



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΟΥ»

Μεταπτυχιακή Εργασία

**ΥΛΟΠΟΙΗΣΗ ΕΛΕΓΚΤΩΝ ΕΜΠΕΔΗΣΗΣ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΣΥΣΤΗΜΑΤΟΣ
ΣΥΝΕΡΓΑΖΟΜΕΝΩΝ ΡΟΜΠΟΤ**

Χρυσάφης Γεώργιος

Επιβλέπων Καθηγητής: Κυριακόπουλος Κωνσταντίνος

ΑΘΗΝΑ 2021

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη ενός ολοκληρωμένου συστήματος, το οποίο αποτελείται από 3 ρομπότ, με στόχο τη μεταφορά ογκώδους αντικειμένου σε ένα εσωτερικό περιβάλλον εργασίας. Πιο αναλυτικά, το σύστημα θα αποτελείται από ένα ρομπότ - οδηγό και δύο ρομπότ - ακολούθους, οι οποίοι θα πρέπει να προσαρμόζονται πλήρως στις κινήσεις του οδηγού. Το ρομπότ - οδηγός προορίζεται να δρα και να κινείται αυτόνομα εντός του χώρου εργασίας μέσω ενός συστήματος ελέγχου τύπου PID σε μια προσπάθεια να προσομοιαστεί η ανθρώπινη παρουσία στο χώρο. Από την άλλη, οι δύο ακόλουθοι θα πρέπει να συμμορφώνονται στις κινήσεις του οδηγού και να ακολουθούν την τροχιά του, μέσω ενός κατάλληλου βρόχου ελέγχου δύναμης (και πιο συγκεκριμένα ελέγχου εμπέδησης) που υλοποιήθηκε. Και οι 3 ρομποτικοί πράκτορες μεταφέρουν το φορτίο - αντικείμενο, τη στιγμή που τα τελικά εργαλεία δράσης τους είναι άκαμπτα προσαρτημένα σε αυτό. Κατά αυτόν τον τρόπο, το σύστημα πλέον αποτελεί μία κλειστή κινηματική αλυσίδα με άμεση συνέπεια να υπεισέρχονται επιπλέον κινηματικοί περιορισμοί σε αυτό.

Όσον αφορά το κομμάτι του λογισμικού, το σύστημα στο σύνολό του αναπτύσσεται στο Robot Operating System (ROS), ενός πλαισίου ανοιχτού κώδικα ιδανικού για την ανάπτυξη ρομποτικών εφαρμογών. Το ROS είναι μια συλλογή πακέτων λογισμικού που στο χαμηλότερο επίπεδο προσφέρουν μια διεπαφή μετάδοσης μηνυμάτων μεταξύ διαφορετικών διεργασιών. Επιπλέον, παρέχει ένα σύνολο εργαλείων για την περιγραφή και τη μοντελοποίηση των ρομποτικών κατασκευών, έτσι ώστε να γίνει κατανοητή από το υπόλοιπο οικοσύστημα. Αυτό το πρότυπο για την περιγραφή κινηματικών αλυσίδων ονομάζεται URDF (Unified Robot Description Format) και αποτελείται από αρχεία σε XML κωδικοποίηση με τη δομή και τις φυσικές ιδιότητες του ρομπότ. Τέλος, το ROS παρέχει κατάλληλες διεπαφές επικοινωνίας του υλικού με τους υλοποιημένους ελεγκτές και έτσι δύναται να σταλούν, καθώς και να ληφθούν εντολές από το ρομπότ. Τέλος, για την υποστήριξη του ROS χρησιμοποιείται το λειτουργικό σύστημα Linux και η γλώσσα προγραμματισμού C++.

Εν κατακλείδι, αξίζει να αναφερθεί ότι το προαναφερθέν σενάριο υλοποιείται σε ένα τρισδιάστατο περιβάλλον γραφικής προσομοίωσης, το Gazebo. Το Gazebo αποτελεί το πιο διαδομένο ανοιχτού κώδικα περιβάλλον προσομοίωσης, προσανατολισμένο ειδικά για ρομποτικές εφαρμογές, ενώ περιλαμβάνει ενσωματωμένη μηχανή γραφικών.

Abstract

This diploma thesis deals with the development of an integrated system, which consists of 3 robots, with the aim of transporting a large rigid object in an internal workspace environment. More specifically, the system consists of a robot - leader and two robots - followers, which should be capable of being fully adapted to the movements of the leader. The driver robot is intended to act and move autonomously within the workplace through a PID control scheme in an attempt to simulate a human presence. On the other hand, the two followers must comply with the driver's movements and follow his trajectory, through a suitable force control scheme (and more specifically impedance control) which is implemented. All 3 robotic agents carry a single payload - object and their end effector tools are rigidly attached to it. In this way, the system is considered a closed kinematic chain with the consequence of additional kinematic constraints to its movement.

As far as the software part is concerned, the complete system is implemented on Robot Operating System (ROS), an open source framework ideal for developing robotic applications. ROS is a collection of software frameworks and packages, which at the lowest level offer a message passing interface that provides inter - process communication. Moreover, ROS provides a set of tools for describing and modeling the robotic agents, so that it can communicate and be understood by the rest of the ecosystem. This format for describing kinematic chains is called URDF (Unified Robot Description Format), which consists of XML documents with the structure and physical properties of the robot. Finally, ROS provides appropriate hardware interfaces in conjunction with built-in or custom made ROS controllers to send and receive commands to the robot. Finally, the operating system and the programming language that are used are Linux and C++ respectively.

In conclusion, it is worth mentioning that the above scenario is simulated on a 3D graphical simulation environment, Gazebo. Gazebo is the most widely used open source simulation environment, specifically oriented to robotic applications with a built-in graphics and physics engine.

Περιεχόμενα

Περίληψη	3
Abstract.....	5
Περιεχόμενα.....	7
Κατάλογος Σχημάτων	9
<i>Κεφάλαιο 1: Εισαγωγή</i>	11
1.1: Σκοπός Εργασίας.....	11
1.2: Βιβλιογραφική Ανασκόπηση.....	13
1.3: Συνεισφορά Της Εργασίας	20
1.4: Δομή Εργασίας.....	21
<i>Κεφάλαιο 2: Εργαλεία Λογισμικού</i>	23
2.1: Λειτουργικό Σύστημα Linux	23
2.2: Γνωριμία Με Το ROS	24
2.2.1: Οργάνωση του οικοσυστήματος	25
2.2.2: Το λογισμικό RViz.....	28
2.3: Γνωριμία Με Το Gazebo.....	29
2.3.1: Προσομοίωση ενός μοντέλου	30
<i>Κεφάλαιο 3: Δυναμική & Έλεγχος Ρομποτικών Διατάξεων</i>	35
3.1: Διαφορική Κινηματική.....	35
3.2: Δυναμικό Μοντέλο.....	39
3.3: Δυνάμεις Και Ροπές Άκαμπτου Σώματος.....	42
3.4: PID Έλεγχος.....	45
3.5: Έλεγχος Εμπέδησης	47
<i>Κεφάλαιο 4: Ελεγκτής Εμπέδησης</i>	55
4.1: Παρουσίαση Του ROS Control.....	55
4.2: Η Βιβλιοθήκη Orocos KDL.....	60
4.3: Υλοποίηση Του Ελεγκτή Εμπέδησης	61
<i>Κεφάλαιο 5: Ρομποτικοί Βραχίονες</i>	69
5.1: Ο Βραχίονας Mitsubishi PA10	69
5.2: Ο Βραχίονας Panda	70

5.2.1: Προσομοίωση του Panda στο Gazebo	71
Κεφάλαιο 6: Προσομοίωση Και Αποτελέσματα.....	73
6.1: Λειτουργία Του Ελεγκτή Εμπέδησης	73
6.2: Προσομοίωση Συστήματος 3 Συνεργατικών RA10.....	77
6.3: Προσομοίωση Συστήματος 3 Συνεργατικών Panda	81
Κεφάλαιο 7: Γενικά Συμπεράσματα	85
Βιβλιογραφία.....	87

Κατάλογος Σχημάτων

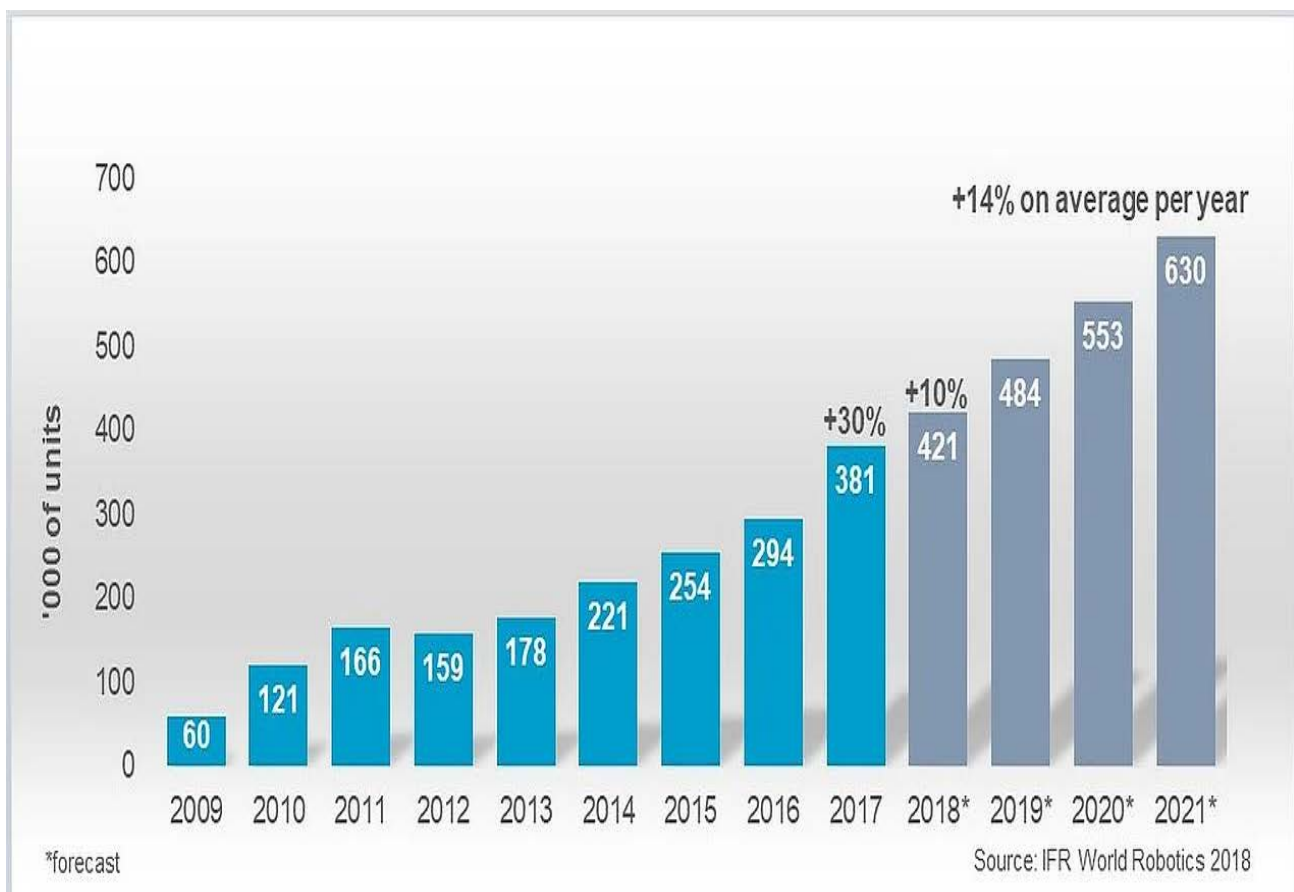
Σχήμα 1.01: Ετήσια προσφορά βιομηχανικών ρομπότ (IFR World Robotics 2018)	11
Σχήμα 1.02: Η γέννηση του Unimate, του πρώτου βιομηχανικού ρομπότ (1962).....	13
Σχήμα 1.03: Η βασική δομή του box pushing problem [4].....	14
Σχήμα 1.04: Συνεργασία ανθρώπου - ρομπότ για τη συναρμολόγηση ηλεκτρονικών εξαρτημάτων	16
Σχήμα 1.05: Σχήμα ταξινόμησης της HRI της αυτονομίας συναρτήσεως της εγγύτητας [11]	17
Σχήμα 1.06: Εφαρμογή ελέγχου εμπέδησης για τον καθαρισμό επίπεδης επιφάνειας ...	18
Σχήμα 2.01: Λογότυπο του ROS	24
Σχήμα 2.02: Η δομή του ROS	25
Σχήμα 2.03: Το ROS ως γράφος	26
Σχήμα 2.04: Η οργάνωση αρχείων του ROS	27
Σχήμα 2.05: Το λογισμικό RViz	28
Σχήμα 2.06: Το λογότυπο του Gazebo	29
Σχήμα 2.07: Απεικόνιση ενός URDF συνδέσμου	32
Σχήμα 2.08: Απεικόνιση μιας URDF άρθρωσης	33
Σχήμα 3.01: Αντιστοίχιση της ταχύτητας των αρθρώσεων με την ταχύτητα του λειτουργικού χώρου για πλεονάζοντες βαθμούς ελευθερίας	37
Σχήμα 3.02: Η θέση του κέντρου μάζας και οι δυνάμεις ενός ενδιάμεσου συνδέσμου μιας κινηματικής αλυσίδας.....	42
Σχήμα 3.03: Δομικό διάγραμμα PID ελεγκτή.....	45
Σχήμα 3.04: Επίδραση των κερδών στην απόκριση του συστήματος	46
Σχήμα 3.05: Είσοδος - έξοδος του ελεγκτή εμπέδησης.....	47
Σχήμα 3.06: Μηχανικό ανάλογο του συστήματος σε επαφή με το ελαστικό εξωτερικό περιβάλλον E.....	48
Σχήμα 3.07: Δομικό διάγραμμα του ελέγχου εμπέδησης (α).....	49
Σχήμα 3.08: Δομικό διάγραμμα του ελέγχου εμπέδησης (β).....	51
Σχήμα 3.09: Συμπεριφορά του ρομπότ σε σχέση με το εξωτερικό περιβάλλον	52
Σχήμα 4.01: Απλουστευμένη δομή του ROS Control.....	55
Σχήμα 4.02: Μηχανή πεπερασμένων καταστάσεων του controller_manager	57

Σχήμα 4.03: Αρχιτεκτονική του ROS Control [16]	58
Σχήμα 4.04: Η σειρά ενεργειών της controller_interface::Controller	59
Σχήμα 4.05: Βρόχος πραγματικού χρόνου στο ROS	60
Σχήμα 4.06: Δομικό διάγραμμα της init()	62
Σχήμα 4.07: Δομικό διάγραμμα της starting()	63
Σχήμα 4.08: Δομικό διάγραμμα της update()	64
Σχήμα 4.09: Δομικό διάγραμμα της stopping().....	66
Σχήμα 4.10: Δομικό διάγραμμα του ελεγκτή	67
Σχήμα 5.01: Ο βραχίονας PA10 στο γραφικό περιβάλλον του Gazebo.....	69
Σχήμα 5.02: Ο βραχίονας Panda στο γραφικό περιβάλλον του Gazebo	71
Σχήμα 6.01: Προσομοίωση του PA10 να χειρίζεται άκαμπτο αντικείμενο.....	73
Σχήμα 6.02: Προσομοίωση του Panda να χειρίζεται άκαμπτο αντικείμενο	74
Σχήμα 6.03: Αντιστάθμιση των δυνάμεων και ροπών του τερματικού εργαλείου δράσης που οφείλονται στα αδρανειακά χαρακτηριστικά.....	75
Σχήμα 6.04: Σφάλμα θέσης του τερματικού εργαλείου δράσης.....	75
Σχήμα 6.05: Σφάλμα ταχύτητας του τερματικού εργαλείου δράσης.....	76
Σχήμα 6.06: Προσομοίωση του συστήματος των PA10 στο Gazebo	77
Σχήμα 6.07: Διεπαφή χρήστη για την αυτόνομη κίνηση του βραχίονα οδηγού.....	78
Σχήμα 6.08: Αρχιτεκτονική της προσομοίωσης στο πλαίσιο του ROS Control	79
Σχήμα 6.09: Κινηματική δομή ολόκληρης της διάταξης των PA10	80
Σχήμα 6.10: Προσομοίωση του συστήματος των Panda στο Gazebo.....	81
Σχήμα 6.11: Οι κόμβοι του εν λειτουργία συστήματος σε μορφή γράφου.....	82
Σχήμα 6.12: Κινηματική δομή ολόκληρης της διάταξης των Panda.....	83

Κεφάλαιο 1: Εισαγωγή

1.1: Σκοπός Εργασίας

Μέχρι πρότινος, το πιο διαδεδομένο είδος ρομπότ που χρησιμοποιούνταν σε παγκόσμια κλίμακα ήταν οι μεμονωμένοι βιομηχανικού τύπου βραχίονες που λειτουργούσαν σε αυτοματοποιημένες γραμμές παραγωγής και σε σαφώς καθορισμένα και προστατευμένα περιβάλλοντα εργασίας χωρίς εμπόδια. Τα κύρια πλεονεκτήματά τους είναι η στιβαρή και αξιόπιστη κατασκευή, η υψηλή ακρίβεια και η επαναληπτικότητα στην κίνησή τους, με αποτέλεσμα να επιλέγονται σε επαναλαμβανόμενες και υψηλού φόρτου χειρωνακτικές εργασίες με σκοπό τη μείωση του κόστους παραγωγής και την αύξηση της παραγωγικότητας μιας γραμμής παραγωγής.



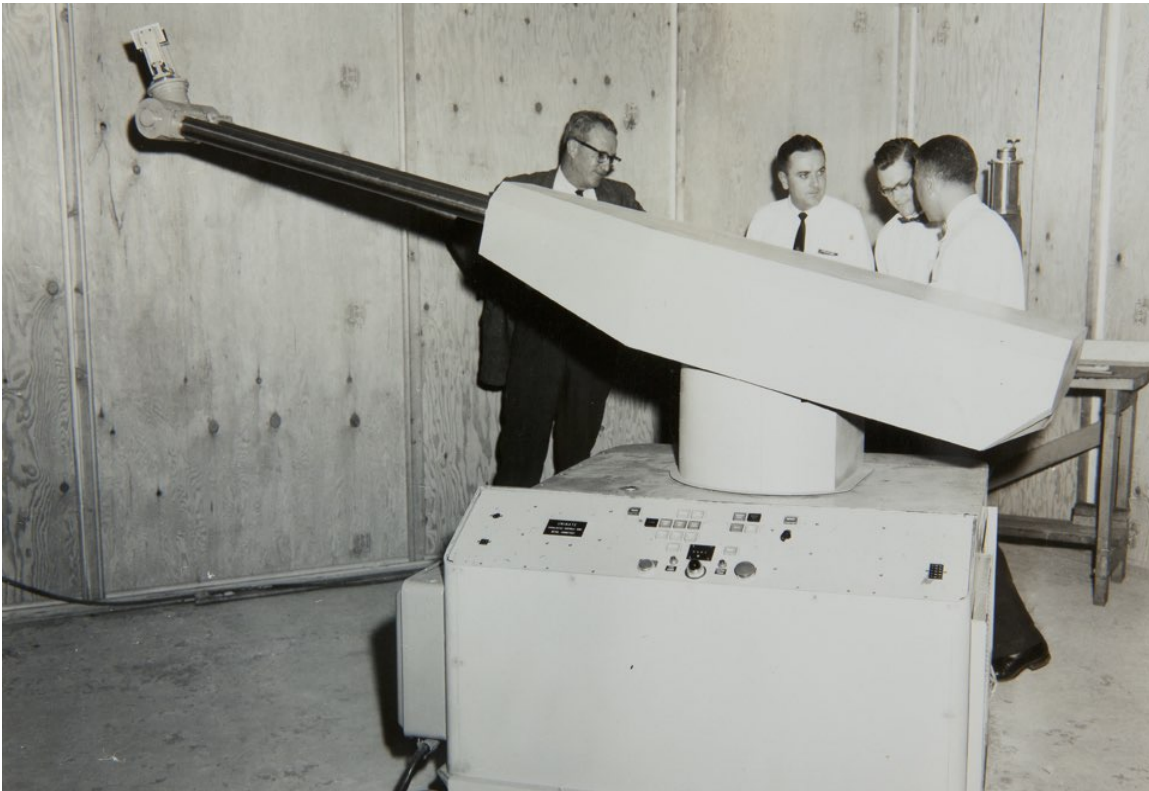
Σχήμα 1.01: Ετήσια προσφορά βιομηχανικών ρομπότ (IFR World Robotics 2018)

Τα τελευταία χρόνια, ωστόσο, αυτό τείνει να αλλάξει με την ερευνητική δραστηριότητα να προσανατολίζεται στο να εξωθήσει τα ρομπότ από έναν «αποστειρωμένο» χώρο εργασίας και να καταστήσει εφικτή την συνύπαρξή τους με ανθρώπους και άγνωστα εμπόδια σε ένα πλήρως λειτουργικό βιομηχανικό περιβάλλον. Για παράδειγμα, έχει καταβληθεί προσπάθεια για τη δημιουργία συστημάτων πολλαπλών κινητών ρομπότ που εμφανίζουν συλλογική και συνεργατική συμπεριφορά. Πληθώρα προβλημάτων και διεργασιών είναι αδύνατο να επιλυθούν με ένα μοναδικό ρομπότ, είτε εξαιτίας φυσικών περιορισμών είτε / και λόγω περιορισμένων πόρων ή πληροφορίας. Αυτοί οι λόγοι αναδεικνύουν τη σημασία μιας συνεργατικής συμπεριφοράς ορισμένων ρομποτικών πρακτόρων, όπου κάθε εταίρος συμβάλλει στην επίτευξη ενός τελικού στόχου. Επιπροσθέτως, η ανθρώπινη ασφάλεια είναι ύψιστης σημασίας υπό οποιαδήποτε συνθήκη - πόσο μάλλον όταν είναι υποχρεωτικό για τα ρομπότ να μοιράζονται κοινό χώρο εργασίας με τον άνθρωπο. Παράλληλα, η αλληλεπίδραση ανθρώπου - ρομπότ (human robot interaction) αποτελεί αντικείμενο έρευνας που στοχεύει στον συνδυασμό των δεξιοτήτων του ανθρώπου και των χαρακτηριστικών ενός ρομπότ, τα οποία μάλιστα αλληλοσυμπληρώνονται ιδανικά.

Λαμβάνοντας υπόψιν τα παραπάνω, η παρούσα εργασία αποσκοπεί σε μία λύση του διάσημου προβλήματος της μεταφοράς ενός αντικειμένου από ένα αρχικό σημείο σε ένα τελικό σημείο - στόχο (box pushing problem) εντός ενός αυθαίρετου εσωτερικού περιβάλλοντος. Το εν λόγω πρόβλημα βρίσκει εφαρμογή σε μια πλειάδα πραγματικών εφαρμογών στη βιομηχανία όπως η μεταφορά αντικειμένων σε μια γραμμή παραγωγής ή η αποθήκευση - εναπόθεση αντικείμενων σε αποθήκη. Αν το αντικείμενο που πρόκειται να μεταφερθεί είναι ιδιαίτερα ογκώδες ή βαρύ, είναι αναγκαίο να χρησιμοποιηθούν παραπάνω του ενός ρομπότ. Στην προκειμένη περίπτωση, θα υλοποιηθεί ένα συνεργατικό σύστημα ανθρώπου - ρομπότ, το οποίο θα περιλαμβάνει 3 πράκτορες. Για λόγους προσομοίωσης, στο ρόλο του ανθρώπου θα τοποθετηθεί ένα ρομπότ - οδηγός, το οποίο, όμως, θα έχει αυτόνομη συμπεριφορά εξωγενώς ορισμένη μέσω κατάλληλου σχήματος ελέγχου. Απεναντίας, τα εναπομείναντα 2 ρομπότ θα πρέπει να συμπεριφέρονται ως ακόλουθοι μέσω ενός διαφορετικού συστήματος ελέγχου εμπέδησης. Κατά αυτόν τον τρόπο επιτυγχάνονται οι προδιαγραφές του προβλήματος, όπου δηλαδή το ρομπότ - ηγέτης είναι επιφορτισμένο με τον σχεδιασμό της τροχιάς κίνησης του αντικείμενου, ενώ στόχος των ακολούθων είναι να συμμορφώνονται σε αυτήν την τροχιά ενεργώντας πλήρως μεμονωμένα. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι τα ρομπότ - ακόλουθοι επιτυγχάνουν τέτοια συμπεριφορά μέσω του βρόχου ελέγχου χωρίς τη χρήση για παράδειγμα οπτικών αισθητήρων. Για την ακρίβεια, δεν γνωρίζουν καν την ύπαρξη των άλλων διατάξεων.

1.2: Βιβλιογραφική Ανασκόπηση

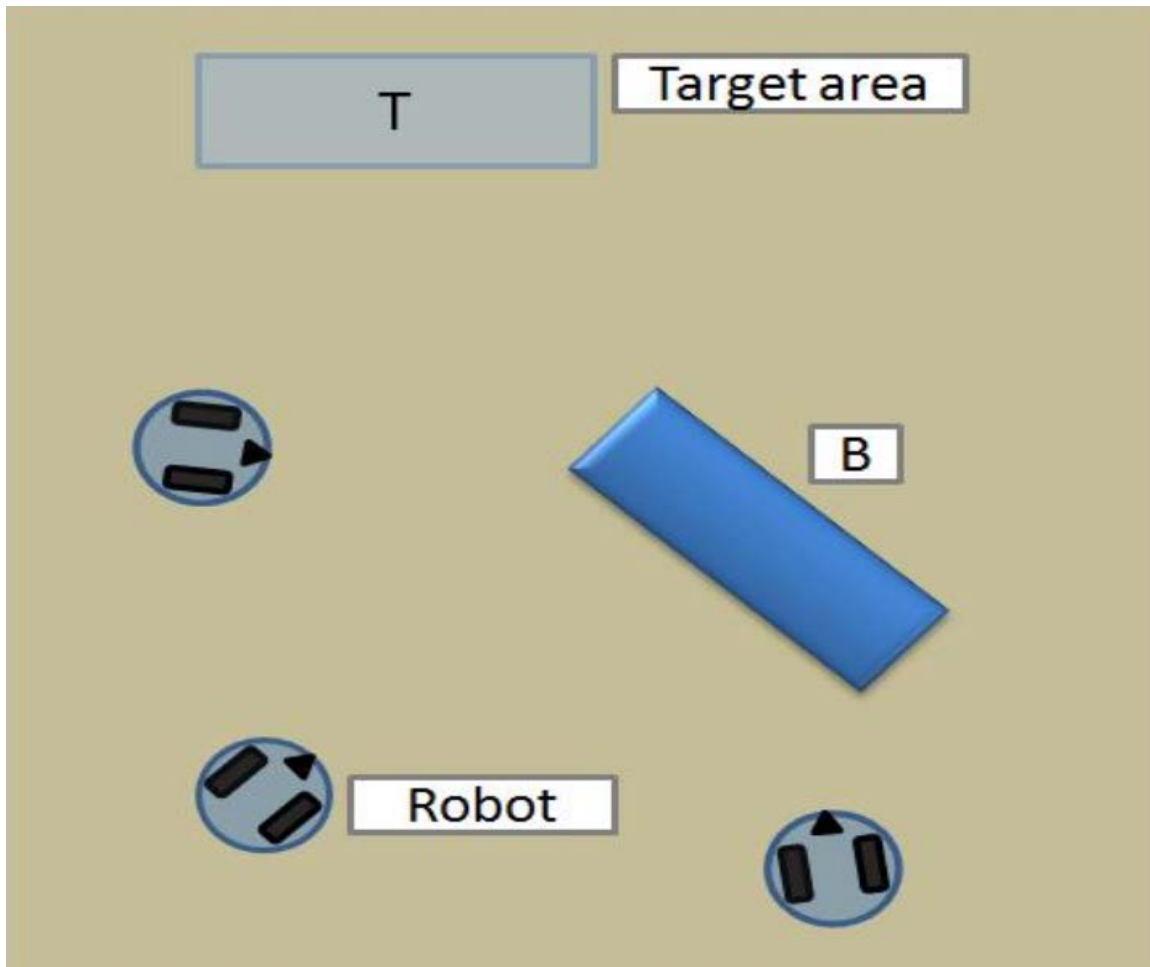
Στην ενότητα που έπεται, θα παρατεθούν κάποια περαιτέρω στοιχεία, πιο λεπτομερή, σχετικά με τη ρομποτική, αλλά και τα επιμέρους πεδία της που πραγματεύεται η εν λόγω εργασία. Αρχικά, η ετυμολογία του όρου προέρχεται από τον όρο “robota” που στις σλάβικες διαλέκτους αναφέρεται στην άμισθη και εξαναγκασμένη εργασία. Για πρώτη φορά, ο όρος εμφανίζεται το 1920 στο θεατρικό έργο του Karel Capek με τίτλο “Rossum’s Universal Robots”, ένα παραμύθι επιστημονικής φαντασίας κατά το οποίο μία αυτόματη κατασκευή στρέφεται εναντίον της ανθρωπότητας [1]. Πλέον, με τον όρο ρομπότ εννοείται μία κατασκευή που έχει τη δυνατότητα να αποκτάει πληροφορίες από το εξωτερικό περιβάλλον, να τις επεξεργάζεται ώστε να λάβει αποφάσεις και τέλος να περιλαμβάνει επενεργητές για την εκτέλεση κάποιας εργασίας.



Σχήμα 1.02: Η γέννηση του Unimate, του πρώτου βιομηχανικού ρομπότ (1962)

Όσον αφορά τα βιομηχανικά ρομπότ, αυτά έκαναν την εμφάνισή τους ως ξεχωριστές κατασκευές περίπου τη δεκαετία του 1960 [2] παράλληλα με την ανάπτυξη των συστημάτων σχεδιασμού με τη βοήθεια υπολογιστή (Computer Aided Design) και των συστημάτων κατασκευών με τη βοήθεια υπολογιστή

(Computer Aided Manufacturing). Όλες αυτές οι τεχνολογίες οδηγούν τη βιομηχανική αυτοματοποίηση σε μια εξελικτική διαδικασία, ο προορισμός της οποίας είναι ακόμα απροσδιόριστος. Ενδεικτικές εφαρμογές τέτοιων ρομπότ είναι το παλετάρισμα, η συναρμολόγηση αντικειμένων, η διαλογή και εναπόθεση αντικειμένων (pick and place), η τροφοδότηση γραμμών παραγωγής και ο έλεγχος ποιότητας.



Σχήμα 1.03: Η βασική δομή του box pushing problem [4]

Από τότε έχει σημειωθεί σημαντική πρόοδος στον τομέα με νέες λειτουργίες και δυνατότητες να προστίθενται συνεχώς, με άμεση συνέπεια ολοένα και να διευρύνεται το φάσμα εφαρμογών στις οποίες ένα ρομπότ μπορεί να λάβει μέρος. Μια από αυτές αποτελεί και η μεταφορά ενός αντικειμένου σε ένα τρισδιάστατο χώρο εργασίας από ένα αρχικό σε ένα τελικό σημείο (Box Pushing Problem), θέμα με το οποίο ασχολείται η συγκεκριμένη εργασία. Το πρόβλημα της ώθησης του αντικειμένου ορίστηκε αρχικά στο πλαίσιο του συστήματος πολλαπλών συνεργατικών ρομπότ [3], με την έρευνα να προσανατολίζεται στον καταμερισμό

των εργασιών και την συνεργασία των επιμέρους ρομποτικών συστημάτων. Ένα πρόβλημα αυτού του είδους έχει αναρίθμητες πρακτικές εφαρμογές. Οι πιο διαδεδομένες είναι η μεταφορά επικίνδυνων αντικειμένων, η αυτοματοποίηση σε ένα βιομηχανικό περιβάλλον, η μεταφορά πολύ ογκωδών αντικειμένων, η διάσωση τραυματιών κ.α..

Έκτοτε, έχουν προταθεί διάφορες στρατηγικές επίλυσης του προβλήματος, το οποίο όπως είναι εύκολα αντιληπτό περιλαμβάνει διάφορα απλούστερα προβλήματα, όπως ο σχεδιασμός τροχιάς, η αποφυγή εμποδίων, η ευρωστία κ.α.. Αρχικά, έχει μελετηθεί ενδελεχώς η επικοινωνία μεταξύ των ρομπότ του συστήματος. Ο Yamada αναφέρει ότι, ανεξάρτητα από τη διαμόρφωση του συστήματος (συγκεντρωτικό ή αποκεντρωμένο), οι ανάγκες επικοινωνίας δύνανται να παραλειφθούν και να αντικατασταθούν από συγκεκριμένους μηχανισμούς συμπεριφοράς που βασίζονται σε τοπικές πληροφορίες [5]. Ωστόσο, τέτοιου είδους στρατηγικές έχουν ως μειονέκτημα την απαίτηση ύπαρξης πλειάδας προκαθορισμένων συμπεριφορών για το χειρισμό νέων προκλήσεων και εμποδίων. Από την άλλη πλευρά, ο Muoz προτείνει τον τρόπο με τον οποίο μπορεί η επικοινωνία να βελτιώσει σημαντικά την απόδοση των συστημάτων πολλαπλών ρομπότ, επιδρώντας θετικά στον συντονισμό, τη συνεργασία και την αποφυγή συγκρούσεων [6]. Άλλες μελέτες αντιμετωπίζουν το πρόβλημα με τη γνωστή στρατηγική κλεισίματος αντικείμενου (Object Closure), κατά την οποία μια σειρά από ρομπότ περικλείουν το αντικείμενο με συνέπεια η θέση του αντικείμενου να μπορεί να ελεγχθεί με σημείο αναφοράς τη θέση του κάθε ρομπότ [7]. Μια ακόμα διαδεδομένη τεχνική, που μάλιστα παραλλαγή της οποίας ακολουθείται στη παρούσα εργασία, είναι αυτή του ωθόντα - παρατηρητή (Pusher - Watcher), όπου ένα ρομπότ καθορίζει την κίνηση του αντικείμενου και καθορίζει τη συμπεριφορά των υπολοίπων [2]. Επιπλέον, πιο πρόσφατες στρατηγικές κάνουν χρήση του μοντέλου που είναι γνωστό ως νοημοσύνη σμήνους (Swarm Intelligence), κατά το οποίο χρησιμοποιούνται ομοιογενή ρομπότ, τα οποία με τη σειρά τους βασίζονται σε αποκεντρωμένες και συλλογικές συμπεριφορές [9]. Τέλος, τα τελευταία χρόνια έχουν εμφανιστεί και τεχνικές που βασίζονται στη μηχανική μάθηση και την τεχνητή νοημοσύνη, οι οποίες όπως είναι λογικό απαιτούν υψηλή υπολογιστική ισχύ [10].

