



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Διπλωματική Εργασία

**Ανάπτυξη Αυτόνομου Ρομποτικού Οχήματος με
Κινηματικό Μοντέλο 4WS4WD**

Εκπόνηση:
Κούρος Γεώργιος

Επιβλέπων:
Πέτρου Λουκάς
Αναπ. Καθηγητής

Θεσσαλονίκη, Οκτώβριος 2016

Περίληψη

Abstract

Development of an Autonomous Robotic Vehicle with a 4WS4WD Kinematic Model

Kouros Georgios

Electrical and Computer Engineering Department

Aristotle University of Thessaloniki, Greece

October 2016

Ευχαριστίες

Πίνακας Περιεχομένων

1 Εισαγωγή	1
1.1 Περιγραφή του Προβλήματος	1
1.2 Συνεισφορά της Διπλωματικής	1
1.3 Διάρθρωση της Διπλωματικής	1
2 Ρομποτική Πλατφόρμα	2
2.1 Ρομποτική Πλατφόρμα <i>Monstertruck</i>	2
2.1.1 Τηλεκατευθυνόμενο Όχημα <i>GroundPounder</i>	2
2.1.2 Σασί Ρομποτικής Πλατφόρμας	3
2.1.3 Ηλεκτρονικός Εξοπλισμός	4
2.2 Σύστημα Μετάδοσης Κίνησης	16
2.2.1 Σύστημα Τετρακίνησης	16
2.2.2 Σύστημα Τετραδιεύθυνσης	18
2.3 Κινηματική Ανάλυση	21
2.3.1 Κινηματικό Μοντέλο Ackermann	22
2.3.2 Κινηματικό Μοντέλο Τετραδιεύθυνσης	25
2.3.3 Κινηματικό Μοντέλο Ρομποτικής Πλατφόρμας <i>Monstertruck</i>	31
3 Αυτόνομη Πλοήγηση σε Αγνωστο Περιβάλλον	36
3.1 Εντοπισμός Θέσης και Χαρτογράφηση	36
3.1.1 Οδομετρία	36
3.1.2 Εκτίμηση Κατάστασης με Συνδυαστική Αντίληψη	38
3.1.3 Ταυτόχρονη Χαρτογράφηση και Εντοπισμός Θέσης	39
3.2 Αυτόνομη Πλοήγηση	41
3.2.1 Κατασκευή Μονοπατιού	42
3.2.2 Αποφυγή Εμποδίων	48
3.2.3 Διάσχιση Μονοπατιού	57
4 Αρχιτεκτονική Συστήματος	67
4.1 Robotic Operating System	67
4.2 Αρχιτεκτονική Λογισμικού Ρομποτικής Πλατφόρμας <i>Monstertruck</i>	68
4.2.1 Software/Hardware Interface	69
4.2.2 Control	70
4.2.3 SLAM	71
4.2.4 Σύστημα Αυτόνομης Πλοήγησης	74
4.2.5 Robot State	81
4.2.6 Visualization	83
4.3 Εργαλεία Προσομοίωσης	84
4.3.1 STDR	85
4.3.2 Gazebo	85

5 Πειράματα και Αποτελέσματα	88
5.1 Πειράματα Κινηματικού Μοντέλου	88
5.1.1 Τροχιές Αρνητικής Τετραδιεύθυνσης	89
5.1.2 Τροχιές Θετικής Τετραδιεύθυνσης	90
5.2 Πειράματα Χαρτογράφησης	91
5.2.1 Χαρτογράφηση στον 2D Προσομοιωτή STDR	91
5.2.2 Χαρτογράφηση στον 3D προσομοιωτή Gazebo	92
5.2.3 Χαρτογράφηση σε Πραγματικό Περιβάλλον με τη Ρομποτική Πλατφόρμα Mon- stertruck	93
5.3 Πειράματα Αξιολόγησης Συστημάτων Αυτόνομης Πλοήγησης	97
5.3.1 Κατασκευή Μονοπατιών	97
5.3.2 Παραμόρφωσης Μονοπατιού με τον Αλγορίθμου Reeds-Shepp Band	100
5.3.3 Πειράματα Διάσχισης Μονοπατιού	102
5.3.4 Πειράματα Αυτόνομης Πλοήγησης με Δυναμική Παραμόρφωση και Δυναμική ¹ Ανακατασκευή Μονοπατιού	109
5.4 Αξιολόγηση Απόδοσης Υπολογιστή Odroid-XU4	110
6 Συμπεράσματα και Μελλοντική Εργασία	111
6.1 Συμπεράσματα	111
6.2 Μελλοντική Εργασία	113
Βιβλιογραφία	115

Λίστα Σχημάτων

2.1	Το τηλεκατευθυνόμενο όχημα <i>GroundPounder</i> , της Redcat Racing.	2
2.2	Τα κουτιά του σασί της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	3
2.3	3D μοντέλο του συναρμολογημένου σασί της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	3
2.4	Raspberry Pi 2	4
2.5	Odroid-XU4	4
2.6	Hokuyo URG-04LX.	6
2.7	Ενδεικτική σάρωση δωματίου.	6
2.8	Compass OS4000.	6
2.9	Logitech Portable Webcam C905.	7
2.10	Αισθητήρες θέσης και ταχύτητας κινητήρων.	8
2.11	Στάδια επεξεργασίας τάσης μπαταρίας για μέτρηση της σε ADC.	9
2.12	Ο κινητήρας Brushed 540 του τηλεκατευθυνόμενου όχηματος <i>GroundPounder</i> .	9
2.13	Ο ελεγκτής ESC B7003SR του τηλεκατευθυνόμενου όχηματος <i>GroundPounder</i> .	9
2.14	Ο κινητήρας maxon EC-max 283858, με κωδικοποιητή και μειωτήρα στροφών.	10
2.15	Ο έξυπνος ελεγκτής κινητήρα, EPOS 24/1, της maxon motor.	10
2.16	Καλωδίωση Κινητήρα, Ελεγκτή και Υπολογιστή.	10
2.17	Μετατροπέας διεπαφής RS232 σε USB.	11
2.18	Σερβοκινητήρας Hitek HS-7954TH.	11
2.19	Pololu Micro Maestro 6-Channel USB Servo Controller	11
2.20	Ο Σερβοκινητήρας Hitek HS-7954TH πάνω στην ρομποτική πλατφόρμα <i>Monstertruck</i> .	12
2.21	Σερβοκινητήρας Dynamixel AX-12A, της Robotis.	12
2.22	Αντάπτορας USB2Dynamixel, της Robotis.	12
2.23	Διάταξη Pitch-Roll του σταθεροποιητή του σαρωτή λέιζερ.	12
2.24	TP-Link Wi-Fi USB Adapter N900 TL-WDN4200.	13
2.25	Οι διακλαδωτές σειριακής διεπαφής USB (USB Hubs) της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	13
2.26	Διασύνδεση διεπαφών των επιμέρους υποσυστημάτων της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	14
2.27	Μπαταρία Gens ace, LiPo, 22.2V, 4000mAh.	14
2.28	Επιμέρους τμήματα συστήματος τροφοδοσίας.	15
2.29	Σύστημα Τροφοδοσίας της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	15
2.30	Σύστημα Μετάδοσης Κίνησης (Drivetrain) της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	16
2.31	Γρανάζια Πινιόν και Ωθησης.	17
2.32	Το κιβώτιο μετάδοσης κίνησης (Transfer Case) του όχηματος <i>GroundPounder</i> .	17
2.33	Το διαφορικό (Differential) του όχηματος <i>GroundPounder</i> .	18
2.34	Λειτουργία ενδεικτικού μηχανισμού διαφορικού.	18
2.35	Ο Μηχανισμός στρέψης, με άξονα έλξης (Drag Link Steering Mechanism) της ρομποτικής πλατφόρμας <i>Monstertruck</i> .	18
2.36	Μετάδοση κίνησης από τον σερβοκινητήρα και το μπράτσο Pitman στον σύνδεσμο έλξης.	19
2.37	Τραπεζοειδής Μηχανισμός στρέψης των τροχών.	20
2.38	Κινηματικό Μοντέλο Ackermann.	23
2.39	Ισοδύναμο μοντέλο ποδηλάτου Ackermann.	24

2.40	Κινηματικό μοντέλο αρνητικής τετραδιεύθυνσης	26
2.41	Κινηματικό μοντέλο θετικής τετραδιεύθυνσης	27
2.42	Ισοδύναμο μοντέλο ποδηλάτου αρνητικής τετραδιεύθυνσης	28
2.43	Ισοδύναμο μοντέλο ποδηλάτου θετικής τετραδιεύθυνσης	29
2.44	Ισοδύναμο κινηματικό μοντέλο ποδηλάτου τετραδιεύθυνσης στο επίπεδο XY	31
2.45	Η σχέση στρέψης μεταξύ εσωτερικού και εξωτερικού τροχού για τα κινηματικά μοντέλα Ackermann, Τετραδιεύθυνσης και της ρομποτικής πλατφόρμας Monstertruck	31
2.46	Το κινηματικό μοντέλο της ρομποτικής πλατφόρμας Monstertruck, σε διάταξη αρνητικής τετραδιεύθυνσης	32
2.47	Το μη ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης, με πλευρική ολίσθηση τροχών, της ρομποτικής πλατφόρμας Monstertruck, σε διάταξη αρνητικής τετραδιεύθυνσης	33
2.48	Ολίσθηση τροχών, λόγω μη ιδανικού μηχανισμού στρέψης	34
3.1	Μετρικός 2D χάρτης και αντίστοιχος τοπολογικός χάρτης σε μορφή γράφου.	39
3.2	Χάρτης πλέγματος κατάληψης κελιού.	40
3.3	Διάγραμμα σταδίων του αλγορίθμου CRSM-SLAM.	40
3.4	Αναπαράσταση ρομπότ ως σημείο και διαστολή εμποδίων, βάσει της ακτίνας του ρομπότ.	43
3.5	Κατευθυνόμενος (αριστερά) και μη κατευθυνόμενος (δεξιά) γράφος.	43
3.6	Δέντρο: συνδεδεμένος, καυτευθυνόμενος και άκυκλος γράφος.	44
3.7	Μέθοδοι αναζήτησης κατά βάθος (αριστερά) και κατά πλάτος (δεξιά) σε δέντρο.	44
3.8	Πλέγμα και γράφοι συνδετικότητας τεσσάρων και οκτώ σημείων (από αριστερά προς τα δεξιά).	45
3.9	Σύγκριση αλγορίθμων αναζήτησης μονοπατιού σε γράφο πλέγματος Dijkstra, Greedy και A*.	46
3.10	Αναζήτηση σε γράφο πλέγματος με A* (αριστερά) και ARA* (δεξιά) για μειούμενο παράγοντα διαστολής ε.	47
3.11	Αναζήτηση σε γράφο πλέγματος με AD* για μειούμενο παράγοντα διαστολής ε.	48
3.12	Πολικό Ιστόγραμμα Πυκνότητας Εμποδίων βάσει του εμποδίων που βρίσκονται μέσα στο ενεργό παράθυρο.	49
3.13	Ενδεικτικός χώρος ταχυτήτων και δυναμικό παράθυρο του αλγορίθμου DWA.	49
3.14	(α) Καθορισμός ελεύθερων τροχών, με βάση τον αλγόριθμο CVM και (β) Χωρισμός του χώρου σε λωρίδες για μετάβαση στην πιο ελεύθερη λωρίδα, με βάση τον αλγόριθμο LCM.	50
3.16	Παράδειγμα μονοπατιού Reeds-Shepp $R_\alpha^+ L_\beta^- R_\gamma^+$	54
3.17	Μη ολονομική φούσκα.	55
3.18	Βέλτιστο ασφαλές μονοπάτι μεταξύ των κέντρων δυο μη ολονομικών φουσκών.	55
3.19	(α') Δημιουργία ζώνης φουσκών, (β') παραμόρφωση και (γ') αντίδραση σε δυναμικό εμπόδιο [23].	56
3.20	Το διάγραμμα σταδίων παραμόρφωσης ολικού μονοπατιού μέσω αλγορίθμου ελαστικής ζώνης και μονοπατιών Reeds-Shepp.	58
3.21	Συναρτήσεις συμμετοχής ασαφούς συνόλου, που χρησιμοποιήθηκαν στον ασαφή ελεγκτή διάσχισης μονοπατιού.	60
3.22	Γενική Αρχιτεκτονική Ασαφούς Ελεγκτή.	61
3.23	Σφάλματα μεταξύ τρέχουσας κατάστασης ρομπότ και τρέχοντα υπό-στόχου που επιλέχθηκαν για την υλοποίηση του αλγορίθμου διάσχισης μονοπατιού.	62
3.24	Βρόχος ελέγχου λειτουργίας διάσχισης μονοπατιού.	62
3.25	Ο ελεγκτής διάσχισης μονοπατιού.	63
3.26	Οι συναρτήσεις συμμετοχής των μεταβλητών εισόδου του ελεγκτή διάσχισης μονοπατιού.	64
3.27	Οι συναρτήσεις συμμετοχής των μεταβλητών εξόδου του ελεγκτή διάσχισης μονοπατιού.	64

4.1	Το διάγραμμα τμημάτων του λογισμικού της ρομποτικής πλατφόρμας Monstertruck.	69
4.2	Το διάγραμμα των κόμβων του τμήματος Software/Hardware Interface της ρομποτικής πλατφόρμας Monstertruck.	70
4.3	Το διάγραμμα των κόμβων του τμήματος Control της ρομποτικής πλατφόρμας Monstertruck.	71
4.4	Διάγραμμα κόμβων του τμήματος χαρτογράφησης και εκτίμησης κατάστασης, με τον αλγόριθμο CRSM-SLAM.	72
4.5	Πλαίσια αναφοράς, βάσει του συστήματος χαρτογράφησης και εκτίμησης κατάστασης με τον αλγόριθμο CRSM-SLAM.	73
4.6	Διάγραμμα κόμβων του τμήματος χαρτογράφησης και εκτίμησης κατάστασης, με τον αλγόριθμο GMapping και με EKF.	73
4.7	Πλαίσια αναφοράς, βάσει του συστήματος χαρτογράφησης και εκτίμησης κατάστασης με τον αλγόριθμο Gmapping και με EKF.	74
4.8	Το τμήμα αυτόνομης πλοήγησης της ρομποτικής πλατφόρμας Monstertruck.	75
4.9	Ο κόμβος move_base με τις βασικές απαιτούμενες διεπαφές [44].	75
4.10	Τα κόστη του επιπέδου διαστολής του 2D χάρτη κόστους [46].	77
4.11	Ο χάρτης πλέγματος κατάληψης κελιού που παράχθηκε από τον αλγόριθμο CRSM-SLAM, για το περιβάλλον robocup του προσομοιωτή STDR και οι αντίστοιχοι χάρτες κόστους.	77
4.12	Παράδειγμα εκτέλεσης ρουτίνας επαναφοράς του αλγορίθμου car_maneuver_recovery.	78
4.13	Ο κόμβος move_base με τα plugins του συστήματος αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού.	79
4.14	Το διάγραμμα δραστηριοτήτων του plugin rsband_local_planner.	80
4.15	Ο κόμβος move_base με τα plugins του συστήματος αυτόνομης πλοήγησης με δυναμική ανακατασκευή μονοπατιού.	80
4.16	Οι βασικές κινήσεις που ορίστηκαν για χρήση στον sbpl_lattice_planner, για την κατασκευή μονοπατιών της ρομποτικής πλατφόρμας Monstertruck.	81
4.17	Το τμήμα Robot State της ρομποτικής πλατφόρμας Monstertruck.	81
4.18	Η ρομποτική πλατφόρμα Monstertruck σε (α') φωτογραφία, (β') 3D μοντέλο και (γ') URDF.	82
4.19	Πλήρης δενδρική δομή των συνδέσμων και αρθρώσεων της ρομποτικής πλατφόρμας Monstertruck.	83
4.20	Μερική δενδρική δομή των σημαντικότερων συνδέσμων και αρθρώσεων της ρομποτικής πλατφόρμας Monstertruck.	83
4.21	Το τμήμα Visualization της ρομποτικής πλατφόρμας Monstertruck.	84
4.22	Περιβάλλον τρισδιάστατης απεικόνισης από τον κόμβο rviz, όπου παρουσιάζεται το μοντέλο του ρομπότ, ο χάρτης, οι χάρτες κόστους, ο στόχος (πράσινο βέλος) το ολικό μονοπάτι(πράσινο), το τοπικό μονοπάτι(μωβ) και η τρέχουσα τροχιά που έχει διασχιστεί(κόκκινο).	84
4.23	Περιβάλλον προσομοίωσης της ρομποτικής πλατφόρμας Monstertruck με τον προσομοιωτή STDR.	86
4.24	Περιβάλλον προσομοίωσης της ρομποτικής πλατφόρμας Monstertruck με τον προσομοιωτή Gazebo.	87
5.1	Το περιβάλλον simple_room που χρησιμοποιήθηκε για την εκτέλεση πειραμάτων αξιολόγησης του κινηματικού μοντέλου στον προσομοιωτή Gazebo.	89
5.2	Τροχιές που παράχθηκαν κατά τα πειράματα αξιολόγησης του κινηματικού μοντέλου σε τροχιές αρνητικής τετραδιεύθυνσης στο περιβάλλον simple_rooms.	90
5.3	Τροχιές που παράχθηκαν κατά τα πειράματα αξιολόγησης του κινηματικού μοντέλου σε τροχιές θετικής τετραδιεύθυνσης στο περιβάλλον simple_rooms.	91
5.4	Χάρτες του περιβάλλοντος robocup.	92

5.5	Τα τμήματα που αφαιρέθηκαν από την αρένα με το ανώμαλο έδαφος, λόγω της αδυναμίας εντοπισμού και προσπελασμότητας από την ρομποτική πλατφόρμα Monstertruck.	93
5.6	Η αρένα του διαγωνισμού RoboCup Rescue 2013 με ομαλό έδαφος και οι αντίστοιχοι παραγόμενοι χάρτες.	95
5.7	Η αρένα του διαγωνισμού RoboCup Rescue 2013 με ομαλό έδαφος και οι αντίστοιχοι παραγόμενοι χάρτες.	96
5.8	Το περιβάλλον sparse_obstacles του προσομοιωτή STDR, που χρησιμοποιήθηκε για τα πειράματα κατασκευής μονοπατιού.	97
5.9	Κατασκευή μονοπατιών με τον αλγόριθμο Dijkstra, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.	98
5.10	Κατασκευή μονοπατιών με τον αλγόριθμο A*, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.	98
5.11	Κατασκευή μονοπατιών με τον αλγόριθμο ARA*, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.	99
5.12	Κατασκευή μονοπατιών με τον αλγόριθμο AD*, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.	99
5.13	Διαδοχική βελτιστοποίηση μονοπατιού με τον αλγόριθμο ARA*.	99
5.14	Διαδοχική βελτιστοποίηση μονοπατιού με τον αλγόριθμο AD*.	99
5.15	Τα περιβάλλοντα του προσομοιωτή STDR που χρησιμοποιήθηκαν για τα πειράματα του αλγορίθμου Reeds-Shepp Band.	100
5.16	Πειράματα παραμόρφωσης μονοπατιού με τον αλγόριθμο Reeds-Shepp Band σε ελεύθερο χώρο στο περιβάλλον simple_rooms.	101
5.17	Πειράματα παραμόρφωσης μονοπατιού με τον αλγόριθμο Reeds-Shepp Band, υπό την επίδραση εμποδίων, στο περιβάλλον sparse_obstacles.	102
5.18	Το περιβάλλον zic_zac_room_no_ramps	103
5.19	Ο χάρτης του περιβάλλοντος zic_zac_room_no_ramps, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).	103
5.20	Διαγράμματα σφαλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον zic_zac_room_no_ramps.	104
5.21	Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον zic_zac_room_no_ramps.	105
5.22	Το περιβάλλον zic_zac_room.	105
5.23	Ο χάρτης του περιβάλλοντος zic_zac_room, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).	105
5.24	Διαγράμματα σφαλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον zic_zac_room.	106
5.25	Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον zic_zac_room.	107
5.26	Το περιβάλλον slope_room.	107
5.27	Ο χάρτης του περιβάλλοντος slope_room, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).	107
5.28	Διαγράμματα σφαλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον slope_room.	108
5.29	Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον slope_room.	109
5.30	Αυτόνομη πλοήγηση με δυναμική παραμόρφωση μονοπατιού, για μετάβαση στις πόζες-στόχους p_i , $i = 1, \dots, 4$.	110
5.31	Αυτόνομη πλοήγηση με δυναμική ανακατασκευή μονοπατιού, για μετάβαση στις πόζες-στόχους p_i , $i = 1, \dots, 4$.	110

