



Εθνικό Μετσόβιο Πολυτεχνείο

Διπλωματική Εργασία

**Έλεγχος βιομηχανικού ρομπότ με βάση ανοιχτό
λογισμικό ψηφιακής καθοδήγησης
εργαλειομηχανών**

Συγγραφέας:
Δ. Τσούμπας

Επιβλέπων:
Γ.Χ. Βοσνιάκος

Ιούλιος 2015

Θα ήθελα να ευχαριστήσω όλους όσους βοήθησαν για την αποπεράτωση της παρούσας διπλωματικής και ιδιαίτερα τον καθηγητή Γ.Χ Βοσνιακό για την εμπιστοσύνη και την βοήθεια. Επίσης ένα μεγάλο ευχαριστώ στην οικογένεια μου για την υπομονή και στήριξη τους.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1.	Εισαγωγή.....	7
1.1.	Διατύπωση του προβλήματος.....	7
1.2.	Σημασία επίλυσης και πιθανές εφαρμογές	8
1.3.	Σχετική έρευνα.....	8
1.4.	Σύνοψη διπλωματικής εργασίας.....	10
ΚΕΦΑΛΑΙΟ 2.	Ρομποτική	11
2.1.	Εισαγωγή στο ρομποτικό βραχίονα	11
2.1.1.	Είδη Αρθρώσεων.	14
2.1.2.	Τύποι Ρομποτικών Βραχιόνων και Χώρος Εργασίας.....	16
2.1.3.	Κινηματική Θεωρία.	23
2.1.4.	Μέθοδοι Προγραμματισμού Ρομπότ.....	38
2.1.5.	Τρόποι Μετάβασης από Σημείο σε Σημείο.	39
2.1.6.	Ανάγκες και Περιορισμοί κατά την Κίνηση.	42
2.2.	Ο ρομποτικός βραχίονας MITSUBISHI RM-501	44
2.2.1.	Μηχανικά Μέρη	45
2.2.2.	Κινηματική θεωρία	60
ΚΕΦΑΛΑΙΟ 3.	Ηλεκτρονική και Έλεγχος	68
3.1.	Εισαγωγή στα ηλεκτρονικά	68
3.1.1.	Βασικές έννοιες	69
3.1.2.	Συνήθεις Διατάξεις Ηλεκτρονικών	72
3.2.	Ηλεκτρονικά ρομποτικού βραχίονα RM-501	82
3.2.1.	Ηλεκτρολογικά σχέδια.....	83
3.2.2.	Οι Πλακέτες 724 και 732	84
3.2.3.	Η Πλακέτα 727 (Αρπάγη).....	93
3.2.4.	Η Πλακέτα Επικοινωνίας με τον Ρομποτικό βραχίονα	95
3.2.5.	Η Πλακέτα επικοινωνίας με Η/Υ	96

ΚΕΦΑΛΑΙΟ 4.	Λογισμικό αριθμητικού ελέγχου	100
4.1.	Η δομή του LinuxCNC	100
4.1.1.	Ιστορία	100
4.1.2.	Τυπική διάταξη LinuxCNC.....	101
4.1.3.	Τι είναι το LinuxCNC	102
4.1.4.	Πώς λειτουργεί το LinuxCNC	102
4.2.	Διαχείριση του ρομποτικού βραχίονα με το πρόγραμμα LinuxCNC	107
4.2.1.	Το αρχείο rm501.c	107
4.2.2.	Το αρχείο HAL.....	108
4.2.3.	Το αρχείο INI.....	118
ΚΕΦΑΛΑΙΟ 5.	Αποτελέσματα και Συμπεράσματα	122
5.1.	Διαδικασία δοκιμών	122
5.2.	Αποτελέσματα	124
5.3.	Συμπεράσματα	131
5.4.	Προτάσεις για μελλοντική μελέτη.....	131
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		133
ΠΑΡΑΡΤΗΜΑ Α	Κατάλογος ηλεκτρονικών RM-501	134
ΠΑΡΑΡΤΗΜΑ Β	ΑΡΧΕΙΑ κώδικα	142

ΚΕΦΑΛΑΙΟ 1. Εισαγωγή

Η ραγδαία ανάπτυξη της βιομηχανίας του πυριτίου στις αρχές τις δεκαετίας του 70, σε συνδυασμό με την απαίτηση των μεγάλων βιομηχανιών για ταχύτητα και ακρίβεια στην γραμμή παραγωγής, συνέβαλαν στην δημιουργία όλο και περισσότερων ρομπότ. Ως ρομπότ μπορούν να θεωρηθούν βάσει ενός γενικού ορισμού, οι μηχανές αυτές οι οποίες είναι σε θέση να πραγματοποιούν αυτόματα ή με καθοδήγηση μια προγραμματισμένη και αναπρογραμματιζόμενη ακολουθία εργασιών. Μια από τις μεγαλύτερες κατηγορίες ρομπότ στην βιομηχανία είναι οι ρομποτικοί βραχίονες.

Οι ρομποτικοί βραχίονες έχουν ευρεία εφαρμογή και ως εργαλειομηχανές μιας και παρουσιάζουν το πλεονέκτημα της επιδεξιότητας στον χώρο εργασίας. Μπορούν δηλαδή να προσεγγίσουν ένα σημείο από τόσες προοπτικές όσες τους επιτρέπουν οι βαθμοί ελευθέριας τους.

Το μεγαλύτερο μειονέκτημα είναι ότι οι κατασκευάστριες εταιρίες ακολουθούν πρωτόκολλα κλειστής αρχιτεκτονικής ελέγχου (closed architecture control). Έτσι, κάθε βραχίονας παρουσιάζει την δική του εξειδικευμένη γλώσσα προγραμματισμού, και τα ηλεκτρονικά και ηλεκτρομηχανικά μέρη που φέρει συνήθως δεν είναι συμβατά με άλλων εταιριών. Αυτή η τακτική δημιουργεί περισσότερα μειονεκτήματα στην περίπτωση διατήρησης παλαιότερων ρομπότ σε περιβάλλον παραγωγής, όπως:

- Απαιτείται περαιτέρω εξειδίκευση για τον χειρισμό του
- Δεν μπορεί να συνεργαστεί εύκολα με εργαλειομηχανές άλλων εταιριών (ενδεχομένως σε ένα έξυπνο κύτταρο παραγωγής)
- Οποιαδήποτε παρέμβαση εξαρτάται από τον κατασκευαστή
- Μεγάλο κόστος συντήρησης
- Δυσκολία συμβατότητας με νεότερη τεχνολογία

Για τους παραπάνω λόγους συχνά είναι προτιμότερη η αγορά ενός νέου ρομποτικού βραχίονα

1.1. Διατύπωση του προβλήματος

Όπως αναφέρθηκε παραπάνω τα μειονεκτήματα που παρουσιάζουν οι ρομποτικοί βραχίονες λόγω του πρωτοκόλλου κλειστής αρχιτεκτονικής ελέγχου έχει κάνει πιο έντονη την προσπάθεια ελέγχου αυτών με πιο ευέλικτους τρόπους. Ένας από αυτούς τους τρόπους είναι ο έλεγχος μέσω G κώδικα. Επίσης ένας σημαντικός αριθμός βραχιόνων αδρανοποιούνται λόγω των παρωχημένων λογισμικών ελέγχου τους.

Στην παρούσα διπλωματική εργασία αναφέρεται η προσπάθεια ελέγχου του ρομποτικού βραχίονα MITSUBISHI MOVEMASTER II RM-501 με ανοιχτό λογισμικό ψηφιακής καθοδήγησης. Σκοπός είναι η υλοποίηση μιας χαμηλού κόστους μεθοδολογίας για τον εκσυγχρονισμό και την λειτουργία του συγκεκριμένου βραχίονα με ανάλογο τρόπο μιας εργαλειομηχανής CNC. Καλείται, δηλαδή, να λειτουργήσει μέσω εντολών G κώδικα με την βοήθεια του λογισμικού LinuxCNC.

Ο ρομποτικός βραχίονας MITSUBISHI MOVEMASTER II RM-501 χρονολογείται από το 1986. Έτσι, πέρα από την απουσία οποιουδήποτε εγχειρίδιου της αρχιτεκτονικής του, ένα ακόμη στοίχημα ήταν η προσπάθεια σύνδεσης της τεχνολογίας του τότε με το σήμερα.

1.2. Σημασία επίλυσης και πιθανές εφαρμογές

Η σημασία της ολοκλήρωσης αυτού του εγχειρήματος είναι πολλαπλή. Αρχικά λόγω του μικρού όγκου του καθώς και του χαμηλού κόστους, το συγκεκριμένο ρομπότ μπορεί να αποτελέσει ένα σημαντικό εργαλείο για εκπαίδευση και πειραματισμό. Επίσης η χρήση G κώδικα του επιτρέπει να συνεργάζεται και με άλλες μηχανές δυναμικά, χωρίς δηλαδή να χρειαστεί να αποθηκευτεί μια νέα αλληλουχία κινήσεων προτού εκτελεστεί κάποια ενέργεια. Μερικά παραδείγματα χρήσης του ρομποτικού βραχίονα μετά τον έλεγχο με σύγχρονο λογισμικό είναι

- Εκμάθηση G κώδικα σε απαιτητικό χώρο εργασίας
- Εφαρμογή εξελιγμένων κινηματικών μοντέλων
- Αποφυγή εμποδίων σε δυναμικό χώρο εργασίας
- Πρωτογενής έλεγχος κώδικα G πριν εφαρμοστεί σε βραχίονες μεγαλύτερων διαστάσεων
- Εφαρμογές συναρμολόγησης
- Βοηθητική παρουσία σε περιβάλλοντα ακατάλληλα για τον άνθρωπο. Θα μπορούσε, για παράδειγμα, να τροφοδοτεί με τήγμα ένα καλούπι χύτευσης.

1.3. Σχετική έρευνα

Μια προσπάθεια ελέγχου ενός ρομποτικού βραχίονα συνδυάζει πολλά επιστημονικά πεδία. Η ηλεκτρονική, ο προγραμματισμός και η μηχανολογία είναι μερικά από αυτά τα πεδία. Έτσι ήταν αναμενόμενο να έχουν υπάρξει παλαιότερα ανάλογα εγχειρήματα. Παρακάτω αναφέρονται συγγενείς εργασίες με την παρούσα.

Οι Ζίβανονιέ et al. [1, 2] παρουσίασαν την επιτυχημένη προσπάθεια διαμόρφωσης μιας επιτραπέζιας εργαλειομηχανής παράλληλης κινηματικής τριών αξόνων (3-dof parallel kinematic machine). Αυτού του τύπου εργαλειομηχανές βρίσκονται σε πειραματικό στάδιο με καλές προοπτικές μιας και δείχνουν να έχουν μεγαλύτερη ακαμψία και μεγαλύτερη αναλογία ισχύος / βάρους. Όμως η πολύπλοκη γεωμετρία αυτών των τύπων μηχανών κάνει την ρύθμιση τους μια δύσκολη διαδικασία. Για τον λόγο αυτό οι παραπάνω ερευνητές αποφάσισαν να ακολουθήσουν μια μεθοδολογία ιεραρχικής ανάλυσης (IDEFO) που συνδυάζει γραφικά και κείμενα τα οποία παρουσιάζονται σε ένα οργανωμένο περιβάλλον. Η επιλογή της διαμόρφωσης της εργαλειομηχανής κάτω από το ανοιχτό λογισμικό περιβάλλον LinuxCNC συνετέλεσε θετικά στο έργο τους.

Οι Li et al. [3] περιγράφουν την αντίστροφη κινηματική θεωρία μιας εργαλειομηχανής παράλληλης κινηματικής έξι βαθμών ελευθερίας (6-dof parallel kinematic machine) βασιζόμενοι στις αρχές της πλατφόρμας Stewart. Και εδώ όπως και στην παραπάνω έρευνα η επαλήθευση και ο έλεγχος της κινηματικής θεωρίας έγινε προσομοιώνοντας την μηχανή με την βοήθεια του ανοιχτού λογισμικού LinuxCNC.

Οι Ferenc et al. [4] αναφέρονται στα πλεονεκτήματα που παρουσιάζει ο έλεγχος ανοιχτής αρχιτεκτονικής πραγματικού χρόνου (open architecture real-time control) στους αρθρωτούς ρομποτικούς βραχίονες. Σκοπός του άρθρου είναι να εξηγήσει το σκεπτικό του ελέγχου ανοιχτής αρχιτεκτονικής (OAC) και να παρουσιάσει τις κύριες διεθνείς δραστηριότητες που προτείνουν παρόμοιες προσεγγίσεις.

Οι Tajti et al. [5] αναπτύσσουν τρία διαφορετικά πειραματικά συστήματα με διάφορες εργαλειομηχανές. Σκοπός είναι να χρησιμοποιηθεί ανοιχτό ενδιάμεσο λογισμικό RT-middleware για την επαναχρησιμοποίηση παλαιών ρομπότ τα οποία είναι λειτουργικά αλλά φέρουν παρωχημένα ηλεκτρονικά. Στο άρθρο επίσης περιγράφεται ο συνδυασμός της αξιοπιστίας του λογισμικού LinuxCNC με την ευελιξία του λογισμικού RT-middleware. Τέλος οι ερευνητές παρουσιάζουν την διάταξη για τον έλεγχο ρομποτικού βραχίονα τύπου SCARA με τα δυο προαναφερθέντα λογισμικά.

Η ίδια ερευνητική ομάδα σε άλλο άρθρο [6] διαβλέπει την ανάγκη για άμεση επικοινωνία των συστημάτων ενδο-επιχειρησιακού σχεδιασμού (Enterprise Resource Systems :ERP) με τις μηχανές μιας γραμμής παραγωγής και προτείνει την κατασκευή ηλεκτρονικής πλακέτας παράλληλης επικοινωνίας (PCI). Σκοπός αυτής της πλακέτας είναι να δέχεται πληροφορίες από την εργαλειομηχανή (όπως άμεση διακοπή, συντήρηση, σφάλμα μηχανής κ.α.) και να ενημερώνει τα συστήματα ενδο-επιχειρησιακού σχεδιασμού. Η πλακέτα φέρει ηλεκτρονικά τέτοια ώστε να δίνεται η δυνατότητα άμεσης χρήσης της μηχανής (plug&play) και το λογισμικό που αναλαμβάνει την διαχείριση αυτών των μηνυμάτων είναι το LinuxCNC εφόσον πρόκειται για ανοιχτό λογισμικό με δυνατότητα προσαρμογής για την διαχείριση οποιουδήποτε ανάλογου σήματος.

Οι Bomfim et al. [7] αναγνωρίζουν τον αυξανόμενο αριθμό ρομπότ που τίθενται σε αχρηστία λόγω του ξεπερασμένου εξοπλισμού τους, και αποφασίζουν να διαμορφώσουν έναν παλιό ρομποτικό βραχίονα της δεκαετίας του 70 (ASEA IRB6-S2) χρησιμοποιώντας τα προγράμματα MATLAB και Mach3. Πρόκειται για ένα εγχείρημα χαμηλού κόστους.

Οι Korácsi et al. [8] παρουσιάζουν τον συγχρονισμό ενός αρθρωτού ρομποτικού βραχίονα Mitsubishi RM-501, με μια επιτραπέζια φρέζα. Για την άμεση συνεργασία των δυο εργαλειομηχανών χρησιμοποιήθηκαν εικονικά μοντέλα με την βοήθεια του λογισμικού VirCa, ενώ ο χειρισμός τους πραγματοποιήθηκε με το ανοιχτό λογισμικό LinuxCNC.

Οι Staroveški et al. [9] κάνουν μια εκτεταμένη ανάλυση της δομής του λογισμικού LinuxCNC. Χρησιμοποιούν γραφήματα για την κατανόηση των επιμέρους επιπέδων και παρουσιάζουν τον έλεγχο μιας πειραματικής επιτραπέζιας φρέζας τριών βαθμών ελευθερίας με το λογισμικό αυτό.

Οι Crneković και Zorc [10] παρουσιάζουν την προσπάθεια εκσυγχρονισμού του αρθρωτού ρομποτικού βραχίονα Mitsubishi RM-501 παρεμβάλλοντας αναμεσα στην μονάδα διαχείρισης του ρομπότ και τον υπολογιστή μια συσκευή κινηματικού ελέγχου έτσι ώστε να μπορεί να επικοινωνεί με νεότερης γενιάς υπολογιστικά συστήματα.

Οι Erlic et al. [11] στην προσπάθεια τους να μελετήσουν πιο εξελιγμένα μοντέλα κινηματικής, βρήκαν δύσκολο το χειρισμό των αρθρωτών ρομποτικών βραχιόνων Mitsubishi RM-501 και PUMA προχωρούν σε τροποποίηση των δυο αυτών ρομπότ. Η παρέμβαση που έγινε για το Mitsubishi RM-501 ήταν η

αντικατάσταση της μονάδας διαχείρισης από διαφορετικό hardware ενώ για το ρομπότ PUMA χρησιμοποιήθηκαν οι drivers MARK II.

1.4. Σύνοψη διπλωματικής εργασίας

Όπως αναφέρθηκε η παρούσα διπλωματική εργασία παρουσιάζει την προσπάθεια ελέγχου του ρομποτικού βραχίονα MITSUBISHI MOVEMASTER II RM-501 με το ανοιχτό λογισμικό ψηφιακής καθοδήγησης LinuxCNC. Το πεδίο που ασχολείται με παρόμοια εγχειρήματα καλείται μηχανοτρονική (mechatronics) και αποτελεί συνδυασμό των πεδίων της μηχανολογίας, της ηλεκτρονικής και του προγραμματισμού. Στην παρούσα διπλωματική κάθε ένα από αυτά τα πεδία αποτελεί ένα ανεξάρτητο κεφάλαιο με πρώτο μέρος την απαιτούμενη εισαγωγική γνώση, η οποία στη συνέχεια εφαρμόζεται στο δεύτερο μέρος για το συγκεκριμένο ρομποτικό βραχίονα .

Αναλυτικότερα, τα επιμέρους κεφάλαια περιλαμβάνουν τα εξής:

Κεφάλαιο 2 - Ρομποτική: το πρώτο μέρος του κεφαλαίου αναφέρεται στην μηχανική των ρομποτικών βραχιόνων και πως κατηγοριοποιούνται βάσει αυτής. Έπειτα, αναπτύσσονται οι γενικές θεωρίες ευθείας και αντίστροφης κινηματικής. Στο δεύτερο μέρος παρουσιάζονται τα μηχανικά μέρη του ρομποτικού βραχίονα RM-501, οι επενεργητές και τα γεωμετρικά χαρακτηριστικά του. Τέλος, αναπτύσσονται η κινηματική και υπολογίζεται ο όγκος εργασίας του συγκεκριμένου βραχίονα.

Κεφάλαιο 3 - Ηλεκτρονική: το πρώτο μέρος του κεφαλαίου κάνει μια γενική επισκόπηση βασικών αρχών της ηλεκτρονικής και εστιάζει σε διατάξεις που θα φανούν χρήσιμες. Το δεύτερο μέρος του κεφαλαίου επικεντρώνεται στην κατανόηση και την αποσαφήνιση των ηλεκτρονικών που φέρει ο ρομποτικός βραχίονας RM-501, καθώς και στην τεχνική που χρησιμοποιήθηκε για τη σύνδεση αυτού με τον ηλεκτρονικό υπολογιστή.

Κεφάλαιο 4 – Λογισμικό ελέγχου: Όπως και στα προηγούμενα δυο κεφάλαια έτσι και εδώ στο πρώτο μέρος γίνεται αναφορά στην δομή και την λειτουργία του λογισμικού που χρησιμοποιείται. Στο δεύτερο μέρος αναπτύσσεται η κατάλληλη διαμόρφωση/παραμετροποίηση του λογισμικού για την ορθή επικοινωνία με το ρομποτικό βραχίονα RM-501.

Κεφάλαιο 5 – Αποτελέσματα & Συμπεράσματα : Παρουσιάζονται ενδεικτικά αποτελέσματα πειραματισμού με απλές εντολές G κώδικα για τον έλεγχο της ορθής επικοινωνίας του ρομποτικού βραχίονα με τον ηλεκτρονικό υπολογιστή. Από τα αποτελέσματα καθώς και από την διαδικασία διαμόρφωσης απορρέουν συμπεράσματα και ιδέες περαιτέρω ανάπτυξης για το μέλλον.

ΚΕΦΑΛΑΙΟ 2. Ρομποτική

2.1. Εισαγωγή στο ρομποτικό βραχίονα

Ο όρος ρομπότ (robot) προέρχεται από την Τσεχοσλοβακία και σημαίνει «αναγκαστική εργασία». Ο όρος βιομηχανικό ρομπότ (industrial robot) καθιερώθηκε το 1954 από τον G. C. Devol (ΗΠΑ). Ο Devol περιέγραψε πως μπορεί να κατασκευαστεί ένα ελεγχόμενο μηχανικό χέρι το οποίο μπορεί να εκτελεί διάφορες εργασίες στη βιομηχανία. Το πρώτο βιομηχανικό ρομπότ κατασκευάστηκε και τέθηκε σε λειτουργία το 1961 από την εταιρεία Unimation. Έκτοτε τέθηκαν σε λειτουργία χιλιάδες ρομπότ στην Αμερική, την Ιαπωνία και την Ευρώπη.

Τα βιομηχανικά ρομπότ δεν έχουν τη μορφή ή τη συμπεριφορά τού ανθρώπου, αλλά μπορούν να εκτελούν εργασίες που κάνει ο άνθρωπος και ουσιαστικά αποτελούν μηχανές που δίνουν στη βιομηχανία μεγαλύτερη ευελιξία από άλλες μορφές αυτοματισμού.

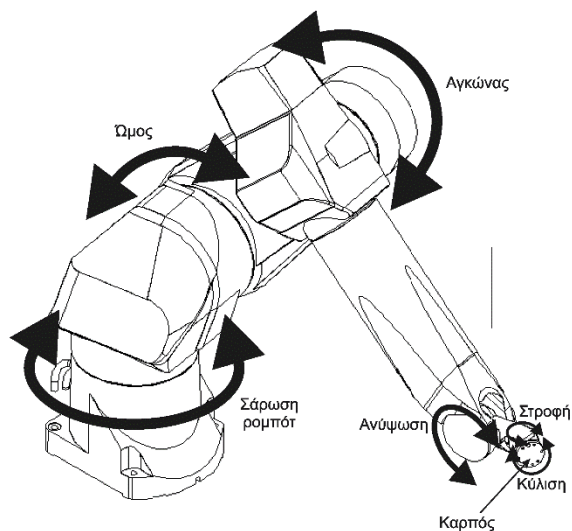
Σήμερα οι ηλεκτρονικοί υπολογιστές αποτελούν ένα ουσιαστικό μέρος τού βιομηχανικού αυτοματισμού. Κατευθύνουν γραμμές παραγωγής και ελέγχουν συστήματα κατασκευής (π.χ. εργαλειομηχανές, συγκολλητές, κοπτικές διατάξεις Laser κ.α.). Τα βιομηχανικά ρομπότ χρησιμοποιούν υπολογιστές σαν μια βασική συνιστώσα τού ελέγχου τους, αλλά οι αυξημένες δυνατότητές τους τα αναβαθμίζουν σε εξελιγμένα συστήματα αυτοματισμού, που μπορούν να εκτελούν ποικίλες εργασίες στα βιομηχανικά συστήματα, και γενικά συμμετέχουν στον πλήρη αυτοματισμό των εργασιών.

Περιγραφή.

Η τυπική μορφή ενός βιομηχανικού ρομπότ φαίνεται στο σχήμα 2.1. Ένα βιομηχανικό ρομπότ μιμείται το ανθρώπινο χέρι. Επειδή τα βιομηχανικά ρομπότ εκτελούν διάφορους χειρισμούς, ονομάζονται και μηχανικοί χειριστές ή απλά χειριστές (manipulators). Πολλοί επιστήμονες τα ονομάζουν και μηχανικά χέρια (mechanical arms).

Ένα ρομπότ έχει αρθρώσεις ανάλογες αυτών του ανθρώπινου χεριού όπως: «ώμο», «αγκώνα», και «καρπό». Ο καρπός μπορεί να περιστρέφεται γύρω από τρεις άξονες. Το διαμήκη που έχει τη διεύθυνση των δακτύλων (η κίνηση ονομάζεται κύλιση: roll), τον εγκάρσιο που βρίσκεται στο επίπεδο της παλάμης και είναι κάθετος προς τα δάκτυλα (κίνηση ανύψωσης: pitch) και τον κάθετο άξονα που είναι κάθετος προς τους άλλους δύο, δηλαδή προς το επίπεδο της παλάμης (κίνηση στροφής: yaw).

Οι αρθρώσεις οδηγούνται από κινητήρες ή γενικότερα από επενεργητές (actuators) που μπορεί να είναι ηλεκτρικοί, πνευματικοί ή υδραυλικοί. Οι επενεργητές (κινητήρες, έμβολα κ.ά.) δίνουν στο ρομπότ πολύ μεγαλύτερη ισχύ από αυτή που μπορεί να αναπτύξει ο άνθρωπος.



ΣΧΗΜΑ 2.1 Τυπική μορφή βιομηχανικού ρομπότ.

Ο υπολογιστής ενός σύγχρονου ρομπότ, περιέχει ένα πρόγραμμα ελέγχου (control program) και ένα πρόγραμμα εργασίας (task program). Το πρόγραμμα ελέγχου δίνεται από τον κατασκευαστή και ρυθμίζει την κίνηση κάθε μιας άρθρωσης του ρομπότ. Το πρόγραμμα εργασίας δίνεται από το χρήστη και καθορίζει τις κινήσεις που χρειάζεται για να εκτελεσθεί κάθε φορά η επιθυμητή εργασία. Ένα πρόγραμμα εργασίας μπορεί να παραχθεί είτε περνώντας το ρομπότ από τις θέσεις που απαιτεί η συγκεκριμένη εργασία είτε χρησιμοποιώντας κατάλληλες γλώσσες προγραμματισμού για να περιγραφεί η τροχιά και οι κινήσεις. Όταν χρησιμοποιείται μια γλώσσα προγραμματισμού, το ρομπότ περιέχει έναν κατάλληλο επεξεργαστή που μεταφράζει το πρόγραμμα εργασίας και παρέχει τα δεδομένα που χρειάζεται το πρόγραμμα ελέγχου για να καθοδηγήσει το ρομπότ στις επιθυμητές κινήσεις. Το πρόγραμμα ελέγχου χρησιμοποιεί το πρόγραμμα εργασίας σαν είσοδο και συνεπώς για κάθε εργασία ο χρήστης πρέπει να γράψει ένα διαφορετικό πρόγραμμα εργασίας.

Χρήση.

Τα ρομπότ είναι ιδανικά για εργασίες μη ευχάριστες ή επικίνδυνες, όπως είναι λ.χ. το βάψιμο με ψεκασμό ή ο χειρισμός ραδιενεργών ουσιών. Επίσης, τα ρομπότ μπορούν να εργάζονται σε ανιαρές και επαναλαμβανόμενες εργασίες που είναι όμως απλές και συνήθως χρειάζονται ανειδίκευτους εργάτες. Εργασίες που απαιτούν την ανθρώπινη ευφυΐα και δημιουργικότητα δεν μπορούν να εκτελεστούν από τα σημερινά ρομπότ.

Γενικά, τα βιομηχανικά ρομπότ μπορούμε να πούμε ότι παρέχουν:

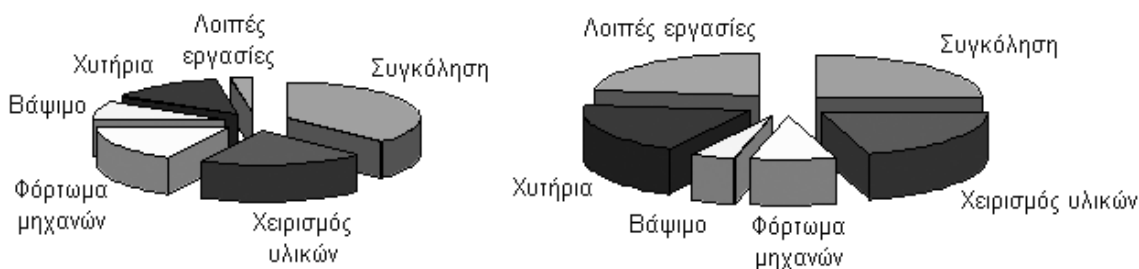
- Υποκατάστατο ανθρώπινου χεριού
- Ευελιξία (εύκολες μεταβολές προϊόντος, πολλά προϊόντα)
- Ευφυΐα και ειδικούς αισθητήρες (π.χ. όραση)
- Ταχύτητα και ακρίβεια

- Μεγαλύτερη δύναμη από του ανθρώπου
- Γρήγορη ανάπτυξη - ολοκλήρωση
- Οικονομικές λύσεις για μικρή παραγωγή
- Σταθερή ποιότητα προϊόντος και παραγωγικότητα

Οι κυριότερες βιομηχανικές εφαρμογές των ρομπότ είναι οι ακόλουθες:

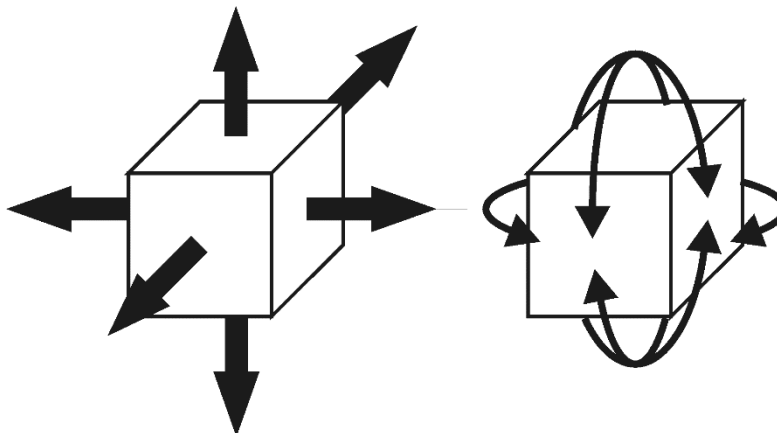
- Φόρτωμα και ξεφόρτωμα μηχανών
- Σημειακή συγκόλληση
- Συγκόλληση τόξου
- Βάψιμο ψεκασμού
- Συναρμολόγηση εξαρτημάτων
- Επίβλεψη
- Κατεργασίες (διάτρηση, κοπή κ.α.)
- Έγχυση μετάλλων σε καλούπια χύτευσης

Το σχήμα 2.2 δείχνει την κατανομή χρησιμοποίησης των Ρομπότ σε διάφορες βιομηχανικές εφαρμογές στην Αμερική και την Ιαπωνία τη δεκαετία του '90. Πρέπει να σημειωθεί εδώ ότι για κάθε μια κατηγορία εφαρμογών απαιτείται διαφορετική ακρίβεια θέσης και διαφορετικά χαρακτηριστικά του συστήματος ελέγχου. Έτσι, λ.χ. ένα ρομπότ βαφής ή φορτώματος-ξεφορτώματος δεν μπορεί να αντικαταστήσει ένα ρομπότ συναρμολόγησης, επειδή έχουν διαφορετικά επίπεδα ακρίβειας.



ΣΧΗΜΑ 2.2 Κατανομή απασχόλησης ρομπότ στη βιομηχανία (α) Η.Π.Α. και (β) Ιαπωνία.

Στο χώρο των τριών διαστάσεων η σχετική κίνηση που μπορεί να πραγματοποιήσει ένα συμπαγές σώμα, ως προς ένα άλλο που θεωρείται ακίνητο, χωρίζεται σε 6 βασικές επιμέρους κινήσεις. 3 μεταφορικές κινήσεις στις διευθύνσεις των αξόνων του τρισσορθογωνίου συστήματος συντεταγμένων, και 3 περιστροφικές κινήσεις γύρω από αυτούς.



ΣΧΗΜΑ 2.3 Οι 6 βαθμοί ελευθερίας ενός συμπαγούς σώματος
(α) μεταφορικοί και (β) περιστροφικοί.

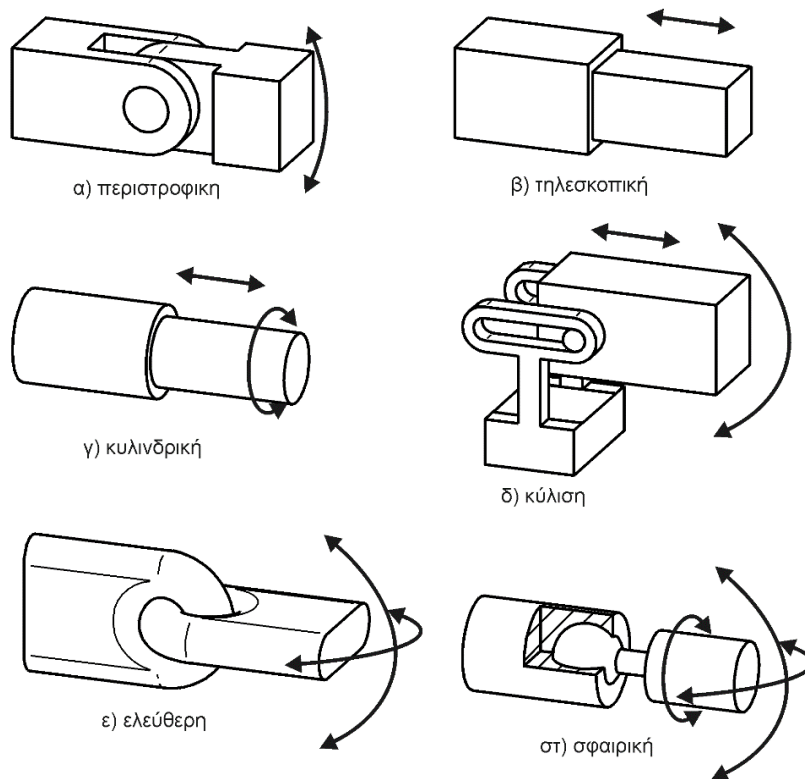
Άρθρωση (joint) ονομάζεται η κατάλληλα διαμορφωμένη διάταξη που συνδέει δύο σώματα μεταξύ τους, περιορίζοντας ένα ή, συχνότερα, περισσότερους από τους μεταξύ τους βαθμούς ελευθερίας, δηλαδή εμποδίζει την πραγματοποίηση μιας ή περισσότερων από τις παραπάνω 6 δυνατότητες κίνησης, μειώνοντας την ελευθερία κίνησης του ενός ως προς το άλλο σώμα.

2.1.1.Είδη Αρθρώσεων.

Οι αρθρώσεις που συναντώνται συχνότερα σε μηχανισμούς είναι οι εξής:

- Η περιστροφική άρθρωση δίνει ένα βαθμό ελευθερίας αφήνοντας το σώμα να περιστραφεί σε ένα επίπεδο και αποκόπτει κάθε άλλη δυνατότητα κίνησης. Επειδή είναι η συνηθέστερη μορφή σύνδεσης δύο σωμάτων, έχει καθιερωθεί να καλείται και απλά «άρθρωση».
- Η τηλεσκοπική άρθρωση δίνει και αυτή ένα βαθμό ελευθερίας αφήνοντας το σώμα να μετατοπίζεται στη διεύθυνση ενός από τους άξονες, και αποκόπτει κάθε άλλη δυνατότητα κίνησης.
- Η κυλινδρική άρθρωση δίνει δύο βαθμούς ελευθερίας αφήνοντας μία μεταφορική κίνηση στη διεύθυνση ενός άξονα και μία περιστροφική γύρω από τον άξονα αυτό.
- Η άρθρωση κύλισης δίνει και αυτή δύο βαθμούς ελευθερίας, δηλαδή μία μεταφορική και μία περιστροφική κίνηση, άλλα σε αυτήν την περίπτωση ο άξονας της περιστροφικής κίνησης είναι κάθετος στη διεύθυνση του άξονα της μεταφορικής κίνησης.
- Η ελεύθερη άρθρωση δίνει δύο βαθμούς ελευθερίας αφήνοντας δύο περιστροφικές κινήσεις και εμποδίζοντας όλες τις υπόλοιπες.
- Η σφαιρική άρθρωση δίνει τρεις βαθμούς ελευθερίας αφήνοντας και τις τρεις περιστροφικές κινήσεις ελεύθερες και εμποδίζοντας όλες τις μεταφορικές.

Σημείωση: Δεν είναι πάντα εφικτό με μία άρθρωση να πετύχουμε τις επιθυμητές κινήσεις, έτσι πολλές φορές καταφεύγουμε σε συνδυασμούς με περισσότερα σώματα αρθρωμένα μεταξύ τους.



ΣΧΗΜΑ 2.4 Τυπικές αρθρώσεις.

Αρθρώσεις σε Ρομποτικούς Βραχίονες.

Οι ρομποτικοί βραχίονες, αν και είναι μηχανισμοί με πολύ μεγάλη ευελιξία, στην πλειονότητά τους βασίζονται σε βραχίονες που συνδέονται μεταξύ τους με αρθρώσεις ενός βαθμού ελευθερίας.

Έτσι, οι ρομποτικοί βραχίονες, για να μπορέσουν να επιτύχουν την προσδοκώμενη ευελιξία, σχεδιάζονται με πολλές (συνήθως από τρεις μέχρι έξι) αρθρώσεις, κατάλληλα τοποθετημένες ώστε να επιτρέπουν στους επενεργητές να κινούνται εύκολα.

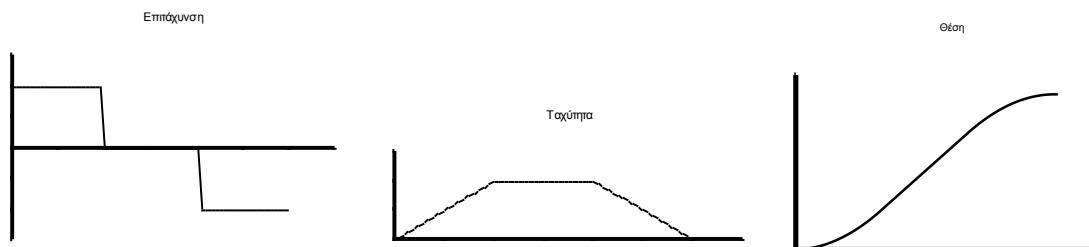
Οι αρθρώσεις αυτές είναι οι:

- Περιστροφική άρθρωση (revolute joint) που επιτρέπει την ελεγχόμενη περιστροφή γύρω από τον άξονά της. Η περιστροφική άρθρωση κυρίως λόγω της μεγάλης ταχύτητάς της είναι η κοινότερη άρθρωση σε ρομποτικούς βραχίονες.
- Γραμμικές αρθρώσεις που χωρίζονται σε ολίσθησης (sliding) και πρισματικές (prismatic) και επιτρέπουν την ελεγχόμενη μετατόπιση προς μία κατεύθυνση. Οι αρθρώσεις αυτές χαρακτηρίζονται για τη μεγάλη ακρίβειά τους και χρησιμοποιούνται κυρίως σε διατάξεις συναρμολόγησης.

Ταχύτητες και Επιταχύνσεις Αρθρώσεων.

Η κίνηση ενός ρομποτικού βραχίονα νοείται ως το άθροισμα των επιμέρους κινήσεων κάθε άρθρωσης με φυσικούς νόμους που διέπει κάθε μία από αυτές.

Κάθε άρθρωση χαρακτηρίζεται από μία οριακή (μέγιστη) ταχύτητα που μπορεί να αποκτήσει και από μία οριακή (μέγιστη) επιτάχυνση ή επιβράδυνση αντίστοιχα που μπορεί να ασκήσει πάνω του ο αντίστοιχος επενεργητής.



ΣΧΗΜΑ 2.5 Τυπικά διαγράμματα επιτάχυνσης, ταχύτητας και θέσης μιας άρθρωσης.

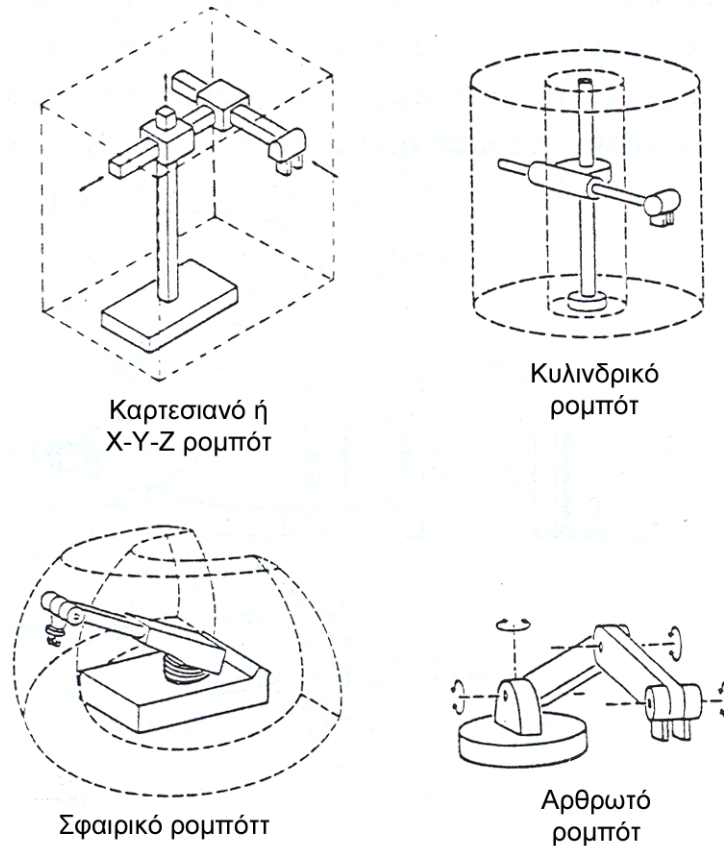
Στην πλειονότητα των ρομποτικών αρθρώσεων η κίνηση επιτυγχάνεται με τη βοήθεια ηλεκτρικών κινητήρων. Όπως φαίνεται και από το σχήμα 2.5 η κίνηση που πραγματοποιεί κάθε άρθρωση είναι ομαλά επιταχυνόμενη ή επιβραδυνόμενη αντίστοιχα.

2.1.2. Τύποι Ρομποτικών Βραχιόνων και Χώρος Εργασίας.

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

Δομικά, τα ρομπότ ταξινομούνται ανάλογα με το σύστημα συντεταγμένων του βραχίονα ως εξής (σχήμα 2.6):

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

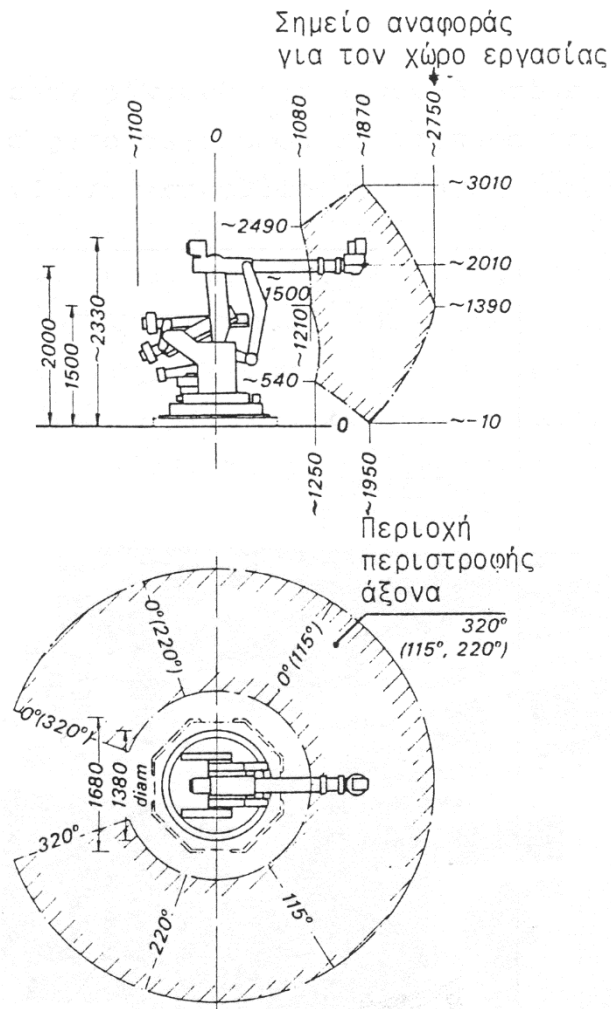


ΣΧΗΜΑ 2.6 Οι τέσσερις βασικοί δομικοί τύποι βιομηχανικού ρομπότ.

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

Ένα από τα βασικά χαρακτηριστικά ενός βιομηχανικού ρομπότ είναι το μέγεθος και το σχήμα του χώρου εργασίας του (work volume ή reach envelope). Το σχήμα του χώρου εργασίας εξαρτάται από τα συστήματα συντεταγμένων των αξόνων του ρομπότ, ενώ το μέγεθος του χώρου εργασίας εξαρτάται από τις διαστάσεις του βραχίονα του ρομπότ. Παράδειγμα του χώρου εργασίας δίνεται στο σχήμα 2.7.

Πρέπει να σημειωθεί ότι με την προσθήκη του τελικού στοιχείου δράσης (εργαλείου) στον καρπό τού ρομποτικού βραχίονα ο χώρος εργασίας μεταβάλλεται ανάλογα με τον τύπο και το μέγεθος του εργαλείου.



ΣΧΗΜΑ 2.7 Χώρος εργασίας τού ρομπότ ΚΥΚΑ IR 601/60.

Παρακάτω δίνουμε τις βασικές, χαρακτηριστικές ιδιότητες για κάθε έναν τύπο ρομπότ. Οι εφαρμογές, στις οποίες χρησιμοποιούνται οι τέσσερις τύποι ρομπότ, συνοψίζονται στο σχήμα 2.1.8.



ΣΧΗΜΑ 2.8 Χρήση των καρτεσιανών, κυλινδρικών, σφαιρικών και αρθρωτών ρομπότ στις διάφορες βιομηχανικές εφαρμογές.

2.1.2.1. Ρομπότ Καρτεσιανών Συντεταγμένων.

Το κυρίως σώμα ενός ρομπότ του τύπου αυτού αποτελείται από τρεις γραμμικούς άξονες όπως φαίνεται στο σχήμα 2.6. Η δομή μπορεί να είναι όμοια με των εργαλειομηχανών (βάση, τραπέζι εργασίας κ.α.) αλλά τότε ο λόγος μεταξύ του χώρου εργασίας, του ρομπότ και του χώρου που καταλαμβάνει είναι μικρότερος. Γενικά, τα χαρακτηριστικά (ηλεκτρονικό υλικό, hardware, πρόγραμμα ελέγχου κ.λπ.) ενός καρτεσιανού ρομπότ είναι όμοια με εκείνα των εργαλειομηχανών αριθμητικού ελέγχου (CNC).

Έτσι, η διακριτική ικανότητα και η επαναληψιμότητα ενός καρτεσιανού ρομπότ μπορεί να είναι πολύ καλή όπως και στις εργαλειομηχανές. Σε πολλά καρτεσιανά ρομπότ η βάση δεν είναι σταθερή αλλά μπορεί να κινείται μέσα σε ορισμένα όρια. Ο καρπός ενός καρτεσιανού ρομπότ μπορεί να ακολουθήσει μια ευθύγραμμη τροχιά, αν κάθε άξονας κινηθεί με σταθερή ταχύτητα. Άλλο πλεονέκτημα των καρτεσιανών ρομπότ είναι η σταθερότητα της διακριτικής ικανότητας θέσης. Αυτό δε συμβαίνει στα μη καρτεσιανά ρομπότ.

Παρά τα πλεονεκτήματα αυτά, τα καρτεσιανά ρομπότ δεν είναι προτιμητέα στη βιομηχανία. Αυτό συμβαίνει γιατί δεν έχουν μηχανική ευελιξία, για παράδειγμα, δεν μπορούν να φθάσουν αντικείμενα που βρίσκονται στο πάτωμα ή δεν είναι ορατά από τη βάση τους. Επίσης, η ταχύτητα λειτουργίας στο οριζόντιο επίπεδο είναι συνήθως μικρότερη από την αντίστοιχη ταχύτητα των ρομπότ που έχουν περιστρεφόμενη βάση. Παραδείγματα καρτεσιανών ρομπότ είναι τα ρομπότ DEA Pragma, Renault Acta Cribier και Mitsubishi Robitus (για συγκολλήσεις σημείων) και το ρομπότ Shin Meiwa για συγκολλήσεις τόξου.

2.1.2.2. Ρομπότ Κυλινδρικών Συντεταγμένων.

Το κυρίως σώμα ενός ρομπότ του τύπου αυτού αποτελείται από έναν οριζόντιο βραχίονα στερεωμένο σε μια κατακόρυφη κολώνα. Η κολώνα είναι με τη σειρά της στερεωμένη πάνω σε μια περιστρεφόμενη βάση (σχήμα 2.6). Ο οριζόντιος βραχίονας κινείται προς τα εμπρός και προς τα πίσω κατά τη διεύθυνση του διαμήκους άξονά του και επίσης ανεβοκατεβαίνει στην κολώνα. Κολώνα και βραχίονας στρέφονται σαν ένα σώμα πάνω στη βάση γύρω από τον κατακόρυφο άξονα.

Η διακριτική ικανότητα ενός κυλινδρικού ρομπότ δεν είναι σταθερή αλλά εξαρτάται από την απόσταση r μεταξύ της κολώνας και του εργαλείου κατά μήκος του οριζόντιου βραχίονα.

Η διακριτική ικανότητα θέσης του κυλινδρικού ρομπότ γύρω από τον άξονα είναι κατά 1-2 τάξεις μεγέθους χειρότερη από την αντίστοιχη των καρτεσιανών ρομπότ ή των εργαλειομηχανών. Τούτο είναι ένα από τα μειονεκτήματα των κυλινδρικών απέναντι στα καρτεσιανά ρομπότ. Τα κυλινδρικά όμως ρομπότ λόγω του περιστρεφόμενου άξονα προσφέρουν μεγαλύτερη ταχύτητα στο άκρο του βραχίονα. Βέβαια, η ταχύτητα αυτή περιορίζεται από το φορτίο στο εργαλείο του ρομπότ και από τη θέση του βραχίονα. Επίσης η δυναμική συμπεριφορά των ρομπότ που έχουν στρεφόμενους άξονες εξαρτάται από τη ροπή αδρανείας του όλου συστήματος ως προς τη βάση, που εξαρτάται από το βάρος που σηκώνει το ρομπότ και από την απόσταση του βάρους αυτού από τον άξονα της βάσης. Από την άλλη, επειδή η ενεργός αυτή ροπή αδρανείας μεταβάλλεται με το χρόνο και με τη θέση, η δυναμική συμπεριφορά του κυλινδρικού (σφαιρικού και αρθρωτού) ρομπότ είναι χειρότερη από εκείνη του καρτεσιανού ρομπότ που δεν έχει στρεφόμενο άξονα.

2.1.2.3. Ρομπότ Σφαιρικών (Πολικών) Συντεταγμένων.

Τα ρομπότ του τύπου αυτού αποτελούνται από μια στρεφόμενη βάση, ένα ανυψούμενο στέλεχος και ένα τηλεσκοπικό βραχίονα που κινείται προς τα μέσα και προς τα έξω, όπως φαίνεται στο σχήμα 2.6.

Το μέγεθος της περιστροφής μετράται από ψηφιακούς κωδικοποιητές που είναι στερεωμένοι στους άξονες. Το βασικό μειονέκτημα των σφαιρικών ρομπότ είναι και πάλι η μικρή διακριτική ικανότητα θέσης των δυο στροφικών αξόνων που μεταβάλλεται με το μήκος του βραχίονα. Τα σφαιρικά ρομπότ, εκτός από το πλεονέκτημα της αυξημένης ταχύτητας κίνησης των στροφικών αξόνων, έχουν και το πλεονέκτημα της αυξημένης ευελιξίας σε σχέση τόσο με τα καρτεσιανά, όσο και με τα κυλινδρικά ρομπότ.

2.1.2.4. Αρθρωτό Ρομπότ.

Τα αρθρωτά ρομπότ αποτελούνται από τρία σταθερά μέλη (συνδέσμους) που ενώνονται με στροφικές αρθρώσεις και είναι τοποθετημένα πάνω σε μια στρεφόμενη βάση (σχήμα 2.6). Η κινηματική διάταξη μοιάζει με εκείνη του ανθρώπινου χεριού. Το εργαλείο (αρπάγη) είναι ανάλογο της παλάμης και προσαρμόζεται στον πήχη (Forearm) μέσω του καρπού. Ο «αγκώνας» συνδέει τον πήχη με τον άνω βραχίονα και ο «ώμος» συνδέει τον άνω βραχίονα με τη βάση. Πολλές φορές στην άρθρωση του ώμου διατίθεται και μια περιστροφική κίνηση σε οριζόντιο επίπεδο όπως φαίνεται στο σχήμα 2.6.

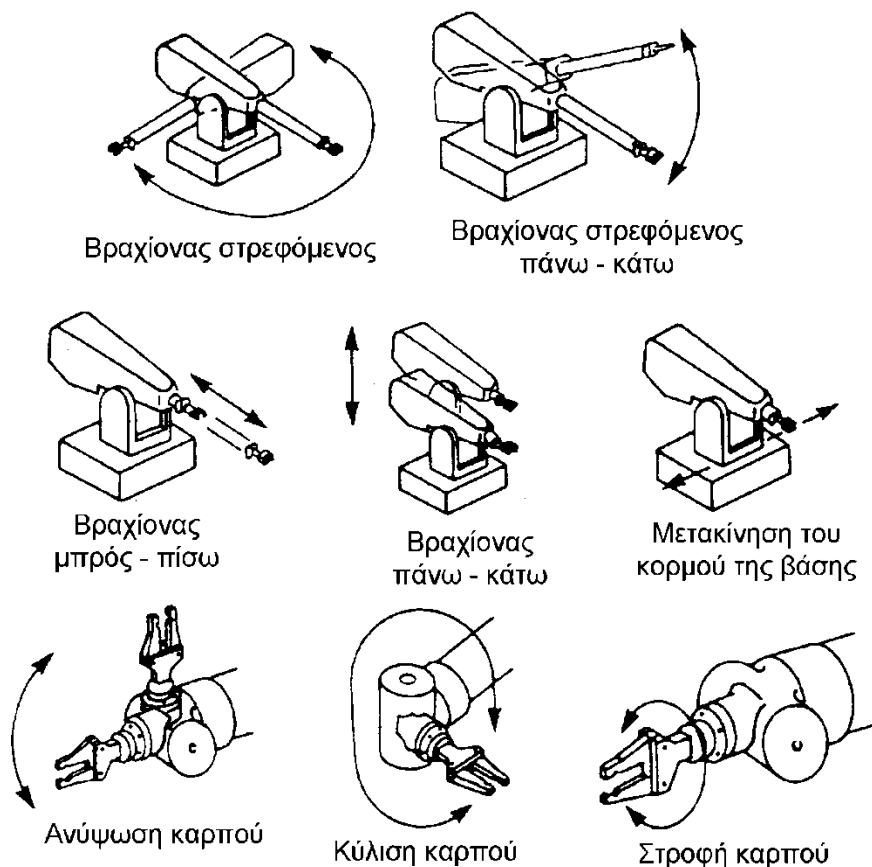
Επειδή το αρθρωτό ρομπότ έχει και τους τρεις άξονες στροφικούς, η διακριτική ικανότητα θέσης εξαρτάται τελείως από τη θέση του βραχίονα. Η ολική ακρίβεια ενός αρθρωτού ρομπότ είναι μικρή γιατί τα σφάλματα των αρθρώσεων συσσωρεύονται στο άκρο του βραχίονα, δηλαδή στη θέση του

καρπού. Τα πλεονεκτήματα των αρθρωτών ρομπότ είναι ότι έχουν την πιο μεγάλη, μηχανική ευελιξία και μπορούν να κινηθούν ταχύτατα.

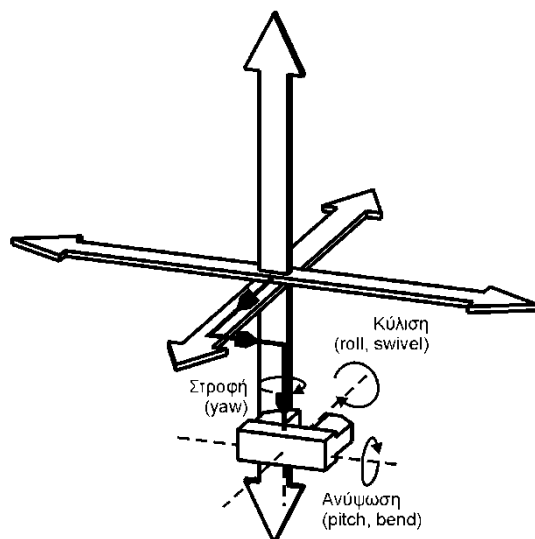
2.1.2.5. Κινήσεις Καρπού και Τελικού Στοιχείου Δράσης.

Το τελικό στοιχείο δράσης (εργαλείο) συνδέεται με το κυρίως σώμα, δηλαδή με το βραχίονα του ρομπότ μέσω του καρπού. Στον καρπό, η κίνηση κύλισης (roll) αντιστοιχεί σε περιστροφή πάνω σε επίπεδο, κάθετο προς το τέλος του βραχίονα, η κίνηση ανύψωσης (pitch, bend) αντιστοιχεί σε περιστροφή πάνω σε ένα κάθετο επίπεδο και η κίνηση στροφής (yaw) αντιστοιχεί σε περιστροφή πάνω σε οριζόντιο επίπεδο.

Οι βασικές κινήσεις του βραχίονα και του καρπού των ρομπότ εικονίζονται στο σχήμα 2.9. Στο σχήμα 2.10 φαίνεται ότι για να φθάσει ένα ρομπότ σε ένα συγκεκριμένο σημείο τού χώρου με συγκεκριμένο προσανατολισμό τού άκρου τού καρπού χρειάζονται έξι βαθμοί ελευθερίας (τρεις για το βραχίονα και τρεις στροφικές για τον καρπό).



ΣΧΗΜΑ 2.9 Βασικές κινήσεις ρομποτικού βραχίονα και καρπού.



ΣΧΗΜΑ 2.10 Οι έξι κινήσεις που χρειάζονται για επιθυμητό προσανατολισμό του καρπού σε επιθυμητό σημείο τού χώρου.