Όπως αναφέρθηκε παραπάνω, η αλληλεπίδραση ανθρώπου - ρομπότ (Human – Robot Interaction) αποτελεί επίσης ένα δημοφιλές αντικείμενο έρευνας τα τελευταία χρόνια, η ανάπτυξη του οποίου ξεκίνησε περίπου το 1990. Πιο συγκεκριμένα, η αλληλεπίδραση ανθρώπου - ρομπότ είναι ένα πεδίο μελέτης αφιερωμένο στην κατανόηση, το σχεδιασμό και την αξιολόγηση ρομποτικών συστημάτων για χρήση από ή με την παρουσία ανθρώπου [12]. Η αλληλεπίδραση, εξ ορισμού, απαιτεί επικοινωνία μεταξύ ρομπότ και ανθρώπου. Η επικοινωνία μεταξύ ανθρώπου και ρομπότ μπορεί να πάρει διάφορες μορφές, αν και αυτές οι μορφές επηρεάζονται σε μεγάλο βαθμό από το αν ο άνθρωπος και το ρομπότ βρίσκονται πολύ κοντά ο ένας στον άλλο ή όχι. Έτσι, η επικοινωνία και, επομένως, η αλληλεπίδραση μπορούν να χωριστούν σε δύο γενικές κατηγορίες:

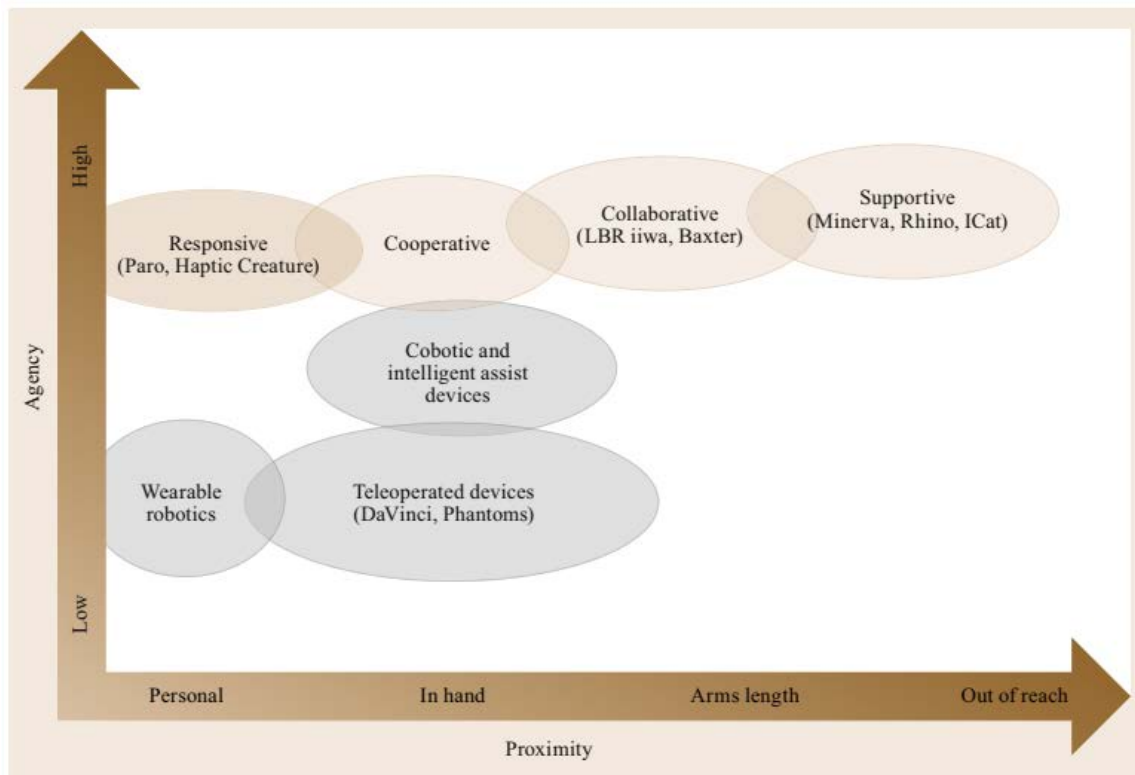
- Απομακρυσμένη αλληλεπίδραση, όπου ο άνθρωπος και το ρομπότ διαχωρίζονται χωρικά ή ακόμη και χρονικά.
- Κοντινή αλληλεπίδραση, όπου οι άνθρωποι και τα ρομπότ μπορεί να βρίσκονται στον ίδιο χώρο.

Μέσα σε αυτές τις γενικές κατηγορίες, είναι χρήσιμο να γίνει μια περαιτέρω κατηγοριοποίηση μεταξύ των εφαρμογών που απαιτούν κινητικότητα, φυσικούς χειρισμούς ή κοινωνική αλληλεπίδραση.



Σχήμα 1.04: Συνεργασία ανθρώπου - ρομπότ για τη συναρμολόγηση ηλεκτρονικών εξαρτημάτων

Η απομακρυσμένη αλληλεπίδραση με κινητά ρομπότ αναφέρεται συχνά ως τηλε-λειτουργία ή εποπτικός έλεγχος (teleoperation), ενώ η εξ αποστάσεως αλληλεπίδραση με φυσικό τρόπο ονομάζεται τηλε-χειρισμός. Από την άλλη, η κοινωνική αλληλεπίδραση περιλαμβάνει κοινωνικές, συναισθηματικές και γνωστικές πτυχές, δηλαδή με άλλα λόγια οι άνθρωποι και τα ρομπότ αλληλεπιδρούν ως εταίροι ή συνεργάτες. Σε γενικές γραμμές, οι κοινωνικές αλληλεπιδράσεις ανθρώπου - ρομπότ φαίνεται να λαμβάνουν χώρα ως επί το πλείστον σε κοντινή απόσταση και όχι απομακρυσμένα.



Σχήμα 1.05: Σχήμα ταξινόμησης της HRI της αυτονομίας συναρτήσει της εγγύτητας [11]

Το πρόβλημα της συνεργασίας διακρίνεται από αυξημένη πολυπλοκότητα και απαιτεί τον συνδυασμό πολλών επιμέρους ερευνητικών διεργασιών ώστε να επιτευχθεί. Το πιο κομβικό ζήτημα που πρέπει να λυθεί, χωρίς μάλιστα την παραμικρή έκπτωση, είναι αυτό της ασφάλειας του ανθρώπου, καθώς ο χώρος λειτουργίας του ρομπότ είναι πλέον προσβάσιμος. Είναι απαραίτητο λοιπόν να αναπτυχθούν αξιόπιστα συστήματα παθητικής ασφάλειας στα ρομπότ με ελαστικά και κινούμενα μέρη χαμηλής αδράνειας, παράλληλα με αλγοριθμικά συστήματα ενεργητικής ασφάλειας για αποφυγή συγκρούσεων.

Εν κατακλείδι, η αλληλεπίδραση ανθρώπου - ρομπότ είναι ένα πεδίο έρευνας και εφαρμογών που βρίσκεται σε διαρκή αύξηση. Ο τομέας περιλαμβάνει πολλά δυσεπίλυτα προβλήματα, ενώ πλέον χρησιμοποιείται σε πληθώρα περιστάσεων επιφέροντας αδιαμφησβήτητα θετικό κοινωνικό αντίκτυπο. Μόνο ένα υποσύνολο των εφαρμογών του είναι σε επιχειρήσεις έρευνας και διάσωσης, σε βοηθητικό ρόλο ατόμων με προβλήματα όρασης ή ηλικιωμένων, σε ιατρικές εφαρμογές, σε βοηθητικό ρόλο εντός του νοικοκυριού και τέλος σε εκπαιδευτικές εφαρμογές.

Η συνεργασία ανθρώπου-ρομπότ με φυσική επαφή συνεπάγεται ανάπτυξη δυνάμεων μεταξύ τους, οι οποίες επηρεάζουν κυρίως την ασφάλεια του ανθρώπου όσο και τη λειτουργία και κατασκευή του ρομπότ. Μέχρι πρότινος, τα βιομηχανικά ρομπότ βρισκόνταν σε προστατευμένο περιβάλλον χωρίς την ύπαρξη εμποδίων. Συνεπώς, χρησιμοποιούνταν με αποτελεσματικότητα αποκλειστικά έλεγχος θέσης για την κίνηση τους. Ωστόσο, όταν το ρομπότ εισαχθεί σε ένα άγνωστο ανελαστικό περιβάλλον, υπάρχουν άμεσες επιπτώσεις στην ευστάθεια του συστήματος ελέγχου. Για παράδειγμα, μια πιθανή δύναμη που θα ασκηθεί στο σώμα του ρομπότ, ανάλογα με το μέτρο και τη διεύθυνσή της, θα δημιουργήσει ένα σφάλμα θέσης το οποίο ο ελεγκτής θα προσπαθήσει να αντισταθμίσει με άγνωστα αποτελέσματα. Σε γενικές γραμμές, όσο μεγαλύτερη είναι η ακαμψία του περιβάλλοντος τόσο μεγαλύτερες δυνάμεις και ροπές θα αναπτυχθούν. Στα πιθανά αποτελέσματα ενός τέτοιου απευκταίου σεναρίου συγκαταλέγονται μεταξύ άλλων η καταστροφή της ρομποτικής κατασκευής μέχρι και ο τραυματισμός κάποιου ανθρώπου.



Σχήμα 1.06: Εφαρμογή ελέγχου εμπέδησης για τον καθαρισμό επίπεδης επιφάνειας

Τα τελευταία χρόνια, λοιπόν, ήταν υψίστης σημασίας να αναπτυχθούν ρομποτικά συστήματα που μπορούν να ενδίδουν σε εξωτερικές δυνάμεις και να προσαρμόζονται σε αυτές με απώτερο σκοπό την ασφαλή συμβίωση με τον άνθρωπο σε κοινό χώρο εργασίας. Ένας τρόπος να επιτευχθεί αυτό είναι να χρησιμοποιηθούν ρομποτικές κατασκευές που διαχειρίζονται τις εξωγενείς δυνάμεις με παθητικό τρόπο και αποτελούνται από ελαστικούς συνδέσμους και αρθρώσεις [13]. Από την άλλη, ο πιο διαδεδομένος τρόπος να επιτευχθεί η επιθυμητή λειτουργία είναι με τα ενεργητικά συστήματα, όπου μέσω του κατάλληλου συστήματος ελέγχου καθορίζονται οι επιθυμητές δυνάμεις επαφής του ρομπότ με το περιβάλλον. Σε αυτή την κατηγορία ελέγχων ανήκει ο έλεγχος δύναμης (Force Control), υποκατηγορίες του οποίου είναι ο έλεγχος εμπέδησης ή σύνθετης μηχανικής αντίστασης (Impedance Control), ο έλεγχος ενδοτικότητας (Admittance Control) και ο υβριδικός έλεγχος δύναμης - θέσης (Hybrid Control). Έτσι, η έρευνα για τον έλεγχο δύναμης στον τομέα της ρομποτικής έχει ανθίσει τις τρεις τελευταίες δεκαετίες. Όπως αναφέρθηκε, ένα τέτοιο ευρύ ενδιαφέρον έχει κίνητρο από τη γενική επιθυμία παροχής ρομποτικών συστημάτων με ενισχυμένες αισθητηριακές δυνατότητες. Ρομπότ με χρήση της δύναμης, της αφής, της απόστασης και της οπτικής ανατροφοδότησης αναμένεται να λειτουργούν αυτόνομα σε μη δομημένα περιβάλλοντα, διαφορετικά από τον τυπικό βιομηχανικό χώρο. Από τις πρώτες εργασίες για την τηλεχειρισμό, η χρήση ελέγχου με ανατροφοδότηση δύναμης σχεδιάστηκε για να βοηθήσει τον άνθρωπο - χειριστή στον απομακρυσμένο χειρισμό αντικειμένων μέσω του ρομπότ. Πιο πρόσφατα, συνεργατικά δάχτυλα ενός επιδέξου ρομποτικού χεριού έχουν αναπτυχθεί και ελέγχονται έτσι ώστε να περιορίζονται οι ανταλλασσόμενες δυνάμεις και να αποφεύγεται η συμπίεση ενός κοινού αντικειμένου. Ο έλεγχος δύναμης παίζει θεμελιώδη ρόλο και στην επίτευξη ισχυρής και ευέλικτης συμπεριφοράς των ρομποτικών συστημάτων σε ανοιχτό περιβάλλον, παρέχοντας ευφυή απάντηση σε απρόβλεπτες καταστάσεις και ενισχύοντας την αλληλεπίδραση ανθρώπου - ρομπότ.

Ο έλεγχος της φυσικής αλληλεπίδρασης μεταξύ ενός ρομπότ και του περιβάλλοντος είναι ζωτικής σημασίας για την επιτυχή εκτέλεση ορισμένων πρακτικών εργασιών όπου το τερματικό εργαλείο δράσης του ρομπότ πρέπει να χειριστεί ένα αντικείμενο ή να εκτελέσει κάποια λειτουργία σε μια επιφάνεια. Χαρακτηριστικά παραδείγματα σε βιομηχανικές εφαρμογές περιλαμβάνουν στίλβωση, κατεργασία ή συναρμολόγηση. Κατά την επαφή, το περιβάλλον μπορεί να θέσει περιορισμούς στην κίνηση που μπορεί να ακολουθήσει η αρπάγη του ρομπότ, οι οποίοι δηλώνονται ως κινηματικοί περιορισμοί. Αυτή η περίπτωση, που ουσιαστικά αντιστοιχεί στην επαφή με μια σκληρή επιφάνεια, γενικά αναφέρεται ως περιορισμένη κίνηση (constrained motion). Σε άλλη περίπτωση, η εργασία χαρακτηρίζεται από μια δυναμική αλληλεπίδραση μεταξύ του ρομπότ και του περιβάλλοντος που μπορεί να είναι αδρανειακή (όπως στην ώθηση ενός μπλοκ), διασκορπιστική (όπως η ολίσθηση σε επιφάνεια με τριβή) ή ελαστική (όπως στην ώθηση ενός ελαστικού αντικειμένου). Σε όλες αυτές τις περιπτώσεις, η χρήση μιας

αμιγούς στρατηγικής ελέγχου κίνησης (motion control) για την αλληλεπίδραση είναι επιρρεπής σε αποτυχία. Η επιτυχής εκτέλεση της εργασίας αλληλεπίδρασης με το περιβάλλον, κάνοντας χρήση του ελέγχου κίνησης, θα μπορούσε να επιτευχθεί μόνο αν η εργασία είχε προγραμματιστεί με απόλυτη ακρίβεια. Αυτό καθιστά υποχρεωτική τη γνώση ενός ακριβούς μοντέλου τόσο της ρομποτικής κατασκευής (κινηματική και δυναμική) όσο και του εξωτερικού περιβάλλοντος (γεωμετρία και μηχανικά χαρακτηριστικά). Το μοντέλο του ρομπότ είναι εφικτό να είναι γνωστό με επαρκή ακρίβεια, αλλά είναι αδύνατο να γίνει μια λεπτομερής περιγραφή του περιβάλλοντος.

Ο έλεγχος δύναμης υπερνικά τις προηγούμενες δυσκολίες. Στη συγκεκριμένη εργασία, θα χρησιμοποιηθεί ο έλεγχος εμπέδησης, οποίος χρησιμοποιείται κατά κόρον σε εφαρμογές αλληλεπίδρασης ανθρώπου - ρομπότ. Επί του πρακτέου, αυτό το σχήμα ελέγχου εξασφαλίζει τον έλεγχο της δυναμικής συμπεριφοράς του ρομπότ, ορίζοντας τη σχέση μεταξύ δύναμης και ταχύτητας παρά τον έλεγχο καθενός μεγέθους μεμονωμένα [14]. Με άλλα λόγια, ο έλεγχος εμπέδησης διαμορφώνει τη μηχανική σύνθετη αντίσταση του συστήματος, η οποία αποτελεί μία χαρτογράφηση των ταχυτήτων στις δυνάμεις. Αυτό το χαρακτηριστικό δίνει το πλεονέκτημα στο μηχανικό να ορίζει την επιθυμητή συμπεριφορά του ρομπότ ανάλογα με την εκάστοτε εφαρμογή και τις επιμέρους συνθήκες του περιβάλλοντος, ρυθμίζοντας με κατάλληλο τρόπο τις παραμέτρους - κέρδη του ελεγκτή. Τέλος, ο έλεγχος σύνθετης μηχανικής αντίστασης μπορεί να εφαρμοστεί είτε στο χώρο των αρθρώσεων, είτε στον καρτεσιανό χώρο του τελικού εργαλείου δράσης.

1.3: Συνεισφορά Της Εργασίας

Σε αυτή την ενότητα, θα παρουσιασθεί επιγραμματικά η συνεισφορά της παρούσας εργασίας. Γενικά, η συγκεκριμένη εργασία αποσκοπεί σε μία λύση του διάσημου προβλήματος της μεταφοράς ενός αντικειμένου από ένα αρχικό σημείο σε ένα τελικό σημείο - στόχο (box pushing problem) εντός ενός αυθαίρετου εσωτερικού περιβάλλοντος. Επίσης, προσφέρεται η υλοποίηση ενός ελεγκτή εμπέδησης πολλαπλών εισόδων - πολλαπλών εξόδων (MIMO) στο πλαίσιο λειτουργίας του ROS Control. Επίσης, υλοποιείται ένα σύστημα με πολλαπλούς ρομποτικούς βραχίονες οι οποίοι δρουν αποκεντρωμένα και ανεξάρτητα, ενώ παράλληλα αναπτύσσουν συνεργατική συμπεριφορά. Επιπλέον, η εργασία προσανατολίζεται προς την ανάπτυξη της φυσικής αλληλεπίδρασης των ρομποτικών διατάξεων με στόχο τη μεταφορά ογκωδών αντικειμένων.

1.4: Δομή Εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από 7 κεφάλαια. Στο πρώτο κεφάλαιο αναπτύσσεται ο σκοπός της εργασίας και παρουσιάζεται συνοπτικά το ολόένα και αυξανόμενο ζήτημα των συνεργατικών ρομπότ. Ακολουθεί η βιβλιογραφική ανασκόπηση, η οποία περιλαμβάνει αρχικά, την παρουσίαση του box pushing problem, την αλληλεπίδραση ανθρώπου - ρομπότ καθώς και στρατηγικών ελέγχου δύναμης, οι οποίες χρησιμοποιούνται κατά κόρον σε περιπτώσεις επαφής μεταξύ ενός ρομπότ και ενός αντικειμένου. Στο δεύτερο κεφάλαιο παρουσιάζονται εν συντομία τα εργαλεία λογισμικού στα οποία θα βασιστεί η εργασία. Γίνεται μια αναφορά στο λειτουργικό σύστημα Linux και περιγράφεται η δομή και η αρχιτεκτονική του πλαισίου ROS. Επίσης, παρουσιάζεται το Gazebo, ένα περιβάλλον προσομοίωσης για ρομποτικές εφαρμογές παντός είδους. Στο επόμενο κεφάλαιο θα αναλυθεί η δυναμική και ο έλεγχος των ρομποτικών διατάξεων. Αρχικά, θα παρουσιαστεί η διαφορική κινηματική μιας κινηματικής αλυσίδας, στη συνέχεια θα μοντελοποιηθεί το δυναμικό πρόβλημα και τέλος θα αναλυθεί η τεχνική του ελέγχου δύναμης. Στο τέταρτο κεφάλαιο θα αναλυθεί η υλοποίηση ενός ελεγκτή εμπέδησης κάνοντας χρήση του ROS Control και μιας βιβλιοθήκης της C++, της KDL. Στο πέμπτο κεφάλαιο θα παρουσιαστούν τα κατασκευαστικά και λειτουργικά χαρακτηριστικά των ρομποτικών διατάξεων και πιο συγκεκριμένα του PA10 και του Panda που θα χρησιμοποιηθούν στις μετέπειτα προσομοιώσεις και θα αναπαρασταθούν στο Gazebo. Στο έκτο κεφάλαιο θα αναλυθούν αρχικά η λειτουργία του ελεγκτή εμπέδησης και μετέπειτα δύο συστήματα συνεργατικών ρομπότ για τη μεταφορά ογκώδους αντικειμένου. Στο έβδομο κεφάλαιο αναφέρονται τα γενικά συμπεράσματα από την εκπόνηση της διπλωματικής εργασίας και προτείνονται μελλοντικές ενέργειες για την επέκτασή της.

Κεφάλαιο 2: Εργαλεία Λογισμικού

2.1: Λειτουργικό Σύστημα Linux

Το κεφάλαιο αυτό αναφέρεται στα εργαλεία που βασίστηκε η ανάπτυξη του λογισμικού του συστήματος. Το λειτουργικό σύστημα του υπολογιστή, ο οποίος εκτελεί το λογισμικό του συστήματος, είναι το λειτουργικό σύστημα Linux. Το Linux είναι ένα λειτουργικό σύστημα το οποίο χαρακτηρίζεται από λογισμικό ανοιχτού κώδικα και διανέμεται δωρεάν. Στο διαδίκτυο κυκλοφορούν πάρα πολλές διανομές, κάθε μία από τις οποίες ενδείκνυται για διαφορετικές χρήσεις και εξυπηρετεί διαφορετικές ανάγκες. Για παράδειγμα, άλλες προσανατολίζονται στη φιλικότητα προς τον χρήστη και στην απλότητα του συστήματος, ενώ άλλες δίνουν προτεραιότητα στη διαχείριση των υπολογιστικών πόρων όπως την επεξεργαστική ισχύ, τη μνήμη και τον αποθηκευτικό χώρο. Σε γενικές γραμμές, οι διανομές Linux προσφέρουν ελευθερία και κατάλληλα εργαλεία στους προγραμματιστές λογισμικού, σε αντιδιαστολή με το λειτουργικό σύστημα Windows, και για αυτό προτιμώνται κατά κόρον σε ανάλογες περιπτώσεις. Η διανομή που χρησιμοποιήθηκε στη συγκεκριμένη εργασία είναι τα Linux Mint Cinnamon, η οποία βασίζεται στη διανομή Ubuntu 18.04. Θεωρείται μια δημοφιλή επιλογή μεταξύ των χρηστών για το λόγο ότι δίνει έμφαση στη λειτουργικότητα, ώστε να παρέχεται καλύτερη υποστήριξη. Όπως είναι λογικό, πριν γίνει οποιαδήποτε προσπάθεια ανάπτυξης λογισμικού είναι απαραίτητη η εξοικείωση με το λειτουργικό σύστημα. Είναι απαραίτητες οι βασικές γνώσεις πλοήγησης και διαχείρισης του συστήματος αρχείων καθώς και εγκατάστασης λογισμικού και εφαρμογών από τον ενσωματωμένο διαχειριστή πακέτων. Όπως σε όλες τις διανομές, βασική προϋπόθεση θεωρείται και η χρήση του τερματικού μέσω του οποίου επιτυγχάνεται η άμεση αλληλεπίδραση με το λειτουργικό σύστημα.

Ένα ακόμα εργαλείο που θα χρησιμοποιηθεί στη συνέχεια είναι και το CMake. Συνήθως, μια εφαρμογή αποτελείται από τον πηγαίο κώδικα, ο οποίος εκτελείται σε συνδυασμό με διάφορες βιβλιοθήκες. Σε περίπτωση που χρειαστεί κάποια αλλαγή για παράδειγμα στις βιβλιοθήκες, τότε αυτές πρέπει πρώτα να μεταγλωττιστούν εκ νέου και έπειτα να ακολουθήσει ο πηγαίος κώδικας. Αυτό δημιουργεί αυξημένη πολυπλοκότητα και διάφορες εξαρτήσεις μεταξύ αρχείων, οι οποίες πρέπει να ληφθούν υπόψιν. Το CMake είναι ένα build εργαλείο που προσφέρει υψηλού επιπέδου λειτουργικότητα, αποτελεί λογισμικό ανοιχτού κώδικα και χρησιμοποιείται για τον έλεγχο της διαδικασίας μεταγλώττισης του λογισμικού με τη χρήση αρχείων διαμόρφωσης (Configuration Files), τα οποία δεν εξαρτώνται από την πλατφόρμα και το μεταγλωττιστή. Ουσιαστικά, το CMake αναλαμβάνει να παράγει τα αρχεία διαμόρφωσης για κάθε λειτουργικό σύστημα

ώστε να είναι δυνατή η μεταγλώττιση του συνολικού κώδικα αργότερα. Στο λειτουργικό σύστημα Linux δημιουργεί ένα Makefile αρχείο, ενώ κατασκευάζει τα build αρχεία σε ξεχωριστή διεύθυνση από αυτήν που βρίσκονται τα αρχεία πηγαίου κώδικα, με σκοπό την ευκολότερη συντήρηση. Τέλος, το CMake προσφέρει πλείστες δυνατότητες παραμετροποίησης των αρχείων διαμόρφωσης μέσω έτοιμων συναρτήσεων.

2.2: Γνωριμία Με Το ROS

Το βασικό περιβάλλον εργασίας που βασίστηκε η παρούσα εργασία είναι το Robot Operating System (ROS), το οποίο είναι ένα ευέλικτο πλαίσιο για την ανάπτυξη λογισμικού που προορίζεται για ρομποτικές εφαρμογές. Ουσιαστικά, πρόκειται για μια συλλογή από εργαλεία, βιβλιοθήκες και πρωτόκολλα τα οποία στοχεύουν στην διευκόλυνση του προγραμματιστή και την απλοποίηση της διαδικασίας δημιουργίας περίπλοκων και σθεναρών ρομποτικών λειτουργιών. Θεωρείται ένα

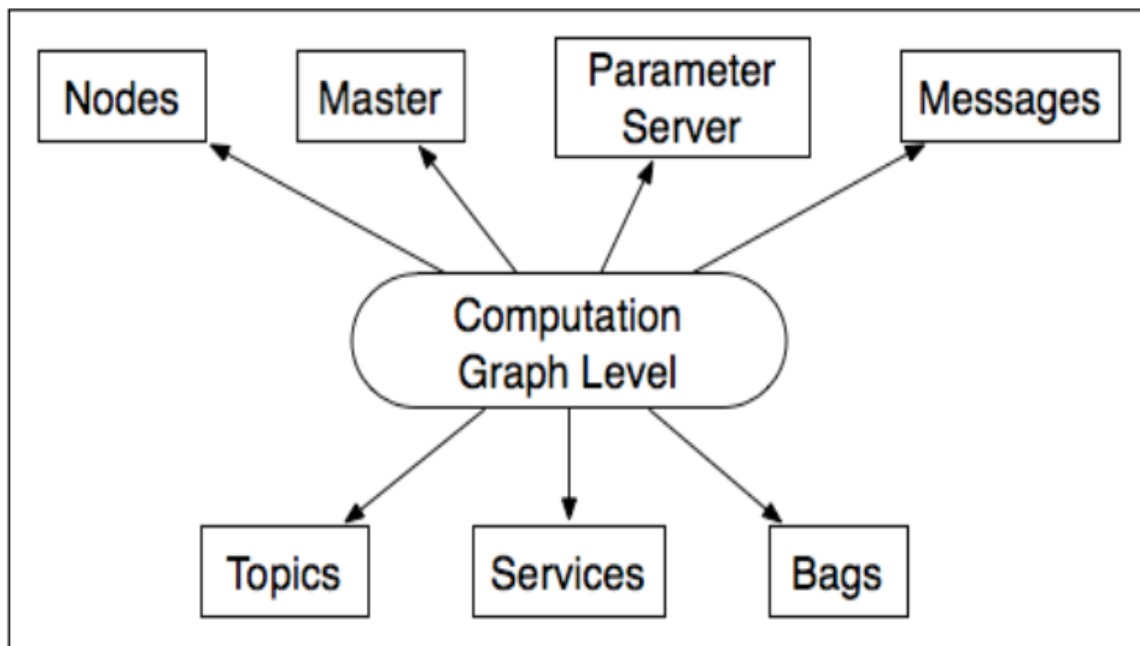


Σχήμα 2.01: Λογότυπο του ROS

μέτα - λειτουργικό σύστημα (meta - operating system), δηλαδή μία πλατφόρμα ανάπτυξης λογισμικού. Το ROS προσφέρει το πλαίσιο μέσω του οποίου

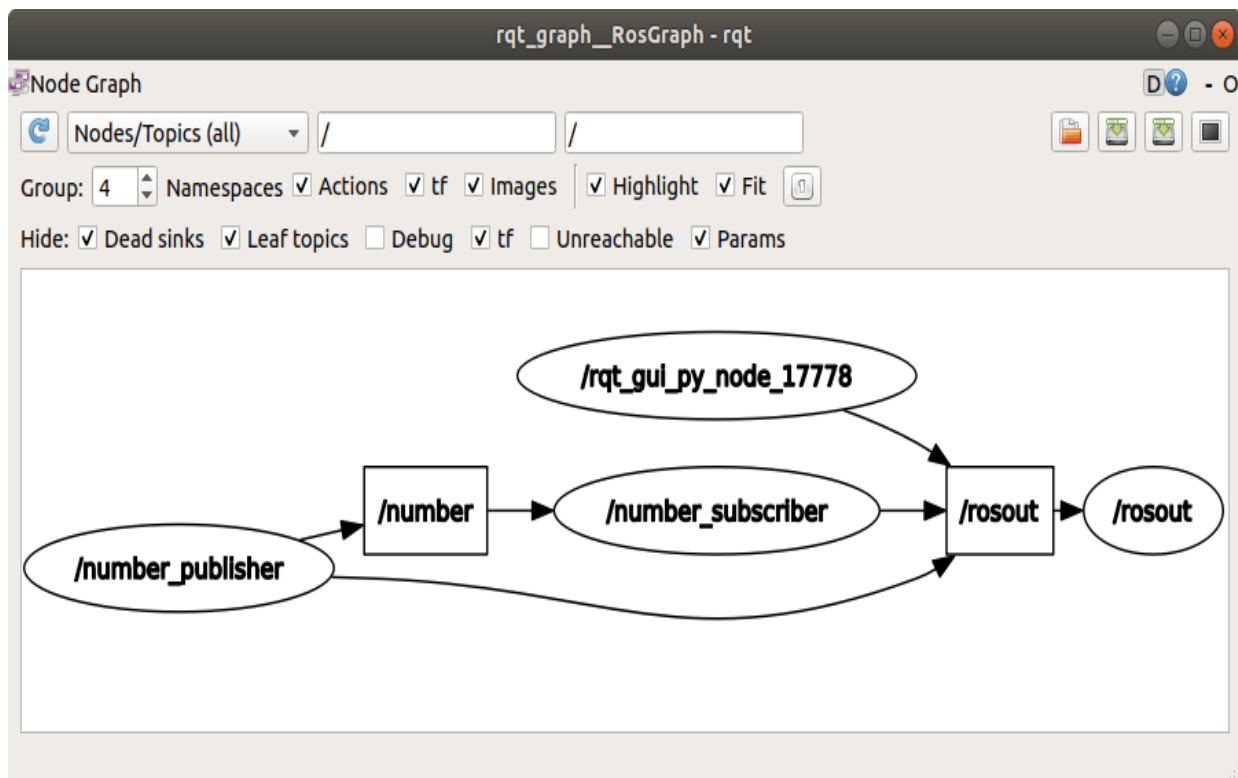
επιτυγχάνεται η επικοινωνία διάφορων στοιχείων μεταξύ τους, καθώς και τους αλγορίθμους οι οποίοι καθορίζουν την απαραίτητη λειτουργία για τη ρομποτική κατασκευή και τέλος τα εργαλεία τα οποία βοηθάνε κατά τη διαδικασία ανάπτυξης λογισμικού. Επιπλέον, διαθέτει προγράμματα οδήγησης υλικού, βιβλιοθήκες, γραφικά περιβάλλοντα χρήσης, δημιουργία και διαχείριση πακέτων και εργαλεία εντοπισμού και διόρθωσης σφαλμάτων. Ένα μεγάλο πλεονέκτημα του ROS είναι ότι λειτουργεί ανεξάρτητα από τη γλώσσα προγραμματισμού του λογισμικού (είναι δηλαδή language agnostic) και επί της ουσίας επιτρέπει την ανάπτυξη ενός πλήρους ρομποτικού συστήματος, η οποία είναι μια εξαιρετικά περίπλοκη διαδικασία, αποκεντρωμένα. Με άλλα λόγια, επιτρέπει την ανάπτυξη ευπροσάρμοστων και ευέλικτων κομματιών κώδικα, ακόμα και σε διαφορετικές γλώσσες προγραμματισμού, που χειρίζονται επί μέρους ζητήματα - τμήματα του συνόλου της εφαρμογής (έλεγχος, διάβασμα αισθητήρων, σχεδιασμός κίνησης κ.α.). Έτσι, καθίσταται πολύ ευκολότερη η συντήρηση και εκσφαλμάτωση του υπάρχοντος λογισμικού, αλλά και η εκμετάλλευση λογισμικού που διατίθεται στο διαδίκτυο. Το ROS είναι και αυτό λογισμικό ανοιχτού κώδικα, διατίθεται δωρεάν και περιοδικά εκδίδονται βελτιωμένες διανομές του. Στη συγκεκριμένη εργασία, χρησιμοποιείται η έκδοση ROS Melodic, η οποία δόθηκε σε κυκλοφορία το 2018 και συνδυάζεται με Ubuntu 18.04.