Λίστα Πινάκων

2.1	Προδιαγραφές Raspberry Pi 2 και Odroid-XU4.	4
2.2	Προδιαγραφές Hokuyo URG-04LX	6
2.3	Προδιαγραφές Compass OS4000	6
2.4	Προδιαγραφές σερβοκινητήρα Dynamixel AX-12A, της Robotis	12
2.5	Προδιαγραφές TP-Link Wi-Fi USB Adapter N900 TL-WDN4200.	13
2.6	Παράμετροι του μηχανισμού μετάδοσης στρέψης των τροχών.	21
3.1	Τα 48 είδη μονοπατιών Reeds-Shepp.	53
3.2	Οι ασαφείς κανόνες του ελεγκτή ταχύτητας.	65
3.3	Οι ασαφείς κανόνες του ελεγκτή της γωνίας στρέψης των μπροστινών τροχών.	65
3.4	Οι ασαφείς κανόνες του ελεγκτή της απόκλισης της γωνίας στρέψης των πίσω τροχών.	66
4.1	Οι αρθρώσεις του μοντέλου URDF της ρομποτικής πλατφόρμας Monstertruck.	82
5.1	Αποτελέσματα πειράματος κινηματικού σε τροχιές αρνητικής τετραδιεύθυνσης.	90
5.2	Αποτελέσματα πειράματος κινηματικού σε τροχιές θετικής τετραδιεύθυνσης.	90
5.3	Οι πόζες - στόχοι των πειραμάτων κατασκευής μονοπατιών στο περιβάλλον <code>sparse_obstacles</code>	98
5.4	Αποτελέσματα των πειραμάτων κατασκευής μονοπατιών.	100
5.5	Αποτελέσματα των πειραμάτων αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού (DPM) και δυναμική ανακατασκευή μονοπατιού (DPR).	109

Κεφάλαιο 1

Εισαγωγή

- 1.1 Περιγραφή του Προβλήματος
- 1.2 Συνεισφορά της Διπλωματικής
- 1.3 Διάρθρωση της Διπλωματικής

Κεφάλαιο 2

Ρομποτική Πλατφόρμα

2.1 Ρομποτική Πλατφόρμα *Monstertruck*

Η ρομποτική πλατφόρμα *Monstertruck* αποτελεί ένα ρομποτικό όχημα, η ανάπτυξη του οποίου, ξεκίνησε στα πλαίσια της ομάδας P.A.N.D.O.R.A. και ολοκληρώθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Αναπτύχθηκε με σκοπό την χρήση σε εφαρμογές αυτόνομων ρομποτικών οχημάτων, για χαρτογράφηση και εξερεύνηση άγνωστων περιβαλλόντων και αναζήτηση σημείων ενδιαφέροντος, αλλά ταυτόχρονα και για πειραματισμό με εναλλακτικά μοντέλα κίνησης οχημάτων.

2.1.1 Τηλεκατευθυνόμενο Όχημα *GroundPounder*

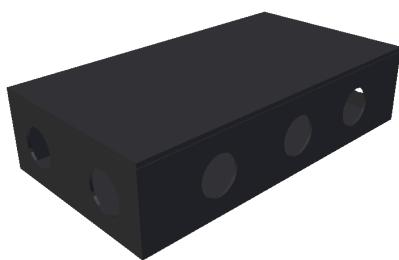
Για την κατασκευή της ρομποτικής πλατφόρμας, χρησιμοποιήθηκε, σαν βάση, το τηλεκατευθυνόμενο όχημα *GroundPounder* της *Redcat Racing*. Ανήκει στην κατηγορία φορτηγών οχημάτων *Monstertruck*, με κλίμακα 1:10 και περιλαμβάνει σκελετό από αλουμίνιο, αναρτήσεις και σύστημα ανεξάρτητης στρέψης των μπροστινών και πίσω τροχών (*Tetradieύθυνση* ή *4-Wheel-Steering*), σε συνδυασμό με μετάδοση της κίνησης και στους τέσσερις τροχούς (*Tetrapakίνηση* ή *4-Wheel-Drive*), προσφέροντας μεγαλύτερη ευελιξία και δυνατότητες, συγκριτικά με τα συμβατικά αυτοκίνητα, που περιλαμβάνουν στρέψη, μόνο, των μπροστινών τροχών (*Μπροστινοδιεύθυνση* ή *Front-Wheel-Steering*) και κίνηση μόνο των μπροστινών (*Μπροστινοκίνηση* ή *Front-Wheel-Drive*) ή μόνο των πίσω τροχών (*Πισωκίνηση* ή *Rear-Wheel-Drive*).



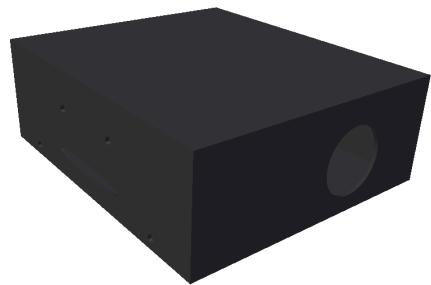
Σχήμα 2.1: Το τηλεκατευθυνόμενο όχημα *GroundPounder*, της Redcat Racing.

2.1.2 Σασί Ρομποτικής Πλατφόρμας

Λόγω, της πληθώρας αισθητήρων, ηλεκτρονικού εξοπλισμού, καλωδιώσεων κλπ. και του περιορισμένου ελεύθερου χώρου πάνω στο όχημα, κρίθηκε σκόπιμο, αυτό, να επεκταθεί, με πρόσθετους χώρους. Για την λύση του προβλήματος, σχεδιάστηκαν, λοιπόν, και κατασκευάστηκαν, από μέλη της ομάδας P.A.N.D.O.R.A. 2014-15, δύο κουτιά, τα οποία προστέθηκαν επάνω στο υπάρχον όχημα, με σκοπό, να περιλάβουν τα επιμέρους υποσυστήματα του ρομπότ.



(α') Κουτί Τροφοδοσίας

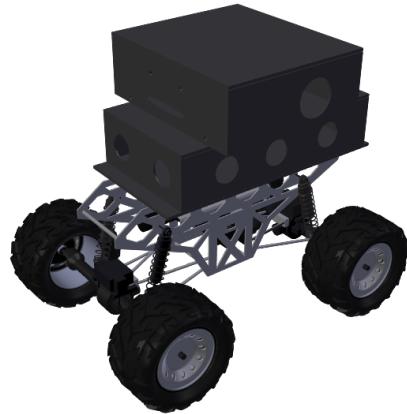


(β') Κουτί Ηλεκτρονικών

Σχήμα 2.2: Τα κουτιά του σασί της ρομποτικής πλατφόρμας Monstertruck.

Το κουτί τροφοδοσίας, του σασί της ρομποτικής πλατφόρμας, που παρουσιάζεται στο σχήμα 2.2α', έχει διαστάσεις $310mm \times 170mm \times 74mm$, και περιλαμβάνει τρύπες, τοποθετημένες περιμετρικά του κουτιού, για πέρασμα καλωδιώσεων. Προορίζεται, όπως λέει και το όνομα του, για την τοποθέτηση του συστήματος τροφοδοσίας της ρομποτικής πλατφόρμας.

Το κουτί ηλεκτρονικών, του σασί της ρομποτικής πλατφόρμας, που παρουσιάζεται στο σχήμα 2.2β', έχει διαστάσεις $210mm \times 240mm \times 84mm$, με δύο τρύπες στα πλάγια του ρομπότ, για τοποθέτηση ανεμιστήρων ψύξης του υπολογιστή, όπως, επίσης και ένα σύνολο από τρύπες στην μπροστινή πλευρά του κουτιού για κεραίες ασύρματης επικοινωνίας WiFi και καλωδιώσεις. Το κουτί ηλεκτρονικών, προορίζεται για την τοποθέτηση του υπολογιστή, των αισθητήρων και των ελεγκτών της ρομποτικής πλατφόρμας.



Σχήμα 2.3: 3D μοντέλο του συναρμολογημένου σασί της ρομποτικής πλατφόρμας Monstertruck.

2.1.3 Ηλεκτρονικός Εξοπλισμός

Υπολογιστής

Το πιο σημαντικό τμήμα ενός ρομποτικού συστήματος και ιδιαίτερα μίας αυτόνομης ρομποτικής πλατφόρμας αποτελεί ο εγκέφαλος του, δηλαδή, το υπολογιστικό του σύστημα, που του επιτρέπει να ελέγχει τα υποσυστήματα του και να εκτελεί διεργασίες και αλγορίθμους. Η επιλογή του υπολογιστικού συστήματος, που εν τέλει, εγκαταστάθηκε στην ρομποτική πλατφόρμα *Monstertruck*, βασίστηκε σε δύο κριτήρια. Πρώτο και βασικότερο κριτήριο επιλογής, αποτέλεσε η υπολογιστική ισχύς και κατά πόσο θα μπορούσε να εκτελεί τους απαιτούμενους αλγορίθμους ταυτόχρονα, αποδοτικά και χωρίς καθυστερήσεις. Το δεύτερο κριτήριο επιλογής, που λήφθηκε υπόψιν, ήταν, η κατανάλωση ισχύος, όσον αφορά τον χρόνο αυτονομίας.

Με βάση τα παραπάνω κριτήρια, τα υπολογιστικά συστήματα που εξετάστηκαν είναι το *Raspberry Pi 2*, του *Raspberry Pi Foundation* και το *Odroid-XU4*, της *Hardkernel*. Και οι δύο υπολογιστές, αυτοί, αποτελούν πλήρεις υπολογιστές, με Κεντρική Μονάδα Επεξεργασίας (CPU), μνήμη RAM, κάρτα γραφικών κλπ., σε εξαιρετικά μικρό μέγεθος και χαμηλή κατανάλωση ισχύος.



Σχήμα 2.4: Raspberry Pi 2



Σχήμα 2.5: Odroid-XU4

Πίνακας 2.1: Προδιαγραφές Raspberry Pi 2 και Odroid-XU4.

Προδιαγραφές	Raspberry Pi 2	Odroid-XU4
CPU	Broadcom BCM2836 Arm7 Quad Core 900MHz Processor	Samsung Exynos5422 ARM® Cortex™-A15 Quad 2.0GHz/ Cortex™-A7 Quad 1.4GHz
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Open GL ES 2.0	Mali™-T628 MP6 OpenGL ES 3.0 / 2.0 / 1.1 and OpenCL 1.1 Full profile
Μνήμη RAM	1GB LPDDR2	2GB LPDDR3
Θύρες USB 2.0	4	1
Θύρες USB 3.0	-	2
Εικόνα	HDMI	HDMI
Ήχος	4 pole Stereo output	HDMI Digital audio output
Αποθηκευτικός Χώρος	Micro SD	Micro SD ή eMMC 5.0
Ethernet	10/100	10/100/1000
Wifi	USB IEEE 802.11b/g/n	USB IEEE 802.11b/g/n 1T1R WLAN
Περιφερειακά - Διεπαφές	40-pin GPIO, UART, SPI, I2C	UART, 30-pin GPIO/IRQ/SPI/ADC 12-pin GPIO/I2S/I2C
Τροφοδοσία	5V, 2A	5V, 4A
Διαστάσεις	85 × 56 × 17mm	82 × 58 × 22mm

Αρχικά, χρησιμοποιήθηκε, στην ρομποτική πλατφόρμα *Monstertruck*, ο υπολογιστής *Raspberry Pi 2*, αλλά, μετά από πειράματα και δοκιμές, με τους απαιτούμενους αλγορίθμους, για την αυτόνομη λειτουργία του οχήματος, διαπιστώθηκε, ότι, ο υπολογιστής *Raspberry Pi 2*, είναι ανεπαρκής για την συγκεκριμένη εφαρμογή. Σαν αποτέλεσμα, στην ρομποτική πλατφόρμα, τελικά χρησιμοποιήθηκε ο υπολογιστής *Odroid-XU4*, που μετά από αντίστοιχα πειράματα, η απόδοση του κρίθηκε πλήρως ικανοποιητική.

Αισθητήρες

Μία εξαιρετικά σημαντική ιδιότητα, κάθε αυτόνομου ρομποτικού συστήματος, αποτελεί η αντίληψη του περιβάλλοντος του. Συγκεκριμένα, η ρομποτική αντίληψη στηρίζεται σε ένα σύνολο αισθητήρων, που επιτρέπουν στο ρομποτικό σύστημα να λαμβάνει πληροφορίες, σχετικά με το περιβάλλον του, σε μορφή κατανοητή και αξιοποιήσιμη από αυτό.

Οι ρομποτικοί αισθητήρες, χωρίζονται σε κατηγορίες, ανάλογα με την πηγή της πληροφορίας, σε *ιδιοδεκτικούς* (*proprioceptive*) ή *εξωδεκτικούς* (*exteroceptive*) [1], εάν η πληροφορία προέρχεται από το ίδιο το ρομποτικό σύστημα, ή από το περιβάλλον του, αντίστοιχα. Παραδείγματα *ιδιοδεκτικών* αισθητήρων, αποτελούν, οι *αισθητήρες μέτρησης θέσης*, *ταχύτητας* και *ροπής* των κινητήρων, *γυροσκόπια*, *αισθητήρες μέτρησης* της φόρτισης των μπαταριών κα. Αντίστοιχα, *εξωδεκτικοί αισθητήρες*, *θεωρούνται*, οι *αισθητήρες επαφής* (*tactile sensors*), οι *ηλεκτρονικές πυξίδες* (*compass, IMU*), *αισθητήρες GPS*, οι *υπέρυθροι*, *υπερηχητικοί* και *λέιζερ αισθητήρες απόστασης* (*range sensors*), όπως επίσης και οι *κάμερες*. Επίσης, χωρίζονται και με βάση την πηγή εκπομπής της πληροφορίας [1] σε *παθητικούς* (*passive*), εάν μετρούν κάποια μορφή ενέργειας που προέρχεται από το περιβάλλον και σε *ενεργητικούς* (*active*), εάν εκπέμπουν ενέργεια στο περιβάλλον και έπειτα, μετρούν την αντίδραση του περιβάλλοντος. Με βάση, τον συγκεκριμένο ορισμό, *αισθητήρες αφής*, *ηλεκτρονικές πυξίδες* και *κάμερες*, αποτελούν *παθητικούς αισθητήρες*, ενώ *κωδικοποιητές* (*encoders*) κινητήρων, *GPS*, *αισθητήρες απόστασης*, αποτελούν *ενεργητικούς αισθητήρες*.

Ένα αυτόνομο ρομποτικό όχημα, είναι προφανές, ότι απαιτεί αισθητήρες, από όλες τις παραπάνω κατηγορίες για να μπορεί να αντιληφθεί και να κινηθεί μέσα στο περιβάλλον του, αλλά και να αντιδράσει μ' αυτό. Ακολούθως, παρουσιάζεται το σύνολο των αισθητήρων, που περιλαμβάνει η ρομποτική πλατφόρμα *Monstertruck*.

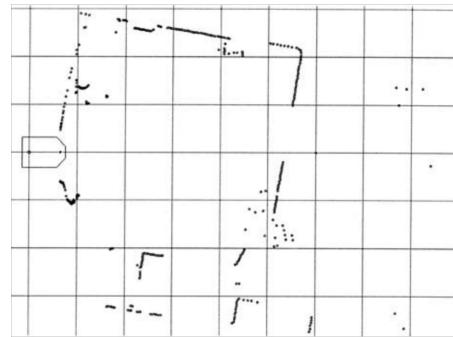
1. Σαρωτής Λέιζερ (*Laser Scanner*):

Οι πιο σημαντικοί αισθητήρες για ένα αυτόνομο ρομποτικό όχημα είναι οι *αισθητήρες απόστασης* (*range sensors*), οι οποίοι, του προσφέρουν πληροφορία, σχετικά με την απόσταση του οχήματος από εμπόδια, επιτρέποντας του, με αυτόν τον τρόπο, μέσω κατάλληλων αλγορίθμων, να χαρτογραφεί τον περιβάλλοντα χώρο του, να ξέρει, ανά πάσα στιγμή, τη θέση του και να πλοηγείται αυτόνομα μέσα σε αυτόν, αποφεύγοντας συγκρούσεις.

Στην ρομποτική πλατφόρμα *Monstertruck*, για τους παραπάνω λόγους, εγκαταστάθηκε ένας *σαρωτής λέιζερ Hokuyo URG-04LX*. Η λειτουργία του βασίζεται στην τεχνική *μέτρησης απόστασης*, μέσω ανίχνευσης φωτός (*Light Detection and Ranging - LIDAR*). Δηλαδή, εκπέμπει έναν παλμό ακτινοβολίας λέιζερ στο περιβάλλον, προς μία κατεύθυνση και καταγράφοντας τον χρόνο που έκανε να επιστρέψει ο οπισθοσκεδαζόμενος παλμός, μπορεί να υπολογίσει την απόσταση του αισθητήρα από το περιβάλλον για εκείνη την κατεύθυνση. Πραγματοποιώντας την μέτρηση αυτή για ένα εύρος γωνιών, ο αισθητήρας προσφέρει μία διδιάστατη αναπαράσταση του περιβάλλοντος.



Σχήμα 2.6: Hokuyo URG-04LX.



Σχήμα 2.7: Ενδεικτική σάρωση δωματίου [2].

Πίνακας 2.2: Προδιαγραφές Hokuyo URG-04LX

Προδιαγραφές	Hokuyo URG-04LX
Τροφοδοσία	5VDC, 500mA
Εμβέλεια	60 - 4 095 mm
Περιοχή Μέτρησης	240°
Ακρίβεια	60 – 1000mm : ±10 1000 – 4095mm : 1%
Γωνιακή Ακρίβεια	0.36°(360°/1024)
Διεπαφή	USB, RS232
Διαστάσεις	50 × 50 × 70mm

2. Πυξίδα (Compass):

Ένα, άλλο είδος αισθητήρων, ιδιαίτερα δημοφιλές και απαραίτητο στις περισσότερες ρομποτικές εφαρμογές, αποτελούν οι αισθητήρες κατεύθυνσης (*heading sensors*) [1]. Στην κατηγορία, αυτή, ανήκουν τα γυροσκόπια (*gyroscopes*), τα κλινόμετρα (*inclinometers*) και οι πυξίδες (*compasses*). Οι αισθητήρες, αυτοί, χρησιμοποιούνται για να καθοριστούν ο προσανατολισμός (*orientation / yaw*) και η κλίση (*pitch, roll*) του ρομποτικού οχήματος, αλλά και σε συνδυασμό με μετρήσεις ταχύτητας για την εκτίμηση της θέσης του (*dead reckoning*).

Η ρομποτική πλατφόρμα *Monstertruck* χρησιμοποιεί την πυξίδα *Compass OS4000*, της *Ocean Server*. Ο αισθητήρας αυτός, συνδυάζει ένα μαγνητόμετρο (*magnetometer*), τριών αξόνων και ένα επιταχυνσιόμετρο (*accelerometer*), τριών αξόνων. Το μαγνητόμετρο χρησιμοποιεί το μαγνητικό πεδίο της γης, για να μετρήσει τον απόλυτο προσανατολισμό, ως προς τους τρεις άξονες x, y, z , ενώ το επιταχυνσιόμετρο, μετράει μεταβολές στην ταχύτητα, ως προς τους τρεις άξονες x, y, z .

Πίνακας 2.3: Προδιαγραφές Compass OS4000



Σχήμα 2.8: Compass OS4000.