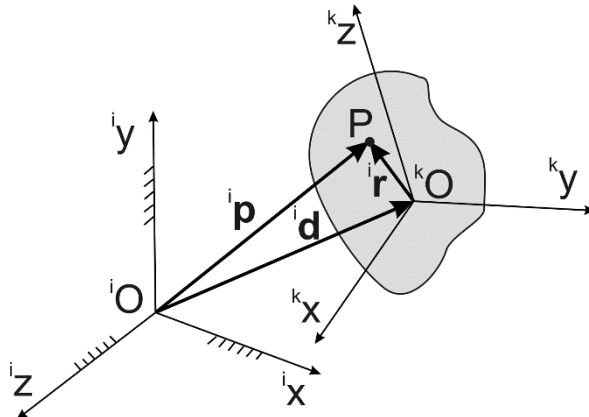
Αν και οι περισσότεροι ρομποτικοί καρποί έχουν τρεις στροφικούς άξονες, υπάρχουν εφαρμογές που χρειάζονται μόνο δύο άξονες του καρπού. Για παράδειγμα, επειδή το πιστόλι συγκόλλησης είναι ένα συμμετρικό εργαλείο, οι περισσότερες εργασίες συγκόλλησης τόξου χρειάζονται καρπό με δύο βαθμούς ελευθερίας. Ο καρπός πρέπει να είναι όσο το δυνατόν πιο ελαφρύς, γιατί έτσι αυξάνεται το μέγιστο επιτρεπτό βάρος που μπορεί να σηκώσει το ρομπότ και μειώνεται η ροπή αδρανείας -πράγμα που βελτιώνει την όλη δυναμική συμπεριφορά τού ρομπότ. Για να μειωθεί το βάρος τού καρπού, οι κινητήρες που οδηγούν τον καρπό τοποθετούνται στη βάση του ρομπότ και η κίνηση μεταδίδεται με αλυσίδες ή στερεούς συνδέσμους.

Γενικά, τα τελικά στοιχεία δράσης διακρίνονται σε δύο κατηγορίες: αρπάγες και εργαλεία για εφαρμογές κατεργασιών (πιστόλια συγκόλλησης, πιστόλια χρωματισμού, τρυπάνια, λειαντές κ.α.)

Οι αρπάγες χρησιμοποιούνται σε εφαρμογές χειρισμού υλικών, φορτώματος μηχανών και συναρμολόγησης. Σε πολλές αρπάγες ο μηχανισμός ενεργοποιείται με ένα πνευματικό έμβολο που κινεί τα δάκτυλα της αρπάγης. Σε εφαρμογές χειρισμού υάλινων προϊόντων ή ανταλλακτικών με πολύ λείες επιφάνειες χρησιμοποιούνται αρπάγες κενού.

2.1.3. Κινηματική Θεωρία.

2.1.3.1. Μετατόπιση Καρτεσιανών Συστημάτων.



ΣΧΗΜΑ 2.11 Χωρόδετο και σωματόδετο καρτεσιανό σύστημα.

Για την περιγραφή της τρισδιάστατης κίνησης μηχανισμών και ρομποτικών συστημάτων, χρησιμοποιείται η διανυσματική ανάλυση. Εισάγονται δύο καρτεσιανά συστήματα συντεταγμένων όπως απεικονίζονται στο σχήμα 2.11:

- Το χωρόδετο καρτεσιανό σύστημα «i» με συντεταγμένες ${}^i(x,y,z)$. Το χωρόδετο καρτεσιανό σύστημα «i» ταυτίζεται με το γεωδαιτικό καρτεσιανό σύστημα, στο οποίο αναφέρονται οι δυνάμεις αδράνειας.
- Το σωματόδετο καρτεσιανό σύστημα «k» με συντεταγμένες ${}^k(x,y,z)$. Το σωματόδετο καρτεσιανό σύστημα «k» είναι απόλυτα συνδεδεμένο με το σώμα και μπορεί να μετατοπισθεί και να στραφεί ως προς το χωρόδετο σύστημα συντεταγμένων «i».

Στο χωρόδετο καρτεσιανό σύστημα «i» (i =inertial) με αρχή το σημείο iO , το σημείο « $P(x,y,z)$ » ενός σώματος περιγράφεται από το διάνυσμα ${}^i\mathbf{p}$:

$${}^i\mathbf{p} = \{ {}^i p_x \quad {}^i p_y \quad {}^i p_z \}^T$$

Το διάνυσμα ${}^i\mathbf{p}$ συντίθεται από τα δύο διανύσματα \mathbf{d} και \mathbf{r} :

$${}^i\mathbf{p} = \begin{Bmatrix} {}^i p_x \\ {}^i p_y \\ {}^i p_z \end{Bmatrix} = {}^i\mathbf{d} + {}^i\mathbf{r} = \begin{Bmatrix} {}^i d_x \\ {}^i d_y \\ {}^i d_z \end{Bmatrix} + \begin{Bmatrix} {}^i r_x \\ {}^i r_y \\ {}^i r_z \end{Bmatrix}$$

και τα δύο αναφέρονται στο χωρόδετο σύστημα «i», όπου:

- ${}^i\mathbf{d}$ είναι το διάνυσμα που περιγράφει τη μετατόπιση της αρχής kO του σωματόδετου καρτεσιανού συστήματος ${}^k(x,y,z)$
- ${}^i\mathbf{r}$ είναι το διάνυσμα που περιγράφει με αρχή το kO σημείο $P(x,y,z)$

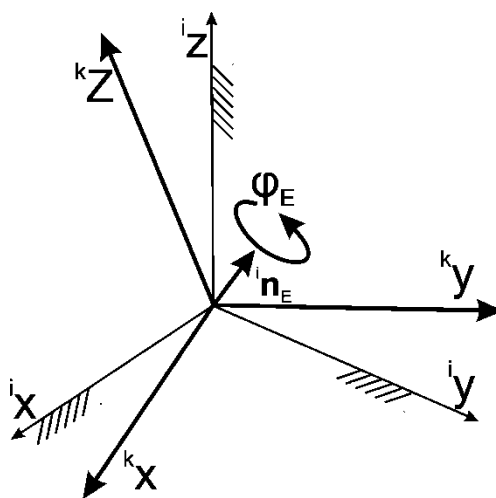
Το σωματόδετο καρτεσιανό σύστημα ${}^k(x,y,z)$ έχει, όπως αναφέρθηκε παραπάνω, αρχή το σημείο kO και διαφοροποιείται από το χωρόδετο λόγω της δυνατότητας τρισδιάστατης περιστροφής του, η οποία μπορεί να περιγραφεί μαθηματικά από το διάνυσμα $\varphi_E(t)$:

$${}^i\boldsymbol{\varphi}_E(t) = \varphi_E(t) \cdot {}^i\mathbf{n}_E$$

όπου $\varphi_E(t)$ είναι το εύρος του διανύσματος περιστροφής και ${}^i\mathbf{n}_E$ το μοναδιαίο διάνυσμα του διανύσματος περιστροφής στο καρτεσιανό σύστημα ${}^i(x,y,z)$. Οι παράμετροι μετασχηματισμού φ_E και ${}^i\mathbf{n}_E$ αποκαλούνται «παράμετροι Euler» (σχήμα 2.12).

Το επί μέρους διάνυσμα ${}^i\mathbf{r}$ (καρτεσιανό σύστημα «i») μετασχηματίζεται στο διάνυσμα ${}^k\mathbf{r}$ (καρτεσιανό σύστημα «k») μέσω ενός 3X3-μητρώου μετασχηματισμού στροφής (R =«rotation»):

$${}^k\mathbf{r} = {}^k\mathbf{R} \cdot {}^i\mathbf{r}$$



ΣΧΗΜΑ 2.12 Παράμετροι περιστροφής Euler φ_E και \mathbf{n}_E

Το μητρώο μετασχηματισμού είναι ίσο με:

$${}^k\mathbf{R} = \cos\theta \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + (1 - \cos\theta) \cdot \begin{bmatrix} \lambda\lambda & \lambda\mu & \lambda\nu \\ \mu\lambda & \mu\mu & \mu\nu \\ \nu\lambda & \nu\mu & \nu\nu \end{bmatrix} + \sin\theta \cdot \begin{bmatrix} 0 & \nu & -\mu \\ -\nu & 0 & \lambda \\ \mu & -\lambda & 0 \end{bmatrix}$$

Οι σταθερές λ , μ και ν είναι ίσες με τα εσωτερικά γινόμενα των μοναδιαίων διανυσμάτων ${}^i\mathbf{n}_x$, ${}^i\mathbf{n}_y$, και ${}^i\mathbf{n}_z$ (στις κατευθύνσεις των αξόνων x , y και z του καρτεσιανού συστήματος «i»):

$${}^i\mathbf{n}_x = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad {}^i\mathbf{n}_y = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad {}^i\mathbf{n}_z = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

με το μοναδιαίο διάνυσμα ${}^i\mathbf{n}_E$:

$$\lambda = {}^i \mathbf{n}_x \cdot {}^i \mathbf{n}_E$$

$$\mu = {}^i \mathbf{n}_y \cdot {}^i \mathbf{n}_E$$

$$\nu = {}^i \mathbf{n}_z \cdot {}^i \mathbf{n}_E$$

Αντιστρέφοντας το μητρώο ${}^k_i \mathbf{R}$ επιτυγχάνεται ο αντίστροφος μετασχηματισμός (από το καρτεσιανό σύστημα «k» στο «i»):

$${}^i \mathbf{r} = {}^i \mathbf{R} \cdot {}^k \mathbf{r} = ({}^k_i \mathbf{R})^{-1} \cdot {}^k \mathbf{r}$$

Όπως μπορεί να αποδειχθεί, το αντίστροφο μητρώο \mathbf{R}^{-1} είναι στη γενική περίπτωση ίσο με το ανάστροφό του \mathbf{R}^T :

$$({}^k_i \mathbf{R})^{-1} = ({}^k_i \mathbf{R})^T = {}^i_k \mathbf{R}$$

Η ιδιότητα αυτή αποκαλείται «ορθογώνια» ιδιότητα του μητρώου μετασχηματισμού και διευκολύνει σημαντικά τη μετάβαση από το ένα καρτεσιανό σύστημα στο άλλο.

Στην περίπτωση που η στροφή $\phi_E(t)$ πραγματοποιείται μόνο γύρω από τον άξονα x και είναι ίση με ϕ , το μητρώο μετασχηματισμού είναι ίσο με:

$${}^k_i \mathbf{R}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

Αντίστοιχα, στην περίπτωση που η στροφή $\phi_E(t)$ πραγματοποιείται μόνο γύρω από τον άξονα y και είναι ίση με θ , το μητρώο μετασχηματισμού είναι ίσο με ($\lambda=\nu=0$):

$${}^k_i \mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

Και τέλος, στην περίπτωση που η στροφή $\phi_E(t)$ πραγματοποιείται μόνο γύρω από τον άξονα z και είναι ίση με ψ , το μητρώο μετασχηματισμού είναι ίσο με:

$${}^k_i \mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Στη γενική περίπτωση το σημείο $P(x,y,z)$ περιγράφεται επομένως στο χωρόδετο σύστημα «i» από το διάνυσμα:

$${}^i \mathbf{p} = {}^i \mathbf{d} + {}^i \mathbf{R} \cdot {}^k \mathbf{r} = {}^i \mathbf{R} \cdot ({}^k \mathbf{d} + {}^k \mathbf{r})$$

στο δε σωματόδετο καρτεσιανό σύστημα «k», από το διάνυσμα:

$${}^k \mathbf{p} = {}^k \mathbf{d} + {}^k \mathbf{R} \cdot {}^i \mathbf{r} = {}^k \mathbf{R} \cdot ({}^i \mathbf{d} + {}^i \mathbf{r})$$

2.1.3.2. Μετασχηματισμός Καρτεσιανών Συστημάτων Euler «Φ-Θ-Ψ».

Μετασχηματισμός Σωματόδετου σε Χωρόδετο Καρτεσιανό Σύστημα:

«i»	←	«2»	←	«1»	←	«k»
	ψ		θ		Φ	

Σύμφωνα με τον Euler, ο μετασχηματισμός ενός σωματόδετου καρτεσιανού συστήματος «k» στο χωρόδετο καρτεσιανό σύστημα «i» μπορεί να πραγματοποιηθεί μετά από μία αλληλουχία στροφών φ, θ και ψ μεταβαίνοντας σταδιακά από το σύστημα «k» στα ενδιάμεσα συστήματα «1» και «2» και τελικά στο σύστημα «i» (σχήμα 2.1.13) ως εξής:

1. Στροφή του σωματόδετου καρτεσιανού συστήματος «k» ως προς τον άξονα

${}^k_x = {}^1x$ κατά τη γωνία φ

και μετάβαση σε ένα ενδιάμεσο καρτεσιανό σύστημα «1». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

$${}^1_k\mathbf{R}(\varphi) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{Bmatrix}$$

2. Στροφή του ενδιάμεσου καρτεσιανού συστήματος «1» ως προς τον άξονα

${}^1_y = {}^2y$ κατά τη γωνία θ

και μετάβαση στο επίσης ενδιάμεσο καρτεσιανό σύστημα «2». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

$${}^2_1\mathbf{R}(\theta) = \begin{Bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{Bmatrix}$$

3. Στροφή του ενδιάμεσου καρτεσιανού συστήματος «2» ως προς τον άξονα

${}^2_z = {}^i z$ κατά τη γωνία ψ

και μετάβαση στο χωρόδετο καρτεσιανό σύστημα «i». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

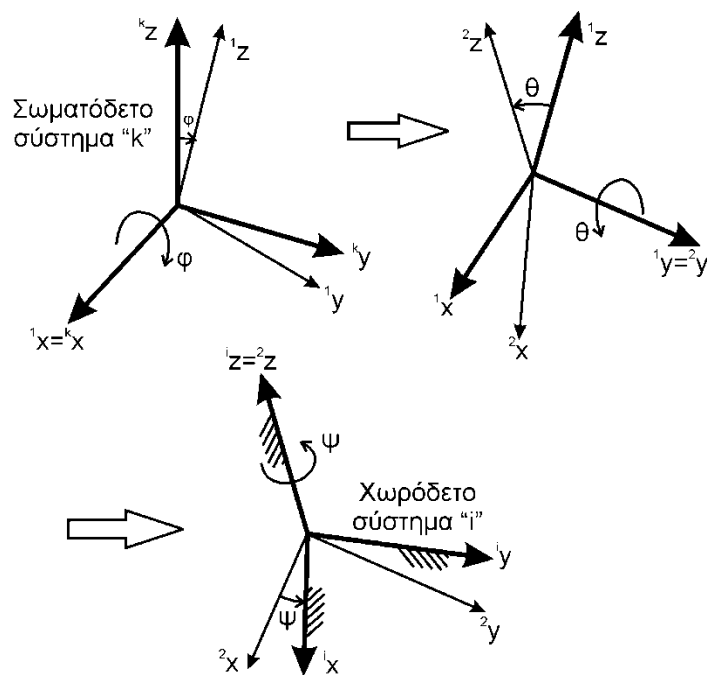
$${}^i_2\mathbf{R}(\psi) = \begin{Bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{Bmatrix}$$

Το τελικό μητρώο μετασχηματισμού από το σωματόδετο «k» στο χωρόδετο καρτεσιανό σύστημα «i» μέσω των στροφών φ-θ-ψ είναι επομένως ίσο με:

$$\begin{aligned}
 {}^i_k \mathbf{R} &= {}^i_k \mathbf{R}(\psi \leftarrow \theta \leftarrow \varphi) = {}^i_2 \mathbf{R}(\psi) \cdot {}^2_1 \mathbf{R}(\theta) \cdot {}^1_k \mathbf{R}(\varphi) \\
 &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta \cdot \sin \varphi & \sin \theta \cdot \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ -\sin \theta & \cos \theta \cdot \sin \varphi & \cos \theta \cdot \cos \varphi \end{bmatrix} \\
 &= \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\theta + c_\psi s_\theta s_\varphi & s_\psi s_\theta + c_\psi s_\theta c_\varphi \\ s_\psi c_\theta & c_\psi c_\theta + s_\psi s_\theta s_\varphi & -c_\psi s_\theta + s_\psi s_\theta c_\varphi \\ -s_\theta & c_\theta s_\varphi & c_\theta c_\varphi \end{bmatrix}
 \end{aligned}$$

όπου $c_x = \cos(x)$ και $s_x = \sin(x)$

Οι ανωτέρω γωνίες φ, θ και ψ λέγονται γωνίες «Euler». Ο μετασχηματισμός είναι «ολονομικός», δηλαδή εξαρτάται από την αλληλουχία των γωνιών φ, θ και ψ. Αυτό σημαίνει ότι είναι άλλη η τελική θέση του σώματος εάν η αλληλουχία των γωνιών είναι η $\varphi \rightarrow \theta \rightarrow \psi$, άλλη όταν είναι η $\varphi \rightarrow \psi \rightarrow \theta$ και άλλη όταν είναι η $\theta \rightarrow \varphi \rightarrow \psi$.



ΣΧΗΜΑ 2.13 Μετασχηματισμός φ→θ→ψ Euler από το σωματόδετο «k» στο χωρόδετο σύστημα «i».

2.1.3.3. Μετασχηματισμός Χωρόδεδου σε Σωματόδετο Καρτεσιανό Σύστημα.

Ο αντίστροφος μετασχηματισμός, δηλαδή ο μετασχηματισμός ενός χωρόδεδου καρτεσιανού συστήματος «i» στο σωματόδετο καρτεσιανό σύστημα «k» μπορεί να πραγματοποιηθεί μετά από μία αλληλουχία.

«i»	→	«2»	→	«1»	→	«k»
	ψ		θ		φ	

στροφών ψ, θ και φ μεταβαίνοντας σταδιακό από το σύστημα «i» στα ενδιάμεσα συστήματα «2» και «1» και τελικά στο σύστημα «k» ως εξής:

1. Στροφή του χωρόδεδου καρτεσιανού συστήματος «i» ως προς τον άξονα

$${}^i z = {}^2 z \text{ κατά τη γωνία } \psi$$

και μετάβαση στο ενδιάμεσο καρτεσιανό σύστημα «2». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

$${}^2_1 \mathbf{R}(\psi) = \begin{Bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{Bmatrix}$$

2. Στροφή του ενδιάμεσου καρτεσιανού συστήματος «2» ως προς τον άξονα

$${}^1 y = {}^2 y \text{ κατά τη γωνία } \theta$$

και μετάβαση στο επίσης ενδιάμεσο καρτεσιανό σύστημα «1». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

$${}^1_2 \mathbf{R}(\theta) = \begin{Bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{Bmatrix}$$

3. Στροφή του ενδιάμεσου καρτεσιανού συστήματος «1» ως προς τον άξονα

$${}^k x = {}^1 x \text{ κατά τη γωνία } \phi$$

και μετάβαση στο σωματόδετο καρτεσιανό σύστημα «k». Το αντίστοιχο μητρώο μετασχηματισμού είναι ίσο με:

$${}^k_1\mathbf{R}(\varphi) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{Bmatrix}$$

Το τελικό μητρώο μετασχηματισμού από το χωρόδετο «i» στο σωματόδετο καρτεσιανό σύστημα «k» μέσω των στροφών ψ-θ-φ είναι επομένως ίσο με:

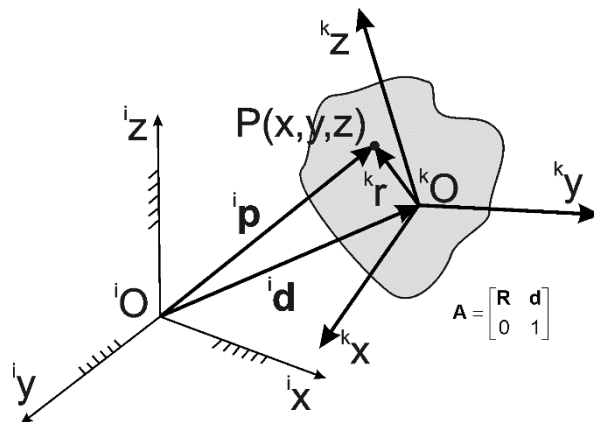
$$\begin{aligned} {}^k_1\mathbf{R} &= {}^k_1\mathbf{R}(\varphi \leftarrow \theta \leftarrow \psi) = {}^k_1\mathbf{R}(\varphi) \cdot {}^2_1\mathbf{R}(\theta) \cdot {}^3_2\mathbf{R}(\psi) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta \cdot \cos\psi & \cos\theta \cdot \sin\psi & -\sin\theta \\ -\sin\psi & \cos\psi & 0 \\ \sin\theta \cdot \cos\psi & \sin\theta \cdot \sin\psi & \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\varphi + c_\psi s_\theta s_\varphi & c_\psi c_\varphi + s_\psi s_\theta s_\varphi & c_\theta s_\varphi \\ s_\psi s_\varphi + c_\psi s_\theta c_\varphi & -c_\psi s_\varphi + s_\psi s_\theta c_\varphi & c_\theta c_\varphi \end{bmatrix} \end{aligned}$$

Η ανωτέρω αλληλουχία των γωνιών ονομάζεται «αλληλουχία ψ→θ→φ» και στα αγγλικά «yaw (ψ) - pitch (θ) - roll (φ)».

Όπως μπορεί κανείς εύκολα να διαπιστώσει, ισχύει η γενική ορθογωνική ιδιότητα των μητρώων μετασχηματισμού, δηλαδή:

$$\left({}^k_1\mathbf{R}\right)^{-1} = \left({}^k_1\mathbf{R}\right)^T \text{ «Ορθογώνια Ιδιότητα»}.$$

2.1.3.4. Η Μέθοδος του Ομογενούς Μετασχηματισμού.



ΣΧΗΜΑ 2.14 Ομογενής μετασχηματισμός «A».

Η μέθοδος του ομογενούς μετασχηματισμού (σχήμα 2.1.14) στοχεύει στην απλοποίηση του μετασχηματισμού:

$${}^i\mathbf{p} = {}^i\mathbf{d} + {}^i\mathbf{R} \cdot {}^k\mathbf{r}$$

ο οποίος περιγράφει τις συντεταγμένες ενός σημείου $P(x,y,z)$. Πράγματι, η παραπάνω εξίσωση γράφεται σύμφωνα με τη μέθοδο του ομογενούς μετασχηματισμού, στη μορφή:

$$\underbrace{\begin{Bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{Bmatrix}}_{{}^i\bar{\mathbf{p}}} = \underbrace{\begin{Bmatrix} \mathbf{p}_{3 \times 1} \\ 1 \end{Bmatrix}}_{{}^i\mathbf{1}} = \underbrace{\begin{bmatrix} {}^i\mathbf{R}_{3 \times 3} & {}^i\mathbf{d}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{1}_{1 \times 1} \end{bmatrix}}_{{}^i\mathbf{A}} \cdot \underbrace{\begin{Bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{Bmatrix}}_{{}^k\bar{\mathbf{r}}} = \underbrace{\mathbf{A}}_{{}^i\mathbf{A}_{4 \times 4}} \cdot \underbrace{\begin{Bmatrix} \mathbf{r}_{3 \times 1} \\ \mathbf{1}_{1 \times 1} \end{Bmatrix}}_{{}^k\bar{\mathbf{r}}_{4 \times 1}} = \underbrace{\mathbf{A}}_{{}^i\mathbf{A}_{4 \times 4}} \cdot \underbrace{\bar{\mathbf{r}}}_{{}^k\bar{\mathbf{r}}_{4 \times 1}}$$

Στην παραπάνω εξίσωση έχουν ορισθεί:

- Το 4x1 διάνυσμα $\bar{\mathbf{r}}$, το οποίο ταυτίζεται γεωμετρικά με το διάνυσμα \mathbf{r} (έχει μόνο προστεθεί η μονάδα στην 4η σειρά)
- Το 4x4 μητρώο \mathbf{A} το οποίο στις θέσεις 3x3 ταυτίζεται με το μητρώο περιστροφής \mathbf{R} , η 4η στήλη του ταυτίζεται με το διάνυσμα \mathbf{d} (έχει μόνο προστεθεί η μονάδα στην 4η σειρά), οι δε 3 πρώτες στήλες της 4ης σειράς του είναι ίσες με το μηδέν

Σύμφωνα με τα παραπάνω, ο πολλαπλασιασμός των δύο μητρώων ${}^i\mathbf{A}$ και ${}^{k-1}\mathbf{A}$ οδηγεί στο εξής μητρώο μετασχηματισμού ${}^{i}_{k-1}\mathbf{A}$:

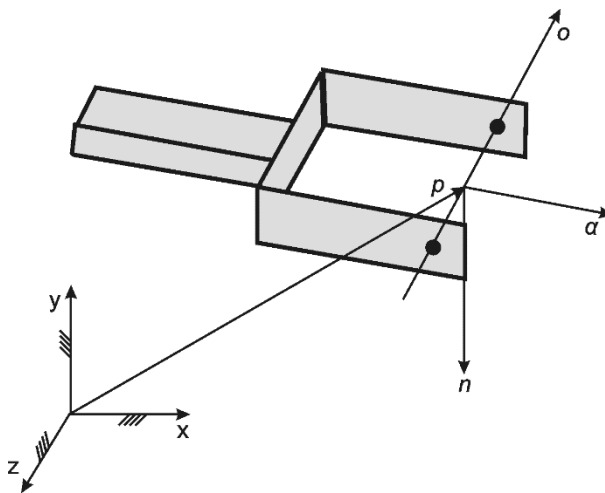
$$\begin{aligned} {}^i_{k-1}\mathbf{A} &= {}^i_k\mathbf{A} \cdot {}^k_{k-1}\mathbf{A} = \begin{bmatrix} {}^i_k\mathbf{R} & {}^i_k\mathbf{d} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^k_{k-1}\mathbf{R} & {}^k_{k-1}\mathbf{d} \\ 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} {}^i_k\mathbf{R} \cdot {}^k_{k-1}\mathbf{R} & {}^i_k\mathbf{R} \cdot {}^k_{k-1}\mathbf{d} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i_{k-1}\mathbf{R} & {}^i_{k-1}\mathbf{d} \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Το μητρώο του ομογενούς μετασχηματισμού μπορεί να γραφτεί και στη μορφή:

$${}^i_k\mathbf{A} = \begin{bmatrix} {}^i_kR_{11} & {}^i_kR_{12} & {}^i_kR_{13} & {}^i_kd_x \\ {}^i_kR_{21} & {}^i_kR_{22} & {}^i_kR_{23} & {}^i_kd_y \\ {}^i_kR_{31} & {}^i_kR_{32} & {}^i_kR_{33} & {}^i_kd_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i\mathbf{n}_x & {}^i\mathbf{n}_y & {}^i\mathbf{n}_z & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου τα διανύσματα ${}^i\mathbf{n}_x=\mathbf{n}$, ${}^i\mathbf{n}_y=\mathbf{o}$ και ${}^i\mathbf{n}_z=\mathbf{a}$ είναι μοναδιαία διανύσματα ${}^k\mathbf{n}_x$, ${}^k\mathbf{n}_y$ και ${}^k\mathbf{n}_z$ του καρτεσιανού συστήματος «k» στις κατευθύνσεις των αξόνων kx , ky και kz τα μετασχηματισμένα στο καρτεσιανό σύστημα «i».

Στην περίπτωση ενός ρομπότ με εργαλείο (βλ. σχήμα 2.15), εισάγεται το σωματόδετο σύστημα καρτεσιανών συντεταγμένων «k», έτσι ώστε:



ΣΧΗΜΑ 2.15 Ρομπότ με εργαλείο.

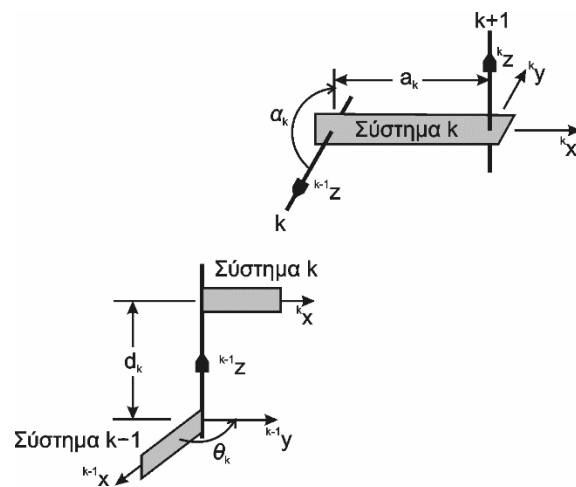
- A. Το διάνυσμα α (z-άξονας) να βρίσκεται στην κατεύθυνση της κίνησης του εργαλείου προς τα εμπρός («approach» vector).
- B. Το διάνυσμα ο (y-άξονας) να βρίσκεται στο επίπεδο του εργαλείου και στη πλάγια κατεύθυνση κίνησης («sliding» vector), και
- Γ. Το διάνυσμα η (x-άξονας) να είναι κάθετο στο επίπεδο ο x α

$n = o \times \alpha$

Το μητρώο συνήθως αποκαλείται και μητρώο «Τα» (T=tool) και χρησιμοποιείται κατά κανόνα ο απλούστερος συμβολισμός:

$${}^i_k \mathbf{A} = \mathbf{T} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{\alpha} & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1.3.5. Ο Μετασχηματισμός Denavit-Hartenberg.



ΣΧΗΜΑ 2.16 Μετασχηματισμός Denavit-Hartenberg.

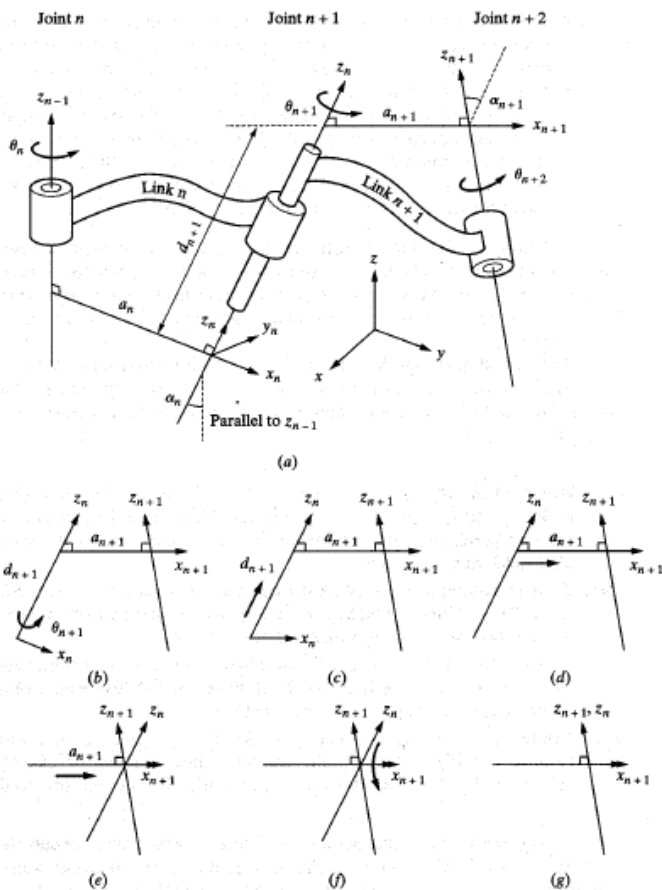
Μετασχηματισμός Denavit-Hartenberg (D-H) από το σύστημα συντεταγμένων k-1 στο σύστημα k.	
Βήμα 1	Περιστροφή του καρτεσιανού συστήματος k-1 γύρω από τον άξονα ${}^{k-1}z$ κατά γωνία θ_k . Άρθρωση $R \rightarrow \theta_k$ μεταβλητή
Βήμα 2	Μετακίνηση του καρτεσιανού συστήματος k-1 κατά μήκος του άξονα ${}^{k-1}z$ κατά μήκος d_k Άρθρωση $P \rightarrow d_k$ μεταβλητή
Βήμα 3	Μετακίνηση του καρτεσιανού συστήματος k-1 κατά μήκος του άξονα ${}^{k-1}x$ κατά a_k Άρθρωση $P \rightarrow a_k$ μεταβλητή
Βήμα 4	Περιστροφή του καρτεσιανού συστήματος k-1 γύρω από τον άξονα ${}^{k-1}x$ κατά τη γωνία α_k Άρθρωση $R \rightarrow \alpha_k$ μεταβλητή

Οι Denavit-Hartenberg (1955) πρότειναν ένα συστηματικό τρόπο τοποθέτησης των καρτεσιανών συστημάτων σε μία κινηματική αλυσίδα, με στόχο την τυποποίηση και απλοποίηση των αναγκαίων μετασχηματισμών.

Πράγματι, σύμφωνα με τη μέθοδο D-H χρησιμοποιείται η εξής τυποποιημένη μετάβαση από ένα καρτεσιανό σύστημα «k-1» στο επόμενο καρτεσιανό σύστημα «k» (σχήματα 2.16 & 2.17).

Το σύστημα «k» προκύπτει από δύο «κοχλιωτές» κινήσεις:

- α. Κοχλιωτή κίνηση κατά μήκος του άξονα ${}^{k-1}z$ με βαθμούς ελευθερίας την απόσταση d_k και τη γωνία θ_k ,
- β. Κοχλιωτή κίνηση κατά μήκος του άξονα kx με βαθμούς ελευθερίας την απόσταση a_k και τη γωνία α_k



ΣΧΗΜΑ 2.17 Τοποθέτηση καρτεσιανών συστημάτων σύμφωνα με τους «D-H».

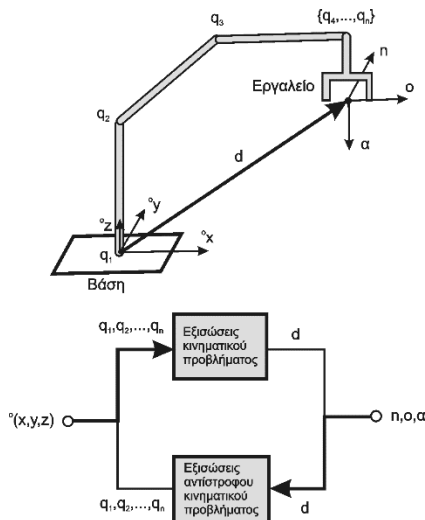
Ο ομογενής μετασχηματισμός από το καρτεσιανό σύστημα «k» στο καρτεσιανό σύστημα «k-1» δίνεται μέσω του τυποποιημένου μητρώου A,

$${}^{k-1}_k \mathbf{D} \equiv {}^{k-1}_k \mathbf{A} = \begin{bmatrix} \cos\theta_k & -\cos\alpha_k \cdot \sin\theta_k & \sin\alpha_k \cdot \sin\theta_k & a_k \cdot \cos\theta_k \\ \sin\theta_k & \cos\alpha_k \cdot \cos\theta_k & -\sin\alpha_k \cdot \cos\theta_k & a_k \cdot \sin\theta_k \\ 0 & \sin\alpha_k & \cos\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ενώ ο αντίστροφος ομογενής μετασχηματισμός από το καρτεσιανό σύστημα «k-1» στο σύστημα «k» είναι ίσος με:

$${}^k_{k-1} \mathbf{D} \equiv {}^k_{k-1} \mathbf{A} = \begin{bmatrix} \cos\theta_k & \sin\theta_k & 0 & -a_k \\ -\cos\alpha_k \cdot \sin\theta_k & \cos\alpha_k \cdot \cos\theta_k & \sin\alpha_k & -d_k \cdot \sin\alpha_k \\ \sin\alpha_k \cdot \sin\theta_k & -\sin\alpha_k \cdot \cos\theta_k & \cos\alpha_k & -d_k \cdot \cos\alpha_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1.3.6. Αντίστροφη Κινηματική Ανάλυση Ρομπότ.



ΣΧΗΜΑ 2.18 Ορισμός του αντίστροφου κινηματικού προβλήματος.

Το αντίστροφο κινηματικό πρόβλημα ρομποτικών μηχανισμών (σχήμα 2.18) συνίσταται στον προσδιορισμό του διανύσματος q των κινηματικών βαθμών ελευθερίας $q_i = \{\theta_i / d_i\}$:

$$q = \begin{Bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{Bmatrix}$$

όταν είναι δεδομένα (συμβολισμός: μπάρα) το διάνυσμα απόστασης d , καθώς και τα διανύσματα προσανατολισμού n, o, α του εργαλείου (ή κάποιου άλλου μέλους) του ρομπότ (σχήμα 2.19):

$$n = n(q_1, q_2, \dots, q_n) = \bar{n}$$

$$o = o(q_1, q_2, \dots, q_n) = \bar{o}$$

$$\alpha = \alpha(q_1, q_2, \dots, q_n) = \bar{\alpha}$$

$$d = d(q_1, q_2, \dots, q_n) = \bar{d}$$

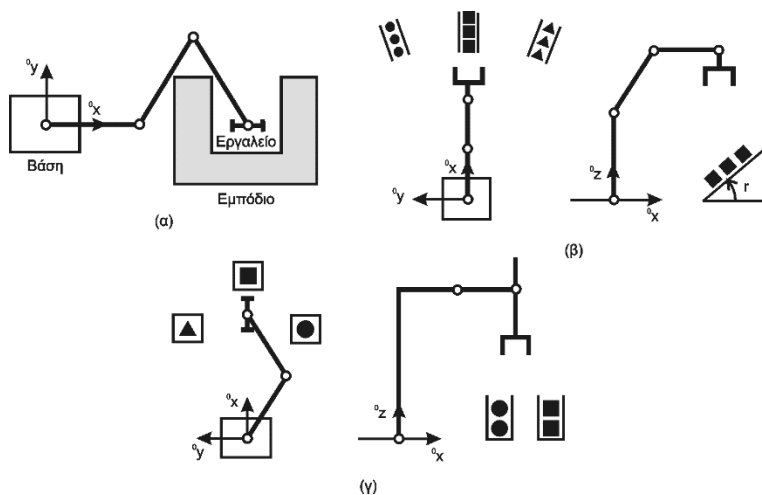
Συνοπτικά αυτό εκφράζεται με την εξίσωση:

$$T = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{\alpha} & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} = T(q_1, q_2, \dots, q_n) \equiv \bar{T} = \begin{bmatrix} \bar{\mathbf{n}} & \bar{\mathbf{o}} & \bar{\mathbf{\alpha}} & \bar{\mathbf{d}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Επειδή το μητρώο T είναι μία μη-γραμμική συνάρτηση των βαθμών ελευθερίας q , το αντίστροφο κινηματικό πρόβλημα, δηλαδή η αναλυτική επίλυση των παραπάνω μη γραμμικών εξισώσεων μπορεί

να μην είναι πάντοτε εφικτή. Στις περιπτώσεις αυτές χρησιμοποιούνται επαναληπτικές-αριθμητικές μέθοδοι επίλυσης.

Γενικά, από τη θέση d του εργαλείου προκύπτουν 3 εξισώσεις υπολογισμού, ενώ από το μητρώο μετασχηματισμού $R=[n \ o \ a]$ το οποίο εμπεριέχει $3 \times 3=9$ στοιχεία, 4 εξισώσεις. Οι υπόλοιπες 5 εξισώσεις (από τις 9 εξισώσεις του μητρώου R) αφορούν εξισώσεις συσχετισμού των παραπάνω στοιχείων μεταξύ τους. Συνεπώς ο προσανατολισμός του εργαλείου περιγράφεται από τους εναπομένοντες 4 βαθμούς ελευθερίας.



ΣΧΗΜΑ 2.19 Παραδείγματα αντίστροφων κινηματικών προβλημάτων.

Μεταβλητές Euler:

Εάν οι 4 βαθμοί ελευθερίας συσχετισθούν με τις μεταβλητές Euler ϕ_E και n_E

$$\phi_E = \phi_E \cdot n_E$$

όπου ϕ_E είναι η γωνία και n_E το μοναδιαίο διάνυσμα του διανύσματος περιστροφής, τότε οι παραπάνω μεταβλητές μπορούν να υπολογισθούν από το μητρώο περιστροφής ως εξής:

1. Επειδή το μητρώο R είναι ορθογωνικό, μία από τις ιδιοτιμές του λ :

$$(R - \lambda I) \cdot n_E = 0$$

είναι ίση με την μονάδα $\lambda=1$ (I =μοναδιαίο μητρώο). Επομένως, οι τρεις συνιστώσες του μοναδιαίου διανύσματος:

$$n_E = \{n_{Ex} \ n_{Ey} \ n_{Ez}\}^T$$

μπορούν να προσδιορισθούν από τον υπολογισμό των ιδιοτιμών του μητρώου $R=[n \ o \ a]$ για $\lambda=1$.

2. Η γωνία ϕ_E υπολογίζεται από το άθροισμα των διαγώνιων όρων του μητρώου R και είναι ίση με:

$$\cos \phi_E = \frac{R_{11} + R_{22} + R_{33} - 1}{2} = \frac{n_x + o_y + a_z - 1}{2}$$

Χρησιμοποιώντας τις παραμέτρους Euler, η θέση και κατεύθυνση του εργαλείου (ή ενός άλλου μέλους του μηχανισμού) προκύπτει τελικά από το διάνυσμα «λ_E»:

Για την περιγραφή του προσανατολισμού τού εργαλείου μπορεί να χρησιμοποιηθεί και η μορφή λ_Q («Quaternion»). Οι 4 νέες μεταβλητές είναι οι:

$$Q_0 = \cos(\phi_E / 2)$$

$$Q_x = n_{Ex} \cdot \sin(\phi_E / 2)$$

$$Q_y = n_{Ey} \cdot \sin(\phi_E / 2)$$

$$Q_z = n_{Ez} \cdot \sin(\phi_E / 2)$$

Με τους παραπάνω ορισμούς το νέο 7x1 διάνυσμα λ_Q προκύπτει:

$$\lambda_Q = \{d_x, d_y, d_z, Q_0, Q_x, Q_y, Q_z\}^T$$

Προφανώς, τόσο οι μεταβλητές Quaternions όσο και οι μεταβλητές Euler μαζί με τις μεταβλητές απόστασης d μπορούν να περιγράψουν τη θέση και τον προσανατολισμό τού εργαλείου (ή του μέλους) ενός μηχανισμού.

Το 6x1 Διάνυσμα λ:

Η επίλυση του αντίστροφου κινηματικού προβλήματος θα μπορούσε να είχε χρησιμοποιήσει και τις γωνίες Euler φ, θ και ψ με αποτέλεσμα τον ορισμό 6 και όχι 7 ανεξαρτήτων μεταβλητών. Όμως, αυτό θα είχε σαν αποτέλεσμα την εισαγωγή των ολονομικών ιδιοτήτων του φ-θ-ψ μετασχηματισμού, κάτι που δεν είναι επιθυμητό.

Από την άλλη πλευρά, επειδή η τρισδιάστατη κίνηση ενός σώματος περιγράφεται συνήθως από 6 μεταβλητές (3 μετατοπίσεις και 3 γωνίες ως προς τους άξονες ενός καρτεσιανού συστήματος), τα 7x1 διανύσματα λ_E και λ_Q δε συμβαδίζουν με τον υπάρχοντα φορμαλισμό.

Συνεπώς, είναι επιθυμητό να ορισθεί ένα 6x1 διάνυσμα «λ» (αντί των 7x1 διανυσμάτων) το οποίο θα εμπεριέχει τις αντίστοιχες πληροφορίες. Το διάνυσμα αυτό μπορεί πράγματι να ορισθεί και είναι ίσο με:

$$\lambda = \left\{ \begin{array}{l} \mathbf{d} \\ \alpha \cdot e^{\beta/\pi} \end{array} \right\}$$

όπου d και ο είναι τα γνωστά 3x1 διανύσματα του μητρώου T. Πρόσθετη είναι η νέα μεταβλητή β, που περιγράφει τη στροφή γύρω από το διάνυσμα α.

Η παραπάνω σχέση προκύπτει από το γεγονός πως το διάνυσμα α είναι η απεικόνιση του μοναδιαίου διανύσματος του άξονα z του εργαλείου στο χωρόδετο σύστημα «i». Επομένως, το μήκος του είναι ίσο με την μονάδα. Άρα, τα δύο διανύσματα d και α περιγράφουν την θέση και κατεύθυνση του εργαλείου του μηχανισμού. Η στροφή β γύρω από τον άξονα z του εργαλείου, εισάγεται με την συνάρτηση $\exp(\beta/\pi)$ (=μέτρο του διανύσματος).

Με τους παραπάνω ορισμούς προκύπτουν οι εξής σχέσεις:

$$\alpha = \frac{1}{\sqrt{\lambda_4^2 + \lambda_5^2 + \lambda_6^2}} \cdot \begin{Bmatrix} \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{Bmatrix} \quad \beta = \pi \cdot \ln\left(\sqrt{\lambda_4^2 + \lambda_5^2 + \lambda_6^2}\right)$$

λαμβάνοντας υπόψη πως $\|\alpha\| = 1$

2.1.4. Μέθοδοι Προγραμματισμού Ρομπότ.

Ο προγραμματισμός ρομπότ πραγματοποιείται με διάφορους τρόπους. Σύμφωνα με την τρέχουσα βιομηχανική πρακτική, διαιρούμε τις μεθόδους προγραμματισμού σε δύο βασικούς τύπους:

- Μεθόδων διδασκαλίας (Leadthrough)
- Μεθόδων με γλώσσες προγραμματισμού ρομπότ

Επίσης, εκτός από τη leadthrough και τις γλώσσες προγραμματισμού, μια άλλη μέθοδος χρησιμοποιείται για τα απλά ρομπότ χαμηλής τεχνολογίας. Η μέθοδος αυτή χρησιμοποιείται σε τύπους μηχανών που αναφέρονται ως ρομπότ περιορισμένης ακολουθίας και που ελέγχονται με τη βοήθεια των μηχανικών στάσεων και των αλλαγών ορίου για να καθορίσουν τα σημεία τέλους των κοινών κινήσεών τους. Ο καθορισμός αυτών των στάσεων και διακοπών μοιάζει με μέθοδο προγραμματισμού. Ωστόσο αυτό το είδος προγραμματισμού θεωρείται ως χειροκίνητη διαδικασία οργάνωσης.

Μέθοδοι Προγραμματισμού Leadthrough.

Στον προγραμματισμό με διδασκαλία, το ρομπότ κινείται κατά μήκος της επιθυμητής πορείας κινήσεων προκειμένου να καταγραφεί η πορεία στη μνήμη των ελεγκτών. Υπάρχουν δύο τρόποι διδασκαλίας:

- Τροφοδοτημένη
- Χειροκίνητη

Η τροφοδοτημένη διδασκαλία χρησιμοποιεί ένα χειριστήριο διδασκαλίας για να ελέγξει τους κινητήρες των αρθρώσεων, και να οδηγήσει το ρομποτικό βραχίονα και τον καρπό μέσω μιας σειράς σημείων στο χώρο. Κάθε σημείο καταγράφεται στη μνήμη για την επόμενη αναπαραγωγή του κατά τη διάρκεια του κύκλου εργασίας. Το χειριστήριο διδασκαλίας είναι συνήθως ένα μικρό φορητό παράθυρο ελέγχου με τους συνδυασμούς διακοπών αναστροφής, πινάκων, και κουμπιών για να ρυθμίσει τις φυσικές μετακινήσεις του ρομπότ και τις ικανότητες προγραμματισμού. Μεταξύ των διάφορων μεθόδων προγραμματισμού ρομπότ, η τροφοδοτημένη διδασκαλία είναι πιθανώς η πιο διαδεδομένη σήμερα. Περιορίζεται κατά μεγάλο μέρος στις -από σημείο σε σημείο- κινήσεις και δε χρησιμοποιείται εύκολα στη συνεχή μετακίνηση λόγω της δυσκολίας που παρουσιάζει το χειριστήριο διδασκαλίας να ρυθμίσει τις σύνθετες γεωμετρικές κινήσεις στο χώρο. Παραδείγματα περιλαμβάνουν εφαρμογές, όπως: μεταφορά κομματιών, φόρτωση και εκφόρτωση μηχανών και σημειακή συγκόλληση.

Η χειροκίνητη διδασκαλία (επίσης μερικές φορές αποκαλούμενη μέθοδος «περάσματος») χρησιμοποιείται ευκολότερα για στο, συνεχούς πορείας, προγραμματισμό όπου ο κύκλος κινήσεων περιλαμβάνει ομαλές, σύνθετες, καμπυλόγραμμες μετακινήσεις τού ρομποτικού βραχίονα. Το πιο κοινό παράδειγμα αυτού του είδους εφαρμογής ρομπότ είναι βαφή ψεκασμού. Ο καρπός του ρομπότ με το πιστόλι ψεκασμού, που συνδέεται ως τελικό στοιχείο δράσης, πρέπει να εκτελέσει ένα ομαλό, κανονικό σχέδιο κινήσεων προκειμένου να εφαρμοστεί το χρώμα ομοιόμορφα σε ολόκληρη την επιφάνεια που βάφεται. Οι συγκολλήσεις τόξου είναι άλλο ένα παράδειγμα στο οποίο ο απαιτείται περιγραφή συνεχούς πορείας.

Στη χειροκίνητη μέθοδο, ο προγραμματιστής πιάνει φυσικά το βραχίονα ρομπότ (και το τελικό στοιχείο δράσης) και τον κινεί με το χέρι κατά μήκος του επιθυμητού κύκλου κινήσεων. Εάν το ρομπότ είναι μεγάλο και είναι δύσκολο να κινηθεί φυσικά, μια ειδική συσκευή προγραμματισμού αντικαθιστά συχνά το πραγματικό ρομπότ. Αυτή η συσκευή έχει βασικά την ίδια γεωμετρία με το ρομπότ, αλλά είναι ευκολότερη στο χειρισμό κατά τη διάρκεια του προγραμματισμού. Ένα κουμπί διδασκαλίας βρίσκεται συχνά κοντά στον καρπό του ρομπότ (ή στην ειδική συσκευή προγραμματισμού) που είναι πιεσμένο κατά τη διάρκεια εκείνων των κινήσεων που θα αποτελέσουν μέρος του προγραμματισμένου κύκλου. Ο κύκλος κινήσεων διαιρείται σε εκατοντάδες ή χιλιάδες, πολύ κοντινά μεταξύ τους, σημεία στο χώρο κατά μήκος της πορείας και αυτά τα σημεία καταγράφονται στη μνήμη των ελεγκτών.

Τα συστήματα ελέγχου και για τις δύο διαδικασίες διδασκαλίας υποστηρίζουν δύο λειτουργίες: λειτουργία διδασκαλίας ή λειτουργία εκτέλεσης. Η λειτουργία διδασκαλίας χρησιμοποιείται για να προγραμματιστεί το ρομπότ και η λειτουργία εκτέλεσης χρησιμοποιείται για να εκτελέσει το πρόγραμμα.

Οι απαιτήσεις ικανότητας των προγραμματιστών είναι σχετικά μέτριες και αυτές οι διαδικασίες μπορούν να εκτελεστούν εύκολα στις εγκαταστάσεις εργασίας.

Μέθοδος με Γλώσσες Προγραμματισμού.

Ο προγραμματισμός ρομπότ με τις γλώσσες προγραμματισμού μοιάζει με τον προγραμματισμό υπολογιστών. Ο προγραμματιστής δακτυλογραφεί το πρόγραμμα χρησιμοποιώντας μία υψηλού επιπέδου γλώσσα. Η διαδικασία εμπλουτίζεται συνήθως με τη χρησιμοποίηση των τεχνικών διδασκαλίας για να διδαχθούν στο ρομπότ οι θέσεις των σημείων στο χώρο εργασίας. Οι γλώσσες προγραμματισμού άρχισαν να αναπτύσσονται τη δεκαετία του '70, με την πρώτη εμπορική γλώσσα να εμφανίζεται το 1979.

2.1.5. Τρόποι Μετάβασης από Σημείο σε Σημείο.

Για ένα ρομπότ με «ν» άξονες, κάθε σημείο στην πορεία του περιγράφεται από «ν» ισότιμες τιμές, που καθεμιά αντιστοιχεί στη θέση μιας άρθρωσης. Καθένας από τους 4 βασικούς τύπους βραχιόνων έχει τρεις άξονες που συνδέουν το βραχίονα και το σώμα και δύο ή τρεις πρόσθετες αρθρώσεις στον καρπό.

Οι αρθρώσεις βραχιόνων και σώματος καθορίζουν τη γενική θέση στο χώρο του τελικού στοιχείου δράσης και ο καρπός καθορίζει τον προσανατολισμό του. Για ένα σημείο στο χώρο ως θέση και προσανατολισμό του τελικού στοιχείου δράσης, υπάρχουν συνήθως περισσότερα από ένα πιθανά

σύνολα τιμών των μεταβλητών των αρθρώσεων που του αντιστοιχούν (σχήμα 2.20). Η προδιαγραφή ενός σημείου στο χώρο δεν καθορίζει μονοσήμαντα τις συντεταγμένες των αρθρώσεων του ρομπότ. Στο πρόγραμμα του ρομπότ διευκρινίζονται ταυτόχρονα θέση και προσανατολισμός του τελικού στοιχείου δράσης σε όλη την πορεία, όπως και οι τιμές των μεταβλητών των αρθρώσεων.



ΣΧΗΜΑ 2.20 Δύο διαφορετικές διαμορφώσεις του άξονα με τον end effector τοποθετημένο στο ίδιο σημείο.

Ανάλογα με την εφαρμογή, όλες οι παραπάνω μέθοδοι προγραμματισμού επιτρέπουν πραγματοποίηση της επιθυμητής κίνησης του βραχίονα με δύο τρόπους, δηλαδή είτε από σημείο σε σημείο (point to point) είτε με συνεχή διαδρομή. Ωστόσο, πρέπει να σημειωθεί πως κατά τον προγραμματισμό ενός κύκλου εργασίας για ένα ρομπότ συνήθως χρησιμοποιούνται και οι δύο τρόποι, ανάλογα με την εργασία που πραγματοποιεί ο βραχίονας σε κάθε κομμάτι του κύκλου εργασίας και πως η κάθε μέθοδος προγραμματισμού είναι καταλληλότερη για τον ένα ή τον άλλο τρόπο μετάβασης.

2.1.5.1. Μετάβαση από Σημείο σε Σημείο.

Η μετάβαση από σημείο σε σημείο είναι ίσως ο παλαιότερος τρόπος προγραμματισμού της κίνησης ρομποτικών βραχιόνων και ο απλούστερος. Στην πραγματικότητα ο προγραμματιστής προσδιορίζει (διδάσκει) δύο ή περισσότερα σημεία στο χώρο και στη συνέχεια «ζητά» από το ρομπότ να πραγματοποιήσει την κίνησή του, περνώντας με καθορισμένη σειρά από αυτά τα σημεία και είτε απλά να περάσει από αυτά είτε να σταματήσει σε κάποιο ή κάποια από αυτά να πραγματοποιήσει κάποια εργασία.

Όλες οι μέθοδοι προγραμματισμού που αναφέρθηκαν υποστηρίζουν τον προγραμματισμό με μετάβαση από σημείο σε σημείο. Μπορεί δηλαδή ο προγραμματιστής είτε με τη βοήθεια της κονσόλας χειρισμού του βραχίονα, να τοποθετήσει το βραχίονα στην επιθυμητή θέση και στη συνέχεια να την αποθηκεύσει στη μνήμη του ελεγκτή του ρομπότ είτε πιάνοντας το βραχίονα με τα χέρια του, να τον περάσει από τα σημεία που χρειάζεται είτε ακόμα και να προγραμματίσει τα σημεία αυτά σε ένα κατάλληλο πρόγραμμα για το ρομπότ. Ωστόσο, συνήθως χρησιμοποιείται μία από τις μεθόδους διδασκαλίας που απαιτούν λιγότερους υπολογισμούς.

Με αυτόν τον τρόπο μετάβασης, η κίνηση που θα πραγματοποιήσει τελικά ο ρομποτικός βραχίονας είναι απροσδιόριστη. Όταν αποθηκεύεται μια θέση του ρομποτικού βραχίονα στο χώρο, στη μνήμη του ρομποτικού ελεγκτή αποθηκεύονται οι τιμές της κάθε άρθρωσης. Στις περισσότερες περιπτώσεις, το πρόγραμμα ελέγχου του ρομποτικού βραχίονα υπολογίζει τις αποστάσεις ή τις γωνίες που πρέπει να διανύσει κάθε άρθρωση για να φτάσει από τη μία θέση στην επόμενη. Στη συνέχεια με βάση τη

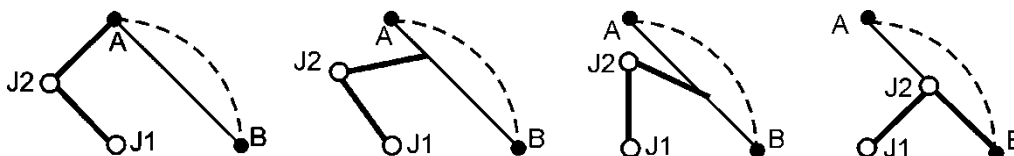
μέγιστη επιτρεπτή ταχύτητα κάθε άρθρωσης, υπολογίζει το χρόνο που χρειάζεται η κάθε άρθρωση για να φτάσει στην επιθυμητή θέση και με βάση των μεγαλύτερο από αυτούς τους χρόνους πραγματοποιεί τη μετάβαση κινώντας κάθε άρθρωση με τέτοια ταχύτητα ώστε όλες οι αρθρώσεις να ξεκινήσουν από το ένα σημείο και να σταματήσουν ή να φτάσουν στο επόμενο, ταυτόχρονα.

Εύκολα γίνεται αντιληπτό πως ειδικά σε αρθρωτούς βραχίονες, η τελική κίνηση που πραγματοποιεί το τελικό στοιχείο δράσης είναι απροσδιόριστη, τόσο ως προς την τροχιά που θα ακολουθήσει, όσο και ως προς τον προσανατολισμό που θα έχει κατά τη διάρκεια της κίνησης. Το γεγονός αυτό καθιστά απαγορευτική τη μέθοδο αυτή σε περιπτώσεις που απαιτείται ακρίβεια σε συγκεκριμένη τροχιά ή με συγκεκριμένο προσανατολισμό. Ωστόσο υπάρχουν μερικοί καλοί λόγοι χρήσης της μεθόδου αυτής. Δηλαδή:

- Επίτευξη της ταχύτερης μετάβασης
- Καθορισμό μιας θέσης εργασίας για το τελικό στοιχείο δράσης
- Αποφυγή εμποδίων

Στην πρώτη περίπτωση, ένα ρομπότ για να μπορέσει να κινηθεί από ένα σημείο σε ένα άλλο ακολουθώντας συγκεκριμένη τροχιά, συνήθως πρέπει να κινήσει τις αρθρώσεις του σε μεγαλύτερο εύρος απ' ό,τι αν επιτρεπόταν να κινηθεί αυθαίρετα μεταξύ των δύο αυτών σημείων. Για παράδειγμα, (σχήμα 2.21) ο βραχίονας με δύο αρθρώσεις J1 και J2 για να κινηθεί αυθαίρετα από το σημείο A στο σημείο B (μετάβαση από σημείο σε σημείο), θα αρκούσε να περιστρέψει μόνο την άρθρωση J1 κατά 90 μοίρες. Στην περίπτωση που θα επιβάλλονταν να κινηθεί σε ευθεία, θα κινήσει και την άρθρωση J2, και μάλιστα πρώτα δεξιόστροφα κατά 40 περίπου μοίρες και στη συνέχεια αριστερόστροφα για άλλο τόσο.