2.2.1: Οργάνωση του οικοσυστήματος



Σχήμα 2.02: Η δομή του ROS

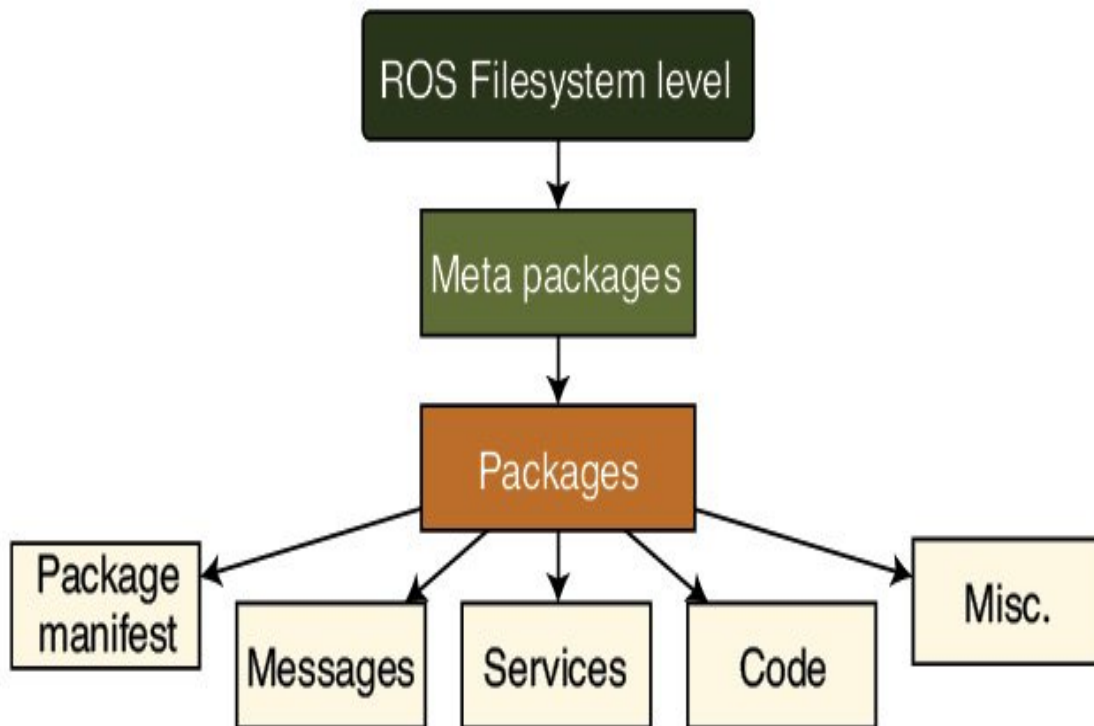
Η λειτουργία του ROS βασίζεται στην ύπαρξη των κόμβων (nodes), οι οποίοι ουσιαστικά αποτελούν μια υπολογιστική διεργασία. Συνήθως, ένα σύστημα αποτελείται από πολλούς κόμβους, καθένας από τους οποίους επιτελεί διαφορετικούς σκοπούς και ρόλους. Έκτος των υπολοίπων, σε κάθε ενεργό σύστημα υπάρχει υποχρεωτικά ο κόμβος master, ο οποίος είναι επιφορτισμένος με τον συντονισμό και την επικοινωνία όλων των υπολοίπων μερών. Το ROS δομείται μέσω μιας αρχιτεκτονικής γράφων, όπου οι κόμβοι μπορούν να επικοινωνούν μεταξύ τους μέσω ειδικών διαύλων επικοινωνίας, των θεμάτων (topics) και των υπηρεσιών (services).



Σχήμα 2.03: Το ROS ως γράφος

Πιο συγκεκριμένα, ένας κόμβος, ο οποίος σε πρακτικό επίπεδο δεν είναι κάτι παραπάνω από ένα εκτελέσιμο αρχείο, μπορεί να επικοινωνήσει με άλλους κόμβους είτε δημοσιεύοντας (publish) είτε μένοντας εγγεγραμμένος (subscribe) σε topic. Τα δεδομένα που διακινούνται μεταξύ των κόμβων ονομάζονται μηνύματα (messages). Ένας ακόμα τρόπος επικοινωνίας είναι και οι υπηρεσίες, δηλαδή μια δομή τύπου πελάτη - εξυπηρετητή (client - server). Για παράδειγμα, ένα απλό μέρος ενός συστήματος μπορεί να είναι ένας κόμβος, ο οποίος δέχεται ως μηνύματα δεδομένα - μετρήσεις ενός αισθητήρα, εκτελεί κάποια υπολογιστική διαδικασία και στην συνέχεια στέλνει τα αποτελέσματα σε άλλον κόμβο. Η

οργανωτική δομή του ROS σε κόμβους είναι αυτή που του προσδίδει αυτή την ευελιξία. Επίσης, παράλληλα με τον κόμβο master, λειτουργεί και ο διακομιστής παραμέτρων (parameter server), όπου αποθηκεύονται διάφορες τιμές μεταβλητών για μετέπειτα χρήση.



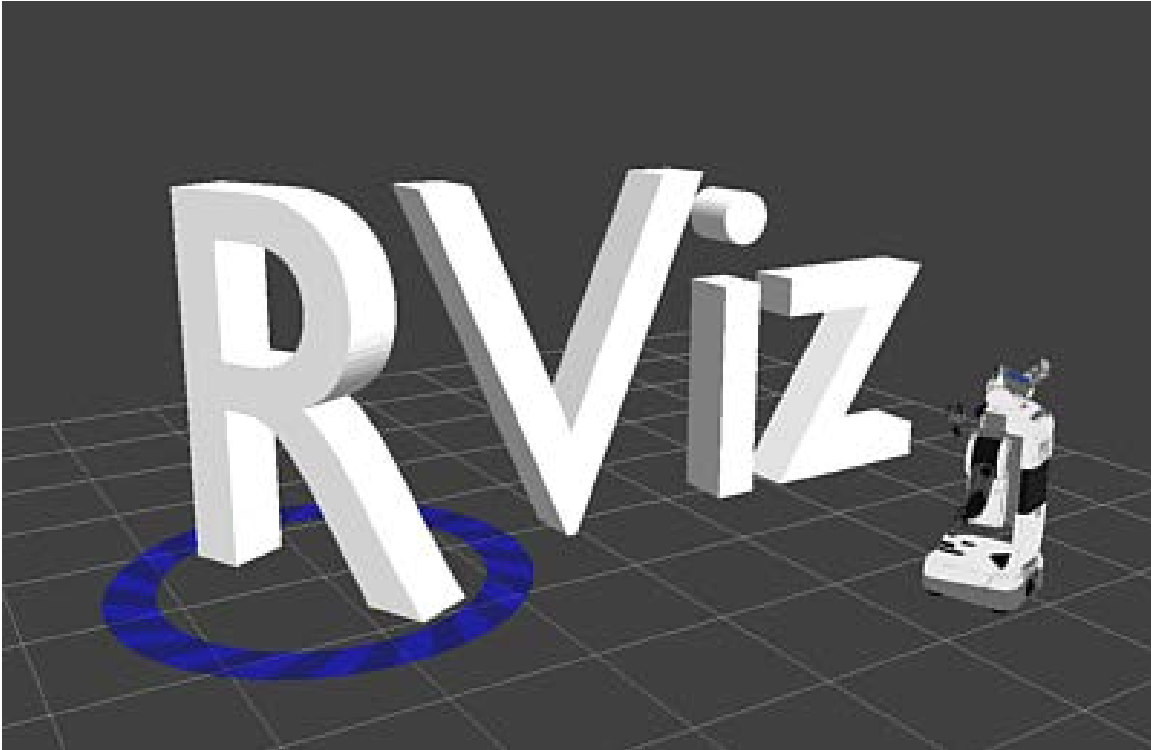
Σχήμα 2.04: Η οργάνωση αρχείων του ROS

Η κύρια μονάδα οργάνωσης του λογισμικού στο ROS είναι τα πακέτα. Κάθε πακέτο περιλαμβάνει βιβλιοθήκες, εκτελέσιμα και τμήματα κώδικα (scripts). Επίσης, περιλαμβάνει ένα αρχείο package.xml όπου ορίζονται οι άδειες χρήσης του, διάφορες πληροφορίες αλλά και οι εξαρτήσεις (dependencies) του από άλλα πακέτα. Τα μέτα - πακέτα (meta - packages) αποτελούν απλά μία ομαδοποίηση ενός αριθμού πακέτων, τα οποία συνήθως δρουν συνεργατικά για την επίτευξη μιας σύνθετης ρομποτικής λειτουργίας.

Το πρώτο βήμα για την έναρξη χρήσης του ROS είναι η δημιουργία ενός catkin χώρου εργασίας (catkin workspace), δηλαδή ενός φακέλου στον οποίο τροποποιούνται, χτίζονται και εγκαθίσταται catkin πακέτα. Ένας τυπικός catkin χώρος εργασίας περιέχει το source space, ο οποίος περιέχει τον πηγαίο κώδικα των πακέτων, το build space όπου χρησιμοποιεί το CMake για να χτιστούν τα

catkin πακέτα και τέλος το devel space όπου τοποθετούνται τα targets πριν εγκατασταθούν.

2.2.2: Το λογισμικό RViz



Σχήμα 2.05: Το λογισμικό RViz

Το λογισμικό Rviz (συντομία του ROS Visualization) αποτελεί έναν τρισδιάστατο οπτικοποιητή (3 Dimensional Visualisator) ο οποίος, όπως προσδίδει και το όνομά του, χρησιμοποιείται παράλληλα με το ROS. Πρόκειται για ένα πολύ ωφέλιμο βοήθημα που χρησιμοποιείται από τους προγραμματιστές κατά τη διάρκεια ανάπτυξης ενός έργου για τον εντοπισμό σφαλμάτων και γενικά για την εποπτεία της εύρυθμης λειτουργίας της ρομποτικής διάταξης. Το εν λόγω εργαλείο προσφέρει ένα εύχρηστο γραφικό περιβάλλον χρήστη και του επιτρέπει να απεικονίσει το τρισδιάστατο μοντέλο της ρομποτικής κατασκευής, την καταγραφή των μετρήσεων των διάφορων αισθητήρων αλλά και αναπαράγει εκ νέου καταγεγραμμένα δεδομένα. Ο προγραμματιστής δύναται μέσω του Rviz να εμφανίσει δεδομένα από κάμερες και λέιζερ ή να ελέγξει τα διάφορα συστήματα συντεταγμένων της κατασκευής, την τρέχουσα στάση του κ.α.. Έτσι, είναι ευκολότερο να διαισθανθεί το τι «βλέπει» το ρομπότ, με άμεση συνέπεια τον ευκολότερο εντοπισμό ποικίλων κατασκευαστικών σφαλμάτων. Με άλλα λόγια,

μέσω του Rviz είναι εφικτή η γραφική απεικόνιση και καταγραφή του συστήματος ανά πάσα στιγμή, η παρακολούθηση της κίνησης και των δεδομένων των αισθητήρων. Φυσικά, τα παραπάνω ισχύουν είτε χρησιμοποιούμε ένα πραγματικό σύστημα στο φυσικό κόσμο, είτε προσομοιώνουμε τη λειτουργία του σε κάποιο περιβάλλον προσομοίωσης.

2.3: Γνωριμία Με Το Gazebo



Σχήμα 2.06: Το λογότυπο του Gazebo

Το Gazebo είναι ο πιο διαδεδομένος τρισδιάστατος προσομοιωτής (3D simulator) που χρησιμοποιείται με το ROS. Πρόκειται για ένα ισχυρό προγραμματιστικό εργαλείο με ενσωματωμένη μηχανή γραφικών, υψηλής ποιότητας γραφικά και φιλικό προς τους προγραμματιστές γραφικό περιβάλλον χρήσης, ενώ όπως και το ROS διατίθεται δωρεάν. Το Gazebo προσφέρει τη δυνατότητα σχεδιασμού ενός ή περισσότερων ρομποτικών συστημάτων σε διαφορετικά σενάρια λειτουργίας, με ακριβή και αποδοτική προσομοίωση αλγορίθμων. Επίσης, ο χρήστης έχει την ευχέρεια να δημιουργήσει εσωτερικούς ή εξωτερικούς χώρους με πληθώρα

διαφορετικών μοντέλων με εξαιρετική ακρίβεια και συνοχή ως προς τις πραγματικές συνθήκες και τα δυναμικά φαινόμενα που λαμβάνουν χώρα. Επιπλέον, προσφέρει διαφορετικές επιλογές ως προς τις μηχανές φυσικής με προεπιλεγμένη την Open Dynamics Engine (ODE). Το Gazebo αποτελείται από δύο μέρη: έναν διακομιστή (ο οποίος λειτουργεί ως μία διεργασία gzserver) και ένα πελάτη (το οποίο λειτουργεί ως μία διεργασία gzclient). Μια ακόμα εντυπωσιακή και πολύτιμη ικανότητα του Gazebo είναι ότι μπορεί να προσομοιώσει επίσης και αισθητήρες συμπεριλαμβανομένων αισθητήρων μέτρησης δύναμης, επιταχυνσιόμετρων ή καμερών.

Οι δυνατότητες προσομοίωσης και απεικόνισης που παρέχει το ROS μέσω του Gazebo ανήκουν στις πολύτιμες ιδιότητές του. Όταν η ανάπτυξη λογισμικού μπορεί να πραγματοποιηθεί μόνο σε συγκεκριμένο υλικό, κατάσταση αλληλένδετη με τον τομέα της ρομποτικής, η παραγωγικότητα μειώνεται. Ένα φυσικό ρομπότ είναι συνήθως ένα εμπόδιο στην ανάπτυξη, επειδή είναι ένας περιορισμένος πόρος που πιθανώς μοιράζεται από μια ομάδα. Επιπλέον, ορισμένα ρομπότ, συμπεριλαμβανομένων οχημάτων και βραχιόνων, απαιτούν από ομάδες να εκτελέσουν πειράματα. Τις περισσότερες φορές, το ρομπότ και η ομάδα βρίσκεται σε αδράνεια τη στιγμή που ένας προγραμματιστής αναζητά σφάλματα στον υπό δοκιμή κώδικα. Με έναν κατάλληλο προσομοιωτή ρομπότ, ο κώδικας μπορεί να αναπτυχθεί υπό προσομοίωση, επιταχύνοντας την όλη διαδικασία και εξασφαλίζοντας τη μέγιστη δυνατή ασφάλεια, όχι μόνο του προσωπικού αλλά και της ρομποτικής διάταξης. Αν ο προσομοιωτής έχει υψηλή πιστότητα και παρέχει μια καλή προσέγγιση της πραγματικής συμπεριφοράς του ρομπότ στο φυσικό κόσμο, λίγες ή καθόλου αλλαγές ενδέχεται να απαιτηθούν για την εκτέλεση του ίδιου κώδικα στο φυσικό σύστημα. Τυπικά, το λογισμικό που αναπτύχθηκε για την προσομοίωση σε ROS δεν απαιτεί αλλαγές για να εκτελεστεί στο φυσικό ανάλογό του. Χρησιμοποιώντας το σύστημα επικοινωνίας του ROS, οι κόμβοι αδιαφορούν για το πού βρίσκεται το άλλο άκρο επικοινωνίας στο σύστημα, αρκεί οι τύποι μηνυμάτων και τα ονόματα των θεμάτων να είναι σωστά. Φυσικά, η επιτυχία και η χρησιμότητα της ανάπτυξης λογισμικού στην προσομοίωση εξαρτάται άμεσα από την ποιότητα του προσομοιωτή.

2.3.1: Προσομοίωση ενός μοντέλου

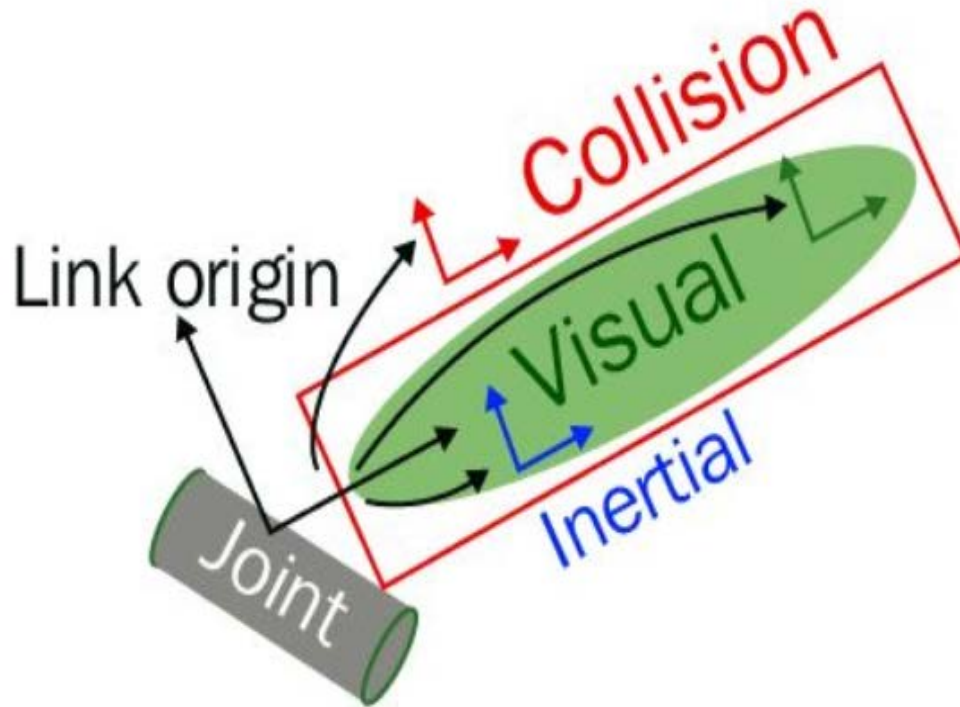
Για να εκτελεστεί μία δυναμική προσομοίωση με επιτυχία και ρεαλιστικότητα, τα μοντέλα των αντικειμένων και εν γένει των κατασκευών πρέπει να πληρούν κάποια κριτήρια και να ορίζονται με λεπτομέρεια. Οι λεπτομέρειες αυτές κατηγοριοποιούνται σε τρεις κατηγορίες: το οπτικό μοντέλο (visual model), το δυναμικό μοντέλο (dynamic model) και το μοντέλο συγκρούσεων (collision model).

Το δυναμικό μοντέλο πρέπει πάντα να περιλαμβάνει οπωσδήποτε τη μάζα και τα αδρανειακά χαρακτηριστικά του αντικειμένου, αφού αυτά συνιστούν την απαραίτητη περιγραφή για ένα αφηρημένο φυσικό μοντέλο. Μόνο ένα αδρανειακό μοντέλο αρκεί ώστε κάποιος να εκτελέσει χρήσιμες προσομοιώσεις. Για παράδειγμα, θα μπορούσε κανείς να υπολογίσει τη δυναμική ενός δορυφόρου που δρα υπό την επίδραση της βαρύτητας.

Όσον αφορά το μοντέλο συγκρούσεων, για να υπολογιστεί η δυναμική των αλληλοεπιδρώντων σωμάτων, συμπεριλαμβανομένων είτε αντικειμένων που συγκρούονται είτε ρομπότ που πιάνουν ή σπρώχνουν ένα αντικείμενο, είναι επίσης απαραίτητο να υπολογίσουμε τις δυνάμεις και τις ροπές που δημιουργούνται λόγω των επαφών. Η προσομοίωση της δυναμικής των επαφών είναι ένα δύσκολο και προκλητικό εγχείρημα και έχει διαφορετικές μεθόδους αντιμετώπισης. Η ODE χρησιμοποιεί την ενέργεια και τη διατήρηση της ορμής για να υπολογίσει το αποτέλεσμα των συγκρούσεων και δεν υπολογίζει λεπτομερώς την στιγμιαία δυναμική των δυνάμεων κατά τη διάρκεια των κρούσεων. Αυτή η επιλογή επιφέρει εξαιρετικά αποτελέσματα στη μοντελοποίηση σύντομων συγκρούσεων, αλλά όχι τόσο καλά στην προσομοίωση δυνάμεων συνεχούς επαφής, συμπεριλαμβανομένων των τροχών στο έδαφος ή των δακτύλων που πιάνουν αντικείμενα. Επίσης, για να συμπεριλάβει τα δυναμικά αποτελέσματα των συγκρούσεων, ο προσομοιωτής πρέπει να είναι σε θέση να συμπεράνει πότε και πού (σε κάθε σώμα) συμβαίνει επαφή. Είναι επομένως απαραίτητο να προσδιοριστεί και η περιβάλλουσα των τρισδιάστατων αντικειμένων με τρόπο κατάλληλο για τον αποτελεσματικό υπολογισμό των δυνάμεων και ροπών. Αυτή η περιβάλλουσα συνιστά το μοντέλο συγκρούσεων και μπορεί να είναι από ένα πρωτόλειο γεωμετρικό σχήμα (κύβος, κύλινδρος κτλ) μέχρι ένα μοντέλο υψηλής πιστότητας προερχόμενο από ένα μοντέλο CAD. Η περιγραφή των επιφανειών μεταφράζεται προσεγγιστικά σε ένα μοντέλο STL, αποτελούμενο από τριγωνικές όψεις, το οποίο είναι υπολογιστικά αποδοτικό για τον υπολογισμό των τομών μεταξύ των ορίων των μοντέλων.

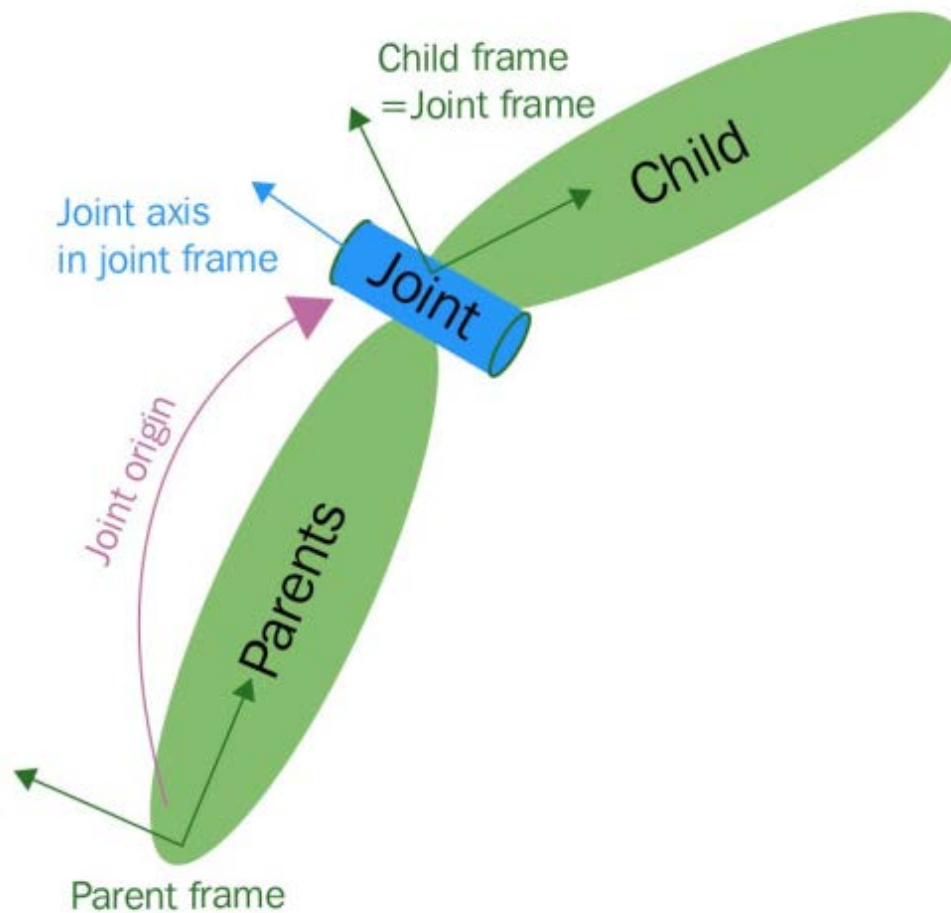
Η τρίτη κατηγορία περιγραφής είναι το οπτικό μοντέλο, το οποίο χρησιμοποιείται αποκλειστικά για τους σκοπούς της γραφικής προβολής. Η γραφική απεικόνιση της υπολογισμένης δυναμικής μπορεί να είναι εξαιρετικά χρήσιμη για την ερμηνεία των αποτελεσμάτων του πειράματος και τον εντοπισμό σφαλμάτων στην ανάπτυξη λογισμικού. Το οπτικό μοντέλο είναι συχνά πανομοιότυπο με το μοντέλο συγκρούσεων, αφού το οπτικό μοντέλο απαιτεί επίσης περιγραφή των γεωμετρικών ορίων του αντικειμένου. Τις περισσότερες φορές, όμως, το οπτικό μοντέλο είναι υψηλότερης πιστότητας, ενσωματώνοντας περισσότερες τριγωνικές όψεις, από το μοντέλο σύγκρουσης. Αυτό συμβαίνει διότι η εμφάνιση ενός μοντέλου απαιτεί αρκετά λιγότερους υπολογιστικούς πόρους σε σχέση με τον υπολογισμό των συγκρούσεων ενός μοντέλου με μεγάλο αριθμό όψεων. Ωστόσο, αυτές οι δύο κατηγορίες μπορούν να περιλαμβάνουν και δικές τους παραμέτρους, όπως συντελεστές τριβής και ελαστικότητας (που δεν έχουν

σημασία για την οπτική εμφάνιση) ή προδιαγραφές χρώματος, ανακλαστικότητας και διαφάνειας (που δεν έχουν επίδραση στο μοντέλο συγκρούσεων).



Σχήμα 2.07: Απεικόνιση ενός URDF συνδέσμου

Ένα μοντέλο ενός ρομπότ, το οποίο πληροί τις παραπάνω ιδιότητες, μπορεί να περιγράψει μέσω ενός ειδικού αρχείου, του Simulation Description Format (SDF). Το πρότυπο SDF υποστηρίζεται από το Gazebo και βασίζεται στην κωδικοποίηση XML. Ωστόσο, δυστυχώς, το πρότυπο SDF δεν υποστηρίζεται από το ROS. Αντί αυτού, χρησιμοποιείται το πρότυπο Unified Robot Description Format (URDF), το οποίο είναι παραπλήσιο μεν, παλαιότερο και με περιορισμένες δυνατότητες δε. Το μοντέλο της ρομποτικής διάταξης ορίζεται ως μια ανοιχτή κινηματική αλυσίδα, δηλαδή σε δομή δέντρου (kinematic tree). Εκεί ορίζονται κάθε άρθρωση και κάθε σύνδεσμος του ρομπότ, με κάθε στοιχείο να περιλαμβάνει τις τρεις προαναφερθέντες κατηγορίες. Επιπλέον, οι αρθρώσεις περιλαμβάνουν και περαιτέρω απαραίτητα χαρακτηριστικά, όπως ο τύπος της άρθρωσης, οι άξονες κίνησης, τα όρια κίνησης κ.ο.κ. Κάθε σύνδεσμος έχει έναν άλλον μοναδικό σύνδεσμο ως γονέα, αλλά δύναται να έχει πολλαπλούς συνδέσμους παιδιά. Κατά αυτόν τον τρόπο εξασφαλίζεται η δομή της ανοιχτής κινηματικής αλυσίδας.



Σχήμα 2.08: Απεικόνιση μιας URDF άρθρωσης

Εν κατακλείδι, αξίζει να αναφέρουμε ότι δεν πρέπει να συγχέει κανείς την ύπαρξη του Rviz με αυτή του Gazebo, μιας και επιτελούν διαφορετικούς σκοπούς. Μέσω του Rviz απεικονίζεται αποκλειστικά η τρέχουσα κατάσταση της ρομποτικής διάταξης, αγνοώντας πλήρως οποιοδήποτε δυναμικό φαινόμενο, όπως τις επιδράσεις της βαρύτητας, τις επιδράσεις τριβών κ.α..

Κεφάλαιο 3: Δυναμική & Έλεγχος Ρομποτικών Διατάξεων

Σε αυτό το κεφάλαιο θα αναλυθεί η δυναμική και ο έλεγχος των ρομποτικών διατάξεων. Αρχικά, θα παρουσιαστεί η διαφορική κινηματική μιας κινηματικής αλυσίδας, στη συνέχεια θα μοντελοποιηθεί το δυναμικό πρόβλημα και τέλος θα αναλυθεί η τεχνική του ελέγχου δύναμης.

3.1: Διαφορική Κινηματική

Για δεδομένη ρομποτική διάταξη, το πρόβλημα του ευθέως διαφορικού κινηματικού μοντέλου ασχολείται με τον προσδιορισμό των εξισώσεων της γραμμικής και γωνιακής ταχύτητας του τερματικού εργαλείου δράσης συναρτήσει των ταχυτήτων των μεταβλητών των αρθρώσεων. Αυτή η αντιστοίχιση περιγράφεται από μια μήτρα, που ονομάζεται Ιακωβιανή μήτρα (Jacobian matrix), η οποία εξαρτάται από τη διαμόρφωση της κινηματικής αλυσίδας. Εναλλακτικά, εάν η θέση και ο προσανατολισμός της τελικής αρπάγης εκφράζεται αναφορικά με το χώρο εργασίας, είναι δυνατόν η Ιακωβιανή μήτρα να υπολογιστεί μέσω διαφορίσης της κινηματικής εξίσωσης ως προς μεταβλητές των αρθρώσεων. Η Ιακωβιανή μήτρα αποτελεί ένα από τα πιο σημαντικά εργαλεία για τη μελέτη μιας κινηματικής αλυσίδας και είναι απαραίτητη για διάφορα ζητήματα που θα παρουσιαστούν σε επόμενα κεφάλαια.

Για δεδομένη κινηματική αλυσίδα με n βαθμούς ελευθερίας, οι εξισώσεις του κινηματικού μοντέλου αυτής, δηλαδή της σχέσης που συνδέει τη θέση και τον προσανατολισμό του τερματικού εργαλείου δράσης στον τρισδιάστατο χώρο ανάλογα τις αποκλίσεις των αρθρώσεων, περιγράφονται από:

$$T_e(q) = \begin{bmatrix} R_e(q) & p_e(q) \\ 0^T & 1 \end{bmatrix}_{4 \times 4} \quad (3.01)$$

όπου $q = [q_1 \ \dots \ q_n]^T$ το διάνυσμα μεταβλητών των αρθρώσεων. Προφανώς, και ο προσανατολισμός ($R_e(q)$) και η θέση ($p_e(q)$) του τερματικού εργαλείου

δράσης εξαρτώνται από τις μεταβλητές των αρθρώσεων. Όπως αναφέρθηκε, ο στόχος της διαφορικής κινηματικής είναι να βρει τη σχέση μεταξύ των ταχυτήτων των αρθρώσεων και των γραμμικών και γωνιακών ταχυτήτων τελικού εργαλείου. Αποδεικνύεται [1] ότι οι αναζητούμενες σχέσεις είναι και οι δύο γραμμικές ως προς τις ταχύτητες των αρθρώσεων:

$$\dot{p}_e = J_p(q) \cdot \dot{q} \quad (3.02)$$

$$\omega_e = J_o(q) \cdot \dot{q} \quad (3.03)$$

όπου \dot{p}_e , ω_e η γραμμική ταχύτητα και η γωνιακή ταχύτητα αντίστοιχα. Επίσης, οι μήτρες $J_p(q)$, $J_o(q)$ έχουν διαστάσεις $(3 \times n)$ και αποτυπώνουν τη συνεισφορά των ταχυτήτων των αρθρώσεων με τη γραμμική και γωνιακή ταχύτητα αντίστοιχα του τελικού εργαλείου δράσης. Σε συμπυκνωμένη μορφή έχουμε:

$$u_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix}_{6 \times 1} = J(q)_{6 \times n} \cdot \dot{q}_{n \times 1} \quad (3.04)$$

Η μήτρα $J(q)_{6 \times n}$ ονομάζεται Ιακωβιανή και εν γένει εξαρτάται από τη διαμόρφωση της διάταξης.

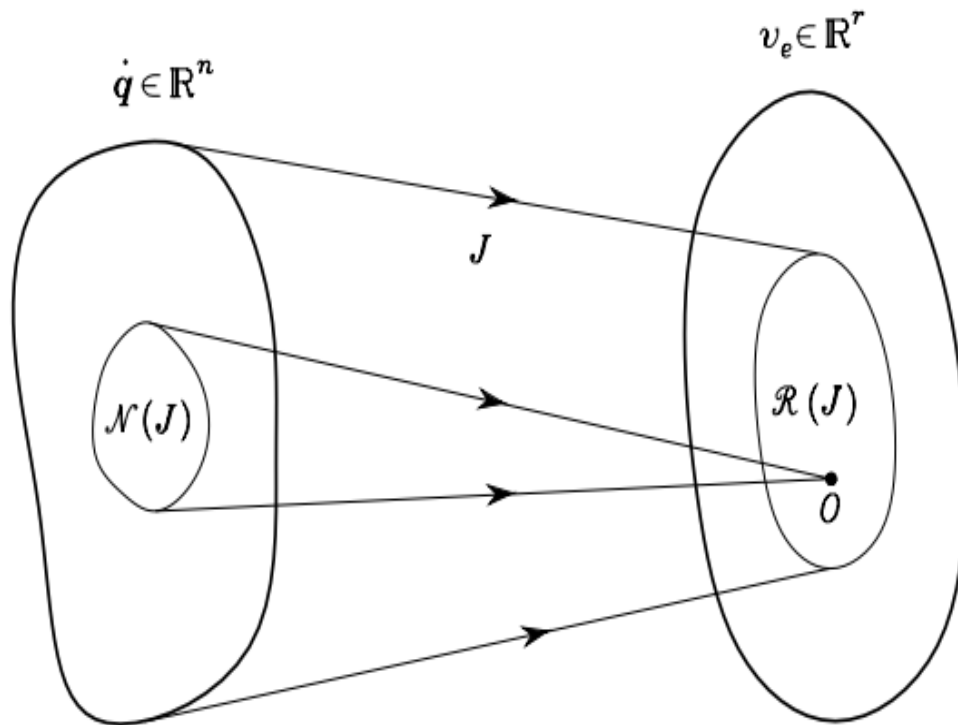
Μέσω της προηγούμενης σχέσης μπορεί να οριστεί υπό συνθήκες και το αντίστροφο διαφορικό μοντέλο:

$$\dot{q} = J(q)^{-1} \cdot u_e \quad (3.05)$$

Φυσικά, αυτή η σχέση υφίσταται με την προϋπόθεση ότι η Ιακωβιανή μήτρα είναι τετραγωνική και αντιστρέψιμη, δηλαδή ότι $n = 6$ και ότι η ορίζουσα $\det|J| \neq 0$. Αν η κατασκευή μεταβεί σε μια διαμόρφωση ώστε η εν λόγω ορίζουσα μηδενιστεί, τότε το ρομποτικό σύστημα μεταβαίνει σε μια ιδιόμορφη κατάσταση (Kinematic Singularity) με πολλαπλά δυσάρεστα αποτελέσματα. Πρώτον, οι ιδιόμορφες διατάξεις αποτελούν διαμορφώσεις του συστήματος, στις οποίες η κινητικότητα του μειώνεται, δηλαδή είναι αδύνατον να επιβληθεί μια συγκεκριμένη τροχιά στο τερματικό εργαλείο δράσης γιατί υπάρχει τουλάχιστον μία διεύθυνση κατά την οποία το ρομπότ δεν μπορεί να κινηθεί. Επίσης, ενδεχομένως υπάρχουν άπειρες λύσεις στο πρόβλημα της αντίστροφης κινηματικής. Επιπλέον, πρόβλημα

δημιουργείται και σε γειτονικές περιοχές των ιδιόμορφων διατάξεων, δηλαδή εκεί όπου η ορίζουσα παίρνει πολύ μικρές τιμές, αφού ελάχιστη ταχύτητα στο λειτουργικό χώρο μετουσιώνεται σε τεράστιες ταχύτητες στο χώρο των αρθρώσεων.