Προδιαγραφές	Compass OS4000
Τροφοδοσία	3.3 – 5VDC, 30mA@3.3V
Σειριακή Διεπαφή	TTL 4800-115000 baud
Επικοινωνίας	8 bit, 1 stop, no parity
Συχνότητα	0.01 – 40Hz
Ακρίβεια Αζιμούθιου	< 0.5°, 0.1° resolution
Ακρίβεια Κλίσης	< 0.5°, 0.1° resolution
Διαστάσεις	15 × 15mm

Η πυξίδα *Compass OS4000*, για την ώρα, χρησιμεύει, μόνο, για την, εύρεση της κλίσης (*pitch, roll*) του ρομπότ, έτσι ώστε να σταθεροποιείται στο οριζόντιο επίπεδο ο σαρωτής λέιζερ, που

αναφέρθηκε παραπάνω, μέσω ενός μηχανισμού σταθεροποιητή *pitch-roll*, που αποτελείται από δύο. σερβοκινητήρες. Παρόλα αυτά, η πυξίδα, μπορεί να χρησιμοποιηθεί, κάλλιστα και για την εκτίμηση κατάστασης (θέση και προσανατολισμός) του οχήματος, για την επέκταση αλγορίθμων ακολούθησης μονοπατιού, με βάση την ομαλότητα του εδάφους, βάση της τρέχουσα κλίσης του οχήματος, αλλά και σε ρουτίνες ασφαλείας, σε περίπτωση επικίνδυνων επιπέδων κλίσης του οχήματος, που μπορεί να προκαλέσουν ανατροπή.

3. Κάμερα:

Η όραση αποτελεί την πιο ισχυρή αίσθηση του ανθρώπου. Προσφέρει ένα τεράστιο όγκο πληροφορίας για το περιβάλλον και διευκολύνει την διάδραση του με αυτό. Στα ρομποτικά συστήματα, η αίσθηση της όρασης προσεγγίζεται με κάμερες, οι οποίες καταγράφουν την ίδια πληροφορία, σε μεγάλο βαθμό που συγκεντρώνει και το ανθρώπινο μάτι.

Στα ρομποτικά συστήματα, κάμερες, μπορεί να χρησιμοποιούνται για επίβλεψη και χειρισμό ρομποτικών συστημάτων, αλλά μεγαλύτερο ενδιαφέρον, παρουσιάζει ο κλάδος της ρομποτικής όρασης, που ασχολείται με την δημιουργία αλγορίθμων, που εξάγουν πληροφορία, από τις εικόνες, που παράγει μία κάμερα. Για παράδειγμα, κάμερες και αλγόριθμοι ρομποτικής όρασης, χρησιμοποιούνται για αναγνώριση αντικειμένων, προσώπων και προτύπων, γενικότερα, αλλά, ακόμα και σε χαρτογράφηση περιβάλλοντος και εκτίμηση κατάστασης (*Visual SLAM* κα.

Στην ρομποτική πλατφόρμα *Monstertruck*, είναι εγκατεστημένη, μία απλή *web* κάμερα *Logitech Portable Webcam C905*, η οποία, χρησιμοποιήθηκε, στα πλαίσια της παρούσας εργασίας, μονάχα για επίβλεψη κατά τον χειρισμό, ή την αυτόνομη λειτουργία της ρομποτικής πλατφόρμας. Παρόλα αυτά, όπως αναφέρθηκε παραπάνω, με την εκμετάλλευση της πληροφορίας από την κάμερα, μέσω κατάλληλων αλγορίθμων, η λειτουργικότητα της ρομποτικής πλατφόρμας, μπορεί να επεκταθεί σημαντικά.



Σχήμα 2.9: Logitech Portable Webcam C905.

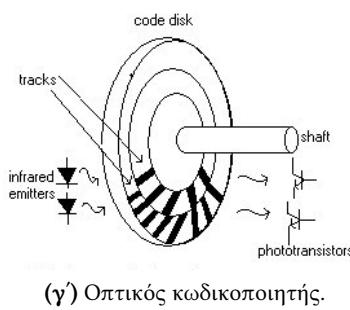
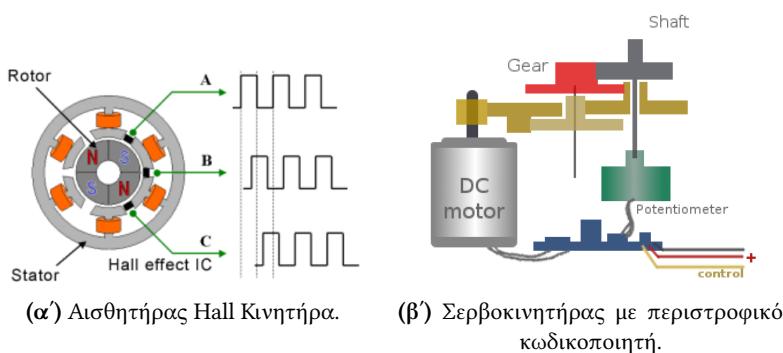
4. Αισθητήρες Θέσης και Ταχύτητας Κινητήρων:

Σε ρομποτικά συστήματα, με κινούμενα μέρη, όπως είναι προφανές, χρησιμοποιούνται κινητήρες και σερβοκινητήρες. Για τον ακριβή έλεγχο και παρακολούθηση, αυτών, είναι απαραίτητη η ύπαρξη αισθητήρων, που προσφέρουν πληροφορία, σχετικά με την θέση, ταχύτητα, επιτάχυνση, φορτίο, ρεύμα, τροφοδοσία ή θερμοκρασία, κατά την λειτουργία τους. Στη ρομποτική πλατφόρμα *Monstertruck*, χρησιμοποιείται, ένας αισθητήρας *Hall*, για τον κινητήρα των τροχών και κωδικοποιητές (*encoders*), για τον κινητήρα και τους σερβοκινητήρες του οχήματος.

Ο αισθητήρας *Hall*, είναι ένας μετατροπέας, που μεταβάλλει την τάση εξόδου του, ως αντίδραση στις μεταβολές ενός μαγνητικού πεδίου. Στους κινητήρες χρησιμοποιείται ως μετρητής των στροφές ανά λεπτό. Είναι οικονομικός αισθητήρας, μπορεί να δουλέψει σε υψηλές συχνότητες

και δεν επηρεάζεται από φαινόμενα θορύβου μηχανικών επαφών (contact bounce), αλλά, έχει μικρή ακρίβεια και είναι επιρρεπείς σε σφάλματα ολίσθησης (drift).

Οι κωδικοποιητές, είναι μία κατηγορία αισθητήρων που χρησιμοποιούνται, για την μέτρηση της θέσης ή ταχύτητας του άξονα ενός κινητήρα. Η μέτρηση αυτή χρησιμοποιείται από το κύκλωμα κλειστού βρόχου, ενός κινητήρα για έλεγχο θέσης ή ταχύτητας. Οι απλοί, συμβατικοί σερβοκινητήρες (*hobby servos*), του εμπορίου, χρησιμοποιούν περιστροφικούς κωδικοποιητές στην μορφή ποτενσιομέτρων (*rotary / shaft encoders*), που μεταβάλλουν την τάση εξόδου τους, ανάλογα με την θέση του άξονα του σερβοκινητήρα. Αντίθετα, οι βιομηχανικοί κινητήρες, συνήθως χρησιμοποιούν οπτικούς κωδικοποιητές (*optical encoders*), οι οποίοι, αποτελούνται από έναν δίσκο με διαφανείς και αδιαφανείς περιοχές και και ζεύγη φωτοεκπομπών και φωτοδεκτών, που διαβάζουν τα μοτίβα του δίσκου και συμπεραίνουν την θέση του άξονα του κινητήρα.

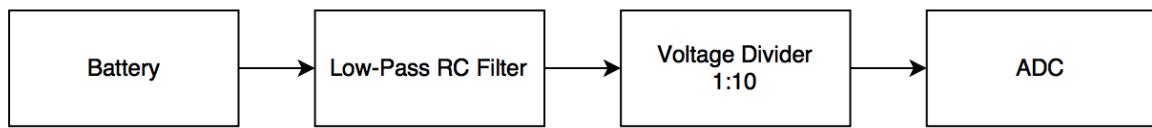


Σχήμα 2.10: Αισθητήρες θέσης και ταχύτητας κινητήρων.

5. Αισθητήρας Μέτρησης Τάσης Μπαταρίας:

Η ρομποτική πλατφόρμα *Monstertruck*, τροφοδοτείται, μέσω, επαναφορτιζόμενων μπαταριών *Λιθίου - Πολυμερών* (*LiPo*), που συνδυάζουν υψηλή χωρητικότητα, μικρό όγκο και βάρος, σε σύγκριση με άλλους τύπους μπαταριών. Ένα σημαντικό πρόβλημα των μπαταριών *LiPo*, είναι η ασφάλεια τους, καθώς σε περίπτωση υπερφόρτισης, αποφόρτισης, βραχυκυκλώματος, κρούσης ή διείσδυσης, μπορεί να προκληθεί καταστροφική ζημιά, όπως ρήξη συσκευασίας, διαρροή ηλεκτρολύτη και φωτιά. Επίσης, κακή χρήση της μπαταρίας, μέσω υπερφορτίσεων και αποφορτίσεων πέρα από τα επιτρεπτά επίπεδα, προκαλεί μείωση της χωρητικότητας και του χρόνου ζωής της μπαταρίας. Καθίσταται, επομένως, απαραίτητη, η χρήση ενός αισθητήρα, που θα μετρά τα επίπεδα τάσης της μπαταρίας και θα τα μεταδίδει στον κεντρικό υπολογιστή του ρομποτικού συστήματος, στον οποίο θα λειτουργεί μία διεργασία, που θα λαμβάνει την πληροφορία αυτή, θα την επεξεργάζεται κατάλληλα (πχ. φιλτράρισμα θορύβου) και θα εξάγει συμπεράσματα και θα ειδοποιεί τον επιβλέπον / χειριστή, σε περίπτωση που η μπαταρία χρειάζεται φόρτιση ή σε περίπτωση που παρεκκλίνει από τα επιτρεπτά όρια.

Για την μέτρηση της τάσης της μπαταρίας, απαιτείται ένας αισθητήρας που θα μετατρέπει την αναλογική τάση σε ψηφιακή πληροφορία. Τον σκοπό αυτό εξυπηρετούν οι *Μετατροπείς Αναλογικού Σήματος σε Ψηφιακό (ADC)*, οι οποίοι μετατρέπουν μία αναλογική τάση σε έναν ψηφιακό αριθμό. Η διαδικασία της μετατροπής, περιλαμβάνει κβαντισμό και περιοδική δειγματοληψία, της τάσης εισόδου και σαν αποτέλεσμα εισάγει ένα μικρό σφάλμα μετατροπής, το οποίο στην προκειμένη περίπτωση, δεν επηρεάζει, σημαντικά, την εφαρμογή. Οι αισθητήρες ADC, περιγράφονται, συνήθως, από μία μέγιστη τάση εισόδου (πχ. 5V). Επειδή, όμως στην ρομποτική πλατφόρμα χρησιμοποιούνται μπαταρίες LiPo, με ονομαστική τάση 22.2V (μέγιστη τάση 25.2V), απαιτείται μία κλιμάκωση της τάσης εισόδου. Για τον λόγο αυτό, η τάση εισόδου του μετατροπέα ADC, κλιμακώνεται, μέσω ενός διαιρέτη τάσης, με σχέση 1:10, από 0-25.2V σε 0-2.52V. Επίσης, λόγω των μεγάλων διαταραχών, στην τάση της μπαταρίας, κατά την λειτουργία της ρομποτικής πλατφόρμας, χρησιμοποιήθηκε ένα χαμηλοπερατό φίλτρο RC.



Σχήμα 2.11: Στάδια επεξεργασίας τάσης μπαταρίας για μέτρηση της σε ADC.

Κινητήρας

Το τηλεκατευθυνόμενο όχημα GroundPounder, αρχικά, περιελάμβανε έναν *Brushed DC* ηλεκτρικό κινητήρα, με μέγιστη ταχύτητα, περίπου, 30 000 rpm, ο οποίος ελεγχόταν από έναν ελεγκτή ESC, με δυνατότητες ελέγχου ταχύτητας και φοράς. Παρόλα αυτά, λόγω της μικρής ακρίβειας, ελέγχου ταχύτητας και την απουσία κωδικοποιητή ή άλλων αισθητήρων για την παροχή μετρήσεων, σχετικά την πραγματική ταχύτητα του κινητήρα, κάθε στιγμή, σε συνδυασμό, με τις υψηλές απαιτήσεις ακρίβειας των ρομποτικών εφαρμογών, κρίθηκε σκόπιμο, το εν λόγω σύστημα κινητήρα και ελεγκτή, να αντικατασταθεί.



Σχήμα 2.12: Ο κινητήρας Brushed 540 του τηλεκατευθυνόμενου οχήματος GroundPounder.

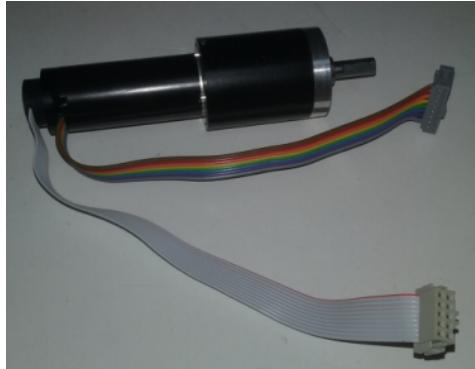


Σχήμα 2.13: Ο ελεγκτής ESC B7003SR του τηλεκατευθυνόμενου οχήματος GroundPounder.

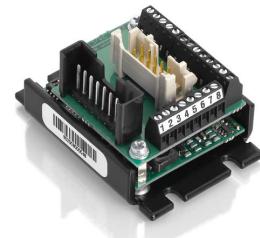
Για την αντικατάσταση, λοιπόν του παραπάνω συστήματος κινητήρα και ελεγκτή, επιλέχθηκαν, από τον διαθέσιμο εξοπλισμό της ομάδας P.A.N.D.O.R.A, ένας κινητήρας, με αντίστοιχο ελεγκτή, της εταιρείας *maxon motor*. Ο κινητήρας *maxon EC-max 283858*, είναι ένας *Brushless EC* κινητήρας, με μέγιστη ταχύτητα 18000 rpm, με ψηφιακό κωδικοποιητή (*encoder*), για μέτρηση της θέσης του όχονα

του κινητήρα, όπως επίσης και αισθητήρα Hall για μέτρηση των στροφών του κινητήρα, ανά λεπτό. Επίσης, περιλαμβάνει ένα κιβώτιο ταχυτήτων - μειωτήρα στροφών(gearbox) v842795-1-5, με σχέση 1:66.

Αντίστοιχα, ως ελεγκτής, επιλέχθηκε ο EPOS 24/1, της maxon motor. Ο ελεγκτής αυτός, αποτελεί ένα μικρού μεγέθους, ψηφιακό, έξυπνο ελεγκτή, με δυνατότητες ελέγχου θέσης, ταχύτητας και ρεύματος, αλλά και δυνατότητες μέτρησης θέσης και ταχύτητας του κινητήρα.

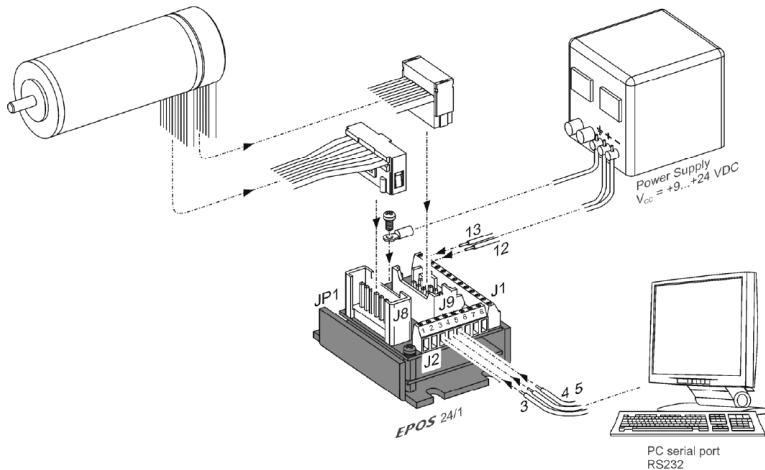


Σχήμα 2.14: Ο κινητήρας maxon EC-max 283858, με κωδικοποιητή και μειωτήρα στροφών.



Σχήμα 2.15: Ο έξυπνος ελεγκτής κινητήρα, EPOS 24/1, της maxon motor [3].

Ο κινητήρας, συνδέεται με τον ελεγκτή EPOS 24/1, μέσω δύο καλωδίων, ένα για τον έλεγχο του κινητήρα και για λήψη μετρήσεων από τον αισθητήρα Hall και το άλλο, για λήψη μετρήσεων από τον ψηφιακό κωδικοποιητή. Ο ελεγκτής EPOS 24/1, επίσης, απαιτεί σύνδεση σε τροφοδοσία 9-24VDC, 1A, ενώ παράλληλα, για επικοινωνία με ηλεκτρονικό υπολογιστή, χρησιμοποιεί το πρωτόκολλο επικοινωνίας RS232, χωρίς χειραψία, μέσω της ελάχιστης συνδεσμολογίας RS232 (RX, TX, Ground).



Σχήμα 2.16: Καλωδίωση Κινητήρα, Ελεγκτή και Υπολογιστή [3].

Ο ελεγκτής EPOS 24/1, όπως προαναφέρθηκε, επιτρέπει την επικοινωνία με τον κεντρικό υπολογιστή της ρομποτικής πλατφόρμας, μέσω του πρωτοκόλλου διεπαφής RS232. Παρόλα αυτά, ο κεντρικός υπολογιστής, δεν διαθέτει διεπαφή RS232 και επομένως, απαιτείται, ένας ενδιάμεσος κόμβος, ο οποίος θα καθιστά δυνατή την επικοινωνία μεταξύ τους. Το ρόλο αυτό, στην προκειμένη περίπτωση, εξυπηρετεί ένα μετατροπέας διεπαφής RS232 σε USB (σχήμα 2.17).



Σχήμα 2.17: Μετατροπέας διεπαφής RS232 σε USB.

Σερβοκινητήρες

Οι σερβοκινητήρες είναι κινητήρες, που επιτρέπουν ακριβή έλεγχο θέσης, ταχύτητας και επιτάχυνσης. Οι συμβατικοί σερβοκινητήρες αποτελούνται από έναν DC κινητήρα, σε συνδυασμό με γρανάζια μετάδοσης, έναν αισθητήρα θέσης, συνήθως περιστροφικό κωδικοποιητή (*rotary encoder*) και ένα κύκλωμα ελέγχου. Οι απλοί, συμβατικοί σερβοκινητήρες ελέγχονται, μέσω σημάτων Διαμόρφωσης Πλάτους Παλμού (PWM), αλλά υπάρχει, βέβαια και μία κατηγορία σερβοκινητήρων, οι λεγόμενοι *έξυπνοι σερβοκινητήρες* (*smart servo motors*), οι οποίοι έχουν ενσωματωμένο μικροελεγκτή. Ο μικροελεγκτής, αυτός, προσφέρει, υψηλότερη ακρίβεια ελέγχου, ευρωστία στο θόρυβο και ομφίδρομη επικοινωνία, μέσω σειριακού πρωτοκόλλου, συνήθως TTL Full-Duplex ή Half-Duplex και RS485. Τέλος, ένα σημαντικό πλεονέκτημα, των έξυπνων σερβοκινητήρων, έναντι των συμβατικών, είναι η παροχή μετρήσεων θέσης (position feedback), εξαιρετικής σημασίας για ρομποτικές εφαρμογές.

Στην ρομποτική πλατφόρμα *Monstertruck* χρησιμοποιούνται συνολικά τέσσερις σερβοκινητήρες, δύο από τους οποίους είναι συμβατικοί και οι άλλοι δύο, έξυπνοι.

Οι δύο συμβατικοί σερβοκινητήρες, χρησιμοποιούνται στο σύστημα τετραδιεύθυνσης της ρομποτικής πλατφόρμας, που θα αναλυθεί στην αντίστοιχη ενότητα, είναι τύπου *Hitek HS-M7990TH*, με ροπή $44kg \cdot cm6V$ και ανάλυση $0.082^\circ/\mu sec$.