Η κίνηση αυτή, ειδικά αν η J2 άρθρωση είναι πιο αργή απ' την J1 μπορεί να απαιτήσει περισσότερο χρόνο για να ολοκληρωθεί απ' ό,τι η μετάβαση από σημείο σε σημείο.



ΣΧΗΜΑ 2.21 Η κίνηση αρθρώσεων για ευθεία μετάβαση.

Στη δεύτερη περίπτωση, το ρομπότ πρέπει να πάρει ένα τεμάχιο από μια δεδομένη θέση ή να εκτελέσει μια σημειακή συγκόλληση σε μια διευκρινισμένη θέση. Κάθε θέση είναι ένα καθορισμένο σημείο στο πρόγραμμα. Αυτή η περίπτωση περιλαμβάνει επίσης τις ασφαλείς θέσεις που απαιτούνται στον κύκλο εργασίας. Για παράδειγμα, είναι απαραίτητο να καθορισθεί ένα ασφαλές, μακρινό σημείο στο χώρο εργασίας από το οποίο το ρομπότ θα ξεκινά τον κύκλο εργασίας.

Η τρίτη περίπτωση χρησιμοποιείται για να καθορίσει ένα ή περισσότερα σημεία στο χώρο απ' τα οποία πρέπει να περάσει το ρομπότ έτσι ώστε να εξασφαλιστεί ότι δεν θα συγκρουστεί με άλλα αντικείμενα που βρίσκονται στο κύτταρο παραγωγής, π.χ. μηχανές, μεταφορικές διατάξεις, και άλλα τμήματα του

εξοπλισμού. Με τον καθορισμό μιας διαδοχής σημείων γύρω από αυτά τα εμπόδια, οι συγκρούσεις μπορούν να αποτραπούν.

2.1.5.2. Μετάβαση Συνεχούς Διαδρομής.

Με αυτόν τον όρο χαρακτηρίζονται όλοι εκείνοι οι τρόποι μετάβασης από ένα σημείο του χώρου σε ένα άλλο, για τους οποίους ο προγραμματιστής έχει προκαθορίσει κάθε ενδιάμεσο σημείο της διαδρομής που θα ακολουθήσει ο ρομποτικός βραχίονας. Αυτό επιτυγχάνεται με τους εξής τρόπους:

- Ευθύγραμμη παρεμβολή
- Κυκλική παρεμβολή
- Ποικίλες ομαλοποιημένες κινήσεις (χειροκίνητη διδασκαλία)

Κατά την ευθύγραμμη παρεμβολή, ο προγραμματιστής του βραχίονα, αφού ορίσει τα σημεία αρχής και τέλους της κίνησης που επιθυμεί, στη συνέχεια ορίζει από το χειριστήριο διδασκαλίας ότι πρέπει να ακολουθηθεί η ευθύγραμμη κίνηση. Τότε ο ελεγκτής του βραχίονα βρίσκει τα κοντινότερα, στην ευθεία αυτή, προσπελάσιμα σημεία και πραγματοποιεί την κίνηση. Η παρεμβολή αυτή προγραμματίζεται εύκολα και με τη βοήθεια των γλωσσών προγραμματισμού, ανάλογα με τις δυνατότητες του βραχίονα.

Η κυκλική παρεμβολή απαιτεί από τον προγραμματιστή να καθορίσει έναν κύκλο στο χώρο εργασίας του ρομπότ. Αυτό γίνεται πιο εύκολα με τον ορισμό τριών σημείων στην περιφέρεια του κύκλου. Ο ελεγκτής κατασκευάζει έπειτα μια προσέγγιση του κύκλου με την επιλογή μιας σειράς προσπελάσιμων σημείων που βρίσκονται πιο κοντά στον καθορισμένο κύκλο. Οι κινήσεις που πραγματοποιούνται από το ρομπότ αποτελούνται στην πραγματικότητα από μικρά ευθύγραμμα τμήματα προσέγγισης του κύκλου. Η κυκλική παρεμβολή προγραμματίζεται ευκολότερα χρησιμοποιώντας μια γλώσσα προγραμματισμού σε σχέση με τις τεχνικές διδασκαλίας.

Στο χειροκίνητο προγραμματισμό διδασκαλίας, όταν ο προγραμματιστής κινεί τον καρπό για να διδάξει τη βαφή ψεκασμού ή τη συγκόλληση τόξων, οι μετακινήσεις αποτελούνται χαρακτηριστικά από συνδυασμούς ομαλών τμημάτων κινήσεων. Αυτά τα τμήματα είναι μερικές φορές περίπου ευθείες, μερικές φορές καμπύλες (όχι απαραίτητα κυκλικές) και μερικές φορές κινήσεις μπρος-πίσω. Αναφερόμαστε σε αυτές τις μετακινήσεις ως ποικίλες, ομαλές κινήσεις και απαιτούν διαδικασία παρεμβολής σε ακολουθία πολύ κοντινών σημείων που καταγράφονται στη μνήμη του ελεγκτή. Αυτές οι θέσεις αποτελούν τα κοντινότερα προσπελάσιμα σημεία στην πορεία.

2.1.6. Ανάγκες και Περιορισμοί κατά την Κίνηση.

Ανάλογα με το κάθε σκέλος του κύκλου εργασίας του ρομπότ, προκύπτουν και οι λιγότερο ή περισσότερο αυστηρές προδιαγραφές, σε ό,τι αφορά τον τρόπο κίνησής του. Οι προδιαγραφές αυτές συνοψίζονται ως εξής:

- Αδιάφορη κίνηση του τελικού στοιχείου δράσης
- Συγκεκριμένη τροχιά
- Συγκεκριμένος προσανατολισμός
- Συγκεκριμένη τροχιά με καθορισμένο προσανατολισμό

- Περιορισμοί από συγκρούσεις κατά την κίνηση του βραχίονα με άλλα αντικείμενα του χώρου εργασίας

Δεν είναι λίγα τα τμήματα του κύκλου εργασίας ενός βραχίονα στα οποία δεν ενδιαφέρει η μορφή της διαδρομής που θα ακολουθήσει το τελικό στοιχείο δράσης, αλλά μόνο η αρχική θέση και η σωστή τοποθέτησή του στην τελική θέση. Τέτοια τμήματα είναι η κίνηση ενός αφόρτωτου βραχίονα από τη θέση απόθεσης τεμαχίων στη θέση παραλαβής -σε περιβάλλον χωρίς εμπόδια- η μετακίνηση από θέση σε θέση για πραγματοποίηση σημειακών συγκολλήσεων ή πολλές φορές και η μετακίνηση του βραχίονα στην πρώτη θέση του κύκλου εργασίας από θέση στάσης. Όπως είναι φανερό, ο προγραμματισμός τέτοιου είδους κινήσεων πραγματοποιείται με τη μέθοδο «από σημείο σε σημείο».

Η κίνηση του βραχίονα κατά τέτοιο τρόπο ώστε το τελικό στοιχείο δράσης να ακολουθεί συγκεκριμένη τροχιά, είναι η πιο συνηθισμένη. Ακόμα και για τις απλές εφαρμογές που περιγράφηκαν παραπάνω, και που σε γενικές γραμμές μπορούν να πραγματοποιηθούν με αδιάφορη κίνηση, εμπόδια στο χώρο εργασίας του βραχίονα μπορεί να επιβάλουν την κίνηση σε συγκεκριμένη τροχιά για την αποφυγή συγκρούσεων. Επίσης, η φόρτωση και εκφόρτωση εργαλειομηχανών επιβάλλει τέτοιου είδους κίνηση για να μπορέσει το ρομπότ να περάσει μέσα από την πόρτα της εργαλειομηχανής. Σε αυτές τις περιπτώσεις χρησιμοποιούμε κάποιον από τους τρόπους μετάβασης συνεχούς διαδρομής και κυρίως τις παρεμβολές ευθείας ή κυκλικής τροχιάς.

Σπανιότερη περίπτωση είναι να ενδιαφέρει μόνο ο προσανατολισμός που θα έχει το τελικό στοιχείο δράσης κατά τη διάρκεια της κίνησης του βραχίονα και όχι η τροχιά που θα ακολουθήσει, π.χ. κατά τη μεταφορά τεμαχίων που λόγω σχήματος ή ευαισθησίας δεν πρέπει να περιστραφούν. Ο προγραμματιστής πρέπει να είναι σε θέση να διακρίνει και να αξιοποιήσει την ελευθερία επιλογής της τροχιάς, κατά τέτοιο τρόπο ώστε να πετύχει την πραγματοποίηση της κίνησης στον ελάχιστο χρόνο.

Οι περισσότερες εφαρμογές των ρομποτικών βραχιόνων σήμερα -όπως η συναρμολόγηση μικρών ή μεγάλων προϊόντων- η συγκόλληση τόξου, η βαφή επίπεδων ή καμπύλων επιφανειών κ.α. απαιτούν την κίνηση του τελικού στοιχείου δράσης όχι μόνο σε καθορισμένη τροχιά αλλά και με συγκεκριμένο προσανατολισμό. Τέτοιες κινήσεις προγραμματίζονται ως επί των πλείστο με το χειροκίνητο προγραμματισμό διδασκαλίας και αποτελούν τα τμήματα αυτά του κύκλου εργασίας όπου η ακρίβεια και η ποιότητα της εργασίας του βραχίονα έχουν μεγαλύτερη βαρύτητα από το χρόνο εκτέλεσης.

Κατά τον προγραμματισμό των κινήσεων ενός βραχίονα περιορισμοί προκύπτουν, όχι μόνον από τα εμπόδια που μπορεί να υπάρχουν κατά μήκος της τροχιάς του τελικού στοιχείου δράσης αλλά και αυτά που μπορεί να παρουσιαστούν και να δημιουργήσουν συγκρούσεις με κάποιον από τους συνδέσμους του βραχίονα, που κατά τον προγραμματισμό μπορεί να μη γίνονται άμεσα αντιληπτές, με πιθανά καταστροφικές συνέπειες.

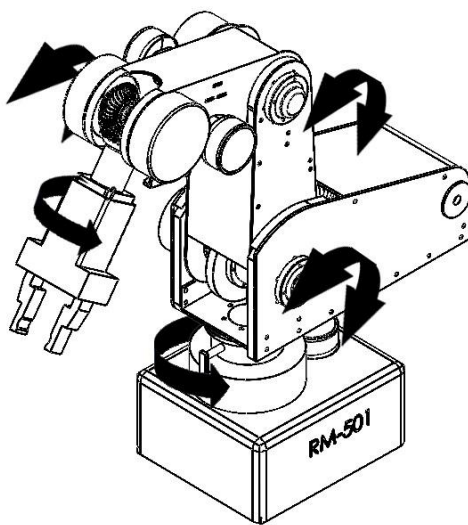
2.2. Ο ρομποτικός βραχίονας MITSUBISHI RM-501

Ο ρομποτικός βραχίονας MITSUBISHI MOVEMASTER RM-501 είναι ένας αρθρωτός βιομηχανικός ρομποτικός βραχίονας 5 βαθμών ελευθέριας, όπως φαίνεται στο σχήμα 2.22 (δεν έχει την δυνατότητα της κύλισης καρπού :yaw). Οι Επενεργητές των αρθρώσεων είναι ηλεκτροκινητήρες συνεχούς ρεύματος, 24V, βηματικού χαρακτήρα με ανάδραση. Πρόκειται για ένα μοντέλο που μπήκε στην γραμμή παραγωγής το 1986 και αποσύρθηκε ένα χρόνο αργότερα. Ήταν το τελευταίο ρομπότ της σειράς RM, το διαδέχτηκε η σειρά RV, μεγαλύτερα ρομπότ με μεγαλύτερο χώρο εργασίας μέγιστης επιδεξιότητας.

Σκοπός της δημιουργίας του ήταν να υπάρξει ένα μικρό, προσιτό ρομπότ σε μικρότερης κλίμακας βιομηχανικές εγκαταστάσεις. Χρησιμοποιήθηκε κυρίως για συναρμολόγηση και την μεταφορά αντικειμένων από μια θέση σε μια άλλη, καθώς επίσης βρήκε και μεγάλη εφαρμογή σε εργαστήρια χημικού και βιολογικού ενδιαφέροντος για την ανάμειξη χημικών. Είχε σχεδιαστεί εξ αρχής να ελέγχεται με δυο τρόπους είτε μέσω ηλεκτρονικού υπολογιστή, είτε μέσω χειριστηρίου. Και οι δυο τρόποι χρησιμοποιούσαν την μετάβαση από σημείο σε σημείο. Οι υπολογιστές της εποχής εκείνης κατασκευάζονταν αποκλειστικά για την κάθε εργασία που απαιτούνταν, συνεπώς ο ρομποτικός βραχίονας δεν είναι άμεσα συμβατός με έναν υπολογιστή γενικού τύπου της σημερινής εποχής.

Η ραγδαία τεχνολογική εξέλιξη καθώς και το δύσχρηστο πρόγραμμα διαχείρισης του ρομποτικού βραχίονα RM-501, δημιούργησαν πολλές προσπάθειες εκσυγχρονισμού του.

Ο ρομποτικός βραχίονας συνοδεύεται από την Μονάδα Διαχείρισης (Drive Unit), που αναλαμβάνει τον έλεγχο και την αποκωδικοποίηση σημάτων για την κίνηση των κινητήρων. Μεγαλύτερη αναφορά πάνω στο Μονάδα Διαχείρισης γίνεται στο Κεφάλαιο 2. Στο παρών κεφάλαιο θα αναφερθούμε στα μηχανικά μέρη και την κινηματική θεωρία του συγκεκριμένου ρομποτικού βραχίονα.



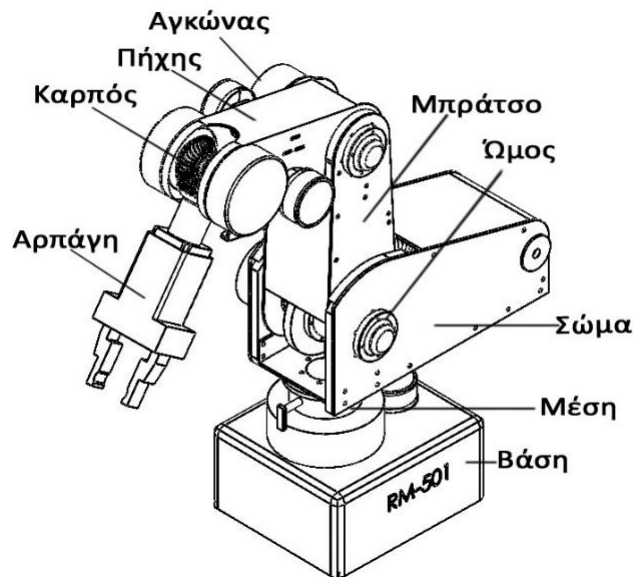
ΣΧΗΜΑ 2.22 Οι βαθμοί ελευθέριας του ρομποτικού βραχίονα RM-501

2.2.1. Μηχανικά Μέρη

Πριν αρχίσουμε οποιαδήποτε αναφορά στα μηχανικά μέρη του ρομπότ, θα θέσουμε κάποια ορολογία για την εύκολη αναφορά και κατανόηση. Θα μπορούσαμε να χωρίσουμε τον ρομποτικό βραχίονα σε δυο βασικά τμήματα όπως φαίνεται στο σχήμα 2.23

- **Τις Αρθρώσεις**
 - Μέση (Waist)
 - Ώμο (Shoulder)
 - Αγκώνα (Elbow)
 - Καρπό (Wrist)

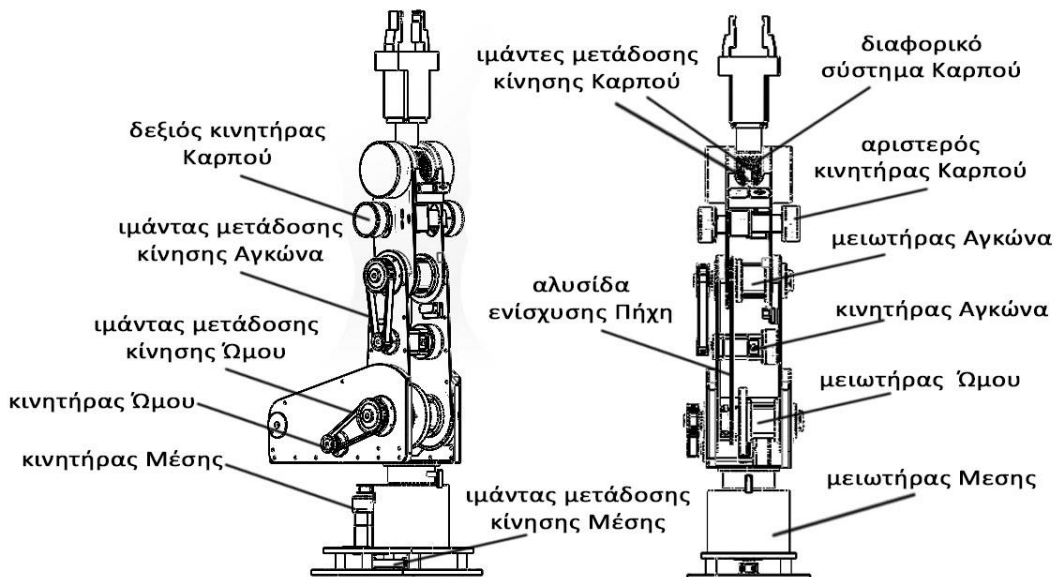
- **και Τα Μέλη**
 - Βάση (Base)
 - Σώμα (Body)
 - Μπράτσο (Upper arm)
 - Πήχη (Forearm)
 - Αρπάγη (Grip)



ΣΧΗΜΑ 2.23 Οι αρθρώσεις και τα μέλη του RM-501

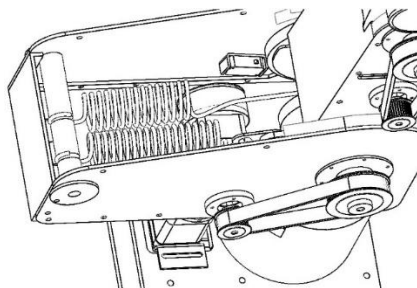
Αφαιρώντας τα καλύμματα, αποκαλύπτεται το εσωτερικό του RM-501. Όπως μπορούμε να παρατηρήσουμε από το σχήμα 2.24 κάθε μέλος «κουβαλάει» τον επενεργητή-κινητήρα της επόμενης άρθρωσης. Οι επενεργητές μεταφέρουν την κίνηση μέσω οδοντωτών ιμάντων στην εκάστοτε άρθρωση. Η κάθε άρθρωση φέρει κατάλληλο μειωτήρα στροφών για την επίτευξη του επιθυμητού λόγου μετάδοσης. Η παρουσία μειωτήρα ενισχύει την ακρίβεια κάθε άρθρωσης καθώς και αποφεύγονται

χαμένα βήματα (miss steps) κατά την λειτουργία. Από την άλλη οι μειωτήρες προσδίδουν βάρος και για το λόγο αυτό απαιτούνται κινητήρες μεγαλύτερης ισχύος για την ανύψωση των μελών του βραχίονα.



ΣΧΗΜΑ 2.24 Εσωτερική επισκόπηση του RM-501

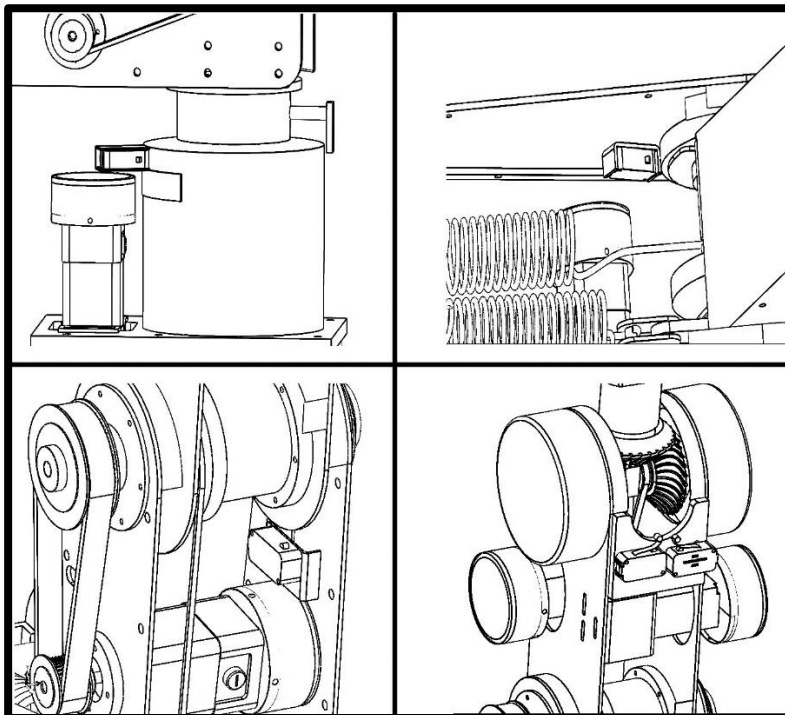
Για την αποφυγή της τοποθέτησης κινητήρων μεγαλύτερης ισχύος, καθώς θα αποτελούσε μια αντιοικονομική λύση και όγκος τους θα απέτρεπε την παραπάνω διάταξη, ο κατασκευαστής χρησιμοποίησε ελατήρια ενίσχυσης (σχήμα 2.25). Είναι τοποθετημένα στο πίσω μέρος του σώματος του βραχίονα. Κάθε ελατήριο αναλαμβάνει να αντισταθμίσει το βάρος του μειωτήρα και των κινητήρων που φέρουν το μπράτσο και πήχη αντίστοιχα. Έτσι ο ρομποτικός βραχίονας μπορεί να διατηρήσει την θέση του ακόμη και όταν οι κινητήρες δεν λειτουργούν. Η ύπαρξη του ελατηρίου του πήχη στο σώμα του βραχίονα και όχι στο μπράτσο δείχνει ανορθόδοξη προσπάθεια εκ πρώτης όψης. Όμως για την διατήρηση της απλότητας κατασκευής καθώς και του μικρού μεγέθους, ο κατασκευαστής μεταφέρει την ενέργεια του ελατηρίου στον πήχη μέσω μιας αλυσίδας όπως φαίνεται στο σχήμα 2.24.



ΣΧΗΜΑ 2.25 Τα ελατήρια ενίσχυσης

Κάτι ακόμη που θα μπορούσαμε να παρατηρήσουμε στο εσωτερικό του βραχίονα είναι οι τερματικοί διακόπτες. Στο σχήμα 2.25 βλέπουμε τους πέντε διακόπτες για τους πέντε βαθμούς ελευθερίας. Όταν

ένας από αυτούς τους διακόπτες ενεργοποιείτε, δηλαδή όταν το μέλος φτάσει σε εκείνο το σημείο όπου πατά τον διακόπτη, τότε το μέλος έχει τερματίσει και δίνεται εντολή στον κινητήρα που ήταν υπεύθυνος για αυτή την κίνηση να σταματήσει. Θα περίμενε κανείς να υπάρχουν δυο διακόπτες για τις δυο ακραίες θέσεις του κάθε μέλους. Παρ'όλ'αυτά για τις άλλες θέσεις ο κατασκευαστής προτίμησε να βάλει φυσικά στοπ (Hard Stops). Τα φυσικά στοπ είναι προληπτικά και αποφεύγεται η κίνηση κοντά σε αυτές τις περιοχές. Έτσι γνωρίζοντας ένα από τα δυο ακραία σημεία του μέλους και το εύρος κίνησης του μπορεί να καθοριστεί ο χώρος δράσης. Οι διακόπτες χρησιμοποιούνται και για την ανίχνευση αρχικής θέσης του ρομποτικού βραχίονα (homing)

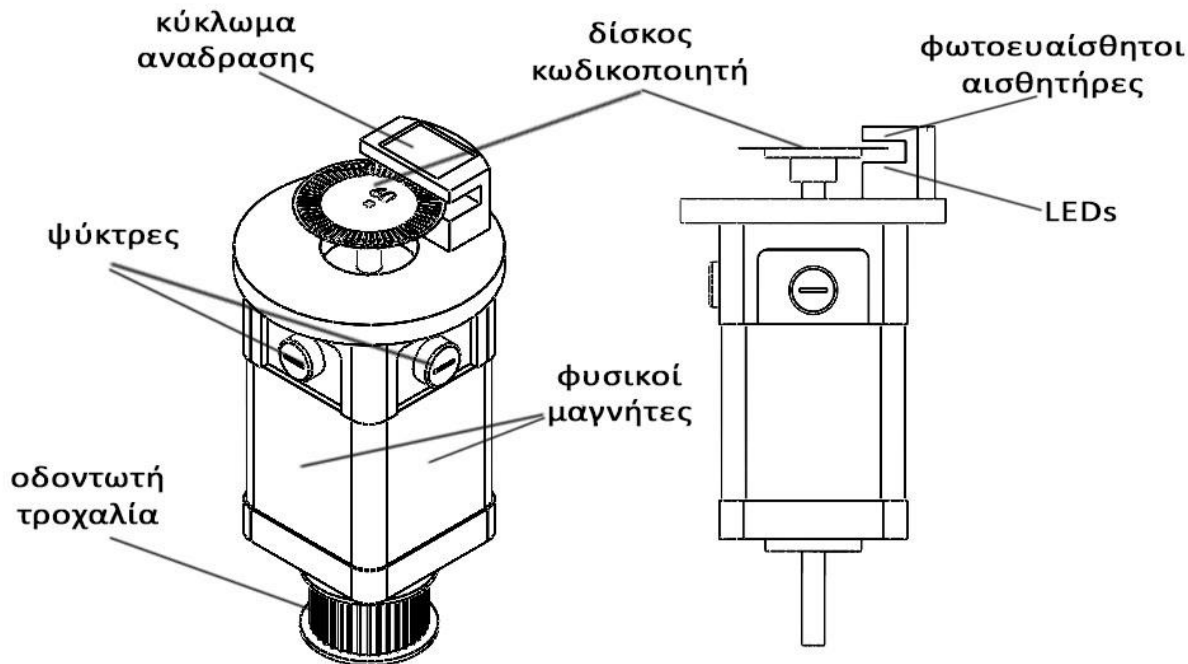


ΣΧΗΜΑ 2.26 Αριστερά πάνω ο διακόπτης της Μέσης, δεξιά πάνω ο διακόπτης Ωμού
Αριστερά κάτω ο διακόπτης του Αγκώνα, δεξιά κάτω οι διακόπτες Περιστροφής και Ανύψωσης

2.2.1.1. Επενεργητές

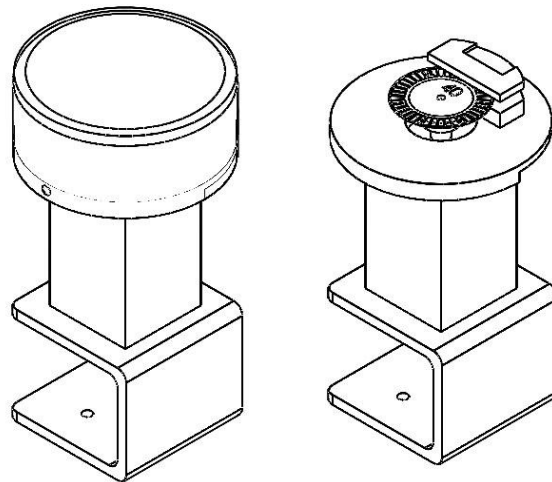
Οι επενεργητές του RM-501 είναι οι κινητήρες που φέρει. Πρόκειται για σερβοκινητήρες συνεχούς ρεύματος (DC) τάσης 24 V με ψήκτρες (brushed DC motor). Στο σχήμα 2.27 βλέπουμε έναν από τους κινητήρες (χωρίς το κάλυμμα κεφαλής), καθώς επίσης και το σύστημα κωδικοποιητή ανάδρασης (encoder). Το εξωτερικό περίβλημα του κινητήρα είναι τέσσερις μόνιμοι μαγνήτες (οι τέσσερις πλευρές) και ο βηματισμός δίνεται από τις ψήκτρες.

Ο κωδικοποιητής ανάδρασης του κινητήρα βασίζεται σε δυο φωτοευαίσθητους αισθητήρες ο ένας δίπλα στον άλλο. Οι αισθητήρες αυτοί όταν ανιχνεύσουν φως κλείνουν το κύκλωμα ανάδρασης, δίνουν δηλαδή λογικό 1 διαφορετικά λογικό 0. Ακριβώς κάτω από τους αισθητήρες βρίσκονται δυο LEDs μόνιμα ανάμενα όσο ο κινητήρας είναι σε λειτουργία. Ο δίσκος κωδικοποιητή (encoder disc) είναι ένας διάτρητος δίσκος τέτοιος ώστε να ενεργοποιεί τους αισθητήρες με φάση $\pi/2$ μεταξύ τους. Όσες περισσότερες είναι οι οπές του δίσκου, τόσο μεγαλύτερη ακρίβεια διόρθωσης του βήματος μπορεί να επιτευχθεί, αφού ένα βήμα θα αντιστοιχεί σε μικρότερο τόξο περιστροφής της τροχαλίας (βλ. σχήμα 2.27) Όταν δίνεται εντολή στον κινητήρα να κινηθεί ένα αριθμό βημάτων, τότε ανάλογα περιστρέφεται και ο δίσκος κωδικοποιητή, με αποτέλεσμα να δημιουργηθούν δυο τετραγωνικά σήματα στο κύκλωμα ανάδρασης. Τα σήματα αυτά είναι οι πληροφορίες για την θέση, την κατεύθυνση και την ταχύτητα του κινητήρα. Εκτενέστερα αυτό το θέμα θα παρουσιαστεί στο Κεφάλαιο 3.



ΣΧΗΜΑ 2.27 Δομή κινητήρα και σύστημα ανάδρασης

Ο παραπάνω κινητήρας χρησιμοποιείται στην κίνηση της Μέσης του Ώμου και του Αγκώνα και έχουν αριθμό οπών στον δίσκο κωδικοποιητή ίσο με 60. Για τον Καρπό χρησιμοποιούνται δυο μικρότεροι σε διαστάσεις κινητήρες με τα ίδια όμως χαρακτηριστικά κ με μόνη διαφορά ότι ο αριθμός οπών του δίσκου κωδικοποιητή είναι 40 (βλ. σχήμα 2.28).

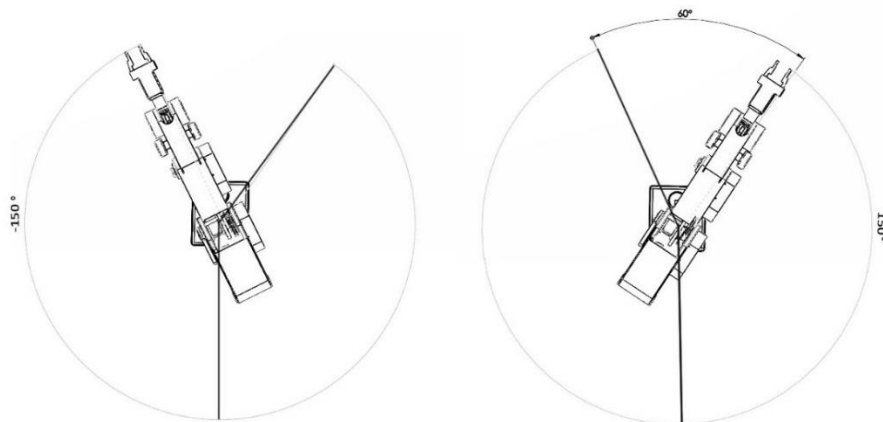


ΣΧΗΜΑ 2.28 Οι κινητήρες του καρπού

2.2.1.2. Αρθρώσεις

Μέση

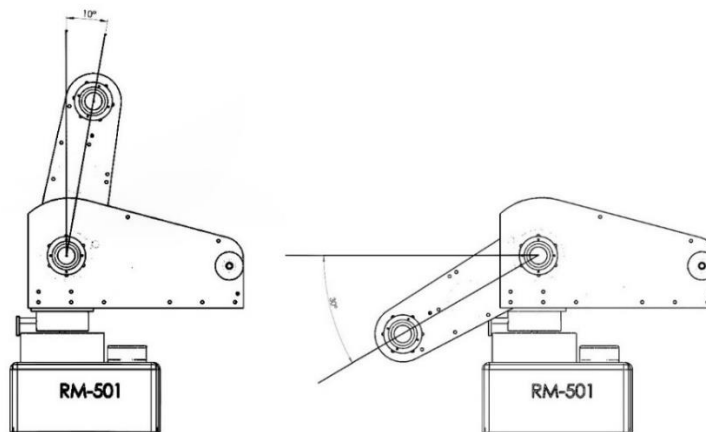
Εμβέλεια Άρθρωσης	300°
Άνω ακραία θέση	150°
Κάτω ακραία θέση	-150°
Αριθμός οδοντών τροχαλίας κινητήρα	20
Αριθμός οδοντών τροχαλίας άρθρωσης	45
Ολική σχέση μετάδοσης	40 steps/1°
Μήκος ιμάντα μετάδοσης	250 mm



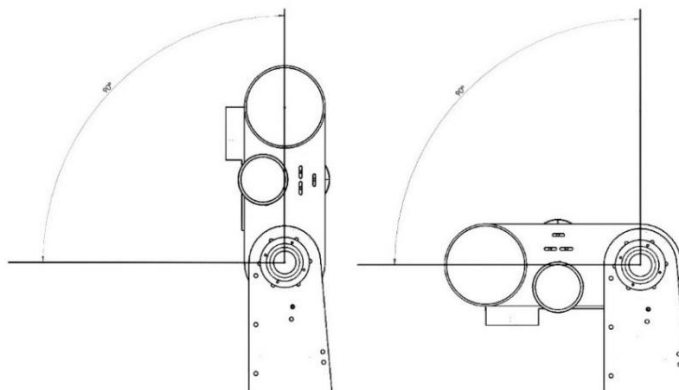
ΣΧΗΜΑ 2.29 Οι δυο ακραίες θέσεις της Μέσης

Ώμος

Εμβέλεια Άρθρωσης	130°
Άνω ακραία θέση	100°
Κάτω ακραία θέση	-30°
Αριθμός οδοντών τροχαλίας κινητήρα	37
Αριθμός οδοντών τροχαλίας άρθρωσης	80
Ολική σχέση μετάδοσης	40 steps/1°
Μήκος ιμάντα μετάδοσης	325mm

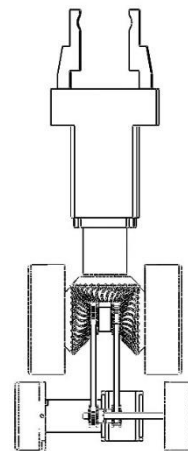
**ΣΧΗΜΑ 2.30** Οι δυο ακραίες θέσεις του Ώμου**Αγκώνας**

Εμβέλεια Άρθρωσης	90°
Άνω ακραία θέση	0°
Κάτω ακραία θέση	-90°
Αριθμός οδοντών τροχαλίας κινητήρα	37
Αριθμός οδοντών τροχαλίας άρθρωσης	80
Ολική σχέση μετάδοσης	40 steps/1°
Μήκος ιμάντα μετάδοσης	325 mm

**ΣΧΗΜΑ 2.31** Οι δυο ακραίες θέσεις του Αγκώνα

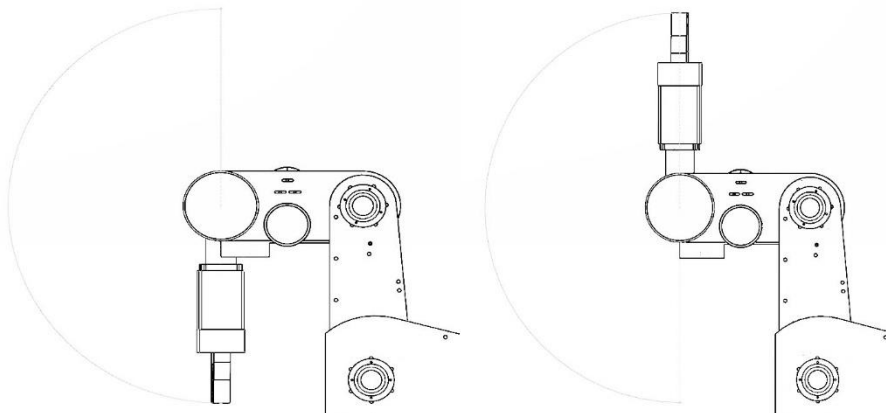
Καρπός

Η ιδιαιτερότητα του καρπού είναι ότι έχει δυο βαθμούς ελευθέριας σε μια άρθρωση. Αυτό επιτυγχάνεται με μια διαφορική διάταξη (βλ. σχήμα 2.32). όταν οι δυο κινητήρες έχουν την ίδια κατεύθυνση περιστροφής τότε ο καρπός κινείται πάνω-κάτω (Ανύψωση) ενώ όταν έχουν διαφορετική κατεύθυνση περιστροφής τότε ο καρπός περιστρέφεται. Οι κινητήρες πρέπει να είναι απολυτά συγχρονισμένοι. Επίσης πρέπει να δοθεί ιδιαίτερη προσοχή έτσι ώστε να μην υπάρχει ταυτόχρονη απαίτηση για ανύψωση και περιστροφή του καρπού μιας και είτε δεν θα προσεγγίσει ποτέ το σημείο στόχο είτε μπορεί να τερματίσει βίαια σε κάποια ακραία θέση. Έτσι λοιπόν οι δυο αυτοί βαθμοί ελευθέριας θα πρέπει να εκτελούνται σειριακά.



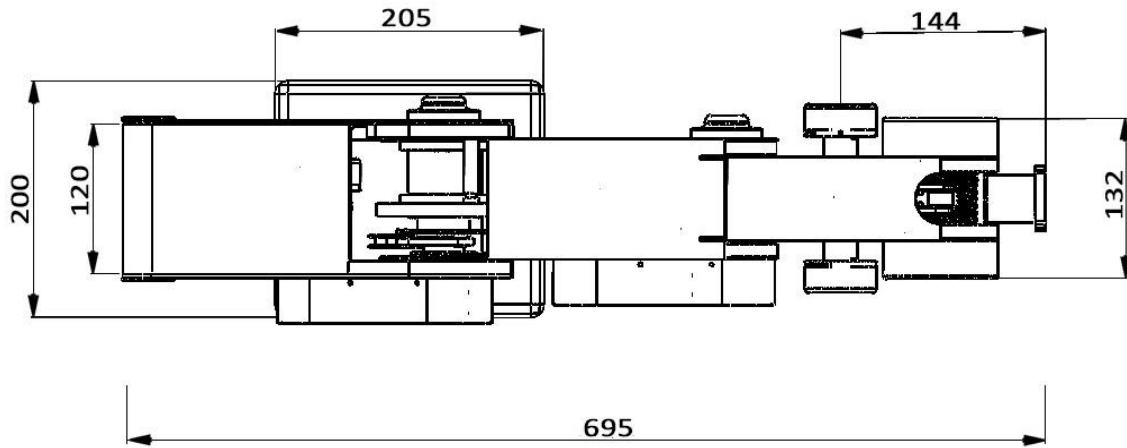
ΣΧΗΜΑ 2.32 Το διαφορικό σύστημα του καρπού

Εμβέλεια Ανύψωσης	180°
Άνω ακραία θέση Ανύψωσης	90°
Κάτω ακραία θέση Ανύψωσης	-90°
Εμβέλεια Περιστροφής	360°
Άνω ακραία θέση Περιστροφής	180°
Κάτω ακραία θέση Περιστροφής	-180°
Αριθμός οδοντών τροχαλίας κινητήρων	20
Αριθμός οδοντών τροχαλίας αρθρώσεων	30
Ολική σχέση μετάδοσης Ανύψωσης	26,66 steps/1°
Μήκος ιμάντων μετάδοσης	217 mm

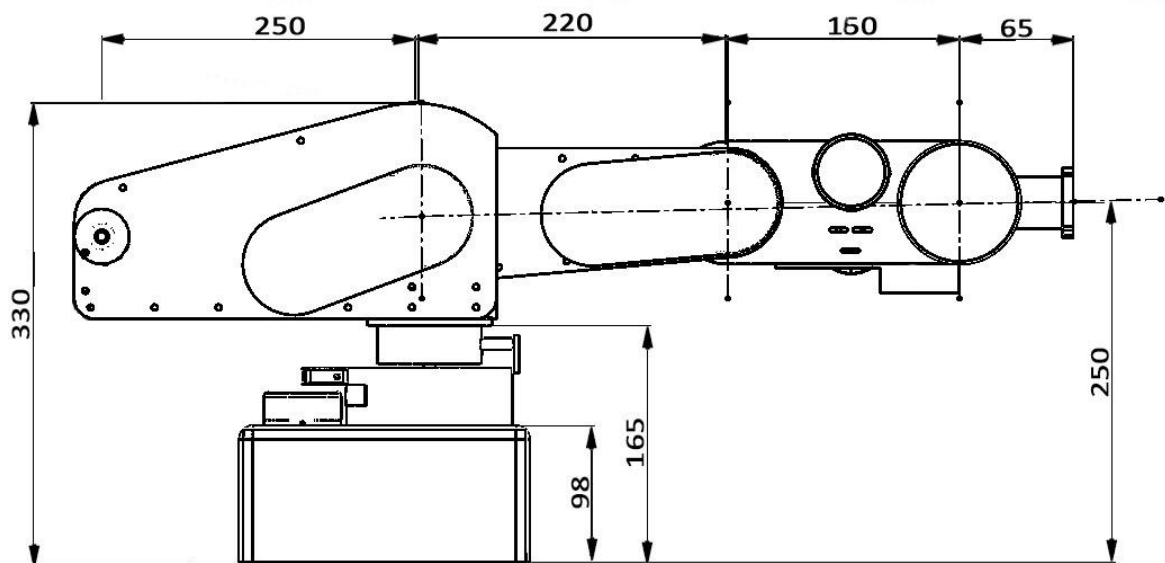


ΣΧΗΜΑ 2.33 Οι δυο ακραίες θέσεις του Καρπού κατά την ανύψωση

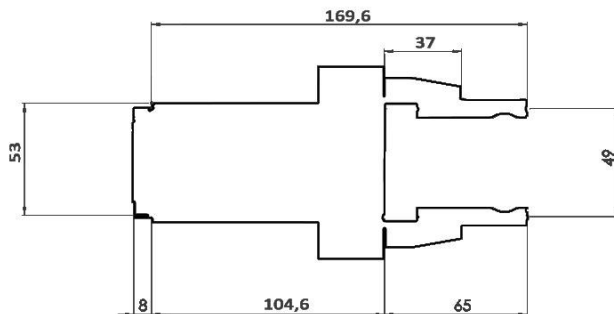
2.2.1.3. Γεωμετρικά χαρακτηριστικά



ΣΧΗΜΑ 2.34 Κάτοψη βραχίονα όταν βρίσκεται σε θέση πλήρης έκτασης



ΣΧΗΜΑ 2.35 Πλάγια όψη βραχίονα όταν βρίσκεται σε θέση πλήρης έκτασης



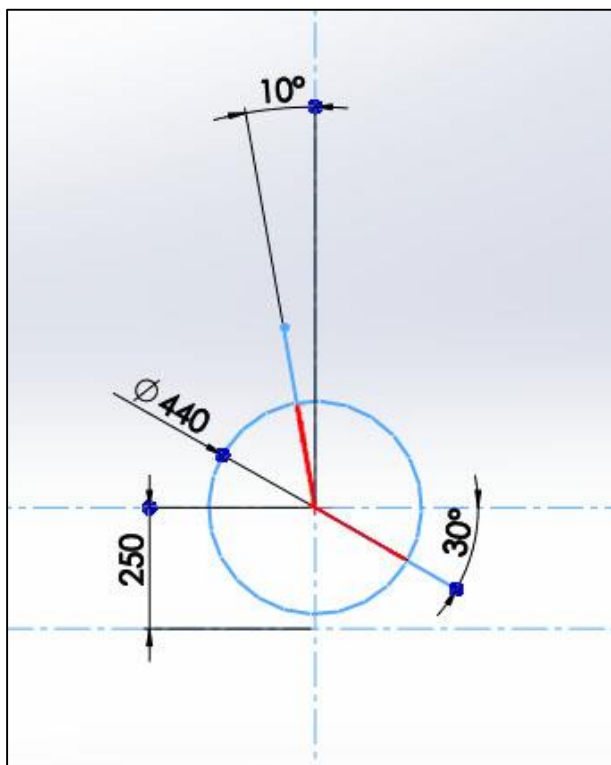
ΣΧΗΜΑ 2.36 Διαστάσεις Αρπάγης

2.2.1.4. Χώρος Εργασίας

Όπως αναφέρθηκε και προηγουμένως ο χώρος εργασίας του ρομποτικού βραχίονα είναι όλα τα πιθανά σημεία που μπορεί να φτάσει το τελικό στοιχείο δράσης. Ο χώρος εργασίας μεταβάλλεται δραματικά ανάλογα με το εργαλείο που προσαρτάται στον ρομποτικό βραχίονα. Ο κατασκευαστής του RM-501 παρουσιάζει στο εγχειρίδιο τον χώρο εργασίας χωρίς την αρπάγη. Έτσι θα πρέπει να υπολογίσουμε τον νέο χώρο εργασίας, με την αρπάγη σαν τελικό στοιχείο δράσης. Παρακάτω παρουσιάζεται μια γεωμετρική προσέγγιση του χώρου εργασίας του ρομποτικού βραχίονα RM-501.

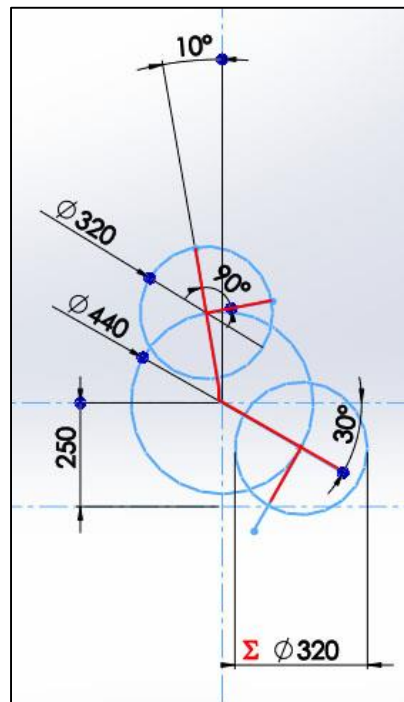
Θεωρούμε ως αρχή των αξόνων την βάση του ρομποτικού βραχίονα, ο άξονας Z είναι κάθετος στο επίπεδο της βάσης και ο άξονας X είναι παράλληλος με την προβολή του Μπράτσου στο επίπεδο βάσης.

Έστω ότι η άρθρωση του ωμού έχει την δυνατότητα περιστροφής 360° τότε θα σχημάτιζε έναν κύκλο που το κέντρο του απέχει από την αρχή των αξόνων 250mm με ακτίνα 220mm όπως φαίνεται στο σχήμα 2.37. Με κόκκινο χρώμα είναι τα μέλη του ρομποτικού βραχίονα σε τερματικές θέσεις. Η άρθρωση όμως μπορεί να κινηθεί 10° αριστερά του άξονα Z και 30° κάτω από τον άξονα X. Πέρα από αυτά τα σημεία το Μπράτσο ακινητοποιείται και αναλαμβάνουν οι υπόλοιπες αρθρώσεις την προσέγγιση των περαιτέρω σημείων.



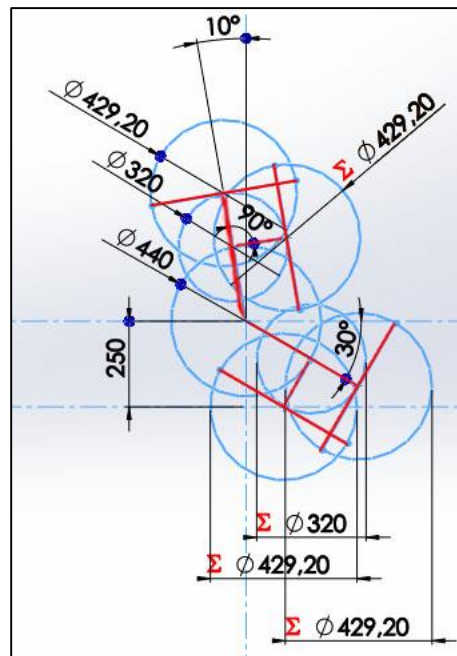
ΣΧΗΜΑ 2.37

Όμοια έστω ότι ο αγώνας, στις θέσεις που έχει τερματίσει το Μπράτσο, έχει την δυνατότητα περιστροφής 360° τότε θα σχημάτιζε κύκλους με ακτίνα 160mm. Όμως η άρθρωση έχει σαν άνω όριο να ευθυγραμμίζεται ο Πήχης με το Μπράτσο και ως κάτω όριο ο πήχης να είναι κάθετος στο Μπράτσο. Πέρα από αυτά τα όρια αναλαμβάνει ο καρπός να προσεγγίσει τα περεταίρω σημεία.



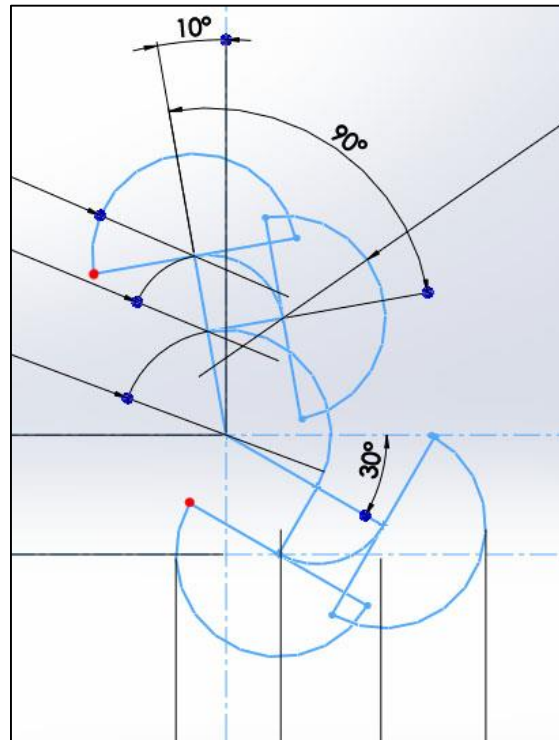
ΣΧΗΜΑ 2.38

Με την ίδια διαδικασία και για τον καρπό προκύπτουν τέσσερις νέοι κύκλοι με ακτίνα 241,6mm που με την σειρά του έχουν σαν όρια τις δυο κάθετες θέσης του καρπού από την πήχη



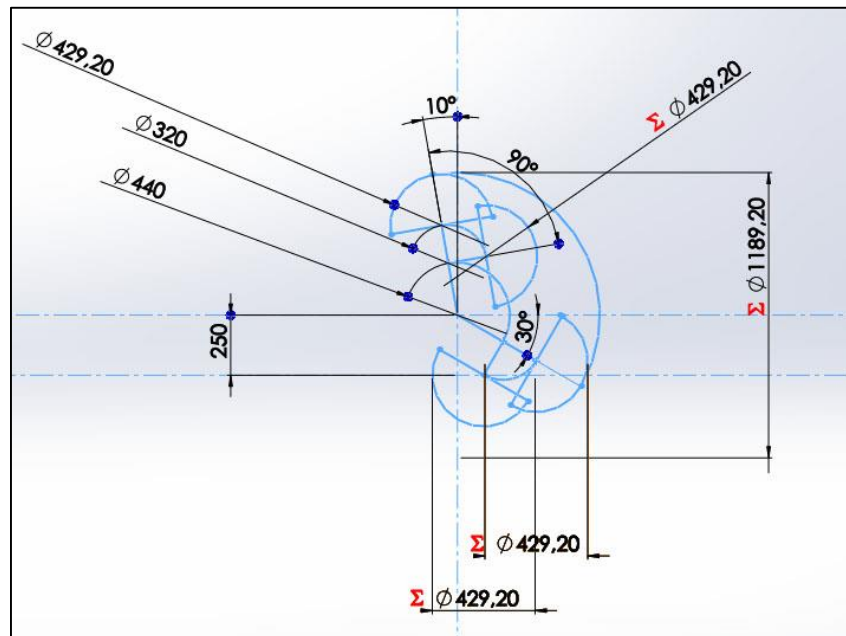
ΣΧΗΜΑ 2.39

Αφαιρώντας τα τόξα των κύκλων που δεν μπορούν να προσεγγιστούν από τα μέλη προκύπτει το παρακάτω σχήμα 2.40. Τα δυο σημεία με κόκκινο χρώμα είναι τα ακρότατα σημεία που μπορεί να προσεγγίσει ο ρομποτικός βραχίονας σε κάποια θέση.



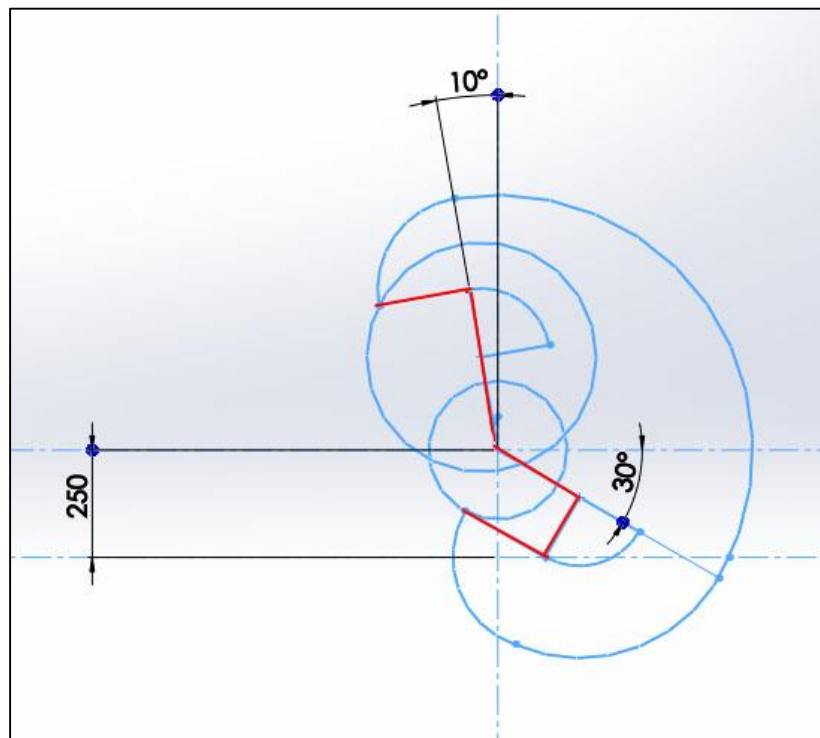
ΣΧΗΜΑ 2.40

Λαμβάνοντας υπόψιν τον τόξο που διαγράφει το αποληκτήριο όταν ο βραχίονας είναι σε πλήρη έκταση (με ακτίνα $220+160+214,6=594,6$) από το ένα άκρο του Ωμου στο άλλο προκύπτει το παρακάτω σχήμα 2.41



ΣΧΗΜΑ 2.41

Αφού ο ωμός τερματίσει στην κάτω θέση ο αγκώνας έχει την δυνατότητα να κινηθεί 90 μοίρες με τον καρπό να εξακολουθεί να είναι σε έκταση. Έτσι το αποληκτήριο διαγραφεί τόξο με ακτίνα $160+214,6=374,6\text{mm}$



ΣΧΗΜΑ 2.42

Στο άνω ακρότατο σημείο πρέπει να εξετάσουμε την περίπτωση όπου ο καρπός μένει στην άνω τελική θέση ενώ μια από τις άλλες δυο αρθρώσεις επιστρέφουν σε άλλη θέση

- κατά την επιστροφή του αγκώνα από την άνω τελική θέση το αποληκτήριο διαγραφεί τόξο με ακτίνα όση η υποτεινούσα του τριγώνου που σχηματίζεται από τον αγκώνα, καρπό και αποληκτήριο
- κατά την επιστροφή του ώμου από την άνω τελική θέση το αποληκτήριο διαγραφεί τόξο με ακτίνα όση η υποκινούσα του τριγώνου που σχηματίζεται από τον ώμο, καρπό και αποληκτήριο

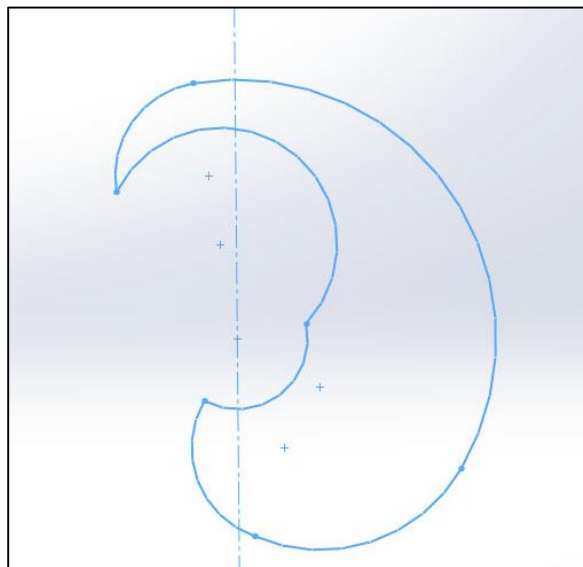
παρατηρούμε ότι το τόξο που αποτελεί όριο του χώρου δράσης είναι αυτό που διαγράφεται όταν κινείται ο Αγκώνας

Στο κάτω ακρότατο σημείο πρέπει να εξετάσουμε την περίπτωση όπου ο καρπός μένει στην κάτω τελική θέση ενώ μια από τις άλλες δυο αρθρώσεις επιστρέφουν σε άλλη θέση

- κατά την επιστροφή του αγκώνα από την κάτω τελική θέση το αποληκτήριο διαγραφεί τόξο με ακτίνα όση η υποκινούσα του τριγώνου που σχηματίζεται από τον αγκώνα, καρπό και αποληκτήριο
- κατά την επιστροφή του ώμου από την κάτω τελική θέση το αποληκτήριο διαγραφεί τόξο με ακτίνα όση η απόσταση του ώμου από το αποληκτήριο

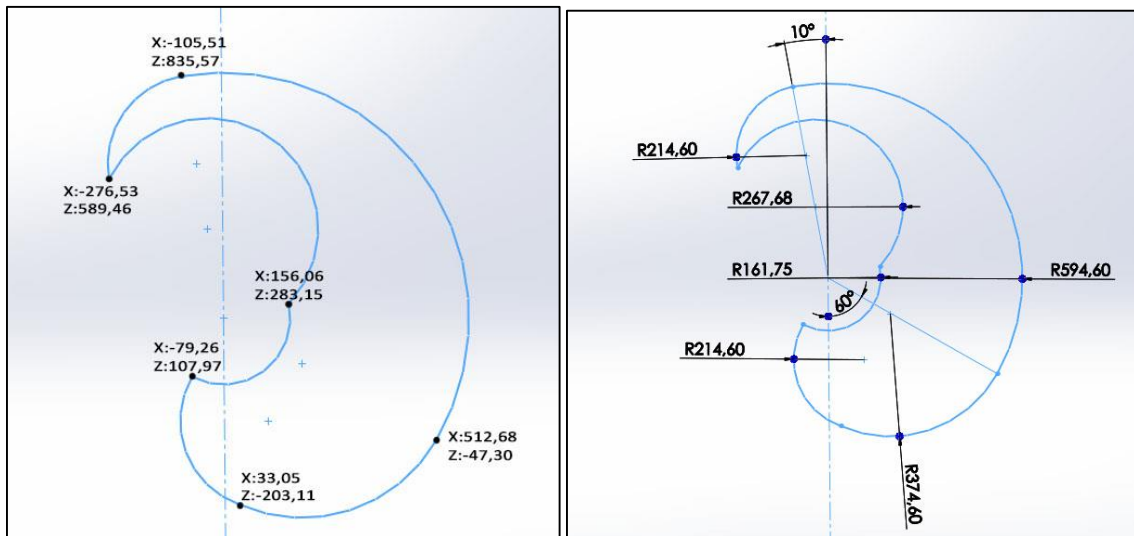
παρατηρούμε ότι το τόξο που αποτελεί όριο του χώρου δράσης είναι αυτό που διαγράφεται όταν κινείται ο Όμος.