Οι ιδιόμορφες διατάξεις μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες, τις συνοριακές και τις εσωτερικές. Οι συνοριακές ιδιόμορφες διατάξεις εμφανίζονται όταν ο χειριστής είναι είτε πλήρως προτεταμένος είτε πλήρως ανακλημένος. Μπορεί να γίνει κατανοητό ότι αυτές οι εκφυλισμένες διατάξεις δεν αποτελούν πραγματικό πρόβλημα, καθώς μπορούν να αποφευχθούν υπό τον όρο ότι το ρομπότ δεν οδηγείται στα όρια του χώρου εργασίας του. Αντιθέτως, οι εσωτερικές ιδιόμορφες διατάξεις εμφανίζονται εντός του προσβάσιμου χώρου εργασίας και γενικά προκαλούνται είτε από την ευθυγράμμιση δύο ή περισσότερων αξόνων κίνησης είτε από την επίτευξη συγκεκριμένων διαμορφώσεων. Σε αντίθεση με τις προηγούμενες, αυτές οι περιπτώσεις αποτελούν σοβαρό πρόβλημα, αφού μπορούν να εμφανιστούν οπουδήποτε στον χώρο εργασίας για μια προγραμματισμένη τροχιά στο λειτουργικό χώρο.



Σχήμα 3.01: Αντιστοίχιση της ταχύτητας των αρθρώσεων με την ταχύτητα του λειτουργικού χώρου για πλεονάζοντες βαθμούς ελευθερίας

Για όλους τους παραπάνω λόγους, είναι, λοιπόν, ζωτικής σημασίας οι ιδιόμορφες διατάξεις να ανιχνεύονται ώστε να αποφευχθούν εγκαίρως. Αυτό

συνήθως γίνεται χρησιμοποιώντας κάποιους πλεονάζοντες βαθμούς ελευθερίας της ρομποτικής διάταξης. Αρχικά, όπως είναι εύκολα κατανοητό, απαιτούνται 6 βαθμοί ελευθερίας για να οριστεί η θέση και ο προσανατολισμός στον τρισδιάστατο χώρο. Οπότε, αν μία ρομποτική διάταξη περιλαμβάνει περισσότερους (δηλαδή $n > 6$, τότε υπάρχουν $n - 6$ πλεονάζοντες βαθμοί ελευθερίας, υπό την προϋπόθεση ότι η Ιακωβιανή μήτρα είναι πλήρους τάξης (δηλαδή $rank[J] = 6$). Οι πλεονάζοντες βαθμοί ελευθερίας εισάγουν τη δυνατότητα απειρίας εφικτών κινήσεων στις αρθρώσεις για δοσμένη επιθυμητή κίνηση στο τελικό στοιχείο δράσης και γίνεται να χρησιμοποιηθούν για την ικανοποίηση επιπρόσθετων κινηματικών περιορισμών, όπως η αποφυγή εμποδίων και η αποφυγή ιδιόμορφων διατάξεων. Έτσι, το πρόβλημα της αντίστροφης διαφορικής μοντελοποίησης ανάγεται σε ένα πρόβλημα βελτιστοποίησης κατά το οποίο η ταχύτητα u_e του τερματικού εργαλείου δράσης καθώς και η Ιακωβιανή μήτρα είναι πλήρους τάξης και γνωστές, ενώ οι ταχύτητες των αρθρώσεων \dot{q} ζητούμενες. Πιο συγκεκριμένα, επιλέγοντας την ελαχιστοποίηση μιας τετραγωνικής συνάρτησης κόστους της μορφής:

$$g(\dot{q}) = \frac{1}{2} \dot{q}^T W \dot{q} \quad (3.06)$$

όπου W μια συμμετρική θετικά ορισμένη μήτρα διαστάσεων $n \times n$. Χρησιμοποιώντας τη μέθοδο των πολλαπλασιαστών Lagrange:

$$g(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T W \dot{q} + \lambda^T (u_e - J \dot{q}) \quad (3.07)$$

Κατά τα γνωστά, χρησιμοποιώντας τις συνθήκες πρώτης τάξης:

$$\left(\frac{\partial g}{\partial \dot{q}}\right)^T = 0 \quad \left(\frac{\partial g}{\partial \lambda}\right)^T = 0 \quad (3.08)$$

καταλήγουμε στις σχέσεις:

$$\dot{q} = W^{-1} J^T \lambda \quad \lambda = (J W^{-1} J^T)^{-1} u_e \quad (3.09)$$

Και τελικά, επιλέγοντας $W = I$ έχουμε τη βέλτιστη λύση:

$$\dot{q} = J^T (JJ^T)^{-1} \cdot u_e = J^\dagger \cdot u_e \quad (3.10)$$

Έχοντας ορίσει την ψεύδο - αντίστροφη μήτρα της Ιακωβιανής:

$$J^\dagger = J^T (JJ^T)^{-1} \quad (3.11)$$

Ενώ ταυτοχρόνως ικανοποιούνται οι συνθήκες δεύτερης τάξης $\left(\frac{\partial^2 g}{\partial \dot{q}^2} = W > 0\right)$.

Αξίζει να αναφέρουμε ότι η Εξ. (3.10) ισχύει μόνο αν η Ιακωβιανή μήτρα είναι πλήρους τάξης. Όμως, ακόμα και στην γειτονιά των ιδιόμορφων διατάξεων δύναται να δημιουργηθούν αριθμητικά προβλήματα υπολογισμού της Εξ. (3.11). Για αυτό το λόγο, προτείνεται η χρησιμοποίηση μιας εναλλακτικής λύσης στη γειτονιά των ιδιόμορφων διατάξεων:

$$J^* = J^T (JJ^T + k^2 I)^{-1} \quad (3.12)$$

όπου k είναι ένας συντελεστής απόσβεσης και I η ταυτοτική μήτρα.

3.2: Δυναμικό Μοντέλο

Η εξαγωγή του δυναμικού μοντέλου μιας ρομποτικής κατασκευής παίζει σημαίνοντα ρόλο στην προσομοίωση κίνησής της και στο σχεδιασμό των αλγορίθμων ελέγχου. Επιπλέον, η προσομοίωση της κίνησης του συστήματος επιτρέπει τη δοκιμή και εκσφαλμάτωση διάφορων στρατηγικών ελέγχου και τεχνικών σχεδιασμού τροχιάς χωρίς την ανάγκη χρήσης του φυσικού συστήματος. Επίσης, η ανάλυση του δυναμικού μοντέλου μπορεί να φανεί χρήσιμη για το μηχανικό σχεδιασμό των πρωτότυπων κατασκευών, αφού ο υπολογισμός των δυνάμεων και των ροπών που απαιτούνται για την εκτέλεση τυπικών κινήσεων παρέχει χρήσιμες πληροφορίες για το σχεδιασμό των αρθρώσεων και των ενεργοποιητών.

Με τον όρο δυναμικό μοντέλο εννοούμε την περιγραφή της σχέσης που συνδέει τις ροπές των αρθρώσεων τ_i με την κίνηση της ρομποτικής διάταξης q, \dot{q}, \ddot{q} . δύο μεθόδους για την εξαγωγή των εξισώσεων κίνησης ενός χειριστή στον κοινό χώρο. Γενικά, υπάρχουν δύο τρόποι για την εξαγωγή του δυναμικού μοντέλου. Η πρώτη μέθοδος βασίζεται στη μεθοδολογία Lagrange και είναι εννοιολογικά απλούστερη. Η δεύτερη μέθοδος βασίζεται στις εξισώσεις Newton – Euler και αποδίδει το μοντέλο σε αναδρομική μορφή και είναι υπολογιστικά πιο αποτελεσματικό. Στη συνέχεια, θα παρουσιαστεί συνοπτικά η μέθοδος Lagrange. Η λαγκρανζιανή συνάρτηση (Lagrangian) ενός μηχανικού συστήματος δίνεται από την:

$$\mathcal{L} = T - U \quad (3.13)$$

όπου T, U η κινητική και δυναμική ενέργεια αντίστοιχα. Η κινητική ενέργεια ισούται με το άθροισμα των συνεισφορών των κινήσεων τόσο των συνδέσμων όσο και των αρθρώσεων:

$$T(q, \dot{q}) = \sum_{i=1}^n (T_{ii} + T_{ji}) = \frac{1}{2} \dot{q}^T B(q) \dot{q} \quad (3.14)$$

όπου η μήτρα αδράνειας $B(q)$ είναι μια συμμετρική θετικά ορισμένη μήτρα διαστάσεων $n \times n$. Αντίστοιχα, η δυναμική ενέργεια δίνεται από τη σχέση:

$$U(q) = \sum_{i=1}^n (U_{ii} + U_{ji}) = - \sum_{i=1}^n (m_{li} g_0^T r_{li} + m_{ji} g_0^T r_{ji}) \quad (3.15)$$

όπου m οι μάζες, g_0^T το διάνυσμα της επιτάχυνσης της βαρύτητας εκφρασμένο ως προς το σύστημα συντεταγμένων της βάσης και r_{li}, r_{ji} τα διανύσματα απόστασης του κέντρου μάζας του συνδέσμου και των αξόνων των αρθρώσεων αντίστοιχα από το σύστημα συντεταγμένων της βάσης. Τότε, η δυναμική εξίσωση Lagrange προκύπτει από:

$$\tau_i = \frac{d}{dt} \left(\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q_i} \quad (3.16)$$

Και μετά από μαθηματικές πράξεις εν τέλει καταλήγουμε στο δυναμικό μοντέλο:

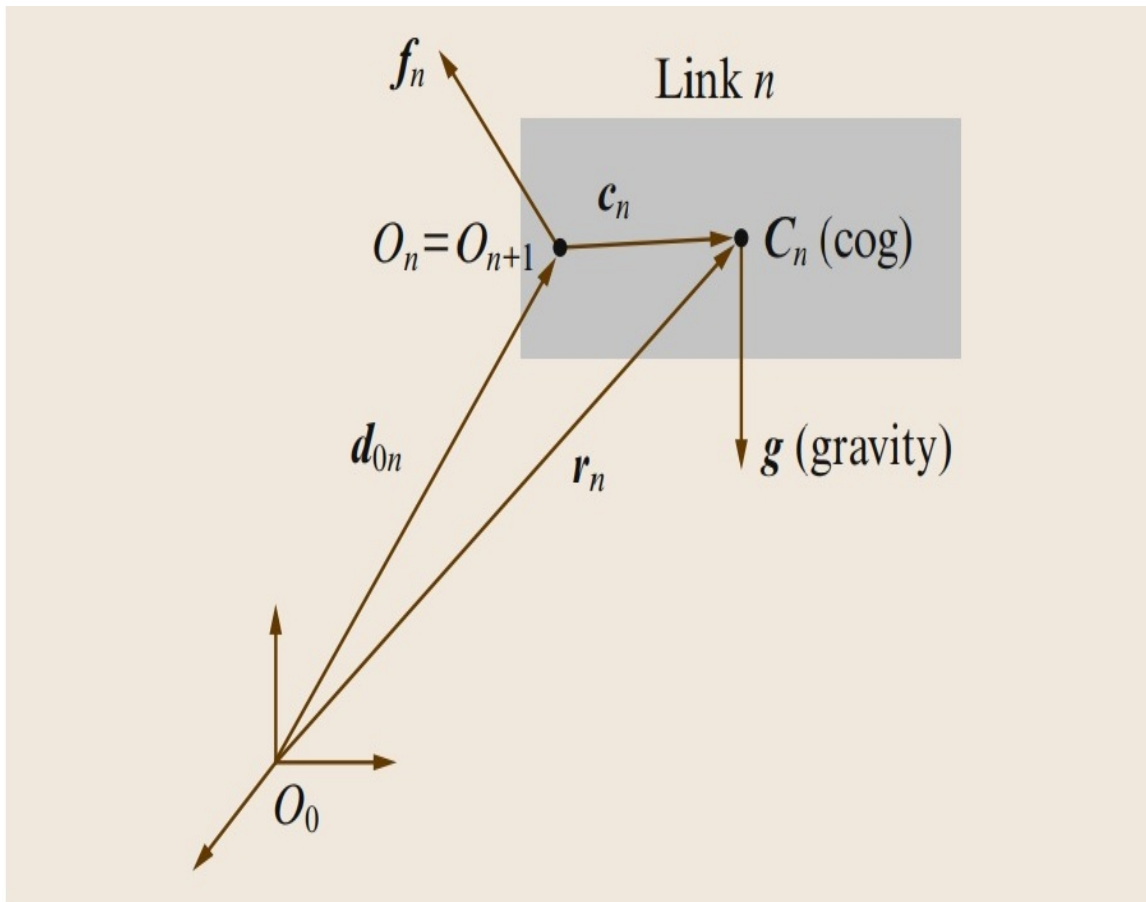
$$\tau - J^T(q)F_e = B(q)\ddot{q} + h(q, \dot{q}) + g(q) \quad (3.17)$$

όπου $h(q, \dot{q})$ το διάνυσμα των φυγόκεντρων όρων και των όρων Coriolis διάστασης $n \times 1$, $g(q)$ το διάνυσμα των βαρυτικών όρων και $J^T(q)F_e$ οι εξωτερικές δυνάμεις και ροπές που ασκούνται στην κατασκευή σε περίπτωση που αυτή έρχεται σε επαφή με το εξωτερικό περιβάλλον. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι η Εξ. (3.17) εμπεριέχει μόνο τους όρους που οφείλονται στις δυνάμεις που προκύπτουν από τη μηχανική άκαμπτου σώματος. Πιο συγκεκριμένα, για λόγους απλότητας, αγνοούνται οι ιξώδεις ροπές τριβής και οι ροπές που οφείλονται λόγω των τριβών Coulomb.

Το δυναμικό μοντέλο χρησιμοποιείται για τη λύση δύο προβλημάτων, το πρόβλημα της ευθείας δυναμικής και το πρόβλημα της αντίστροφης δυναμικής. Το πρόβλημα της ευθείας δυναμικής αποτελεί τον υπολογισμό της επιτάχυνσης των μεταβλητών των αρθρώσεων \ddot{q} (κατά συνέπεια και q, \dot{q}) για δεδομένες ροπές. Από την άλλη, το πρόβλημα της αντίστροφης δυναμικής συνίσταται στον προσδιορισμό των ροπών των αρθρώσεων τ που απαιτούνται για τη δημιουργία της κίνησης που καθορίζεται από τις επιταχύνσεις, τις ταχύτητες και τις θέσεις των αρθρώσεων. Συνήθως, η επίλυση του προβλήματος της αντίστροφης δυναμικής είναι χρήσιμη για το σχεδιασμό της τροχιάς του ρομποτικού συστήματος και του συστήματος ελέγχου. Μόλις καθοριστεί μια τροχιά για το σύστημα καθορίζοντας τις επιθυμητές θέσεις, τις ταχύτητες και τις επιταχύνσεις (συνήθως ως αποτέλεσμα μιας αντίστροφης κινηματικής διαδικασίας), και δεδομένου ότι οι εξωτερικές δυνάμεις στο τερματικό εργαλείο δράσης είναι γνωστές - μετρήσιμες, τότε η αντίστροφη δυναμική επιτρέπει τον υπολογισμό των ροπών που πρέπει να εφαρμοστούν στις αρθρώσεις για να επιτευχθεί η επιθυμητή κίνηση. Αυτή η υπολογιστική διαδικασία αποδεικνύεται πολύ βοηθητική τόσο για την επαλήθευση της επιβαλλόμενης τροχιάς όσο και για την αντιστάθμιση διάφορων μη μοντελοποιημένων και μη γραμμικών όρων στο δυναμικό μοντέλο του συστήματος. Από την αριθμητική σκοπιά, το εν λόγω πρόβλημα διαχειρίζεται από τον αναδρομικό αλγόριθμο Newton - Euler με μεγάλη αποτελεσματικότητα και γραμμική υπολογιστική πολυπλοκότητα (μόλις $O(n)$) [15].

3.3: Δυνάμεις Και Ροπές Άκαμπτου Σώματος

Σε αυτή την ενότητα θα παρουσιαστεί μια μεθοδολογία για τον υπολογισμό των δυνάμεων και ροπών που ασκεί ένα άκαμπτο σώμα, το οποίο εκτελεί μια τροχιά, ένεκα των αδρανειακών χαρακτηριστικών του [11]. Το άκαμπτο σώμα ενδέχεται να αποτελεί μέρος μιας κινηματικής αλυσίδας ή ακόμα και να είναι ένα μεμονωμένο αντικείμενο το οποίο κρατείται από μία αρπάγη. Γενικά, τα αδρανειακά χαρακτηριστικά ενός σώματος περιγράφονται πλήρως από τις τιμές 10 παραμέτρων. Αυτές είναι η μάζα του, το κέντρο μάζας του ως προς ένα σύστημα συντεταγμένων αναφοράς και 6 στοιχεία του μητρώου αδρανείας, λαμβάνοντας υπόψιν τη συμμετρικότητα της μήτρας.



Σχήμα 3.02: Η θέση του κέντρου μάζας και οι δυνάμεις ενός ενδιάμεσου συνδέσμου μιας κινηματικής αλυσίδας

Σύμφωνα με το «Σχήμα 3.02», το κέντρο μάζας του σώματος αναφέρεται ως C_n , το οποίο βρίσκεται σε θέση $r_n = C_n - O_0$, θεωρώντας ως σύστημα αναφοράς το

σύστημα συντεταγμένων της βάσης, και σε θέση $c_n = C_n - O_n$ ως προς το σύστημα του ίδιου σώματος. Χρησιμοποιώντας ως σύστημα αναφοράς αυτό του σώματος n και ξεκινώντας από τη χωρική εξίσωση της κίνησης ενός άκαμπτου σώματος, η οποία ορίζει ότι οι δυνάμεις και ροπές που επιδρούν στο σώμα ισούνται με το ρυθμό μεταβολής της ορμής του:

$$f_n = \frac{d}{dt}(I_n v_n) = I_n a_n + u_n \times I_n u_n \quad (3.18)$$

Τότε, αντικαθιστώντας τις τιμές των χωρικών επιταχύνσεων, ταχυτήτων και αδρανειών:

$$f_n = \begin{bmatrix} \bar{I}_n & m_n S(c_n) \\ m_n S(c_n)^T & m_n \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \dot{\omega}_n \\ \dot{u}_n \end{bmatrix} + \begin{bmatrix} S(\omega_n) & S(u_n) \\ \mathbf{0} & S(\omega_n) \end{bmatrix} \times \begin{bmatrix} \bar{I}_n & m_n S(c_n) \\ m_n S(c_n)^T & m_n \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \omega_n \\ u_n \end{bmatrix}$$

Καταλήγοντας:

$$f_n = \begin{bmatrix} \bar{I}_n \dot{\omega}_n + S(\omega_n) \bar{I}_n \omega_n - m_n S(\ddot{d}_{0n}) c_n \\ m_n \ddot{d}_{0n} + S(\dot{\omega}_n) m_n c_n + S(\omega_n) S(\omega_n) m_n u_n \end{bmatrix} \quad (3.19)$$

Ενώ έχουμε χρησιμοποιήσει τις σχέσεις:

$$\dot{u}_n = \ddot{d}_{0n} - \omega_n \times u_n \quad (3.20)$$

Και η ποσότητα $S(p)$ αφορά τη skew - symmetric μήτρα:

$$S(p) = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \quad (3.21)$$

Στη συνέχεια, θα προσπαθήσουμε να εκφράσουμε την Εξ. (3.19) σε μητρική μορφή ορίζοντας τις σχέσεις:

$$L(p) = \begin{bmatrix} p_x & p_y & p_z & 0 & 0 & 0 \\ 0 & p_x & 0 & p_y & p_z & 0 \\ 0 & 0 & p_x & 0 & p_y & p_z \end{bmatrix} \quad (3.22)$$

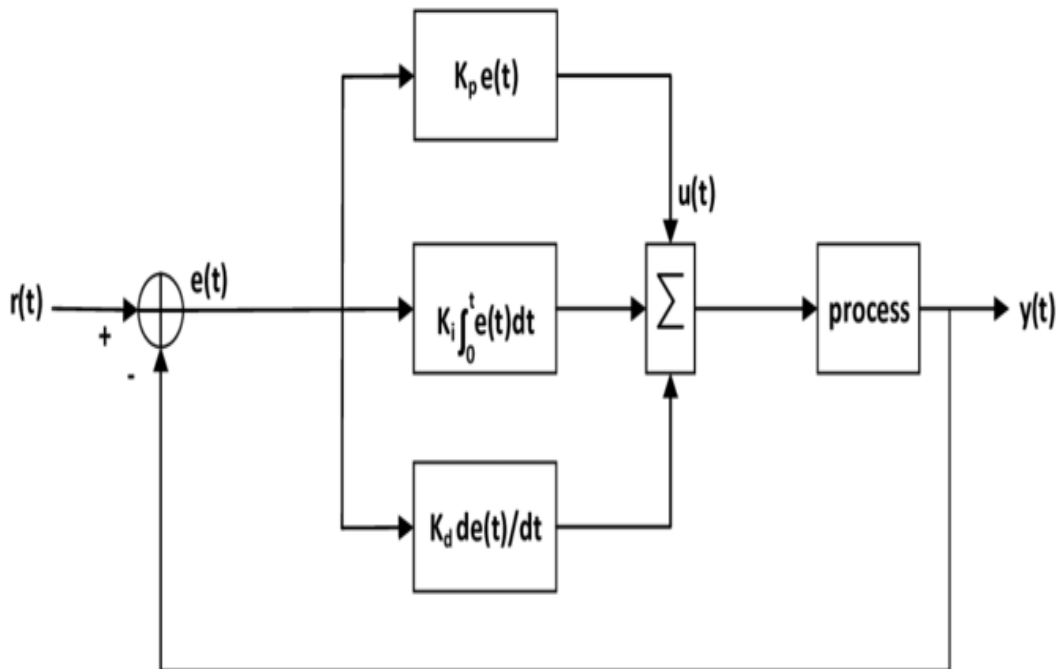
$$l(\bar{I}_n) = \begin{bmatrix} I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} \quad (3.23)$$

Τελικά, αντικαθιστώντας καταλήγουμε στην τελική σχέση που περιλαμβάνει τόσο τις δυνάμεις όσο και τις ροπές:

$$f_n = \begin{bmatrix} 0 & -S(\ddot{d}_{on}) & L(\dot{\omega}_n) + S(\omega_n)L(\omega_n) \\ \ddot{d}_{on} & S(\dot{\omega}_n) + S(\omega_n)S(\omega_n) & 0 \end{bmatrix}_{6 \times 10} \cdot \begin{bmatrix} m_n \\ m_n c_n \\ l(\bar{I}_n) \end{bmatrix}_{10 \times 1} \quad (3.24)$$

3.4: PID Έλεγχος

Σε αυτή την ενότητα γίνεται περιγραφή και ανάλυση της μακρά πιο διαδεδομένης τεχνικής ελέγχου ανατροφοδότησης, τον PID έλεγχο. Γενικά, ο PID ελεγκτής ή αλλιώς ελεγκτής τριών όρων, είναι ένας αντισταθμιστής σειράς που επεμβαίνει στον απ' ευθείας κλάδο του κλειστού συστήματος και ρυθμίζει το σήμα που οδηγεί τον επενεργητή (actuator) σε ένα σύστημα, λαμβάνοντας υπόψιν την απόκλιση (σφάλμα) της επιθυμητής τιμής της εξόδου από την πραγματική τιμή αυτής.



Σχήμα 3.03: Δομικό διάγραμμα PID ελεγκτή

Στο παραπάνω σχήμα απεικονίζεται η γενική δομή του PID ελεγκτή όπου:

- process: η διαδικασία - σύστημα προς έλεγχο
- $r(t)$: η επιθυμητή έξοδος του συστήματος προς έλεγχο (ή σήμα αναφοράς)
- $y(t)$: η έξοδος του συστήματος προς έλεγχο
- $e(t)$: το σφάλμα της επιθυμητής τιμής της εξόδου από την πραγματική τιμή της εξόδου, δηλαδή $e(t) = r(t) - y(t)$. Το σφάλμα αυτό αποτελεί και την είσοδο στον αντισταθμιστή σειράς

Όπως είναι εύκολα αντιληπτό, ο στόχος του ελέγχου είναι να μηδενιστεί το σφάλμα $e(t)$, δηλαδή ιδανικά η έξοδος που επιθυμούμε να έχει το σύστημα κλειστού βρόχου να ταυτίζεται με την πραγματική έξοδο του. Η παραπάνω διαδικασία περιγράφεται από τη διαφορική εξίσωση του PID ελεγκτή:

$$u_t = k_p e(t) + k_i \int e(t) dt + k_d \frac{d[e(t)]}{dt} \quad (3.25)$$

Το σήμα αυτό θα σταλεί στο σύστημα προς έλεγχο και στη συνέχεια θα λάβουμε ένα νέο σήμα εξόδου. Η νέα έξοδος θα σταλεί ξανά πίσω στο αισθητήριο για να ανιχνεύσει και αυτό με τη σειρά του το νέο σήμα σφάλματος. Ο ελεγκτής θα πάρει αυτό το νέο σήμα σφάλματος και θα υπολογίσει ξανά την παράγωγο και το ολοκλήρωμα και η ίδια διαδικασία θα επαναλαμβάνεται αέναα.

Παράμετρος	Χρόνος Ανύψωσης	Υπερύψωση	Χρόνος Αποκατάστασης	Μόνιμο Σφάλμα
K_p	Μείωση	Αύξηση	Μικρή αλλαγή	Μείωση
K_i	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
K_d	Μικρή αλλαγή	Μείωση	Μείωση	Μικρή αλλαγή

Σχήμα 3.04: Επίδραση των κερδών στην απόκριση του συστήματος

Οι διάφορες σταθερές ονομάζονται κέρδη του ελεγκτή και κάθε μία επιτελεί διαφορετικό ρόλο. Η χρήση αναλογικού ελέγχου k_p έχει ως αποτέλεσμα την ελάττωση του χρόνου ανύψωσης (κάνει την απόκριση του συστήματος γρηγορότερη). Επίσης, ρυθμίζει το μόνιμο σφάλμα και τείνει να το μηδενίσει αλλά ποτέ δεν καταφέρνει να το μηδενίσει και παραμένει ένα μόνιμο σφάλμα ανάλογα με τον τύπο του συστήματος. Το κέρδος ολοκληρώματος k_i αυξάνει τον τύπο του συστήματος με συνέπεια να μηδενίζει το μόνιμο σφάλμα αλλά μεταβάλλει προς το χειρότερο τη μεταβατική συμπεριφορά καθώς προκαλεί την εμφάνιση ταλαντώσεων. Τέλος, το διαφορικό κέρδος k_d βελτιώνει τη μεταβατική συμπεριφορά απομακρύνοντας τους επικρατούντες πόλους από το φανταστικό άξονα, όμως χειροτερεύει την μόνιμη συμπεριφορά και καταστρέφει την ποιότητα

του σήματος εισόδου που εισάγεται στο σύστημα (λόγω της παραγωγίσιμης του θορύβου που υπάρχει στην έξοδο).

Το μεγάλο πλεονέκτημα του PID ελεγκτή είναι ότι βασίζεται στη μέτρηση της εξόδου της διεργασίας και όχι στη γνώση της υποκείμενης διεργασίας και για αυτό το λόγο χρησιμοποιείται κατά κόρον. Ρυθμίζοντας κατάλληλα τις τρεις παραμέτρους, μπορούμε να χειριστούμε τις ειδικές απαιτήσεις κάθε διεργασίας. Ωστόσο, η χρήση του αλγορίθμου του PID δεν εγγυάται ούτε τη βέλτιστη εκπλήρωση των απαιτούμενων προδιαγραφών της απόκρισης του συστήματος, ούτε εξασφαλίζει την ευστάθεια του.

3.5: Έλεγχος Εμπέδησης

Στη παρούσα ενότητα παρουσιάζεται η βασική μεθοδολογία μιας διαδεδομένης στρατηγικής ελέγχου δύναμης, του ελέγχου εμπέδησης ή ελέγχου σύνθετης μηχανικής αντίστασης. Ο έλεγχος εμπέδησης ανήκει στην κατηγορία του ενεργού ελέγχου αλληλεπίδρασης (Active Interaction Control) και πιο συγκεκριμένα αποτελεί έναν έμμεσο έλεγχο δύναμης (Indirect Force Control), με την έννοια ότι επιτυγχάνεται έλεγχος δύναμης διαμέσου του ελέγχου κίνησης (Motion Control), χωρίς να απαιτείται ένας αυτούσιος βρόχος ανάδρασης δύναμης.

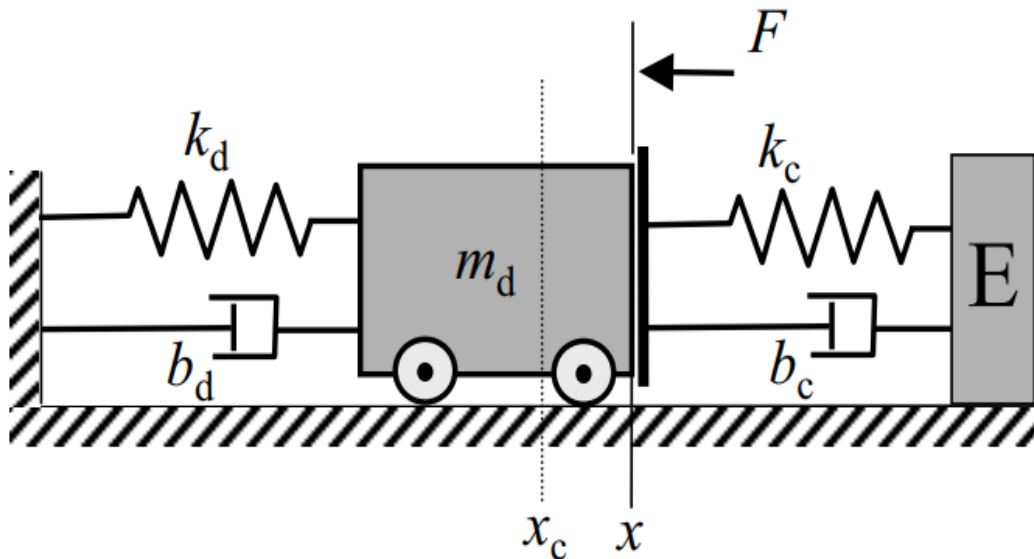


Σχήμα 3.05: Είσοδος - έξοδος του ελεγκτή εμπέδησης

Ο έλεγχος εμπέδησης αποτελεί επέκταση του ελέγχου ελαστικότητας, ο οποίος όμως εξετάζει μόνο τη στατική περίπτωση. Ουσιαστικά, ο ελεγκτής δέχεται σαν είσοδο κίνηση και εξάγει δυνάμεις. Ένα σημαντικό πλεονέκτημα της συγκεκριμένης αρχιτεκτονικής ελέγχου είναι ότι προσφέρει μια ενιαία προσέγγιση για όλες τις

διαφορετικές φάσεις του χειρισμού. Πιο συγκεκριμένα, τόσο η περίπτωση που ο ρομποτικός βραχίονας κινείται ελεύθερα στον χώρο, όσο και η περίπτωση της αλληλεπίδρασής του με το περιβάλλον (φάση επαφής) μπορούν να διαχειριστούν με επιτυχία από τον ίδιο νόμο ελέγχου χωρίς να είναι απαραίτητη η εναλλαγή μεταξύ διαφορετικών σχημάτων ελέγχου.

Η κεντρική ιδέα του ελεγκτή σύνθετης μηχανικής αντίστασης, όπως δηλώνει και η ονομασία του, είναι ότι καθορίζει τη δυναμική συμπεριφορά του ρομπότ σύμφωνα με ένα εικονικό μοντέλο μάζας - ελατηρίου - αποσβεστήρα. Η εργασία που επιτελείται πρέπει να περιγραφεί τόσο ως προς τη σχέση της κίνησης της ρομποτικής κατασκευής όσο και σε σχέση με τη δύναμη αλληλεπίδρασης με το περιβάλλον, καθώς και από τη σχέση μεταξύ αυτών των δύο μεγεθών. Ως εκ τούτου, κατά τη διάρκεια μιας εφαρμογής, ο ελεγκτής δεν έχει σχεδιαστεί ώστε να διαμορφώνει και να ρυθμίζει καταλλήλως την κίνηση του ρομποτικού βραχίονα ή τη δύναμη που ασκεί στην επιφάνεια επαφής, αλλά την εμπεδητική συμπεριφορά του, που αποτελεί μια ανταλλακτική σχέση μεταξύ των δύο. Εναλλακτικά, η δυναμική συμπεριφορά του συστήματος κλειστού βρόχου θα μοντελοποιείται από το μηχανικό ανάλογο του συστήματος με μία δεδομένη μάζα, μία απόσβεση και μία δυσκαμψία.

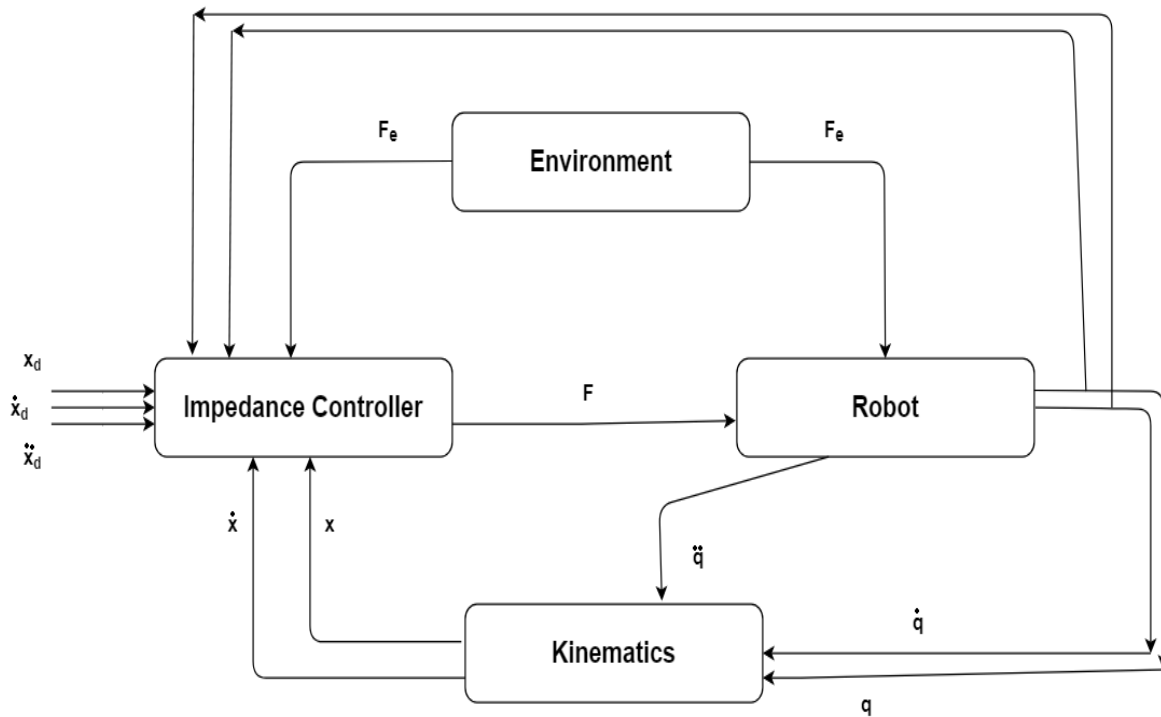


Σχήμα 3.06: Μηχανικό ανάλογο του συστήματος σε επαφή με το ελαστικό εξωτερικό περιβάλλον E

Άρα, σε μαθηματικούς όρους η επιθυμητή συμπεριφορά του συστήματος κλειστού βρόχου επιζητούμε να εκφράζεται από μια εξίσωση της μορφής:

$$M_m(\ddot{x} - \ddot{x}_d) + D_m(\dot{x} - \dot{x}_d) + K_m(x - x_d) = -F_e \quad (3.26)$$

όπου η μήτρα επιθυμητής αδράνειας M_m , η μήτρα επιθυμητής απόσβεσης D_m και η μήτρα επιθυμητής ακαμψίας είναι διαστάσεων $n \times n$, θετικά ορισμένες και συνήθως διαγώνιες. Επίσης, ο υποδείκτης d υποδηλώνει τα επιθυμητά μεγέθη.