Ο έλεγχος των δύο σερβοκινητήρων πραγματοποιείται από έναν ελεγκτή *Micro Maestro 6 - Channel USB Servo Controller*, της *Pololu*. Ο ελεγκτής, αυτός, προσφέρει αποδοτικό έλεγχο σερβοκινητήρων, υψηλής ακρίβειας, με ενσωματωμένο έλεγχο ταχύτητας και επιτάχυνσης, έξι κανάλια ελέγχου και επικοινωνία, μέσω σειριακού πρωτοκόλλου *USB*.



Σχήμα 2.18: Σερβοκινητήρας Hitek HS-7954TH.



Σχήμα 2.19: Pololu Micro Maestro 6-Channel USB Servo Controller

Οι δύο έξυπνοι σερβοκινητήρες, χρησιμοποιούνται ως μηχανισμός σταθεροποίησης (*Pitch-Roll Stabilizer*) του Σαρωτή Λέιζερ, που αναφέρθηκε παραπάνω, λαμβάνοντας υπόψιν πληροφορία για την κλίση του οχήματος, μέσω της πυξίδας *Compass OS4000*. Ο μηχανισμός αυτός είναι απαραίτητος για την αξιόπιστη χαρτογράφηση χώρου με ανώμαλο έδαφος.

Οι έξυπνοι σερβοκινητήρες του μηχανισμού σταθεροποίησης του Σαρωτή Λέιζερ, είναι τύπου *Dynamixel AX-12A*, της *Robotis*. Οι έξυπνοι σερβοκινητήρες *Dynamixel AX-12*, έχουν την δυνατότητα, να



Σχήμα 2.20: Ο Σερβοκινητήρας Hitek HS-7954TH πάνω στην ρομποτική πλατφόρμα Monstertruck.

παίρνουν μετρήσεις, σχετικά με την ταχύτητα, θέση, θερμοκρασία, τάση και φορτίο και να αντιδρούν ανάλογα με την περίπτωση και να μεταδίδουν αυτήν την πληροφορία, στον υπολογιστή του ρομποτικού συστήματος.



Σχήμα 2.21: Σερβοκινητήρας Dynamixel AX-12A, της Robotis.

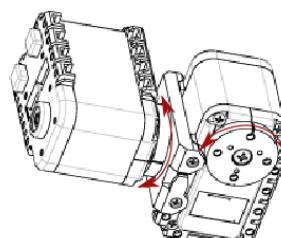
Πίνακας 2.4: Προδιαγραφές σερβοκινητήρα Dynamixel AX-12A, της Robotis

Προδιαγραφές	Dynamixel AX-12A
Τροφοδοσία	9 – 12VDC, 900mA
Σειριακή Διεπαφή Επικοινωνίας	3-pin TTL Half-Duplex 7343bps 1Mbps
Εύρος	300°
Μέγιστη Ροπή	15.3kg · cm
Μέγιστη Ταχύτητα (χωρίς φορτίο)	59 RPM 0.169sec/60°
Feedback	Θέσης, Φορτίου, Θερμοκρασίας, Τάσης
Διαστάσεις	32 × 50 × 40mm

Η επικοινωνία, μεταξύ του υπολογιστή και των έξυπνων σερβοκινητήρων, επιτυγχάνεται μέσω του αντάπτορα *USB2Dynamixel*, ο οποίος επικοινωνεί με τον υπολογιστή, μέσω σειριακού πρωτοκόλλου USB και με τους έξυπνους σερβοκινητήρες, μέσω σειριακής επικοινωνίας TTL. Επίσης, οι δύο έξυπνοι σερβοκινητήρες, συνδέονται, μεταξύ τους, σειριακά, μέσω τοπολογίας *Daisy Chain*.



Σχήμα 2.22: Αντάπτορας USB2Dynamixel, της Robotis.



Σχήμα 2.23: Διάταξη Pitch-Roll του σταθεροποιητή του σαρωτή λέιζερ.

Ασύρματη Επικοινωνία

Η προετοιμασία και ο χειρισμός της ρομποτικής πλατφόρμας *Monstertruck*, ή η επίβλεψη της, κατά την αυτόνομη λειτουργία, από τον χειριστή/επιβλέποντα, απαιτεί έναν πρόσθετο υπολογιστή, ο οποίος θα συνιστά τον σταθμό χειρισμού/επίβλεψης. Η επικοινωνία, μεταξύ των δύο υπολογιστικών συστημάτων, μπορεί να πραγματοποιηθεί, είτε ενσύρματα, μέσω μίας διασύνδεσης διεπαφής *Ethernet*, είτε ασύρματα, μέσω ενός πομποδέκτη ασύρματης επικοινωνίας *Wi-Fi*, σε συνδυασμό με έναν δρομολογητή *Wi-Fi* (*Wi-Fi router*) που αποτελεί και την πιο πρακτική λύση, αν αναλογιστεί κανείς, ότι σε αντίθετη περίπτωση, ο χειριστής, θα έπρεπε να κυνηγάει το ρομπότ από πίσω, με κίνδυνο, πρόκλησης ατυχήματος, πιθανή αποσύνδεση, αλλά και πιθανή παρεμβολή στις μετρήσεις των αισθητήρων. Η απαίτηση, αυτή, ικανοποιείται, στη ρομποτική πλατφόρμα *Monstertruck*, μέσω ενός αντάπτορα *TP-Link WiFi N900 TL-WDN4200*.

Πίνακας 2.5: Προδιαγραφές TP-Link Wi-Fi USB Adapter N900 TL-WDN4200.



Σχήμα 2.24: TP-Link Wi-Fi USB Adapter N900 TL-WDN4200.

Προδιαγραφές	TP-Link Wi-Fi USB Adapter N900 TL-WDN4200
Σύνδεση	USB 2.0
Ταχύτητα	Dual Band $2 \times 450 Mbps$
Πρότυπο	IEEE 802.11b/g/n
Συχνότητα	2.4/5GHz
Ασφάλεια	WEP (64-128bit)

Διασύνδεση Υποσυστημάτων

Στις παραπάνω ενότητες, αναφέρθηκαν τα επιμέρους υποσυστήματα της ρομποτικής πλατφόρμας *Monstertruck* και έγινε φανερό, ότι δεν χρησιμοποιούνται οι ίδιες διεπαφές επικοινωνίας σε κάθε συσκευή, αλλά και ότι κάθε συσκευή, χρησιμοποιεί το δικό της πρωτόκολλο επικοινωνίας. Το μόνο κοινό όλων των υποσυστημάτων, είναι η διασύνδεση και η συγκέντρωση της πληροφορίας, στον κεντρικό κόμβο του συστήματος, τον υπολογιστή *Odroid-XU4*.

Ένα σημαντικό πρόβλημα, του υπολογιστή *Odroid-XU4*, αποτελεί ο ανεπαρκής, για την συγκεκριμένη εφαρμογή, αριθμός θυρών διεπαφής σειριακής επικοινωνίας USB. Επομένως, για την ταυτόχρονη λειτουργία, όλων των επιμέρους αισθητήρων και ελεγκτών του συστήματος, κρίθηκε απαραίτητη η προσθήκη δύο διακλαδωτών USB (*USB Hubs*) (σχήμα 2.25X). Η διασύνδεση των διεπαφών, όλων των επιμέρους επιμέρους υποσυστημάτων, της ρομποτικής πλατφόρμας *Monstertruck*, παρουσιάζεται στο σχήμα 2.26.

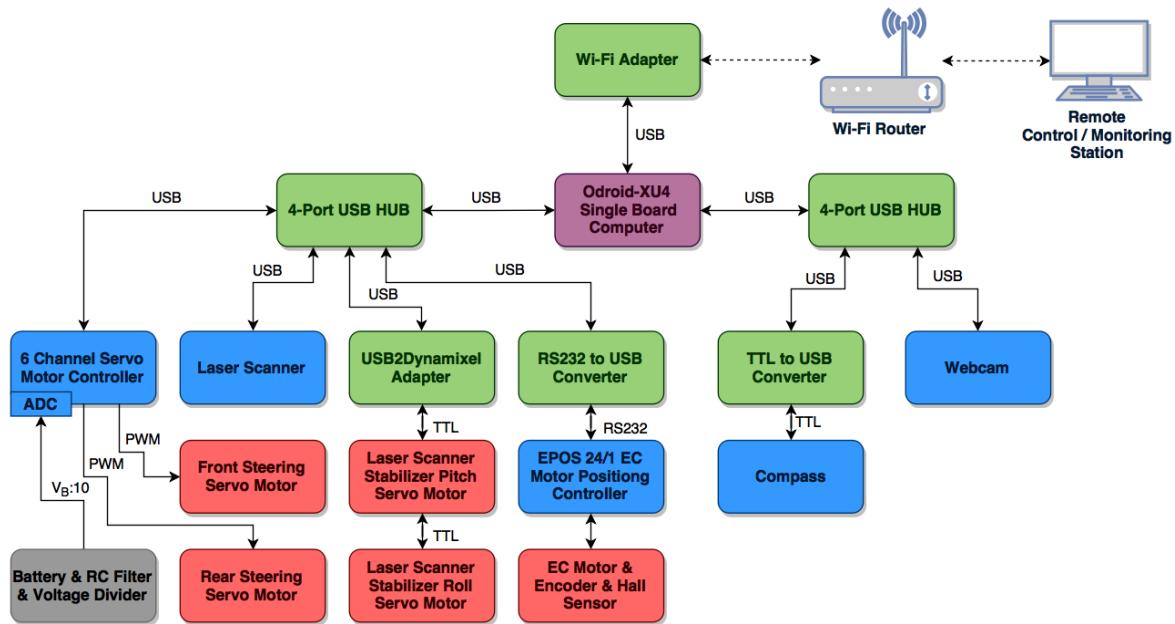


(α') Akasa AK-HB-01-BK 4-port USB hub Black.



(β') Akasa AK-HB-01WH C 4-PORT USB hub White.

Σχήμα 2.25: Οι διακλαδωτές σειριακής διεπαφής USB (*USB Hubs*) της ρομποτικής πλατφόρμας *Monstertruck*.



Σχήμα 2.26: Διασύνδεση διεπαφών των επιμέρους υποσυστημάτων της ρομποτικής πλατφόρμας *Monstertruck*.

Σύστημα Τροφοδοσίας

Από την παρουσίαση των επιμέρους υποσυστημάτων της ρομποτικής πλατφόρμας *Monstertruck*, που πραγματοποιήθηκε στις προηγούμενες παραγράφους, προκύπτει, ότι, κάθε υποσύστημα - συσκευή, περιλαμβάνει διαφορετικές προδιαγραφές τροφοδοσίας. Επομένως, απαιτείται, ένα εκτενές και πλήρες σύστημα τροφοδοσίας, που να προσφέρει τις απαιτούμενες προδιαγραφές για κάθε υποσύστημα ξεχωριστά, για την ταυτόχρονη λειτουργία, όλων μαζί, αλλά και να επιτρέπει περιθώρια επέκτασης. Επίσης, θα πρέπει να περιλαμβάνει, επαρκής απομόνωση της τροφοδοσίας των ευαίσθητων ηλεκτρονικών υποσυστημάτων, από άλλα υποσυστήματα που εισάγουν θόρυβο στις γραμμές τροφοδοσίας, όπως οι κινητήρες και οι σερβοκινητήρες.

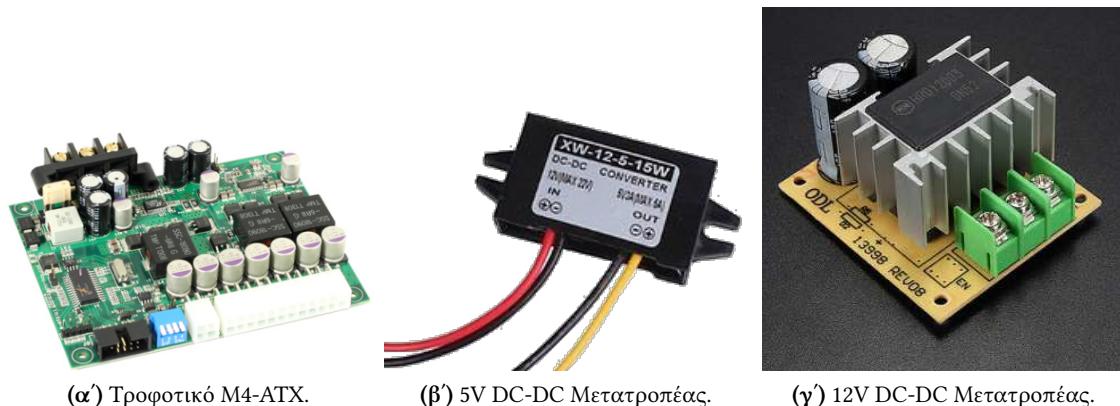
Όπως, παρουσιάζεται και στο σχήμα 2.29, ως πηγή τροφοδοσίας της ρομποτικής πλατφόρμας *Monstertruck*, χρησιμοποιείται μία μπαταρία Λιθίου-Πολυμερών (σχήμα 2.27) με ονομαστική τάση 22.2V, μέγιστη τάση 25.2V (100% φόρτιση) και 3700/4000/5000mAh. Η τροφοδοσία που παρέχει η μπαταρία, τροφοδοτείται σε έναν διακλαδωτή (*Motor Distribution Board*), από τον οποίο τροφοδοτούνται ο ελεγκτής *EPOS 24/1* που τροφοδοτεί και τον κινητήρα του οχήματος, ο 12V DC-DC μετατροπέας και το τροφοδοτικό M4-ATX.



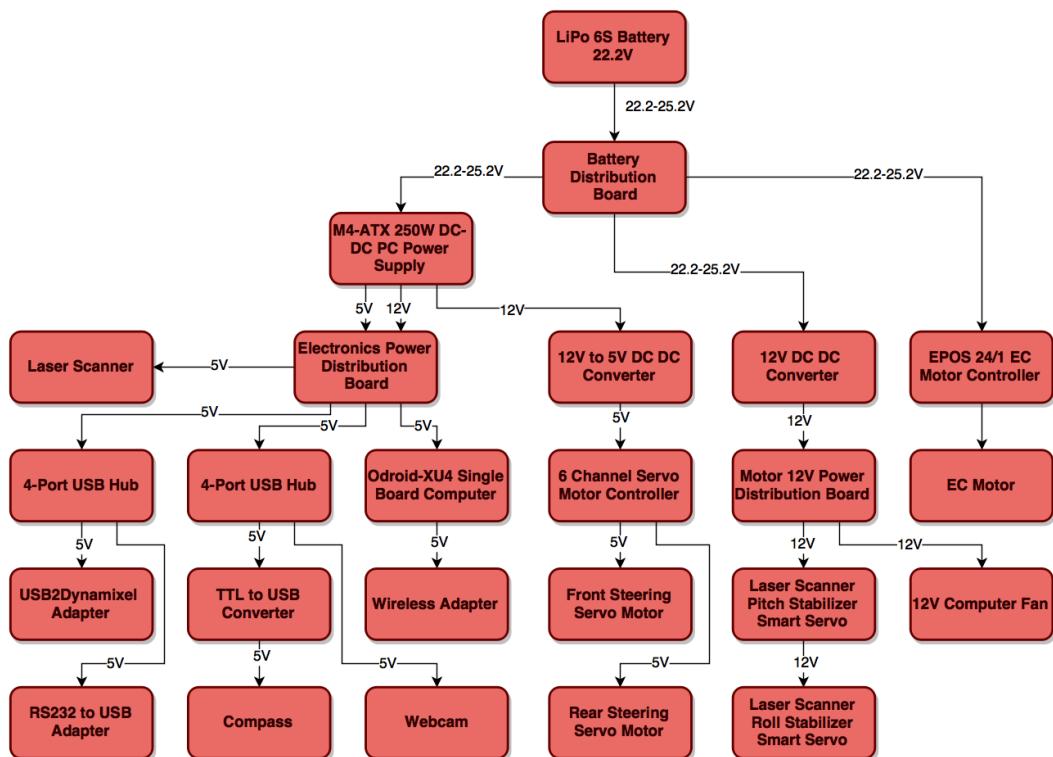
Σχήμα 2.27: Μπαταρία Gens ace, LiPo, 22.2V, 4000mAh.

Ο 12V DC-DC μετατροπέας τροφοδοτεί με 12V, μέσω ενός διακλαδωτή (*Motor Distribution Board*) τους έξι πνούς σερβοκινητήρες *Dynamixel*, του σταθεροποιητή *Pitch-Roll* του σαρωτή λέιζερ, όπως επίσης και έναν ανεμιστήρα, υπεύθυνο, για την ψύξη του υπολογιστή *Odroid-XU4*.

Το τροφοδοτικό M4-ATX, παράγει εξόδους τροφοδοσίας 5V και 12V και τροφοδοτεί την πλειονότητα των ηλεκτρονικών υποσυστημάτων της ρομποτικής πλατφόρμας. Αρχικά, τροφοδοτεί απευθείας έναν 5V DC-DC μετατροπέα, ο οποίος χρησιμοποιείται για να τροφοδοτεί τους σερβοκινητήρες του συστήματος στρέψης - τετραδιεύθυνσης της ρομποτικής πλατφόρμας, απομονώνοντας, ταυτόχρονα την τροφοδοσία των σερβοκινητήρων από την τροφοδοσία των υπόλοιπων ηλεκτρονικών υποσυστημάτων, που παρουσιάζουν ενασθησία στον θόρυβο. Τα υπόλοιπα ηλεκτρονικά υποσυστήματα της ρομποτικής πλατφόρμας, τροφοδοτούνται, από το τροφοδοτικό M4-ATX, μέσω ενός διακλαδωτή (Electronics Distribution Board), είτε άμεσα, όπως ο υπολογιστής Odroid-XU4, ο σαρωτής λέιζερ και οι διακλαδωτές USB (USB Hubs), είτε μέσω του υπολογιστή και των διακλαδωτών USB.



Σχήμα 2.28: Επιμέρους τμήματα συστήματος τροφοδοσίας.



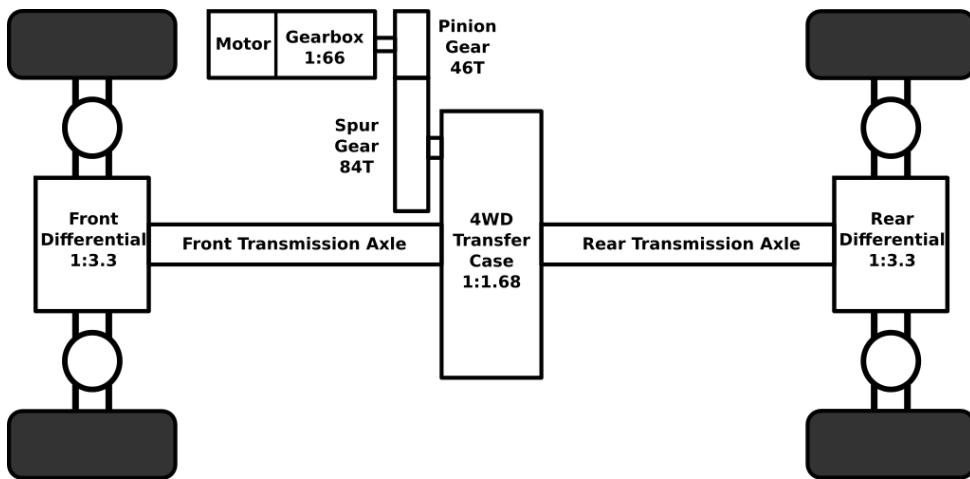
Σχήμα 2.29: Σύστημα Τροφοδοσίας της ρομποτικής πλατφόρμας *Monstertruck*.

2.2 Σύστημα Μετάδοσης Κίνησης

Ένα αυτοκίνητο όχημα, για να κινηθεί, απαιτεί την ύπαρξη ενεργοποιητών(*actuators*), οι οποίοι μετατρέπουν την ενέργεια από μία πηγή τροφοδοσίας και ένα σήμα ελέγχου σε μηχανική κίνηση. Τον σκοπό αυτό, εξυπηρετούν οι κινητήρες και στην προκειμένη περίπτωση, για την επίτευξη της μηχανικής κίνησης του υλοποιημένου ρομποτικού οχήματος, χρησιμοποιούνται ένας κινητήρας, ο οποίος, μέσω ενός συστήματος μετάδοσης κίνησης, μεταδίδει την περιστροφική κίνηση του και στους τέσσερις τροχούς του οχήματος (*tetrakínηση*) και δύο σερβοκινητήρες, οι οποίοι στρίβουν και τους τέσσερις τροχούς, με ανεξάρτητη στρέψη των μπροστινών, από τους πίσω τροχούς (*tetradiεύθυνση*). Στην συνέχεια, παρουσιάζεται η ανάλυση των μηχανισμών *tetrakínησης* και *tetradiεύθυνσης* της ρομποτικής πλατφόρμας *Monstertruck*.