Ο χώρος εργασίας του ρομποτικού βραχίονα, με την αρπάγη ως απολυτήριο δράσης, για κάποια γωνία της Μέσης είναι το περίβλημα που δημιουργείται από τα προαναφερθείσα τόξα όπως φαίνεται παρακάτω.



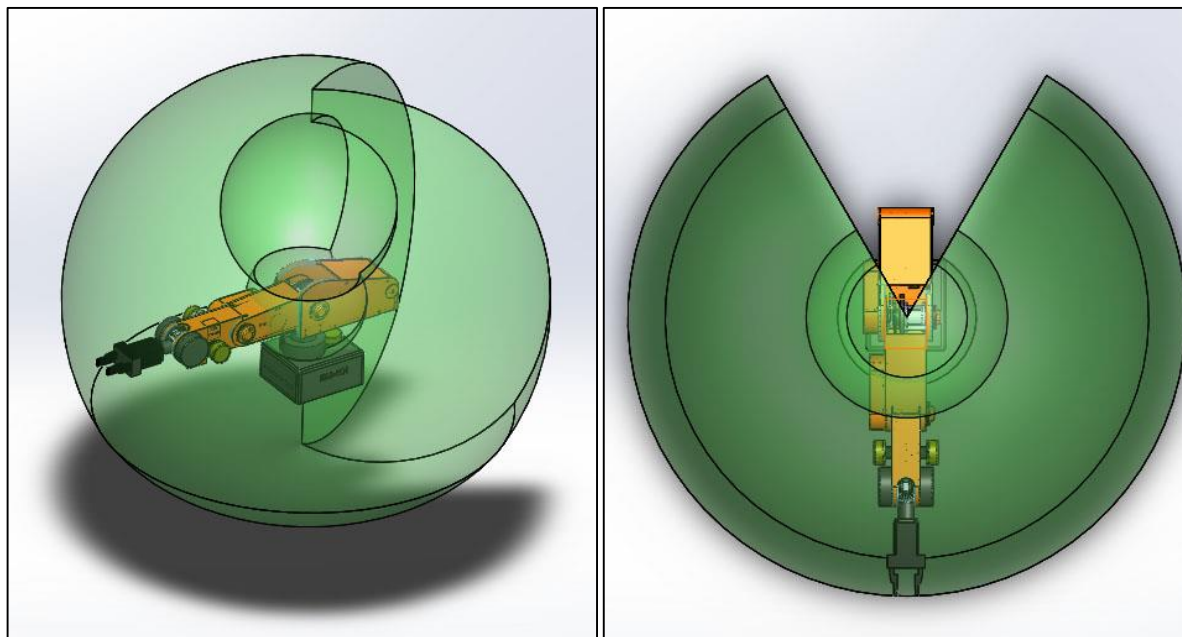
ΣΧΗΜΑ 2.43

Τελικά με την με την βοήθεια προγράμματος σχεδίασης μπορούμε να προσδιορίσουμε τα γεωμετρικά χαρακτηριστικά του χώρου εργασίας σε γωνία Μέσης ίση με 0.



ΣΧΗΜΑ 2.44 Το προφίλ του χωρίου εργασίας με την αρπαγή ως αποληκτήριο

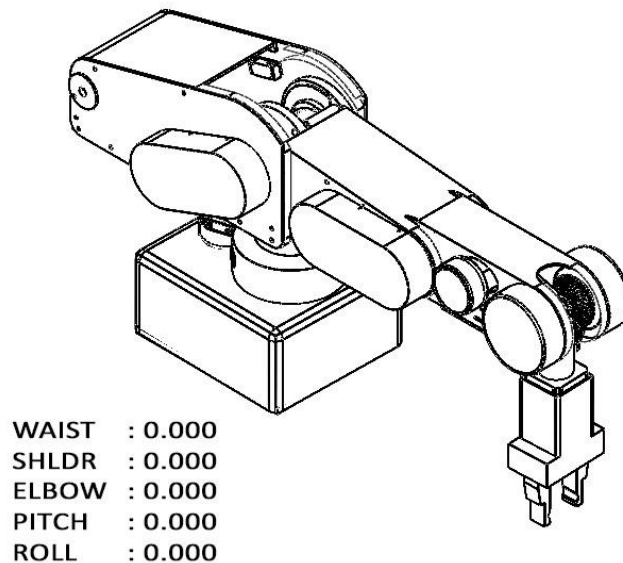
Ο όγκος εργασίας προσιδιάζεται σαν τον χώρο εργασίας για κάθε τιμή της άρθρωσης της Μέσης που δεν ξεπερνά όμως τα όρια του ρομποτικού βραχίονα δηλαδή για $\pm 150^\circ$



ΣΧΗΜΑ 2.45 Ο όγκος εργασίας

2.2.2. Κινηματική θεωρία

Για να προκύψουν όσο το δυνατόν απλούστερες εξισώσεις κατά την επίλυση της κινηματικής θα πρέπει να επιλεγεί μια θέση μηδέν του RM-501 τέτοια ώστε να μηδενίζονται όσος το δυνατόν περισσότεροι παράμετροι Denavit-Hartenberg. Ως θέση μηδέν ορίζεται η θέση όπου όλες οι γωνίες των αρθρώσεων είναι μηδέν. Μια τέτοια θέση φαίνεται στο σχήμα 2.46.



ΣΧΗΜΑ 2.46 Ο ρομποτικός βραχίονας στην θέση μηδέν

Με βάση την γεωμετρία του βραχίονα, την παραπάνω οριζόμενη θέση ως θέση μηδέν και ότι η αρχή των αξόνων του χώρου βρίσκεται στην βάση του βραχίονα έτσι ώστε ο άξονας να Z είναι κάθετος στο επίπεδο της βάσης και ο άξονας X παράλληλος με την προβολή του Μπράτσου στο επίπεδο βάσης, κατασκευάζουμε τον πίνακα με τις παραμέτρους Denavit-Hartenberg

Μέλος	α_i	a_i	d_i	θ_i
1	0	0	d_1	θ_1
2	90°	0	0	θ_2
3	0	a_2	0	θ_3
4	0	a_3	0	θ_4
5	90°	0	d_5	θ_5

Όπως αναφέρθηκε και στο εισαγωγικό μέρος ο αντίστροφος ομογενής μετασχηματισμός από το καρτεσιανό σύστημα "i-1" στο σύστημα "i" είναι ίσος με:

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

Επομένως για κάθε μετάβαση από το μέλος i-1 στο μέλος i έχω:

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

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

$${}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Ως c_i, s_i με $i=1, 2, 3, 4, 5$ ορίζονται το $\cos(\theta_i)$ και $\sin(\theta_i)$ αντίστοιχα

2.2.2.1. Ευθεία Κινηματική

Ισχύει ότι

$${}^0_5T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T$$

Παρατηρούμε ότι τα μέλη 2, 3 και 4 έχουν τους άξονες παράλληλους συνεπώς υπολογίζουμε αρχικά τον πίνακα 1_3T ,

$${}^1_3T = {}^1_2T {}^2_3T = \begin{bmatrix} c_{23} & -s_{23} & 0 & c_2 a_2 \\ 0 & 0 & -1 & 0 \\ s_{23} & c_{23} & 0 & s_2 a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου από τις τριγωνομετρικές ταυτότητες έχω,

$$c_{23} = c_2 c_3 - s_2 s_3 = \cos(\theta_2 + \theta_3)$$

$$s_{23} = s_2 c_3 + c_2 s_3 = \sin(\theta_2 + \theta_3)$$

όμοια,

$${}^1_4T = {}^1_3T {}^3_4T = \begin{bmatrix} c_{234} & -s_{234} & 0 & c_{23} a_3 + c_2 a_2 \\ 0 & 0 & -1 & 0 \\ s_{234} & c_{234} & 0 & s_{23} a_3 + s_2 a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου,

$$c_{234} = c_{23} c_4 - s_{23} s_4$$

$$s_{234} = s_{23} c_4 + c_{23} s_4$$

Θα υπολογίσουμε αρχικά τον πίνακα 1_5T γιατί θα φανεί χρήσιμος στους υπολογισμούς της αντίστροφης κινηματικής

$${}^1_5T = {}^1_4T {}^4_5T = \begin{bmatrix} c_5 c_{234} & -s_5 c_{234} & s_{234} & d_5 s_{234} + c_{23} a_3 + c_2 a_2 \\ -s_5 & -c_5 & 0 & 0 \\ c_5 s_{234} & -s_5 s_{234} & -c_{234} & -d_5 c_{234} + s_{23} a_3 + s_2 a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ο τελικός πίνακας θα έχει την μορφή,

$${}^0_5T = {}^0_1T {}^1_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου,

$$r_{11} = c_5 c_1 c_{234} + s_5 s_1$$

$$r_{12} = -s_5 c_1 c_{234} + c_5 s_1$$

$$r_{13} = c_1 s_{234}$$

$$r_{21} = c_5 s_1 s_{234} - s_5 c_1$$

$$r_{22} = -s_5 s_1 s_{234} - c_5 c_1$$

$$r_{23} = s_1 s_{234}$$

$$r_{31} = c_5 s_{234}$$

$$r_{32} = -s_5 s_{234}$$

$$r_{33} = -c_{234}$$

$$p_x = d_5 c_1 s_{234} + a_3 c_1 c_{23} + a_2 c_1 c_2$$

$$p_y = d_5 s_1 s_{234} + a_3 s_1 c_{23} + a_2 s_1 c_2$$

$$p_z = d_1 - d_5 c_{234} + a_3 s_{23} + a_2 s_2$$

2.2.2.2. Αντίστροφη κινηματική

Για την αντιστροφή κινηματική γνωρίζω τον πίνακα

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Και θέλω να υπολογίσω τις γωνίες για κάθε άρθρωση

Από την ευθεία κινηματική έχω,

$${}^0_5T = {}^0_1T(\theta_1) {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T(\theta_5)$$

Σύμφωνα με τις ιδιότητες πινάκων ισχύει,

$$[{}^0_1T(\theta_1)]^{-1} {}^0_5T = {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T(\theta_5)$$

Και όπως αποδείχτηκε για τους ομογενείς πίνακες ισχύει,

$$[{}^0_1T(\theta_1)]^T {}^0_5T = {}^1_5T$$

$$\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_5 c_{234} & -s_5 c_{234} & s_{234} & d_5 s_{234} + c_{23} a_3 + c_2 a_2 \\ -s_5 & -c_5 & 0 & 0 \\ c_5 s_{234} & -s_5 s_{234} & -c_{234} & -d_5 c_{234} + s_{23} a_3 + s_2 a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Εξισώνοντας το στοιχείο (2,4) του πρώτου μέλους με το αντίστοιχο του δευτέρου μέλους έχω

$$-s_1 p_x + c_1 p_y = 0$$

Θέτω,

$$p_x = \rho \cos \varphi$$

$$p_y = \rho \sin \varphi$$

Όπου

$$\rho = \sqrt{p_x^2 + p_y^2}$$

$$\varphi = \text{Atan2}(p_y, p_x)$$

Έτσι

$$-s_1 c_\varphi + c_1 s_\varphi = 0$$

Από τις τριγωνομετρικές ταυτότητες προκύπτει

$$\sin(\varphi - \theta_1) = 0$$

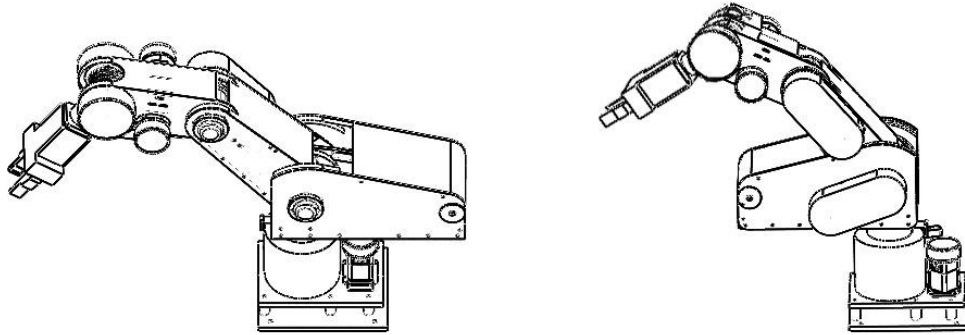
$$\cos(\varphi - \theta_1) = \pm 1$$

$$\varphi - \theta_1 = \text{Atan2}(0, \pm 1)$$

$$\theta_1 = \text{Atan2}(p_y, p_x) - \pi$$

$$\theta_1 = \text{Atan2}(p_y, p_x) + \pi$$

Παρατηρούμε ότι υπάρχουν δυο λύσεις. Όμως τα όρια του βραχίονα μας επιτρέπουν μια από αυτές κάθε φορά, ανάλογα με την τελική θέση του εργαλείου



ΣΧΗΜΑ 2.47 Δυο συμμετρικές προσεγγίσεις από την Μέση. Η δεξιά απορρίπτεται
Εξισώνοντας τα στοιχεία (1, 3) και (3, 3) του πρώτου μέλους με το δεύτερο έχω

$$s_{234} = c_1 r_{13} + c_1 r_{23}$$

$$c_{234} = -r_{33}$$

$$\theta_{234} = \text{Atan2}(c_{234}, s_{234})$$

Με γνωστά τα s_{234} και c_{234} εξισώνοντας τα στοιχεία (1, 4) (3, 4) προκύπτει

$$s_{23}a_3 + s_2a_2 = k_1$$

$$c_{23}a_3 + c_2a_2 = k_2$$

Όπου k_1 και k_2 γνωστές ποσότητες

$$k_1 = d_5 c_{234} + p_z$$

$$k_2 = c_1 p_x + s_1 p_y - d_5 s_{234}$$

Προσθέτοντας τα τετράγωνα των k_1 και k_2 προκύπτει

$$k_1^2 + k_2^2 = 2a_2a_3c_3 + a_3^2 + a_2^2$$

Με μόνο άγνωστο το $\cos(\theta_3)$ εξισώνοντας ως προς αυτόν έχω,

$$c_3 = \frac{k_1^2 + k_2^2 - a_3^2 - a_2^2}{2a_2a_3}$$

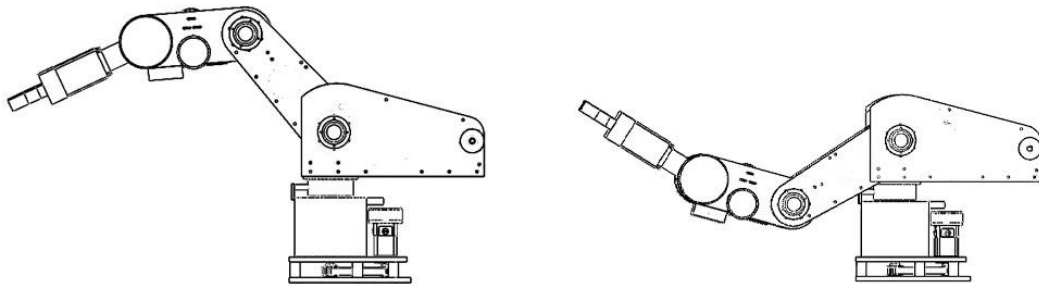
Από τις τριγωνομετρικές ταυτότητες προκύπτει και το ημίτονο

$$s_3 = \pm\sqrt{1 - c_3^2}$$

$$\theta_3 = \text{Atan2}(c_3, \pm s_3)$$

Ο βραχίονας στον σύνδεσμο του αγκώνα έχει την δυνατότητα να κινηθεί μέχρι την ευθεία του ώμου έτσι η συμμετρική κάτω λύση απορρίπτεται. Επομένως

$$\theta_3 = \text{Atan2}(c_3, -s_3)$$



ΣΧΗΜΑ 2.48 Δυο συμμετρικές προσεγγίσεις από την Μέση. Η δεξιά απορρίπτεται

Από την $c_{23}a_3 + c_2a_2 = k_2$ αναλύοντας την τριγωνομετρική ταυτότητα προκύπτει

$$c_2(c_3a_3 + a_2) - s_2(s_3a_3) = k_2$$

Θέτω,

$$(s_3a_3) = \rho \cos \varphi$$

$$(c_3a_3 + a_2) = \rho \sin \varphi$$

Όπου

$$\rho = \sqrt{(c_3a_3 + a_2)^2 + (s_3a_3)^2}$$

$$\varphi = \text{Atan2}((c_3a_3 + a_2), (s_3a_3))$$

Έτσι

$$-s_2c_\varphi + c_2s_\varphi = k_2/\rho$$

Από τις τριγωνομετρικές ταυτότητες προκύπτει

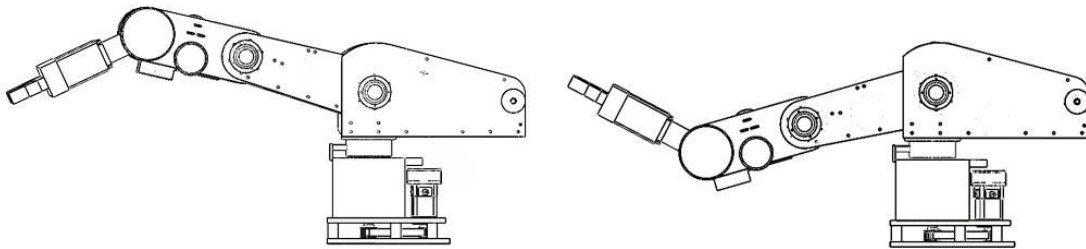
$$\sin(\varphi - \theta_2) = k_2/\rho$$

$$\cos(\varphi - \theta_2) = \pm \sqrt{1 - \frac{k_2^2}{\rho^2}}$$

$$\varphi - \theta_2 = \text{Atan2}\left(\frac{k_2}{\rho}, \pm \sqrt{1 - \frac{k_2^2}{\rho^2}}\right)$$

$$\theta_2 = \text{Atan2}((c_3 a_3 + a_2), (s_3 a_3)) - \text{Atan2}(k_2, \pm \sqrt{(c_3 a_3 + a_2)^2 + (s_3 a_3)^2 - k_2^2})$$

Παρατηρούμε ότι και εδώ έχουμε δυο συμμετρικές λύσεις. Οι λύσεις αυτές αναφέρονται σε όλες τις θέσεις από εκεί που ο αγκώνας είναι σε πλήρη έκταση έως να επανέλθει σε πλήρη έκταση. Ο καρπός αναλαμβάνει να δώσει τις λύσεις αυτές



ΣΧΗΜΑ 2.49 Οι δυο συμμετρικές ακραίες προσεγγίσεις από τον Ώμο.

Με γνωστές τις γωνίες θ_{234} , θ_3 , θ_2 έχουμε

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3$$

Εξισώνοντας τα στοιχεία (1, 2) και (2, 2) του πρώτου μέλους με το δεύτερο έχω

$$s_5 = s_1 r_{11} - c_1 r_{12}$$

$$c_5 = s_1 r_{21} - c_1 r_{22}$$

$$\theta_5 = \text{Atan2}(s_5, c_5)$$

Στην περίπτωση της περιστροφής του καρπού λόγω της συμμετρικότητας που παρουσιάζει η αρπάγη, η περιστροφή 90 είναι αρκετή για να καθορίσει όλες τις πιθανές θέσεις.

ΚΕΦΑΛΑΙΟ 3. Ηλεκτρονική και Έλεγχος

3.1. Εισαγωγή στα ηλεκτρονικά

Η ηλεκτρονική είναι ένας κλάδος της επιστήμης της φυσικής, όσον αφορά τη θεωρητική μελέτη, και το κύριο γνωστικό αντικείμενο της επιστήμης ηλεκτρονικού μηχανικού, όσον αφορά την πρακτική εφαρμογή για την επίλυση τεχνολογικών προβλημάτων. Αφορά τη διερεύνηση της λειτουργίας και κατασκευής κυκλωμάτων και συσκευών που λειτουργούν με τον έλεγχο ροής ηλεκτρονίων και άλλων ηλεκτρικά φορτισμένων σωμάτων, σε συσκευές όπως οι ηλεκτρονικές λυχνίες και οι ημιαγωγοί.

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.

Ένας απλός τρόπος να μελετήσουμε ένα ηλεκτρονικό σύστημα είναι να το χωρίσουμε στα ακόλουθα:

- Είσοδος - Ηλεκτρικοί ή μηχανικοί αισθητήρες (ή μορφοτροπίες), παίρνουν τα σήματα (ηλεκτρικά ή μαγνητικά κύματα, ή μετρήσεις θερμοκρασίας, πίεσης κλπ.) από τον φυσικό κόσμο και τα μετατρέπουν σε ηλεκτρικό ρεύμα.
- Κυκλώματα επεξεργασίας σημάτων - Αποτελούνται από ηλεκτρονικά εξαρτήματα διασυνδεδεμένα μαζί για να επεξεργάζονται, μετατρέπουν και μεταφράζουν ηλεκτρικά σήματα.
- Έξοδος - Μορφοποιητές που μετατρέπουν το ηλεκτρικό ρεύμα/τάση σε χρήσιμη (για τον άνθρωπο) μορφή (bits πληροφορίας, εικόνα, ήχο, κίνηση κλπ.)

Για παράδειγμα μια τηλεόραση έχει σαν είσοδο το σήμα εκπομπής από την κεραία ή το σήμα από κάποιο καλωδιακό σύστημα. Διάφορα κυκλώματα επεξεργασίας σημάτων μέσα στην τηλεόραση αναπαράγουν τις πληροφορίες για την φωτεινότητα, χρώμα και ήχο από το σήμα. Η έξοδος γίνεται στην οθόνη η οποία αναπαράγει το είδωλο και ένα μεγάφωνο αναπαράγει τον ήχο από το ηλεκτρικά σήματα.

Στην παρούσα ενότητα θα γίνει αναφορά μόνο στα ζητήματα που θα μας απασχολήσουν στην δεύτερη ενότητα του κεφαλαίου 2.

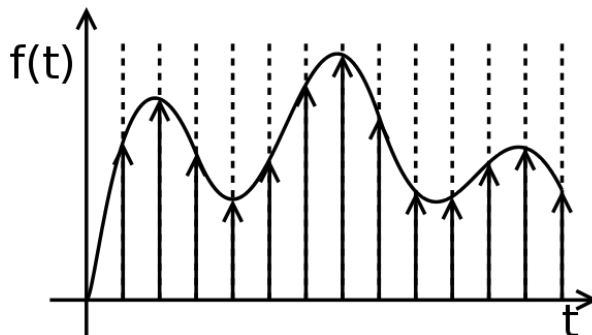
3.1.1. Βασικές έννοιες

3.1.1.1. Ψηφιακό Σήμα

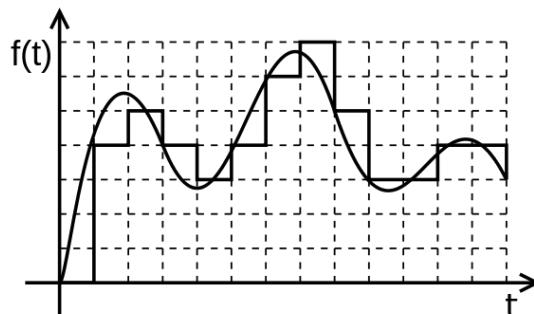
Ο όρος ψηφιακό σήμα αναφέρεται σε περισσότερες από μια έννοιες. Μπορεί να αναφέρεται σε ένα σήμα διακριτού χρόνου το οποίο μπορεί να πάρει συγκεκριμένες (διακριτές) τιμές στον άξονα του χρόνου. Σε αυτή τη περίπτωση μιλάμε για ένα σήμα το οποίο παράγεται μέσω μιας μεθόδου ψηφιακής διαμόρφωσης και θεωρείται περισσότερο ως αναλογικό σήμα (δηλ. ένα επεξεργασμένο αναλογικό σήμα για τη μετατροπή του σε ψηφιακό).

Μπορεί επίσης να αναφέρεται στην κυματομορφή ενός σήματος συνεχούς χρόνου σε ένα ψηφιακό σύστημα το οποίο μπορεί να αναπαρασταθεί σε μια αλληλουχία από bits. Σε αυτή τη περίπτωση μιλάμε για ένα εξ ολοκλήρου ψηφιακό σήμα.

Σήματα Διακριτού Χρόνου



ΣΧΗΜΑ 3.1 Ένα σήμα διακριτού χρόνου



ΣΧΗΜΑ 3.2 Ψηφιακό σήμα

Ένα αναλογικό σήμα είναι μια χρονικά μεταβαλλόμενη τιμή δεδομένων ή όπως είναι ο ορισμός του, μια ομαλά μεταβαλλόμενη τιμή ηλεκτρικής τάσης ή έντασης ρεύματος (δηλ. ένα σήμα με πλάτος χρονικά μεταβαλλόμενο) η οποία μπορεί να περιγραφεί από μια μαθηματική συνάρτηση, με το χρόνο να αποτελεί την ανεξάρτητη και την τιμή του σήματος, κάθε χρονική στιγμή, την εξαρτημένη μεταβλητή. Ένα διακριτό σήμα είναι το αποτέλεσμα που παίρνουμε μέσω της μεθόδου της

δειγματοληπτικής μείωσης από το αρχικό αναλογικό σήμα: δηλαδή, η τιμή των δεδομένων σημειώνεται ανά τακτά χρονικά διαστήματα (π.χ. μικροδευτερόλεπτο) και όχι συνεχώς (όπως είναι εκ φύσεως τα μηχανικά κύματα).

Αν οι ατομικές τιμές του σήματος αντί να μετρηθούν επακριβώς, επάνω στον άξονα του χρόνου, είναι εναρμονισμένες με κάποια ορισμένη ακρίβεια, τότε η ροή δεδομένων που προκύπτει είναι το ψηφιακό σήμα. Η διαδικασία προσέγγισης αυτής της ακρίβειας (δηλ. μιας συγκεκριμένης τιμής), μέσα από ένα σταθερό αριθμό ψηφίων (δηλ. bit) ονομάζεται ψηφιοποίηση.

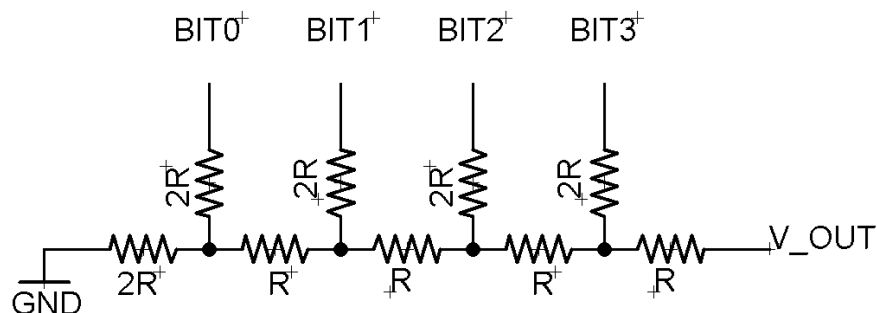
Σε γενικές γραμμές, ένα ψηφιακό σήμα είναι ένα ψηφιοποιημένο σήμα διακριτού χρόνου. Το διακριτό σήμα είναι το αποτέλεσμα της επεξεργασίας ενός αναλογικού σήματος με τη μέθοδο της δειγματοληπτικής μείωσης.

Σχεδόν σε όλες τις εφαρμογές, τα ψηφιακά σήματα αναπαριστώνται μέσω των δυαδικών αριθμών, έτσι ώστε να μπορούν να μετρηθούν σε bit. Επειδή 7 bits (δυαδικά ψηφία) μπορούν να καταγράψουν 128 διακριτές τιμές (0 έως 127), αυτό το σύστημα είναι το πιο ικανό από οποιοδήποτε άλλο για να εκφραστεί ένα τεράστιο πλήθος τιμών.

Μετατροπή Ψηφιακού σήματος σε Αναλογικό (DAC)

Υπάρχουν πολλές διατάξεις για την μετατροπή του αναλογικού σήματος σε ψηφιακό. Οι περισσότερες όμως είναι παραλλαγές της κλίμακας αντιστάσεων R-2R (resistor ladder R-2R).

Στο σχήμα 3.3 παρουσιάζεται μια κλίμακα αντιστάσεων τεσσάρων bit. Κάθε bit έχει την δική του βαρύτητα και συμβάλει στην μεταβολή της τάσης εξόδου. Προφανώς κάθε είσοδος μπορεί να πάρει δυο τιμές είτε 0V είτε την ονομαστική τάση του κυκλώματος V_{ref} . Αναφέρθηκε προηγουμένως ότι τέσσερα bit μπορούν να δώσουν $2^4 = 16$ συνδυασμούς. Έτσι η τάση στην έξοδο της κλίμακας V_{out} μπορεί να πάρει 16 διαφορετικές τιμές



ΣΧΗΜΑ 3.3 Κλίμακα R2R

Το bit 0 είναι το πρώτο ψηφίο (LSB-Least Significant Bit) ενώ το bit 3 είναι το τελευταίο ψηφίο (MSB Most Significant Bit)

Η τάση εξόδου υπολογίζεται ως εξής:

$$V_{out} = V_{ref} \frac{X}{2^N}$$

Οπού

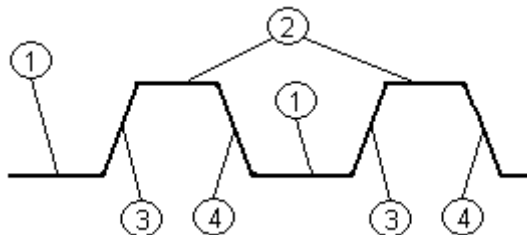
- N είναι ο αριθμός των εισόδων της σκάλας
- X είναι ο αριθμός που δημιουργούν τα N bits σε δεκαδικό σύστημα

Έτσι αν για την παραπάνω κλίμακα (4-bit) δίνεται ως είσοδος 1011 στο δεκαδικό σύστημα είναι ο αριθμός 11 και έστω ότι το κύκλωμα λειτουργεί με ρεύμα τάσης 5V τότε η τάση στην έξοδο της σκάλας θα είναι

$$V_{out} = 5 \frac{11}{2^4} = 3.43 V$$

Η κυματομορφή στα ψηφιακά συστήματα

Τόσο στους υπολογιστές όσο και σε οποιοδήποτε άλλο ψηφιακό σύστημα, η κυματομορφή του σήματος εναλλάσσεται μεταξύ δύο επιπέδων τάσης (0 και 4,8V) οι οποίες αναπαριστούν αντίστοιχα τις δύο τιμές του δυαδικού συστήματος (0 και 1). Έτσι, αναφερόμαστε σε αυτή τη κυματομορφή ως **ψηφιακό σήμα**. Παρ' όλο που πρόκειται για μια αναλογική κυματομορφή τάσεως, το ονομάζουμε ψηφιακό διότι εναλλάσσεται μεταξύ δύο σταθερών καταστάσεων.



ΣΧΗΜΑ 3.4 Η κυματομορφή ενός ψηφιακού σήματος: (1) χαμηλό επίπεδο τάσης, (2) υψηλό επίπεδο, (3) μετάβαση σε υψηλό επίπεδο τάσης, (4) μετάβαση σε χαμηλό επίπεδο

Το σήμα του ρολογιού χρονισμού είναι ένα ειδικό ψηφιακό σήμα το οποίο χρησιμοποιείται για τον συγχρονισμό των ψηφιακών κυκλωμάτων. Στο σχήμα () φαίνεται η συγκεκριμένη κυματομορφή. Οι λογικές αλλαγές ενεργοποιούνται είτε από την αύξηση είτε από την μείωση του σήματος. Στο σχήμα () βλέπουμε ένα παράδειγμα του πρακτικού παλμού. Έτσι, όταν λέμε:

- **Αύξηση του σήματος** : εννοούμε την διαδικασία της μετάβασης από χαμηλή σε υψηλή τάση.
- **Μείωση του σήματος** : εννοούμε την διαδικασία της μετάβασης από υψηλή σε χαμηλή τάση.

3.1.2. Συνήθειες Διατάξεις Ηλεκτρονικών

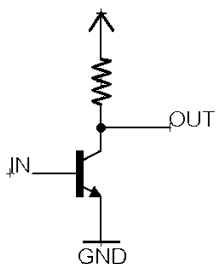
3.1.2.1. Λογικές Πύλες

Χρησιμοποιώντας διπολικά NPN τρανζίστορ σαν διακόπτες μπορούμε να δημιουργήσουμε διατάξεις τέτοιες ώστε κάνουμε πράξεις. Το ηλεκτρικό κύκλωμα που εκτελεί μία λογική πράξη ονομάζεται λογική ή ψηφιακή πύλη και παριστάνεται από ένα σύμβολο. Τα δυαδικά ψηφία 1 και 0, που ουσιαστικά παριστάνουν τις δύο καταστάσεις αληθής (true-T), ψευδής (false-F), στη φυσική τους υπόσταση είναι δυο διακριτά επίπεδα ηλεκτρικής τάσης (συνήθως στην ιδανική περίπτωση 5V και 0V).

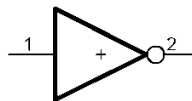
Λογική πύλη NOT

Η λογική πύλη NOT δημιουργεί αντιστροφή του σήματος εισόδου

ΚΥΚΛΩΜΑ ΠΥΛΗΣ



ΛΟΓΙΚΟ ΣΥΜΒΟΛΟ



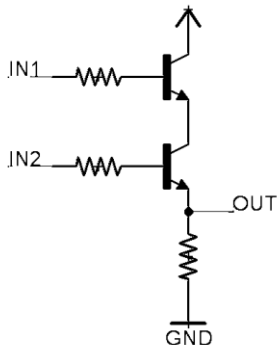
ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

IN	OUT
T	F
F	T

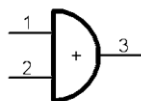
Λογική πύλη AND

Η έξοδος είναι αληθής (1), όταν και οι δυο εισοδοι είναι αληθείς (1)

ΚΥΚΛΩΜΑ ΠΥΛΗΣ



ΛΟΓΙΚΟ ΣΥΜΒΟΛΟ



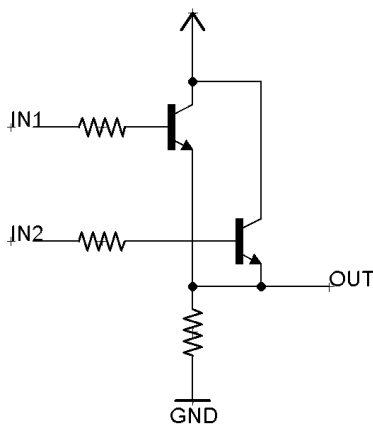
ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

IN1	IN2	OUT
F	F	F
T	F	F
F	T	F
T	T	T

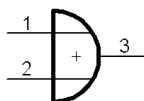
Λογική πύλη OR

Η έξοδος είναι αληθής (true) (1), εάν μια από τις εισόδους ή και οι δυο είναι αληθείς (1)

ΚΥΛΩΜΑ ΠΥΛΗΣ ΣΥΜΒΟΛΟ



ΛΟΓΙΚΟ



ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

IN1	IN2	OUT
F	F	F
T	F	T
F	T	T
T	T	T

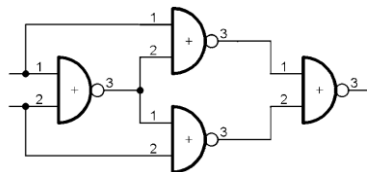
Παρατηρούμε στα κυκλώματα των παραπάνω λογικών πυλών ότι αν η έξοδος είναι από την μεριά της τροφοδοσίας της τάσης και όχι από την γείωση τότε προκύπτουν δυο ακόμη πύλες οι NAND και NOR.

Λογική πύλη XOR

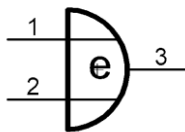
Η έξοδος είναι αληθής (1), όταν ή μία εκ των δύο εισόδων είναι αληθής (1), αλλά όχι και οι δύο ταυτόχρονα:

Η λογική πύλη XOR δημιουργείται από τέσσερις πύλες NAND

ΚΥΛΩΜΑ ΠΥΛΗΣ



ΛΟΓΙΚΟ ΣΥΜΒΟΛΟ



ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

IN1	IN2	OUT
F	F	F
T	F	T
F	T	T
T	T	T

3.1.2.2. Κυκλώματα Ακολουθιακής λογικής

Στα παραπάνω κυκλώματα συνδυαστικής λογικής, οι τιμές της εξόδου σ' οποιαδήποτε χρονική στιγμή είναι συνάρτηση μόνο των τιμών της εισόδου της ίδιας χρονικής στιγμής. Στην ακολουθιακή λογική οι τιμές της εξόδου των κυκλωμάτων επηρεάζονται από τις παρούσες αλλά και τις προηγούμενες τιμές της εισόδου. Υπάρχει δηλαδή ανάδραση.

Δύο είναι οι κύριες κατηγορίες των ακολουθιακών κυκλωμάτων:

Μη χρονισμένα: Αλλάζουν κατάσταση σύμφωνα με τις αλλαγές των εισόδων τους. Απαιτούνται ειδικές τεχνικές σχεδιασμού.

Χρονισμένα: Τα σήματα ανάδρασης διακόπτονται από καταχωρητές που ενεργοποιούνται από παλμούς ρολογιού χρονισμού. Συνεπώς η κατάστασή του κυκλώματος αλλάζει σύμφωνα με τους παλμούς του ρολογιού χρονισμού. Η κατάσταση του κυκλώματος ορίζεται από το περιεχόμενο των στοιχείων της μνήμης.

Flip – Flops

Τα flip-flops διαθέτουν δύο σταθερές καταστάσεις (1 και 0), και παρέχουν μνήμη που αποθηκεύει πληροφορία ενός (1) bit. Υπάρχουν διάφοροι τύποι flip-flops, οι οποίοι ταξινομούνται σύμφωνα με τον τρόπο λειτουργίας τους. Τα flip-flops αποτελούν τα βασικά δομικά στοιχεία για το σχεδιασμό των ακολουθιακών κυκλωμάτων

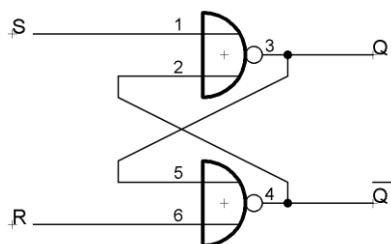
Μη χρονισμένα Κυκλώματα Ακολουθιακής λογικής

Set-Reset flip-flop

Πρόκειται για ένα τύπο flip flop που θέλει προσοχή κατά την χρήση του γιατί υπάρχουν καταστάσεις εισόδων που δεν είναι αποδεκτές όπως έχει σημειωθεί στον πίνακα λογικής με (!!). Ο λόγος που δεν είναι αποδέκτες είναι ότι απαιτείται χρόνος για μεταφερθούν τα δεδομένα από την είσοδο στην έξοδο και κάθε μια από τις λογικές πύλες ακόμα και αν είναι από το ίδιο υλικό έχει τον δικό της χρόνο συνεπώς οι έξοδοι για αυτές τις καταστάσεις εισόδου θα είναι τυχαίες.

Τα SR flip-flops έχουν δύο τύπους τα NOR SR flip-flops και τα NAND SR flip-flops και καθορίζονται από τις λογικές πύλες που χρησιμοποιούν. Παρακάτω παρουσιάζονται οι δύο τύποι με τους αντίστοιχους λογικούς πίνακες.

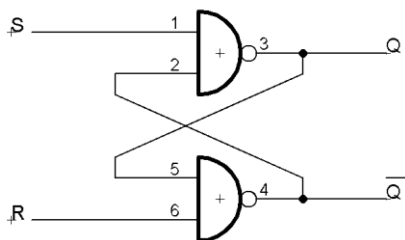
NOR SR flip-flop



ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

S	R	Q
F	F	HOLD
T	F	T
F	T	F
T	T	!!

NAND SR flip-flop



ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

S	R	Q
F	F	!!
T	F	F
F	T	T
T	T	HOLD

Χρονισμένα Κυκλώματα Ακολουθιακής λογικής (Clocked Circuits)

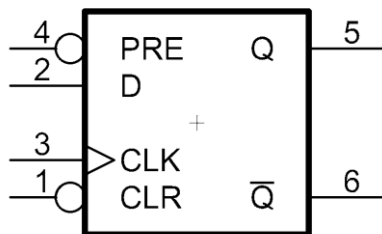
Τα χρονισμένα κυκλώματα χρησιμοποιούν ρολόι χρονισμού. Το ρολόι είναι ένας τετραγωνικός παλμός. Τα δεδομένα περνάνε από τις εισόδους στην έξοδο όταν το ρολόι χρονισμού, είτε μεταβεί από χαμηλή τάση σε υψηλή (positive edge triggered \uparrow), είτε από υψηλή σε χαμηλή (negative edge triggered \downarrow), είτε όταν απλά αλλάξει κατάσταση (positive and negative edge triggered $\uparrow\downarrow$). Αυτό ορίζεται από τον τρόπο κατασκευής του κυκλώματος.

Κατά τον συμβολισμό το ρολόι χρονισμού συνήθως παριστάνεται ως βέλος (>) είτε ως CLK

D flip-flops

Το D flip-flop έχει μια είσοδο (D) και δυο εξόδους (η μια είναι αντίθετη της άλλης). Όταν το ρολόι χρονισμού μεταβάλλεται από υψηλή τάση σε χαμηλή τότε η κατάσταση (state) στην είσοδο (D) μεταφέρεται στην έξοδο (Q, Q-bar). Εδω θεωρήσαμε αυθαίρετα ότι το flip flop ενεργοποιείται από την πτώση τάσης είναι όμως μια πληροφορία που την δίνει ο κατασκευαστής του κυκλώματος.

D flip-flop ΛΟΓΙΚΟ ΣΥΜΒΟΛΟ

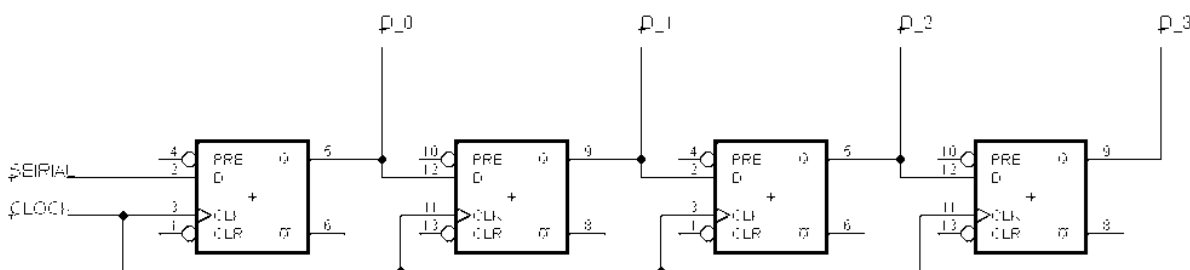


ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

D	Q	Q-bar
F	F \downarrow	T \downarrow
T	T \downarrow	F \downarrow

Καταχωρητής Ολίσθησης (Shift Register)

Εάν συνδέσουμε D flip flops σε σειρά και ενεργοποιούνται όλα με το ίδιο ρολόι τότε έχουμε δημιουργήσει έναν καταχωρητή ολίσθησης τόσων bits όσα και τα flip flops. Στο σχήμα 3.5 αναπαρίσταται ένας καταχωρητής τεσσάρων bits. Κάθε φορά που το ρολόι στέλνει ένα παλμό τότε τα δεδομένα κυλούν από το D_0 προς το D_3 κατά μια θέση. Οι καταχωρητές ολίσθησης χρησιμοποιούνται για την μετατροπή σειριακών δεδομένων σε παράλληλα, αλλά και για την αρμονική επικοινωνία διαφορετικών κυκλωμάτων (master – slave communications)

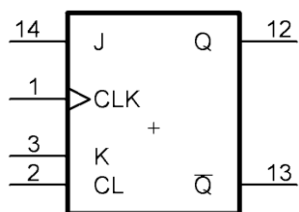


ΣΧΗΜΑ 3.5 Καταχωρητής Ολίσθησης 4-bits

JK flip-flops

Το JK είναι ένα χρονισμένο flip flop με δυο εισόδους αντί μιας όπως το D flip flop. Η ιδιαιτερότητα του JK flip flop είναι ότι όπως φαίνεται και στο λογικό πίνακα όταν και τα οι δυο εισοδοι έχουν λογικό 1 τότε η έξοδος αλλάζει συνεχώς κατάσταση για κάθε επόμενο παλμό (Toggle)

JK flip-flop

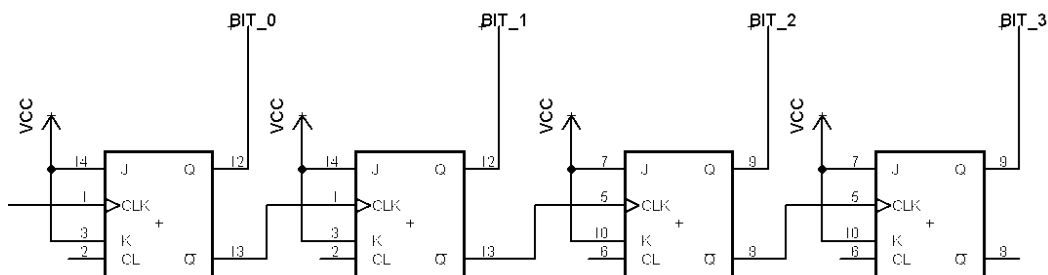


ΠΙΝΑΚΑΣ ΛΟΓΙΚΗΣ

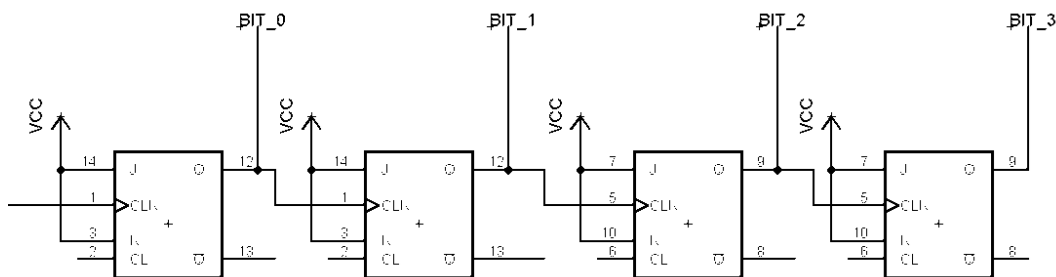
J	K	Q	\bar{Q}
F	F	HOLD	TOGGLE
T	F	T	F
F	T	F	T
T	T	TOGGLE	HOLD

Έτσι, ενώνοντας τις δυο εισόδους ενός JK flip flop προκύπτει ένα νέο flip flop τύπου T (Toggle). Τα JK flip flops βρίσκουν μεγάλη εφαρμογή στην δημιουργία ασύγχρονων καταμετρητών (ripple counters).

Παρακάτω παρουσιάζονται οι δυο διατάξεις JK flip flop για καταμέτρηση +1 (Up Counter) και -1 (Down Counter) τεσσάρων bit.



ΣΧΗΜΑ 3.6 Up Counter



ΣΧΗΜΑ 3.7 Down Counter

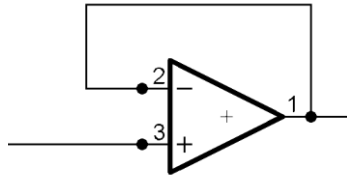
Παρατηρούμε ότι το πρώτο flip flop χρειάζεται δυο παλμούς για να επανέλθει στην προηγούμενη κατάσταση το bit 0, το δεύτερο τέσσερις για το bit 1 το τρίτο οκτώ για το bit 2 και το τέταρτο δεκαέξι για το bit 3. Κάθε παλμός λοιπόν προσθέτει ή αφαιρεί μια μονάδα (στο δεκαδικό σύστημα).

3.1.2.3. Τελεστικοί Ενισχυτές (Operational Amplifiers)

Οι ενισχυτές (amplifiers) μπορούν να λειτουργήσουν και ως διαφορικοί τελεστές για ψηφιακά κυκλώματα ένας τελεστικός ενισχυτής έχει δυο εισόδους και μια έξοδο. Η είσοδος που σημειώνεται με (-) είναι η αναστρέφουσα είσοδος και αυτή με (+) είναι η μη αναστρέφουσα είσοδος. Όταν σταλεί το σήμα εξόδου του τελεστικού ενισχυτή πίσω στην αναστρέφουσα είσοδος τότε έχουμε διαμόρφωση. Με βάση αυτή την αρχή υπάρχουν τέσσερις βασικές διατάξεις

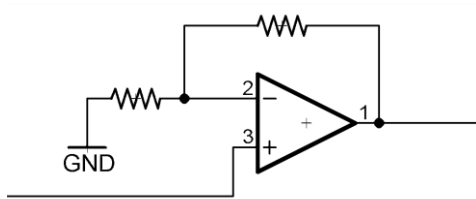
- Ο Ακολουθητής τάσης
- Ο Μη Αναστρέφων Ενισχυτής
- Ο Αναστρέφων Ενισχυτής
- Και Ο Αθροιστής

Απλός Ακολουθητής



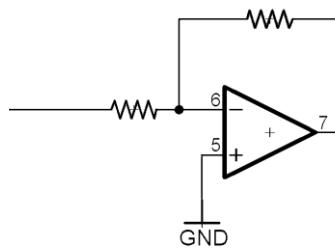
Πρόκειται για την πιο απλή χρήση ενός τελεστικού ενισχυτή και συμβάλει ως φίλτρο για την διόρθωση θορύβων.

Μη Αναστρέφων Ενισχυτής



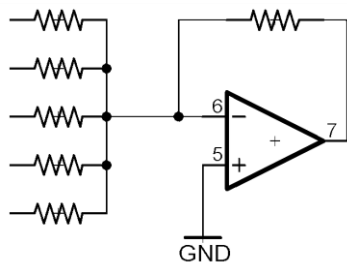
Έχει την δυνατότητα ενίσχυσης ή μείωσης της τάσης χωρίς την δυνατότητα της ανάστροφης διαδικασίας

Αναστρέφων Ενισχυτής



Έχει την δυνατότητα ενίσχυσης ή μείωσης της τάσης με την δυνατότητα της ανάστροφης διαδικασίας

Αθροιστής



Αποδίδει την ολική τάση που δίνεται από τις εισόδους

3.1.2.4. Αποληκτικές Διατάξεις (Terminations)

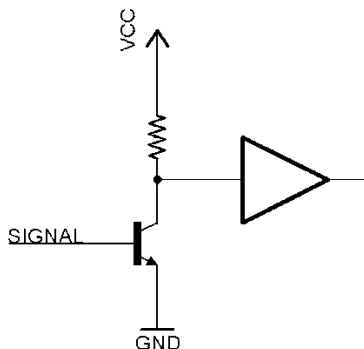
Ένα κύκλωμα μπορεί να βρίσκεται σε μια από τις τρεις καταστάσεις: κλειστό (ON) ανοιχτό (OFF) και χωρίς σύνδεση (not connected: NC). Στην κατάσταση NC το κύκλωμα δεν έχει σταθερότητα (floating), δεν ξέρουμε δηλαδή πως θα συμπεριφερθεί σε περίπτωση που του δοθεί ρεύμα. Για την αποφυγή αυτής της κατάστασης προτιμώνται κάποιες διατάξεις η οποίες θέτουν το κύκλωμα σε μια από τις δυο επιτρεπτές καταστάσεις ON και OFF. Τις διατάξεις αυτές τις συναντούμε είτε κατά την επικοινωνία των κυκλωμάτων με αλλά, είτε κατά την σύνδεση με πηγές τάσης για το λόγο αυτό ονομάζονται αποληκτικές διατάξεις (terminations). Υπάρχουν πολλές διατάξεις εμείς όμως θα αναφερθούμε στις τρεις κυριότερες

- Αποληκτική διάταξη Ανόδου (pull up termination)
- Αποληκτική διάταξη Καθόδου (pull down termination)
- Αποληκτική διάταξη Thevenin (thevenin termination)

Αποληκτική διάταξη Ανόδου (pull up termination)

Και οι τρεις διατάξεις ακολουθούν μια βασική αρχή. Ότι το ρεύμα ακολουθεί πάντα την πιο σύντομη διαδρομή, δηλαδή την διαδρομή που θα συναντήσει την μικρότερη αντίσταση.

Στην περίπτωση της διάταξης Ανόδου το κύκλωμα είναι μονιμά συνδεδεμένο με την τροφοδοσία όπως φαίνεται στο σχήμα. Συνεπώς είναι στην κατάσταση ON. Όταν κλείσει ο διακόπτης -στην περίπτωση του σχήματος τον ρόλο του διακόπτη έχει αναλαμβάνει ένα τρανζίστορ- τότε το ρεύμα ακολουθεί την πιο σύντομη διαδρομή ρέει προς την γείωση χωρίς να περάσει από το κύκλωμα, έτσι το κύκλωμα βρίσκεται στην κατάσταση OFF.

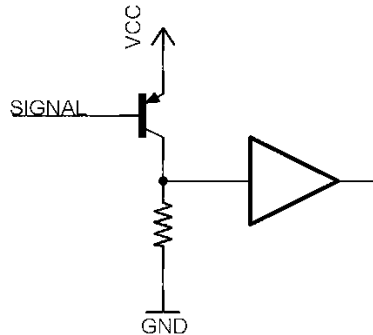


ΣΧΗΜΑ 3.8 Αποληκτική διάταξη Ανόδου

Αποληκτική διάταξη Καθόδου (pull down termination)

Στην περίπτωση της διάταξης καθόδου το κύκλωμα είναι μονιμά γειωμένο Όταν ο διακόπτης είναι κλειστός το ρεύμα περνά από το κύκλωμα μόνο αν η αντίσταση που φαίνεται στο σχήμα είναι μεγαλύτερη από την αντίσταση του κυκλώματος, για τον λόγο αυτό θα πρέπει να επιλέγεται με

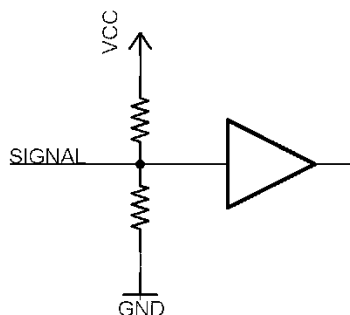
προσοχή και συνήθως είναι διπλάσια από την αντίσταση του κυκλώματος, Τότε το κύκλωμα είναι σε κατάσταση ON. μόλις ανοίξει ο διακόπτης το ρεύμα που έχει διατηρήσει το κύκλωμα ρέει προς την γείωση και τελικά το κύκλωμα έρχεται σε κατάσταση OFF.



ΣΧΗΜΑ 3.9 Αποληκτρία διάταξη Καθόδου

Αποληκτρία διάταξη Thevenin (thevenin termination)

Η Αποληκτρία διάταξη thevenin μοιάζει να συνδυάζει τις δυο παραπάνω διατάξεις. Αυτό που εκμεταλλεύεται η διάταξη thevenin είναι τον νόμο του Ohm. Έστω R_{TH} η αντίσταση που φαίνεται στο σχήμα να συνδέεται με την τροφοδοσία (Vcc) και R_{TL} η αντίσταση που συνδέεται με την γείωση τότε $V_{TH}=V_{TL} = I (R_{TH} R_{TL}) / (R_{TH} + R_{TL})$ έτσι παρατηρούμε ότι ανάλογα με το ύψος του ρεύματος και την κατάλληλη επιλογή αντιστάσεων η διάταξη μπορεί να λειτουργήσει είτε σαν διάταξη ανόδου είτε σαν διάταξη καθόδου. Το πλεονέκτημα της διάταξης thevenin είναι ότι σπαταλά λιγότερη ενέργεια όταν τροφοδοτείται το κύκλωμα και ότι βοηθά σημαντικά στην μείωση του θορύβου

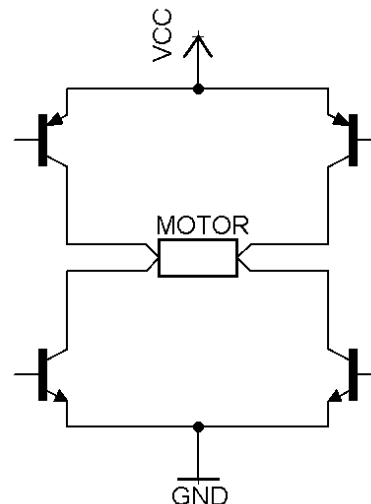


ΣΧΗΜΑ 3.10 Αποληκτρία διάταξη Thevenin

3.1.2.5. Γέφυρα H

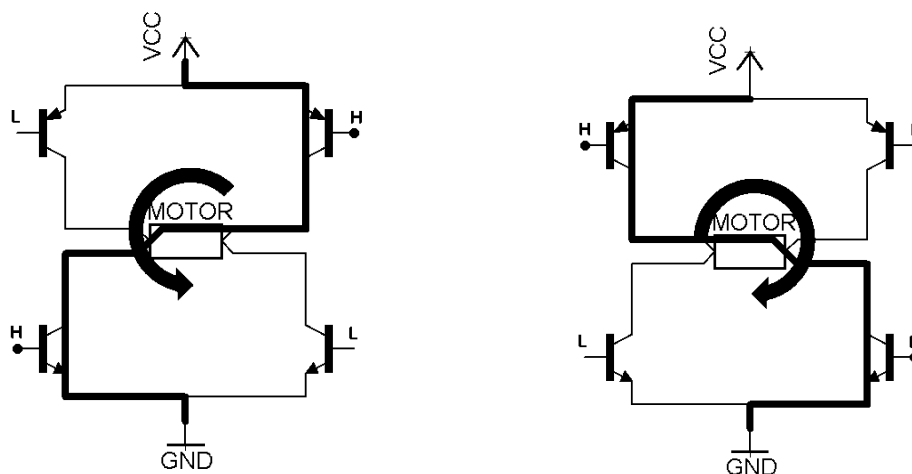
Η γέφυρα H είναι μια διάταξη τρανζίστορ που επιτρέπει σε έναν κινητήρα να αλλάζει κατεύθυνση περιστροφής. Ονομάστηκε γέφυρα H επειδή οπτικά μοιάζει με το γράμμα H. Όπως φαίνεται και στο σχήμα 3.11η γέφυρα αποτελείται από τέσσερα τρανζίστορ, 2 NPN που συνδέονται με τα άκρα του κινητήρα και την γείωση και 2 PNP που συνδέονται με την τροφοδοσία και τα άκρα του κινητήρα. Ανοιγοκλείνοντας ανά ζεύγη (PNP – NPN) τα τρανζίστορ έχουμε την επιθυμητή φορά. Για παράδειγμα αν δοθεί H (high) στην βάση βλ. σχήμα 3.13 τότε το ρεύμα θα ρέει όπως δείχνει η έντονη γραμμή και ο κινητήρας θα στραφεί ωρολογιακά. Αν τώρα δοθεί τάση στις βάσεις των τρανζίστορ όπως δείχνει το σχήμα 3.12 τότε ο κινητήρας θα στραφεί αντί-ωρολογιακά.