Σχήμα 3.07: Δομικό διάγραμμα του ελέγχου εμπέδησης (α)

Χωρίς βλάβη της γενικότητας, οι όροι τριβών που επιδρούν στο ρομποτικό σύστημα μπορούν να παραληφθούν από το δυναμικό μοντέλο, το οποίο είναι από την Εξ. (3.17). Παράλληλα εκφράζοντας τη ροπή των αρθρώσεων συναρτήσει της Ιακωβιανής μήτρας:

$$\tau - J^T(q)F_e = B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.27)$$

$$\tau = J^T(q) \cdot F_\alpha \quad (3.28)$$

Συνδυάζοντας τις προηγούμενες σχέσεις και λύνοντας ως προς \ddot{q} :

$$\ddot{q} = B^{-1}(q) \cdot (J^T(q)F_\alpha - J^T(q)F_e - C(q, \dot{q})\dot{q} - g(q)) \quad (3.29)$$

Η επιτάχυνση, όμως, του τελικού εργαλείου δράσης στον καρτεσιανό χώρο περιγράφεται από την παρακάτω σχέση:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (3.30)$$

Συνδυάζοντας τις προηγούμενες σχέσεις και λύνοντας ως προς \ddot{x} :

$$\ddot{x} = JB^{-1}J^T \cdot (F_\alpha - F_e) - JB^{-1}C \cdot \dot{q} - JB^{-1}g + \dot{J}\dot{q} \quad (3.31)$$

Για να παραχθεί ο νόμος ελέγχου, ως προς τη δύναμη:

$$F_\alpha = (JB^{-1}J^T)^{-1} \cdot (\ddot{x} + JB^{-1}C \cdot \dot{q} + JB^{-1}g - \dot{J}\dot{q}) + F_e \quad (3.32)$$

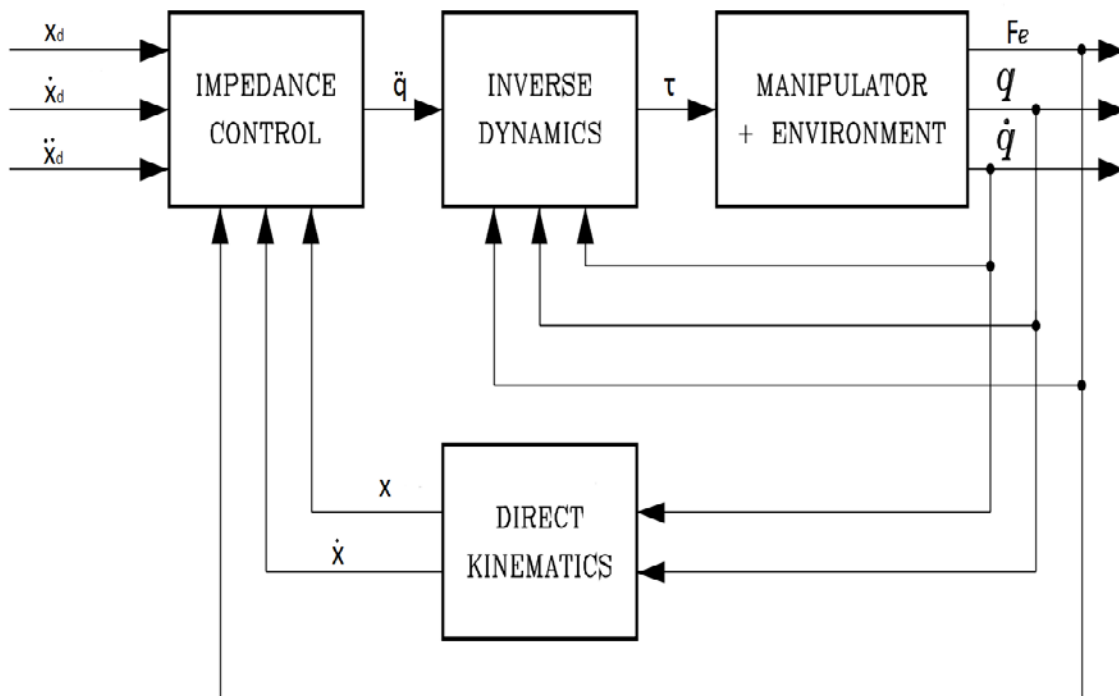
Ταυτόχρονα, όμως, από την Εξ. (3.26) ισχύει:

$$\ddot{x} = \ddot{x}_d + M_m^{-1}[K_m(x_d - x) + D_m(\dot{x}_d - \dot{x}) - F_e] \quad (3.33)$$

Αντικαθιστώντας την Εξ. (3.32) στην Εξ. (3.31) καταλήγουμε:

$$F_\alpha = (JB^{-1}J^T)^{-1}M_m^{-1} \cdot [K_m(x_d - x) + D_m(\dot{x}_d - \dot{x})] + (JB^{-1}J^T)^{-1} \cdot (JB^{-1}C \cdot \dot{q} + JB^{-1}g - j\dot{q}) + (I - (JB^{-1}J^T)^{-1}M_m^{-1})F_e + (JB^{-1}J^T)^{-1}\ddot{x}_d \quad (3.34)$$

Άρα, μπορούμε να χρησιμοποιήσουμε την Εξ. (3.28) για να εφαρμοστούν οι κατάλληλες ροπές στις αρθρώσεις.



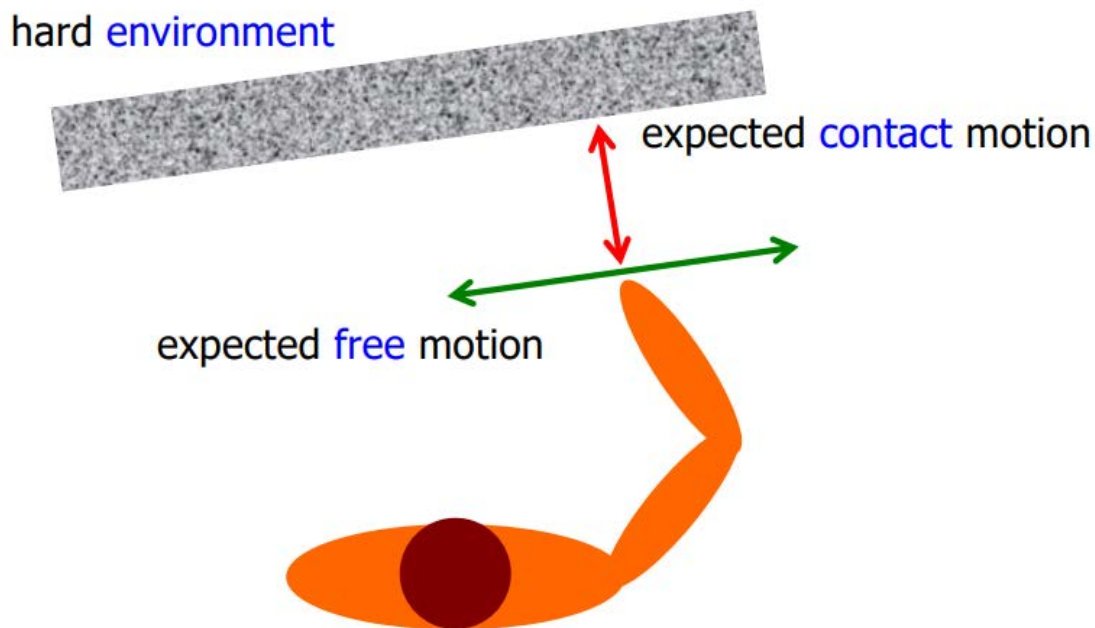
Σχήμα 3.08: Δομικό διάγραμμα του ελέγχου εμπέδησης (β)

Επίσης, υπάρχει και μια παραπλήσια παραλλαγή της μεθόδου. Χρησιμοποιώντας την Εξ. (3.30) και λύνοντας ως προς \ddot{q} :

$$\ddot{q} = J^+(\ddot{x} - j\dot{q}) \quad (3.35)$$

Στη συνέχεια, αντικαθιστούμε την Εξ. (3.33) στην παραπάνω σχέση και λύνοντας το πρόβλημα της αντίστροφης δυναμικής, όπως έχει περιγραφεί σε προηγούμενη ενότητα, προκύπτουν οι ροπές των αρθρώσεων.

Όπως είναι αντιληπτό από τα παραπάνω, για μια γενική εργασία υπό το πρίσμα του ελέγχου εμπέδησης, απαιτούνται έξι συνιστώσες δύναμης για να περιγραφεί πλήρως η εξωτερική δύναμη επαφής: τρία στοιχεία δύναμης, ένα για κάθε άξονα κίνησης, και τρεις ροπές αντίστοιχα. Για το λόγο αυτό, στην πλειοψηφία των εφαρμογών ελέγχου εμπέδησης (και γενικότερα δύναμης), συμπεριλαμβανομένων των βιομηχανικών εφαρμογών, τοποθετείται ένας αισθητήρας δύναμης - ροπής (Force - Torque Sensor) μεταξύ του κυρίου μέρους της ρομποτικής κατασκευής και του τερματικού εργαλείου δράσης. Σε μια τέτοια περίπτωση, το βάρος και η αδράνεια του εργαλείου, το οποίο είναι τοποθετημένο μεταξύ του αισθητήρα και του εξωτερικού περιβάλλοντος είτε θεωρείται αμελητέο είτε αντισταθμίζεται κατάλληλα από τις μετρήσεις δύναμης/ροπής του αισθητήρα.



Σχήμα 3.09: Συμπεριφορά του ρομπότ σε σχέση με το εξωτερικό περιβάλλον

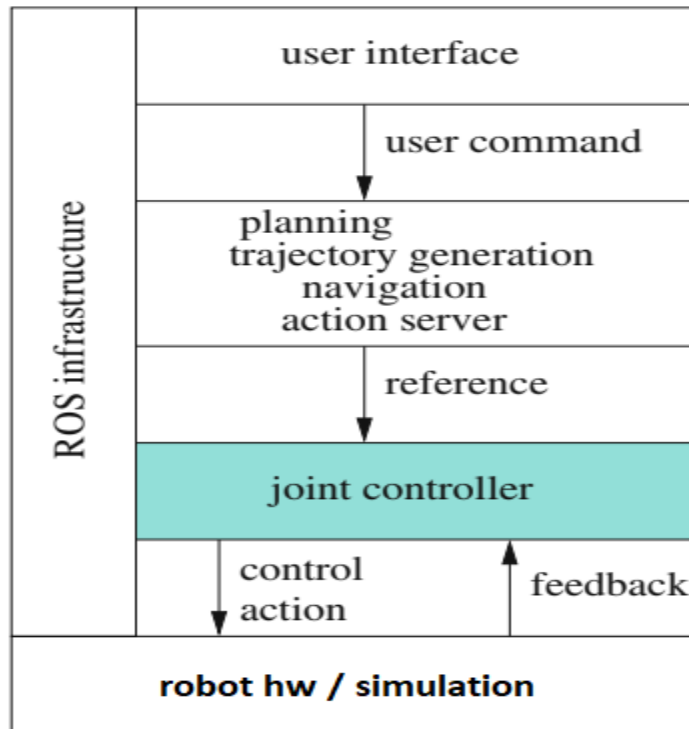
Η επιλογή καλών παραμέτρων σύνθετης αντίστασης διασφαλίζοντας μια ικανοποιητική συμπεριφορά δεν είναι εύκολη υπόθεση. Στην πραγματικότητα, η δυναμική του συστήματος κλειστού βρόχου είναι διαφορετική κατά τη διάρκεια της ελεύθερης κίνησης στο χώρο και κατά τη διάρκεια της αλληλεπίδρασης. Οι στόχοι ελέγχου είναι διαφορετικοί επίσης, δεδομένου ότι η παρακολούθηση και η απόρριψη των διαταραχών πρέπει να εξασφαλίζεται σε ελεύθερο χώρο, ενώ, κατά

τη διάρκεια της αλληλεπίδρασης, ο κύριος στόχος είναι η επίτευξη μιας κατάλληλης δυναμικής συμπεριφοράς συμμόρφωσης για την τελική αρπάγη αναλόγως της εφαρμογής. Επιπλέον, η δυναμική του ελεγχόμενου συστήματος κατά τη διάρκεια της αλληλεπίδρασης εξαρτάται και από τη δυναμική του περιβάλλοντος. Άρα, ο χρήστης πρέπει να επιλέξει κατάλληλα κέρδη, δηλαδή κατάλληλες τιμές για τις μήτρες επιθυμητής αδράνειας, απόσβεσης και ακαμψίας ανάλογα με το είδος της εφαρμογής, δεδομένου ότι πρέπει να ισορροπήσει μεταξύ της ακρίβειας κίνησης του χειριστή και διατηρώντας τις εξωτερικές δυνάμεις κάτω από ένα ανώτατο όριο. Πιο συγκεκριμένα, σε εφαρμογές όπου απαιτείται η επαφή με δύσκαμπτο περιβάλλον πρέπει να διατηρηθούν οι δυνάμεις επαφής εντός κάποιων ορίων και αυτό επιτυγχάνεται επιλέγοντας μεγάλες τιμές στα διαγώνια στοιχεία της μήτρας αδράνειας και μικρές τιμές αντίστοιχα στη μήτρα δυσκαμψίας. Αντιθέτως, κατά τη διάρκεια της κίνησης στον ελεύθερο χώρο επιλέγονται μεγάλες τιμές στη μήτρα δυσκαμψίας και μικρότερες στη μήτρα αδράνειας, ώστε να επιτευχθεί μεγάλη ακρίβεια με βάση την επιθυμητή τροχιά. Τέλος, και στις δύο περιπτώσεις οι τιμές της μήτρας απόσβεσης επιλέγονται ούτως ώστε να επιτευχθεί καλή μεταβατική συμπεριφορά.

Κεφάλαιο 4: Ελεγκτής Εμπέδησης

4.1: Παρουσίαση Του ROS Control

Το ROS Control είναι ένα API που έχει αναπτυχθεί από την κοινότητα του ROS για να επιτρέψει την πρόσβαση του προγραμματιστή στους διαφορετικούς ενεργοποιητές (actuators). Ουσιαστικά, πρόκειται για ένα μέτα - πακέτο, δηλαδή για μια συλλογή πακέτων που δρουν συνεργατικά, όπως έχει εξηγηθεί σε προηγούμενη ενότητα. Το κύριο κίνητρο για την ανάπτυξη του πλαισίου του ROS Control ήταν η έλλειψη επιπέδων επικοινωνίας πραγματικού χρόνου στο ROS. Επιπλέον, το ROS Control υλοποιεί λύσεις για τον κύκλο ζωής του ελεγκτή καθώς και τη διαχείριση των πόρων υλικού. Χρησιμοποιώντας αυτό το API, επιτυγχάνουμε το λογισμικό του ελεγκτή να διαχωρίζεται πλήρως από το λογισμικό του ενεργοποιητή. Για παράδειγμα, κάποιος θα μπορούσε να αναπτύξει μία στρατηγική ελέγχου και να την δοκιμάσει σε διαφορετικό υλικό χωρίς ουσιαστικές αλλαγές στο λογισμικό. Επίσης, θα μπορούσε κάποιος να ελέγξει την απόκριση και τη συμπεριφορά διαφορετικών τεχνικών ελέγχου με το ίδιο υλικό - ρομπότ για να διαπιστώσει την καταλληλότερη για τις ανάγκες του.



Σχήμα 4.01: Απλουστευμένη δομή του ROS Control

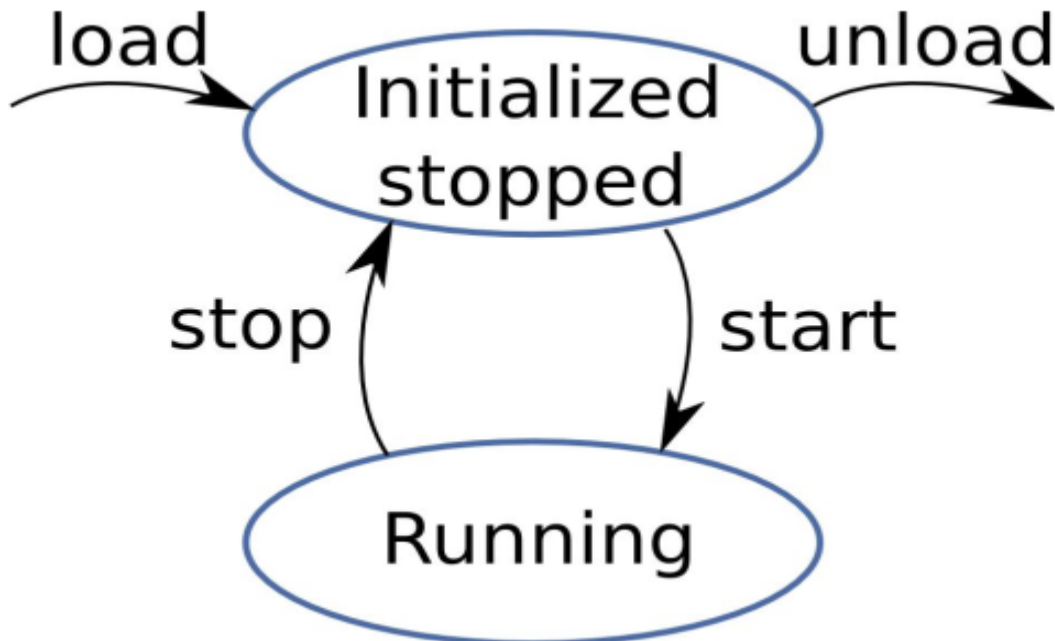
Το ROS Control έχει ορισμένα χαρακτηριστικά που το καθιστούν κυρίαρχο έναντι των εναλλακτικών του:

- λειτουργία σε πραγματικό χρόνο η οποία επιτρέπει την εκτέλεση κώδικα σε ταχύτητα εκατοντάδων Hz.
- μια απλή διεπαφή διαχειριστή, που δίνει πρόσβαση στους ενεργοποιητές και χειρίζεται τυχούσες διενέξεις πόρων.
- μια διασύνδεση ασφαλείας, η οποία γνωρίζει τους περιορισμούς του υλικού των αρθρώσεων (όρια λειτουργίας) και διασφαλίζει ότι οι εντολές που αποστέλλονται στους ενεργοποιητές βρίσκονται μεταξύ των ορίων τους.
- ένα σύνολο έτοιμων ελεγκτών προς χρήση.

Το ROS Control έχει υλοποιημένα διάφορα επιμέρους πακέτα που περιλαμβάνουν μεταξύ άλλων ελεγκτές, διαχειριστές ελεγκτών, διεπαφές υλικού, διεπαφές διαφορετικών μεταδόσεων. Πιο συγκεκριμένα, τα βασικότερα πακέτα είναι:

- *control_toolbox*: αυτό το πακέτο περιέχει απλά εργαλεία που χρησιμοποιούνται συχνά από τους ελεγκτές (πχ γεννήτριες σημάτων και θορύβου).
- *controller_interface*: υλοποιεί μια βασική κλάση για τη διασύνδεση με τους ελεγκτές, την κλάση Controller.
- *controller_manager*: υλοποιεί την κλάση Controller Manager για τη διαχείριση των ελεγκτών, η οποία περιέχει τις μεθόδους load, unload, start, stop.
- *hardware_interface*: υλοποιεί τις βασικές κλάσεις για τη διασύνδεση του ελεγκτή με το υλικό, την RobotHW και την JointHandle.
- *joints_limits_interface*: υλοποίηση της βασικής κλάσης για τον ορισμό των ορίων λειτουργίας των αρθρώσεων. Τα όρια των αρθρώσεων ορίζονται και μέσω των αρχείων URDF, απλά το συγκεκριμένο πακέτο προσφέρει επιπλέον και όρια επιταχύνσεων.
- *transmission_interface*: υλοποίηση των κλάσεων για την διεπαφή μετάδοσης διαφορετικών μεγεθών.
- *realtime_tools*: περιέχει ένα σύνολο από εργαλεία που χρησιμοποιούνται σε εργασίες πραγματικού χρόνου.
- *control_msgs*: περιέχει ένα σύνολο από εύχρηστου τύπου μηνύματα ROS

- *ros_controllers*: για την ακρίβεια πρόκειται για ένα μέτα - πακέτο που περιέχει διάφορους απλούς SISO (Single Input Single Output) ελεγκτές. Πιο συγκεκριμένα περιέχει ελεγκτές PID για τη δύναμη - ροπή, θέση και ταχύτητα. Επιπλέον, περιλαμβάνει τον *joint_state_controller*, ο οποίος εν αντιθέσει με ότι δηλώνει το όνομά του, λειτουργεί απλά ως publisher για τις μεταβλητές της θέσης και της ταχύτητας των αρθρώσεων του ρομπότ.



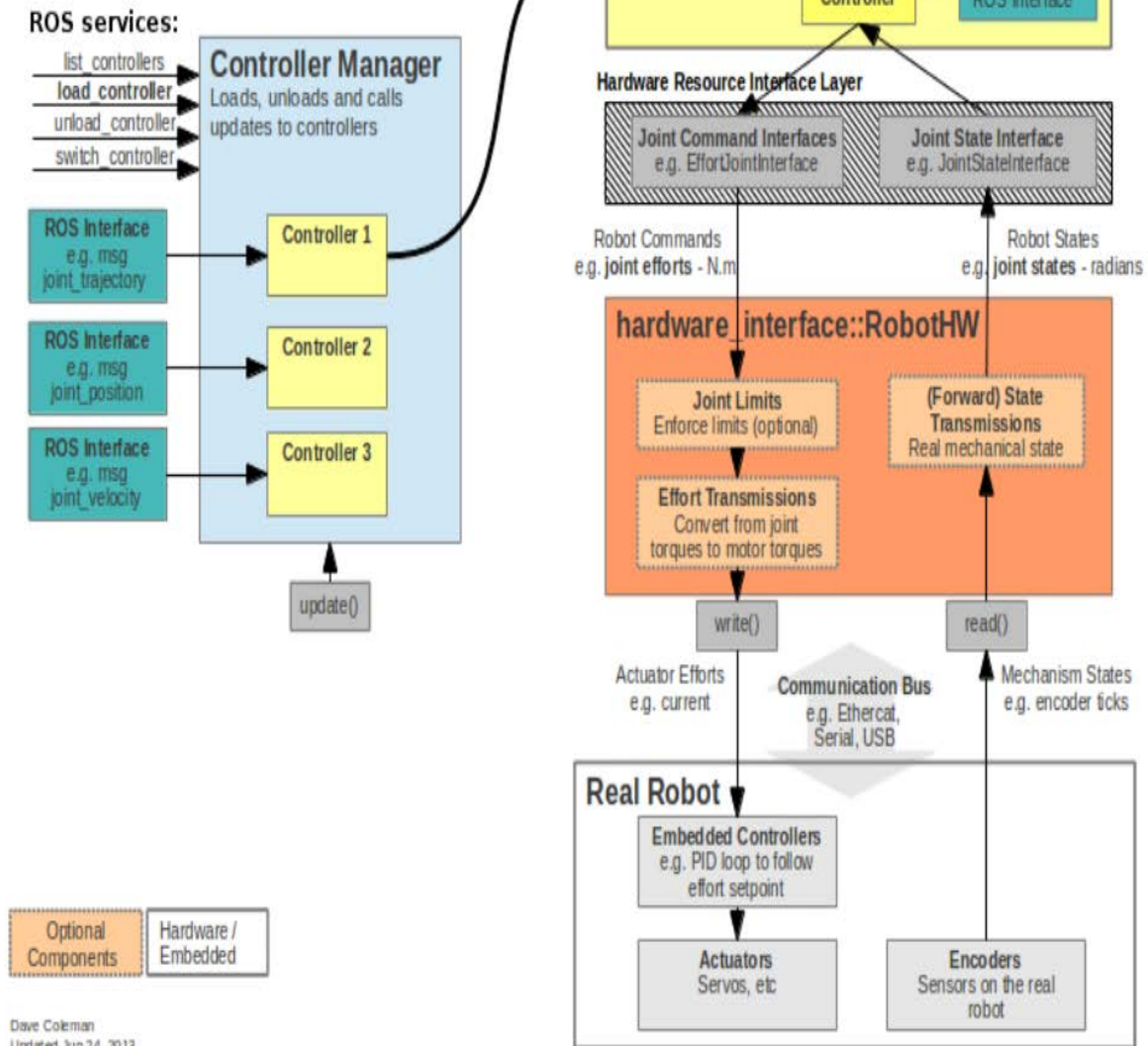
Σχήμα 4.02: Μηχανή πεπερασμένων καταστάσεων του controller_manager

Η ραχοκοκαλιά του ROS Control είναι το Στρώμα Αφηρημένου Υλικού (Hardware Abstraction Layer), το οποίο χρησιμεύει ως ένας σύνδεσμος για διαφορετικά ρομπότ, είτε προσομοιωμένα είτε και πραγματικά. Αυτή η αφαίρεση παρέχεται από την κλάση `hardware_interface::RobotHW`. Αντικείμενα αυτής της κλάσης μοντελοποιούν διάφορα μέρη του υλικού ενός ρομπότ όπως ηλεκτρικούς ή πνευματικούς επενεργητές, κωδικοποιητές και αισθητήρες δύναμης - ροπής. Επίσης, παρέχει τη δυνατότητα για τη σύνθεση ήδη υλοποιημένων αντικειμένων της `RobotHW`, λειτουργία που είναι ιδανική για την κατασκευή συστημάτων ελέγχου για ρομπότ με εξαρτήματα τα οποία προέρχονται από διαφορετικούς προμηθευτές, ο καθένας εκ των οποίων παρέχει το δική του συγκεκριμένο αντικείμενο `RobotHW`. Μέσω αυτού του πακέτου επιτυγχάνεται η εύκολη συντήρηση και η ανεξαρτησία των ελεγκτών από το υλικό. Επιπλέον, η κλάση `hardware_interface::JointHandle` συμβάλλει εξίσου στο χειρισμό του υλικού, αφού

παρέχει συναρτήσεις για την ανάγνωση των θέσεων, των ταχυτήτων και ροπών των αρθρώσεων καθώς και για την εφαρμογή ροπών σε αυτές. Αυτές οι συναρτήσεις προσπελαίνουν τα αντικείμενα της RobotHW μέσω δεικτών, η οποία υλοποιεί την πραγματική πρόσβαση στο πραγματικό ή προσομοιωμένο ρομπότ.

ROS Control

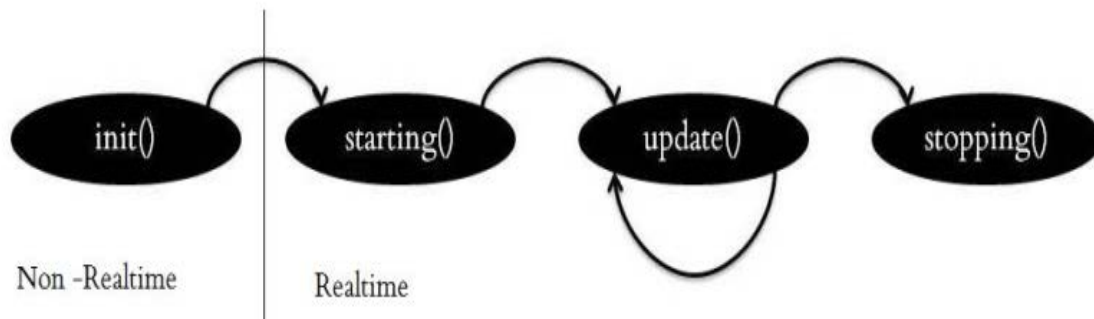
Data flow of controllers



Σχήμα 4.03: Αρχιτεκτονική του ROS Control [16]

Το πακέτο `controller_manager` είναι υπεύθυνο για τη διαχείριση του κύκλου ζωής των ελεγκτών. Ο κύκλος ζωής των ελεγκτών δεν είναι στατικός, αλλά δύναται να τροποποιηθεί κατά τη διάρκεια εκτέλεσης μέσω τυπικών ROS services που παρέχονται από το `controller_manager`. Άρα, αυτό το πακέτο φορτώνει plugins και ελεγκτές (οι ελεγκτές ορίζονται ως plugins στο ROS Control), τα αρχικοποιεί, τα εκκινεί και διακόπτει τη λειτουργία τους σύμφωνα με τις εντολές του χρήστη. Τέλος, διασφαλίζει ότι ο ελεγκτής δεν θα θέσει τιμές εκτός των ορίων των αρθρώσεων.

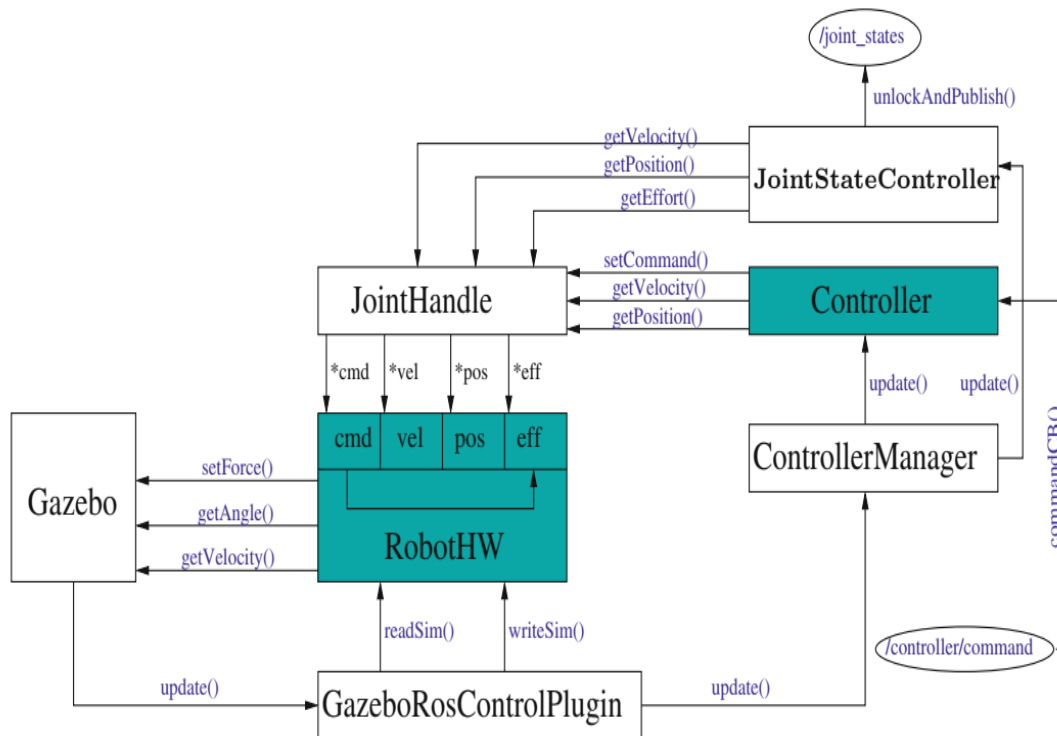
Ο βασικός ελεγκτής χαμηλού επιπέδου που θέλουμε να υλοποιήσουμε πρέπει να κληρονομήσει την βασική κλάση που ονομάζεται `controller_interface :: Controller` από το πακέτο `controller_interface`. Αυτή η κλάση περιέχει τέσσερις θεμελιώδεις συναρτήσεις: `init()`, `start()`, `update()` και `stop()`, οι οποίες χρησιμοποιούνται σε κάθε ελεγκτή. Η πρώτη συνάρτηση που εκτελείται όταν φορτώνεται ένας ελεγκτής είναι η `init()`. Η `init()` απλώς θα την αρχικοποιήσει τον ελεγκτή χωρίς να εκκινήσει τη λειτουργία του, ενώ η εν λόγω διαδικασία μπορεί να διαρκέσει οποιοδήποτε χρονικό διάστημα πριν από την εκκίνηση. Εκτελείται ακριβώς μία φορά, όπως και η μέθοδος που ακολουθεί, η `start()`. Η `start()` ουσιαστικά αποτελεί το πρώτο βήμα λειτουργίας του ελεγκτή. Στη συνέχεια, ακολουθεί η `update()`, η οποία είναι η σημαντικότερη μέθοδος. Η `update()` περιέχει το κύριο αλγόριθμο ελέγχου και εκτελείται επαναλαμβανόμενα με συγκεκριμένη συχνότητα. Τέλος, η `stop()` εκτελείται για μία φορά μόλις τερματιστεί η λειτουργία του ελεγκτή.



Σχήμα 4.04: Η σειρά ενεργειών της `controller_interface::Controller`

Το ROS δεν αποτελεί ένα σύστημα πραγματικού χρόνου. Οι κόμβοι του ROS δεν εκτελούνται σε πραγματικό χρόνο και τα ROS topics είναι ένας ασύγχρονος μηχανισμός επικοινωνίας, για τον οποίο δεν υπάρχουν εγγυήσεις για το χρόνο παράδοσης του μηνύματος. Οι κόμβοι και τα topics είναι επαρκή για διεργασίες υψηλότερου επιπέδου. Ωστόσο, για ελεγκτές χαμηλού επιπέδου υπάρχει ο βρόχος πραγματικού χρόνου του ROS, ο οποίος εκτελείται με συγκεκριμένη συχνότητα και όπου οι υπολογισμοί είναι ασφαλείς και η επικοινωνία

βασίζεται σε σύγχρονες κλήσεις συναρτήσεων. Φυσικά, αυτό δεν αρκεί για να διασφαλιστεί ο πραγματικός χρόνος, αφού αυτό εξαρτάται από το υποκείμενο λειτουργικό σύστημα. Στο προεπιλεγμένο Linux, η ακολουθία των εργασιών δεν βασίζεται σε αυστηρή προτεραιότητα από το χρήστη και μόνο ειδικά προνομιούχες διαδικασίες μπορούν να τύχουν την πολιτική προγραμματισμού τους σε πραγματικό χρόνο και μόνο με συγκεκριμένη αναβάθμιση του λειτουργικού συστήματος (PREEMP_RT kernel patch).