2.2.1 Σύστημα Τετρακίνησης

Η μετάδοση της κίνησης, από τον μοναδικό κινητήρα του οχήματος, προς τους τέσσερις τροχούς, δηλαδή η τετρακίνηση επιτυγχάνεται, μέσω του συστήματος μετάδοσης κίνησης (*drivetrain*), όπως φαίνεται στο σχήμα 2.30. Το σύστημα αυτό περιλαμβάνει, συνολικά τέσσερα στάδια μετάδοσης της περιστροφικής κίνησης του κινητήρα, προς κάθε τροχό, όπου κάθε στάδιο εισάγει ένα λόγο μείωσης των στροφών, αλλά ταυτόχρονα, αντίστοιχο λόγο αύξησης της ροπής στρέψης.



Σχήμα 2.30: Σύστημα Μετάδοσης Κίνησης (Drivetrain) της ρομποτικής πλατφόρμας *Monstertruck*.

Το πρώτο στάδιο, αποτελεί το κιβώτιο ταχυτήτων - μειωτήρας (*gearbox*) του κινητήρα, το οποίο περιγράφεται από ένα λόγο μετάδοσης $\lambda_{gearbox} = 1 : 66$. Ο λόγος, αυτός, μειώνει την μέγιστη περιστροφική ταχύτητα του κινητήρα, από $18\,000\,rpm$ σε $18\,000 : 66 = 272.72\,rpm$.

Το δεύτερο στάδιο μετάδοσης, αποτελείται από δύο γρανάζια, το *Πινιόν* (*Pinion Gear*) και *Ωθησης* (*Spur Gear*). Το γρανάζι *Πινιόν*, μεταδίδει την κίνηση από τον κινητήρα στο γρανάζι *Ωθησης*, το οποίο, με τη σειρά του μεταδίδει την κίνηση στο επόμενο στάδιο μετάδοσης. Το γρανάζι *Πινιόν* περιλαμβάνει 46 οδοντώσεις, ενώ το γρανάζι *Ωθησης*, 81, έχοντας ως αποτέλεσμα ένα λόγο μετάδοσης $\lambda_{spur_pinion} = 46 : 81 = 1 : 1.76$. Με την μείωση του δεύτερου σταδίου, η μέγιστη ταχύτητα μειώνεται στα $272.72 : 1.76 = 154.96\,rpm$.

Το τρίτο στάδιο μετάδοσης και σημαντικότερο, για την επίτευξη *tetrakínησης*, περιλαμβάνει το κιβώτιο μετάδοσης (*Transfer Case*) του μπροστινού και του πίσω άξονα, που φαίνεται στο σχήμα 2.32. Το κιβώτιο μετάδοσης, περιλαμβάνει δύο γρανάζια, το γρανάζι μετάδοσης (*Transmission Gear*) και το



Σχήμα 2.31: Γρανάζια Πινιόν και Ωθησης.

διαφορικό γρανάζι (Differential Gear). Η περιστροφική κίνηση μεταδίδεται, από το γρανάζι Ωθησης, προς το γρανάζι μετάδοσης, μέσω ενός μηχανισμού σφιγκτήρα ολίσθησης (*slipper clutch*), που επιτρέπει την αποσύμπλεξη των γραναζιών, μέσω ολίσθησης, σε περίπτωση, υψηλής ροπής στους τροχούς, που θα μπορούσαν να προκαλέσουν ζημιά στα γρανάζια και στους άξονες μετάδοσης. Υπό φυσιολογικές συνθήκες, το γρανάζι μετάδοσης, μεταδίδει την περιστροφική κίνηση προς το διαφορικό γρανάζι και έπειτα προς τους δύο άξονες μετάδοσης, με λόγο μετάδοσης $\lambda_{transfer_case} = 1 : 1.68$. Επομένως, έχουμε μία επιπλέον μείωση των στροφών, με αποτέλεσμα $154.96 : 1.68 = 92.24 rpm$.



Σχήμα 2.32: Το κιβώτιο μετάδοσης κίνησης (Transfer Case) του οχήματος GroundPounder.

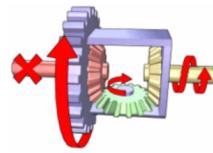
Το τέταρτο και τελευταίο στάδιο μετάδοσης της κίνησης, αποτελείται από δύο διαφορικά (*differential*), ένα για τους μπροστινούς τροχούς και ένα για τους πίσω. Το διαφορικό είναι ένας μηχανισμός, ο οποίος μετατρέπει την κατεύθυνση κίνησης, από την ευθύγραμμη, του άξονα μετάδοσης, στην εγκάρσια, των ημιαξόνων κάθε τροχού. Παράλληλα, επιτρέπει σε δύο τροχούς, έναν αριστερό και ένα δεξιό, να κινούνται με διαφορετική περιστροφική ταχύτητα ή ροπή, ανάλογα με την πρόσφυση σε κάθε έναν, από αυτούς. Ο μηχανισμός, αυτός, είναι απαραίτητος, καθώς, όταν ένα όχημα προσπαθεί να στρίψει, ακολουθώντας μία καμπύλη, οι τροχοί που βρίσκονται στην εξωτερική πλευρά της καμπύλης, διανύουν μεγαλύτερη απόσταση, από τους εσωτερικούς τροχούς και άρα θα πρέπει να κινούνται με μεγαλύτερη ταχύτητα. Τέλος, το κάθε διαφορικό εισάγει, ακόμη, έναν τελευταίο λόγο μείωσης των στροφών, της τάξης του $\lambda_{differential} = 1 : 3.3$, οπότε η τελική μέγιστη ταχύτητα περιστροφής των τροχών προκύπτει $92.24 : 3.3 = 27.95 rpm$.

Επομένως, η σχέση μετάδοσης της περιστροφικής ταχύτητας, από τον κινητήρα, στους τροχούς προκύπτει:

$$\omega_{wheel} = \omega_{motor} / (\lambda_{gearbox} \times \lambda_{spur_pinion} \times \lambda_{transfer_case} \times \lambda_{differential}) = \omega_{motor} / 644 \quad (2.1)$$



Σχήμα 2.33: Το διαφορικό (Differential) του οχήματος GroundPounder.

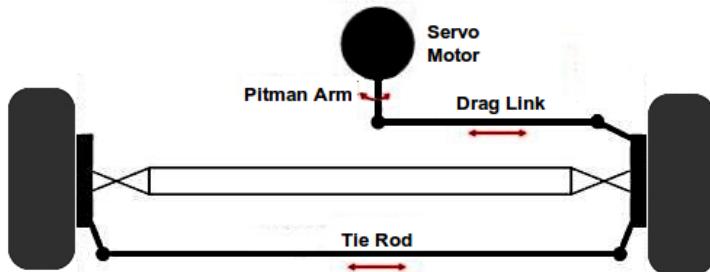


Σχήμα 2.34: Λειτουργία ενδεικτικού μηχανισμού διαφορικού.

2.2.2 Σύστημα Τετραδιεύθυνσης

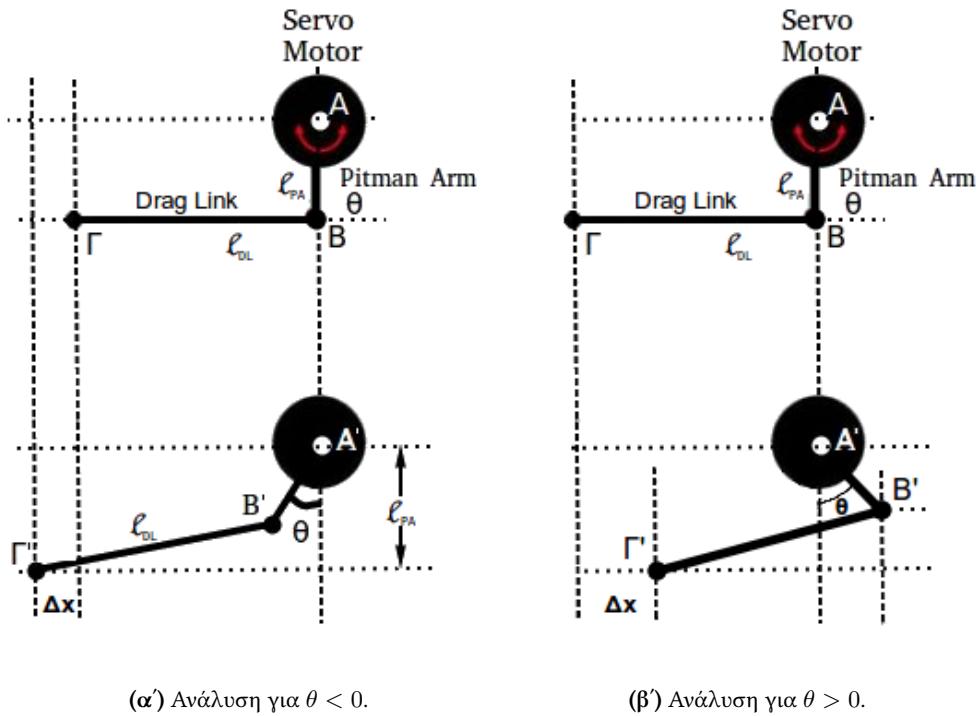
Η ρομποτική πλατφόρμα *Monstertruck*, περιλαμβάνει δύο σερβοκινητήρες, υπεύθυνους για την ανεξάρτητη στρέψη των μπροστινών και πίσω τροχών. Η ανεξάρτητη αυτή στρέψη, επιτρέπει στο όχημα να λειτουργεί με μπροστινή στρέψη (*Μπροστινοδιεύθυνση - FWS*), πίσω στρέψη (*Πίσωδιεύθυνση - RWS*), ή ταυτόχρονη στρέψη (*Τετραδιεύθυνση - 4WS*) των τροχών. Στην *τετραδιεύθυνση*, οι μπροστινοί τροχοί, μπορεί να στρίβουν, είτε, με την ίδια φορά με τους πίσω τροχούς, οπότε μιλάμε για *θετική τετραδιεύθυνση*, είτε με αντίθετη, οπότε μιλάμε για *αρνητική τετραδιεύθυνση*.

Εφόσον, υπάρχει ένας σερβοκινητήρας, για τους μπροστινούς τροχούς και ένας για τους πίσω, γεννάται το ερώτημα, πώς μεταδίδεται η στρέψη από έναν σερβοκινητήρα σε δύο τροχούς, έναν αριστερό και έναν δεξιό. Ο μηχανισμός, που λύνει το πρόβλημα, στην προκειμένη περίπτωση, ονομάζεται *Μηχανισμός Στρέψης*, μέσω *Συνδέσμου Έλξης* (*Drag Link Steering Mechanism*). Ο μηχανισμός αυτός, στην προκειμένη περίπτωση, αποτελείται από έναν σερβοκινητήρα, ένα μπράτσο *Pitman* (*Pitman Arm*), έναν σύνδεσμο έλξης (*Drag Link*) και έναν σύνδεσμο ένωσης των τροχών (*Tie Rod*), όπως επίσης και τις αρθρώσεις στρέψης των τροχών (*Wheel Steering Knuckles*), όπως και παρουσιάζεται στο σχήμα 2.35.



Σχήμα 2.35: Ο Μηχανισμός στρέψης, με όξονα έλξης (*Drag Link Steering Mechanism*) της ρομποτικής πλατφόρμας *Monstertruck*.

Η περιστροφική κίνηση του σερβοκινητήρα, μεταδίδεται μέσω του μπράτσου *Pitman* και μετατρέπεται σε μεταφορική κίνηση του συνδέσμου έλξης, η οποία με τη σειρά της μετατρέπεται σε στρέψη της άρθρωσης του δεξιού τροχού. Η στρέψη, τώρα, της δεξιάς άρθρωσης, παρασύρει τον σύνδεσμο ένωσης των αρθρώσεων στρέψης των τροχών σε μεταφορική κίνηση, η οποία, έχει σαν αποτέλεσμα την στρέψη και του αριστερού τροχού. Ακολούθως, αναλύεται η λειτουργία του μηχανισμού στρέψης, σε δύο βήματα. Πρώτα υπολογίζεται η μετατόπιση Δx του συνδέσμου έλξης, συναρτήσει της γωνίας στρέψης θ του σερβοκινητήρα και έπειτα, συναρτήσει της γωνίας στρέψης δ της άρθρωσης τροχού, με την οποία, είναι συνδεδεμένος ο σύνδεσμος έλξης.



Σχήμα 2.36: Μετάδοση κίνησης από τον σερβοκινητήρα και το μπράτσο Pitman στον συνδέσμο έλξης.

Λαμβάνοντας την παραδοχή, ότι η άκρη (σημείο Γ) του συνδέσμου έλξης, κινείται, μόνο οριζόντια και όχι κάθετα και με βάση το σχήμα 2.36 και απλή γεωμετρική ανάλυση, μπορεί να εξαχθεί η σχέση μεταξύ της γωνίας στρέψης θ του σερβοκινητήρα και της μετατόπισης Δx του συνδέσμου έλξης, ως:

$$\Delta x = \begin{cases} \sqrt{l_{DL}^2 - l_{PA}^2(1 - \cos(\theta))^2} + l_{PA} \sin(|\theta|) - l_{DL}, & \theta < 0 \\ 0, & \theta = 0 \\ -\sqrt{l_{DL}^2 - l_{PA}^2(1 - \cos(\theta))^2} - l_{PA} \sin(|\theta|) - l_{DL}, & \theta > 0 \end{cases} \quad (2.2)$$

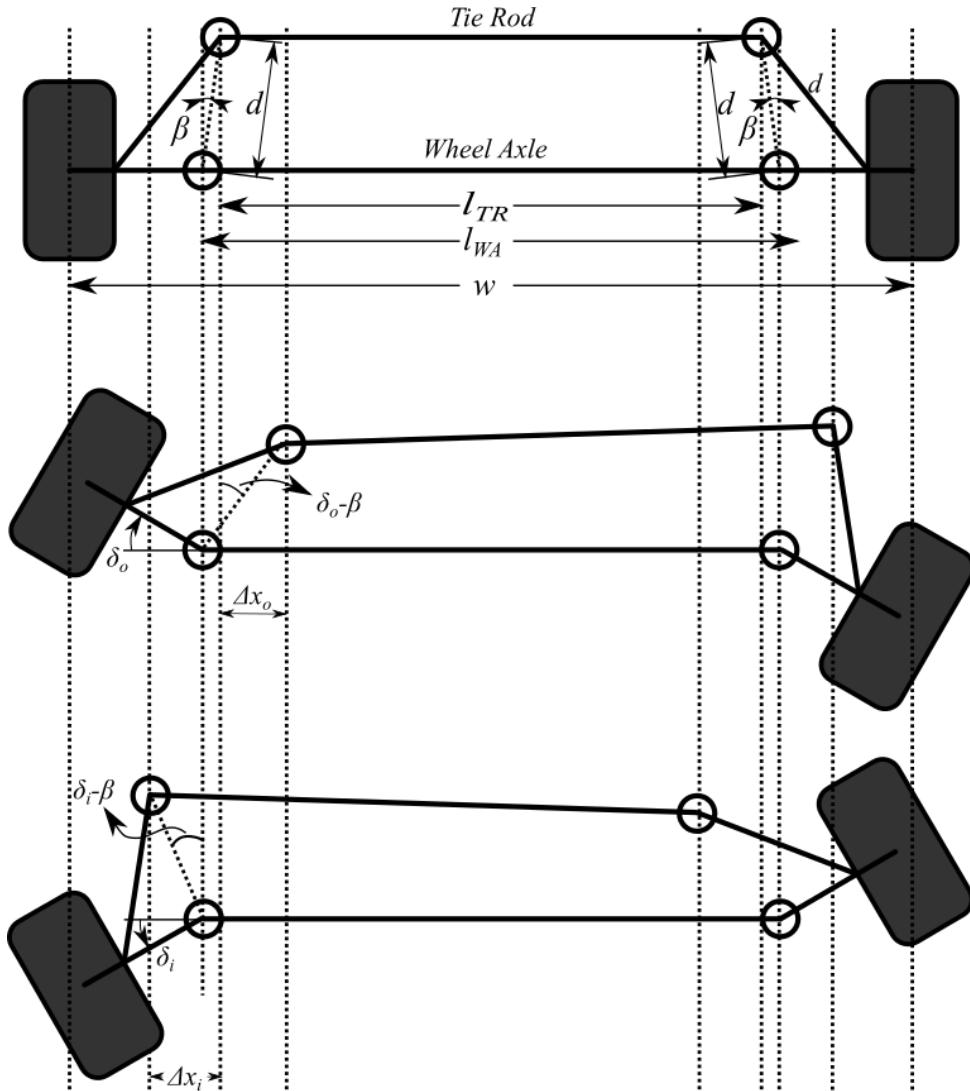
όπου l_{DL} είναι το μήκος του συνδέσμου έλξης και l_{PA} είναι το μήκος του μπράτσου *Pitman*.

Με βάση το σχήμα 2.37, μέσω γεωμετρικής ανάλυσης, προκύπτει ότι, η μετατόπιση του συνδέσμου έλξης Δx , συναρτήσει της γωνίας στρέψης, του άμεσα συνδεδεμένου τροχού, δηλαδή του αριστερού, στην προκειμένη περίπτωση, για τις περιπτώσεις, που ο τροχός είναι στην εσωτερική ή στην εξωτερική πλευρά της στροφής, λαμβάνεται μέσω των ακόλουθων σχέσεων.

$$\Delta x_i = d \sin(\delta_i - \beta) + d \sin(\beta) \quad (2.3)$$

$$(2.4)$$

$$\Delta x_o = d \sin(\delta_o + \beta) - d \sin(\beta) \quad (2.5)$$



Σχήμα 2.37: Τραπεζοειδής Μηχανισμός στρέψης των τροχών.

Εξισώνοντας, τώρα, τις σχέσεις (2.2), (2.3), (2.4), προκύπτει η σχέση μεταξύ της γωνίας στρέψης του σερβοκινητήρα και της γωνίας στρέψης του αριστερού τροχού, για τις περιπτώσεις που είναι εσωτερικά (δ_i) στην στροφή και εξωτερικά (δ_o):

$$\delta_i = \beta + \sin^{-1} \left(\frac{\sqrt{l_{DL}^2 - l_{PA}^2(1 - \cos(\theta))} + l_{PA} \sin |\theta| - l_{DL} - d \sin \beta}{d} \right) \quad (2.6)$$

$$\delta_o = -\beta + \sin^{-1} \left(\frac{-\sqrt{l_{DL}^2 - l_{PA}^2(1 - \cos(\theta))} - l_{PA} \sin |\theta| - l_{DL} + d \sin \beta}{d} \right) \quad (2.7)$$

Ο υπολογισμός της γωνίας στρέψης, του απέναντι τροχού, δηλαδή, στην προκειμένη περίπτωση, του δεξιού τροχού, προκύπτει από τις εξισώσεις τραπεζοειδούς μηχανισμού στρέψης τροχών [4]:

$$\sin(\beta + \delta_i) + \sin(\beta - \delta_o) = \frac{L}{d} + \sqrt{\left(\frac{L}{d} - w \sin \beta\right)^2 - \left(\cos(\beta - \delta_o) - \cos(\beta + \delta_i)\right)^2} \quad (2.8)$$

Αντικαθιστώντας την τιμή για την δ_i , στην εξίσωση (2.8) θα πρέπει να εφαρμοστεί, ένας επαναληπτικός αλγόριθμος, για την εύρεση της δ_o και αντίστροφα.

Για την απλοποίηση της όλης διαδικασίας μετατροπής της γωνίας στρέψης του σερβοκινητήρα, σε γωνίες στρέψης των τροχών και αντίστροφα, χρησιμοποιήθηκαν πολυωνυμικές προσεγγίσεις των σχέσεων μεταξύ αυτών, λύνοντας, επαναληπτικά και για όλες τις δυνατές τιμές, τις εξισώσεις (2.6), (2.7), (2.8) και για τις τιμές των παραμέτρων του μηχανισμού στρέψης που φαίνονται στον πίνακα 2.6.