Αυτό που παρατηρείτε στην γέφυρα H είναι ότι το κύκλωμα που ανοιγοκλείνει τα τρανζίστορ είναι ανεξάρτητο από αυτό που τροφοδοτεί τον κινητήρα. Έτσι για παράδειγμα με καταλληλά τρανζίστορ μπορούμε να ελέγχουμε την κατεύθυνση ενός κινητήρα DC 24V με σήματα 3.3V.



ΣΧΗΜΑ 3.11 Η Γέφυρα H

Υπάρχουν πολλές παραλλαγές τις γέφυρας H όπως, η μισή γέφυρα (Half H-bridge), η χρήση τρανζίστορ MOSFET, η χρήση τρανζίστορ Darlington κ.α. πολλοί κατασκευαστές έχουν δημιουργήσει ολοκληρωμένα κυκλώματα, Η γέφυρας, αρκετά μικρών διαστάσεων που όμως έχουν εφαρμογή σε μικρούς κινητήρες. Οι μεγαλύτεροι κινητήρες υπερθερμαίνουν τα τρανζίστορ και για το λόγο αυτό θα πρέπει να είναι μεμονωμένα και να έχει προβλεφθεί η ψύξη τους.



ΣΧΗΜΑ 3.12 Η ροή του ρεύματος για ωρολογιακή και αντιωρολογιακή φορά

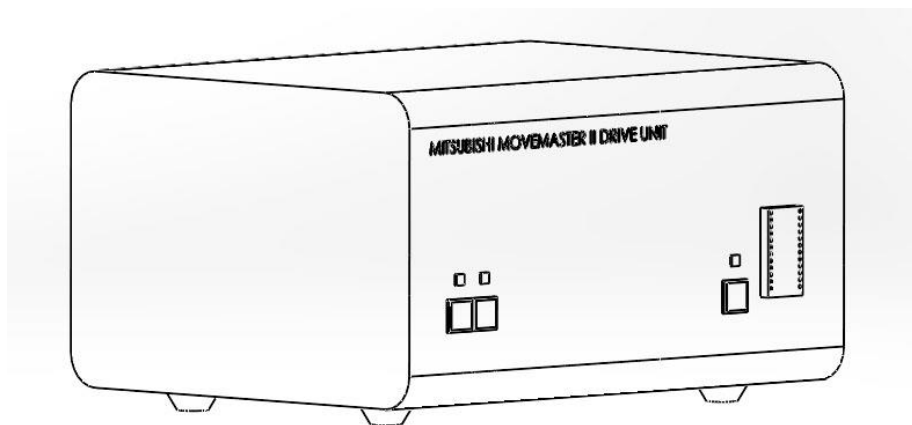
3.2. Ηλεκτρονικά ρομποτικού βραχίονα RM-501

Ο ρομποτικός βραχίονας RM-501 συνοδεύεται από μια μονάδα ελέγχου (Drive Unit) η οποία εμπεριέχει τον μικροεπεξεργαστή που παράγει τα σήματα για τον έλεγχο του βραχίονα, την μνήμη για την αποθήκευση πορείας, και τους Σέρβο-Ενισχυτές (Servo-Amplifiers). Επίσης έχει την δυνατότητα να συνεργαστεί με περιφερικά συστήματα μέσω θυρών I/O και με αυτήν επικοινωνούν ο υπολογιστής και το χειριστήριο διδασκαλίας (Teaching Box).

Παρότι ο κατασκευαστής έχει φροντίσει την δυνατότητα της επικοινωνίας με διάφορους τρόπους, δεν είναι δυνατή η επικοινωνία με σύγχρονο Η/Υ. Επίσης ο μικροεπεξεργαστής είναι προγραμματισμένος έτσι ώστε η μονάδα να διαχειρίζεται μόνο με την μέθοδο διδασκαλίας τον ρομποτικό βραχίονα. Ακόμη και η γλώσσα που δημιουργήθηκε για τον έλεγχο του ρομπότ από τον κατασκευαστή είναι γραπτός τρόπος διδασκαλίας διαχείρισης

Για την ανάγκη επικοινωνίας με σύγχρονο Η/Υ, έπρεπε να απομονωθεί και να αχρηστευθεί ο μικροεπεξεργαστής. Με το πρόγραμμα LinuxCNC υπάρχει η δυνατότητα να παραχθούν τα σήματα από τον ίδιο τον υπολογιστή μέσω της παράλληλης θύρας. Έτσι από την μονάδα ελέγχου χρειαζόμαστε μόνο τους Σέρβο-Ενισχυτές και τους χειριστές ανάδρασης για κάθε σερβοκινητήρα. Όμως τα κυκλώματά των Σέρβο-Ενισχυτών δεν έχουν ξεκάθαρη επικοινωνία με τον επεξεργαστή. Δεδομένου της απουσίας οποιουδήποτε τεχνικού εγχειρίδιου, έπρεπε να γίνει αποδόμηση και σχεδίαση των ηλεκτρολογικών σχημάτων (schematics) των Σέρβο-Ενισχυτών. Η αποδόμηση της μονάδας ελέγχου ήταν μια περιπλοκή και χρονοβόρα διαδικασία.

Σε αυτή την ενότητα θα ασχοληθούμε με την αποκρυπτογράφηση των πλακετών των Σέρβο-Ενισχυτών.

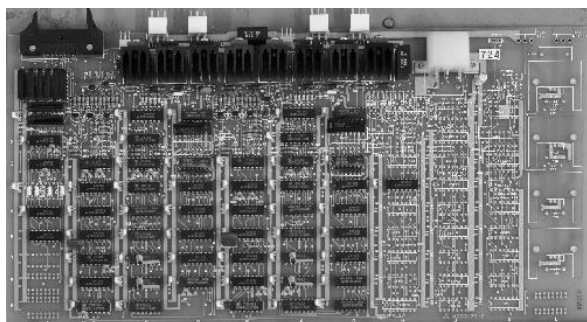


ΣΧΗΜΑ 3.13 Η μονάδα ελέγχου

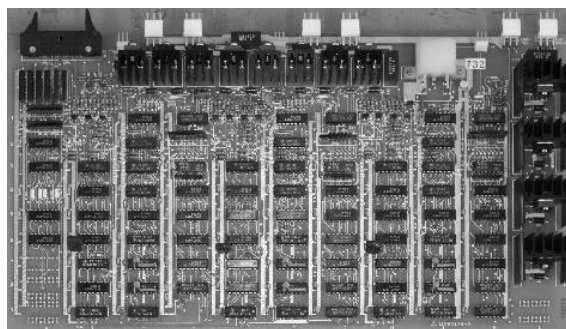
3.2.1. Ηλεκτρολογικά σχέδια

Κατά την διαδικασία αποδόμησης της μονάδας ελέγχου διαπιστώθηκε ότι τα κυκλώματα των σερβοενισχυτών ήταν τυπωμένα σε τρεις πλακέτες. Οι πλακέτες ήταν συνδεδεμένες σε σειρά με τον μικροεπεξεργαστή με θύρες 20 καρφιών (20-pin connectors).

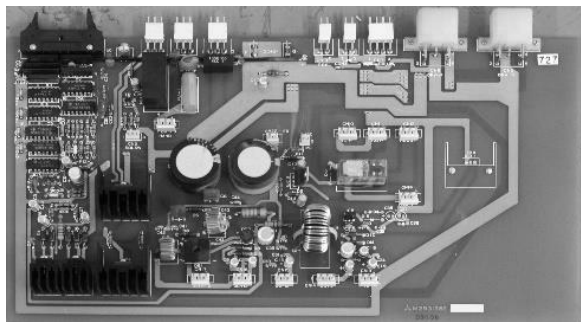
Η αποκρυπτογράφηση των κυκλωμάτων έγινε στις τέσσερις πλακέτες που παρουσιάζονται παρακάτω. Η πλακέτα 724 (βλ. Σχ. 3.14) επικοινωνεί με τους κινητήρες της μέσης και του αγκώνα ενώ η 732 (βλ. Σχ. 3.17) με του ωμού και τους δύο κινητήρες του καρπού. Στην πλακέτα 727 (βλ. Σχ. 3.16) ένα μέρος της αναλαμβάνει τον έλεγχο της αρπάγης ενώ το άλλο μέρος της διαχειρίζεται τα ρεύματα όλης της μονάδας. Σε αυτήν θα συναντήσουμε ρεύματα τάσεων 220V, $\pm 24V$, $\pm 12V$ και 5V. Η τέταρτη πλακέτα που φαίνεται στο σχήμα 3.15 βρίσκεται μέσα στο σώμα το ρομποτικού βραχίονα και δεν κάνει τίποτε άλλο από το μειώνει το θόρυβο των σημάτων που πρόκειται να σταλούν πίσω στην μονάδα ελέγχου



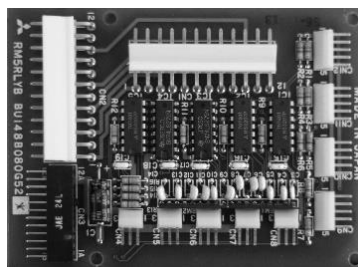
ΣΧΗΜΑ 3.14 Πλακέτα 724 για τον έλεγχο της Μέσης και του Αγκώνα



ΣΧΗΜΑ 3.16 Πλακέτα 732 για τον έλεγχο του Ώμου και του Καρπού



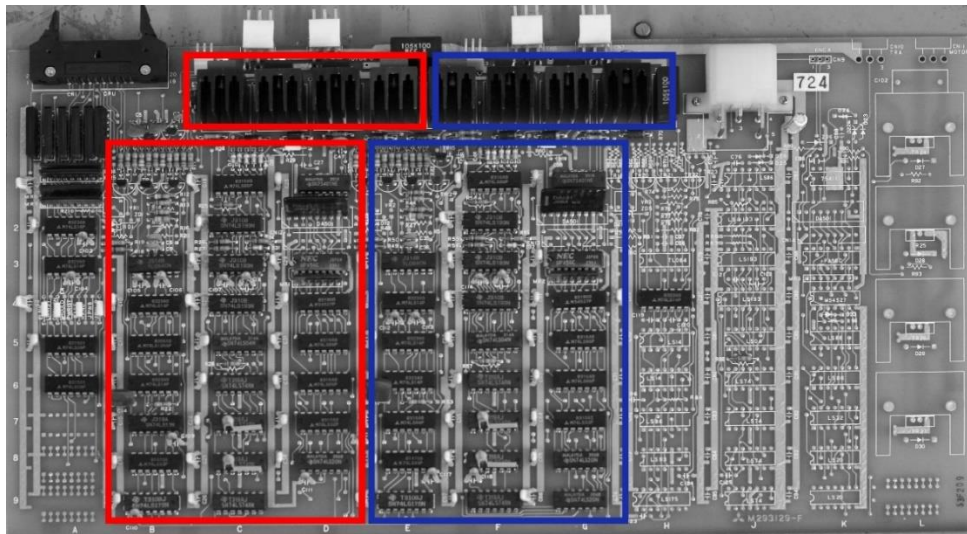
ΣΧΗΜΑ 3.15 Πλακέτα 727 για τον έλεγχο της αρπάγης και την διαχείριση των ρευμάτων



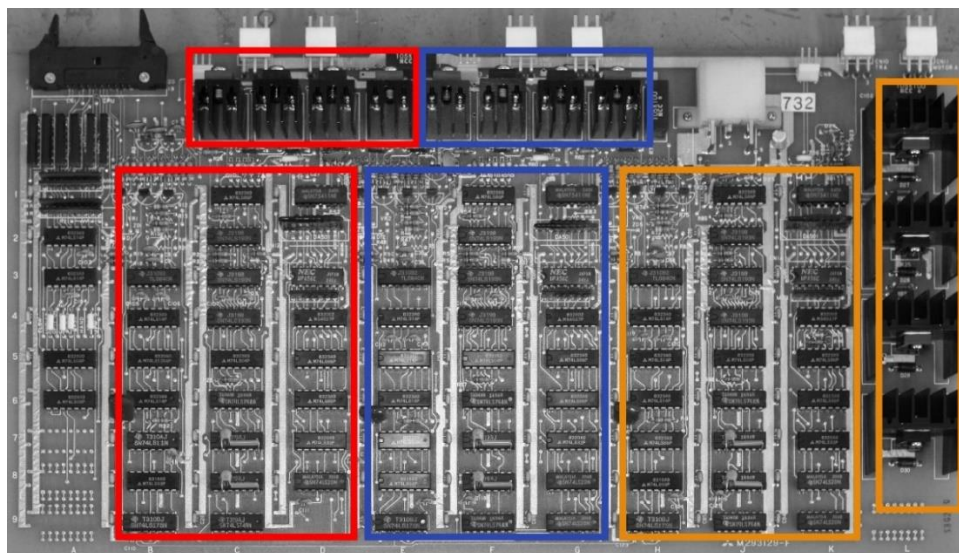
ΣΧΗΜΑ 3.17 Πλακέτα επικοινωνίας του ρομπότ με την μονάδα ελέγχου

3.2.2. Οι Πλακέτες 724 και 732

Παρατηρούμε ότι οι πλακέτες 724 και 732 είναι πανομοιότυπες. Το μόνο που αλλάζει είναι ότι η πρώτη δεν φέρει όλα τα ηλεκτρονικά στοιχεία για τα οποία σχεδιάστηκε. Το γεγονός ότι οι πλακέτες είναι ίδιες και ότι η 724 διαχειρίζεται δυο κινητήρες ενώ η 732 τρεις, δείχνει την υποψία ανεξάρτητων κυκλωμάτων πάνω σε μια πλακέτα. Πράγματι όπως παρατηρούμε και από τα σχήματα 3.2.6 και 3.2.7 κάθε πλακέτα περιλαμβάνει τρία όμοια ανεξάρτητα κυκλώματα. Τα ηλεκτρονικά στοιχεία που «πατάνε» πάνω σε ψήκτρες είναι τα ζευγάρια NPN και PNP τρανζίστορ της γέφυρας H για κάθε κινητήρα.



ΣΧΗΜΑ 3.18 Τα δυο ανεξάρτητα κυκλώματα πάνω στην πλακέτα 724



ΣΧΗΜΑ 3.19 Τα τρία ανεξάρτητα κυκλώματα πάνω στην δεύτερη πλακέτα 732

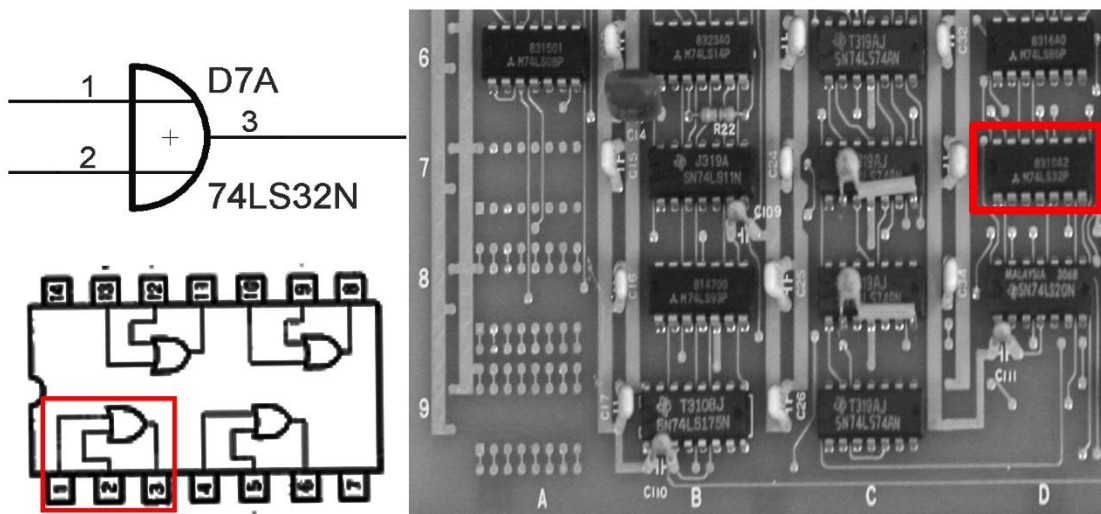
Έτσι, η αποκρυπτογράφηση μπορεί να περιοριστεί σε ένα από τα πέντε κυκλώματα, και πως επικοινωνούν αυτά με τον επεξεργαστή, δηλαδή την αποκρυπτογράφηση των 20 καρφιών των θυρών.

Παρόλ'αυτα Η αποκρυπτογράφηση έγινε από σημείο σε σημείο σε όλη την πλακέτα με την χρήση πολύμετρου και τον κατάλογο του Παραρτήματος Α. Τα αποτελέσματα αποτυπώνονται στα ηλεκτρολογικά σχέδια που παραπέιθονται ξεχωριστά. Λόγω της μεγάλης κλίμακας στην οποία σχεδιάστηκαν δεν μπορούν να προβληθούν σαν σχήματα στην ροή του κειμένου.

Στην χαρτογράφηση δεν έχουν ληφθεί υπόψη οι πυκνωτές για την διόρθωση θορύβου μια και τα σχέδια αποτελούν εποπτικό μέσο και η παρουσία εκατοντάδων πυκνωτών που δεν προσφέρουν στο λογικό διάγραμμα θα προκαλούσε σύγχυση.

Αξίζει να σημειωθεί ότι τα λογικά σύμβολα των σχεδίων είναι σημειωμένα με τέτοιο τρόπο έτσι ώστε να αντιπροσωπεύουν ακριβώς την λογική πύλη του κυκλώματος πάνω στην πλακέτα

Για παράδειγμα στο σχήμα 3.20 βλέπουμε το λογικό σύμβολο μιας λογικής πύλης OR ο κωδικός του ολοκληρωμένου κυκλώματος είναι 74LS32N και βρίσκεται στην πλακέτα στην θέση D7. Τα νούμερα στις εισόδους και την έξοδο της πύλης αντιπροσωπεύουν την λογική πύλη από τις έξι που έχει το συγκεκριμένο ολοκληρωμένο κύκλωμα σύμφωνα με το Παράρτημα Α

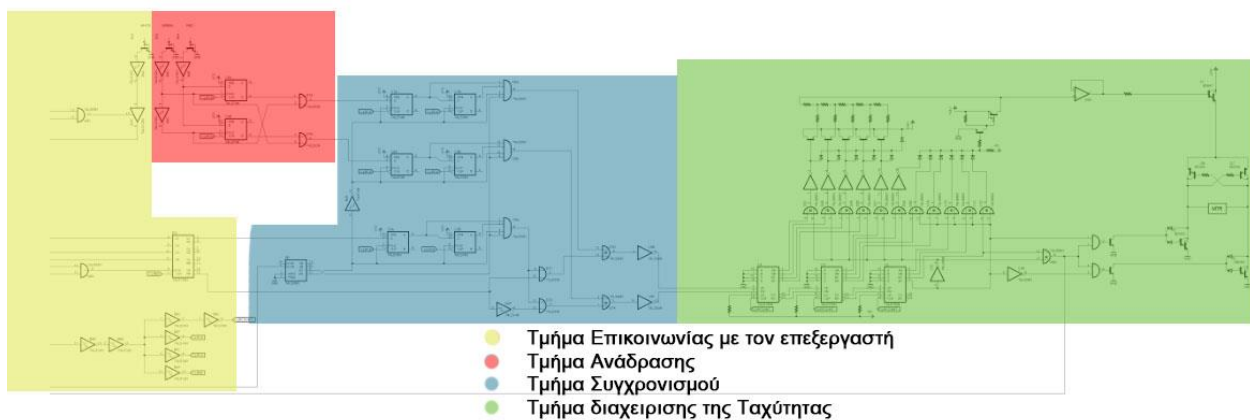


ΣΧΗΜΑ 3.20 Η κωδικοποίηση κάθε λογικού συμβόλου στα ηλεκτρολογικά σχέδια

Μετά την αποτύπωση των πλακετών σαν ηλεκτρολογικά σχέδια θα πρέπει να γίνει η αποκρυπτογράφηση τους. Θα πρέπει δηλαδή να αποκωδικοποιηθούν όλες αυτές οι λογικές πύλες, για ποιο λόγο υπάρχουν και πως συμβάλουν στο κύκλωμα. Προφανώς αυτή η διαδικασία αρκεί να γίνει σε μόνο ένα κύκλωμα από τα συνολικά πέντε τα οποία ακολουθούν την ίδια αρχή.

Απομονώνοντας ένα κύκλωμα σέρβο-ενισχυτή και παρατηρώντας το μπορούμε να το χωρίσουμε σε 4 επιμέρους τμήματα

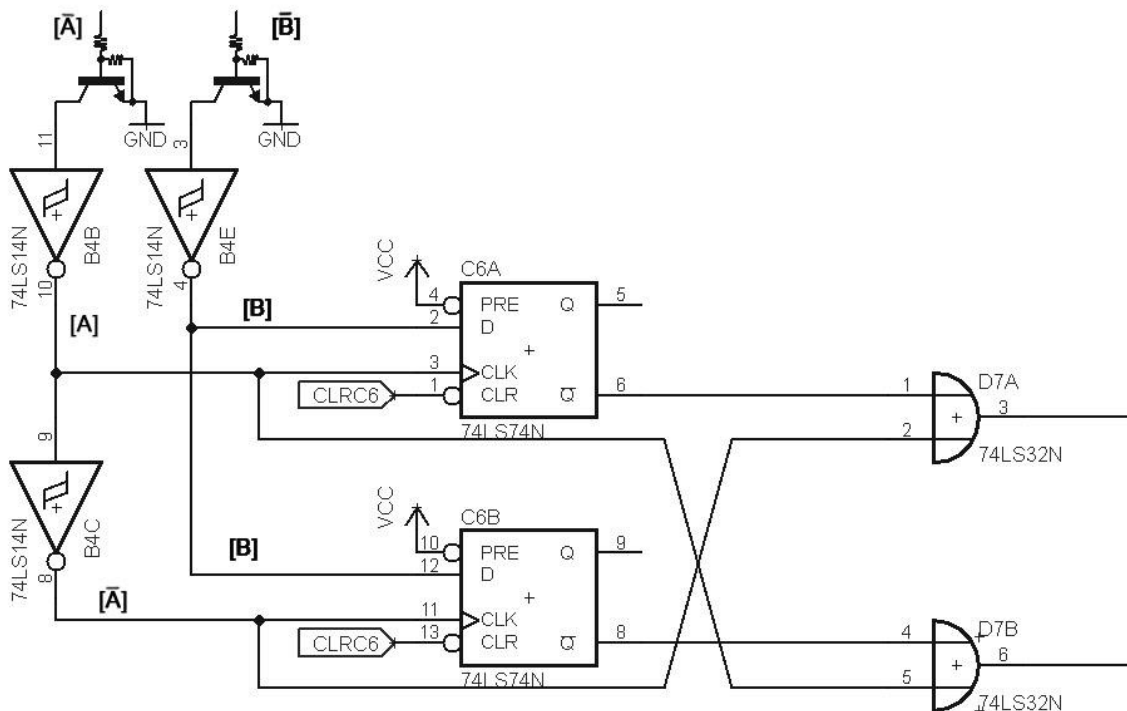
- Το τμήμα επικοινωνίας με τον επεξεργαστή
- Το τμήμα ανάδρασης
- Το τμήμα συγχρονισμού
- Το τμήμα διαχείρισης της ταχύτητας



ΣΧΗΜΑ 3.21 Τα επιμέρους τμήματα ενός από τα κυκλώματα σερβοενισχυτών

Παρακάτω θα γίνει μια πιο εκτενής περιγραφή κάθε τμήματος

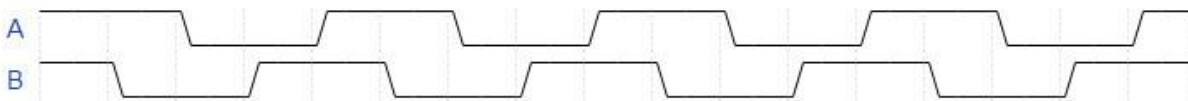
3.2.2.1. Τμήμα ανάδρασης



ΣΧΗΜΑ 3.22 Το κύκλωμα Ανάδρασης

Στο πάνω μέρος του σχήματος 3.22 παρατηρούμε δυο τρανζίστορ. Στα τρανζίστορ αυτά έρχονται τα σήματα ανάδρασης από τον κωδικοποιητή (encoder) του εκάστοτε κινητήρα.

Όπως αναφέρθηκε στο δεύτερο μέρος του πρώτου κεφαλαίου το κύκλωμα ανάδρασης κάθε κινητήρα παράγει κατά την λειτουργία του δυο τετραγωνικά σήματα που απέχουν διαφορά περιόδου $\pi/2$. Έστω ότι αυτά τα σήματα είναι το [A] και το [B]



ΣΧΗΜΑ 3.23 Τα δυο τετραγωνικά σήματα με διαφορά φάσης $\pi/2$ που παράγει ο κινητήρας

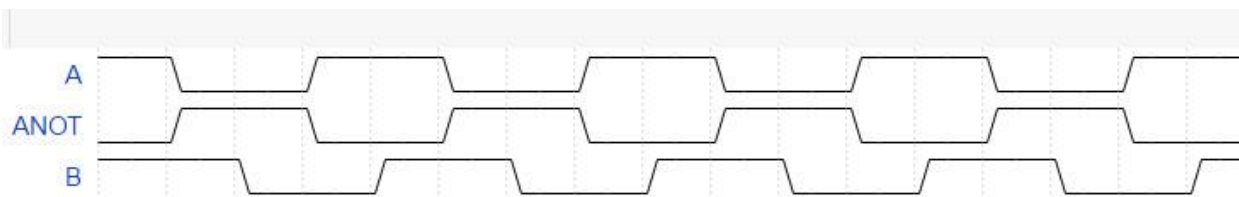
Η συχνότητα αυτών των δυο σημάτων μαρτυρά την ταχύτητα του κινητήρα. Η κατεύθυνση περιστροφής καθορίζεται από ποιο από τα δυο σήματα προηγείται. Αυτό ακριβώς εκμεταλλεύεται το παραπάνω κύκλωμα. Παρατηρούμε ότι το ένα σήμα λειτουργεί σαν ρολόι χρονισμού (έστω ότι είναι το [A]) για δυο flip flops τύπου D ενώ το άλλο (το [B]) ως είσοδος. Επίσης το ρολόι χρονισμού [A] για τα δυο flip flop αντιστρέφεται $[\bar{A}]$. Τέλος δυο OR πύλες συγκρίνουν το ρολόι χρονισμού ενός flip flop με την αρνητική έξοδο του αλλού. Πρέπει να σημειωθεί ότι το 74LS74 «περνά» τα δεδομένα εισόδου στην

έξοδο κατά την θετική πλευρά μετάβασης του παλμού χρονισμού (positive edge triggered D-type flip flops) (βλ. Παράρτημα Α).

Αυτό που συμβαίνει στο παραπάνω κύκλωμα είναι:

Όταν προηγείται το [A] τότε το πάνω flip flop «βλέπει» πάντα 1 ενώ το κάτω πάντα 0. Συνεπώς οι αρνητικές έξοδοι των δυο flip flop πάνω και κάτω θα δίνουν 0 και 1 αντίστοιχα. Μια λογική πύλη OR δίνει πάντα 1 εκτός αν και τα δυο είναι 0. Έτσι, το πάνω OR θα διαβάσει το σήμα του [A] ενώ το κάτω θα δίνει 1

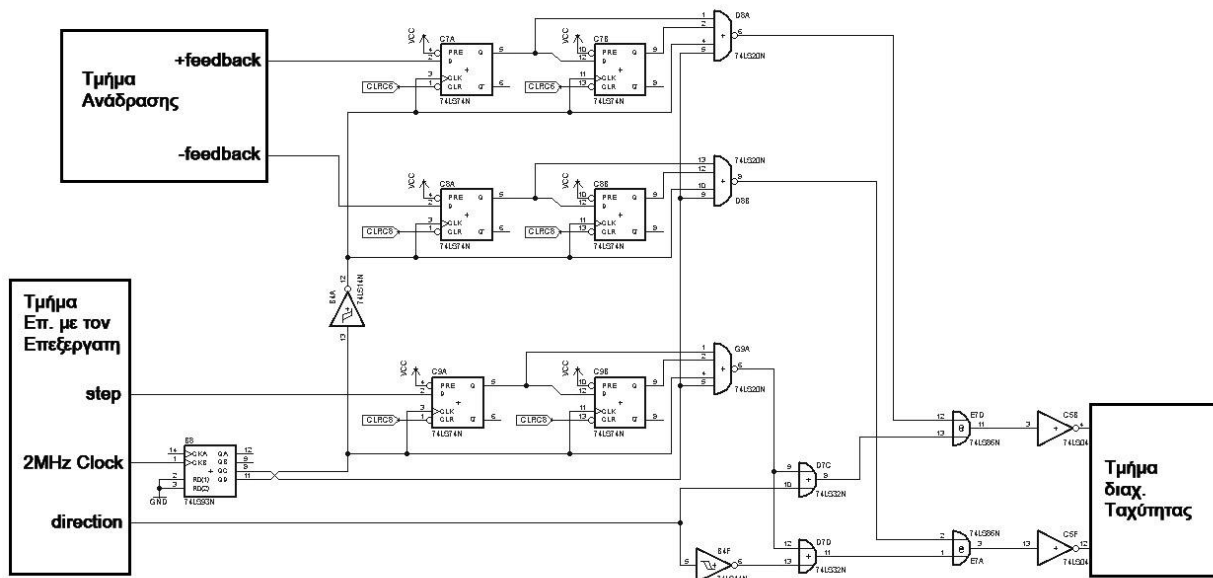
Αντίστοιχα, όταν προηγείται το [B] τότε το \bar{A} προηγείται του [B] έτσι με την ίδια λογική το πάνω OR θα δίνει 1 ενώ το κάτω θα διαβάσει το σήμα του \bar{A}



ΣΧΗΜΑ 3.24 Το [B] προηγείται του [A] και το ρολόι χρονισμού \bar{A}

Συνεπώς, όταν η άρθρωση στρέφεται ωρολογιακά τότε ο παλμός ανάδρασης θα δίνει σήματα για να την στρέψει αντιωρολογιακά και το αντίθετο. Αναλύοντας το τμήμα συγχρονισμού θα γίνει περισσότερο αντιληπτή η λειτουργία ανάδρασης.

3.2.2.2. Τμήμα συγχρονισμού



ΣΧΗΜΑ 3.25 Το κύκλωμα Συγχρονισμού

Με μια γρήγορη επισκόπηση του σχήματος 3.25 παρατηρούμε μια ομοιότητα διαχείρισης για κάθε σήμα που έρχεται από το τμήμα ανάδρασης και ενός από τα σήματα που προέρχεται από το τμήμα επικοινωνίας με τον επεξεργαστή. Συμπεραίνουμε ότι αυτό το σήμα είναι τα βήματα (steps) που πρέπει να παράγει ο υπολογιστής.

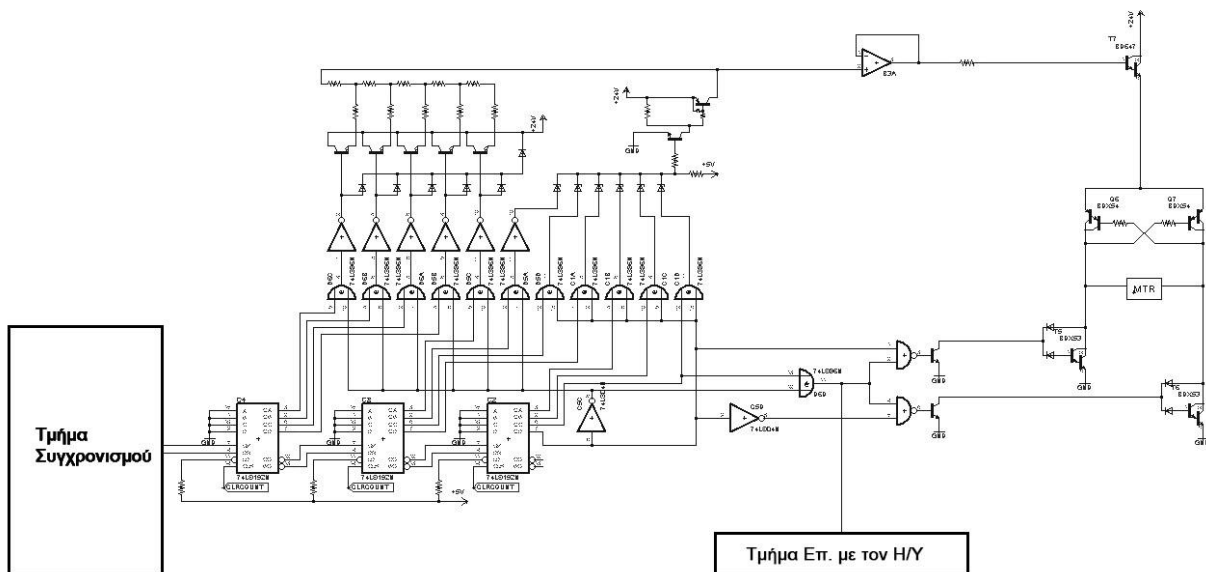
Ένα από τα αλλά δυο σήματα που προέρχονται από το τμήμα επικοινωνίας με τον επεξεργαστή είναι ρολόι χρονισμού για τον καταμετρητή 4-bit M74LS175P (βλ. Παράρτημα Α) και χρησιμοποιείται το τέταρτο ψηφίο εξόδου (Q_3) για τον χρονισμό όλων των flip-flops του τμήματος συγχρονισμού. Από μετρήσεις με παλμογράφο γνωρίζουμε ότι το σήμα αυτό είναι ένα σταθερό τετραγωνικό σήμα συχνότητας 2MHz. Έχει πολύ μεγαλύτερη συχνότητα από ένα συνηθισμένο βηματισμό που κυμαίνεται από 25÷60 KHz. Έτσι συμπεραίνουμε ότι σκοπός του είναι να συγχρονίσει το βηματισμό που προέρχεται από τον επεξεργαστή με τον βηματισμό ανάδρασης που παράγει ο έλεγχος. Ο συγχρονισμός αυτός επιτυγχάνεται ως εξής:

- Το τέταρτο ψηφίο του M74LS175P παράγει σήματα με πλάτος τέσσερις φορές μεγαλύτερο από το αρχικό έτσι το σήμα έχει πλέον συχνότητα 0,5MHz
- Ανά 0,5 MHz μεταφέρεται η πληροφορία του βηματισμού από τον επεξεργαστή από το ένα flip flop τύπου D στο άλλο (2-bit shift-register) ενώ η πληροφορία βηματισμού ανάδρασης μεταφέρεται με φάση $\pi/2$ (αναστρέφεται το ρολόι χρονισμού).
- Οι NAND πύλες τεσσάρων εισοδών (βλ. Παράρτημα Α : SN74LS20N) έχουν τον ρολό του τελικού συντονιστή. Δίνουν σήμα όταν το τρίτο και το τέταρτο ψηφίο του M74LS175P είναι λογικό 1 (δηλαδή ανά $0,5+0,5*0,66=0,83$ MHz) και δεν υπάρχει μεταβολή βήματος ανάμεσα σε δυο χρονισμούς.

Το τρίτο σήμα που έρχεται από το τμήμα επικοινωνίας με τον επεξεργαστή συγκρίνεται -αυτό και το ανάστροφο του- με το βηματισμό από τον επεξεργαστή (ύστερα από τον συγχρονισμό του) μέσω λογικών πυλών OR. Το σήμα βηματισμού θα περάσει από την πύλη OR μόνο όταν ο άλλος όρος προς σύγκριση θα είναι 0. Με άλλα λόγια, το τρίτο σήμα παίζει το ρόλο της φοράς, είναι, δηλαδή, η επιθυμητή κατεύθυνση περιστροφής του κινητήρα και κατά συνέπεια της εκάστοτε άρθρωσης.

Για σταθερή ταχύτητα κάθε παλμός βηματισμού που θα «περάσει» στο τμήμα διαχείρισης της Ταχύτητας θα διορθωθεί αμέσως μετά από τον παλμό που θα προκαλέσει η ανάδραση. Αυτό επιτυγχάνεται με τις λογικές πύλες XOR όπου συγκρίνονται οι δυο παλμοί.

3.2.2.3. Τμήμα Διαχείρισης της Ταχύτητας



ΣΧΗΜΑ 3.26 Το κύκλωμα Διαχείρισης της Ταχύτητας

Το τμήμα διαχείρισης της Ταχύτητας αποτελείται από τρεις UP-DOWN καταμετρητές συνδεδεμένους σε σειρά δημιουργώντας, έναν γενικό καταμετρητή 12-bit, δηλαδή ένα εύρος 4096 ψηφίων (βλ. Παράρτημα Α: SN74LS173N). Οι έξοδοι του καταμετρητή καταλήγουν σε μετατροπή ψηφιακού σε αναλογικό σήμα με κλίμακα αντιστάσεων R-2R.

Δεξιά του Σχήματος είναι η διάταξη γέφυρας-H που καθορίζει την φορά περιστροφής του κινητήρα. Παρατηρούμε επίσης ότι το τελευταίο ψηφίο του καταμετρητή (Most Significant Bit) καθορίζει την φορά περιστροφής του κινητήρα. Συμπεραίνεται λοιπόν ότι χρησιμοποιείται η τεχνική συμπλήρωτος ως προς δυο (two's complement) για τον καθορισμό της φοράς. Έτσι το εύρος των τιμών σε δεκαδικό σύστημα θα είναι $-2048 \div 2048$ αρκετά καλό για μια ομαλή μετάβαση ψηφιακού σε αναλογικό σήμα.

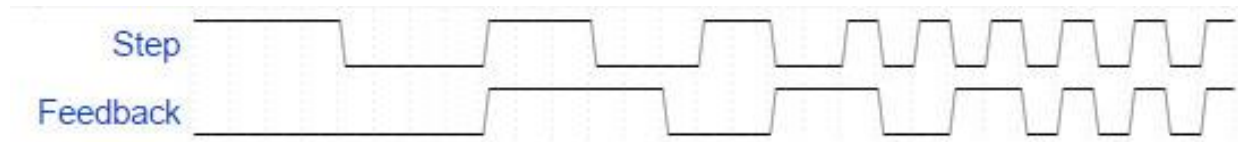
Το σήμα που στέλνεται στο τμήμα επικοινωνίας με τον επεξεργαστή είναι για τον έλεγχο υπερχείλισης (overflow) του καταμετρητή. Σε περίπτωση δηλαδή που ο καταμετρητής βγει εκτός του εύρους του θα πρέπει να επανακαθοριστούν οι αρχικές τιμές (reset).

Κάτι ακόμη που μπορούμε να επισημάνουμε είναι ο τελεστικός ενισχυτής που λειτουργεί ως απλός ακολουθητής και ανοίγει το αναλογικό τρανζίστορ διακυμαίνοντας την τάση βάσης από $0V \div 24V$ καθορίζοντας έτσι την ταχύτητα περιστροφής του κινητήρα.

Πως όμως επιτυγχάνεται αύξηση (επιτάχυνση) ή μείωση (επιβράδυνση) της ταχύτητας του κινητήρα;

Προηγουμένως αναφέρθηκε ότι για σταθερή ταχύτητα κάθε παλμός βηματισμού θα διορθωθεί αμέσως μετά από τον παλμό που θα προκαλέσει η ανάδραση. Έστω ότι επιδιώκουμε επιτάχυνση του κινητήρα. Τότε δίνονται από τον επεξεργαστή βήματα μικρότερου πλάτους -πιο γρήγορα- (σχήμα 3.27), επειδή η ανάδραση πάντα έπεται θα διορθώνει το αμέσως προηγούμενο παλμό έτσι οι καταμετρητές θα έχουν δεχθεί +1 (ή -1 ανάλογα με την κατεύθυνση) αυξάνοντας την τάση βάσης του τρανζίστορ κατάντι του

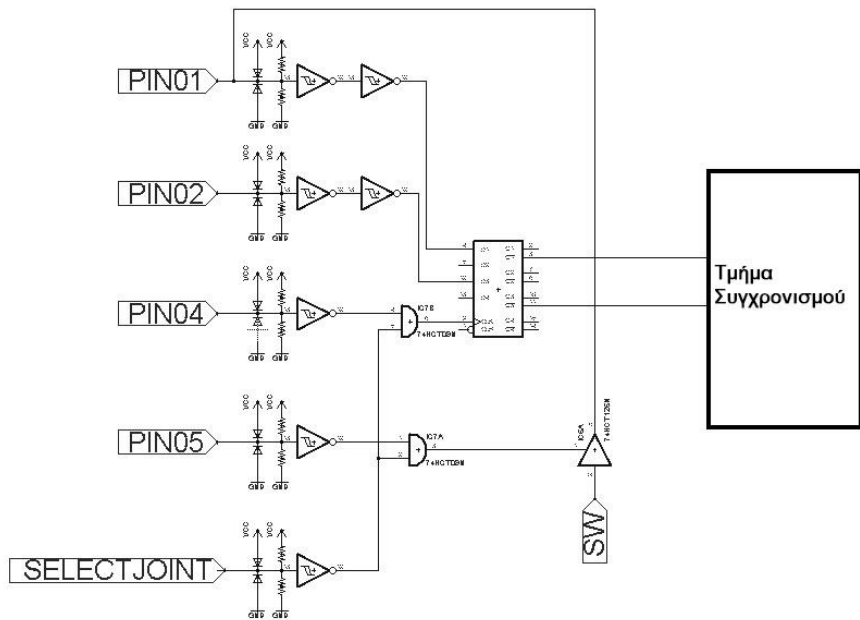
τελεστικού ενισχυτή. Με αύξηση της τάσης βάσης ενός τρανζίστορ τύπου Darlington, έχουμε αύξηση της τάσης στον εκπομπό του τρανζίστορ (βλ. Παράρτημα Α) και κατά συνέπεια σε όλη την γέφυρα-Η. Η ίδια διαδικασία συμβαίνει όταν ο κινητήρας επιβραδύνεται και τελικά σταματά.



ΣΧΗΜΑ 3.27 Η συμπεριφορά των σημάτων βηματισμού και ανάδρασης κατά την επιτάχυνση

3.2.2.4. Το τμήμα επικοινωνίας με τον Επεξεργαστή

Το τμήμα αυτό αναφέρεται τελευταίο μιας και είναι απόρροια όλων των άλλων τμημάτων και τελικά το ζητούμενο της αποκρυπτογράφησης των πλακετών. Το παρακάτω Σχήμα παρουσιάζει τα αποληκτρια καρφιά (pins) που είναι απαραίτητα για τον καθορισμό ενός από τα πέντε κυκλώματα σέρβο-ενισχυτών. Δεν παριστάνονται τα καρφιά του σταθερού χρονισμού 2MHz και του επανακαθοριστή (Master Reset) μιας και εφαρμόζονται σε όλα τα κυκλώματα χωρίς να αλλάζει η κατάσταση τους.



ΣΧΗΜΑ 3.28 Το τμήμα επικοινωνίας με τον Επεξεργαστή

Από το τμήμα του συγχρονισμού έχει καθοριστεί ότι το PIN01 είναι τα βήματα που έρχονται από τον επεξεργαστή και το PIN02 η επιθυμητή φορά περιστροφής του κινητήρα. Όμως το PIN01 δείχνει έχει δυο εισόδους! Στην ουσία το PIN01 όταν πατηθεί ο τερματικός διακόπτης (SW) στέλνει λογικό 1 με αντίθετη κατεύθυνση για να κινηθεί η άρθρωση με την ελάχιστη ταχύτητα έως ότου αποδεσμευτεί ο

διακόπτης. Το PIN01 όμως θα πρέπει να έχει μια είσοδο κάθε φορά. Για αυτό τον λόγο υπάρχουν τα PIN04 και PIN05.

Έτσι κατά την μετάβαση του PIN04 από λογικό 1 σε λογικό 0 «περνάνε» τα δεδομένα βηματισμού και κατεύθυνσης στο τμήμα συγχρονισμού (write enable ή strobe) και όταν το PIN05 είναι 0 το PIN01 διαβάζει την τιμή του τερματικού διακόπτη (read enable)

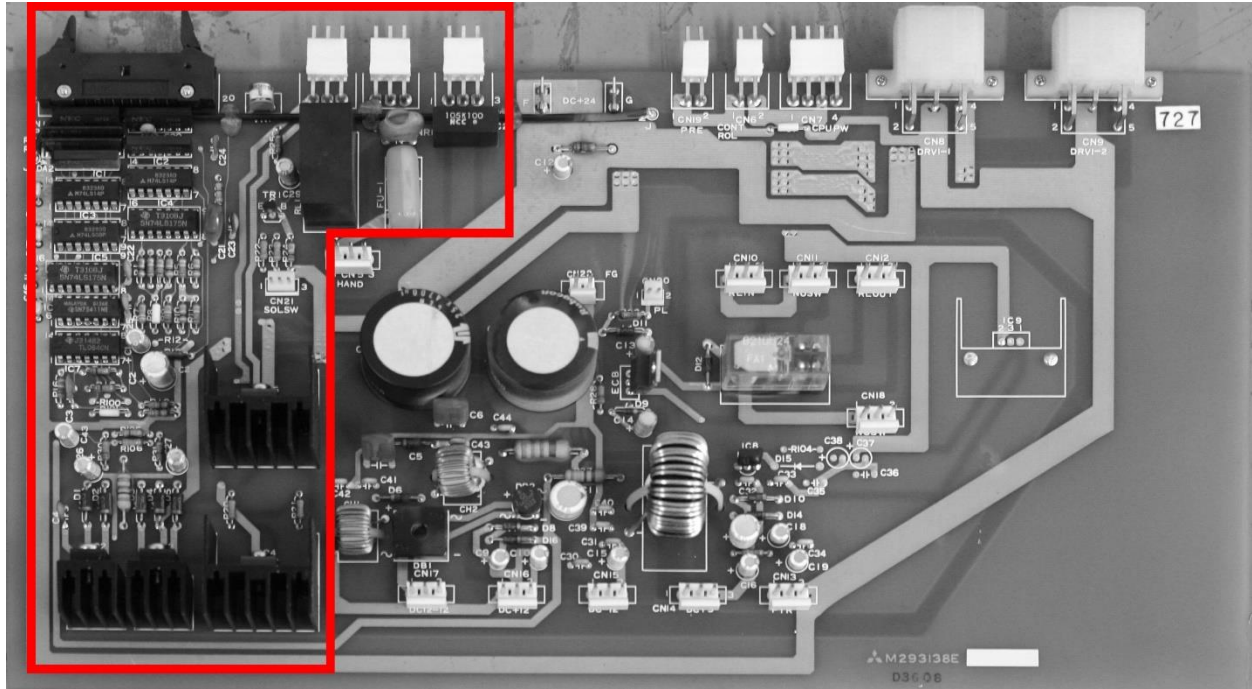
Τα PIN01 PIN02 PIN04 και PIN05 είναι κοινά καρφιά και για τα πέντε ανεξάρτητα κυκλώματα των σέρβο ενισχυτών. Για να αναφέρονται λοιπόν σε ένα από τα κυκλώματα κάθε φορά θα πρέπει να υπάρχει κάποιο καρφί που θα καθορίζει το κύκλωμα. Αυτό το καρφί παρουσιάζεται ως select joint στο σχήμα 3.2.16 και είναι διαφορετικό για κάθε κύκλωμα

Παρατηρούμε ότι όλα τα καρφιά είναι ανεστραμμένα αυτό συμβαίνει γιατί ο κατασκευαστής προτιμά τα μη συνδεδεμένα καρφιά να διατηρούν λογικό 1 για την αποφυγή της κατάστασης NC (not connected). Για τον λόγο αυτό βλέπουμε την αποληκτρία διάταξη Thevenin (Thevenin termination). Οι δυο δίοδοι που συνδέονται με τα καρφιά προστατεύουν το κύκλωμα από αιφνίδιες ακραίες αλλαγές τάσης (ESD protection)

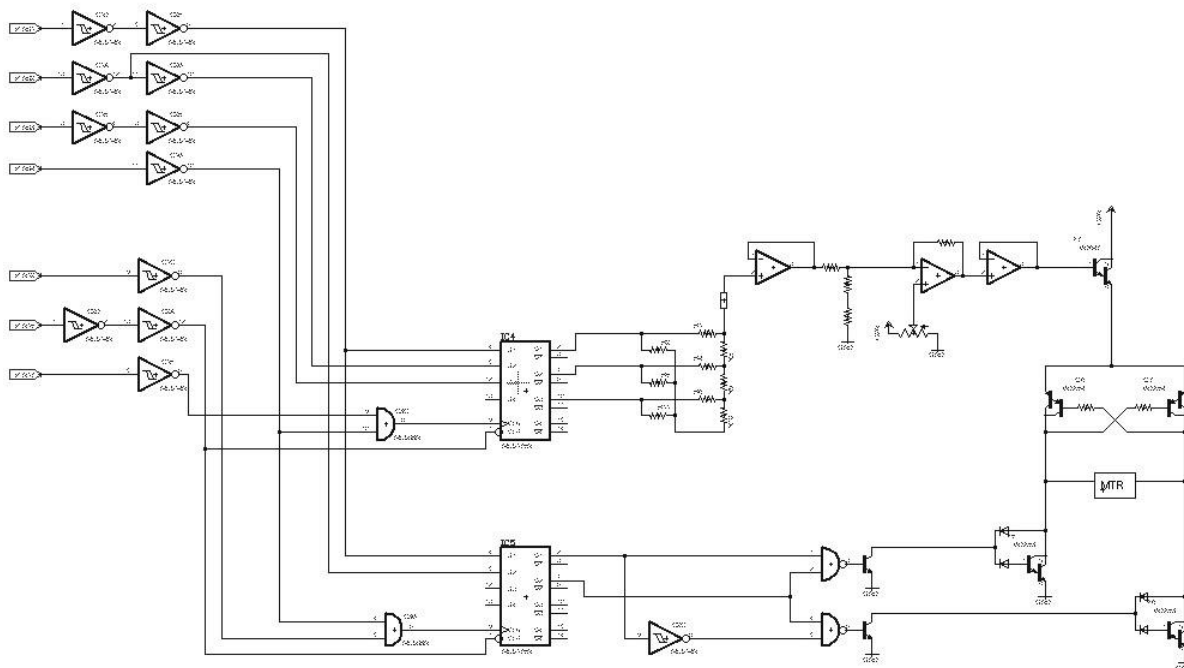
Συνοψίζοντας και Παρατηρώντας τα σχέδια της κάθε πλακέτας καταλήγουμε στον παρακάτω πίνακα

ΚΑΡΦΙ	ΛΕΙΤΟΥΡΓΙΑ	ΤΥΠΟΣ
Pin01	Βηματισμός	Είσοδος
Pin02	Κατεύθυνση	Είσοδος
Pin04	Γράψε	Είσοδος
Pin05	Διάβασε	Είσοδος
Pin07	Διάλεξε Αγκώνα	Είσοδος
Pin08	Διάλεξε Όμο	Είσοδος
Pin09	Διάλεξε Αριστερό Καρπό	Είσοδος
Pin10	Διάλεξε Δεξί Καρπό	Είσοδος
Pin11	Διάλεξε Μέση	Είσοδος
Pin13	Υπερχείλιση 724	Έξοδος
Pin14	Υπερχείλιση 732	Έξοδος
Pin15	Επανακαθοριστής	Είσοδος
Pin16	Ρολόι Χρονισμού 2MHz	Είσοδος

3.2.3. Η Πλακέτα 727 (Αρπάγη)



ΣΧΗΜΑ 3.29 Το κύκλωμα της αρπάγης πάνω στην πλακέτα 727



ΣΧΗΜΑ 3.30 Η Χαρτογράφηση του κυκλώματος αρπάγης

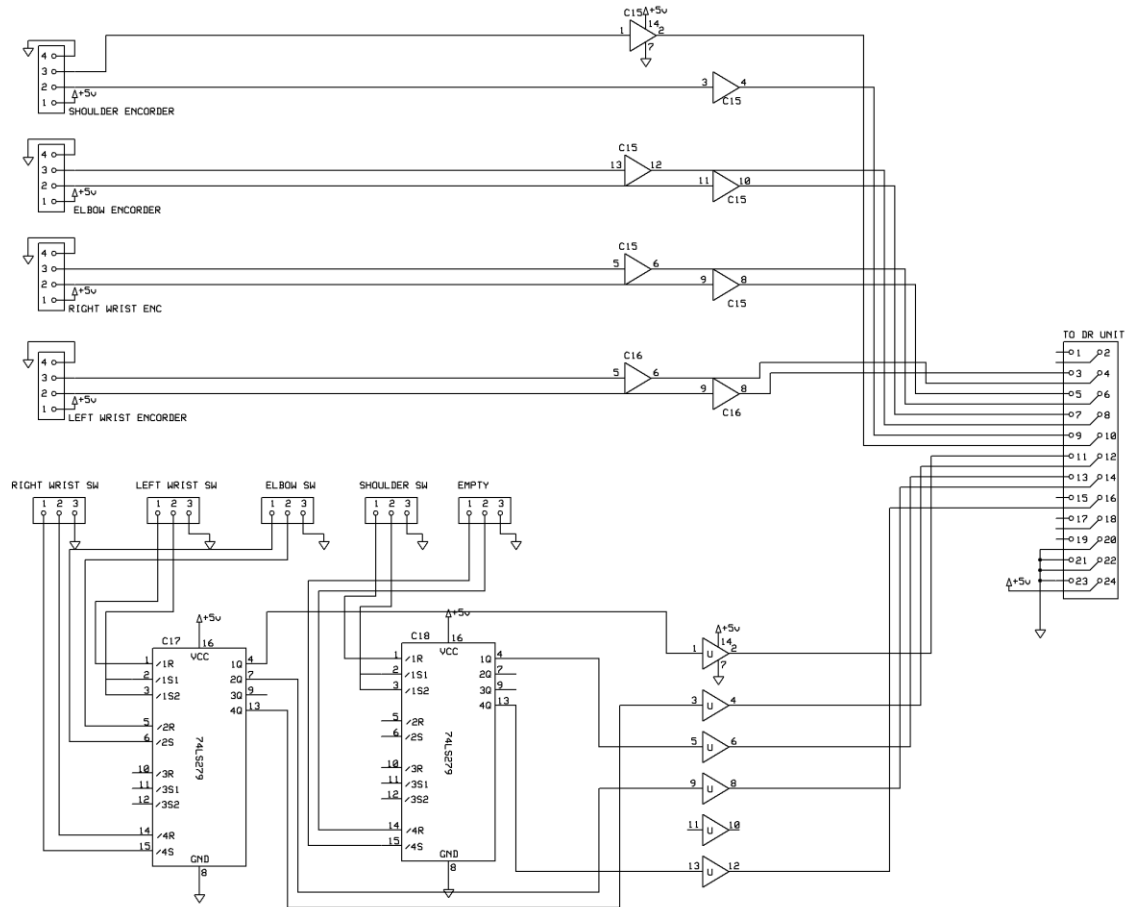
Από την πλακέτα 727 θα μας απασχολήσει μόνο το τμήμα της αρπάγης. Η γέφυρα-Η καθώς και μια παρόμοια πιο απλή διάταξη με αυτές που συναντήσαμε στις πλακέτες 724 και 732 δηλώνει ότι και η αρπάγη αποτελείται από κινητήρα βηματικού τύπου. Μάλιστα εδώ γίνεται διάκριση στον βηματισμό και στην κατεύθυνση με δυο διαφορετικά flip flop. Έτσι υπάρχουν δυο καρφιά που παίζουν το ρόλο του «Διάλεξε». Το PIN12 έχει σαν λειτουργία το διάλεξε κατεύθυνση και το PIN17 διάλεξε ταχύτητα.