Σχήμα 4.05: Βρόχος πραγματικού χρόνου στο ROS

4.2: Η Βιβλιοθήκη Orocos KDL

Orocos είναι το ακρωνύμιο της φράσης Open Robot Control Software. Ουσιαστικά, πρόκειται για την ανάπτυξη ενός έργου με λογισμικό ανοιχτού κώδικα με κύριο στόχο την ανάπτυξη ενός γενικού σκοπού, δωρεάν λογισμικού πλαισίου για τον έλεγχο ρομποτικών κατασκευών [34]. Το έργο Orocos υποστηρίζει τέσσερις C++ βιβλιοθήκες: την Real - Time Toolkit, την Kinematics and Dynamics Library, την Bayesian Filtering Library και την Orocos Component Library. Στην παρούσα εργασία, θα χρησιμοποιηθεί η βιβλιοθήκη Kinematics and Dynamics Library (KDL),

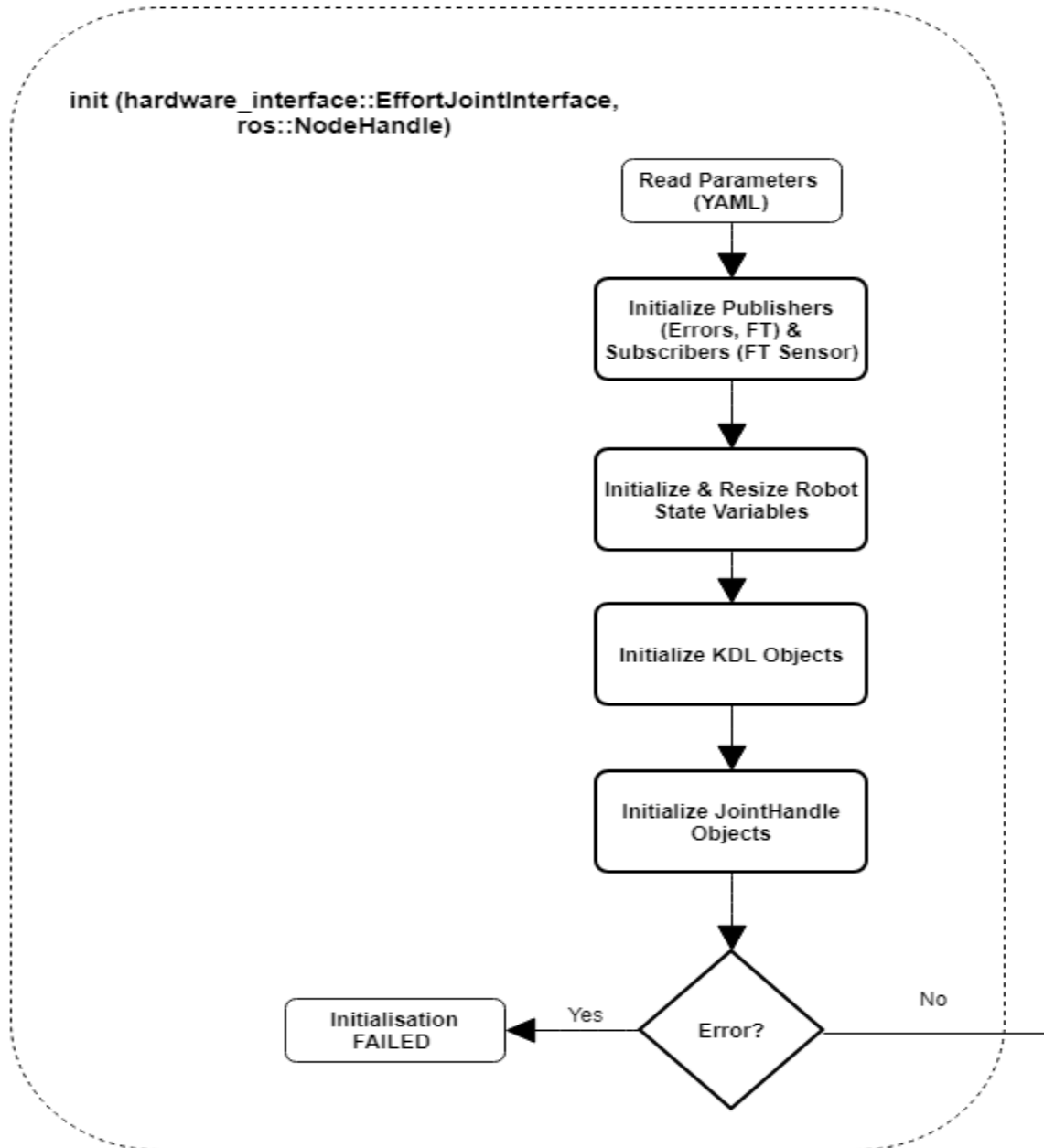
η οποία αποτελεί μια βιβλιοθήκη που είναι συμβατή με το ROS και επιτρέπει τον υπολογισμό σε πραγματικό χρόνο των κινηματικών και δυναμικών μοντέλων κινηματικών αλυσίδων, όπως ρομπότ, βιομηχανικά μοντέλα ανθρώπου, εργαλειομηχανές κ.α.. Επίσης παρέχει εξειδικευμένες επιμέρους βιβλιοθήκες για την γεωμετρική περιγραφή αντικειμένων, για την περιγραφή κινηματικών αλυσίδων διάφορων μορφών (σειριακές, παράλληλες, κινητές, ανθρωποειδείς) και για την περιγραφή της δυναμικής τους συμπεριφοράς. Στη συγκεκριμένη εργασία, θα γίνει χρήση κατά κόρον της κλάσης `ChainIdSolver_RNE`, η οποία επιλύει το πρόβλημα της αντίστροφης δυναμικής. Βασίζεται στον αναδρομικό αλγόριθμο των Newton - Euler για υπολογίσει τις ροπές των αρθρώσεων, ενώ δέχεται σαν είσοδο τη θέση, ταχύτητα και επιτάχυνση των αρθρώσεων καθώς και τις εξωτερικές δυνάμεις που επιδρούν σε κάθε σύνδεσμο της αλυσίδας. Επίσης, η KDL θα χρησιμοποιηθεί για την επίλυση του ευθύ κινηματικού μοντέλου, της Ιακωβιανής μήτρας της ρομποτικής κατασκευής καθώς και την διαφορίση αυτής, υπολογισμοί που απαιτούνται στην υλοποίηση του βρόχου ελέγχου της σύνθετης μηχανικής αντίστασης. Τέλος, θα χρησιμοποιηθεί το πακέτο `kdl_parser`, το οποίο μετατρέπει την περιγραφή της ρομποτικής διάταξης, η οποία είναι εκφρασμένη στο πρότυπο URDF, στη κινηματική αλυσίδα τύπου KDL για να χειριστεί στη συνέχεια αναλόγως.

4.3: Υλοποίηση Του Ελεγκτή Εμπέδησης

Σε αυτήν την ενότητα θα παρουσιαστεί η υλοποίηση ενός ελεγκτή εμπέδησης, βασιζόμενοι στα εργαλεία που έχουν παρουσιαστεί στα προηγούμενα κεφάλαια. Η υλοποίηση, όπως είναι λογικό, θα βασίζεται στο πλαίσιο του ROS Control και εν γένει του ROS, ενώ για τους υπολογισμούς των κινηματικών και δυναμικών μοντέλων θα χρησιμοποιηθεί όπου είναι απαραίτητο η βιβλιοθήκη Orocos KDL. Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί είναι η C++. Όπως έχει αναφερθεί, οι ROS Controllers αναπτύσσονται ως plugins που περιέχουν μια βασική κλάση με τέσσερις μεθόδους (`init()`, `starting()`, `update()`, `stopping()`).

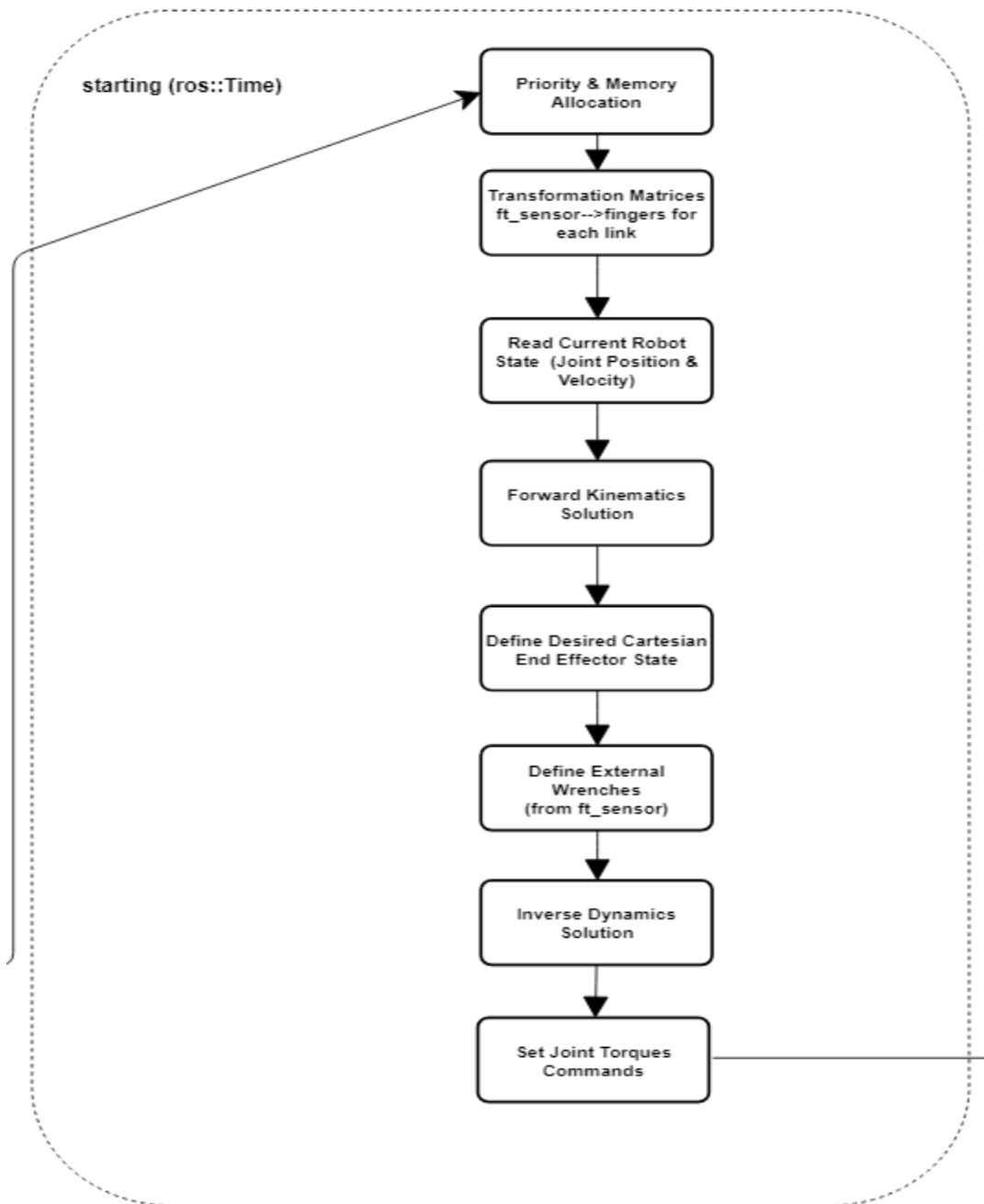
Η διαδικασία εκτέλεσης του κώδικα ξεκινάει μόλις επιτευχθεί η φόρτωση του μέσω ενός κόμβου τύπου «spawner» του πακέτου `controller_manager`. Φυσικά, αφού κληθεί και εκτελεστεί ο δημιουργός (`constructor`) της κλάσης εμφανίζοντας κατάλληλο μήνυμα, η πρώτη μέθοδος που εκτελείται είναι η `init()`. Η `init()` επιστρέφει μία τιμή τύπου `boolean` και δέχεται ως ορίσματα εισόδου έναν δείκτη τύπου `hardware_interface::EffortJointInterface` και ένα τύπου `ros::NodeHandle`. Η χρήση του `EffortJointInterface` είναι μονόδρομος, αφού ο ελεγκτής εμπέδησης αποτελεί έναν ελεγκτή δύναμης - ροπής και οι αρθρώσεις πρέπει να δέχονται εντολές ροπής. Η μέθοδος `init()` εκτελείται και όπως υποδηλώνει το όνομά της

αρχικοποιεί κάποια αντικείμενα και μεταβλητές που θα χρησιμοποιηθούν στην πορεία. Πιο συγκεκριμένα, αρχικά διαβάζει τις τιμές κάποιων παραμέτρων από τον διακομιστή παραμέτρων και αρχικοποιεί τους publishers και subscribers, τις μεταβλητές κατάστασης του ρομπότ, τα αντικείμενα της KDL και τελικά τα αντικείμενα της JointHandle για να καταστεί εφικτή η αλληλεπίδραση με τις αρθρώσεις. Η `init()` εκτελείται μόνο μία φορά και αν όλα βαίνουν καλώς επιστρέφεται αληθής τιμή και η εκτέλεση συνεχίζεται. Σε διαφορετική περίπτωση, η διαδικασία αποτυγχάνει και τερματίζεται.

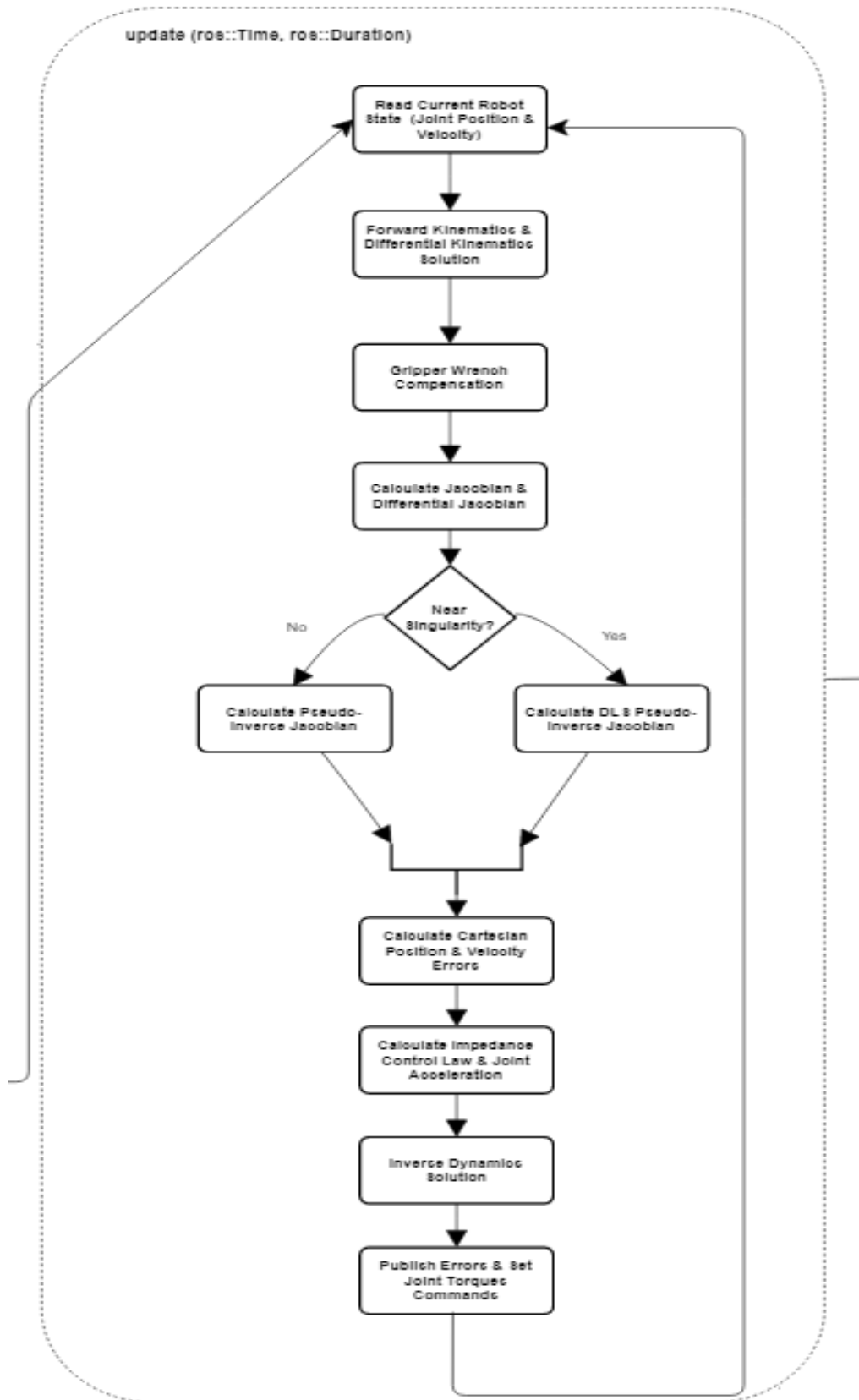


Σχήμα 4.06: Δομικό διάγραμμα της `init()`

Μετά την `init()`, ακολουθεί η εκτέλεση της `starting()`. Η `starting()` είναι τύπου `void` και δέχεται ως όρισμα εισόδου ένα αντικείμενο `ros::Time`. Αρχικά, καθορίζεται η υψηλή προτεραιότητα εκτέλεσης της στο λειτουργικό σύστημα Linux και δεσμεύεται η απαιτούμενη μνήμη. Στη συνέχεια, διαβάζεται η παρούσα κινηματική κατάσταση της αλυσίδας, δηλαδή η θέση και η ταχύτητα των αρθρώσεων, και υπολογίζονται οι μήτρες μετασχηματισμού από το σημείο του αισθητήρα δύναμης προς κάθε σύνδεσμο που έπεται. Αυτοί οι υπολογισμοί είναι χρήσιμοι για τη συνέχεια.



Σχήμα 4.07: Δομικό διάγραμμα της `starting()`

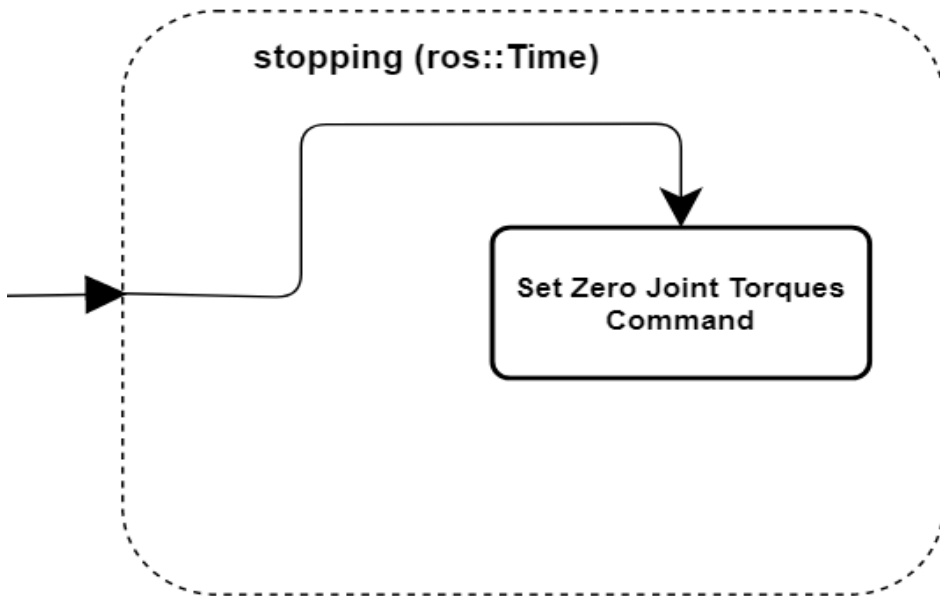


Σχήμα 4.08: Δομικό διάγραμμα της update()

Έπειτα, λύνεται το ευθύ κινηματικό μοντέλο με τη βοήθεια της κλάσης `KDL::ChainFKSolverPos` και θέτεται ως επιθυμητή καρτεσιανή θέση η υπάρχουσα, η οποία αποτελεί και τη θέση αρχικοποίησης της ρομποτικής διάταξης. Η επιθυμητή ταχύτητα και επιτάχυνση θέτονται μηδενικές. Στη συνέχεια, ορίζονται οι εξωτερικές δυνάμεις και ροπές που ασκούνται σε κάθε σύνδεσμο της αλυσίδας μέσω του αισθητήρα δύναμης - ροπής και λύνεται το πρόβλημα της αντίστροφης δυναμικής με τη βοήθεια της `ChainIdSolver_RNE` για μηδενική ταχύτητα και επιτάχυνση των αρθρώσεων. Τέλος, οι ροπές που έχουν υπολογιστεί δίνονται ως εντολές στις αρθρώσεις.

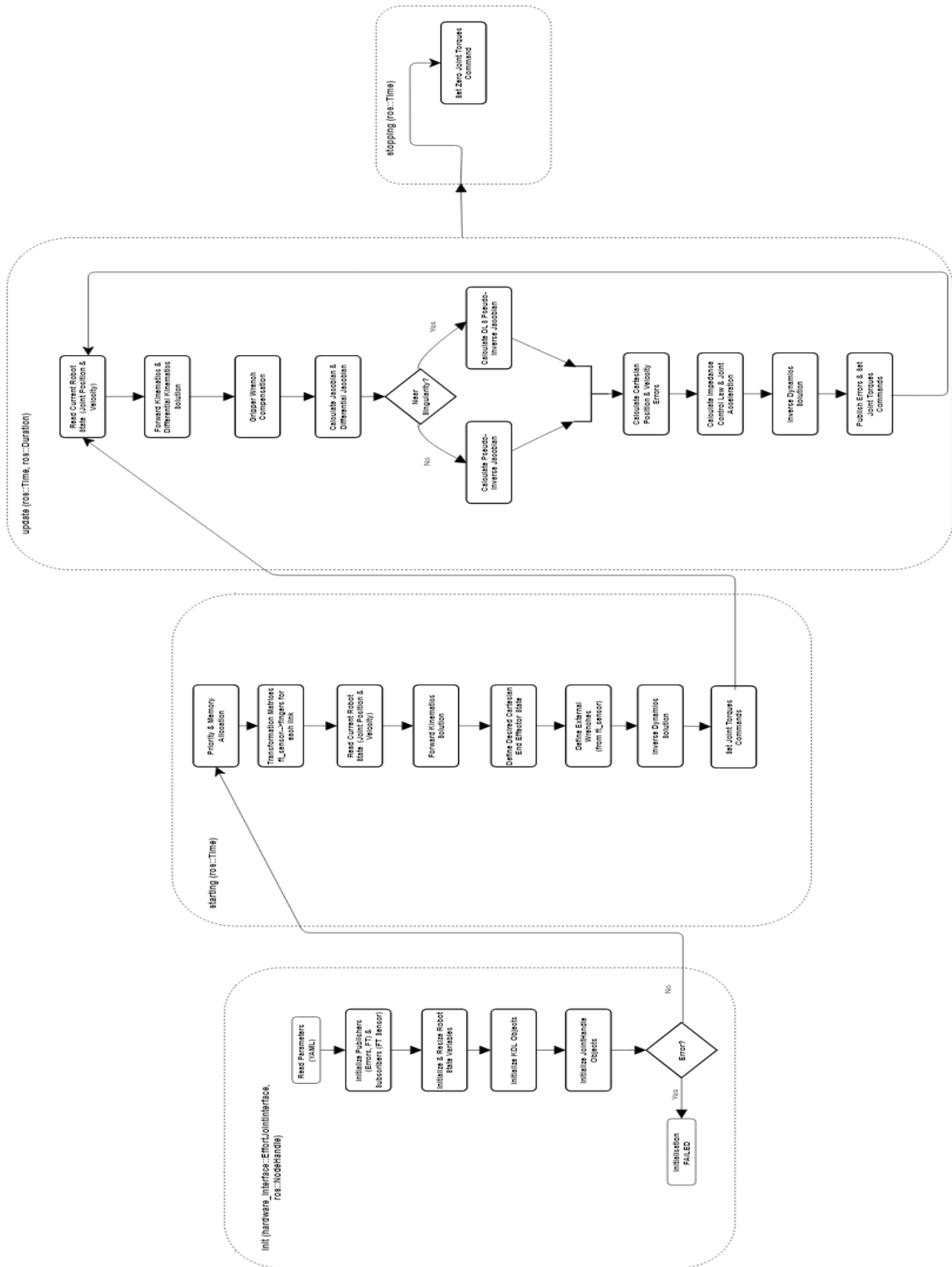
Μετά την `starting()`, ακολουθεί η εκτέλεση της `update()`. Η `update()` είναι τύπου `void` και δέχεται ως όρισμα εισόδου ένα αντικείμενο `ros::Time` και ένα `ros::Duration`. Η πρώτη της λειτουργία είναι να προσπελαύνει την παρούσα κινηματική κατάσταση της αλυσίδας, δηλαδή η θέση και η ταχύτητα των αρθρώσεων και να υπολογίζει το ευθύ κινηματικό και διαφορικό μοντέλο. Έπειτα, υπολογίζονται οι δυνάμεις και ροπές των συνδέσμων που έπονται του αισθητήρα δύναμης - ροπής χρησιμοποιώντας τις μήτρες μετασχηματισμού από την `starting()` και διορθώνεται η μέτρηση. Σημειώνεται ότι δεν μεσολαβούν αρθρώσεις ενδιάμεσα των συνδέσμων και κατά συνέπεια οι συγκεκριμένες μήτρες είναι σταθερές. Στη συνέχεια, υπολογίζεται η Ιακωβιανή μήτρα καθώς και η παράγωγος της με τις κλάσεις `KDL::ChainJntToJakSolver` και `KDL::ChainJntToJakDotSolver` αντίστοιχα. Επίσης, εξετάζεται αν βρισκόμαστε στη γειτονιά κάποιας ιδιόμορφης διάταξης μέσω υπολογισμού της τάξης της Ιακωβιανής μήτρας διατηρώντας υψηλό κατώφλι ευαισθησίας και υπολογίζεται η ψεύδο - αντίστροφη της με την Εξ. 3.11 ή Εξ. 3.12 αντίστοιχα. Στη συνέχεια, υπολογίζονται τα σφάλματα θέσης και ταχύτητας με βάση τις επιθυμητές τιμές που έχουν καθοριστεί και υπολογίζεται ο νόμος ελέγχου σύμφωνα με την Εξ. 3.33. Έπειτα, μετατρέπεται η επιτάχυνση από το χώρο εργασίας στο χώρο των αρθρώσεων μέσω της Εξ. 3.35 και λύνεται το πρόβλημα της αντίστροφης δυναμικής. Τελικά, εκδίδονται τα υπολογισθέντα σφάλματα σε κατάλληλο `topic` και οι ροπές που έχουν υπολογιστεί δίνονται ως εντολές στις αρθρώσεις. Όπως έχει αναφερθεί, η `update()` εκτελείται συνεχώς με σταθερή και καθορισμένη συχνότητα για όσο διαρκεί η λειτουργία του συστήματος. Τέλος, αξίζει να αναφέρουμε ότι σε περίπτωση που απαιτούνταν το τερματικό εργαλείο δράσης να ακολουθήσει κάποια τροχιά και όχι να ισορροπεί σε σταθερή θέση, θα μπορούσε να επιτευχθεί εύκολα θέτοντας διαφορετικές τιμές στις επιθυμητές θέσεις - ταχύτητες.

Μόλις ο χρήστης δώσει εξωγενώς μια εντολή τερματισμού λειτουργίας, η εκτέλεση της μεθόδου `update()` τερματίζεται και καλείται η `stopping()`. Η μέθοδος `stopping()` είναι τύπου `void` και δέχεται ως όρισμα εισόδου ένα αντικείμενο `ros::Time`. Εκτελείται αποκλειστικά μία φορά και η λειτουργία της είναι να θέσει μηδενικές τιμές στις ροπές των αρθρώσεων.



Σχήμα 4.09: Δομικό διάγραμμα της stopping()

Εν κατακλείδι, αξίζει να αναφέρουμε τη λειτουργία εισαγωγής δεδομένων στον ελεγκτή από το διακομιστή παραμέτρων κατά τη διάρκεια εκτέλεσης της `init()`. Πριν τη φόρτωση και εκτέλεση του ελεγκτή, φορτώνονται οι τιμές κάποιων παραμέτρων στο διακομιστή παραμέτρων του ROS μέσω ενός αρχείου `yaml`. Η `yaml` είναι μία γλώσσα αναπαράστασης σειριακών δεδομένων σε μορφή κατανοητή από τον άνθρωπο, όχι πολύ διαφορετική από την `xml`. Πιο συγκεκριμένα, ο ελεγκτής εμπέδησης δέχεται τιμές παραμέτρων που αφορούν τα ονόματα των αρθρώσεων που πρόκειται να ελεγχθούν, το όνομα της ρομποτικής διάταξης, τα ονόματα του πρώτου και τελευταίου συνδέσμου της διάταξης, το όνομα του συνδέσμου στο οποίο εκφράζονται οι μετρήσεις του αισθητήρα δύναμης - ροπής, το όνομα του `topic` από το οποίο δέχεται ο ελεγκτής αυτές τις μετρήσεις και τέλος τις τιμές των κερδών του νόμου ελέγχου. Αυτή η λειτουργία είναι πολύ σημαντική διότι κατά αυτόν τον τρόπο επιτυγχάνεται η προσαρμοστικότητα του ελεγκτή σε κάθε ρομποτικό σύστημα χωρίς να απαιτείται η παραμικρή αλλαγή στον πηγαίο κώδικα. Για να εφαρμοστεί ο ελεγκτής σε διαφορετική ρομποτική διάταξη αρκεί η φόρτωση ενός διαφορετικού `yaml` αρχείου με τα απαραίτητα δεδομένα, με την υπόλοιπη διαδικασία να παραμένει αυτοματοποιημένη.

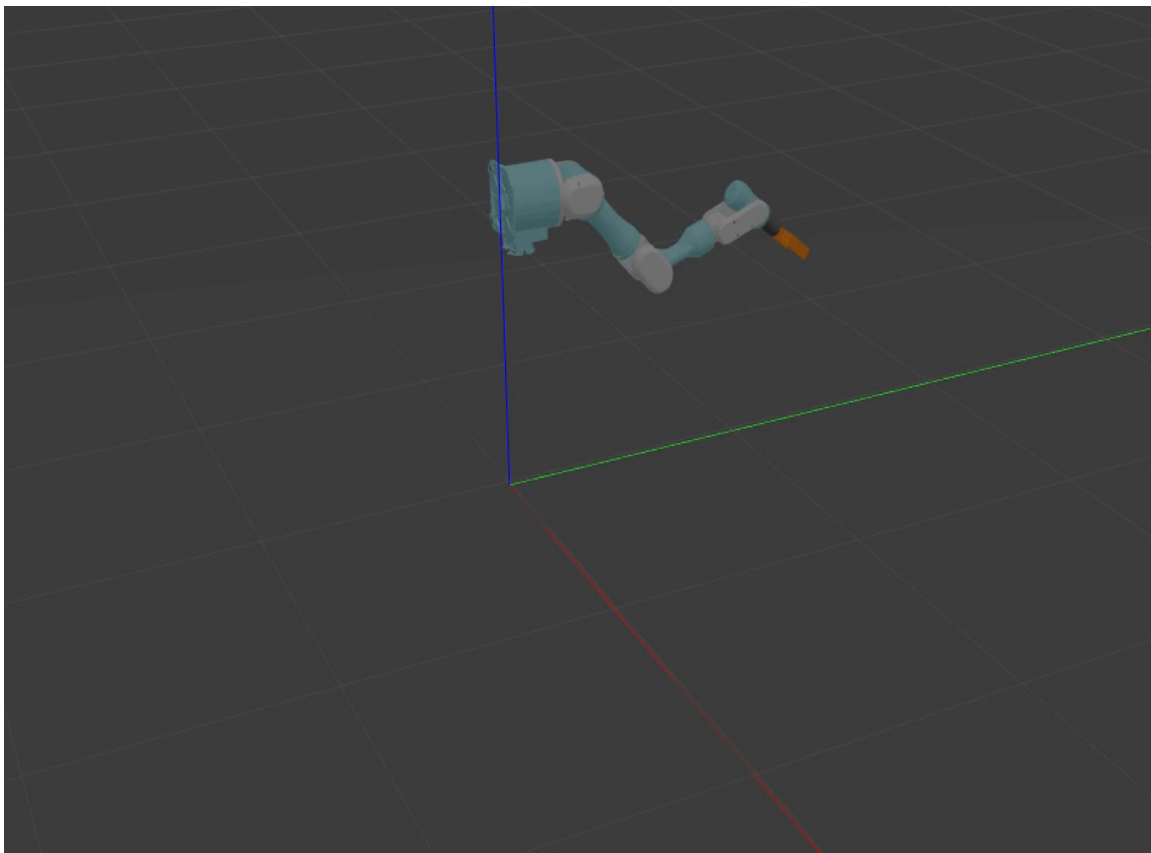


Σχήμα 4.10: Δομικό διάγραμμα του ελεγκτή

Κεφάλαιο 5: Ρομποτικοί Βραχίονες

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα κατασκευαστικά και λειτουργικά χαρακτηριστικά των ρομποτικών διατάξεων που θα χρησιμοποιηθούν στις μετέπειτα προσομοιώσεις.