Πίνακας 2.6: Παράμετροι του μηχανισμού μετάδοσης στρέψης των τροχών.

Παράμετρος	Τιμή
d	30mm
β	5°
l_{TR}	225mm
l_{WA}	230mm
l_{PA}	20mm
l_{DA}	100mm

$$\begin{aligned}
 \delta_{lf} &= -0.029 \cdot \theta_f^2 - 0.6515 \cdot \theta_f - 0.0006 \\
 \delta_{lr} &= 0.029 \cdot \theta_r^2 + 0.6515 \cdot \theta_r + 0.0006 \\
 \delta_{rf} &= -0.0058 \cdot \delta_{lf}^2 + 1.0203 \cdot \delta_{lf} \\
 \delta_{rr} &= 0.0058 \cdot \delta_{lr}^2 - 1.0203 \cdot \delta_{lr} \\
 \theta_f &= 0.1047 \cdot \delta_{lf}^2 + 1.5362 \cdot \delta_{lf} + -0.0009 \\
 \theta_r &= -0.1047 \cdot \delta_{lr}^2 - 1.5362 \cdot \delta_{lr} - 0.0009
 \end{aligned} \tag{2.9}$$

όπου

θ_f : γωνία στρέψης του μπροστινού σερβοκινητήρα

θ_r : γωνία στρέψης του πίσω σερβοκινητήρα

δ_{lf} : γωνία στρέψης του μπροστινού αριστερού τροχού

δ_{lr} : γωνία στρέψης του πίσω αριστερού τροχού

δ_{rf} : γωνία στρέψης του μπροστινού δεξιού τροχού

δ_{rr} : γωνία στρέψης του πίσω δεξιού τροχού

2.3 Κινηματική Ανάλυση

Ένα σύγχρονο αυτοκίνητο όχημα, στην πλειονότητα των περιπτώσεων, για να κινηθεί, περιλαμβάνει έναν κινητήρα, ο οποίος είναι υπεύθυνος για την περιστροφική κίνηση των μπροστινών (μπροστινοκίνηση), πίσω (πισωκίνηση) ή όλων (τετρακίνηση) των τροχών. Επίσης, περιλαμβάνει ένα σύστημα στρέψης των τροχών, είτε μπροστινών (μπροστινοδιεύθυνση), είτε πισινών (πισωδιεύθυνση), είτε και των τεσσάρων (τετραδιεύθυνση), έτσι ώστε να μπορεί να ακολουθεί καμπύλες τροχιές και

όχι μόνο ευθύγραμμες. Η κινηματική ανάλυση του οχήματος, που παρουσιάζεται στην παρούσα ενότητα, προσπαθεί να περιγράψει την επίδραση του ελέγχου κίνησης των τροχών του, στην κίνηση του οχήματος και στις μεταβολές της κατάστασης του.

Η κατάσταση ενός άκαμπτου (rigid) ρομποτικού οχήματος, συνήθως περιγράφεται, από έξι μεταβλητές, τις καρτεσιανές συντεταγμένες του x , y , z , ως προς ένα αυθαίρετο εξωτερικό σύστημα συντεταγμένων και τις γωνίες *Euler yaw, pitch, roll*. Στην προκειμένη περίπτωση, το πρόβλημα που εξετάζεται, περιορίζεται σε επίπεδο περιβάλλον και επομένως, ως κατάσταση του οχήματος, λαμβάνεται, η πόζα του q , η οποία περιγράφεται από τις καρτεσιανές συνταγμένες του x , y στο επίπεδο και τον προσανατολισμό του θ , ως προς αυθαίρετο εξωτερικό σύστημα συντεταγμένων.

$$\mathbf{q} = [x \ y \ \theta]^T \quad (2.10)$$

Αντίστοιχα, η ταχύτητα ενός ρομποτικού οχήματος στο επίπεδο, ως προς ένα αυθαίρετο εξωτερικό σύστημα συντεταγμένων ορίζεται ως η μεταβολή της πόζας του q , ως προς τον χρόνο.

$$\dot{\mathbf{q}} = [\dot{x} \ \dot{y} \ \dot{\theta}]^T \quad (2.11)$$

Ενώ, η ταχύτητα ενός ρομποτικού οχήματος, στο επίπεδο, ως προς το κέντρο μάζας του C είναι

$$\mathbf{v} = [v_{cx} \ v_{cy} \ \omega_c]^T \quad (2.12)$$

και αντιστοιχεί σε μία κυκλική τροχιά, ακτίνας R . Εφόσον, ένα διάνυσμα ταχυτήτων, αντιστοιχεί σε μία κυκλική τροχιά του κέντρου μάζας C του οχήματος, τότε, μία επιθυμητή καμπύλη τροχιά, μπορεί να προσεγγιστεί, από ένα σύνολο τόξων κύκλου και τις αντίστοιχες ταχύτητες τους.

2.3.1 Κινηματικό Μοντέλο Ackermann

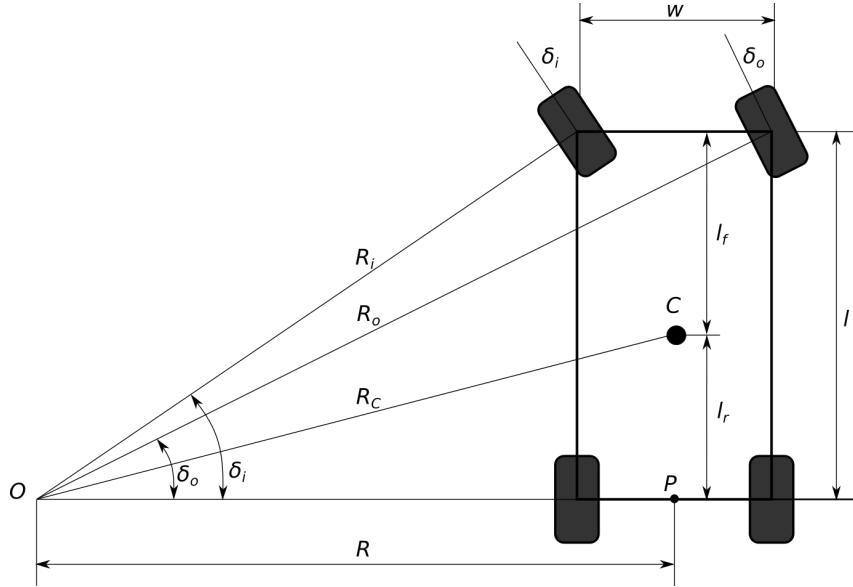
Το *Κινηματικό Μοντέλο Ackermann*, αποτελεί το δημοφιλέστερο και πιο διαδεδομένο κινηματικό μοντέλο στην αυτοκινητοβιομηχανία. Αναπτύχθηκε από τον Γερμανό μηχανικό Georg Lankensperger στο Μόναχο, το 1817, αλλά το δίπλωμα ευρεσιτεχνίας κατοχυρώθηκε από τον Rudolph Ackermann, το 1818, για ιππήλατες άμαξες. Τελικά, επεκτάθηκε και στην αυτοκινητοβιομηχανία και χρησιμοποιείται μέχρι και σήμερα.

Σκοπός του κινηματικού μοντέλου *Ackermann* είναι η αποφυγή της πλευρικής ολίσθησης των τροχών, ενός τετράτροχου οχήματος κατά την ακολούθηση καμπύλων τροχιών. Για την εξυπηρέτηση αυτού του σκοπού, λοιπόν, το κινηματικό μοντέλο *Ackermann*, στηρίζεται σε μία συνθήκη, την λεγόμενη συνθήκη *Ackermann*, μεταξύ των τροχών στρέψης ενός οχήματος, που αν ικανοποιείται προβλέπει την κίνηση των τροχών, χωρίς πλευρική ολίσθηση [4]. Η συνθήκη *Ackermann*, υποστηρίζει ότι για να κινείται ένα τετράτροχο όχημα χωρίς να ολισθαίνουν πλευρικά οι τροχοί του, θα πρέπει οι κάθετοι, στους τροχούς, άξονες να τέμνονται σε ένα κοινό σημείο, το οποίο ονομάζεται *Στιγμαίο Κέντρο Περιστροφής* (*Instantaneous Center of Rotation - ICR*) [5] και αποτελεί το κέντρο της στιγμαίας κυκλικής τροχιάς που ακολουθεί το όχημα.

Λαμβάνοντας υπόψιν το σχήμα 2.38 προκύπτουν οι γωνίες στρέψης των τροχών ως

$$\tan(\delta_i) = \frac{l}{R - \frac{w}{2}} \quad \text{ή} \quad \cot(\delta_i) = \frac{R - \frac{w}{2}}{l} \quad (2.13)$$

$$\tan(\delta_o) = \frac{l}{R + \frac{w}{2}} \quad \text{ή} \quad \cot(\delta_o) = \frac{R + \frac{w}{2}}{l} \quad (2.14)$$



Σχήμα 2.38: Κινηματικό Μοντέλο Ackermann.

Αφαιρώντας τις εξισώσεις (2.13), (2.14), προκύπτει η συνθήκη Ackermann.

$$\cot \delta_i - \cot \delta_o = w/l \quad (2.15)$$

όπου

δ_i : γωνία στρέψης εσωτερικού (inner), ως προς την στροφή, τροχού.

δ_o : γωνία στρέψης εξωτερικού (outer), ως προς την στροφή, τροχού.

L : μεταξόνιο (wheelbase).

w : μετατρόχιο (track).

R : ακτίνα τροχιάς που εκτελεί το μεσαίο σημείο μεταξύ των πίσω τροχών.

Το όχημα, που παρουσιάζεται στο σχήμα 2.38, πραγματοποιεί μία κυκλική τροχιά γύρω από το Στιγματό Κέντρο Περιστροφής O . Αν λάβουμε ως σημείο αναφοράς το κέντρο του πίσω άξονα (σημείο P), τότε το όχημα εκτελεί μία κυκλική τροχιά, ακτίνας R , γύρω από το σημείο O . Η ακτίνα R , μπορεί να υπολογιστεί προσθέτοντας τις εξισώσεις (2.13), (2.14), ως

$$R = l \cdot \frac{\cot \delta_i + \cot \delta_o}{2} = l \cot \delta = \frac{l}{\tan \delta} \quad (2.16)$$

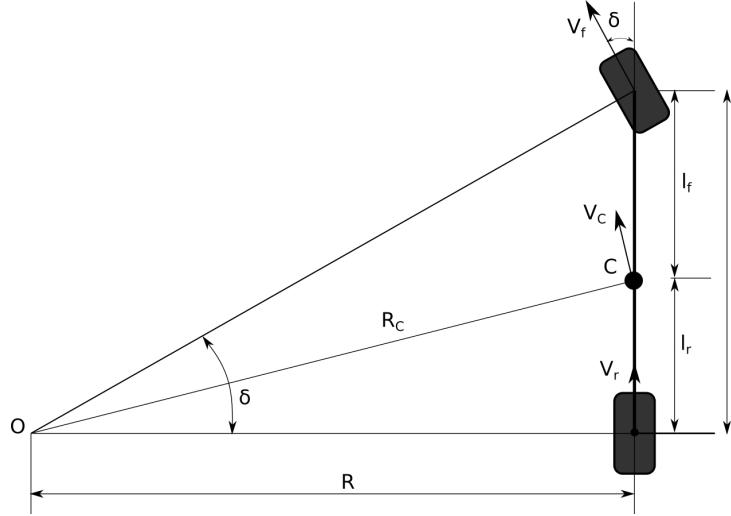
όπου δ είναι η γωνία στρέψης του ισοδύναμου κινηματικού μοντέλου ποδηλάτου (σχήμα 2.39).

Αντίστοιχα, ο υπολογισμός της ακτίνας R_c του κέντρου μάζας του οχήματος, με βάση το σχήμα 2.38, προκύπτει

$$R_c = \sqrt{R^2 + l_R^2} = \sqrt{R^2 + (l - l_F)^2} \quad (2.17)$$

όπου, l_R και l_F είναι η απόσταση του κέντρου μάζας του οχήματος από τον άξονα των μπροστινών και πίσω τροχών αντίστοιχα, όπου $l_R + l_F = l$.

Κατά την κίνηση σε κυκλική τροχιά, κάθε τροχός του οχήματος διαγράφει διαφορετική τροχιά από τους υπόλοιπους. Για παράδειγμα, στο σχήμα 2.38, για να καταστεί δυνατή η εκτέλεση της τροχιάς, γύρω από το O , χωρίς ολίσθηση, θα πρέπει οι τροχοί να κινούνται ίδια γωνιακή ταχύτητα $\dot{\theta}$,



Σχήμα 2.39: Ισοδύναμο μοντέλο ποδηλάτου Ackermann.

ως προς το κέντρο O και με διαφορετική γραμμική ταχύτητα, και άρα, καθένας, να περιστρέφεται με διαφορετική περιστροφική ταχύτητα ω , ανάλογη της ακτίνας της τροχιάς που διαγράφει. Επομένως, ισχύει

$$\dot{\theta}_{ir} = \dot{\theta}_{or} = \dot{\theta}_{if} = \dot{\theta}_{of} = \dot{\theta} \quad (2.18)$$

και αντικαθιστώντας στην εξίσωση (2.18), την σχέση μεταξύ γωνιακής και γραμμικής ταχύτητας

$$v = \dot{\theta} \cdot R \quad (2.19)$$

η εξίσωση (2.18) μετασχηματίζεται στη μορφή

$$\frac{v_{ir}}{R_{ir}} = \frac{v_{or}}{R_{or}} = \frac{v_{if}}{R_{if}} = \frac{v_{of}}{R_{of}} = \dot{\theta} = \frac{v_c}{R_c} \quad (2.20)$$

όπου

ir : εσωτερικός πίσω τροχός

or : εξωτερικός πίσω τροχός

if : εσωτερικός μπροστινός τροχός

of : εξωτερικός μπροστινός τροχός

Επομένως, με βάση τις δύο παραπάνω παρατηρήσεις, και την σχέση (2.20) μπορεί να υπολογιστεί η γραμμική ταχύτητα κάθε τροχού, ως

$$v_{if} = \dot{\theta} \cdot R_{if} = \dot{\theta} \cdot \sqrt{l^2 + (R - \frac{w}{2})^2} \quad (2.21)$$

$$v_{of} = \dot{\theta} \cdot R_{of} = \dot{\theta} \cdot \sqrt{l^2 + (R + \frac{w}{2})^2} \quad (2.22)$$

$$v_{ir} = \dot{\theta} \cdot R_{ir} = \dot{\theta} \cdot (R - \frac{w}{2}) \quad (2.23)$$

$$v_{or} = \dot{\theta} \cdot R_{or} = \dot{\theta} \cdot (R + \frac{w}{2}) \quad (2.24)$$

Χρησιμοποιώντας, τις εξισώσεις (2.21) - (2.24), όπως επίσης και την σχέση μεταξύ γραμμικής ταχύτητας v , περιστροφικής ταχύτητας ω και ακτίνας τροχού r

$$v = \omega \cdot r \quad (2.25)$$

προκύπτουν οι περιστροφικές ταχύτητες των τροχών, ως

$$\omega_{ir} = \frac{\dot{\theta}}{r} \cdot R_{ir} = \frac{\dot{\theta}}{r} \cdot \left(R - \frac{w}{2}\right) \quad (2.26)$$

$$\omega_{or} = \frac{\dot{\theta}}{r} \cdot R_{or} = \frac{\dot{\theta}}{r} \cdot \left(R + \frac{w}{2}\right) \quad (2.27)$$

$$\omega_{if} = \frac{\dot{\theta}}{r} \cdot R_{if} = \frac{\dot{\theta}}{r} \cdot \sqrt{l^2 + \left(R - \frac{w}{2}\right)^2} \quad (2.28)$$

$$\omega_{of} = \frac{\dot{\theta}}{r} \cdot R_{of} = \frac{\dot{\theta}}{r} \cdot \sqrt{l^2 + \left(R + \frac{w}{2}\right)^2} \quad (2.29)$$

Τέλος, οι ταχύτητες του κέντρου του άξονα των πίσω τροχών, ως προς ένα αυθαίρετο εξωτερικό σύστημα συντεταγμένων, βάση του *ισοδύναμου κινηματικού μοντέλου Ackermann*, είναι:

$$\dot{X}_r = v_r \cdot \cos \theta \quad (2.30)$$

$$\dot{Y}_r = v_r \cdot \sin \theta \quad (2.31)$$

$$\dot{\Theta}_r = \frac{v_r}{l} \cdot \tan \delta \quad (2.32)$$

Ενώ, οι ταχύτητες του κέντρου του άξονα των μπροστινών τροχών, ως προς ένα αυθαίρετο εξωτερικό σύστημα συντεταγμένων, είναι:

$$\dot{X}_f = v_f \cdot \cos(\theta + \delta) \quad (2.33)$$

$$\dot{Y}_f = v_f \cdot \sin(\theta + \delta) \quad (2.34)$$

$$\dot{\Theta}_f = \frac{v_f \cdot \sin \delta}{l} \quad (2.35)$$

2.3.2 Κινηματικό Μοντέλο Τετραδιεύθυνσης

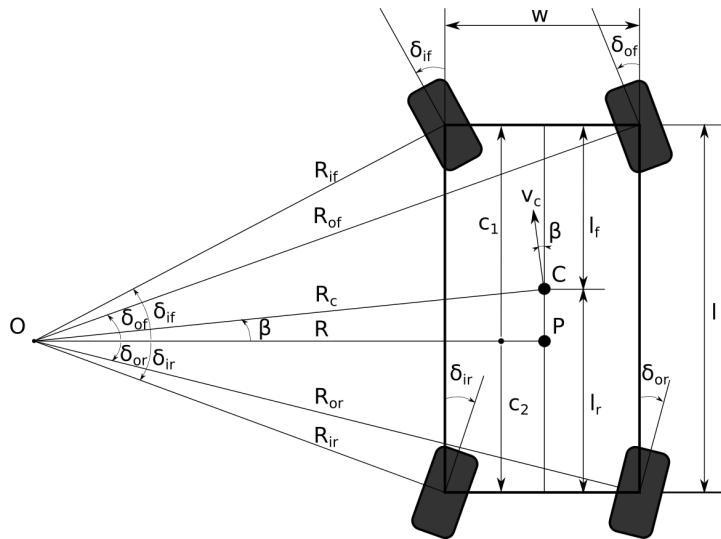
Το *κινηματικό μοντέλο τετραδιεύθυνσης* αποτελεί επέκταση του *κινηματικού μοντέλου Ackermann*, χρησιμοποιώντας, ταυτόχρονη στρέψη των μπροστινών και πίσω τροχών του οχήματος. Προσφέρει, με αυτόν τον τρόπο μεγαλύτερη ευελιξία, μέσω μειωμένης ακτίνας τροχιάς, συγκριτικά με το απλό μοντέλο *Ackermann* ($R_{4WS,min} < R_{2WS,min}$), ενώ, ακόμα, παρέχει και δυνατότητα διαγώνιας κίνησης, μέσω παραλληλης στρέψη των τροχών (crab steering). Επίσης, στην ειδική περίπτωση, που κάθε τροχός, μπορεί να κινηθεί και να στραφεί ανεξάρτητα από τους άλλους, το όχημα μπορεί να πραγματοποιήσει επί τόπου στροφή (0-point-turn, $R_{min} = 0$), κάτι που δεν εφαρμόζεται στην ρομποτική πλατφόρμα *Monstertruck*.