Παρατηρούμε και την παρουσία του PIN03 που δεν κάνει τίποτε άλλο από το να προσθέτει ακόμη 5V στην κλίμακα R-2R για την αύξηση της τάσης στη γέφυρα-Η

Με τον καθορισμό όλων των καρφιών και από τις τρεις πλακέτες που επικοινωνούν με τον επεξεργαστή έχω τον τελικό πίνακα

ΚΑΡΦΙ	ΛΕΙΤΟΥΡΓΙΑ	ΤΥΠΟΣ
Pin01	Βηματισμός	Είσοδος
Pin02	Κατεύθυνση	Είσοδος
Pin03	Βοηθητικό στην Αρπάγη	Είσοδος
Pin04	Γράψε	Είσοδος
Pin05	Διάβασε	Είσοδος
Pin06	Εκτός Σύνδεσης	
Pin07	Διάλεξε Αγκώνα	Είσοδος
Pin08	Διάλεξε Όμο	Είσοδος
Pin09	Διάλεξε Αριστερό Καρπό	Είσοδος
Pin10	Διάλεξε Δεξί Καρπό	Είσοδος
Pin11	Διάλεξε Μέση	Είσοδος
Pin12	Διάλεξε Κατεύθυνση Αρπάγης	Είσοδος
Pin13	Υπερχείλιση 724	Έξοδος
Pin14	Υπερχείλιση 732	Έξοδος
Pin15	Επανακαθοριστής	Είσοδος
Pin16	Ρολόι Χρονισμού 2MHz	Είσοδος
Pin17	Διάλεξε Ταχύτητα Αρπάγης	Είσοδος
Pin18	Εκτός Σύνδεσης	
Pin19	Εκτός Σύνδεσης	
Pin20	Εκτός Σύνδεσης	

3.2.4. Η Πλακέτα Επικοινωνίας με τον Ρομποτικό βραχίονα



ΣΧΗΜΑ 3.31 Το κύκλωμα Επικοινωνίας με τον Ρομποτικό βραχίονα

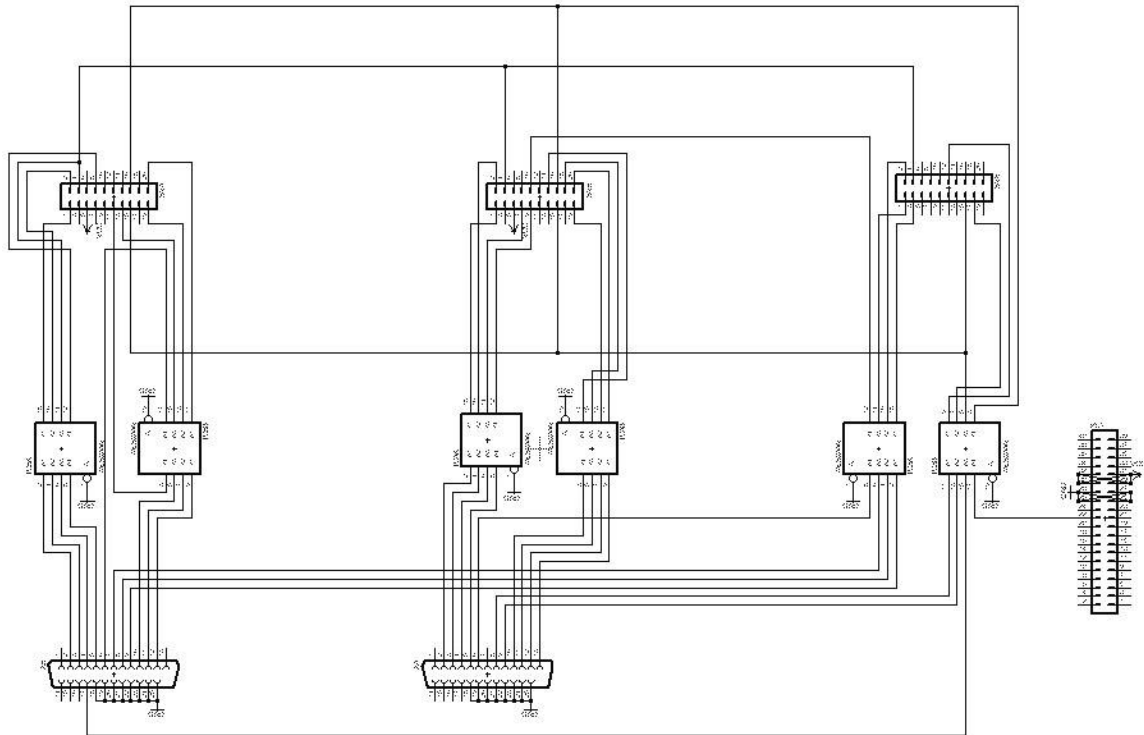
Η πλακέτα Επικοινωνίας με τον Ρομποτικό βραχίονα χαρτογραφήθηκε για λόγους πληρότητας. Η Πλακέτα αυτή είναι υπεύθυνη για τον καθαρισμό του θορύβου από τα σήματα των κωδικοποιητών των κινητήρων που στέλνονται στα τμήματα ανάδρασης της μονάδας ελέγχου. Η πλακέτα αναλαμβάνει επίσης να μετατρέψει το διπολικό σήμα από τους τερματικούς διακόπτες σε μονοπολικό αυτό γίνεται με set-reset flip flops (βλ. Παράρτημα Α: SN74LS279)

3.2.5. Η Πλακέτα επικοινωνίας με Η/Υ

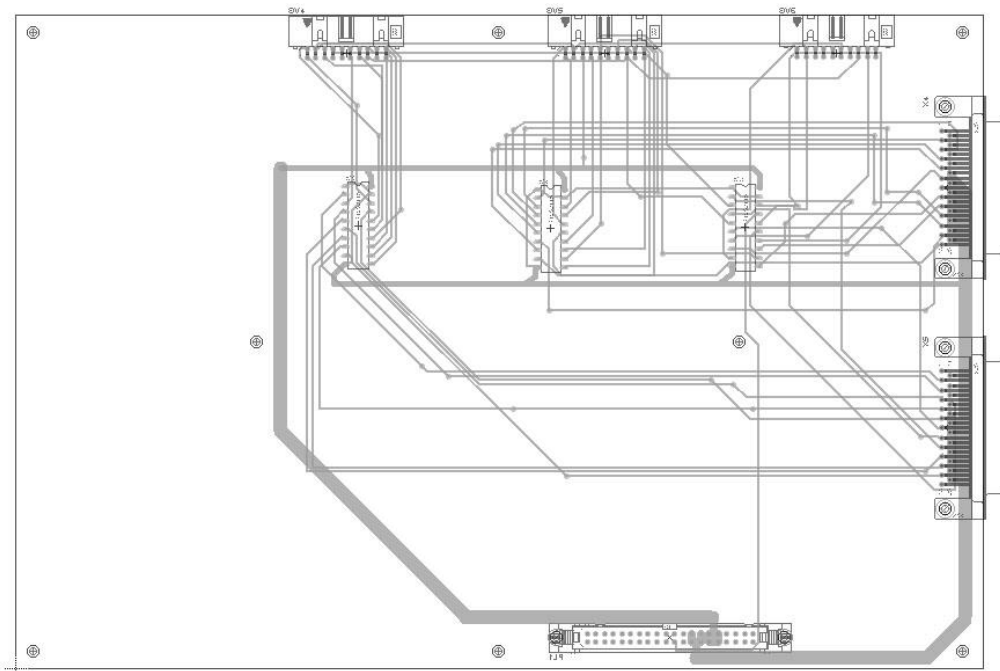
Όπως αναφέρθηκε και στην εισαγωγή αυτής της ενότητας πρέπει να αντικατασταθεί ο επεξεργαστής που αναλαμβάνει την παραγωγή βημάτων και κατεύθυνσης από σύγχρονο ηλεκτρονικό υπολογιστή. Για να γίνει έπρεπε να τυπωθεί μια νέα πλακέτα (βλ. σχήμα 3.32). Για την σχεδίαση της λήφθηκαν υπόψιν τα εξής:

- Παρότι οι πλακέτες 724 732 και 727 ήταν συνδεδεμένες μεταξύ τους σε σειρά και κατέληγαν στην πλακέτα του επεξεργαστή σε μια θύρα 20 καρφιών, αποφασίστηκε η νέα πλακέτα να έχει τρεις θύρες 20 καρφιών που η κάθε μια θα επικοινωνεί με μια πλακέτα. Ο λόγος για τον οποίο έγινε αυτό εξηγείται στην δεύτερη ενότητα του τρίτου κεφαλαίου.
- Για τον εντοπισμό των τερματικών σημείων από τον ηλεκτρονικό υπολογιστή, η διαχείριση των τερματικών διακοπών πρέπει να γίνεται από το πρόγραμμα LinuxCNC και όχι από τους ίδιους τους σέρβο ενισχυτές. Για τον λόγο αυτό το σήμα από τους διακόπτες αποφασίστηκε να σταλεί στα κενά καρφιά 18 19 και 20 της κάθε πλακέτας έτσι ώστε να αποτελεί είσοδο (input) στην παράλληλη θύρα του υπολογιστή. Επίσης για την απενεργοποίηση του ενεργοποιητή «διάβασε» (read enable), στέλνεται μόνιμο λογικό 1 στο καρφί PIN05
- Ο υπολογιστής δεν μπορεί να παράγει τετραγωνικό σήμα 2MHz από την παράλληλη θύρα έτσι εκμεταλλευτήκαμε την κρυσταλλικό ταλαντωτή (crystal oscillator) του παλιού κυκλώματος για να παράγουμε το κατάλληλο σήμα.
- Για την αμφότερη μετάδοση καθαρών σημάτων ανάμεσα στον υπολογιστή και την μονάδα ελέγχου του ρομποτικού βραχίονα χρησιμοποιήθηκαν τα ολοκληρωμένα κυκλώματα SN74LS244 (βλ. Παράρτημα Α).
- Τέλος έπρεπε να προβλεφθεί η κοινή γείωση αναμεσα στον υπολογιστή και την μονάδα ελέγχου, καθώς επίσης και τα 5V τροφοδοσίας των ολοκληρωμένων κυκλωμάτων SN74LS244.

Με γνώμονα τα παραπάνω καθώς και με την βοήθεια του λογισμικού eagle σχεδιάστηκε η πλακέτα όπως φαίνεται στο σχήμα 3.21. Το μεγάλο της μέγεθος παρόλο το μικρό κύκλωμα που φέρει έχει σαν σκοπό να αντικαταστήσει την θέση της πλακέτας που αχρηστεύθηκε μέσα στην Μονάδα Ελέγχου.



ΣΧΗΜΑ 3.32 Το Κύκλωμα της πλακέτας



ΣΧΗΜΑ 3.33 Η Πλακέτα με τα στοιχεία της

Η αντιστοιχία των καρφιών των δυο παράλληλων θυρών με αυτά των πλακετών 724 732 και 727 παρουσιάζεται στον παρακάτω πίνακα

ΠΑΡΑΛΛΗΛΗ ΘΥΡΑ 0	ΟΝΟΜΑΣΙΑ ΣΗΜΑΤΟΣ	ΚΑΡΦΙ ΠΛΑΚΕΤΑΣ
0.Pin1	-	
0.Pin2	STEP_1	724.Pin1
0.Pin3	DIR_1	724.Pin2
0.Pin4	SROBE	724.Pin4 732.Pin4 727.Pin4
0.Pin5	ELBOW SELECT	724.Pin8
0.Pin6	WAIST SELECT	724.Pin11
0.Pin7	HAND_1	727.Pin1
0.Pin8	HAND_2	727.Pin2
0.Pin9	HAND_3	727.Pin3
0.Pin10	OVERFLOW 724	724.Pin14
0.Pin11	SWITCH ELBOW	724.Pin18
0.Pin12	SWITCH WAIST	724.Pin20
0.Pin13	-	
0.Pin14	-	
0.Pin15	-	
0.Pin16	-	
0.Pin17	MASTER RESET	724.Pin15 732.Pin15 727.Pin15
0.Pin18	GND	
0.Pin19	GND	
0.Pin20	GND	
0.Pin21	GND	
0.Pin22	GND	
0.Pin23	GND	
0.Pin24	GND	
0.Pin25	GND	

ΠΑΡΑΛΛΗΛΗ ΘΥΡΑ 1	ΟΝΟΜΑΣΙΑ ΣΗΜΑΤΟΣ	ΚΑΡΦΙ ΠΛΑΚΕΤΑΣ
1.Pin1	STEP_2	732.Pin1
1.Pin2	-	
1.Pin3	DIR_2	732.Pin2
1.Pin4	SHOULDER SELECT	732.Pin7
1.Pin5	LEFT WRIST SELECT	732.Pin9
1.Pin6	RIGHT WRIST SELECT	732.Pin10
1.Pin7	-	
1.Pin8	HAND DIRECTION	727.Pin12
1.Pin9	HAND POWER	727.Pin17
1.Pin10	OVERFLOW 732	
1.Pin11	SWITCH SHOULDER	732.Pin18
1.Pin12	SWITCH PITCH	732.Pin19
1.Pin13	SWITCH ROLL	732.Pin20
1.Pin14	-	
1.Pin15	-	
1.Pin16	-	
1.Pin17	-	
1.Pin18	GND	
1.Pin19	GND	
1.Pin20	GND	
1.Pin21	GND	
1.Pin22	GND	

1.Pin23	GND
1.Pin24	GND
1.Pin25	GND

ΚΕΦΑΛΑΙΟ 4. Λογισμικό αριθμητικού ελέγχου

4.1. Η δομή του LinuxCNC

4.1.1. Ιστορία

Το λογισμικό LinuxCNC υπάρχει από τις αρχές της δεκαετίας. Αρχικά, είχε το όνομα EMC (Enhanced Machine Controller) και δημιουργήθηκε από το Εθνικό Ινστιτούτο Τυποποίησης και Τεχνολογίας (NIST) των Η.Π.Α. Σκοπός ήταν να δημιουργηθεί ένα τυποποιημένο πακέτο ελέγχου μηχανών CNC με τη χρήση ηλεκτρονικού υπολογιστή. Η πρώτη προσπάθεια ελέγχου έγινε χρησιμοποιώντας πλακέτες ευφυούς ελέγχου PMAC και λογισμικού EMC και δομήθηκε πάνω σε έκδοση Windows NT «real time» για τον έλεγχο μεγάλης φρέζας.

Η ανάγκη για λιγότερο ακριβό έλεγχο, καθώς και η αναβάθμιση ξεπερασμένων και δυσλειτουργικών λογισμικών έκανε τον Matt Shaver να απευθυνθεί στο Εθνικό Ινστιτούτο Τυποποίησης και Τεχνολογίας και να ζητήσει να γίνει το λογισμικό προσιτό τόσο από οικονομική άποψη όσο και από θέμα ευελιξίας. Έτσι, γεννήθηκε η ιδέα του λογισμικού EMC.

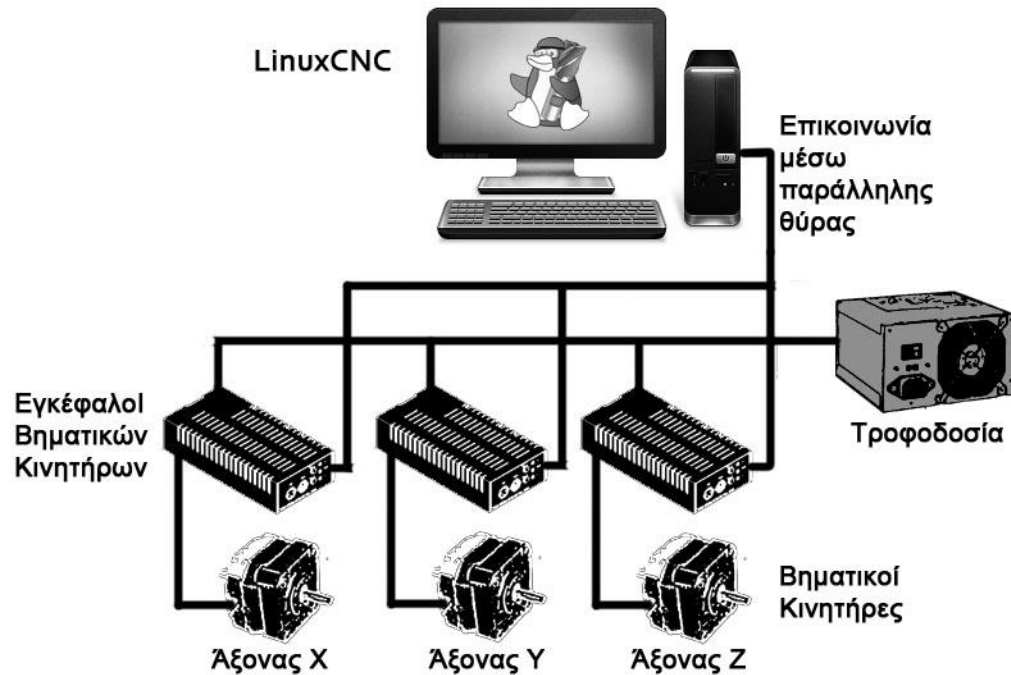
Η πρώτη παρέμβαση είχε σκοπό να «ξεφορτωθούν» το ακριβό και αναξιόπιστο «real time» Windows NT, με αφορμή μια νέα επέκταση του λογισμικού Linux που υποστήριζε πράξεις-ενέργειες σε πραγματικό χρόνο. Το επόμενο βήμα ήταν να αντικαταστήσουν τις ακριβές, ευφυείς πλακέτες, οι οποίες ήταν υπεύθυνες για τον έλεγχο της κίνησης. Η πρώτη πλακέτα που δημιουργήθηκε για αυτό το σκοπό ήταν η Servo-To-Go και ήταν ο δίαυλος ελέγχου των κινητήρων από τον υπολογιστή. Το λογισμικό αναβαθμίστηκε επιτρέποντας τον PID έλεγχο καθώς και την επικοινωνία με το μεταφραστή RS274 (G-code interpreter).

Όταν ξεκίνησαν να «χτίζονται» τα πρώτα CNC με αυτό το λογισμικό, το ενδιαφέρον από την κοινότητα των εργαλειομηχανών άρχισε να μεγαλώνει. Επειδή το EMC ήταν ένα ανοιχτό πρόγραμμα, μπορούσε ο καθένας να το μεταποιήσει. Έτσι, υπήρξαν πολλές προσπάθειες βελτίωσής του από διάφορες προσεγγίσεις. Το μεγάλο πρόβλημα ήταν ότι κάθε τέτοια προσπάθεια απαιτούσε παρεμβάσεις και ανακατατάξεις στον πυρήνα των Linux, στο τμήμα του κώδικα πραγματικού χρόνου (RTAI). Το πρόβλημα ξεπεράστηκε αφού δημιουργήθηκε ένα CD όπου εγκαθιστούσε ταυτόχρονα το πρόγραμμα EMC, το λογισμικό Linux καθώς και το συμπληρωματικό κομμάτι του πυρήνα, των Linux.

Το 2003 η κοινότητα του EMC οργάνωσε το πρώτο συνέδριο. Μια κάπως ανορθόδοξη προσπάθεια που όμως έδωσε την ιδέα μιας εικονικής πλακέτας (HAL: Hardware Abstractive Layer), η οποία κάνει πιο κατανοητή την επικοινωνία του προγράμματος με τον «έξω κόσμο».

Πλέον, το πρόγραμμα EMC ή LinuxCNC -όπως μετονομάστηκε- απευθύνεται σε ένα πολύ μεγάλο κοινό. Η ελευθέρια επιλογής γραφικού περιβάλλοντος, καθώς και η ευελιξία του προγράμματος να προσαρμόζεται σε μηχανές έως και εννέα βαθμών ελευθέριας, το έχουν καθιερώσει ως ένα από τα γνωστότερα προγράμματα ελέγχου εργαλειομηχανών.

4.1.2. Τυπική διάταξη LinuxCNC



ΣΧΗΜΑ 4.1 Τυπική διάταξη LinuxCNC

Το παραπάνω σχήμα μας δείχνει μια απλή διάταξη μιας εργαλειομηχανής τριών αξόνων χρησιμοποιώντας το λογισμικό LinuxCNC. Ο υπολογιστής, μέσω του LinuxCNC, μεταφράζει τον G-κώδικα και αναλόγως με τη διαμόρφωση της εργαλειομηχανής αναλαμβάνει στρατηγικά την κίνηση των Βηματικών Κινητήρων, στέλνοντας τα απαραίτητα σήματα. Τα σήματα αυτά, συμπεριλαμβάνονται στον Οδηγό (Driver Board) του εκάστοτε Βηματικού Κινητήρα και τα μετατρέπει σε βήματα του κινητήρα.

4.1.3. Τι είναι το LinuxCNC

Το LinuxCNC είναι ένα λογισμικό σύστημα για τον έλεγχο, μέσω υπολογιστή, εργαλειομηχανών όπως: φρέζες, τόνους, ρομποτικούς βραχίονες και γενικά οποιαδήποτε μηχανή υπόκειται σε έλεγχο και δεν ξεπερνά τους εννέα βαθμούς ελευθέριας. Είναι ένα ανοιχτό πρόγραμμα (open source) και ακολουθεί τα GNU General Public License και Lesser GNU General Public License (GPL & LGPL).

Το LinuxCNC συνεργάζεται είτε με σερβοκινητήρες αναλογικού ή ψηφιακού τύπου (PWM), ανοικτού ή κλειστού βρόγχου είτε με απλούς βηματικούς κινητήρες. Μπορεί να υποστηρίξει κινηματικά μοντέλα μη καρτεσιανού επιπέδου, όπως κίνηση εξάποδων μηχανών. Επίσης, επικοινωνεί με μεγάλη ποικιλία περιφερειακών συστημάτων.

Παρόλα αυτά, δεν έχει τη δυνατότητα να επικοινωνεί με σχέδια CAM, ούτε με G-κώδικα προερχόμενο από CAM εφαρμογές ακόμη.

Γενικά, το LinuxCNC προσφέρει:

- Επιλογή μέσα από 6 γραφικά περιβάλλοντα.
- Δυνατότητα δημιουργίας γραφικού περιβάλλοντος της επιλογής του διαμορφωτή.
- Μεταφραστή G-κώδικα σε θέσεις και ταχύτητες (RS-274).
- Κίνηση Πραγματικού Χρόνου (Real Time Motion).
- Δυνατότητα προγραμματισμού με PLC (Classic Ladder).
- Εικονική πλακέτα για ευελιξία στη διαμόρφωση του προγράμματος πάνω σε κάθε εργαλειομηχανή (HALayer).
- Επικοινωνία με διάφορα περιφερειακά συστήματα, όπως αισθητήρες, διακόπτες, LED.
- Εύκολη εγκατάσταση.

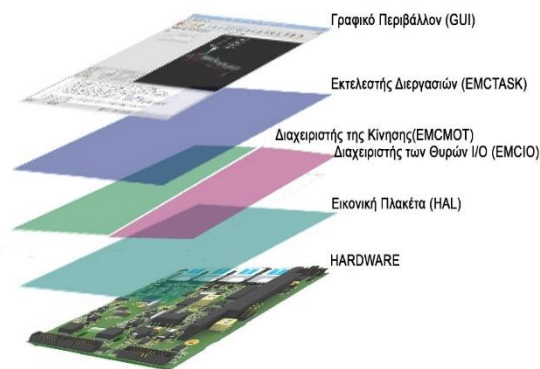
4.1.4. Πώς λειτουργεί το LinuxCNC

Το LinuxCNC δεν είναι απλά ένα ακόμη πρόγραμμα για κάποια εργαλειομηχανή. Μπορεί και αναλαμβάνει τον έλεγχο σερβοκινητήρων και γενικά οποιαδήποτε συσκευή με την οποία συνεργάζεται η εργαλειομηχανή.

Υπάρχουν τέσσερα βασικά τμήματα-μονάδες στο λογισμικό του προγράμματος:

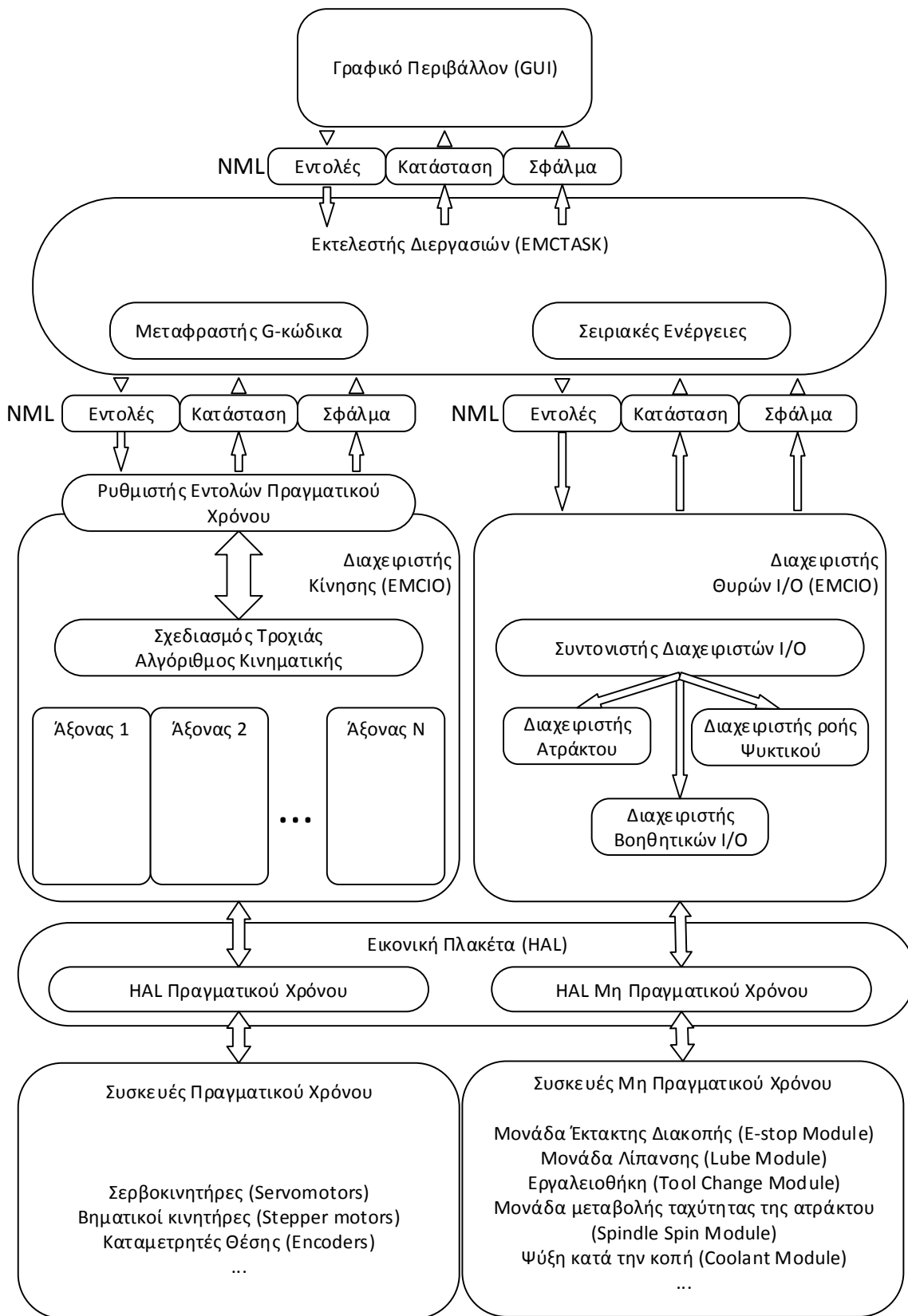
- Γραφικό Περιβάλλον (GUI)
- Εκτελεστής Διεργασιών (EMCTASK)
- Διαχειριστής Κίνησης (EMCMOT)
- Διαχειριστής Θυρών I/O (EMCIO)

Επιπλέον, υπάρχει ένα επίπεδο (HAL) που συμπεριφέρεται σαν εικονική πλακέτα, το οποίο βοηθάει στη διαμόρφωση του προγράμματος πάνω σε κάποια εργαλειομηχανή χωρίς να απαιτείται η γραφή πρωτογενούς κώδικα από το Διαμορφωτή (Integrator).



ΣΧΗΜΑ 4.2 Τα Επίπεδα του LinuxCNC

Παρουσιάζεται ένα διάγραμμα ροής με τις κυρίες διεργασίες του προγράμματος LinuxCNC.



4.1.4.1. Γραφικό Περιβάλλον (GUI)

Το Γραφικό Περιβάλλον (Graphical User Interface) είναι ο σύνδεσμος με τον οποίο επικοινωνεί ο χρήστης με το LinuxCNC. Σε αυτό γράφονται οι εντολές G-κώδικα από το χρήστη αλλά και η εργαλειομηχανή δίνει πληροφορίες, όπως η τρέχουσα θέση του εργαλείου, λίπανση κ.α.

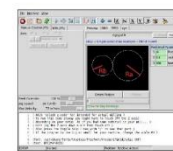
Το LinuxCNC είναι ένα ανοικτό λογισμικό με αποτέλεσμα να έχουν δημιουργηθεί πολλά γραφικά περιβάλλοντα, με έξι από αυτά να είναι τα επικρατέστερα:

- **AXIS GUI.** Πρόκειται για το πιο ανθρωποστραφές περιβάλλον περιλαμβάνοντας, monitors, εικονικά LED και γενικά αρκετά γραφικά, για την εύκολη χρήση τού προγράμματος.
- **NGC GUI.** Είναι ένα περιβάλλον το οποίο χτίζει G-κώδικα από υπορουτίνες. Ο συνδυασμός των υπορουτίνων, μπορούν να αποτελέσουν ένα ολοκληρωμένο πρόγραμμα. Το περιβάλλον NGC μπορεί να χρησιμοποιηθεί είτε μόνο του, είτε να κληθεί από το AXIS GUI.
- **Touchy GUI.** Περιβάλλον το οποίο χτίστηκε για panel monitors (touch screens).
- **TkLinuxCNC GUI.** Είναι ένα από τα πρώτα περιβάλλοντα και δομήθηκε πάνω σε Tk-Tcl πλατφόρμα. Έχει πολύ μικρές, υπολογιστικές απαιτήσεις και για αυτό έχει μεγάλη εφαρμογή σε επεξεργαστές μικρών δυνατοτήτων.
- **MINI GUI.** Σχεδιάστηκε με σκοπό το περιβάλλον να καταλαμβάνει πλήρη οθόνη και να μην έχει τη φιλοσοφία τού παραθύρου. Βρίσκει εφαρμογή σε μικρές οθόνες που συχνά συνοδεύουν μια εργαλειομηχανή.
- **KEYSTICK GUI.** Είναι η μορφή γραφικού περιβάλλοντος εξαιρετικά απλή στη χρήση. Σαν φιλοσοφία έχει την απλή εφαρμογή G-κώδικα

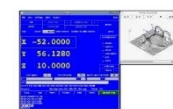
Όλα τα περιβάλλοντα είναι σε μια συνεχή διαδικασία βελτίωσης και μπορεί ο καθένας να λάβει μέρος σε αυτή αρκεί να ακολουθεί τους ορούς GPL.



Axis GUI



NGC GUI



TkLinuxCNC GUI



Touchy GUI



Mini GUI



Keystick GUI

ΣΧΗΜΑ 4.3

Διάφορα GUI

4.1.4.2. Εκτελεστής Διεργασιών (EMCTASK)

Ο Εκτελεστής Διεργασιών στην ουσία είναι μια συντονιστική μονάδα η οποία παρεμβάλλεται αναμεσα στο Γραφικό Περιβάλλον και τους Διαχειριστές Κίνησης και I/O. Σκοπός του είναι να επικοινωνεί και να διευθετεί τις ακολουθίες εντολών. Επίσης, μέσα σε αυτό το τμήμα γίνεται η μετάφραση του G-κώδικα (RS-274). Ο τρόπος που επικοινωνεί με τις άλλες τρεις μονάδες, GUI, EMCIO, EMC MOT, είναι μέσω εντολών NML. Οι εντολές NML είναι οι κατάλληλες για την επικοινωνία με τη μονάδα πραγματικού χρόνου EMC MOT καθώς έχουν τη δυνατότητα να συγκεντρώνουν πολλές πληροφορίες στον ίδιο ρυθμιστή (buffer).

4.1.4.3. Διαχειριστής Κίνησης (EMCMOT)

Ο Διαχειριστής Κίνησης είναι υπεύθυνος για το χειρισμό των κινήσεων της εργαλειομηχανής. Διενεργεί σε πραγματικό χρόνο και περιέχει δυο βασικές υπομονάδες:

- Τον Σχεδιαστή Τροχιάς (Trajectory Planner)
- Και τους Άξονες (AXIS)

Ο Σχεδιαστής Τροχιάς δέχεται τις μεταφρασμένες εντολές G-κώδικα και τις τροποποιεί, αναλόγως με το είδος και τη μορφή της εργαλειομηχανής. Για να γίνει αυτό, θα πρέπει ο διαμορφωτής του προγράμματος (integrator) να εισάγει κινηματικούς αλγόριθμους (kinematics), σύμφωνα με τη διάταξη των επενεργητών και τον τρόπο κίνησης της εργαλειομηχανής. Για παράδειγμα, για την κίνηση ενός ρομποτικού βραχίονα χρειαζόμαστε αλγόριθμο αντίστροφης κινηματικής ενώ μια φρέζα τριών βαθμών ελευθέρια ενδεχομένως να μη χρειάζεται αλγόριθμο, μιας και η μεταβολή της θέσης του εργαλείου μπορεί να εκφραστεί ένα προς ένα (X,Y,Z : κινητήραςX, κινητήραςY, κινητήραςZ).

Οι Άξονες (AXIS) είναι τόσο όσοι και οι κινητήρες. Στην ουσία οι Άξονες αναλαμβάνουν να δώσουν την κάθε πληροφορία για την λειτουργία του εκάστοτε κινητήρα, όπως επίσης να λάβουν την όποια πληροφορία (feedback) από τον καταμετρητή θέσης (encoder) σερβοκινητήρα. Στις αρμοδιότητες των αξόνων είναι και ο PID έλεγχος. Ο μέγιστος αριθμός Αξόνων που μπορεί να υποστηρίξει το LinuxCNC είναι εννέα, περιλαμβάνοντας σχεδόν όλες τις δυνατές διατάξεις εργαλειομηχανών, αφού μόνο κάποιες εφαρμογές ευφυούς ελέγχου, όπως μηχανικά χεριά, ξεπερνούν τις εννέα αρθρώσεις. Οι Άξονες (AXIS) συχνά συγχέονται με το Γραφικό Περιβάλλον AXIS GUI ή τους άξονες του καρτεσιανού συστήματος. Για το λόγο αυτό είναι απαραίτητο να γίνεται κατανοητό πότε αναφερόμαστε στην υπομονάδα (sub module) Άξονας (AXIS) και πότε στο καρτεσιανό σύστημα.

4.1.4.4. Διαχειριστής Θυρών I/O (EMCIO)

Ο Διαχειριστής Θυρών I/O αναλαμβάνει όλες τις διεργασίες που δεν απαιτείται πραγματικός χρόνος. Επικοινωνεί με το hardware και το Γραφικό Περιβάλλον, συναλλάσσοντας πληροφορίες για τα περιφερικά συστήματα της εργαλειομηχανής όπως: η λίπανση, η ροή ψυκτικού, η αλλαγή εργαλείου, η ταχύτητα της ατράκτου (για τόννευση) και γενικά όποια άλλη εντολή M G-κώδικα. Για την ομαλή λειτουργία της μονάδας I/O υπάρχει μια υπομονάδα που συντονίζει όλες τις παραπάνω λειτουργίες, ο συντονιστής διαχειριστών (coordinating controller). Γενικά, στη μονάδα I/O θα συναντήσουμε τρεις διαχειριστές:

- Διαχειριστής Ατράκτου (spindle speed controller)
- Διαχειριστής Ροής Ψυκτικού Μέσου (coolant controller)
- Διαχειριστής Βοηθητικών I/O (auxiliary controller)

Οι διαχειριστές αυτοί όπως και οι άξονες (axis) της μονάδας EMCMOT έχουν κάποια εικονικά «καρφιά» (pins) για να επικοινωνούν με την Εικονική Πλακέτα (HAL).

4.1.4.5. Εικονική Πλακέτα (Hardware Abstractive Layer)

Μια βασική αρχή του LinuxCNC είναι να προσαρμόζεται ευκολά σε κάθε τύπο εργαλειομηχανής. Αυτό δεν θα ήταν εφικτό αν δεν υπήρχε η εικονική πλακέτα (HAL). Η Εικονική Πλακέτα είναι ο συνδεδετικός κρίκος αναμεσα στο πρόγραμμα και τον «έξω κόσμο». Είναι μια ιδέα που δημιουργήθηκε για την αποφυγή επιπρόσθετου hardware για κάθε μηχανή -πράγμα ακριβό και ενδεχομένως αναξιόπιστο- αλλά και την αποφυγή γραφής κώδικα που θα έπρεπε να αγγίξει τον πυρήνα του λογισμικού (kernel). Έτσι, δημιουργήθηκε ένα επίπεδο στο οποίο ο Διαμορφωτής Προγράμματος μπορεί να καλέσει κάποια εικονικά στοιχεία (HAL Components) να τα ενώσει με εικονικά καλώδια-σήματα (HAL signals) πετυχαίνοντας έτσι την επιθυμητή επικοινωνία του προγράμματος με τον «έξω κόσμο».

Τα στοιχεία αυτά είτε είναι κλασικές διατάξεις ηλεκτρονικής: inverters, flip flops, multiplexers κ.α. είτε πρωτότυπες διατάξεις για τις απαιτήσεις του προγράμματος: παραγωγοί βημάτων (step generators), μετρητές βημάτων (step counters), παραγωγοί σημάτων (signal generators), κ.α. Η κοινότητα του LinuxCNC συχνά βελτιώνει τα στοιχεία αυτά ή δημιουργεί νέα, ανάλογα με τις ανάγκες που προκύπτουν.

Η Εικονική Πλακέτα (HAL) αναλαμβάνει την επικοινωνία με τις φυσικές συσκευές διαχωρίζοντας τις σε πραγματικού (real-time) και μη πραγματικού χρόνου (non real-time). Τις συσκευές που απαιτείται πραγματικός χρόνος για την απόκρισή τους, διαχειρίζονται στοιχεία πραγματικού χρόνου (real time components), ενώ για αυτές που δεν απαιτείται πραγματικός χρόνος, τις διαχειρίζεται το HALUI. Το HALUI είναι ένα εικονικό στοιχείο που επικοινωνεί με φυσικούς κόμβους, διακόπτες, LED κ.α.

4.2. Διαχείριση του ρομποτικού βραχίονα με το πρόγραμμα LinuxCNC

Για την διαχείριση του ρομποτικού βραχίονα με το πρόγραμμα LinuxCNC θα πρέπει να επεξεργαστούμε όλες τις πληροφορίες που δοθήκαν στα παραπάνω κεφάλαια. Θα πρέπει δηλαδή να συντάξουμε ένα αρχείο C έτσι ώστε ο βραχίονας να υπακούει στην κινηματική θεωρία, ένα αρχείο HAL έτσι ώστε να υπάρξει η επικοινωνία της μονάδας ελέγχου με τον υπολογιστή και τέλος το αρχείο .ini που θα καθορίζει τις παραμέτρους για το συγκεκριμένο βραχίονα (λογούς μετάδοσης, μονάδες αναφοράς (machine units), ποιο λειτουργικό περιβάλλον επιθυμούμε, κ.α.).

Στις παρακάτω υποενότητες θα αναφερθούμε για καθένα από τα τρία αρχεία ξεχωριστά. Τα αρχεία παρατίθενται στο ΠΑΡΑΡΤΗΜΑ Β

4.2.1. Το αρχείο `rm501.c`

Το αρχείο `rm501.c` είναι στην ουσία ο αλγόριθμος της κινηματικής θεωρίας που παρουσιάστηκε στην υποενότητα 2.2 του Κεφαλαίου 2. Όπως παρατηρείται στο αρχείο (ΠΑΡΑΡΤΗΜΑ Β) έχουν γραφτεί ως υπορουτίνες η ευθεία και αντιστροφή κινηματική και καλούνται από το βασικό αλγόριθμο. Η διαταξη αυτή κατακερματίζει σε επιμέρους ανεξάρτητα τμήματα τον αλγόριθμο και επιτρέπει την εύκολη ανεύρεση και αντιμετώπιση λαθών (debugging).

Η ευθεία κινηματική δείχνει να μην είναι απαραίτητη μιας και καλούμαστε να καθορίσουμε τις μοίρες των αρθρώσεων δεδομένου του σημείου προσέγγισης. Παράλυτα είναι χρήσιμη γιατί επαληθεύει τις τιμές από την αναστροφή κινηματική δίνοντας έναν πιο σταθερό αλγόριθμο, όπως επίσης βοηθά και στην απόρριψη της συμμετρικής λύσης που προκύπτει από την επίλυση της αντιστροφής κινηματικής. Αυτό επιτυγχάνεται με κάποιους δείκτες (flags) οι οποίοι ενεργοποιούνται, αν οι επίλυση της ευθείας κινηματικής με τις τιμές των αρθρώσεων της αντιστροφής, αποκλίνει από τις συντεταγμένες που δίνονται από το πρόγραμμα. Επίσης η επίλυση της ευθείας κινηματικής προβάλλεται στην οθόνη (αν επιλεγεί το γραφικό περιβάλλον AXIS) με αποτέλεσμα να αποτελεί ένα καλό δείκτη κατά την προσπάθεια σταθεροποίησης του αλγορίθμου.

Για τις πράξεις μεταξύ πινάκων καθώς και για την μετετράπη του ομογενοποιημένου πίνακα σε συντεταγμένες και αντιστροφή, χρησιμοποιήθηκαν οι μαθηματικές βιβλιοθήκες `rtapi_math.h` και `rosemath.h`. Επίσης κατασκευάστηκε και η βιβλιοθήκη `rm501.h` (ΠΑΡΑΡΤΗΜΑ Β) η οποία στην ουσία φιλοξενεί τις ανοχές σφάλματος αναμεσά στην ευθεία και την αντιστροφή κινηματική για την μεταξύ τους επικοινωνία.

Για την επικοινωνία του αλγορίθμου με το τμήμα διαχείρισης τροχιάς (trajectory planner) θα πρέπει οι αρθρώσεις να αναφέρονται ως στοιχεία του πίνακα `joint[]` και οι συντεταγμένες του χώρου x, y, z ως `world -> a, world -> b, world -> c`, αντίστοιχα.

Τέλος, το αρχείο θα πρέπει να τροποποιηθεί (compile) έτσι ώστε να εκτελείται σαν εικονικό στοιχείο (HAL component) στον αρχείο HAL και να καλείται πριν από οποιοδήποτε άλλο στοιχείο έτσι ώστε να αποτελεί κομμάτι του διαχειριστή τροχιάς.

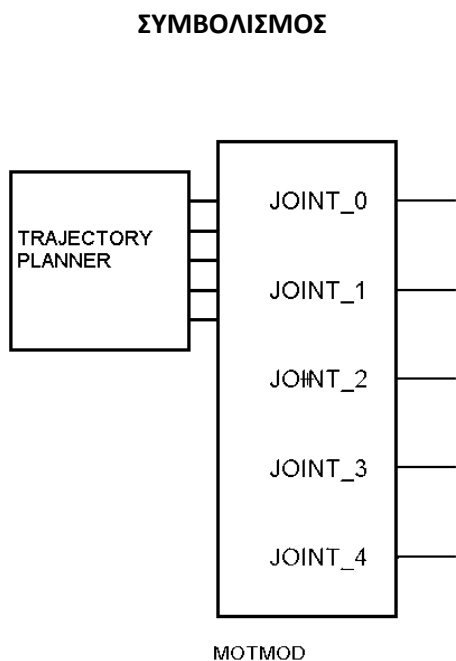
4.2.2. Το αρχείο HAL

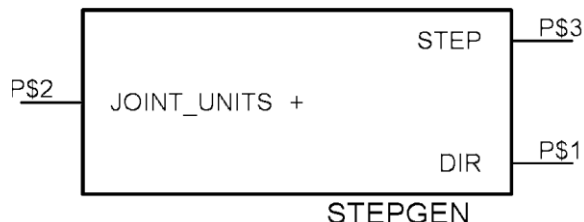
4.2.2.1. Εικονικά στοιχεία (HAL COMP)

Πριν γίνει η αναφορά στο αρχείο `rm501.hal` θα πρέπει να παραθέσουμε τα εικονικά στοιχεία (HAL components) που θα χρησιμοποιηθούν για την επικοινωνία του υπολογιστή με τις πλακέτες της μονάδας ελέγχου. Επειδή το αρχείο HAL έχει τους δικούς του συντακτικούς κανόνες και μια απλή εποπτεία ίσως είναι κουραστική, κάθε εικονικό στοιχείο συνοδεύεται από ένα σύμβολο σχεδιασμένο αυθαίρετα και χωρίς να περιλαμβάνει όλες τις παραμέτρους και τα εικονικά καρφιά (hal pins), παρά μόνο αυτά που είναι απαραίτητα για την κατανόηση της διάταξης της εικονικής πλακέτας HAL.

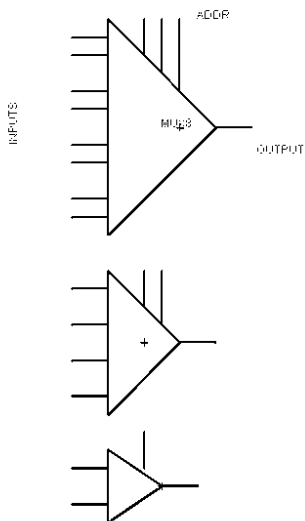
Παρακάτω αναφέρονται εν συντομία τα εικονικά στοιχεία που χρησιμοποιήθηκαν στο αρχείο `rm501.hal`. για μια πιο εκτενή περιγραφή κάθε στοιχείου ανατρέξτε στο εγχειρίδιο HAL.

MOTMOD

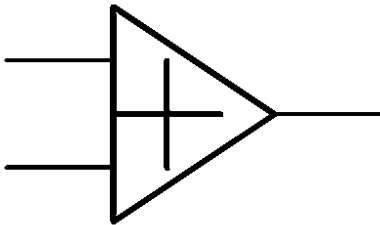


STEPGEN**ΣΥΜΒΟΛΙΣΜΟΣ****ΠΕΡΙΓΡΑΦΗ**

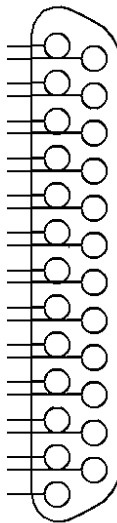
Προέρχεται από την συντομογραφία STEP GENerator. Είναι δηλαδή το εικονικό στοιχείο το οποίο δέχεται την πληροφορία της θέσης της άρθρωσης και παράγει τα βήματα (παλμούς) έτσι ώστε να σταλούν στον σερβοκινητήρα. Κατά τον συμβολισμό έγινε μια υπεραπλούστευση του στοιχείου μιας και έχει πολλές παραμέτρους οι οποίες πρέπει να ληφθούν υπόψιν. Παρολαυτα οι παράμετροι αυτοί μπορούν να ρυθμιστούν από το αρχείο INI συνεπώς δεν υπήρχε λογος αναφοράς τους. Για περισσότερες πληροφορίες καθώς και για όλους τους τύπους των stepgen μπορείτε να ανατρέξετε στο εγχειρίδιο HAL

MUX_GENERIC**ΣΥΜΒΟΛΙΣΜΟΣ****ΠΕΡΙΓΡΑΦΗ**

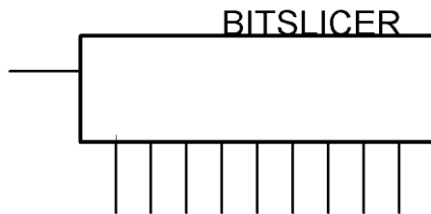
Προέρχεται από την συντομογραφία multiplexer generic. Είναι το εικονικό στοιχείο το οποίο δίνει σαν έξοδο την είσοδο εκείνη που υποδεικνύει η διεύθυνση (ADDR). Είναι μια γενικοποιημένη μορφή multiplexer, δηλαδή αντί να υπάρχει ξεχωριστό στοιχείο για 2 εισόδους, 4 εισόδους, 8 εισόδους, κλπ. υπάρχει ένα που θέτοντας κατάλληλα τις παραμέτρους μπορεί να αποτελέσει στοιχείο από 2 έως και 32 εισόδων. Η διεύθυνση δίνεται σε δυαδική μορφή έτσι για ένα multiplexer 2εισόδων χρειαζόμαστε 1bit οπού 0 θα είναι η ΕΙΣΟΔΟΣ_1 και 1 θα είναι η ΕΙΣΟΔΟΣ_2, ενώ για ένα multiplexer 8 εισόδων χρειαζόμαστε 3 bit έτσι αν θέλουμε να καλέσουμε την εισοδο-6 θα πρέπει να δοθεί διεύθυνση 101. Για το πώς καλείτε το στοιχείο καθώς και πως καθορίζονται οι παράμετροι, ανατρέξτε στο εγχειρίδιο HAL

SUM2**ΣΥΜΒΟΛΙΣΜΟΣ****ΠΕΡΙΓΡΑΦΗ**

Το εικονικό αυτό στοιχείο αποδίδει το άθροισμα 2 αριθμών. Υπάρχουν και άλλα παρόμοια στοιχεία (SUM4, SUM8) τα οποία αποδίδουν το άθροισμα των αριθμών των εισόδων που αναφέρεται στο όνομα τους. Αξίζει να σημειωθεί ότι υπάρχουν παράμετροι που επιτρέπουν τον πολλαπλασιασμό των εισόδων με κάποιο συντελεστή πριν την πρόσθεση τους. Έτσι αν πολλαπλασιάσουμε την ΕΙΣΟΔΟ_2 με -1 τότε το στοιχείο αποδίδει το υπόλοιπο 2 αριθμών!

PARPORT**ΣΥΜΒΟΛΙΣΜΟΣ****ΠΕΡΙΓΡΑΦΗ**

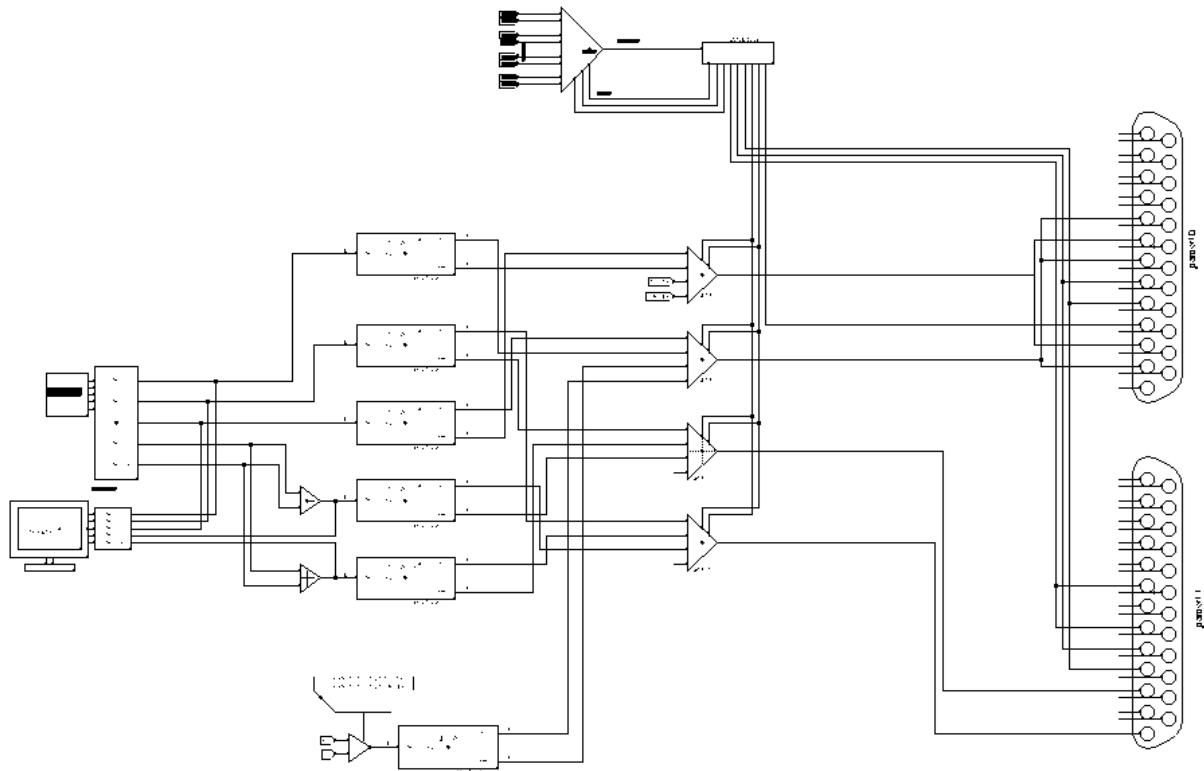
Προσέρχεται από την συντομογραφία PARallel PORT και είναι το εικονικό στοιχείο εκείνο που καλείται να στείλει ή να λάβει την πληροφορία από την παράλληλη θύρα. Έχει παραμέτρους που καθορίζει ποιες θύρες θα είναι εισοδοι (read) και ποιες έξοδοι (write), όπως ορίζει το πρωτόκολλο της εκάστοτε μητρικής. Επίσης το στοιχείο μπορεί να αντιστρέφει την τιμή για κάθε καρφή (pin), κάτι εξαιρετικά εργονομικό, διαφορετικά θα απαιτούνταν να καλέσουμε ένα στοιχείο NOT για κάθε καρφή (pin) που θα θάλαμε να αντιστρέψουμε.

BITSLICER**ΣΥΜΒΟΛΙΣΜΟΣ****ΠΕΡΙΓΡΑΦΗ**

Αυτό το εικονικό στοιχείο λαμβάνει σαν είσοδο έναν δεκαεξαδικό ή δεκαδικό αριθμό τον μετατρέπει σε δυαδική μορφή και αποδίδει τα bit τα οποία το διαμορφώνουν. Για παράδειγμα ο δεκαεξαδικός αριθμός x00015 μεταφράζεται σε 10101 συνεπώς το bitslicer θα μας δώσει στην ΕΞΟΔΟ_1= 1, ΕΞΟΔΟ_2=0, ΕΞΟΔΟ_3=1 κ.ο.κ. Το στοιχείο μπορεί να αποδώσει έως 32 bit σαν έξοδο.

4.2.2.2. Δομή του Αρχείου HAL

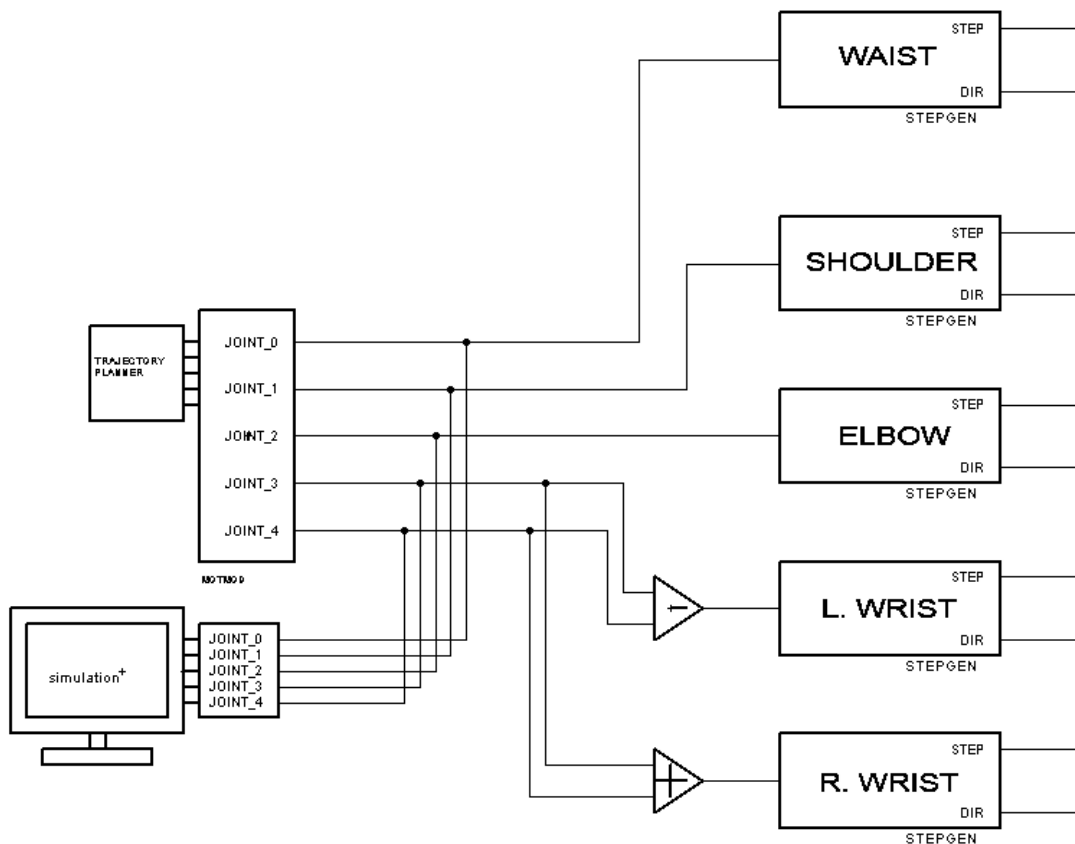
Έπειτα και από την παράθεση των εικονικών στοιχείων που κλήθηκαν για να διαμορφώσουν την εικονική πλακέτα, παρακάτω παρουσιάζεται αυθαίρετα η σχηματική αναπαράσταση της δομής του αρχείου HAL.



ΣΧΗΜΑ 4.4 Το κύκλωμα της εικονικής Πλακέτας HAL

Παρατηρώντας το σχήμα διαπιστώνουμε ότι έχουν χρησιμοποιηθεί δυο παράλληλες θύρες. Επιλεχτήκαν δυο παράλληλες θύρες γιατί δεν επαρκούσαν τα καρφιά εξόδου μιας μόνο θύρας. Επίσης παρατηρείται ότι, ανάμεσα στα sterpens και τις parport μεσολαβούν multiplexers. Τα multiplexers αναλαμβάνουν να δώσουν την πληροφορία των sterpens όλων των αρθρώσεων σε δυο μόνο καρφιά όπως απαιτούν οι πλακέτες της μονάδας διαχείρισης (drive unit). Βέβαια εδώ χρησιμοποιούνται 4 multiplexers δηλαδή 4 καρφιά (pins) εξόδου, αυτό έχει να κάνει με την ταχύτερη απόκριση, εκμεταλλευόμενοι τις δυο αυτόνομες πλακέτες της μονάδας διαχείρισης. Αναλυτικότερα θα αναφερθούμε παρακάτω όταν θα κάνουμε τμηματικά την εποπτεία του παραπάνω εικονικού κυκλώματος. Η συνδεσμολογία που παρουσιάζεται παραπάνω είναι πιστή τις συνδεσμολογίας που αναφέρεται στο αρχείο HAL και καθορίζεται από την αρχιτεκτονική διάταξη της μονάδας διαχείρισης.

Στην προσπάθεια να γίνει πλήρως αντιληπτή η παραπάνω διάταξη θα επιχειρήσουμε να παρουσιάσουμε τμηματικά το κύκλωμα



ΣΧΗΜΑ 4.5 Το τμήμα του κυκλώματος που αφορά την προσομοίωση και την μετατροπή αριθμητικών μεγεθών σε σήματα

Το σχήμα αναφέρεται στο τμήμα όπου τα δεδομένα έχουν τροποποιηθεί στο σχεδιαστή τροχιάς μέσω του αρχείου `rm501.c` και τροποποιούνται σε βήματα μέσω των `stepgens`. Στην ουσία τα δεδομένα αυτά είναι οι μοίρες για κάθε άρθρωση, έτσι μπορούμε να έχουμε μια ειοπτεία των δεδομένων μέσω μιας απλής προσομοίωσης. Για την προσομοίωση χρησιμοποιήθηκε ένα ελεύθερο πρόγραμμα σχεδιασμού το `vismach` και αποτέλεσε εξαιρετικό εποπτικό μέσο για την ρύθμιση και τον έλεγχο της κινηματικής θεωρίας.