5.1: Ο Βραχίονας Mitsubishi PA10



Σχήμα 5.01: Ο βραχίονας PA10 στο γραφικό περιβάλλον του Gazebo

Το Mitsubishi PA10 είναι ένα ρομπότ επτά βαθμών ελευθερίας, το οποίο έχει κατασκευαστεί από την εταιρεία Mitsubishi Heavy Industries. Οι αρθρώσεις του ρομπότ είναι περιστροφικού τύπου και περιλαμβάνουν ως επενεργητές τριφασικούς σέρβο - κινητήρες εναλλασσόμενου ρεύματος και αρμονικά κιβώτια ταχυτήτων. Το βάρος του βραχίονα είναι περίπου 40 kg συνοδευόμενα από 18 kg

της μονάδας ελέγχου, ενώ έχει τη δυνατότητα μεταφοράς φορτίου έως και 10 kg. Ο συγκεκριμένος βραχίονας χρησιμοποιείται κατά κόρον στην έρευνα για νέες προσεγγίσεις στον τομέα της βιομηχανικής ρομποτικής, εστιάζοντας κυρίως σε εργασίες διαχείρισης της εφοδιαστικής αλυσίδας και παραγωγής. Η ανοιχτή αρχιτεκτονική του ρομπότ (σε υλικό και λογισμικό) παρέχει τη δυνατότητα ελέγχου και τροποποιήσεων οποιασδήποτε πτυχής της συμπεριφοράς του ρομπότ καθώς και συμπερίληψη νέων πληροφοριών - μετρήσεων αισθητήρων στον βρόχο ελέγχου. Νέες εξελίξεις στις τεχνικές οπτικής εξυπηρέτησης (οπτικές πληροφορίες που χρησιμοποιούνται ως σήματα ανατροφοδότησης για τον έλεγχο της κίνησης του ρομπότ), μέθοδοι ελέγχου αλληλεπίδρασης για ασφαλή και ισχυρή αλληλεπίδραση με το περιβάλλον και η χρήση πληροφοριών RFID για ευφυή χειρισμό πακέτων είναι μερικές από τις λύσεις που έχουν εφαρμοστεί και δοκιμάζονται σε αυτήν τη διάταξη. Ως τελικός στόχος είναι τα αποτελέσματα αυτής της έρευνας να μεταφερθούν στη βιομηχανική παραγωγή.

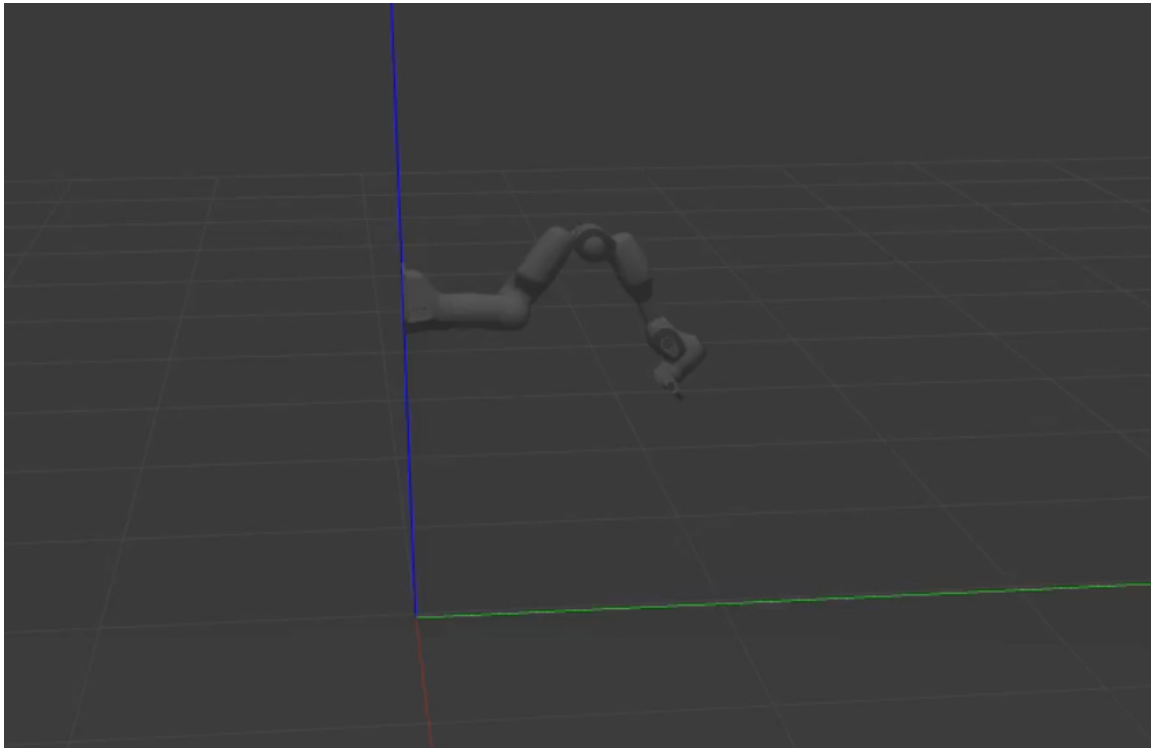
5.2: Ο Βραχίονας Panda

Το Panda είναι ένα ρομπότ επτά βαθμών ελευθερίας, το οποίο έχει κατασκευαστεί από την εταιρεία Franka Emika. Η Franka Emika είναι μια εταιρεία με έδρα το Μόναχο που ιδρύθηκε από τον διακεκριμένο καθηγητή ρομποτικής και τεχνητής νοημοσύνης Sami Haddadin. Το Panda έχει σχεδιαστεί για να είναι ελαφρύ και περιλαμβάνει περισσότερους από εκατό αισθητήρες. Χαρακτηρίζεται από τις ιδιότητες ενός κλασσικού άκαμπτου βιομηχανικού ρομπότ με επαναληψιμότητα θέση μικρότερη του +/- 0,1 mm (ISO 9283) και μια αμελητέα απόκλιση διαδρομής ακόμη και σε ταχύτητες του τερματικού εργαλείου δράσης μέχρι 2 m/s. Αυτό επιτρέπει ακριβή, στιβαρή και γρήγορη εκτέλεση των διεργασιών. Ο συγκεκριμένος ρομποτικός βραχίονας ενδείκνυται και βρίσκει εφαρμογή σε πολλούς τομείς όπως οι βιομηχανικές γραμμές παραγωγής, σε ερευνητικά εργαστήρια και σε κέντρα ιατρικής περίθαλψης. Πρόκειται για ένα βραχίονα με επτά βαθμούς ελευθερίας, αισθητήρες θέσης υψηλής ανάλυσης και αισθητήρες ροπής υψηλής ακρίβειας. Οι αρθρώσεις του ρομπότ είναι περιστροφικού τύπου και περιλαμβάνουν ως επενεργητές κινητήρες συνεχούς ρεύματος χωρίς ψήκτες. Το Panda μπορεί να χρησιμοποιηθεί για την εφαρμογή δυνάμεων με ελάχιστο όριο 0.05 N προκειμένου να εκτελέσει ευαίσθητες και ακριβείς εργασίες όπως η τοποθέτηση αντικειμένων και βίδωμα εξαρτημάτων. Η συνεχής και ακριβής τελειοποίηση των δυνάμεων αποτελεί επίσης προϋπόθεση για εφαρμογές όπως η στίλβωση και η λείανση. Επίσης, το Panda διαθέτει ρυθμιζόμενες λειτουργίες καθοδήγησης οι οποίες αντισταθμίζουν τη βαρύτητα και την τριβή ώστε να διασφαλιστεί η ομαλή φυσική αλληλεπίδραση μεταξύ του ανθρώπου και της κατασκευής. Οι εξελιγμένοι αισθητήρες του σε συνδυασμό με τους αλγορίθμους ελέγχου και το εσωτερικό

μοντέλο επιτρέπουν την έγκαιρη ανίχνευση και αντίδραση σε ανεπιθύμητες συγκρούσεις μέσα σε χιλιοστά του δευτερολέπτου. Τέλος, η ρομποτική διάταξη έχει βάρος περίπου 18 kg συνοδευόμενα από 7 kg της μονάδας ελέγχου ενώ έχει τη δυνατότητα μεταφοράς φορτίου έως 3 kg.

5.2.1: Προσομοίωση του Panda στο Gazebo

Σκοπός της παρούσας εργασίας είναι η προσομοίωση της συμπεριφοράς του ρομποτικού βραχίονα Panda, όπως και των υπολοίπων διατάξεων, σε συγκεκριμένα σενάρια λειτουργίας στο περιβάλλον προσομοίωσης του Gazebo. Δυστυχώς, στην περίπτωση του Panda και της Frank Emika, δεν παρέχονται επίσημα αρχεία URDF από την εταιρεία, τα οποία να περιλαμβάνουν όλα τα απαραίτητα στοιχεία και λειτουργικά και κατασκευαστικά χαρακτηριστικά ώστε να επιτευχθεί η αναπαράσταση του βραχίονα στο Gazebo επιτυχώς.



Σχήμα 5.02: Ο βραχίονας Panda στο γραφικό περιβάλλον του Gazebo

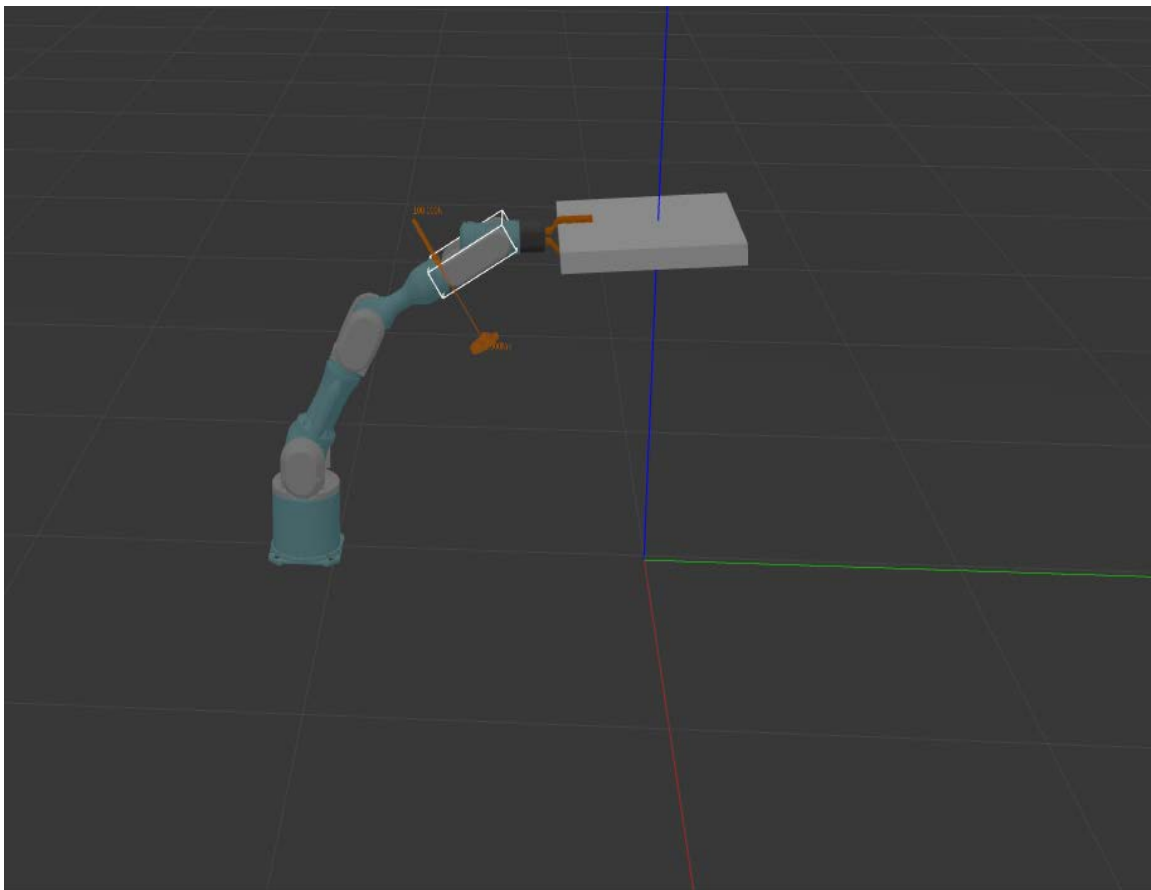
Ως εκ τούτου για τη συνέχεια της εργασίας και βασιζόμενοι στη δυναμική αναγνώριση της ρομποτικής διάταξης από τους [17], έγιναν οι απαραίτητες τροποποιήσεις στα αρχεία περιγραφής της:

- Πάκτωση του βραχίονα και σύνδεση του με το παγκόσμιο σύστημα συντεταγμένων
- Εισαγωγή αποσβέσεων και τριβών στις αρθρώσεις της κατασκευής
- Εισαγωγή μάζας για κάθε σύνδεσμο της διάταξης
- Εισαγωγή μητρώων αδράνειας και την τοποθεσία του κέντρου μάζας για κάθε σύνδεσμο της διάταξης
- Εισαγωγή του plugin `gazebo_ros_control` (για την σύνδεση του ROS Control με το Gazebo) καθώς και ορισμός για κάθε επενεργητή των `hardware_interface::EffortJointInterface`

Κεφάλαιο 6: Προσομοίωση Και Αποτελέσματα

6.1: Λειτουργία Του Ελεγκτή Εμπέδησης

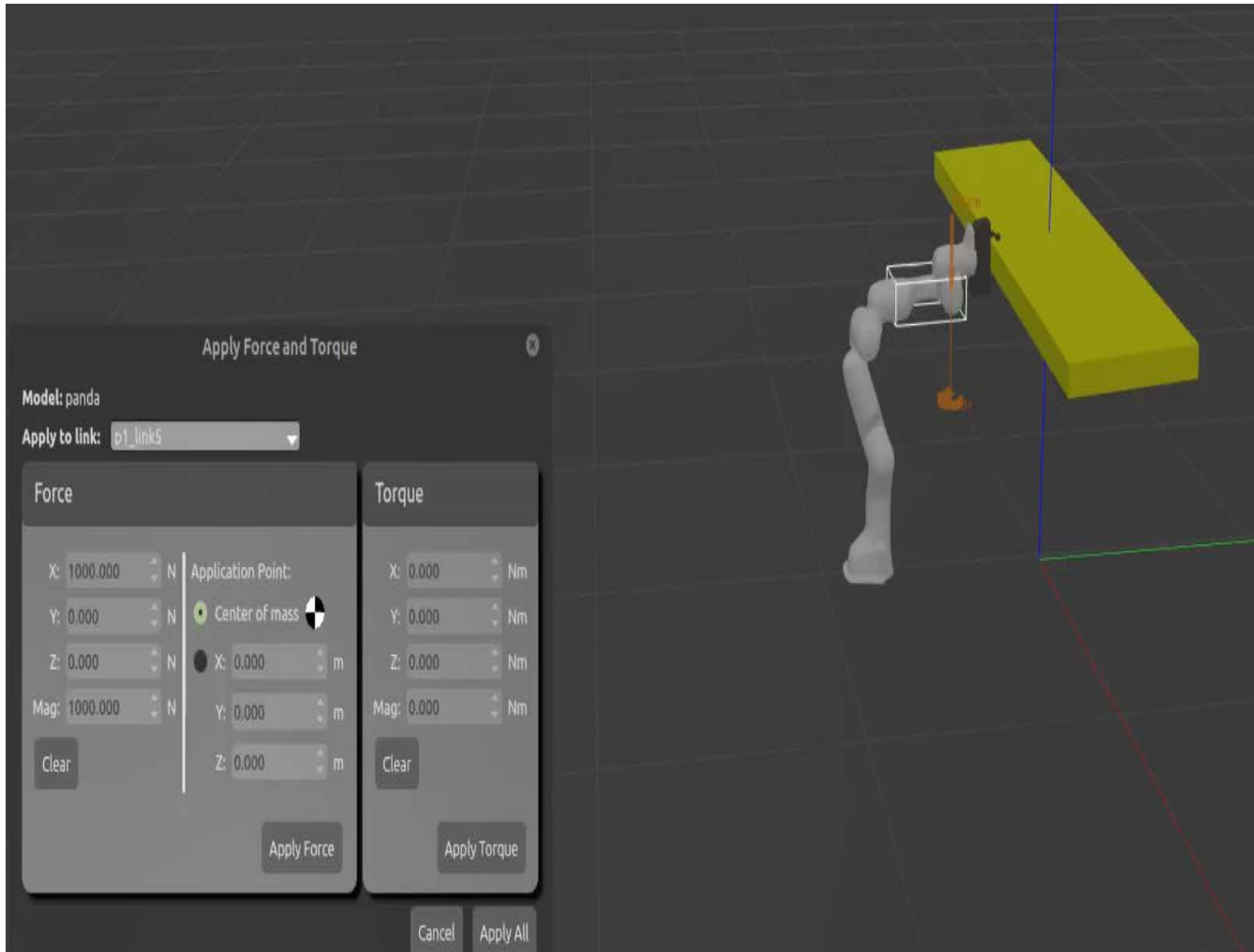
Σε αυτή την ενότητα θα ελεγχθεί η λειτουργία του ελεγκτή εμπέδησης στο περιβάλλον προσομοίωσης του Gazebo. Για το λόγο αυτό θα χρησιμοποιηθούν οι βραχίονες Panda και PA10, όπως έχουν περιγραφεί στο κεφάλαιο 5.



Σχήμα 6.01: Προσομοίωση του PA10 να χειρίζεται άκαμπτο αντικείμενο

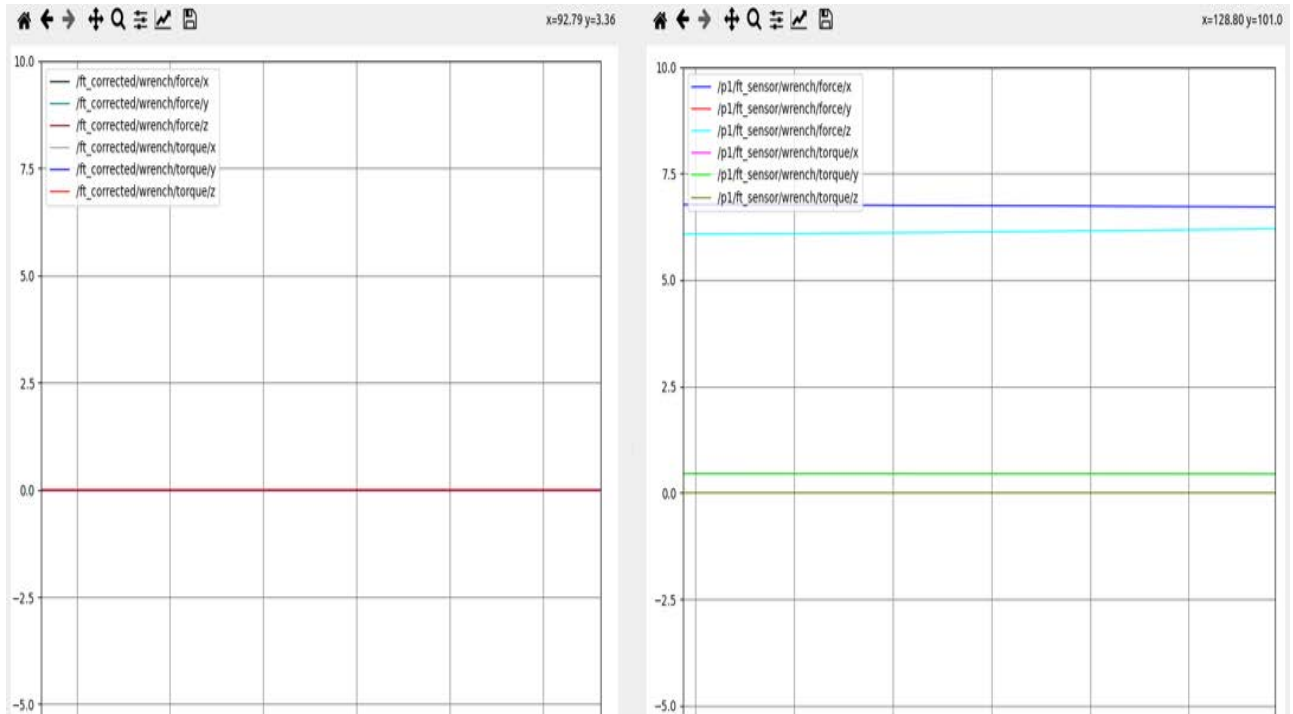
Το σενάριο λειτουργίας είναι ότι φορτώνεται ο βραχίονας στο περιβάλλον του Gazebo. Στο τερματικό εργαλείο δράσης είναι τοποθετημένο ένα αντικείμενο, τη στιγμή που η σύνδεση μεταξύ του αντικειμένου και της αρπάγης του ρομπότ είναι

άκαμπτη. Η συμπεριφορά του βραχίονα εξαρτάται αποκλειστικά από τον ελεγκτή εμπέδησης. Αυτό σημαίνει ότι θα πρέπει να προσαρμόζεται στις εξωγενείς δυνάμεις του περιβάλλοντος και να συμμορφώνεται σε αυτές.

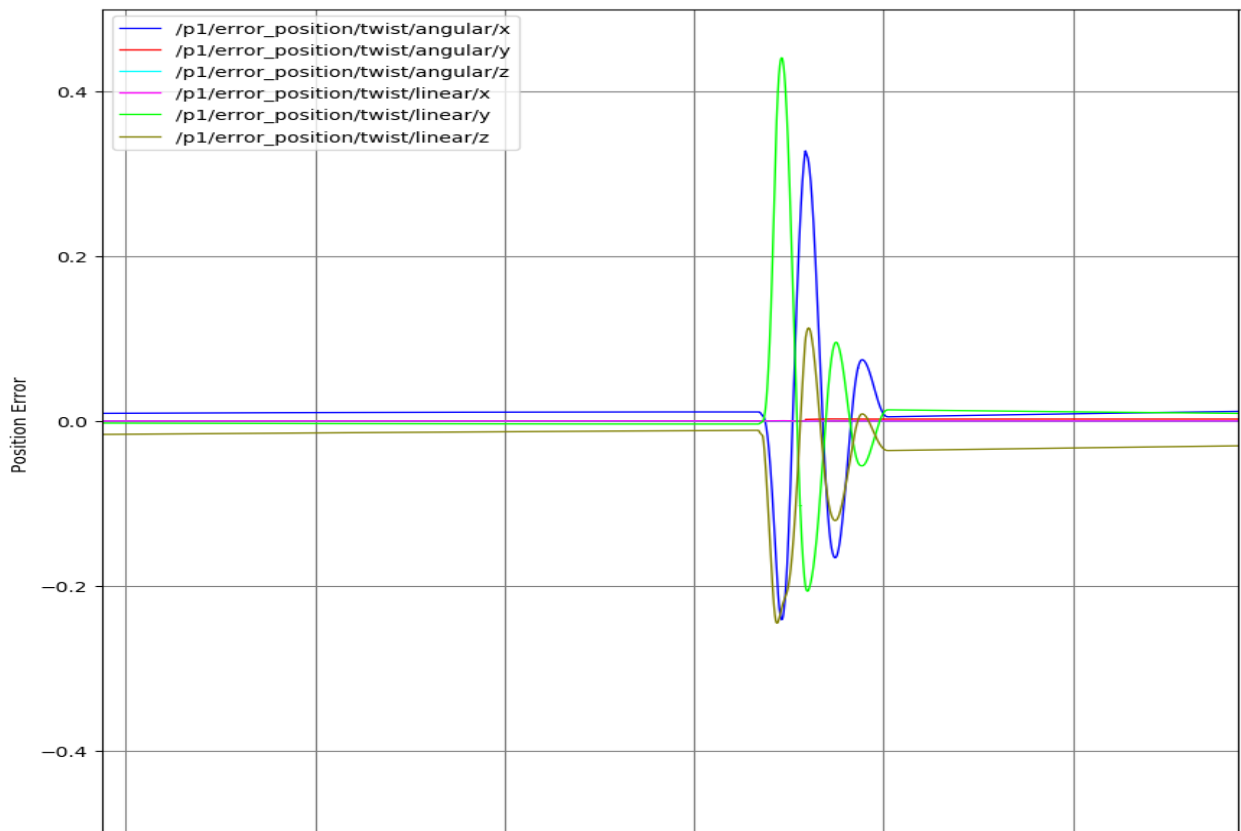


Σχήμα 6.02: Προσομοίωση του Panda να χειρίζεται άκαμπτο αντικείμενο

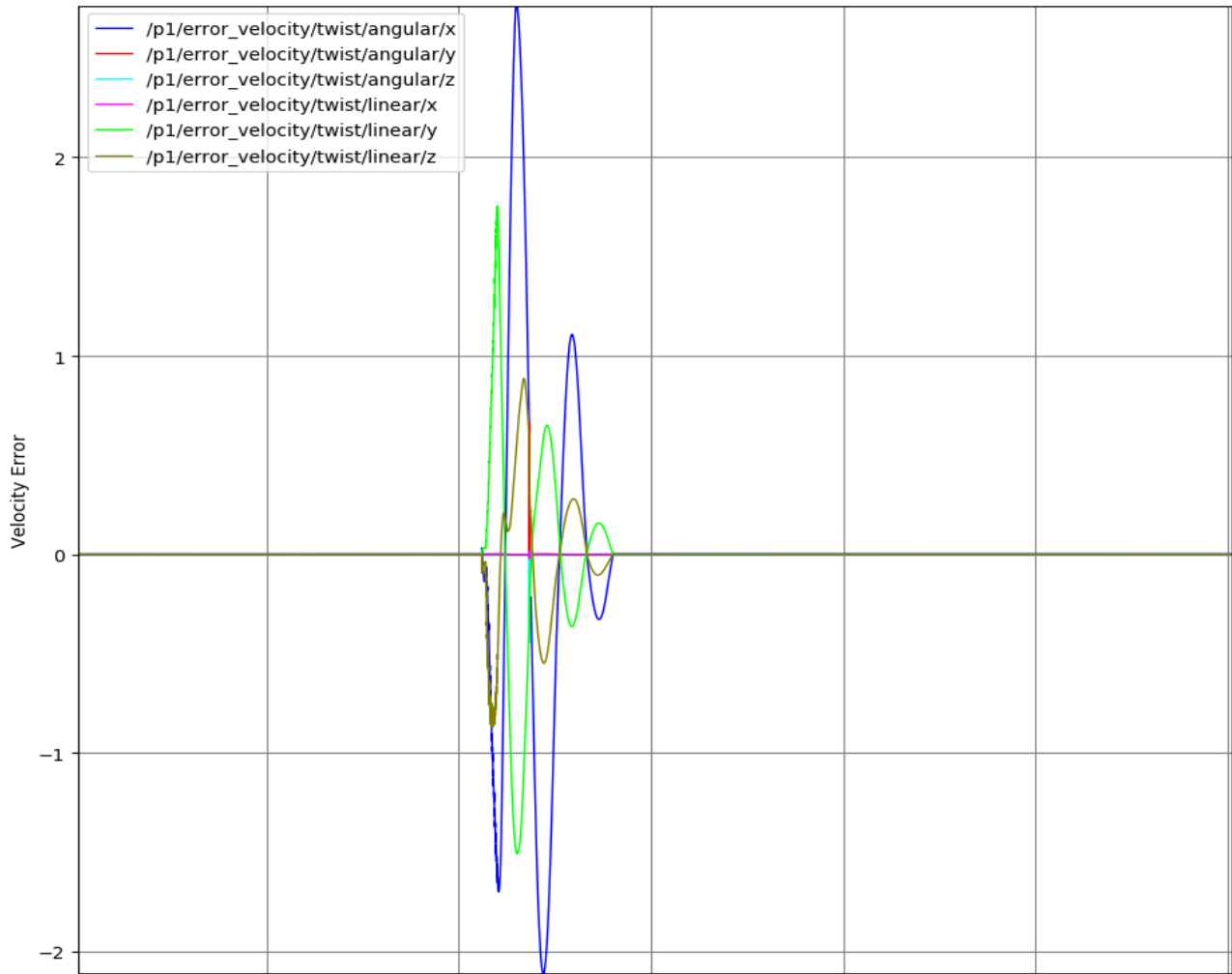
Αρχικά, ο βραχίονας ισορροπεί σε σταθερή θέση σε απόλυτη συμφωνία με τον τρόπο που υλοποιήθηκε ο ελεγκτής εμπέδησης, όπως αυτός περιεγράφηκε στο κεφάλαιο 4. Για να διαπιστωθεί η σωστή λειτουργία του ελεγκτή, θα ασκήσουμε τυχαίου μέτρου και κατεύθυνσης εξωτερικές δυνάμεις και ροπές στο κυρίως σώμα του βραχίονα. Αυτό επιτυγχάνεται πολύ εύκολα χρησιμοποιώντας την κατάλληλη διεπαφή από το γραφικό περιβάλλον χρήσης του Gazebo. Η προσδοκώμενη συμπεριφορά είναι αρχικά ο βραχίονας να εκτραπεί από τη θέση ισορροπίας του και για όσο διάστημα ασκούνται οι εξωτερικές δυνάμεις. Στη συνέχεια, μόλις οι ενεργούμενες εξωτερικές δυνάμεις τερματιστούν, ο βραχίονας θα πρέπει να επιστρέψει στην πρότερη θέση ισορροπίας σε εύλογο χρονικό διάστημα.



Σχήμα 6.03: Αντιστάθμιση των δυνάμεων και ροπών του τερματικού εργαλείου δράσης που οφείλονται στα αδρανειακά χαρακτηριστικά



Σχήμα 6.04: Σφάλμα θέσης του τερματικού εργαλείου δράσης

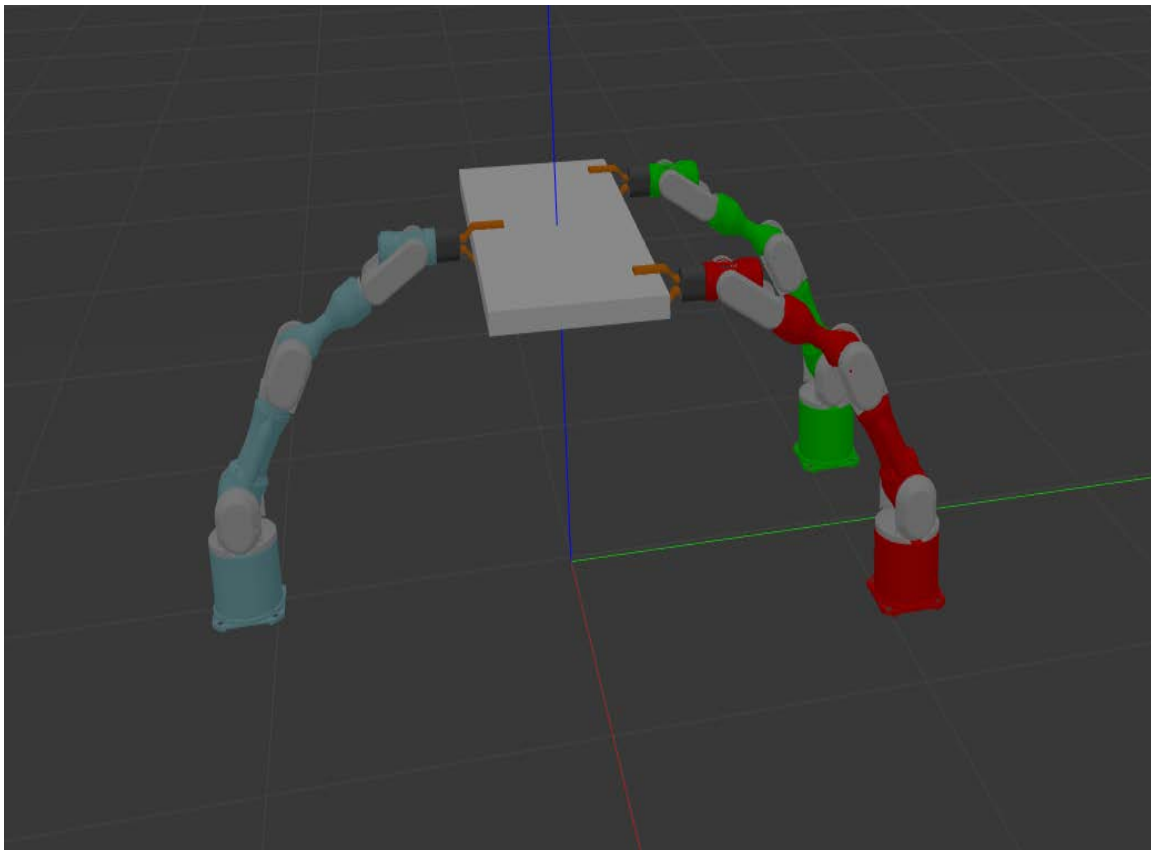


Σχήμα 6.05: Σφάλμα ταχύτητας του τερματικού εργαλείου δράσης

Η αναμενόμενη συμπεριφορά επιβεβαιώνεται πλήρως. Κατά τη διάρκεια εφαρμογής των δυνάμεων και ροπών ο βραχίονας εκτρέπεται από τη θέση ισορροπίας και το σφάλμα θέσης και ταχύτητας του τελικού εργαλείου δράσης μεγαθύνεται. Στη συνέχεια, μόλις οι δυνάμεις και ροπές πάψουν, ο βραχίονας επιστρέφει στην επιθυμητή θέση με μηδενική ταχύτητα.