Στην παρούσα ενότητα, θα μας απασχολήσει η κινηματική ανάλυση του μοντέλου τετραδιεύθυνσης, για τις λειτουργίες, της *αρνητικής στρέψης* (counter/negative steering) και της *θετικής στρέψης* (same/positive/crab steering) των τροχών. Η αρνητική τετραδιεύθυνση, χρησιμοποιείται σε αυτοκίνητα, για χαμηλές ταχύτητες ($< 40 km/h$), με στόχο την αυξημένη ευελιξία, μέσω πραγματοποίησης πιο στενών ελιγμών. Αντίθετα, η θετική τετραδιεύθυνση, χρησιμοποιείται για, υψηλές ταχύτητες ($> 40 km/h$),

για πιο ομαλή αλλαγή λωρίδων, μέσω πιο μικρών μεταβολών στην ακτίνα της τροχιάς που εκτελεί. Σε ρομποτικές εφαρμογές, όπως και η παρούσα, όπου οι ταχύτητες είναι πολύ μικρότερες, το μοντέλο θετικής στρέψης μπορεί να χρησιμοποιηθεί σε αλγορίθμους κατασκευής μονοπατιών, για την επέκταση του ρεπερτορίου των δυνατών κινήσεων, αλλά και για την ενίσχυση αλγορίθμων διάσχισης μονοπατιού, μέσω διόρθωσης απόκλισης, σε περιπτώσεις παρεκκλίνουσας συμπεριφοράς, λόγω εξωτερικών παραγόντων, όπως ολίσθηση, ή ατελειών του κινηματικού ή του δυναμικού μοντέλου του οχήματος. Επίσης, στην ειδική περίπτωση θετικής τετραδιεύθυνσης, όπου οι πίσω τροχοί στρέφονται με την ίδια γωνία και φορά με τους μπροστινούς, το όχημα κινείται πλαγίως, με μηδενική γωνιακή ταχύτητα και άρα μηδενική μεταβολή προσανατολισμού.

Η κινηματική ανάλυση του μοντέλου τετραδιεύθυνσης, ακολουθεί ίδια κατεύθυνση με την κινηματική ανάλυση του μοντέλου Ackermann, μέσω μία αντίστοιχης συνθήκης στρέψης των τροχών (steering condition) [4]. Η συνθήκη αυτή, την οποία θα καλούμε, συνθήκη τετραδιεύθυνσης, ορίζει την σχέση μεταξύ των γωνιών στρέψης και των τεσσάρων τροχών, έτσι ώστε, οι, κάθετοι στους τροχούς, άξονες να τέμνονται σε ένα κοινό σημείο (σχήματα 2.40, 2.41).

Η ανάλυση του κινηματικού μοντέλου τετραδιεύθυνσης και η εξαγωγή των εξισώσεων και σχέσεων, που ακολουθεί στην συνέχεια, χρησιμοποιεί την διάταξη αρνητικής τετραδιεύθυνσης, που παρουσιάζεται στο σχήμα 2.40, αλλά ισχύει, παράλληλα και για την διάταξη θετικής τετραδιεύθυνσης, γεγονός που μπορεί να αποδειχθεί με απλή γεωμετρική ανάλυση του σχήματος 2.41.



Σχήμα 2.40: Κινηματικό μοντέλο αρνητικής τετραδιεύθυνσης.

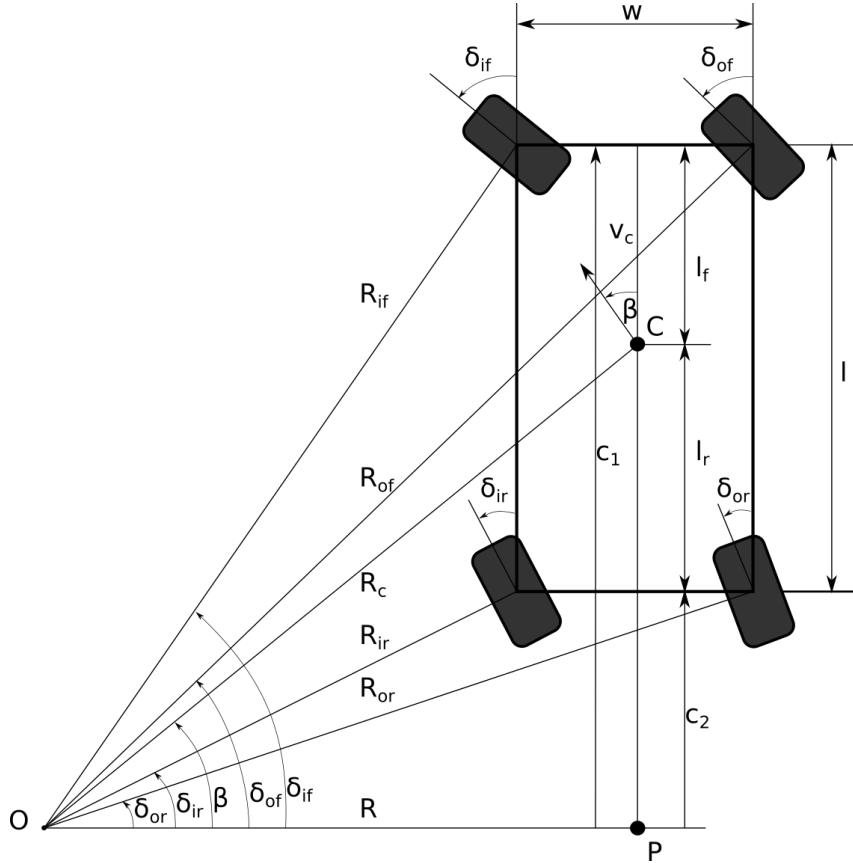
Μέσω γεωμετρικής ανάλυσης του σχήματος 2.40, βάση του ορισμού της συνθήκης τετραδιεύθυνσης, προκύπτουν οι γωνίες στρέψης των τροχών του οχήματος:

$$\tan \delta_{if} = \frac{c_1}{R - \frac{w}{2}} \quad (2.36)$$

$$\tan \delta_{of} = \frac{c_1}{R + \frac{w}{2}} \quad (2.37)$$

$$\tan \delta_{ir} = \frac{c_2}{R - \frac{w}{2}} \quad (2.38)$$

$$\tan \delta_{or} = \frac{c_2}{R + \frac{w}{2}} \quad (2.39)$$



Σχήμα 2.41: Κινηματικό μοντέλο θετικής τετραδιεύθυνσης.

Αντιστρέφοντας και αφαιρώντας τις σχέσεις (2.36), (2.37) και (2.38), (2.39) προκύπτουν οι συνθήκες τετραδιεύθυνσης των μπροστινών και πίσω τροχών, αντίστοιχα.

$$\cot \delta_{of} - \cot \delta_{if} = \frac{w}{c_1} \quad (2.40)$$

$$\cot \delta_{or} - \cot \delta_{ir} = \frac{w}{c_2} \quad (2.41)$$

Έπειτα, προσθέτοντας τις προκύπτουσες εξισώσεις (2.40), (2.41), λαμβάνεται η μαθηματική εξίσωση της συνθήκης τετραδιεύθυνσης.

$$\frac{1}{\cot \delta_{of} - \cot \delta_{if}} + \frac{1}{\cot \delta_{or} - \cot \delta_{ir}} = \frac{c_1 - c_2}{w} = \frac{l}{w} \quad (2.42)$$

Η γωνία β πλευρικής ολίσθησης (sideslip angle) [6] του κέντρου μάζας C του οχήματος, υπολογίζεται, εύκολα, από το ισοδύναμο μοντέλο ποδηλάτου τετραδιεύθυνσης (σχήμα 2.42).

$$\tan \beta = \frac{l_r \cdot \tan \delta_f + l_f \cdot \tan \delta_r}{l} \quad (2.43)$$

όπου

δ_f : γωνία στρέψης μπροστινού τροχού ισοδύναμου μοντέλου ποδηλάτου τετραδιεύθυνσης

δ_r : γωνία στρέψης πίσω τροχού ισοδύναμου μοντέλου ποδηλάτου τετραδιεύθυνσης

Οι γωνίες στρέψης δ_f και δ_r , μπορούν να υπολογιστούν για ένα όχημα με τετραδιεύθυνση ως

$$\cot \delta_f = \frac{\cot \delta_{if} + \cot \delta_{of}}{2} = \frac{R}{c_1} \quad (2.44)$$

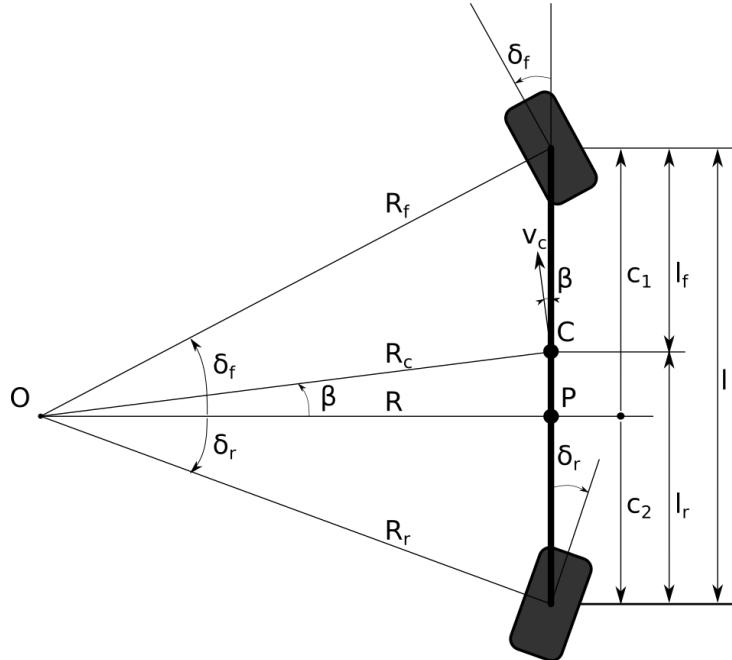
$$\cot \delta_r = \frac{\cot \delta_{ir} + \cot \delta_{or}}{2} = \frac{R}{c_2} \quad (2.45)$$

Αντιστρέφοντας και αφαιρώντας τις εξισώσεις (2.45), (2.44), υπολογίζεται η ακτίνα R της τροχιάς, του σημείου P

$$R = \frac{l}{\tan \delta_f - \tan \delta_r} \quad (2.46)$$

και έπειτα η ακτίνα R_c της τροχιάς του κέντρου μάζας C του οχήματος.

$$R_c = \frac{R}{\cos \beta} = \frac{l}{\cos \beta \cdot (\tan \delta_f - \tan \delta_r)} \quad (2.47)$$



Σχήμα 2.42: Ισοδύναμο μοντέλο ποδηλάτου αρνητικής τετραδιεύθυνσης.

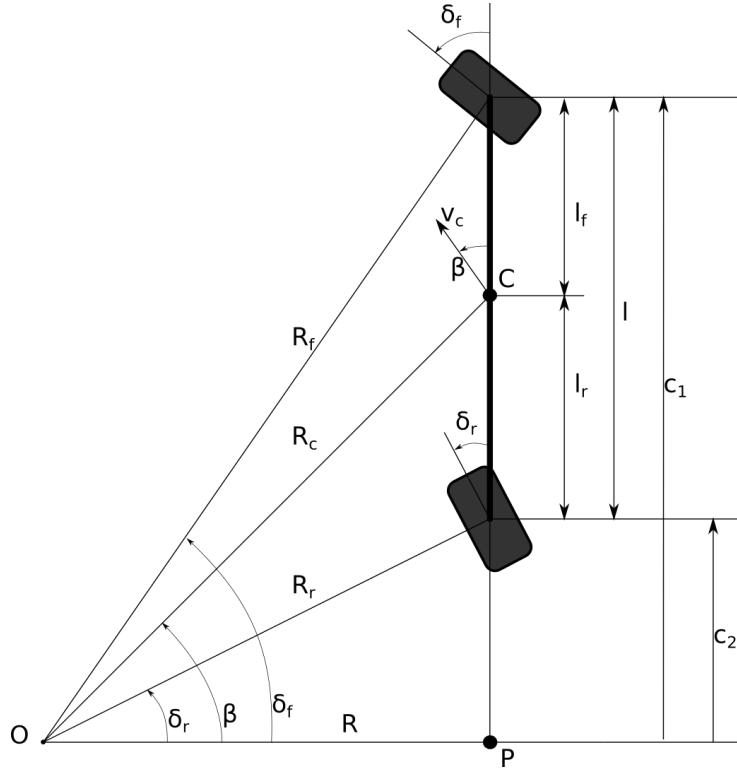
Για τον υπολογισμό των ταχυτήτων θα χρησιμοποιηθούν και πάλι, οι εξισώσεις (2.18)-(2.20), εφόσον ισχύουν και στην προκειμένη περίπτωση, σε συνδυασμό με τις σχέσεις των ακτίνων τροχιάς κάθε τροχού, όπως προκύπτει από το σχήμα 2.40.

$$R_{if} = \frac{R - \frac{w}{2}}{\cos \delta_{if}} \quad (2.48)$$

$$R_{of} = \frac{R + \frac{w}{2}}{\cos \delta_{of}} \quad (2.49)$$

$$R_{ir} = \frac{R - \frac{w}{2}}{\cos \delta_{ir}} \quad (2.50)$$

$$R_{or} = \frac{R + \frac{w}{2}}{\cos \delta_{or}} \quad (2.51)$$



Σχήμα 2.43: Ισοδύναμο μοντέλο ποδηλάτου θετικής τετραδιεύθυνσης.

Επομένως, οι γραμμικές ταχύτητες των τροχών προκύπτουν, ως

$$v_{if} = \dot{\theta} \cdot R_{if} = \frac{v_c}{R_c} \cdot \frac{R - \frac{w}{2}}{\cos \delta_{if}} \quad (2.52)$$

$$v_{of} = \dot{\theta} \cdot R_{of} = \frac{v_c}{R_c} \cdot \frac{R + \frac{w}{2}}{\cos \delta_{of}} \quad (2.53)$$

$$v_{ir} = \dot{\theta} \cdot R_{ir} = \frac{v_c}{R_c} \cdot \frac{R - \frac{w}{2}}{\cos \delta_{ir}} \quad (2.54)$$

$$v_{or} = \dot{\theta} \cdot R_{or} = \frac{v_c}{R_c} \cdot \frac{R + \frac{w}{2}}{\cos \delta_{or}} \quad (2.55)$$

και αντίστοιχα οι περιστροφικές ταχύτητες των τροχών, με βάση την εξίσωση (2.25), προκύπτουν

$$\omega_{if} = \frac{v_{if}}{r} = \frac{v_c}{r \cdot R_c} \cdot \frac{R - \frac{w}{2}}{\cos \delta_{if}} \quad (2.56)$$

$$\omega_{of} = \frac{v_{if}}{r} = \frac{v_c}{r \cdot R_c} \cdot \frac{R + \frac{w}{2}}{\cos \delta_{of}} \quad (2.57)$$

$$\omega_{ir} = \frac{v_{if}}{r} = \frac{v_c}{r \cdot R_c} \cdot \frac{R - \frac{w}{2}}{\cos \delta_{ir}} \quad (2.58)$$

$$\omega_{or} = \frac{v_{if}}{r} = \frac{v_c}{r \cdot R_c} \cdot \frac{R + \frac{w}{2}}{\cos \delta_{or}} \quad (2.59)$$

Με βάση την παραπάνω κινηματική ανάλυση, οι εξισώσεις των ταχυτήτων, του οχήματος, ως προς το κέντρο μάζας του C , μπορούν να υπολογιστούν, χρησιμοποιώντας τις σχέσεις

$$\tan \beta = \frac{v_{c,y}}{v_{c,x}} \Leftrightarrow v_{c,y} = v_{c,x} \cdot \tan \beta \quad (2.60)$$

$$v_c^2 = v_{c,x}^2 + v_{c,y}^2 \Leftrightarrow v_c^2 = v_{c,x}^2 - v_{c,x}^2 \cdot \tan^2 \beta \quad (2.61)$$

$$v_c = \omega_c \cdot R_c \quad (2.62)$$

από τις οποίες, τελικά προκύπτουν οι εξισώσεις των ταχυτήτων του οχήματος, ως προς το κέντρο μάζας του C :

$$v_{c,x} = \frac{v_c}{\sqrt{1 + \tan^2 \beta}} \quad (2.63)$$

$$v_{c,y} = \frac{v_c \cdot \tan \beta}{\sqrt{1 + \tan^2 \beta}} \quad (2.64)$$

$$\omega_c = \frac{v_c \cdot \cos \beta \cdot (\tan \delta_f - \tan \delta_r)}{l} \quad (2.65)$$

όπου

v_c : η συνισταμένη γραμμική ταχύτητα του κέντρου μάζας του οχήματος

$v_{c,x}$: η γραμμική επιμήκης ταχύτητα του κέντρου μάζας του οχήματος

$v_{c,y}$: η γραμμική εγκάρσια ταχύτητα του κέντρου μάζας του οχήματος

ω_c : η γωνιακή ταχύτητα του κέντρου μάζας του οχήματος

Τέλος, οι ταχύτητες κέντρου μάζας C του οχήματος, ως προς ένα αυθαίρετο εξωτερικό σύστημα συντεταγμένων, προκύπτουν βάση του *ισοδύναμου κινηματικού μοντέλου ποδηλάτου τετραδιεύθυνσης* (σχήμα 2.42) [7].

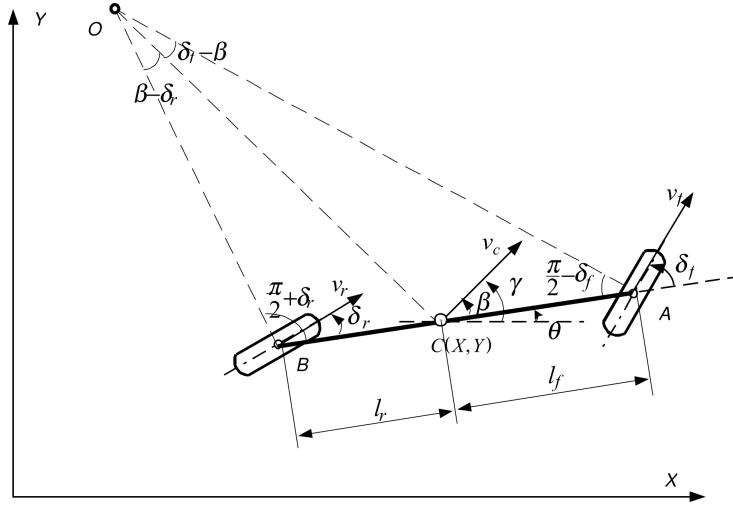
$$\dot{X} = v_c \cdot \cos(\theta + \beta) \quad (2.66)$$

$$\dot{Y} = v_c \cdot \sin(\theta + \beta) \quad (2.67)$$

$$\dot{\Theta} = \omega_c = \frac{v_c \cdot \cos \beta \cdot (\tan \delta_f - \tan \delta_r)}{l} \quad (2.68)$$

όπου,

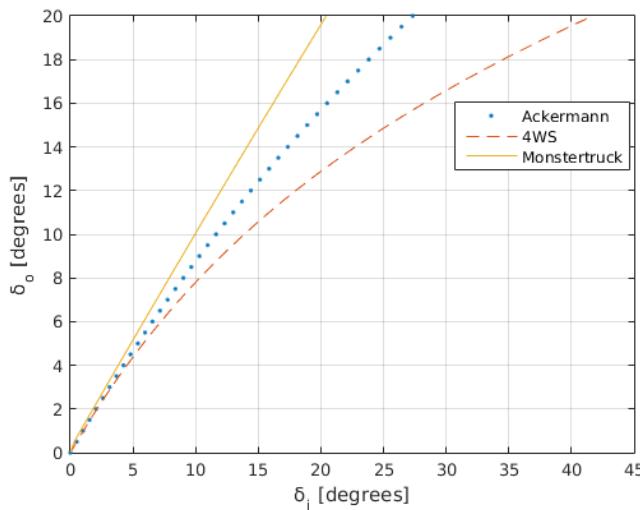
$$v_c = \frac{v_f \cdot \cos \delta_f + v_r \cos \delta_r}{2 \cdot \cos \beta} \quad (2.69)$$



Σχήμα 2.44: Ισοδύναμο κινηματικό μοντέλο ποδηλάτου τετραδιεύθυνσης στο επίπεδο XY [7].