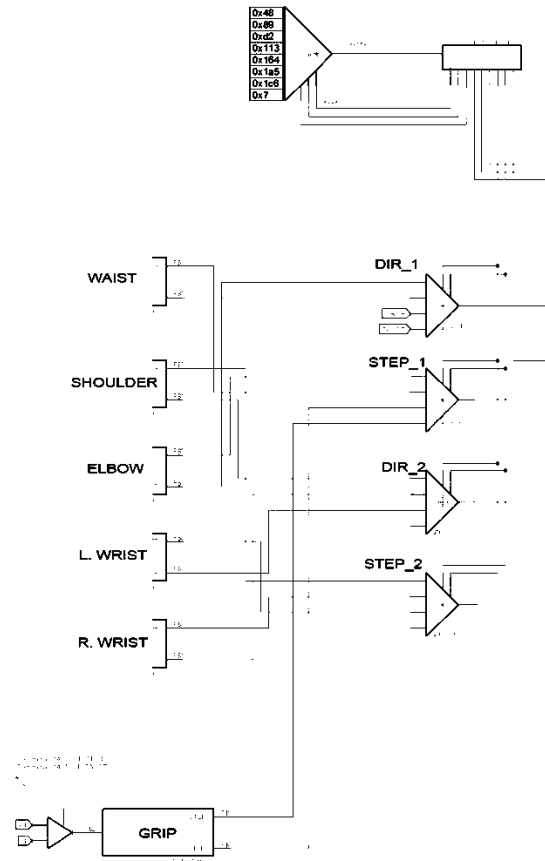
Επίσης παρατηρούμε ότι οι γωνίες των αρθρώσεων `joint_3` και `joint_4` περνούν από στοιχεία αθροίσματος `sum2` πριν καταλήξουν στα `stepgen`. Αυτό συμβαίνει γιατί οι δυο κινητήρες του καρπού

έχουν διαφορεική διάταξη ένα από τα sum2 προσθέτει το αρνητική γωνία της άρθρωσης που αναφέρεται για την περιστροφή έτσι ώστε να δώσει την αντίθετη φορά αλλά με τις ίδιες μοίρες στον κινητήρα. Ας δώσουμε ένα μικρό παράδειγμα: Έστω ότι θέλουμε να ο καρπός να περιστραφεί (roll) 30° ωρολογιακά, τότε το joint_4 του σχήματος θα έχει την τιμή 30 και το joint_3 την τιμή 0. Συνεπώς στον κινητήρα του αριστερού καρπού θα δοθεί $0+(-30) = -30^\circ$ και στον κινητήρα του δεξιού καρπού $0+30 = 30^\circ$ έτσι ο καρπός θα περιστρέφει 30° ωρολογιακά. Έστω τώρα ότι θέλουμε ο καρπός να μετακινηθεί 30° (pitch) τότε joint_3 = 30 , joint_4=0. Συνεπώς στον κινητήρα του αριστερού καρπού θα δοθεί $30+(-0) = 30^\circ$ και στον κινητήρα του δεξιού καρπού $30+0 = 30^\circ$, έτσι και οι δυο κινητήρες θα στραφούν με 30° προς την ίδια κατεύθυνση. Πρέπει να σημειωθεί ότι αυτό το τέχνασμα προϋποθέτει ότι δεν υπάρχει ταυτόχρονη μεταφορά και περιστροφή του καρπού συνεπώς θα πρέπει να έχει προβλεφθεί στην κινηματική να εκτελούνται σειριακά.

Τέλος αυτό το τμήμα αποδίδει τα βήματα και τις κατευθύνσεις που πρέπει να κάνει κάθε κινητήρας με βάση τις τιμές που του δόθηκαν.

Αφού παράξουμε τα βήματα του κάθε κινητήρα καλούμαστε να βρούμε λύση σε ένα περιορισμό που έχει τεθεί από την αρχιτεκτονική των πλακετών της μονάδας διαχείρισης. Ο περιορισμός είναι ότι τα βήματα όλων των κινητήρων δίνονται για κάθε πλακέτα στο pin01 και η κατεύθυνση στο pin02, ενώ για κάθε έναν από τους 5 κινητήρες υπάρχει ένα pin που δηλώνει σε ποιον αναφερόμαστε.

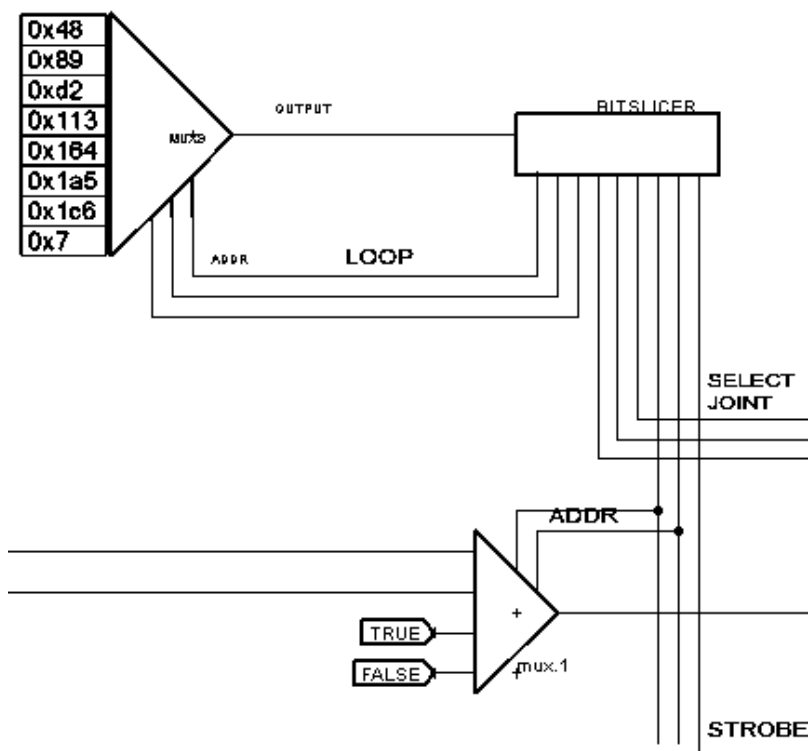
Συνεπώς το αρχείο hal θα πρέπει να αποδίδει βήμα και κατεύθυνση του κινητήρα που είναι ενεργοποιημένος ,δηλαδή που έχει ενεργό το pin. Στην περίπτωση του RM-501 τα select pins είναι ενεργά με λογική τιμή 0 (active low).



ΣΧΗΜΑ 4.6 Το τμήμα του κυκλώματος που αφορά την προσομοίωση και την μετατροπή αριθμητικών μεγεθών σε σήματα

Για να αντιμετωπίσουμε αυτόν τον περιορισμό χρησιμοποιήσαμε multiplexers έτσι κάθε ένα από αυτά θα επιλεγεί μια είσοδο ως έξοδο ανάλογα με την διεύθυνση που του δίνεται. Όμως Με την επιλογή των multiplexers δημιουργήθηκε ένα μεγάλο ερώτημα. Πως καλείται η σωστή διεύθυνση και πως επιλέγεται το σωστό pin για κάποιον κινητήρα και μάλιστα σύγχρονα;

Η απάντηση βρέθηκε με ένα τέχνασμα που προτάθηκε από του forum της κοινότητας του LinuxCNC και αυτό ήταν να δημιουργηθεί ένα αυτοτροφοδοτούμενο multiplexer το οποίο θα στέλνει την όποια πληροφορία με την βοήθεια ενός bitslicer.



ΣΧΗΜΑ 4.7 Το τέχνασμα του αυτοτροφοδοτούμενου multiplexer

Όπως φαίνεται και στο σχήμα σε κάθε μια από τις εισόδους του αυτοτροφοδοτούμενου multiplexer έχει αποθηκευτεί ένας δεκαεξαδικός HEX αριθμός τέτοιος ώστε το πρώτο bit να είναι η πληροφορία για το καρφί γράψε (strobe) τα δυο επόμενα bit να μας δίνουν την πληροφορία σχετικά με την διεύθυνση των multiplexers τα τρία επόμενα bit την επιλογή ενός κινητήρα και τέλος τα τρία τελευταία bit την διεύθυνση η οποία θα δώσει ως έξοδο τον επόμενο δεκαεξαδικό αριθμό.

Η ανατροφοδότηση του multiplexer δείχνει να μην έχει λογική. Όμως η πλακέτα hal είναι εικονική, στην ουσία τα στοιχεία είναι υπορουτίνες που εκτελούνται επαναλημμένα ανάλογα με το thread που την έχει κατατάξει ο προγραμματιστής (βλέπε εγχειρίδιο HAL). Έτσι η υπορουτίνα ενός multiplexer καλεί να δώσει τις τιμή εξόδου, αυτής της χρονικής περιόδου ,με βάση την διεύθυνση που δόθηκε την προηγούμενη χρονική περίοδο. (Ως χρονική περίοδος ορίζονται τα νανοδευτερόλεπτα που απαιτούνται για να εκτελεστεί η υπορουτίνα και καθορίζεται κυρίως από τις δυνατότητες του επεξεργαστή).

Η επιλογή του πίνακα των δεκαεξαδικών αριθμών προέκυψε με βάση την απαιτούμενη πληροφορία Όπως αναφέρθηκε στο κεφάλαιο II η λειτουργία του pin05 αχρηστεύθηκε γιατί δεν μας ικανοποιούσε, συνεπώς η εντολή γράψε που λαμβάνει το pin04 μπορεί να είναι ένα μόνιμο τετραγωνικό σήμα η αλλιώς να είναι εναλλαγή 0 και 1 στο πρώτο bit των δεκαεξαδικών αριθμών. Τα multiplexers θα έχουν διευθύνσεις έτσι ώστε να περνά μια από τις τέσσερεις εισόδους στην έξοδο για κάθε δεκαεξαδικό

αριθμό. Η επιλογή κινητήρα θα γίνεται έτσι ώστε όλοι οι άλλοι κινητήρες που βρίσκονται στην ίδια πλακέτα να είναι ανενεργοί. Και τέλος τα τρία bit που είναι υπεύθυνα για την ανατροφοδότηση θα πρέπει να έχουν την διεύθυνση της επομένης εσόδου. Ως κύκλος σάρωσης ορίζεται μια ολοκληρωμένη σάρωση, όταν δηλαδή στο αυτοτροφοδοτούμενο mux δοθεί η διεύθυνση 000

Παρακάτω παρουσιάζεται ο πίνακας που καθορίζει την επιλογή των δεκαεξαδικών αριθμών

HEX No	MUX LOOP	JOINT SELECT	ADDR MUX4	STROBE
0x48	001	001	00	0
0x89	010	001	00	1
0xD2	011	010	01	0
0x113	100	010	01	1
0x164	101	100	10	0
0x1A5	110	100	10	1
0xC6	111	000	11	0
0x7	000	000	11	1

Σαν μια πρώτη λογική ήταν να χρησιμοποιήσουμε ένα multiplexer για το βηματισμό και ένα για την κατεύθυνση και για τις 3 πλακέτες της μονάδας διαχείρισης έτσι με 6 κινητήρες χρειαζόμασταν δυο multiplexers 6 εσόδων και 1 παράλληλη θύρα. Προτιμήθηκε όμως να χρησιμοποιηθεί ένα ζεύγος multiplexers για κάθε πλακέτα. Αυτό έγινε γιατί το αυτοτροφοδοτούμενο multiplexer μοιράζει την ίδια πληροφορία σε δυο πανομοιότυπες πλακέτες με αποτέλεσμα να μειώνεται στο μισό ο κύκλος σάρωσης.

Επιστέφοντας στο σχήμα παρατηρούμε ότι ο sterpen για την αρπάγη έχει μια ιδιαιτερότητα αυτό συμβαίνει γιατί η πλακέτα που φιλοξενεί το κύκλωμα της αρπάγης είναι ελαφρώς διαφοροποιημένο έχει δυο καρφιά διάλεξε. Η συνδεσμολογία δείχνει καθώς ανορθόδοξη αλλά μιας και η πλακέτα 724 έχει μόνο δυο κινητήρες υπήρχε η δυνατότητα να χρησιμοποιήσουμε τις άλλες δυο κενές εισόδους του multiplexer για τον συντονισμό της αρπάγης χωρίς να καλέσουμε περεταίρω εικονικά στοιχεία. Επίσης το sterpen της αρπάγης ενεργοποιείτε κατευθείαν από το περιβάλλον του χρήστη μέσω εντολών M. με την εντολή M101 η αρπάγη κλείνει και με την εντολή M102 ανοίγει.

Τέλος κάτι που δεν παρουσιάζεται σχηματικά αλλά αναφέρεται στο αρχείο HAL είναι οι πληροφορίες που παίρνει το πρόγραμμα από τον βραχίονα (inputs). Οι πληροφορίες αυτές είναι οι διακοπές οριακών θέσεων και οι υπερχειλιστές.

Οι διακοπές οριακών θέσεων συνδέονται με εικονικό pin που έχει προβλεφθεί από τον προγραμματιστές του LinuxCNC ενώ οι υπερχειλιστές συγκρίνονται με ένα εικονικό στοιχείο OR η έξοδος του οποίου συνδέεται με τον επανακαθοριστή των πλακετών (Master Reset).

4.2.3. Το αρχείο INI

Το αρχείο `rm501.ini` είναι σαφώς πιο ευανάγνωστο από τα άλλα δυο αρχεία. Όπως όλα τα αρχεία INI συνάσσεται για τον καθορισμό και την αρχικοποίηση κάποιων παραμέτρων. Έχει και αυτό τους δικούς του συντακτικούς κανόνες και αποτελεί μαζί με το αρχείο HAL την παρέμβαση του συντονιστή για την προσαρμογή του προγράμματος LinuxCNC στα δεδομένα τις κάθε μηχανής. Το εγχειρίδιο INI είναι αρκετά αναλυτικό και παρέχει την όποια πληροφορία χρειάζεται ο συντονιστής.

Στην παρούσα υποενότητα θα αναφερθούμε στην επιλογή των τιμών των παραμέτρων και όχι στην σύνταξη του αρχείου.

Οι πρώτες παράμετροι που μας προβληματίζουν είναι η επιλογή νανοδευτερολέπτων για της μεταβλητές `BASE_PERIOD`, `SERVO_PERIOD` και `TRAJ_PERIOD`. Στην ουσία οι τιμές αυτές δηλώνουν ανα πόσα νανοδευτερόλεπτα θα εκτελούνται οι υπορουτίνες που έχουν να κάνουν με τα τμήματα `base_thread`, `servo_thread` και τις υπορουτίνες του διαχειριστή τροχιάς αντίστοιχα. Προφανώς θα θέλαμε τις μικρότερες δυνατές τιμές όσον αφορά κυρίως το `BASE_PERIOD`. Όμως κάθε υπολογιστής έχει τις δικές του υπολογιστικές δυνατότητες. Για το λόγο αυτό η κοινότητα του LinuxCNC διαθέτει ένα πρόγραμμα που υπολογίζει τη μέγιστη καθυστέρηση που μπορεί να έχει ο υπολογιστής σε νανοδευτερόλεπτα (`max jitter`) έτσι ώστε να τεθεί η τιμή του `BASE_PERIOD` λίγο πιο πάνω από αυτό το όριο.

Στην περίπτωση του υπολογιστή που συνοδεύει τον ρομποτικό βραχίονα RM-501 η τιμή του `base_period` δεν μπορούσε να τεθεί κάτω από 50000 ns με το λειτουργικό σύστημα Ubuntu 10.04 για το λόγο αυτό και μετά από μια όχι και τόσο βατή διαδικασία το πρόγραμμα LinuxCNC μεταφορτώθηκε σε λειτουργικό σύστημα Ubuntu 12.04 όπου μας επέτρεψε να θέσουμε στην μεταβλητή `BASE_PERIOD` την τιμή 40000 ns. Οι άλλες δυο τιμές εκτιμώνται βάση της πρώτης και με δοκιμές καταλέξαμε στις παρακάτω τιμές

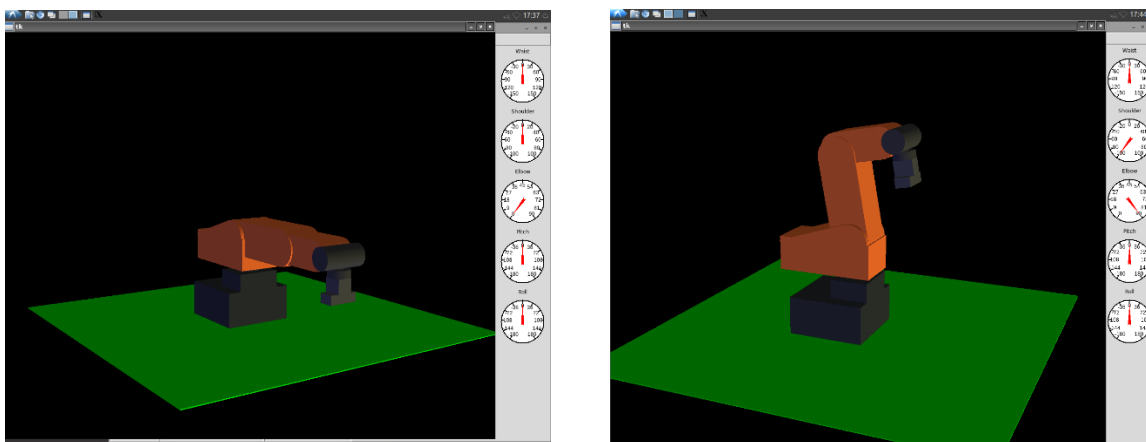
```
BASE_PERIOD      =      40000
SERVO_PERIOD     =     1000000
TRAJ_PERIOD      =     10000000
```

Κάθε μια άρθρωση παραμετροποιείται ξεχωριστά στο αρχείο με μια σειρά από μεταβλητές όπως οριακές θεάσεις μέγιστη ταχύτητα και πολλά ακόμη που βοηθούν στην προσαρμογή του προγράμματος σε διάφορους τύπους εργαλειομηχανών. Παρακάτω παρατίθενται οι μεταβλητές που χρειάστηκαν για την ρύθμιση κάθε μιας άρθρωσης του ρομποτικού βραχίονα.

- **TYPE**: καθορίζει τον τύπο της άρθρωσης αν θα περιστροφική ή επιμήκης (στην περίπτωση μας όλες οι αρθρώσεις είναι περιστροφικές)

- MAX_VELOCITY: είναι η μέγιστη δυνατή ταχύτητα μονάδες μηχανής ανα δευτερόλεπτο. Η μέγιστη ταχύτητα καθορίζεται κυρίως από την δυνατότητα παραγωγής βήματος συνεπώς έχει σχέση με την τιμή του BASE_PERIOD και το μήκος βήματος. Σε περίπτωση που τεθεί μεγαλύτερη τιμή από την δυνατή το πρόγραμμα θέτει την μέγιστη δυνατή αυτόματα.
- MAX_ACCELERATION: η μέγιστη δυνατή επιτάχυνση
- BACKLASH: είναι μια μεταβλητή η όποια σκοπό έχει να εξομαλύνει τυχόν διαφορές που παρουσιάζονται κατά την διαχείριση τις εργαλειομηχανής από το πρόγραμμα
- MIN_LIMIT: η κάτω οριακή τιμή που μπορεί να λάβει η άρθρωση (soft limit), ποια είναι αυτή καθορίζεται από ποια ορίσαμε ως θέση μηδέν (0,0,0,0,0) κατά την κινηματική θεωρία
- MAX_LIMIT: η άνω οριακή τιμή που μπορεί να λάβει η άρθρωση (soft limit)

Το να έρθει μια εργαλειομηχανή στην αρχική της θέση καμιά φορά είναι πιο δύσκολο από ότι δείχνει. Στην περίπτωση του ρομποτικού βραχίονα RM 501 η αρχική θέση δεν συμπίπτει με την θέση μηδέν όπως φαίνεται και στο σχήμα 3.2.5



ΣΧΗΜΑ 4.8 Αριστερά η θέση μηδέν και δεξιά η αρχική θέση

Ως αρχική θέση (HOME) ορίζεται η θέση που έρχεται ο βραχίονας πριν ξεκινήσει να εκτελεί οποιαδήποτε εντολή. Η θέση αυτή μπορεί να είναι οποιαδήποτε αλλά συνήθως επιλέγεται μια θέση όπου ο ρομποτικός βραχίονας δεν έχει εκτεθειμένα μέλη. Θέτοντας την εργαλειομηχανή σε λειτουργία προτού εκτελέσει οποιαδήποτε εντολή G κώδικα πρέπει να έρθει στην αρχική θέση, η διαδικασία αυτή ονομάζεται αρχικοποίηση (homing). Για την ακριβή οριοθέτηση της αρχικής θέσης μιας εργαλειομηχανής υπάρχουν μια σειρά από μεταβλητές όπως παρουσιάζονται παρακάτω

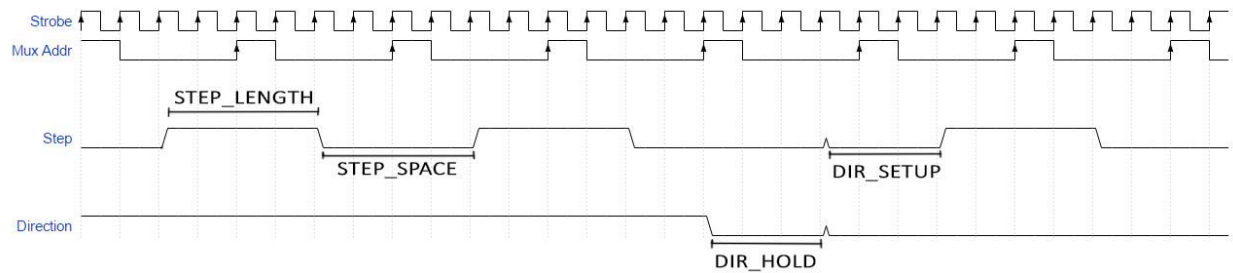
- HOME: η θέση στην οποία θα βρίσκεται η άρθρωση αφού ολοκληρωθεί η διαδικασία αρχικοποίησης

- HOME SEQUENCE: είναι η σειρά με την οποία θα αρχικοποιηθούν όλες οι αρθρώσεις. Αν δηλαδή η μεταβλητή αυτή είναι 0 για την άρθρωση του Ωμου και 1 για την Μέση, τότε θα φέρει πρώτα σε αρχική θέση τον Ωμο και έπειτα τη Μέση
- HOME SEARCH VEL: η ταχύτητα με την οποία κατά την αρχικοποίηση η εργαλειομηχανή προσεγγίζει τον διακόπτη οριακής θέσης
- HOME LATCH VEL: η ταχύτητα με την οποία αποδεσμεύεται ο διακόπτης αρχικοποίησης
- HOME IGNORE LIMITS: στην περίπτωση που ο διακόπτης αρχικοποίησης είναι ίδιος με τον οριακό διακόπτη τότε τίθεται η τιμή YES έτσι ώστε να αγνοηθεί ο τερματικός διακόπτης κατά την διαδικασία αρχικοποίησης
- HOME OFFSET: η θέση στην οποία βρίσκεται ο διακοπής αρχικοποίησης (home switch)
- HOME IS SHARED: δέχεται τιμές στην περίπτωση που ένας διακόπτης εξυπηρετεί δυο ή περισσότερες αρθρώσεις προς αρχικοποίηση

Επίσης μπορούν να συνταχθούν μεταβλητές έτσι ώστε να μεταβάλουν τιμές σε παραμέτρους εικονικών στοιχείων HAL. Στην περίπτωση του ρομποτικού βραχίονα RM-501 λόγω της ιδιομορφίας που παρουσίαζαν οι πλακέτες επιβάλλεται να μεταβληθούν οι παράμετροι που αφορούν τον βηματισμό (stepping configuration). Για τον λόγο αυτό δημιουργήθηκαν οι παρακάτω μεταβλητές οι οποίες αναφέρονταν στις αντίστοιχες παραμέτρους του στοιχείου stepgen για κάθε άρθρωση:

- SCALE: η ολική σχέση μετάδοσης
- STEPGEN MAXVEL: η μέγιστη ταχύτητα παραγωγής βήματος
- STEPGEN MAXACCEL: η μέγιστη επιτάχυνση παραγωγής βήματος
- STEP LENGTH: το μήκος βήματος σε νανοδευτερόλεπτα
- STEP SPACE: η απόσταση μεταξύ δυο βημάτων σε νανοδευτερόλεπτα
- DIR HOLD: ο χρόνος αναμονής κατά την αλλαγή κατεύθυνσης έως ότου ξεκινήσει ο χρόνος καθορισμού του πρώτου βήματος με την νέα κατεύθυνση
- DIR SETUP: ο χρόνος καθορισμού του πρώτου βήματος με την νέα κατεύθυνση

Οι μεταβλητές βηματισμού επιλέχτηκαν έτσι ώστε να μην χαθεί κάποιο βήμα (miss-step). Στο σχήμα παρουσιάζεται η μέθοδος που χρησιμοποιήθηκε για την σωστή επιλογή των τιμών των παραπάνω μεταβλητών.



ΣΧΗΜΑ 4.9 Οι κυματομορφές κατά την λειτουργία ενός κινητήρα

Όπως αναφερθήκαμε στην υποενότητα 3.2.2 του κεφαλαίου 3 για να δεχθεί ένα βήμα ο κινητήρας κάποιας άρθρωσης θα πρέπει κατά την μετάβαση από Low σε High του strobe και την μετάβαση από Low σε High του multiplexer για την συγκεκριμένη διεύθυνση να έχει υπάρξει μεταβολή του βήματος. Όμως το strobe έχει περίοδο $2 \times \text{BASE_PERIOD}$ και το mux έχει $4 \times \text{BASE_PERIOD}$. Συνεπώς για να «βλέπει» πάντα μεταβολή ο κινητήρας θα πρέπει τα STEP_LENGTH και STEP_SPACE να είναι $4 \times \text{BASE_PERIOD}$ και τα DIR_SETUP και DIR_HOLD $3 \times \text{BASE_PERIOD}$.

ΚΕΦΑΛΑΙΟ 5. Αποτελέσματα και Συμπεράσματα

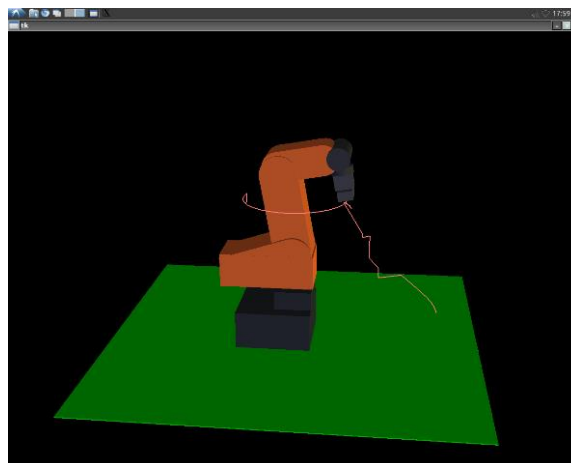
5.1. Διαδικασία δοκιμών

Μετά και την ολοκλήρωση την διαμόρφωσης του λογισμικού κάνοντας μια σύντομη ανασκόπηση της διαδικασίας παρατηρούμε ότι

- Έχει καθοριστεί η Ευθεία και αντίστροφη κινηματική του βραχίονα
- Έχει καθοριστεί ο όγκος εργασίας όπως αυτός προκύπτει με αποληκτήριο εργαλείο την αρπάγη
- Αποσαφηνίστηκε η αρχή λειτουργίας της μονάδας διαχείρισης
- Τροποποιήθηκε κατάλληλα η μονάδα διαχείρισης για την επικοινωνία της με σύγχρονο ηλεκτρονικό υπολογιστή
- Διαμορφώθηκε κατάλληλα το λογισμικό για την ορθή επικοινωνία του με τον βραχίονα

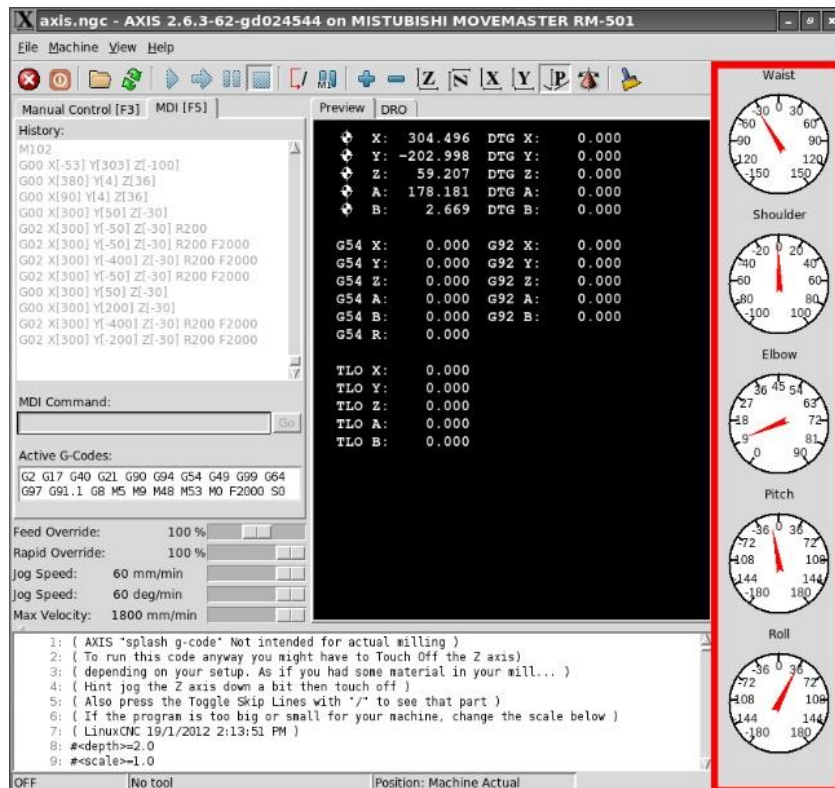
Κατά συνέπεια, πρέπει να ελεγχθεί κατά πόσο ο βραχίονας δέχεται σωστά τα 'ερεθίσματα' από τον υπολογιστή και το αντίστροφο.

Για την διεξαγωγή δοκιμών χρησιμοποιήθηκε καταρχάς προσομοίωση του ρομπότ με την βοήθεια εικονικού 3D γραφήματος (βλ. σχήμα 5.1) Με αυτό τον τρόπο εξετάζονται τυχόν ατέλειες της κινηματικής, τερματικές θέσεις και γενικά δοκιμές που είναι προτιμότερο να μην γίνονται απευθείας στον βραχίονα (off machine tests). Το πρόγραμμα που χρησιμοποιήθηκε για την σχεδίαση ήταν το VisMach ένα δωρεάν ανοιχτό λογισμικό τρισδιάστατης σχεδίασης. Αποθηκεύτηκε σαν αρχείο rpython και τροποποιήθηκε αναλόγως ώστε να έχει επικοινωνία με το αρχείο HAL. Εδώ πρέπει να σημειωθεί ότι το μεγαλύτερο μέρος του σχεδιασμού και διαμόρφωσης του 3D γραφήματος έχει γίνει από τον Jakob Flierl και χρησιμοποιήθηκε με την συγκατάθεση του. Στο σχήμα 5.1 παρατηρούμε ότι κατά την κίνηση του βραχίονα διαγράφεται το μονοπάτι που ακολουθεί. Στην περίπτωση του σχήματος το ίχνος αναφέρεται στην διαδικασία εύρεσης της αρχικής θέσης από την θέση μηδέν.



ΣΧΗΜΑ 5.1 3D Γράφημα για την Προσομοίωση του Ρομποτικού βραχίονα

Στο γραφικό περιβάλλον AXIS προστέθηκαν δείκτες (βλ. Σχ. 5.2) έτσι ώστε να φαίνονται οι γωνίες των αρθρώσεων και πως αυτές μεταβάλλονται για κάθε απαίτηση νέας θέσης. Αυτό πραγματοποιήθηκε με την δημιουργία ενός αρχείου xhtml και με καθοδήγηση από το εγχειρίδιο διαμόρφωσης (integrator manual). Η τροποποίηση αυτή βοηθά στην εκτίμηση σφαλμάτων θέσης.



ΣΧΗΜΑ 5.2 Οι δείκτες γωνιών για κάθε άρθρωση

Τέλος, χρησιμοποιήθηκε ο ίδιος ο ρομποτικός βραχίονας. Κυρίως έγιναν δοκιμές για την διαδικασία εύρεσης αρχικής θέσης όπως επίσης για την εύρεση μέγιστης ταχύτητας και κατά ποσό συμβαδίζει με το 3D γράφημα

Για τις δοκιμές επιλέχθηκαν τρία σενάρια :

Σενάριο 1: ευθύγραμμη κίνηση με μέγιστη ταχύτητα από την αρχική θέση στην θέση μηδέν

Σενάριο 2: ευθύγραμμη κίνηση με πρόωση 2000 από την θέση μηδέν σε θέση ελάχιστης επιδεξιότητας

Σενάριο 3: καμπυλόγραμμη κίνηση με πρόωση 2000 με σταθερές αρθρώσεις Όμου, Αγκώνα και Καρπού

Τα σενάρια επιλέχθηκαν για να παρουσιάσουν κάποιες ιδιομορφίες και να παροτρύνουν για περισσότερη έρευνα επί του βραχίονα

Για λόγους πληρότητας δόθηκε ένα δείγμα G κώδικα που αφορά την συλλογή και τοποθέτηση ενός τεμαχίου από μια θέση μέγιστης επιδεξιότητας σε μια άλλη

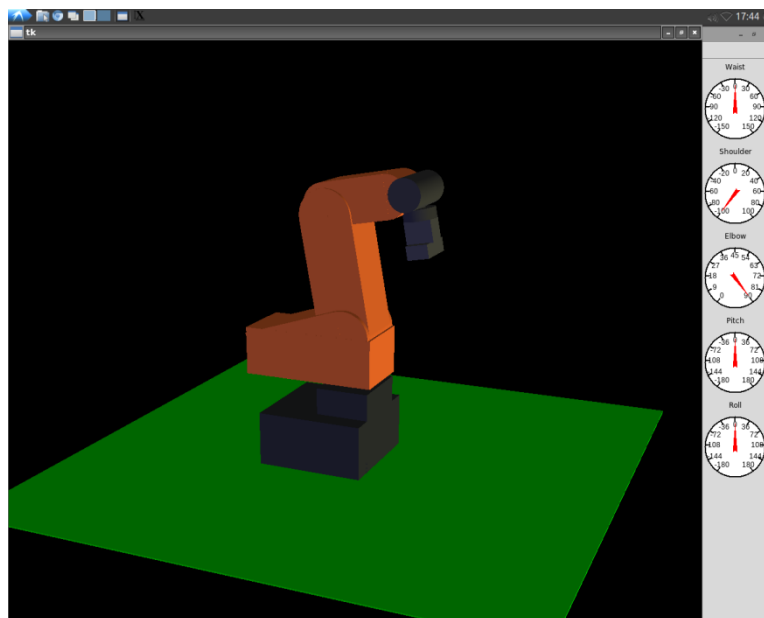
5.2. Αποτελέσματα

Σενάριο 1: οι συντεταγμένες του τελικού στοιχείου στην αρχική θέση προκύπτουν από την επίλυση της ευθείας κινηματικής για τις δεδομένες γωνίες των αρθρώσεων και είναι [X:82,10 Y:0 Z:-205,78]. Οι συντεταγμένες του τελικού στοιχείου στην θέση μηδέν προκύπτουν από την επίλυση της ευθείας κινηματικής για γωνίες όλων των αρθρώσεων ίσες με μηδέν και είναι [X:380 Y:0 Z:-35,4]

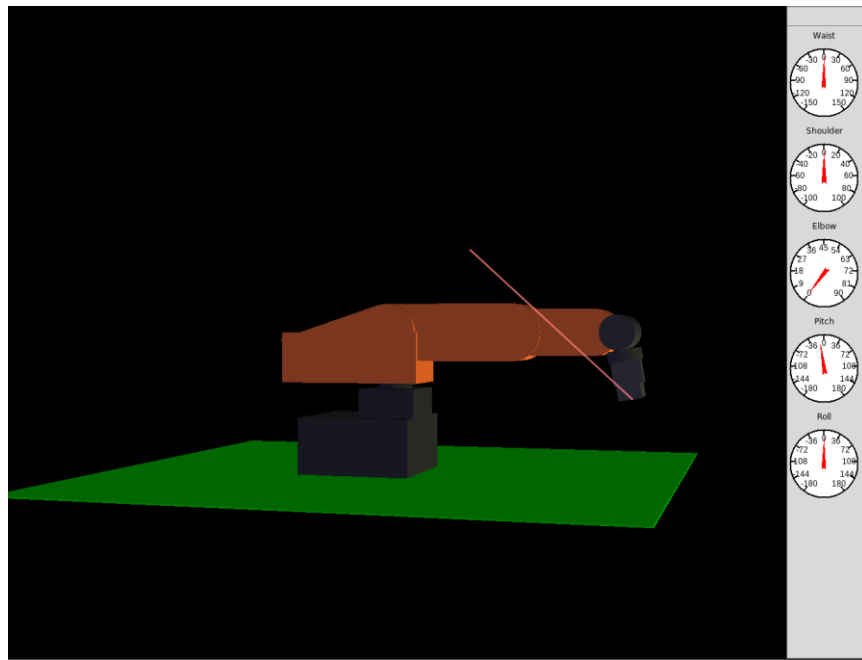
Συνεπώς για ευθύγραμμη κίνηση με μέγιστη ταχύτητα από την αρχική θέση στην θέση μηδέν η εντολή κώδικα G θα είναι

G00 X[380] Y[0] Z[-35,4]

Όπως παρατηρούμε από τα σχήματα 5.3 και 5.4 η προσομοίωση εκτέλεσε με επιτυχία την εντολή για ευθύγραμμη απαίτηση. Όμως η άρθρωση του καρπού κατά την ανύψωση δεν τερμάτισε στο ακριβές μηδέν. Αυτό εκτιμάται ότι οφείλεται κυρίως σε σφάλμα διάχυσης κατά την εκτέλεση της κινηματικής.



ΣΧΗΜΑ 5.3 Η αρχική θέση πριν την εντολή G00 του σεναρίου 1



ΣΧΗΜΑ 5.4 Η θέση μετά την εκτέλεση της εντολής G00 του σεναρίου 1

Στον πραγματικό χώρο ο βραχίονας ακολούθησε πιστά την πορεία που προβάλλεται στην προσομοίωση. Στο σχήμα 5.5 γίνεται πιο εμφανής η αδυναμία του καρπού να έρθει στην θέση μηδέν για την ανύψωση.



ΣΧΗΜΑ 5.5 Η θέση στον πραγματικό χώρο μετά την εντολή G00 του σεναρίου 1

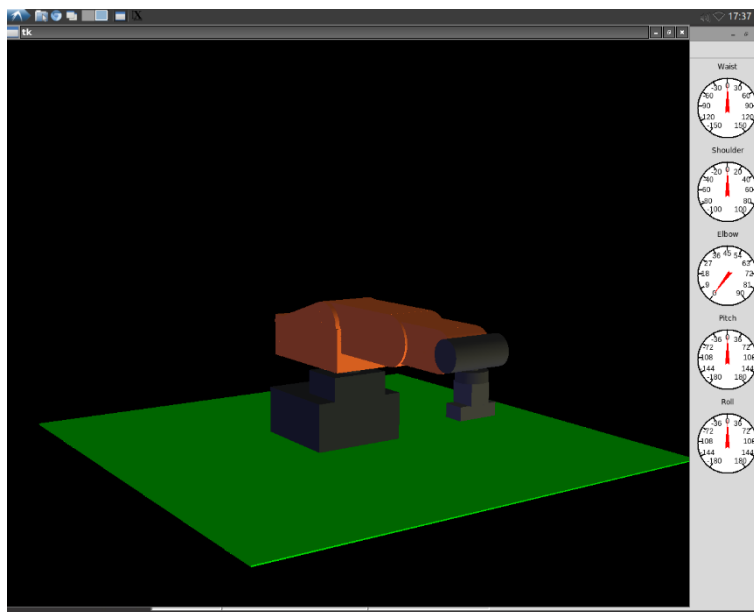
Σενάριο 2: Στο δεύτερο σενάριο επιλέχτηκε μια θέση στην οποία ο βραχίονας θα τερματίσει κάποια άρθρωση έτσι ώστε να παρατηρηθεί η συμπεριφορά του, πχ η ύπαρξη εμφανούς κραδασμού (jogging) και συνακόλουθη επίδραση στην ακρίβεια του τελικού στοιχείου. Η θέση που επιλέχτηκε είναι [X: 90 Y:0 Z:-35,4] έτσι ώστε να «πέφτει» πάνω στον άξονα Y για οπτικούς λόγους και να τερματίζει η άρθρωση του Αγκώνα.

Οι εντολές σε G κώδικα για την κίνηση από την αρχική θέση στην θέση μηδέν και έπειτα στο σημείο (90, 0, -35,4) είναι

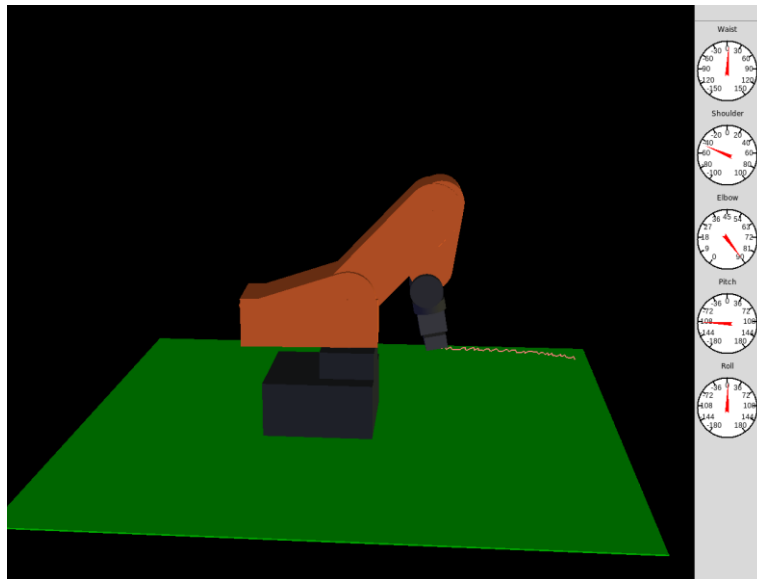
```
G00 X[380] Y[0] Z[-35,4]
```

```
G01 X[90] Y[0] Z[-35,4] F[2000]
```

Κατά την προσομοίωση αγνοήσαμε τα όρια των αρθρώσεων ώστε να υπάρξει μεγαλύτερη διαδρομή με πιο εμφανή αποτελέσματα. Στην προσομοίωση ο βραχίονας εκτέλεσε ευθεία πορεία αλλά όχι τόσο ομαλή. Στο σημείο αυτό πρέπει να θυμίσουμε ότι ο έλεγχος της ανάδρασης γίνεται από τους σερβοενισχυτές και όχι από το πρόγραμμα LinuxCNC. Έτσι δεν έχουμε πληροφορία της θέσης του βραχίονα στον πραγματικό χώρο. Για το λόγο αυτό η ανωμαλίες που παρουσιάζονται κατά την προσομοίωση της διαδρομής του τελικού στοιχείου στο παρόν σενάριο δεν οφείλονται στα προάγοντες του πραγματικού συστήματος (πχ. επενεργητές, τερματικοί θέσης κλπ.).



ΣΧΗΜΑ 5.6 Η θέση πριν την εντολή G01 του σεναρίου 2



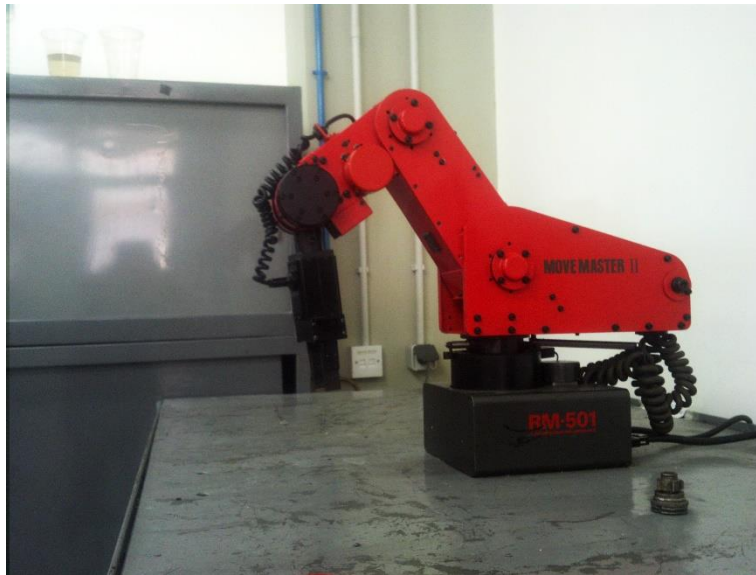
ΣΧΗΜΑ 5.6 Η θέση μετά την εντολή G01 και η διαδρομή του τελικού στοιχείου στο σενάριο 2

Παρουσιάζονται κυματισμοί κατά την πορεία του βραχίονα από την μια θέση στην άλλη, λόγω της ιδιομορφίας της αρχιτεκτονικής σε συνδυασμό με τις κακές επιδόσεις του επεξεργαστή του ΗΥ. Από το κεφάλαιο 4 γνωρίζουμε ότι χρειαζόμαστε 4x BASE_PERIOD για να «ξαναδει» βήμα κάποιος κινητήρας. Ο επεξεργαστής του υπολογιστή που ελέγχει τον βραχίονα μπορεί να δώσει οριακά BASE_PERIOD = 40000ns αρκετά μεγάλο για να παρουσιάζει ανωμαλίες η ευθύγραμμη πορεία. Βέβαια εδώ θα αναρωτηθεί κάποιος γιατί δεν υπήρξε το ίδιο φαινόμενο και για την κίνησή του πρώτου σεναρίου. Η απάντηση είναι ότι κατά το πρώτο σενάριο η ταχύτητα πρόωσης ήταν η μέγιστη δυνατή. Ο τρόπος αντιμετώπισης του φαινομένου είναι η επιλογή καλύτερου επεξεργαστή, να μειωθεί το BASE_PERIOD έτσι ώστε να γίνουν ανεπαίσθητες οι ανωμαλίες στην πορεία του τελικού στοιχείου και για μικρές προώσεις.

Στον πραγματικό χώρο δεν μπορούσε να δοθεί αυτή η εντολή μια και ξεπερνούσε τα όρια του αγκώνα κατά πολύ. Έτσι το σημείο που επιλέχτηκε για να την ευθύγραμμη κίνηση με τερματισμό του αγκώνα είναι αυτό της επόμενης εντολής

```
G01 X[144,52] Y[0] Z[-35,4] F[2000]
```

Η θέση που προέκυψε δεν ήταν πάνω στην ευθεία που δημιουργείται από την απαίτηση των σταθερών Y, Z. Αυτό συνέβη γιατί η θέση ξεπερνά τα όρια της άρθρωσης Αγκώνα με αποτέλεσμα όταν αυτός τερμάτισε (όταν έκλεισε ο διακόπτης οριακής θέσης) ο βραχίονας προσπάθησε να προσεγγίσει περαιτέρω κατά το δυνατόν το επιθυμητό σημείο.



ΣΧΗΜΑ 5.7 Η θέση στον πραγματικό χώρο μετά την εντολή G01 στο σενάριο 2

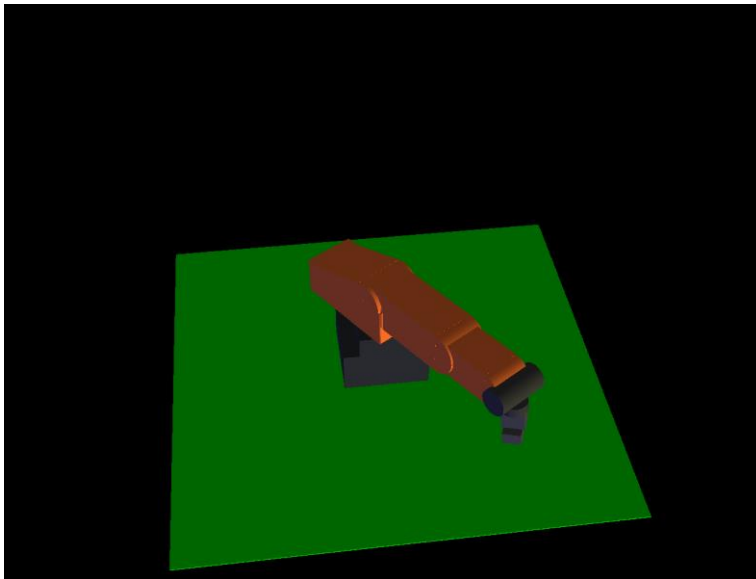
Σενάριο 3: Στο τρίτο σενάριο επιλέγεται μια θέση ώστε οι αρθρώσεις του Ώμου του Αγκώνα και του Καρπού να είναι μηδέν και τυχαία τιμή της άρθρωσης της Μέσης. Αυτή η θέση είναι η [X: 190 Y:330 Z:-35,4]. Επιλέγεται ως τερματική θέση η συμμετρική της ως προς το άξονα Y δηλαδή η [X: 190 Y:-330 Z:-35,4]. Η επιλογή αυτών των δυο σημείων έγινε έτσι ώστε η ακτίνα καμπυλότητας της πορείας να είναι σίγουρα εντός του χώρου εργασίας του βραχίονα. Η ακτίνα είναι ίση με το άθροισμα των μηκών του Μπράτσου και του Πήχη δηλαδή 380mm.

Έτσι, για καμπυλόγραμμη κίνηση με πρόωση 2000 για σταθερές αρθρώσεις Ώμου Αγκώνα και Καρπού, από την αρχική θέση οι εντολές G κώδικα είναι

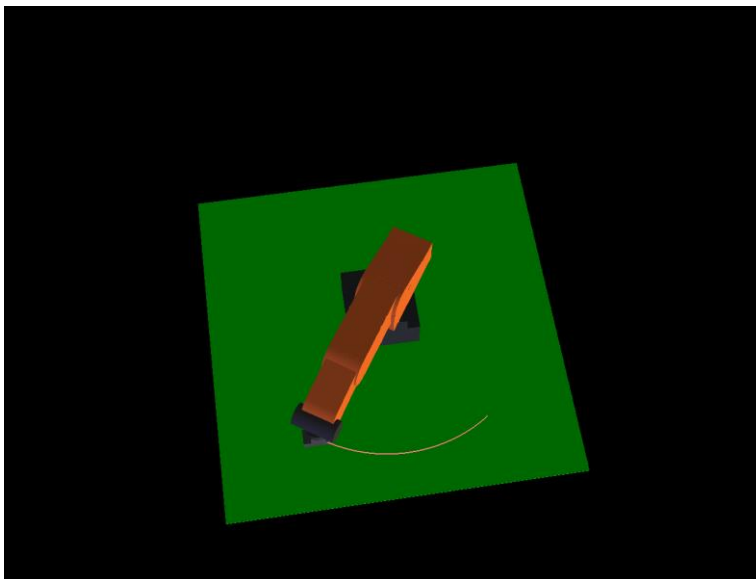
```
G00 X[190] Y[330] Z[-35,4]
```

```
G02 X[190] Y[-330] Z[-35,4] R[380] F[2000]
```

Από τα σχήματα παρατηρούμε ότι ο βραχίονας εκτελεί πορεία καμπυλοειδή με ακτίνα 380mm.

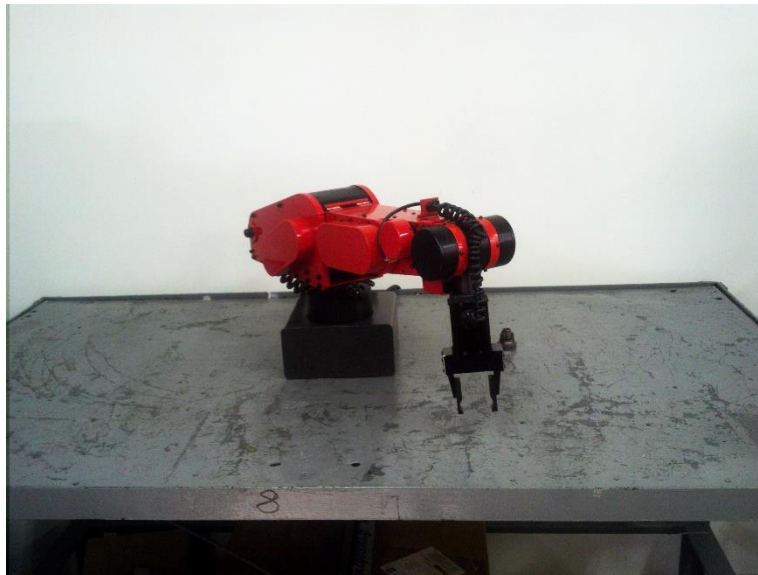


ΣΧΗΜΑ 5.8 Η θέση πριν την εντολή G02 του σεναρίου 3



ΣΧΗΜΑ 5.9 Η θέση μετά την εντολή G02 και η πορεία του τελικού στοιχείου του σεναρίου 3

Οι απαίτηση του σεναρίου για την μεταβολή μόνο της άρθρωσης της μέσης έδωσε μια ομαλή και ακριβή πορεία και για τον βραχίονα στον πραγματικό χώρο (βλ. Σχήμα 5.8 – 5.11).



ΣΧΗΜΑ 5.10 Η θέση στον πραγματικό χώρο πριν την εντολή G02 του σεναρίου 3



ΣΧΗΜΑ 5.11 Η θέση στον πραγματικό χώρο μετά την εντολή G02 του σεναρίου 3

5.3. Συμπεράσματα

Έπειτα και από την διεξαγωγή και των δοκιμών μπορούν να εξαχθούν τα εξής συμπεράσματα.

Ο ρομποτικός βραχίονας ακολουθεί πιστά το 3D γράφημα. Αυτό πρακτικά επιβεβαιώνει

- Την ορθή επικοινωνία του βραχίονα με τον υπολογιστή
- Την ορθή αποσαφήνιση των ηλεκτρονικών της μονάδας διαχείρισης
- Την σωστή επιλογή παραμέτρων κατά την διαμόρφωση του αρχείου INI
- Την επιτυχία του 'τεχνάσματος' με το αυτοτροφοδοτούμενο multiplexer κατά την διαμόρφωση του αρχείου HAL

Η διαδικασία εύρεσης της αρχικής θέσης (homing) εκτελείται με επιτυχία. Κάτι τέτοιο σημαίνει

- Σωστή τροποποίηση των σερβοενισχυτών για την άμεση επικοινωνία του λογισμικού με τους διακόπτες αρχικοποίησης
- Σωστή επιλογή μέγιστων ταχυτήτων και επιταχύνσεων σε μέτρο και φορά
- Ορθή επιλογή παραμέτρων για την διαδικασία εύρεσης της αρχικής θέσης

Προσέγγιση σημείων του χώρου εργασίας με δεδομένη πορεία. Παρόλα αυτά δεν επιτεύχθηκε η ακρίβεια των τιμών που είχε την δυνατότητα να δώσει ο ρομποτικός βραχίονας, διότι:

- Η κινηματική θεωρία είναι μερικώς λειτουργική
- Υπάρχουν σημεία ιδιομορφίας όπου ο παρών κώδικας κινηματικής εκφυλίζει τις τιμές

5.4. Προτάσεις για μελλοντική μελέτη

Ο ρομποτικός βραχίονας Mitsubishi RM-501 μετά τις τροποποιήσεις δείχνει ικανός να αποτελέσει πλατφόρμα για έρευνα και πειραματισμό. Η παρούσα πτυχιακή με την χαρτογράφηση και την αποσαφήνιση των ηλεκτρονικών του βραχίονα επιτρέπει την περαιτέρω βελτιστοποίηση του με χαμηλό κόστος. Μερικές προτάσεις είναι :

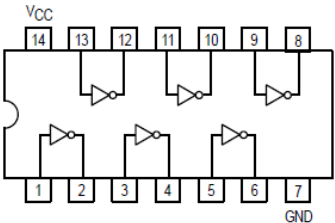
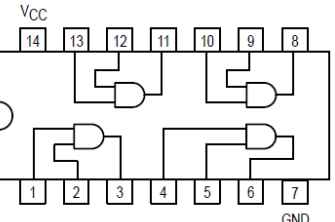
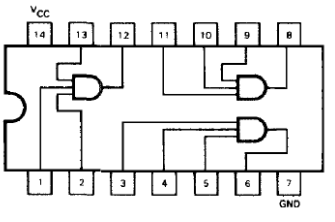
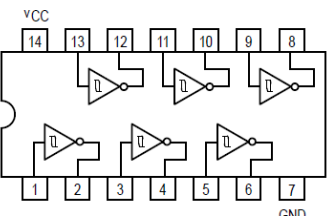
- Βελτίωση του κώδικα κινηματικής. Ο ρομποτικός βραχίονας λόγω της μικρής διαδρομής που παρουσιάζουν οι αρθρώσεις του έχει πολλά σημεία που προκύπτουν από την επίλυση της κινηματικής αλλά δεν είναι προσβάσιμα. Θα πρέπει να εκτιμηθούν αυτά τα σημεία και να τροποποιηθεί κατάλληλα ο κώδικας έτσι ώστε να αποκλείει στον βραχίονα την προσπάθεια προσέγγισης τους. Επίσης πρέπει να προβλεφθεί το σφάλμα διάχυσης που παρατηρήσαμε στο σενάριο 1. Τέλος συχνά κατά την εκτέλεση της κινηματικής προκύπτουν αρνητικές τιμές της συνάρτησης atan2 με αποτέλεσμα να εκφυλίζονται οι αντίστοιχες λύσεις.
- Βελτίωση της επικοινωνίας του υπολογιστή με τον βραχίονα. Υπάρχουν αρκετά διαθέσιμα καρφιά στις παράλληλες θύρες για την αύξηση της πληροφορίας που θα μπορούσαμε να πάρουμε από τον βραχίονα. Έτσι θα μπορούσε να δίνεται η πληροφορία θέσης του βραχίονα. Αυτό μπορεί να επιτευχθεί εάν ο Υπολογιστής είτε αναλάβει τον έλεγχο ανάδρασης είτε απλά συλλέγει την πληροφορία για εποπτικούς λόγους. Επίσης, θα μπορούσε να επικοινωνεί το κουμπί άμεσης διακοπής με το λογισμικό.