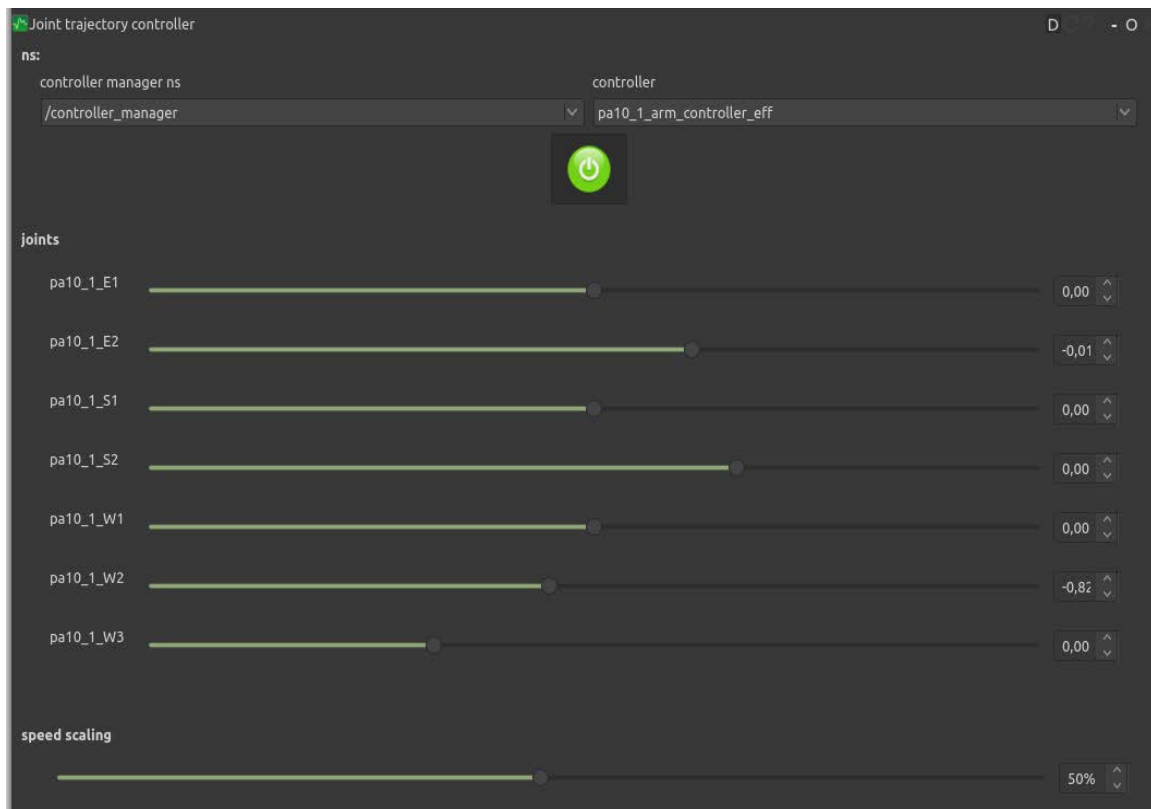
6.2: Προσομοίωση Συστήματος 3 Συνεργατικών PA10

Σε αυτή την ενότητα θα παρουσιαστεί η προσομοίωση ενός ολοκληρωμένου συστήματος, το οποίο αποτελείται από 3 βραχίονες PA10, με στόχο τη μεταφορά ογκώδους αντικειμένου σε ένα εσωτερικό περιβάλλον εργασίας. Πιο αναλυτικά, το σύστημα θα αποτελείται από ένα ρομπότ - οδηγό και δύο ρομπότ - ακολούθους, οι οποίοι θα πρέπει να προσαρμόζονται πλήρως στις κινήσεις του οδηγού. Το ρομπότ - οδηγός προορίζεται να δρα και να κινείται αυτόνομα εντός του χώρου εργασίας μέσω ενός συστήματος ελέγχου τύπου PID σε μια προσπάθεια να προσομοιαστεί η ανθρώπινη παρουσία στο χώρο. Από την άλλη, οι δύο ακόλουθοι θα πρέπει να συμμορφώνονται στις κινήσεις του οδηγού και να ακολουθούν την τροχιά του, μέσω του κατάλληλου βρόχου ελέγχου εμπέδησης που υλοποιήθηκε. Και οι 3 ρομποτικοί πράκτορες μεταφέρουν το φορτίο - αντικείμενο, τη στιγμή που τα τελικά εργαλεία δράσης τους είναι άκαμπτα προσαρτημένα σε αυτό. Κατά αυτόν τον τρόπο, το σύστημα πλέον αποτελεί μία κλειστή κινηματική αλυσίδα με άμεση συνέπεια να υπεισέρχονται επιπλέον κινηματικοί περιορισμοί σε αυτό.



Σχήμα 6.06: Προσομοίωση του συστήματος των PA10 στο Gazebo

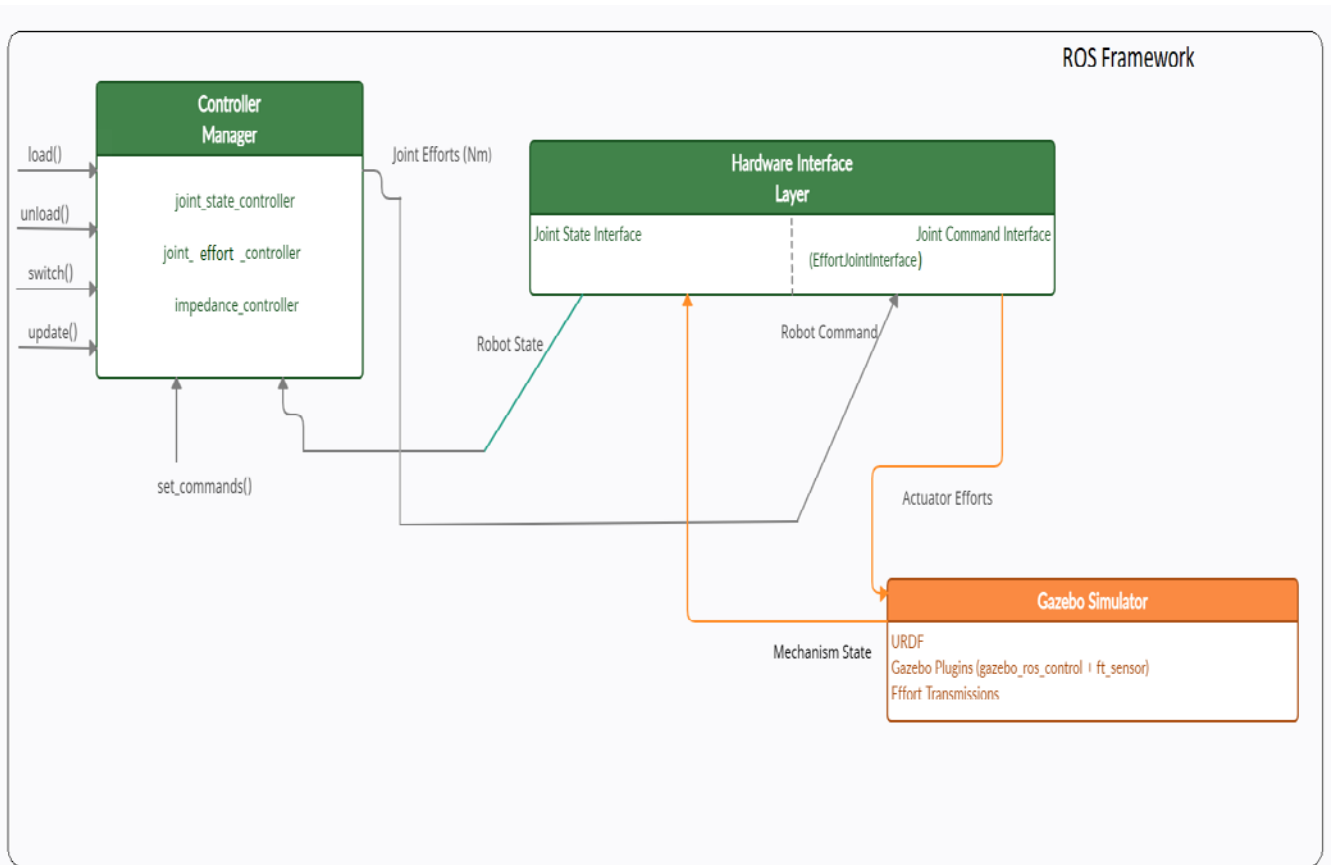
Για λόγους προσομοίωσης, στο ρόλο του ανθρώπου θα τοποθετηθεί ένα ρομπότ - οδηγός, το οποίο, όμως, θα έχει αυτόνομη συμπεριφορά εξωγενώς ορισμένη μέσω κατάλληλου σχήματος ελέγχου. Απεναντίας, τα εναπομείναντα 2 ρομπότ θα πρέπει να συμπεριφέρονται ως ακόλουθοι μέσω του συστήματος ελέγχου εμπέδησης, όπως αυτό έχει περιγραφεί. Κατά αυτόν τον τρόπο επιτυγχάνονται οι προδιαγραφές του προβλήματος, όπου δηλαδή το ρομπότ - ηγέτης είναι επιφορτισμένο με τον σχεδιασμό της τροχιάς κίνησης του αντικείμενου, ενώ στόχος των ακολούθων είναι να συμμορφώνονται σε αυτήν την τροχιά ενεργώντας πλήρως μεμονωμένα. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι τα ρομπότ - ακόλουθοι επιτυγχάνουν τέτοια συμπεριφορά μέσω του βρόχου ελέγχου χωρίς τη χρήση για παράδειγμα οπτικών αισθητήρων. Για την ακρίβεια, δεν γνωρίζουν καν την ύπαρξη των άλλων διατάξεων.



Σχήμα 6.07: Διεπαφή χρήστη για την αυτόνομη κίνηση του βραχίονα οδηγού

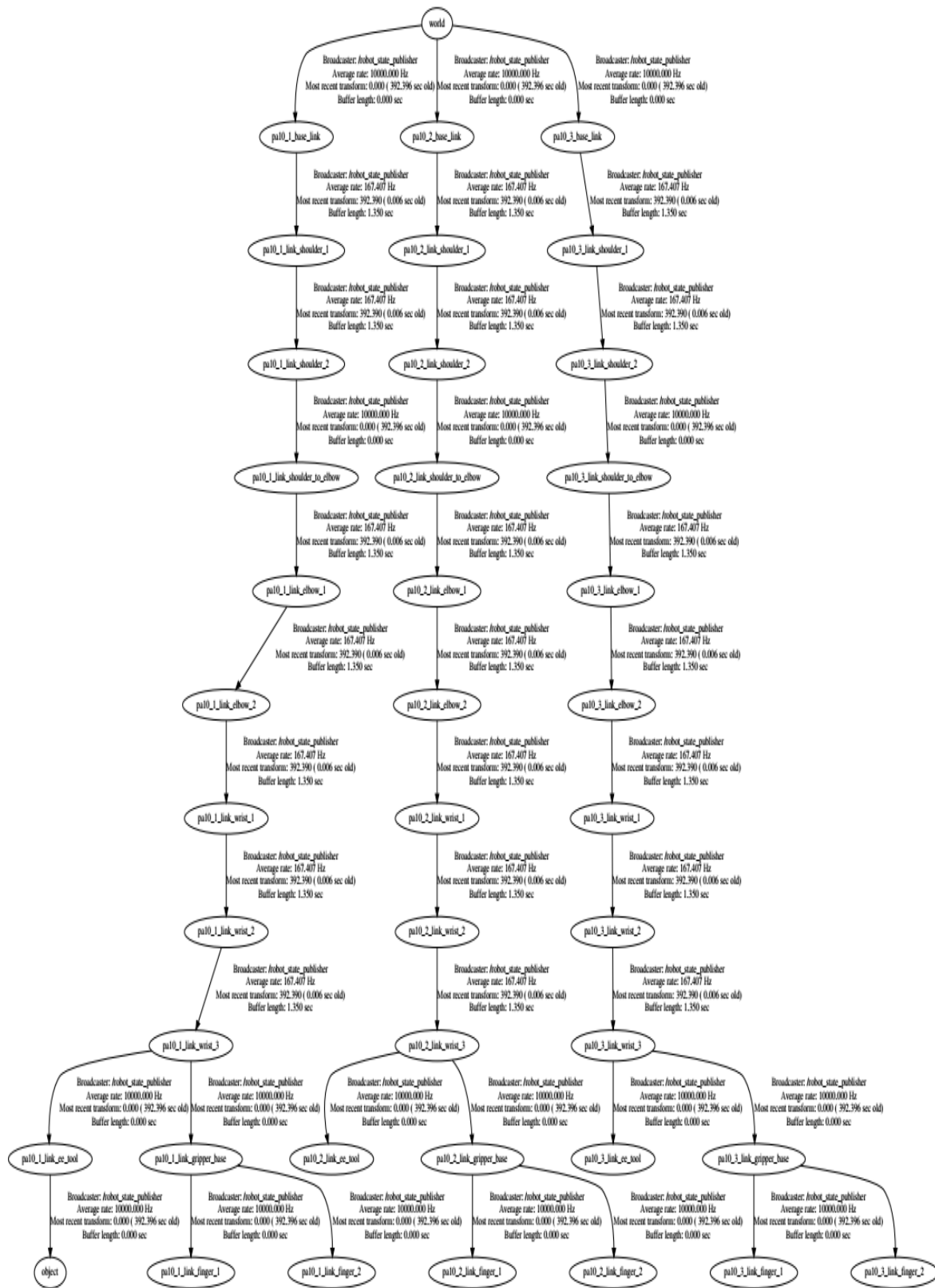
Όσον αφορά το σχήμα ελέγχου του ρομπότ - οδηγού, χρησιμοποιήθηκε ένας έτοιμος `joint_trajectory_controller` του πακέτου `ros_controllers`. Πιο συγκεκριμένα, πρόκειται για μια ομάδα ελεγκτών τύπου PID, καθένας εκ των οποίων αντιστοιχεί σε μία άρθρωση της ρομποτικής διάταξης. Όπως και στην περίπτωση των ρομπότ - ακολούθων, χρησιμοποιείται το `hardware_interface::effort_joint_interface`. Η

αυτόνομη κίνηση επιτυγχάνεται μέσω ειδικής διεπαφής χρήστη που παρέχεται από το πακέτο `rgt_gui`.



Σχήμα 6.08: Αρχιτεκτονική της προσομοίωσης στο πλαίσιο του ROS Control

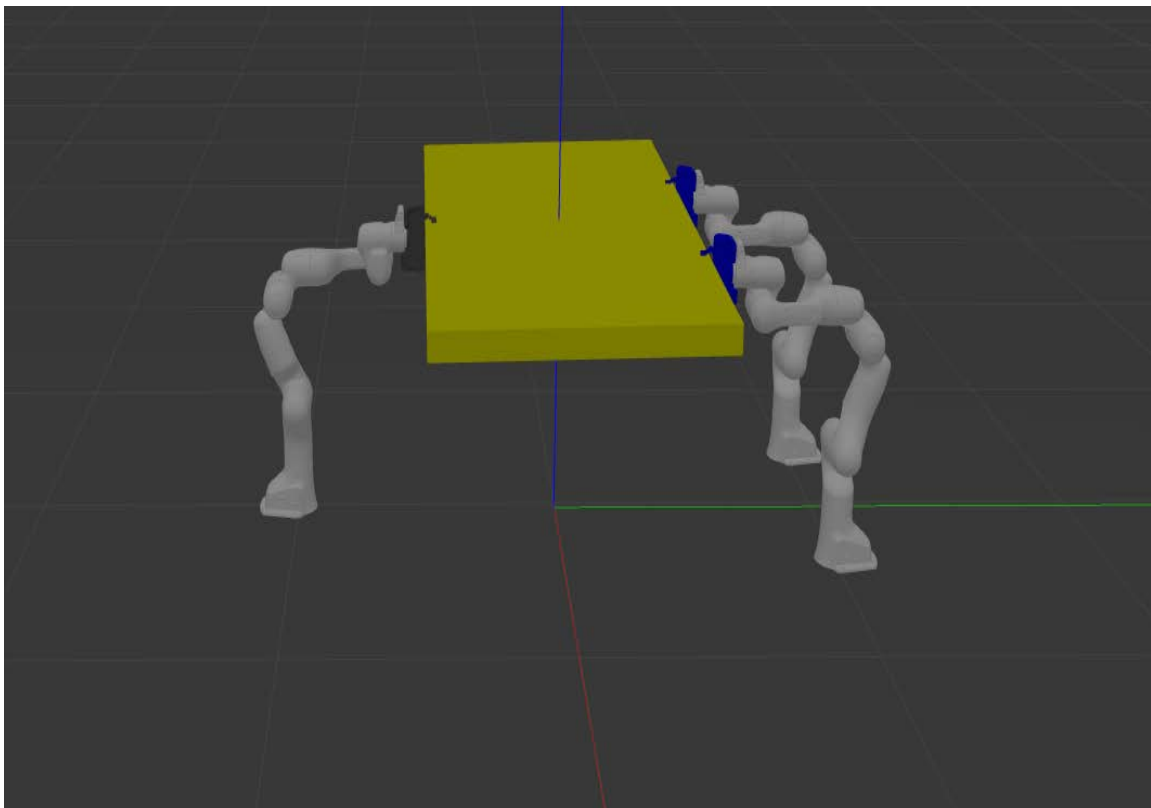
Η όλη προσομοίωση, όπως έχει αναφερθεί, ορίζεται στο πρότυπο Unified Robot Description Format (URDF), το οποίο αποτελεί το επίσημο πρότυπο που υποστηρίζει το ROS. Σε γενικές γραμμές, το μοντέλο μιας ρομποτικής διάταξης ορίζεται ως μια ανοιχτή κινηματική αλυσίδα, δηλαδή σε δομή δέντρου (kinematic tree). Ωστόσο, για τη συγκεκριμένη περίπτωση και λόγω της άκαμπτης σύνδεσης των ρομπότ με το αντικείμενο δημιουργείται μια κλειστή κινηματική αλυσίδα. Το URDF πρότυπο δεν υποστηρίζει τέτοιες κινηματικές δομές. Για αυτό το λόγο είναι αναγκαίο, μέσα στο URDF αρχείο, να παρεμβληθούν άκαμπτοι σύνδεσμοι σε πρότυπο SDF μεταξύ των τερματικών εργαλείων δράσης των ρομπότ με το αντικείμενο, το οποίο πρότυπο υποστηρίζει αποτελεσματικά κλειστές κινηματικές αλυσίδες. Σε αυτή την περίπτωση, αυτή η τεχνική λειτουργεί αποτελεσματικά και επιτυγχάνονται τα ζητούμενα.



Σχήμα 6.09: Κινηματική δομή ολόκληρης της διάταξης των PA10

6.3: Προσομοίωση Συστήματος 3 Συνεργατικών Panda

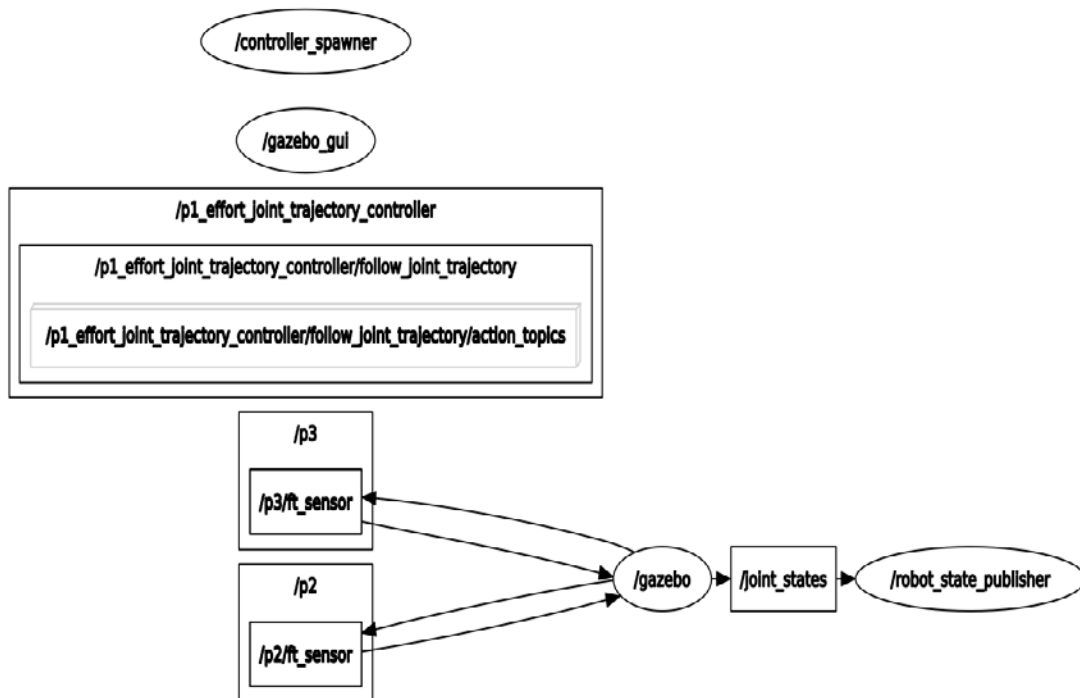
Σε αυτή την ενότητα θα παρουσιαστεί η προσομοίωση ενός ολοκληρωμένου συστήματος, το οποίο αποτελείται από 3 βραχίονες Panda, με στόχο τη μεταφορά ογκώδους αντικειμένου σε ένα εσωτερικό περιβάλλον εργασίας. Η δομή του συστήματος παραμένει αυτούσια με το σύστημα των PA10, δηλαδή το σύστημα αποτελείται από ένα ρομπότ - οδηγό και δύο ρομπότ - ακολούθους, οι οποίοι θα πρέπει να προσαρμόζονται πλήρως στις κινήσεις του οδηγού. Οι τεχνικές ελέγχου που εφαρμόζονται είναι ακριβώς οι ίδιες, με το ρομπότ - οδηγό να ελέγχεται μέσω του `joint_trajectory_controller` του πακέτου `ros_controllers`. Φυσικά, και εδώ επιλέγεται το `hardware_interface::EffortJointInterface`. Από την άλλη, οι δύο ακόλουθοι θα πρέπει να συμμορφώνονται στις κινήσεις του οδηγού και να ακολουθούν την τροχιά του, μέσω του γνωστού βρόχου ελέγχου εμπέδησης που υλοποιήθηκε.



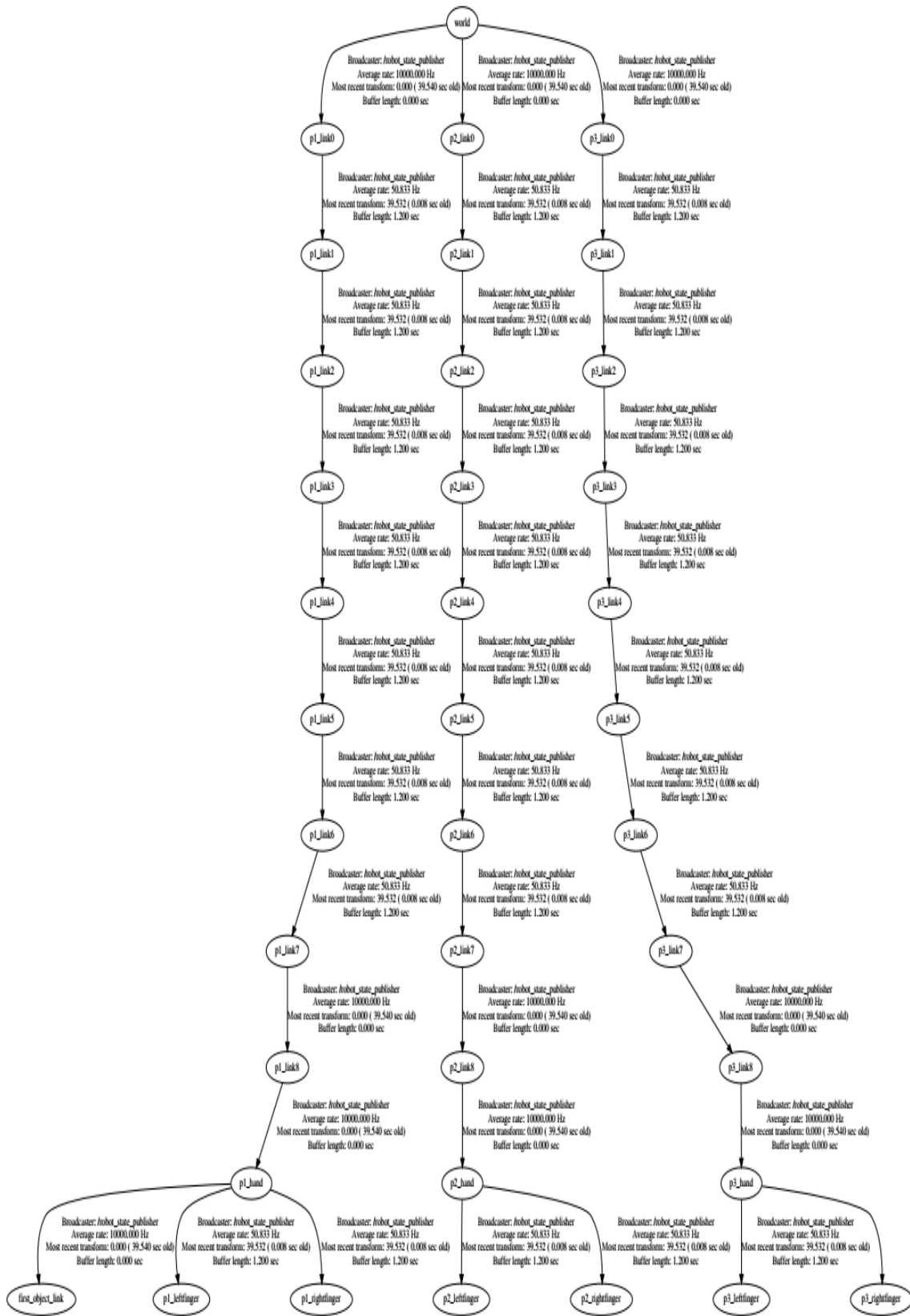
Σχήμα 6.10: Προσομοίωση του συστήματος των Panda στο Gazebo

Η όλη προσομοίωση, όπως έχει αναφερθεί, ορίζεται στο πρότυπο Unified Robot Description Format (URDF), το οποίο αποτελεί το επίσημο πρότυπο που

υποστηρίζει το ROS. Σε γενικές γραμμές, το μοντέλο μιας ρομποτικής διάταξης ορίζεται ως μια ανοιχτή κινηματική αλυσίδα, δηλαδή σε δομή δέντρου (kinematic tree). Και οι 3 ρομποτικοί πράκτορες μεταφέρουν το φορτίο - αντικείμενο, τη στιγμή που τα τελικά εργαλεία δράσης τους είναι άκαμπτα προσαρτημένα σε αυτό. Κατά αυτόν τον τρόπο, το σύστημα πλέον αποτελεί μία κλειστή κινηματική αλυσίδα με άμεση συνέπεια να υπεισέρχονται επιπλέον κινηματικοί περιορισμοί σε αυτό και η πολυπλοκότητα να αυξάνεται. Όπως έχει αναφερθεί, το URDF πρότυπο δεν υποστηρίζει τέτοιες κινηματικές δομές. Μία λύση που μπορεί να δοθεί σε αυτό το ζήτημα είναι με την παρεμβολή άκαμπτων συνδέσμων σε πρότυπο SDF μεταξύ των τερματικών εργαλείων δράσης των ρομπότ με το αντικείμενο, το οποίο πρότυπο υποστηρίζει αποτελεσματικά κλειστές κινηματικές αλυσίδες. Ωστόσο, αυτή η λύση δεν δίνεται χωρίς κόστος, αφού λόγω του ανεπίσημου χαρακτήρα της εγείρονται ζητήματα συμβατότητας του προτύπου με το οικοσύστημα του ROS. Επίσης, ενδέχεται να προκύψουν ζητήματα ευστάθειας με τη physics engine του Gazebo, λόγω της μεγάλης πολυπλοκότητας του μοντέλου και των επιπλέον κινηματικών περιορισμών που υπεισέρχονται. Όπως είναι λογικό, όσο η πολυπλοκότητα αυξάνεται τόσο πιθανότερο είναι τέτοια προβλήματα να κάνουν την εμφάνισή τους.



Σχήμα 6.11: Οι κόμβοι του εν λειτουργία συστήματος σε μορφή γράφου



Σχήμα 6.12: Κινηματική δομή ολόκληρης της διάταξης των Panda

Κεφάλαιο 7: Γενικά Συμπεράσματα

Σε αυτό το κεφάλαιο πραγματοποιείται μία σύντομη ανασκόπηση της παρούσας διπλωματικής εργασίας. Εξάγονται συμπεράσματα όσον αφορά τα αποτελέσματα των προσομοιώσεων και προτείνονται μελλοντικές επεκτάσεις.

Κατά την εκπόνηση της εργασίας, αρχικά υλοποιήθηκε ένας ελεγκτής εμπέδησης και ελέγχθηκε η λειτουργία του στο περιβάλλον του Gazebo. Στη συνέχεια, σχεδιάστηκε ένα συνεργατικό σύστημα ρομπότ, το οποίο θα περιλαμβάνει 3 πράκτορες. Πιο συγκεκριμένα, έχει τοποθετηθεί ένα ρομπότ - οδηγός, το οποίο έχει αυτόνομη συμπεριφορά εξωγενώς ορισμένη μέσω κατάλληλου σχήματος ελέγχου PID. Η συγκεκριμένη επιλογή έγινε εν είδει υποκατάστασης της ανθρώπινης παρουσίας στο σύστημα. Απεναντίας, τα εναπομείναντα 2 ρομπότ θα πρέπει να συμπεριφέρονται ως ακόλουθοι μέσω του συστήματος ελέγχου εμπέδησης. Κατά αυτόν τον τρόπο επιτυγχάνεται το ρομπότ - οδηγός να είναι επιφορτισμένο με τον σχεδιασμό της τροχιάς κίνησης του αντικείμενου, ενώ στόχος των ακολούθων είναι να συμμορφώνονται σε αυτήν την τροχιά ενεργώντας πλήρως μεμονωμένα, προσφέροντας μια λύση στο πρόβλημα της μεταφοράς ογκωδών αντικειμένων.

Σαν μια πρώτη επέκταση του υπάρχοντος συστήματος θα μπορούσε να είναι η προσάρτηση των βραχιόνων σε τροχήλατες πλατφόρμες, με απώτερο σκοπό τη διεύρυνση του λειτουργικού χώρου της διάταξης. Επιπλέον, θα μπορούσε να αντικατασταθεί το ρομπότ - οδηγός με ένα αυτούσιο μοντέλο ανθρώπου με περαιτέρω ανάλυση της δυναμικής του συμπεριφοράς. Φυσικά, μία ακόμα σημαντική εξέλιξη θα μπορούσε να χαρακτηριστεί η επαλήθευση της εγκυρότητας του συστήματος στο φυσικό κόσμο μέσω διεξαγωγής πειραμάτων.

Βιβλιογραφία

- [1] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Springer
- [2] B. Roth, "Principles of Automation," Future Directions in Manufacturing Technology, *Based on the Unilever Research and Engineering Division Symposium held at Port Sunlight, April 1983, Published by Unilever Research, UK*
- [3] Gerkey, B. P. & Mataric, M. J. (2002). Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination, *Robotics and Automation, 2002. Proceedings ICRA'02. IEEE International Conference on 1: 464–469*
- [4] Parra-González, E. F., Ramírez-Torres, G., & Toscano-Pulido, G. Motion Planning for Cooperative Multi-robot Box-Pushing Problem. *Lecture Notes in Computer Science, 382–391. doi:10.1007/978-3-540-88309-8_39*
- [5] Yamada, S., Saito, J.: Adaptive action selection without explicit communication for multi-robot box-pushing. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 31(3), 398–404 (2001)*
- [6] Munoz Melendez, A., Drogoul, A.: Analyzing multi-robot box-pushing. *Avances en la Ciencia de la Computacin. Memoria de los Talleres del Quinto Encuentro Internacional de Computacin. Universidad de Colima, pp. 530–539 (2004)*
- [7] Wang, Z., Kumar, V.: Object closure and manipulation by multiple cooperating mobile robots. In: *Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2002, vol. 1, pp. 394–399 (2002)*
- [8] Craig, J. J. (2004). *Introduction to Robotics: Mechanics and Control (3rd Edition)*. Prentice Hall
- [9] Li, Y., Chen, X.: Modeling and simulation of a swarm of robots for box-pushing task. In: *12th Mediterranean Conference on Control and Automation, Kusadasi, Aydin, Turkey (2004)*
- [10] Wang, Y., de Silva, W.C.: Multi-robot box-pushing single-agent q-learning vs team q-learning. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 3694–3699 (October 2006)*

- [11] Siciliano, B. and Khatib, O., editors (2008). Springer Handbook of Robotics. *Springer Berlin Heidelberg, Berlin, Heidelberg*.
- [12] Goodrich, M. A., & Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3), 203–275. doi:10.1561/11000000005
- [13] Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., and Hirzinger, G. (2007). The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: An International Journal*, 34(5):376–385
- [14] Hogan, N. (1984). Impedance Control: An Approach to Manipulation. *In American Control Conference, volume 107, pages 304–313. IEEE*
- [15] Featherstone, R. (2008). Rigid Body Dynamics Algorithms. doi:10.1007/978-1-4899-7560-7
- [16] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtké and E. Fernandez Perdomo "ros_control: A generic and simple control framework for ROS", *The Journal of Open Source Software*, 2017.
- [17] Gaz, C. R., Cognetti, M., Oliva, A. A., Robuffo Giordano, P., & De Luca, A. (2019). Dynamic Identification of the Franka Emika Panda Robot with Retrieval of Feasible Parameters Using Penalty-based Optimization. *IEEE Robotics and Automation Letters*, 1–1. doi:10.1109/lra.2019.2931248
- [18] Corke, P. (2011). Robotics, Vision and Control. *Springer Berlin Heidelberg*
- [19] Dorf, R. and Bishop, R. (2011). Modern Control Systems. *Prentice Hall*.
- [20] Murray, R. M., Li, Z., and Sastry, S. S. (1994). A Mathematical Introduction to Robotic Manipulation, volume 29.
- [21] Ott, C. (2008). Cartesian Impedance Control of Redundant and Flexible-Joint Robots, *volume 49 of Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg*.
- [22] Kim, T., Kim, H. S., Kim J., "Position-based Impedance Control for Force Tracking of a Wall-Cleaning Unit," *International Journal of Precision Engineering and Manufacturing, Vol. 17, No. 3, March 2016, pp 323-329*.

- [23] O. Khatib, "A unified approach for motion and force control of robot manipulators: the operational space formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43-53, February 1987.
- [24] L. Biagiotti, H. Liu, G. Hirzinger, and C. Melchiorri, "Cartesian impedance control for dexterous manipulation," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada, vol. 4*, pp. 3270-3275, October 2003.
- [25] Δημέας Φ., (2016). Ανάπτυξη Συστημάτων ελέγχου για συνεργασία ρομπότ - ανθρώπου σε χειρισμούς αντικειμένων. *Phd Thesis*
- [26] Koubaa, A. (Ed.). (2016). Robot Operating System (ROS). Studies in Computational Intelligence. The Complete Reference (Volume 1) doi:10.1007/978-3-319-26054-9
- [27] Koubaa, A. (Ed.). (2017). Robot Operating System (ROS). Studies in Computational Intelligence. The Complete Reference (Volume 2) doi:10.1007/978-3-319-54927-9
- [28] Joseph, L., (2018). Mastering ROS for Robotics Programming. 2nd ed. Birmingham - Mumbai, Packt.
- [29] Martinez, A., Fernandez, E., (2013). Learning ROS for Robotics Programming, 1st ed. Birmingham - Mumbai, Packt.
- [30] Joseph, L., (2017). ROS Robotics Projects. 1st ed. Birmingham - Mumbai, Packt.
- [31] Wyatt, N., (2018). A Systematic Approach to Learning Robot Programming with ROS., *CRC Press*.
- [32] Fetter, Lages. (2016). Implementation of Real-Time Joint Controllers. 10.1007/978-3-319-26054-9_26
- [33] K.J. Åström, B. Wittenmark, (1984) Computer Controlled Systems—Theory and Design, Englewood Cliffs, *PrenticeHall*.
- [34] Bruyninckx, H., Open robot control software: the orocos project, in Proceedings of the 2001 *IEEE International Conference on Robotics and Automation*, (IEEE Press, Seoul, 2001), pp.2523–2528

- [35] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sep 2004, vol. 3 (IEEE Press, Sendai, 2004), pp. 2149–2154
- [36] K. Ogata, *Modern Control Engineering* (1970). Englewood Cliffs. *Prentice-Hall*
- [37] Deitel, P. J., & Deitel, H. (2016). *C++ How to Program (10th Edition)*. Pearson.