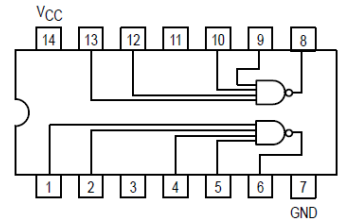
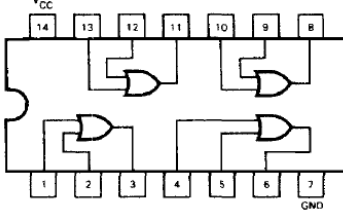
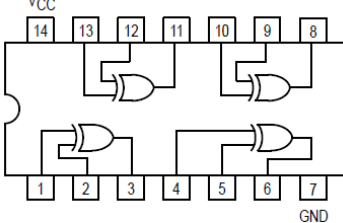
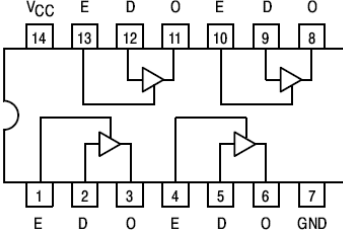
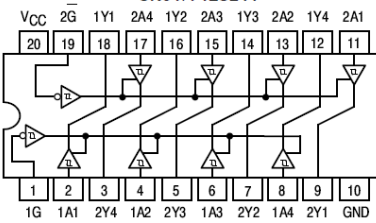
- Δυνατότητα αλλαγής αποληκτήριο εργαλείου. Στην παρούσα εργασία η αρπάγη αντιμετωπίστηκε ως το μόνο αποληκτήριο εργαλείο. Υπάρχει όμως η δυνατότητα να διαμορφωθεί κατάλληλα το λογισμικό και να δημιουργηθεί μια εργαλειοθήκη ώστε το εκάστοτε αποληκτήριο εργαλείο να επιλέγεται από μια σειρά διατιθέμενων εργαλείων.
- Αύξηση της ταχύτητας παραγωγής βημάτων. Η επιλογή αυτής της διαμόρφωσης έχει το μειονέκτημα ότι απαιτεί μεγάλο επεξεργαστικό χρόνο. Για το λόγο αυτό θα πρέπει να βρεθούν εναλλακτικές έτσι ώστε να μειωθεί ο χρόνος παραγωγής βημάτων (4x BASE_PERIOD).
- Τέλος, έχει σχεδιαστεί με την βοήθεια του λογισμικού SOLIDWORKS ένα λεπτομερές τρισδιάστατο μοντέλο του βραχίονα. Αυτό θα μπορούσε να χρησιμοποιηθεί για την μελέτη ελέγχου του ρομποτικού βραχίονα σε δυναμικό περιβάλλον κ.α.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Živanović, S., Glavonjić, M., Dimić, Z. Methodology for Configuring Desktop 3-axis Parallel Kinematic Machine, *FME Transactions*, 37, 2009, 107-115.
- [2] Glavonjic, M., Milutinovic, D., Zivanovic, S., Dimic, Z., Kvrđić, V. Desktop 3-axis parallel kinematic milling machine, *International Journal of Advanced Manufacturing Technology*, 46, 2010, 51–60/
- [3] Li, B. C., Yang, D., Lu, R. Z.. Kinematics Analysis and Control System Design of 6-DOF Parallel Kinematic Machine with Matlab and EMC2. In *Advanced Materials Research*, Vol. 102, 2010, 363-367.
- [4] Ferenc, G., Dimić, Z., Lutovac, M., Vidaković, J., Kvrđić, V. Open architecture platforms for the control of robotic systems and a proposed reference architecture model. *Transactions of FAMENA*, 37(1), 2013, 89-100.
- [5] Tajti, F., Szayer, G., Kovacs, B., Korondi, P.. Universal RT-middleware robot controller. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, 7862-7867.
- [6] Tajti, F., Szayer, G., Kovács, B., Korondi, P. Industrial robotics for ERP controlled smart factories, *Recent Innovations in Mechatronics (RIIM)*, 2 (1-2). 2015, 1-9.
- [7] Bomfim, Bracarense, Coelho, Lima, Gontijo, A low-cost methodology applied to remanufacturing of robotic manipulators, *Proc of the XX Brazilian Congress of Automation, Belo Horizonte*, 2014, 1506-1513.
- [8] Kopácsi, S., Nacsá, J., Kisari, Á. Synchronized robot control for incremental sheet forming in the cyberspace. *INFORMATIKA-A GÁBOR DÉNES FŐISKOLA KÖZLEMÉNYEI*, 14(2), 2012, 8-13.
- [9] Staroveški, T., Brezak, D., Udiljak, T., Majetić, D.. Implementation of a Linux-based CNC open control system. In *12th International Scientific Conference on Production Engineering, Computer Integrated Manufacturing and High Speed Machining, CIM 2009, 17-20/6/2009, Biograd, Croatia*, 209-216.
- [10] Zorc, D., Crneković, M. Kinematic Controller for a Mitsubishi RM501 Robot. *Transactions of FAMENA*, 36(1), 2012, 69-77.
- [11] Erlic, M., Jones, K., Lu, W. S.. Hardware interface configuration for motion control of the PUMA-560 and the Mitsubishi RM-501 robots. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1991, 266-269.
- [12] Mitsubishi Electric, *Move Master II, Instruction Manual*, 1986.
- [13] LinuxCNC, *Integrator Manual V2.5*, 2012-04-01
- [14] LinuxCNC, *HAL Manual V2.5*, 2012-04-01
- [15] LinuxCNC, *User Manual V2.6.8-21-g85ec543*, 2015-07-12

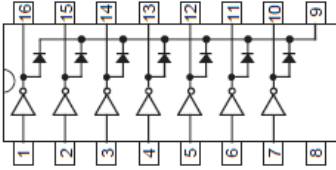
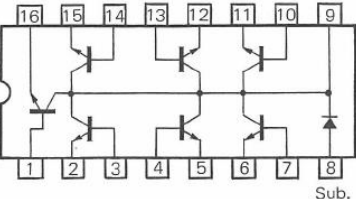
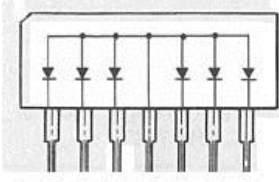
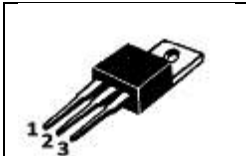
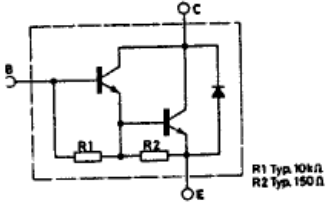
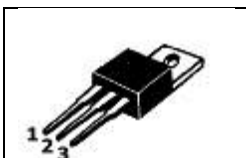
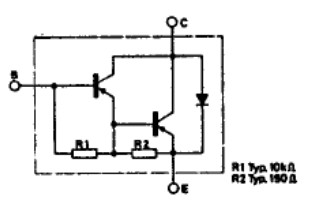
ΠΑΡΑΡΤΗΜΑ Α Κατάλογος ηλεκτρονικών RM-501

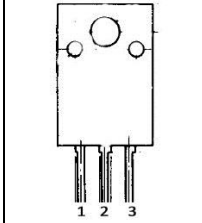
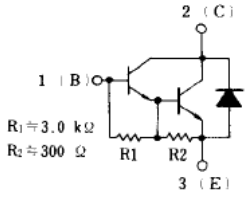
Ολοκληρωμένα Κυκλώματα σειράς LS																							
Όνομα	Διάταξη	Λογικός Πίνακας T: TRUE F: FALSE X: Άσχετο	Περιγραφή																				
M74LS04N SN74LS04N		<table border="1"> <thead> <tr> <th>IN</th> <th>OU</th> </tr> </thead> <tbody> <tr> <td></td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> </tr> </tbody> </table>	IN	OU		T	T	F	F	T	6 Λογικές Πύλες NOT												
IN	OU																						
	T																						
T	F																						
F	T																						
M74LS08N		<table border="1"> <thead> <tr> <th>IN1</th> <th>IN 2</th> <th>OUT</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> <td>F</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> </tbody> </table>	IN1	IN 2	OUT	F	F	F	T	F	F	F	T	F	T	T	T	4 Λογικές Πύλες AND Δυο Εισόδων					
IN1	IN 2	OUT																					
F	F	F																					
T	F	F																					
F	T	F																					
T	T	T																					
SN74LS11N		<table border="1"> <thead> <tr> <th>IN 1</th> <th>IN2</th> <th>IN3</th> <th>OUT</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>X</td> <td>X</td> <td>F</td> </tr> <tr> <td>X</td> <td>F</td> <td>X</td> <td>F</td> </tr> <tr> <td>X</td> <td>X</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> <td>T</td> </tr> </tbody> </table>	IN 1	IN2	IN3	OUT	F	X	X	F	X	F	X	F	X	X	F	F	T	T	T	T	3 Λογικές Πύλες AND Τριών Εισόδων
IN 1	IN2	IN3	OUT																				
F	X	X	F																				
X	F	X	F																				
X	X	F	F																				
T	T	T	T																				
M74LS14N		<table border="1"> <thead> <tr> <th>IN</th> <th>OU</th> </tr> </thead> <tbody> <tr> <td></td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> </tr> </tbody> </table>	IN	OU		T	T	F	F	T	6 Λογικές πύλες NOT Με διόρθωση θορύβου (Κατάλληλο για φιλτράρισμα)												
IN	OU																						
	T																						
T	F																						
F	T																						

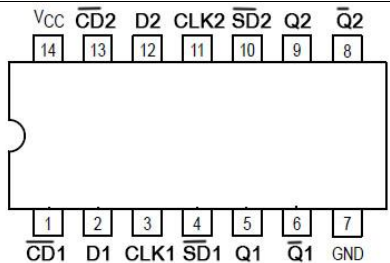
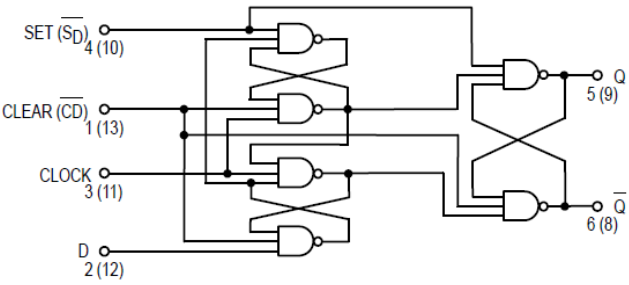
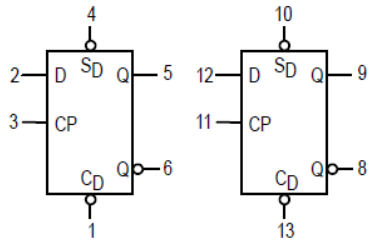
<p>SN74LS20N</p>		<table border="1"> <thead> <tr> <th>IN 1</th> <th>IN 2</th> <th>IN 3</th> <th>IN 4</th> <th>OU T</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>X</td> <td>X</td> <td>X</td> <td>T</td> </tr> <tr> <td>X</td> <td>F</td> <td>X</td> <td>X</td> <td>T</td> </tr> <tr> <td>X</td> <td>X</td> <td>F</td> <td>X</td> <td>T</td> </tr> <tr> <td>X</td> <td>X</td> <td>X</td> <td>F</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> <td>T</td> <td>F</td> </tr> </tbody> </table>	IN 1	IN 2	IN 3	IN 4	OU T	F	X	X	X	T	X	F	X	X	T	X	X	F	X	T	X	X	X	F	T	T	T	T	T	F	<p>2 Λογικές Πύλες NAND Τεσσάρων Εισόδων</p>
IN 1	IN 2	IN 3	IN 4	OU T																													
F	X	X	X	T																													
X	F	X	X	T																													
X	X	F	X	T																													
X	X	X	F	T																													
T	T	T	T	F																													
<p>M74LS32N</p>		<table border="1"> <thead> <tr> <th>IN1</th> <th>IN 2</th> <th>OUT</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>F</td> <td>T</td> </tr> <tr> <td>F</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> </tbody> </table>	IN1	IN 2	OUT	F	F	F	T	F	T	F	T	T	T	T	T	<p>4 Λογικές Πύλες OR Δυο Εισόδων</p>															
IN1	IN 2	OUT																															
F	F	F																															
T	F	T																															
F	T	T																															
T	T	T																															
<p>M74LS86N</p>		<table border="1"> <thead> <tr> <th>IN1</th> <th>IN 2</th> <th>OUT</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>F</td> <td>T</td> </tr> <tr> <td>F</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>F</td> </tr> </tbody> </table>	IN1	IN 2	OUT	F	F	F	T	F	T	F	T	T	T	T	F	<p>4 Λογικές Πύλες XOR Δυο Εισόδων</p>															
IN1	IN 2	OUT																															
F	F	F																															
T	F	T																															
F	T	T																															
T	T	F																															
<p>SN74LS126N</p>		<table border="1"> <thead> <tr> <th>INPU T</th> <th>OU T</th> </tr> </thead> <tbody> <tr> <td>E D O</td> <td>T</td> </tr> <tr> <td>T F F</td> <td>F</td> </tr> <tr> <td>T T T</td> <td>T</td> </tr> <tr> <td>F X OF</td> <td>F</td> </tr> </tbody> </table>	INPU T	OU T	E D O	T	T F F	F	T T T	T	F X OF	F	<p>4 Buffers Η έξοδος (O) λαμβάνει το σήμα της εισόδου (D) όταν «ανοίξει η πύλη» (E)</p>																				
INPU T	OU T																																
E D O	T																																
T F F	F																																
T T T	T																																
F X OF	F																																
<p>SN74LS244N</p>		<p>-</p>	<p>8 Buffers Με διόρθωση θορύβου (Κατάλληλο για φιλτράρισμα)</p>																														

<p>SN74LS279N</p>		<table border="1"> <thead> <tr> <th colspan="3">INPUT</th> <th>OUT</th> </tr> <tr> <th>S</th> <th>S₂</th> <th>R</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td></td> <td></td> </tr> <tr> <td>F</td> <td>F</td> <td>F</td> <td>T</td> </tr> <tr> <td>F</td> <td>X</td> <td>T</td> <td>T</td> </tr> <tr> <td>X</td> <td>F</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>L</td> <td>L</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> <td>HOLD</td> </tr> </tbody> </table>	INPUT			OUT	S	S ₂	R	Q	1				F	F	F	T	F	X	T	T	X	F	T	T	T	T	L	L	T	T	T	HOLD	<p>4 Set Reset Flip Flops Τριών Εισόδων</p>
INPUT			OUT																																
S	S ₂	R	Q																																
1																																			
F	F	F	T																																
F	X	T	T																																
X	F	T	T																																
T	T	L	L																																
T	T	T	HOLD																																
<p>M75401</p>		<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>F</td> <td>T</td> </tr> </tbody> </table>	INPUT		OUT	A	B	Y	F	F	F	T	F	F	F	T	F	F	F	T	<p>2 NAND (για την επικοινωνία με περιφερικά κυκλώματα υψηλών και πολύ υψηλών ρευμάτων) Μέγιστη Τάση 30V</p>														
INPUT		OUT																																	
A	B	Y																																	
F	F	F																																	
T	F	F																																	
F	T	F																																	
F	F	T																																	
<p>M75411</p>		<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> <tr> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>F</td> <td>T</td> </tr> </tbody> </table>	INPUT		OUT	A	B	Y	F	F	F	T	F	F	F	T	F	F	F	T	<p>2 NAND (για την επικοινωνία με περιφερικά κυκλώματα υψηλών και πολύ υψηλών ρευμάτων) Μέγιστη Τάση 55V</p>														
INPUT		OUT																																	
A	B	Y																																	
F	F	F																																	
T	F	F																																	
F	T	F																																	
F	F	T																																	

Εξειδικευμένα Ολοκληρωμένα Κυκλώματα και άλλα Ηλεκτρονικά			
Όνομα	Διάταξη	Τεχνικά Στοιχεία	Περιγραφή
<p>TL054CN</p>		<p>Απολυτές Μέγιστες Τιμές</p> <p>VCC ±18 V</p> <p>V_i (input Voltage) ±15 V</p> <p>V_{id} (Differential Input Voltage) ±30 V</p>	<p>4 Τελεστικοί ενισχυτές</p>

<p>M54523P</p>		<p>7 NPN Darlington τρανζίστορ σε σειρά με δίοδο Καταλληλά να αποδώσουν μεγάλα ρεύματα με μικρά ρεύματα για είσοδο</p>				
<p>μPA56C</p>	 <p style="text-align: right;">Sub.</p>	<p>Έχει αποσυρθεί από την αγορά</p>	<p>7 NPN τρανζίστορ σε σειρά</p>			
<p>μPA64H</p>		<p>Έχει αποσυρθεί από την αγορά</p>	<p>6 δίοδοι σε σειρά</p>			
<p>BDX53</p>	 <table border="1" data-bbox="477 1318 721 1432"> <tr> <td>1. ΒΑΣΗ</td> </tr> <tr> <td>2. ΣΥΛΛΕΚΤΗΣ</td> </tr> <tr> <td>3. ΕΚΠΟΜΠΟΣ</td> </tr> </table>	1. ΒΑΣΗ	2. ΣΥΛΛΕΚΤΗΣ	3. ΕΚΠΟΜΠΟΣ	 <p>R1 Τύπ. 10kΩ R2 Τύπ. 150Ω</p>	<p>NPN Darlington Τρανζίστορ</p>
1. ΒΑΣΗ						
2. ΣΥΛΛΕΚΤΗΣ						
3. ΕΚΠΟΜΠΟΣ						
<p>BDX54</p>	 <table border="1" data-bbox="477 1617 721 1730"> <tr> <td>1. ΒΑΣΗ</td> </tr> <tr> <td>2. ΣΥΛΛΕΚΤΗΣ</td> </tr> <tr> <td>3. ΕΚΠΟΜΠΟΣ</td> </tr> </table>	1. ΒΑΣΗ	2. ΣΥΛΛΕΚΤΗΣ	3. ΕΚΠΟΜΠΟΣ	 <p>R1 Τύπ. 10kΩ R2 Τύπ. 150Ω</p>	<p>PNP Darlington Τρανζίστορ</p>
1. ΒΑΣΗ						
2. ΣΥΛΛΕΚΤΗΣ						
3. ΕΚΠΟΜΠΟΣ						

<p>2SD1309</p>	 <p>1.ΒΑΣΗ 2.ΣΥΛΛΕΚΤΗΣ 3.ΕΚΠΟΜΠΟΣ</p>	 <p>$R_1 \approx 3.0 \text{ k}\Omega$ $R_2 \approx 300 \Omega$</p>	<p>NPN Darlington Τρανζίστορ</p>
-----------------------	--	--	--------------------------------------

Χρονισμένα Ολοκληρωμένα Κυκλώματα (Clocked Components)																																					
<p>Όνομα - Περιγραφή</p> <p>SN74LS74AN</p> <p>2, Τύπου D, Flip Flops ανεξάρτητου χρονισμού, ενεργοποίησης από θετική μεριά μεταβολής του χρονισμού</p>	<p>Διάταξη</p> 																																				
<p>Λογικό Διάγραμμα για κάθε Flip Flop</p> 	<p>Συμβολισμός</p>  <p>$V_{CC} = \text{PIN } 14$ $GND = \text{PIN } 7$</p>																																				
Λογικός Πίνακας																																					
<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>SET</th> <th>CLEAR</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>T</td> <td>F</td> <td>X</td> </tr> <tr> <td>F</td> <td>T</td> <td>X</td> </tr> <tr> <td>F</td> <td>F</td> <td>X</td> </tr> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>F</td> </tr> </tbody> </table>			INPUT		OUTPUT	SET	CLEAR	D	T	F	X	F	T	X	F	F	X	T	T	T	T	T	F	<table border="1"> <thead> <tr> <th colspan="2">OUTPUT</th> </tr> <tr> <th>Q</th> <th>\bar{Q}</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> </tr> <tr> <td>T</td> <td>T</td> </tr> <tr> <td>F</td> <td>F</td> </tr> </tbody> </table>		OUTPUT		Q	\bar{Q}	F	T	T	F	T	T	F	F
INPUT		OUTPUT																																			
SET	CLEAR	D																																			
T	F	X																																			
F	T	X																																			
F	F	X																																			
T	T	T																																			
T	T	F																																			
OUTPUT																																					
Q	\bar{Q}																																				
F	T																																				
T	F																																				
T	T																																				
F	F																																				
<p>T: TRUE F: FALSE X: Οτιδήποτε ↑: Ενεργοποίηση κατά την μεταβολή του χρονισμού(CLOCK) από F →T</p>																																					

Χρονισμένα Ολοκληρωμένα Κυκλώματα (Clocked Components)																					
Όνομα - Περιγραφή	Διάταξη																				
<p>SN74LS175N</p> <p>4 Τύπου D Flip Flops σύγχρονου χρονισμού, ενεργοποίησης από θετική μεριά μεταβολής του χρονισμού με δυνατότητα επανακαθορισμού (reset)</p>																					
Λογικό Διάγραμμα	Συμβολισμός																				
<p>VCC = PIN 16 GND = PIN 8 ○ = PIN NUMBERS</p>	<p>VCC = PIN 16 GND = PIN 8</p>																				
Λογικός Πίνακας																					
<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th colspan="2">OUTPUT</th> </tr> <tr> <th>MR</th> <th>D</th> <th>Q</th> <th>\bar{Q}</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>X</td> <td>F</td> <td>T</td> </tr> <tr> <td>T</td> <td>T</td> <td>T↑</td> <td>F↑</td> </tr> <tr> <td>T</td> <td>F</td> <td>F↑</td> <td>T↑</td> </tr> </tbody> </table>	INPUT		OUTPUT		MR	D	Q	\bar{Q}	F	X	F	T	T	T	T↑	F↑	T	F	F↑	T↑	<p>T: TRUE F: FALSE X: Οτιδήποτε ↑: Ενεργοποίηση κατά την μεταβολή του χρονισμού(CP) από F →T</p>
INPUT		OUTPUT																			
MR	D	Q	\bar{Q}																		
F	X	F	T																		
T	T	T↑	F↑																		
T	F	F↑	T↑																		

Χρονισμένα Ολοκληρωμένα Κυκλώματα (Clocked Components)																																																																																																							
Όνομα - Περιγραφή	Διάταξη-Συμβολισμός																																																																																																						
<p>M74LS175P</p> <p>4-Bit καταμετρητής. Λαμβάνει ένα σήμα χρονισμού και κατά την μετάβαση από T→F μετρά 1 και το προσθέτει στο προηγούμενο σύνολο. Το σύνολο αυτό αποδίδεται είτε δυαδικά είτε οκταδικά. Διαλέγοντας την κατάσταση (T ή F) των R_{D1} R_{D2} μπορεί να επιλεγεί με ποιο τρόπο από τους δυο πρέπει να αποδίδεται το σύνολο.</p> <table border="1"> <tr> <th>R_{D1}</th> <th>R_{D2}</th> <th></th> </tr> <tr> <td>F</td> <td>T</td> <td>Μόνο Δυαδικό BIN</td> </tr> <tr> <td>T</td> <td>F</td> <td>Μόνο Οκταδικό OCT</td> </tr> <tr> <td>F</td> <td>F</td> <td>BIN+OCT</td> </tr> <tr> <td>T</td> <td>T</td> <td>Δεν μετρά</td> </tr> </table>	R_{D1}	R_{D2}		F	T	Μόνο Δυαδικό BIN	T	F	Μόνο Οκταδικό OCT	F	F	BIN+OCT	T	T	Δεν μετρά																																																																																								
R_{D1}	R_{D2}																																																																																																						
F	T	Μόνο Δυαδικό BIN																																																																																																					
T	F	Μόνο Οκταδικό OCT																																																																																																					
F	F	BIN+OCT																																																																																																					
T	T	Δεν μετρά																																																																																																					
Λογικό Διάγραμμα	Ορολογία																																																																																																						
	<p>\bar{T}_1: Ρολί χρονισμού για καταμέτρηση δυαδικά \bar{T}_2: Ρολί χρονισμού για καταμέτρηση οκταδικά R_{D1}: Επιλογή δυαδικής καταμέτρησης R_{D2}: Επιλογή οκταδικής καταμέτρησης Q_A: Καταμέτρηση LSB Q_B: Καταμέτρηση Q_C: Καταμέτρηση Q_D: Καταμέτρηση MSB</p>																																																																																																						
Λογικός Πίνακας																																																																																																							
	<table border="1"> <thead> <tr> <th>A/A</th> <th>\bar{T}</th> <th>Q_A</th> <th>Q_B</th> <th>Q_C</th> <th>Q_D</th> </tr> </thead> <tbody> <tr><td>0</td><td></td><td>F</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>1</td><td>↓</td><td>T</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>2</td><td>↓</td><td>F</td><td>T</td><td>F</td><td>F</td></tr> <tr><td>3</td><td>↓</td><td>T</td><td>T</td><td>F</td><td>F</td></tr> <tr><td>4</td><td>↓</td><td>F</td><td>F</td><td>T</td><td>F</td></tr> <tr><td>5</td><td>↓</td><td>T</td><td>F</td><td>T</td><td>F</td></tr> <tr><td>6</td><td>↓</td><td>F</td><td>T</td><td>T</td><td>F</td></tr> <tr><td>7</td><td>↓</td><td>T</td><td>T</td><td>T</td><td>F</td></tr> <tr><td>8</td><td>↓</td><td>F</td><td>F</td><td>F</td><td>T</td></tr> <tr><td>9</td><td>↓</td><td>T</td><td>F</td><td>F</td><td>T</td></tr> <tr><td>10</td><td>↓</td><td>F</td><td>T</td><td>F</td><td>T</td></tr> <tr><td>11</td><td>↓</td><td>T</td><td>T</td><td>F</td><td>T</td></tr> <tr><td>12</td><td>↓</td><td>F</td><td>F</td><td>T</td><td>T</td></tr> <tr><td>13</td><td>↓</td><td>T</td><td>F</td><td>T</td><td>T</td></tr> <tr><td>14</td><td>↓</td><td>F</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>15</td><td>↓</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> </tbody> </table>	A/A	\bar{T}	Q_A	Q_B	Q_C	Q_D	0		F	F	F	F	1	↓	T	F	F	F	2	↓	F	T	F	F	3	↓	T	T	F	F	4	↓	F	F	T	F	5	↓	T	F	T	F	6	↓	F	T	T	F	7	↓	T	T	T	F	8	↓	F	F	F	T	9	↓	T	F	F	T	10	↓	F	T	F	T	11	↓	T	T	F	T	12	↓	F	F	T	T	13	↓	T	F	T	T	14	↓	F	T	T	T	15	↓	T	T	T	T
A/A	\bar{T}	Q_A	Q_B	Q_C	Q_D																																																																																																		
0		F	F	F	F																																																																																																		
1	↓	T	F	F	F																																																																																																		
2	↓	F	T	F	F																																																																																																		
3	↓	T	T	F	F																																																																																																		
4	↓	F	F	T	F																																																																																																		
5	↓	T	F	T	F																																																																																																		
6	↓	F	T	T	F																																																																																																		
7	↓	T	T	T	F																																																																																																		
8	↓	F	F	F	T																																																																																																		
9	↓	T	F	F	T																																																																																																		
10	↓	F	T	F	T																																																																																																		
11	↓	T	T	F	T																																																																																																		
12	↓	F	F	T	T																																																																																																		
13	↓	T	F	T	T																																																																																																		
14	↓	F	T	T	T																																																																																																		
15	↓	T	T	T	T																																																																																																		

Χρονισμένα Ολοκληρωμένα Κυκλώματα (Clocked Components)				
Όνομα - Περιγραφή		Διάταξη		
<p>SN74LS173N</p> <p>Καταμετρητής UP-DOWN τεσσάρων BIT. Ο καταμετρητής αυτός προσθέτει (UP) ή αφαιρεί (DOWN) 1 μονάδα από το σύνολο, κατά την μετάβαση από F→T του CP_U ή του CP_D αντίστοιχα. Έχει την δυνατότητα προκαθορισμού του συνόλου θέτοντας το PL→F και ορίζοντας στα P₀ P₁ P₂ P₃ το σύνολο εκφρασμένο σε δεκαεξαδικό σύστημα. Σε υπερχειλίση (overflow) τα TC_U TC_D αλλάζουν την κατάσταση τους έτσι ώστε να μεταφέρουν στην πληροφορία σε άλλον καταμετρητή UP-DOWN αλλιώς αν δεν υπάρχει άλλος καταμετρητής το MR (reset) γίνεται T και το σύνολο μηδενίζεται</p>				
<p>Ορολογία</p> <p>CP_U : Ρολί χρονισμού για την πρόσθεση μιας μονάδας CP_D : Ρολί χρονισμού για την αφαίρεση μιας μονάδας TC_U : Μετάβαση του παλμού CP_U σε άλλο LS173 TC_D : Μετάβαση του παλμού CP_D σε άλλο LS173 PL : FALSE λαμβάνει προκαθορισμένα δεδομένα MR : TRUE επανακαθορισμός στην αρχική κατάσταση P₀ P₁ P₂ P₃ : Προκαθορισμένα δεδομένα Q₀ Q₁ Q₂ Q₃ : Το σύνολο σε δεκαεξαδική μορφή</p>		<p>Συμβολισμός</p>		
Λογικός Πίνακας				
MR	PL	CP _U	CP _D	MODE
T	X	X	X	RESET
F	F	X	X	PRESET
F	T	T	T	HOLD
F	T	↑	T	+1
F	T	T	↑	-1

ΠΑΡΑΡΤΗΜΑ Β ΑΡΧΕΙΑ κώδικα

rm501.hal

```
# core HAL config file for simulation - 5 axis

# load RT modules
loadrt rm501kins
loadrt [EMCMOT]EMCMOT base_period_nsec=[EMCMOT]BASE_PERIOD
servo_period_nsec=[EMCMOT]SERVO_PERIOD traj_period_nsec=[EMCMOT]TRAJ_PERIOD
num_joints=[TRAJ]AXES

#Load Real Time Components
loadrt hal_parport cfg="0x378 out 0xE000 out"
loadrt stepgen step_type=0,0,0,0,0 ctrl_type=p,p,p,p,p,p
loadrt mux_generic config=uu8,bb4,bb4,bb4,bb4
loadrt bitslice count=1 personality=9
loadrt mux2
loadrt and2
loadrt sum2 count=2
loadrt or2

#Hook Functions to Base Thread
addf or2.0 base-thread
addf stepgen.make-pulses base-thread
addf mux-gen.00 base-thread
addf bitslice.0 base-thread
addf mux-gen.02 base-thread
addf mux-gen.01 base-thread
addf mux-gen.04 base-thread
addf mux-gen.03 base-thread
addf and2.0 base-thread
addf parport.0.write base-thread
addf parport.1.write base-thread
addf parport.0.read base-thread
addf parport.1.read base-thread
addf parport.0.reset base-thread
addf parport.1.reset base-thread

#Hook Functions to Servo Thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf stepgen.update-freq servo-thread
addf stepgen.capture-position servo-thread
addf mux2.0 servo-thread
addf sum2.0 servo-thread
addf sum2.1 servo-thread

# Connect Position Commands from Motion Module to Step Generator
net J0pos axis.0.motor-pos-cmd => stepgen.0.position-cmd
net J1pos axis.1.motor-pos-cmd => stepgen.1.position-cmd
net J2pos axis.2.motor-pos-cmd => stepgen.2.position-cmd
net J3pos axis.3.motor-pos-cmd => sum2.0.in0 sum2.1.in0
net J4pos axis.4.motor-pos-cmd => sum2.0.in1 sum2.1.in1
net Lwpos sum2.0.out => stepgen.3.position-cmd
net Rwpos sum2.1.out => stepgen.4.position-cmd

#Connect Position Commands Feedback from Step Generator to Motion Module
net J0pos axis.0.motor-pos-fb
```

```

net J1pos          axis.1.motor-pos-fb
net J2pos          axis.2.motor-pos-fb
net J3pos          axis.3.motor-pos-fb
net J4pos          axis.4.motor-pos-fb

#net J0pos-fb      stepgen.0.position-fb => axis.0.motor-pos-fb
#net J1pos-fb      stepgen.1.position-fb => axis.1.motor-pos-fb
#net J2pos-fb      stepgen.2.position-fb => axis.2.motor-pos-fb
#net J3pos-fb      stepgen.3.position-fb => axis.3.motor-pos-fb
#net J4pos-fb      stepgen.4.position-fb => axis.4.motor-pos-fb

#Connect Enable Signals for Step Generators
net J0en           axis.0.amp-enable-out => stepgen.0.enable
net J1en           axis.1.amp-enable-out => stepgen.1.enable
net J2en           axis.2.amp-enable-out => stepgen.2.enable
net J3en           axis.3.amp-enable-out => stepgen.3.enable
net J4en           axis.4.amp-enable-out => stepgen.4.enable

# Create a data table for self-driving mux, strobe and joint select
setp mux-gen.00.in-u32-00 0x48
setp mux-gen.00.in-u32-01 0x89
setp mux-gen.00.in-u32-02 0xD2
setp mux-gen.00.in-u32-03 0x113
setp mux-gen.00.in-u32-04 0x164
setp mux-gen.00.in-u32-05 0x1A5
setp mux-gen.00.in-u32-06 0x1C6
setp mux-gen.00.in-u32-07 0x7

# Break the data table to binary
net data           mux-gen.00.out-u32 => bitslice.0.in

# Loop Back for self-driving mux
net addr0          bitslice.0.out-06 => mux-gen.00.sel-bit-00
net addr1          bitslice.0.out-07 => mux-gen.00.sel-bit-01
net addr2          bitslice.0.out-08 => mux-gen.00.sel-bit-02

#Select Output
net sel0           bitslice.0.out-01 => and2.0.in0 mux-gen.01.sel-bit-00
                  mux-gen.02.sel-bit-00 mux-gen.03.sel-bit-00 mux-gen.04.sel-bit-00
net sel1           bitslice.0.out-02 => and2.0.in1 mux-gen.01.sel-bit-01
                  mux-gen.02.sel-bit-01 mux-gen.03.sel-bit-01 mux-gen.04.sel-bit-01

# Link Steps
net elbw_stepstepgen.2.step => mux-gen.02.in-bit-00
net base_stepstepgen.0.step => mux-gen.02.in-bit-01
net hand_stepstepgen.5.step => mux-gen.02.in-bit-03

net shld_stepstepgen.1.step => mux-gen.04.in-bit-00
net lwst_stepstepgen.4.step => mux-gen.04.in-bit-01
net rwst_stepstepgen.3.step => mux-gen.04.in-bit-02

# Link Directions
net elbw_dir stepgen.2.dir => mux-gen.01.in-bit-00
net base_dir stepgen.0.dir => mux-gen.01.in-bit-01
net hand_dir stepgen.5.dir => mux-gen.02.in-bit-02

net shld_dir stepgen.1.dir => mux-gen.03.in-bit-00
net lwst_dir stepgen.4.dir => mux-gen.03.in-bit-01
net rwst_dir stepgen.3.dir => mux-gen.03.in-bit-02

```

```

# Link muxes, strobe to parport
net strobe      bitslice.0.out-00 => parport.0.pin-04-out
net dir_01     mux-gen.01.out-bit => parport.0.pin-03-out parport.0.pin-08-out
net dir_02     mux-gen.03.out-bit => parport.1.pin-03-out
net step_01    mux-gen.02.out-bit => parport.0.pin-02-out parport.0.pin-07-
out parport.0.pin-09-out
net step_02    mux-gen.04.out-bit => parport.1.pin-01-out

#Net select joint
net elbw_shld  bitslice.0.out-03 => parport.0.pin-05-out parport.1.pin-04-out
net base_lwst  bitslice.0.out-04 => parport.0.pin-06-out parport.1.pin-05-out
net hand_rwst  bitslice.0.out-05 => parport.1.pin-08-out parport.1.pin-06-out
net hand_pwr   and2.0.out        => parport.1.pin-09-out

#Net Home Switches
net elbw_sw    parport.0.pin-11-in => axis.2.home-sw-in
net base_sw    parport.0.pin-12-in => axis.0.home-sw-in
net shld_sw    parport.1.pin-11-in => axis.1.home-sw-in
net pitch_sw   parport.1.pin-13-in => axis.3.home-sw-in
net rol_sw     parport.1.pin-12-in => axis.4.home-sw-in

#Net Reset, Errors
#net error_1   parport.0.pin-10-in-not => or2.0.in0
#net error_2   parport.1.pin-10-in-not => or2.0.in1
#net reset     or2.0.out              => parport.0.pin-17-out

#Grip Configuration
setp mux-gen.01.in-bit-02 TRUE
setp mux-gen.01.in-bit-03 FALSE
setp stepgen.5.enable TRUE
setp mux2.0.in0 -1
setp mux2.0.in1 1
net clopen      mux2.0.out => stepgen.5.position-cmd
setp stepgen.5.position-scale 400
setp stepgen.5.steplen 390000
setp stepgen.5.stepspace 390000
setp stepgen.5.dirhold 190000
setp stepgen.5.dirsetup 190000

#Diferential Configuration
setp sum2.0.gain0 1
setp sum2.0.gain1 1
setp sum2.1.gain0 -1
setp sum2.1.gain1 1

#Test
#setp stepgen.5.enable 1
#net sw siggen.0.clock => mux2.0.sel

#Parport Parameters Configuration
setp parport.0.pin-02-out-invert TRUE
setp parport.0.pin-03-out-invert TRUE
setp parport.0.pin-04-out-invert TRUE
setp parport.0.pin-05-out-invert TRUE
setp parport.0.pin-06-out-invert TRUE
setp parport.0.pin-07-out-invert TRUE
setp parport.0.pin-08-out-invert TRUE
setp parport.0.pin-09-out-invert TRUE
setp parport.0.pin-16-out-invert TRUE

```



```

setp parport.0.pin-17-out-invert      TRUE

setp parport.1.pin-01-out-invert      TRUE
setp parport.1.pin-02-out-invert      TRUE
setp parport.1.pin-03-out-invert      TRUE
setp parport.1.pin-04-out-invert      TRUE
setp parport.1.pin-05-out-invert      TRUE
setp parport.1.pin-06-out-invert      TRUE
setp parport.1.pin-07-out-invert      TRUE
setp parport.1.pin-08-out-invert      TRUE
setp parport.1.pin-09-out-invert      TRUE
setp parport.1.pin-14-out-invert      TRUE
setp parport.1.pin-16-out-invert      TRUE
setp parport.1.pin-17-out-invert      TRUE

#Stepping Configuration
setp stepgen.0.steplen                 [AXIS_0]STEP_LENGTH
setp stepgen.1.steplen                 [AXIS_1]STEP_LENGTH
setp stepgen.2.steplen                 [AXIS_2]STEP_LENGTH
setp stepgen.3.steplen                 [AXIS_3]STEP_LENGTH
setp stepgen.4.steplen                 [AXIS_4]STEP_LENGTH

setp stepgen.0.stepspace               [AXIS_0]STEP_SPACE
setp stepgen.1.stepspace               [AXIS_1]STEP_SPACE
setp stepgen.2.stepspace               [AXIS_2]STEP_SPACE
setp stepgen.3.stepspace               [AXIS_3]STEP_SPACE
setp stepgen.4.stepspace               [AXIS_4]STEP_SPACE

setp stepgen.0.dirhold                 [AXIS_0]DIR_HOLD
setp stepgen.1.dirhold                 [AXIS_1]DIR_HOLD
setp stepgen.2.dirhold                 [AXIS_2]DIR_HOLD
setp stepgen.3.dirhold                 [AXIS_3]DIR_HOLD
setp stepgen.4.dirhold                 [AXIS_4]DIR_HOLD

setp stepgen.0.dirsetup                [AXIS_0]DIR_SETUP
setp stepgen.1.dirsetup                [AXIS_1]DIR_SETUP
setp stepgen.2.dirsetup                [AXIS_2]DIR_SETUP
setp stepgen.3.dirsetup                [AXIS_3]DIR_SETUP
setp stepgen.4.dirsetup                [AXIS_4]DIR_SETUP

setp stepgen.0.position-scale          [AXIS_0]SCALE
setp stepgen.1.position-scale          [AXIS_1]SCALE
setp stepgen.2.position-scale          [AXIS_2]SCALE
setp stepgen.3.position-scale          [AXIS_3]SCALE
setp stepgen.4.position-scale          [AXIS_4]SCALE

setp stepgen.0.maxaccel                 [AXIS_0]STEPGEN_MAXACCEL
setp stepgen.1.maxaccel                 [AXIS_1]STEPGEN_MAXACCEL
setp stepgen.2.maxaccel                 [AXIS_2]STEPGEN_MAXACCEL
setp stepgen.3.maxaccel                 [AXIS_3]STEPGEN_MAXACCEL
setp stepgen.4.maxaccel                 [AXIS_4]STEPGEN_MAXACCEL

#setp stepgen.0.maxvel                  [AXIS_0]STEPGEN_MAXVEL
setp stepgen.1.maxvel                  [AXIS_1]STEPGEN_MAXVEL
setp stepgen.2.maxvel                  [AXIS_2]STEPGEN_MAXVEL
setp stepgen.3.maxvel                  [AXIS_3]STEPGEN_MAXVEL
setp stepgen.4.maxvel                  [AXIS_4]STEPGEN_MAXVEL

# Estop Loopback
net estop-loop iocontrol.0.user-enable-out iocontrol.0.emc-enable-in

```

```

#Tool Loading Loopback
net tool-prep-loop      ioccontrol.0.tool-prepare =>   ioccontrol.0.tool-prepared
net tool-change-loop    ioccontrol.0.tool-change  =>   ioccontrol.0.tool-changed

#Simulation
loadusr -W rm50lgui

loadrt scale count=5

addf scale.0 servo-thread
addf scale.1 servo-thread
addf scale.2 servo-thread
addf scale.3 servo-thread
addf scale.4 servo-thread

net J0sim    axis.0.joint-pos-cmd
net J1sim    axis.1.joint-pos-cmd
net J2sim    axis.2.joint-pos-cmd
net J3sim    axis.3.joint-pos-cmd
net J4sim    axis.4.joint-pos-cmd

net J0sim scale.0.in
net J1sim scale.1.in
net J2sim scale.2.in
net J3sim scale.3.in
net J4sim scale.4.in

setp scale.0.gain 1
setp scale.1.gain 1
setp scale.2.gain 1
setp scale.3.gain 1
setp scale.4.gain 1

net J0scaled scale.0.out =>   rm50lgui.joint1
net J1scaled scale.1.out =>   rm50lgui.joint2
net J2scaled scale.2.out =>   rm50lgui.joint3
net J3scaled scale.3.out =>   rm50lgui.joint4
net J4scaled scale.4.out =>   rm50lgui.joint5

```

rm501.ini

```

# EMC controller parameters for MITSUBISHI MOVEMASTER RM-501.

# General note: Comments can either be preceded with a # or ; - either is
# acceptable, although # is in keeping with most linux config files.

# Settings with a + at the front of the comment are likely needed to get
# changed by the user.
# Settings with a - at the front are highly unneeded to be changed
#####
# General section
#####
# General section -----

```

```
[EMC]

#- Version of this INI file
VERSION      =      $Revision$

#+ Name of machine, for use with display, etc.
MACHINE      =      MISTUBISHI MOVEMASTER RM-501

#+ Debug level, 0 means no messages. See src/emc/nml_int/emcglb.h for others
DEBUG       =      0
#DEBUG      =      0x00000007
#DEBUG      =      0x7FFFFFFF

#####
# Sections for display options
#####
[DISPLAY]

# Name of display program, e.g., xemc
DISPLAY     =      axis
#DISPLAY    =      usrmot
#DISPLAY    =      mini
#DISPLAY    =      tkemc

#- Cycle time, in seconds, that display will sleep between polls
CYCLE_TIME  =      0.200

#- Path to help file
HELP_FILE   =      tklinucnc.txt

#- Initial display setting for position, RELATIVE or MACHINE
POSITION_OFFSET =      MACHINE

#- Initial display setting for position, COMMANDED or ACTUAL
POSITION_FEEDBACK =      ACTUAL

#+ Highest value that will be allowed for feed override, 1.0 = 100%
MAX_FEED_OVERRIDE =      2.0

#+ Prefix to be used
PROGRAM_PREFIX =      ../../nc_files/

#- Introductory graphic
INTRO_GRAPHIC =      NTUA-logo.gif
INTRO_TIME    =      6
#PYVCP       =      rm501.xml

# Editor to be used with Axis
EDITOR       =      gedit

#####
# Task controller section
#####
[TASK]

# Name of task controller program, e.g., milltask
TASK        =      milltask

#- Cycle time, in seconds, that task controller will sleep between polls
CYCLE_TIME  =      0.010
```

```
#####
# Part program interpreter section
#####
[RS274NGC]

#- File containing interpreter variables
PARAMETER_FILE           =      rm501.var

# M101 (open grip) and M102 (close grip) files
USER_M_PATH              =/home/mechcnc/linuxcnc/nc_files/ngcgui_lib/mfiles

#####
# Motion control section
#####
[EMCMOT]

EMCMOT                   =      motmod
COMM_TIMEOUT             =      1.0
COMM_WAIT                =      0.010

BASE_PERIOD              =      40000
SERVO_PERIOD             =      1000000
TRAJ_PERIOD              =      10000000

#####
# Hardware Abstraction Layer section
#####
[HAL]

# The run script first uses halcmd to execute any HALFILE
# files, and then to execute any individual HALCMD commands.

# list of hal config files to run through halcmd
# files are executed in the order in which they appear

HALFILE                  =      rm501.hal
#HALFILE                 =      rm501_.hal
#POSTGUI_HALFILE        =      rm501_postgui.hal
#HALCMD                  =      save neta
HALUI                    =      halui

#####
# Trajectory planner section
#####
[TRAJ]
#+ machine specific settings
AXES                     =      5
COORDINATES              =      X Y Z A B
HOME                     =      0 0 0 0 0
LINEAR_UNITS             =      mm
ANGULAR_UNITS            =      deg
CYCLE_TIME               =      0.010
DEFAULT_VELOCITY         =      1.0
MAX_VELOCITY             =      33
DEFAULT_ACCELERATION     =      10
MAX_ACCELERATION         =      20
#POSITION_FILE           =      rm-501_position.txt
#####
# Axes sections
#####
```

```

#*****#
#   WAIST   #
#*****#

[AXIS_0]
TYPE           =      ANGULAR
MAX_VELOCITY   =      30
MAX_ACCELERATION =    200
BACKLASH       =      0.000
MIN_LIMIT      =     -150
MAX_LIMIT      =      150
FERROR        =      2000
MIN_FERROR     =      0.200

HOME           =      0.000
HOME_SEQUENCE  =      0
HOME_SEARCH_VEL =    -15
HOME_LATCH_VEL =      5
HOME_USE_INDEX =     NO
HOME_IGNORE_LIMITS = NO
HOME_OFFSET    =    -150
HOME_IS_SHARED =      0

SCALE          =      40
#STEPGEN_MAXVEL =    31.3
STEPGEN_MAXACCEL =    200
STEP_LENGTH    =   390000
STEP_SPACE     =   390000
DIR_HOLD       =   190000
DIR_SETUP      =   190000

#*****#
#  SHOULDER  #
#*****#

[AXIS_1]
TYPE           =      ANGULAR
MAX_VELOCITY   =      30
MAX_ACCELERATION =    20
BACKLASH       =      0.000
MIN_LIMIT      =     -100
MAX_LIMIT      =      30
FERROR        =      2.000
MIN_FERROR     =      0.200

HOME           =     -100
HOME_SEQUENCE  =      0
HOME_SEARCH_VEL =    -15
HOME_LATCH_VEL =      5
HOME_USE_INDEX =     NO
HOME_IGNORE_LIMITS = NO
HOME_OFFSET    =    -100
HOME_IS_SHARED =      0

SCALE          =      40
STEPGEN_MAXVEL =    31.3
STEPGEN_MAXACCEL =    21
STEP_LENGTH    =   390000
STEP_SPACE     =   390000

```

```

DIR_HOLD          =      190000
DIR_SETUP        =      190000

#####
#   ELBOW   #
#####

[AXIS_2]
TYPE              =      ANGULAR
MAX_VELOCITY     =      30
MAX_ACCELERATION =      20
BACKLASH         =      0.000
MIN_LIMIT        =      -0.01
MAX_LIMIT        =      90.01
FERROR          =      2.000
MIN_FERROR       =      0.200

HOME              =      90
HOME_SEQUENCE    =      0
HOME_SEARCH_VEL  =      15
HOME_LATCH_VEL   =      -5
HOME_USE_INDEX   =      NO
HOME_IGNORE_LIMITS =      NO
HOME_OFFSET      =      90
HOME_IS_SHARED   =      0

SCALE            =      40
STEPGEN_MAXVEL   =      31.3
STEPGEN_MAXACCEL =      21
STEP_LENGTH      =      390000
STEP_SPACE       =      390000
DIR_HOLD         =      190000
DIR_SETUP        =      190000

#####
#   PITCH   #
#####

[AXIS_3]
TYPE              =      ANGULAR
MAX_VELOCITY     =      92
MAX_ACCELERATION =      20
BACKLASH         =      0.000
MIN_LIMIT        =      -180
MAX_LIMIT        =      0
FERROR          =      2.000
MIN_FERROR       =      0.200

HOME              =      -180
HOME_SEQUENCE    =      0
HOME_SEARCH_VEL  =      -15
HOME_LATCH_VEL   =      5
HOME_USE_INDEX   =      NO
HOME_IGNORE_LIMITS =      YES
HOME_OFFSET      =      -180
HOME_IS_SHARED   =      0

```

```

SCALE                =      26.66666
STEPGEN_MAXVEL      =      45
STEPGEN_MAXACCEL    =      21
STEP_LENGTH         =     390000
STEP_SPACE          =     390000
DIR_HOLD            =     190000
DIR_SETUP           =     190000

```

```

#*****#
#   ROLL   #
#*****#

```

```

[AXIS_4]
TYPE                =      ANGULAR
MAX_VELOCITY        =      92
MAX_ACCELERATION    =      20
BACKLASH            =      0.000
MIN_LIMIT           =     -180
MAX_LIMIT           =      180
FERROR              =      2.0000
MIN_FERROR          =      0.200

```

```

HOME                =      0
HOME_SEQUENCE       =      1
HOME_SEARCH_VEL     =     -15
HOME_LATCH_VEL     =      15
HOME_USE_INDEX      =      NO
HOME_IGNORE_LIMITS =      NO
HOME_OFFSET         =     -65
HOME_IS_SHARED      =      0

```

```

SCALE                =      26.6666
STEPGEN_MAXVEL      =      45
STEPGEN_MAXACCEL    =      21
STEP_LENGTH         =     390000
STEP_SPACE          =     390000
DIR_HOLD            =     190000
DIR_SETUP           =     190000

```

```

#####
# section for main IO controller parameters
#####
[EMCIO]

```

```

EMCIO      =      io
CYCLE_TIME =      0.100
TOOL_TABLE =      rm501.tbl

```

rm501.c

```

/*****
* Description: rm501kins.c
* Kinematics for RM-501 Mitsubishi Movemaster Robot
*
* Author:
* License: GPL Version 2

```

```

* System: Linux
*
*****/

#include "rtapi_math.h"
#include "posemath.h"
#include "rm501kins.h"
#include "kinematics.h"
#include "rtapi.h"
#include "rtapi_app.h"
#include "hal.h"

#define sq(x) ((x)*(x))

/* RM-501 Mitsubishi Movemaster DH parameters*/
int d1 = 250;
int a2 = 220;
int a3 = 160;
int d5 = 214.6;

/*****
                                Forward Kinematics
*****/
int kinematicsForward(const double * joint,
                    EmcPose * world,
                    const KINEMATICS_FORWARD_FLAGS * fflags,
                    KINEMATICS_INVERSE_FLAGS * iflags)
{

    double th1 = joint[0] * PM_PI / 180.0;
    double th2 = joint[1] * PM_PI / 180.0;
    double th3 = joint[2] * PM_PI / 180.0;
    double th4 = joint[3] * PM_PI / 180.0;
    double th5 = joint[4] * PM_PI / 180.0;

    double C234 = cos(th2+th3+th4);
    double S234 = sin(th2+th3+th4);
    double C23  = cos(th2+th3);
    double S23  = sin(th2+th3);
    double C1   = cos(th1);
    double S1   = sin(th1);
    double C2   = cos(th2);
    double S2   = sin(th2);
    double C3   = cos(th3);
    double S3   = sin(th3);
    double C5   = cos(th5);
    double S5   = sin(th5);
    PmHomogeneous hom;

```



```

PmPose worldPose;
PmRpy rpy;

/* First column of rotation matrix.*/
hom.rot.x.x = C5*C1*C234 + S1*S5;
hom.rot.x.y = C5*S1*C234 - C1*S5;
hom.rot.x.z = C5*S234;

/* Second column of rotation matrix.*/
hom.rot.y.x = -S5*C1*C234 + S1*C5;
hom.rot.y.y = -S5*S1*C234 - C1*C5;
hom.rot.y.z = -S5*S234;

/* Third column of rotation matrix.*/
hom.rot.z.x = C1*S234;
hom.rot.z.y = S1*S234;
hom.rot.z.z = -C234;

/* Position vector. */
hom.tran.x = d5*C1*S234 + a3*C1*C23 + a2*C1*C2;
hom.tran.y = d5*S1*S234 + a3*S1*C23 + a2*S1*C2;
hom.tran.z = d1 - d5*C234 + a3*S23 + a2*S2;

/*****
                        Flags for Inverse Kinematics
*****/

/*Helpfull Variables*/

double k2      = C1*hom.tran.x + S1*hom.tran.y - d5*S234;
double ith1    = atan2(hom.tran.y, hom.tran.x);
double ith2    = atan2(sqrt(sq(C3*a3+a2)+sq(a3*S3)-sq(k2)), k2) -
atan2(a3*S3 , C3*a3+a2);
double ith5    = atan2(S1*hom.rot.x.x - C1*hom.rot.x.y ,
S1*hom.rot.y.x - C1*hom.rot.y.y);

/* reset flags */
*iflags = 0;

/* Set Waist flag if necessary */
if (fabs(th1 - ith1) < FLAG_FUZZ)
{
*iflags |= RM501_WAIST;
}
/* Set Shoulder flag if necessary */
if (fabs(th2 - ith2) < FLAG_FUZZ)
{
*iflags |= RM501_SHOULDER;
}

```

```

    /* Set Flip flag if necessary */
    if (fabs(th5 - ith5) < FLAG_FUZZ)
    {
        *iflags |= RM501_FLIP;
    }

    /*****

    /* convert hom.rot to world->quat */
    pmHomPoseConvert(&hom, &worldPose);
    pmQuatRpyConvert(&worldPose.rot, &rpy);
    world->tran = worldPose.tran;
    world->a = rpy.r * 180.0/PM_PI;
    world->b = rpy.p * 180.0/PM_PI;
    world->c = rpy.y * 180.0/PM_PI;

    /* return 0 and exit */
    return 0;
}

/*****
                                     Inverse Kinematics
*****/
int kinematicsInverse(const EmcPose * world,
                    double * joint,
                    const KINEMATICS_INVERSE_FLAGS * iflags,
                    KINEMATICS_FORWARD_FLAGS * fflags)
{
    PmHomogeneous hom;
    PmPose worldPose;
    PmRpy rpy;

    double th1, c1, s1;
    double th2;
    double th3, c3, s3;
    double th4;
    double th5, c5, s5;
    double th234, c234, s234;

    /* reset flags */
    *fflags = 0;

    /* convert pose to hom */
    worldPose.tran = world->tran;
    rpy.r = world->a*PM_PI/180.0;
    rpy.p = world->b*PM_PI/180.0;

```

```

rpy.y = world->c*PM_PI/180.0;
pmRpyQuatConvert(&rpy, &worldPose.rot);
pmPoseHomConvert(&worldPose, &hom);

/*****
    Waist-joint[0]-link[1]
    *****/
/* Atan2(0,0) is undefined so we have to take some precautions*/
/* The below lines means: If they asked you x=0 y=0 go home*/

if (hom.tran.y == 0 && hom.tran.x == 0) {
    th1=0;
} else {
    if (*iflags & RM501_WAIST){
        th1 = atan2(hom.tran.y, hom.tran.x);
    }else{
        th1 = atan2(hom.tran.y, hom.tran.x);
        if ( joint[0]< 0){
            th1=th1-PM_PI;
        }else{
            th1=th1+PM_PI;
        }
    }
}

/* Compute cos sin for later calcs*/
c1 = cos(th1);
s1 = sin(th1);

/* Calculate terms for future use */
s234 = c1*hom.rot.z.x + s1*hom.rot.z.y;
c234 = -hom.rot.z.z;
th234 = atan2(s234, c234);

/*****
    Elbow-joint[2]-link[3]
    *****/
double k1 = hom.tran.z - d1 + d5*c234;
double k2 = c1*hom.tran.x + s1*hom.tran.y - d5*s234;

c3 = (sq(k1) + sq(k2) - sq(a3) - sq(a2)) / (2*a2*a3);
if (c3 > 1) c3 = 1;
if (c3 < -1) c3 = -1;
s3 = sqrt(1 - sq(c3));

th3 = atan2(s3, c3);

```

```

/*****
    Shoulder-joint[1]-link[2]
*****/

double Sqrt = sqrt(sq(c3*a3+a2)+sq(a3*s3)-sq(k2));
/* -- FIXME -- flip the condition if necessary */
if (*iflags & RM501_SHOULDER) {
    th2 = atan2(-Sqrt,k2)-atan2(a3*s3,c3*a3+a2);
} else {
    th2 = atan2(Sqrt,k2)-atan2(a3*s3,c3*a3+a2);
}
/*****
    Pitch-joint[3]-link[4]
*****/
th4=th234-th3-th2;

/*****
    Roll-joint[4]-link[5]
*****/
c5 = s1*hom.rot.y.x-c1*hom.rot.y.y;
s5 = sqrt(1 - sq(s5));
if (*iflags & RM501_FLIP) {
    th5 = atan2(s5,c5);
} else {
    th5 = atan2(-s5,c5);
}

/* copy out */
joint[0] = th1*180/PM_PI;
joint[1] = -th2*180/PM_PI;
joint[2] = th3*180/PM_PI;
joint[3] = -th4*180/PM_PI;
joint[4] = th5*180/PM_PI;

    return 0;
}

int kinematicsHome(EmcPose * world,
                  double * joint,
                  KINEMATICS_FORWARD_FLAGS * fflags,
                  KINEMATICS_INVERSE_FLAGS * iflags)
{
    /* use joints, set world */
    return kinematicsForward(joint, world, fflags, iflags);
}

KINEMATICS_TYPE kinematicsType()
{

```

```

    return KINEMATICS_FORWARD_ONLY;
    // return KINEMATICS_BOTH;
}

EXPORT_SYMBOL(kinematicsType);
EXPORT_SYMBOL(kinematicsForward);
EXPORT_SYMBOL(kinematicsInverse);
EXPORT_SYMBOL(kinematicsHome);

MODULE_LICENSE("GPL");

int comp_id;
int rtapi_app_main(void) {
    comp_id = hal_init("rm501kins");
    if(comp_id < 0) return comp_id;

    hal_ready(comp_id);
    return 0;
}

void rtapi_app_exit(void) { hal_exit(comp_id); }

```

rm501.h

```

/*****
* Description: RM501kins.h
* Kinematics for a RM501 robot
* Author:
* License: GPL Version 2
* System: Linux
*****/
* This is the header file to accompany RM501kins.c.
*****/
*/
#ifndef RM501_H
#define RM501_H

#define FLAG_FUZZ            0.000001
#define RM501_WAIST         0x01
#define RM501_SHOULDER     0x02
#define RM501_ELBOW        0x03
#define RM501_FLIP          0x04
#endif /* RM501_H */

```