2.3.3 Κινηματικό Μοντέλο Ρομποτικής Πλατφόρμας *Monstertruck*

Η ρομποτική πλατφόρμα *Monstertruck* περιλαμβάνει ένα μη ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης, με την έννοια, ότι δεν υπακούει στην συνθήκη τετραδιεύθυνσης (2.42). Το γεγονός αυτό, οφείλεται στο μηχανισμό μετάδοσης της στρέψης των τροχών που παρουσιάστηκε στην ενότητα 2.2.2 και ο οποίος ορίζει μία σχέση, μεταξύ εσωτερικού και εξωτερικού τροχού, με αρκετά μεγάλη απόκλιση από την ιδανική συνθήκη τετραδιεύθυνσης 2.42, ή ακόμα και από την ιδανική συνθήκη Ackermann 2.15 όπως παρουσιάζεται και στο σχήμα 2.45.

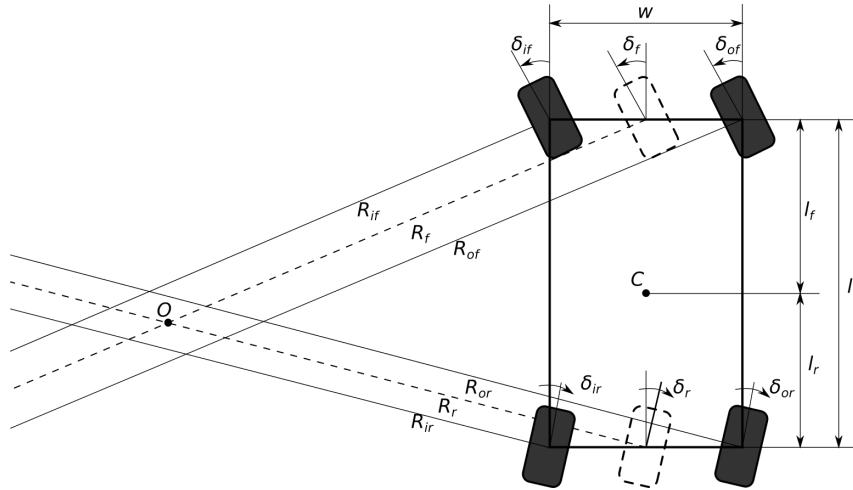


Σχήμα 2.45: Η σχέση στρέψης μεταξύ εσωτερικού και εξωτερικού τροχού για τα κινηματικά μοντέλα Ackermann, Τετραδιεύθυνσης και της ρομποτικής πλατφόρμας *Monstertruck*.

Εφόσον, παραβιάζεται η συνθήκη τετραδιεύθυνσης (2.42), η κίνηση της ρομποτικής πλατφόρμας *Monstertruck*, θα επιβαρύνεται από πλευρική ολίσθηση των τροχών. Παρόλα αυτά, επειδή, η ρομποτική πλατφόρμα, σχεδιάστηκε και προορίζεται για εφαρμογές εξαιρετικά μικρών ταχυτήτων και επομένως τυχόν δυναμικά φαινόμενα, που παρουσιάζονται, κατά την κίνηση, είναι αμελητέα, η κίνηση της ρομποτικής πλατφόρμας, μπορεί να προσεγγισθεί από το ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης.

Για την κινηματική ανάλυση του μοντέλου της ρομποτικής πλατφόρμας *Monstertruck*, θα χρησιμοποιήσουμε τις παραδοχές, ότι η διαφορά της γωνίας στρέψης μεταξύ δεξιού και αριστερού τροχού είναι αμελητέα και ότι σε χαμηλές ταχύτητες, η κίνηση της ρομποτικής πλατφόρμας, περιγράφεται, με αμελητέο σφάλμα, από τις εξισώσεις κίνησης του κινηματικού μοντέλου τετραδιεύθυνσης στο επίπεδο.

Το μη ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης της ρομποτικής πλατφόρμας *Monstertruck*, παρουσιάζεται στο σχήμα 2.46. Στο μοντέλο αυτό, το Στιγμαίο Κέντρο Περιστροφής O , βρίσκεται στην τομή των κάθετων στους τροχούς, αξόνων του ισοδύναμου κινηματικού μοντέλου ποδηλάτου τετραδιεύθυνσης. Σαν αποτέλεσμα η διεύθυνση κάθε τροχού είναι διάφορη της διεύθυνσης της ταχύτητας του κατά μία γωνία α , που ονομάζεται γωνία πλευρικής ολίσθησης τροχού (*side slip angle*) [5], όπως παρουσιάζεται στο σχήμα 2.47.



Σχήμα 2.46: Το κινηματικό μοντέλο της ρομποτικής πλατφόρμας *Monstertruck*, σε διάταξη αρνητικής τετραδιεύθυνσης.

Με βάση το μη ιδανικό μοντέλο τετραδιεύθυνσης, που παρουσιάζεται στο σχήμα 2.47, ο εσωτερικός τροχός στρέφεται κατά ίδια γωνία με τον αντίστοιχο εξωτερικό. Παρόλα αυτά, επειδή το όχημα, εκτελεί μία περιστροφική κίνηση, γύρω από το O , οι εξωτερικοί τροχοί διανύουν μεγαλύτερη απόσταση από τους εσωτερικούς, το οποίο σημαίνει ότι οι τροχοί ολισθαίνουν. Οι γωνίες της συνισταμένης ταχύτητας των τροχών δίνονται από τις ακόλουθες σχέσεις.

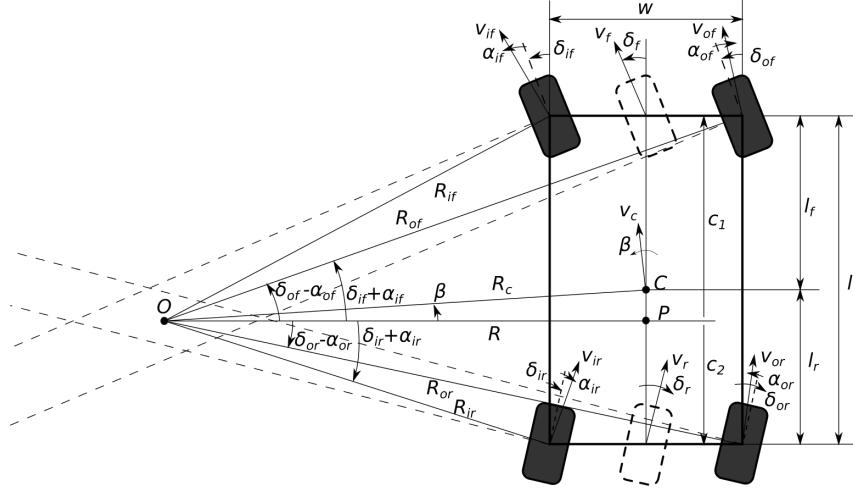
$$\cot(\delta_{if} + \alpha_{if}) = \frac{R - \frac{w}{2}}{c_1} \quad (2.70)$$

$$\cot(\delta_{of} - \alpha_{of}) = \frac{R + \frac{w}{2}}{c_1} \quad (2.71)$$

$$\cot(\delta_{ir} + \alpha_{ir}) = \frac{R - \frac{w}{2}}{c_2} \quad (2.72)$$

$$\cot(\delta_{or} - \alpha_{or}) = \frac{R + \frac{w}{2}}{c_2} \quad (2.73)$$

Αφαιρώντας τις (2.70), (2.71) και (2.72), (2.73) προκύπτουν οι σχέσεις τετραδιεύθυνσης του μη ιδανικού μοντέλου, για τους μπροστινούς και πίσω τροχούς.



Σχήμα 2.47: Το μη ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης, με πλευρική ολίσθηση τροχών, της ρομποτικής πλατφόρμας Monstertruck, σε διάταξη αρνητικής τετραδιεύθυνσης.

$$\cot(\delta_{of} - \alpha_{of}) - \cot(\delta_{if} + \alpha_{if}) = \frac{w}{c_1} \quad (2.74)$$

$$\cot(\delta_{or} - \alpha_{or}) - \cot(\delta_{ir} + \alpha_{ir}) = \frac{w}{c_2} \quad (2.75)$$

(2.76)

Έπειτα, αντιστρέφοντας και αφαιρώντας τις σχέσεις (2.74), (2.75) προκύπτει τελικά, η κινηματική συνθήκη του μη ιδανικού μοντέλου τετραδιεύθυνσης της ρομποτικής πλατφόρμας Monstertruck [5].

$$\frac{1}{\cot(\delta_{of} - \alpha_{of}) - \cot(\delta_{if} + \alpha_{if})} - \frac{1}{\cot(\delta_{or} - \alpha_{or}) - \cot(\delta_{ir} + \alpha_{ir})} = \frac{c_1 - c_2}{w} = \frac{l}{w} \quad (2.77)$$

Με βάση την παραδοχή, ότι το όχημα περιστρέφεται με συνισταμένη γραμμική ταχύτητα v_c και γωνιακή ταχύτητα ω_c , γύρω από το κέντρο O , υπολογίζονται αρχικά οι ακτίνες περιστροφής των τροχών γύρω από το κέντρο O .

$$R_{if} = \frac{R_c}{\cos(\delta_{if} + \alpha_{if})} \quad (2.78)$$

$$R_{of} = \frac{R_c}{\cos(\delta_{of} - \alpha_{of})} \quad (2.79)$$

$$R_{ir} = \frac{R_c}{\cos(\delta_{ir} + \alpha_{ir})} \quad (2.80)$$

$$R_{or} = \frac{R_c}{\cos(\delta_{or} - \alpha_{or})} \quad (2.81)$$

Έπειτα, μπορούμε να υπολογίσουμε τις συνισταμένες ταχύτητες κάθε τροχού γύρω από το κέντρο O , χρησιμοποιώντας την αντίστοιχη ακτίνα τροχιάς.

$$v_{if} = \dot{\theta} \cdot R_{if} = \frac{v_c}{R_c} \cdot \frac{R - \frac{w}{2}}{\cos(\delta_{if} - \alpha_{if})} \quad (2.82)$$

$$v_{of} = \dot{\theta} \cdot R_{of} = \frac{v_c}{R_c} \cdot \frac{R + \frac{w}{2}}{\cos(\delta_{of} - \alpha_{of})} \quad (2.83)$$

$$v_{ir} = \dot{\theta} \cdot R_{ir} = \frac{v_c}{R_c} \cdot \frac{R - \frac{w}{2}}{\cos(\delta_{ir} + \alpha_{ir})} \quad (2.84)$$

$$v_{or} = \dot{\theta} \cdot R_{or} = \frac{v_c}{R_c} \cdot \frac{R + \frac{w}{2}}{\cos(\delta_{or} - \alpha_{or})} \quad (2.85)$$

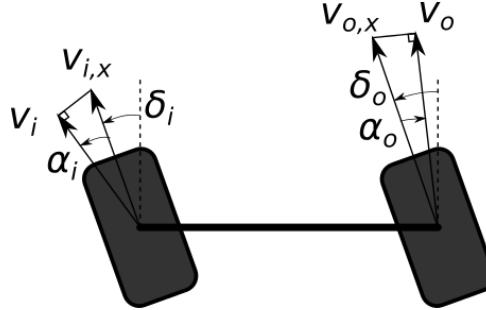
Επομένως, με βάση τις συνισταμένες ταχύτητες v και τις γωνίες πλευρικής ολίσθησης α , (σχήμα 2.48) μπορούμε να υπολογίσουμε τις επιμήκεις ταχύτητες των τροχών ως

$$v_{if,x} = \frac{v_{if}}{\cos \alpha_{if}} \quad (2.86)$$

$$v_{of,x} = \frac{v_{of}}{\cos \alpha_{of}} \quad (2.87)$$

$$v_{ir,x} = \frac{v_{ir}}{\cos \alpha_{ir}} \quad (2.88)$$

$$v_{or,x} = \frac{v_{or}}{\cos \alpha_{or}} \quad (2.89)$$



Σχήμα 2.48: Ολίσθηση τροχών, λόγω μη ιδανικού μηχανισμού στρέψης.

Τέλος, οι περιστροφικές ταχύτητες των τροχών, υπολογίζονται από τις ακόλουθες σχέσεις

$$\omega_{if} = \frac{v_{if,x}}{r} = \frac{v_c \cdot (R - \frac{w}{2})}{r \cdot R_c \cdot \cos \alpha_{if} \cdot \cos(\delta_{if} - \alpha_{if})} \quad (2.90)$$

$$\omega_{of} = \frac{v_{of,x}}{r} = \frac{v_c \cdot (R + \frac{w}{2})}{r \cdot R_c \cdot \cos \alpha_{of} \cdot \cos(\delta_{of} - \alpha_{of})} \quad (2.91)$$

$$\omega_{ir} = \frac{v_{ir,x}}{r} = \frac{v_c \cdot (R - \frac{w}{2})}{r \cdot R_c \cdot \cos \alpha_{ir} \cdot \cos(\delta_{ir} - \alpha_{ir})} \quad (2.92)$$

$$\omega_{or} = \frac{v_{or,x}}{r} = \frac{v_c \cdot (R + \frac{w}{2})}{r \cdot R_c \cdot \cos \alpha_{or} \cdot \cos(\delta_{or} - \alpha_{or})} \quad (2.93)$$

Οι παραπάνω τύποι, μπορούν να χρησιμοποιηθούν και για διαφορετικά κινηματικά μοντέλα τετραδιεύθυνσης, αντικαθιστώντας κατάλληλα τις γωνίες στρέψης δ και τις γωνίες πλευρικής ολίσθησης α . Επίσης, περιγράφουν και το ιδανικό κινηματικό μοντέλο τετραδιεύθυνσης για μηδενικές γωνίες πλευρικής ολίσθησης α των τροχών και για γωνίες στρέψης δ που υπακούν στις εξισώσεις (2.36)-(2.39).

Για τις γωνίες στρέψης της ρομπότικής πλατφόρμας Monstertruck, με βάση την παραδοχή της αμελητέας διαφοράς στρέψης, ισχύουν οι ακόλουθες εξισώσεις.

$$\delta_{if} = \delta_{of} = \delta_f \quad (2.94)$$

$$\delta_{ir} = \delta_{or} = \delta_r \quad (2.95)$$

Επομένως, οι γωνίες πλευρικής ολίσθησης προκύπτουν ως

$$\alpha_{if} = \tan^{-1}\left(\frac{c_1}{R - \frac{w}{2}}\right) - \delta_f \quad (2.96)$$

$$\alpha_{of} = -\tan^{-1}\left(\frac{c_1}{R + \frac{w}{2}}\right) + \delta_f \quad (2.97)$$

$$\alpha_{ir} = \tan^{-1}\left(\frac{c_2}{R - \frac{w}{2}}\right) - \delta_r \quad (2.98)$$

$$\alpha_{or} = -\tan^{-1}\left(\frac{c_2}{R + \frac{w}{2}}\right) + \delta_r \quad (2.99)$$

Τέλος, οι ταχύτητες του κέντρου μάζας C του οχήματος, ως προς ένα αυθαίρετο σύστημα συντεταγμένων, υπολογίζονται από τους τύπους του ιδανικού κινηματικού μοντέλου τετραδιεύθυνσης (2.66)-(2.68).

Κεφάλαιο 3

Αυτόνομη Πλοήγηση σε Άγνωστο Περιβάλλον

Στο παρόν κεφάλαιο θα παρουσιαστούν μέθοδοι και αλγόριθμοι που χρησιμοποιήθηκαν στη ρομποτική πλατφόρμα Monstertuck, για να καταστεί δυνατή η αυτόνομη πλοήγηση της σε ένα άγνωστο περιβάλλον. Θα μας απασχολήσουν θέματα σχετικά με τον εντοπισμό της θέσης του ρομποτικού οχήματος στον χώρο και την χαρτογράφηση του χώρου αυτού, μέσω μετρήσεων από τους αισθητήρες του ρομπότ αλλά και την ασφαλή και κινηματικά εφικτή αυτόνομη πλοήγηση μέσα σε αυτόν, αποσκοπώντας στην άφιξη σε έναν δεδομένο στόχο, ή την πλήρη εξερεύνηση του χώρου.

3.1 Εντοπισμός Θέσης και Χαρτογράφηση

Κατά την αυτόνομη πλοήγηση, ένα ρομποτικό όχημα θα πρέπει να γνωρίζει την θέση και τον προσανατολισμό του, δηλαδή την πόζα του, ως προς ένα συγκεκριμένο αδρανειακό πλαίσιο. Το πρόβλημα εντοπισμού θέσης (localization), συνήθως λύνεται βάσει ενός συνόλου μεθόδων που παρέχουν πληροφορία, σχετικά με την πόζα του ρομπότ. Η πληροφορία αυτή μπορεί να είναι είτε απόλυτη (global localization), είτε σχετική (local localization) [2]. Για παράδειγμα, αισθητήρες GPS, παρέχουν απόλυτη πληροφορία για την πόζα ενός ρομποτικού οχήματος, με βάση ορόσημα (landmarks). Αντίθετα, αισθητήρες, που μετρούν ταχύτητες ή ανιχνεύουν οπτικά χαρακτηριστικά στο περιβάλλον, παρέχουν μία σχετική πληροφορία της νέας πόζας του ρομποτικού οχήματος, ως προς την προηγούμενη.

Συνήθως, στις σύγχρονες ρομποτικές εφαρμογές, χρησιμοποιείται πληθώρα αισθητήρων και άρα πηγών πληροφορίας, είτε σχετική, είτε απόλυτη, σχετικά με την πόζα του ρομποτικού οχήματος και επομένως απαιτούνται μέθοδοι συνδυασμού και συγχώνευσης της πληροφορίας από όλες τις πηγές, για την παραγωγή μίας πιο αξιόπιστης εκτίμησης της κατάστασης του. Για την εξυπηρέτηση του σκοπού αυτού, ιδιαίτερα δημοφιλή λύση, αποτελεί η χρήση πιθανοτικών μεθόδων, όπως τα Φίλτρα Kalman, ο Εντοπισμός Θέσης Markov (Markov Localization) και Monte Carlo (Monte Carlo Localization).

Για να μπορεί ένα ρομπότ να ξέρει ανά πάσα στιγμή την θέση του και να μπορεί να πλοηγηθεί σε ένα άγνωστο περιβάλλον για να φτάσει σε κάποιον στόχο, αποφεύγοντας, ταυτόχρονα, τυχόν εμπόδια, θα πρέπει να διαθέτει μία μέθοδο χαρτογράφησης του περιβάλλοντος του. Επομένως, η χαρτογράφηση αποτελεί μία απαραίτητη ικανότητα για κάθε αυτόνομο ρομπότ, που προορίζεται για χρήση σε άγνωστο περιβάλλον. Μία αξιόπιστη μέθοδος χαρτογράφησης, όμως, προϋποθέτει και μία αποδοτική και εύρωστη μέθοδο εντοπισμού θέσεις. Γι αυτό το λόγο, υπάρχει το ευρέως μελετημένο αντικείμενο αλγορίθμων Ταυτόχρονης Χαρτογράφησης και Εντοπισμού Θέσης (SLAM).

3.1.1 Οδομετρία

Η Οδομετρία είναι μία μέθοδος υπολογισμού της μεταβολής της πόζας ενός ρομποτικού οχήματος. Οι πιο δημοφιλείς μέθοδοι οδομετρίας, σε ρομποτικές εφαρμογές, είναι η οδομετρία τροχών (wheel

odometry) και η οπτική οδομετρία (visual odometry). Η οδομετρία τροχών χρησιμοποιεί τις μετρήσεις των ταχυτήτων των τροχών ενός ρομποτικού οχήματος για την εξαγωγή των ταχυτήτων αυτού και μέσω ολοκλήρωσης των ταχυτήτων, υπολογίζει την μεταβολή της πόζας του οχήματος. Αντίστοιχα, η οπτική οδομετρία, χρησιμοποιεί μία κάμερα για τον υπολογισμό της μεταβολής της πόζας του οχήματος, μέσω των μεταβολών, μεταξύ διαδοχικών εικόνων. Στην προκειμένη περίπτωση, παρόλα αυτά, θα μας απασχολήσει μόνο η περίπτωση της οδομετρίας τροχών.

Η οδομετρία τροχών απαιτεί την μέτρηση της ταχύτητας κάθε τροχού και με βάση το κινηματικό μοντέλο την εξαγωγή των ταχυτήτων του οχήματος. Παρόλα αυτά, η ρομποτική πλατφόρμα Monstetruck, δεν περιλαμβάνει ξεχωριστούς κινητήρες και αισθητήρες μέτρησης ταχύτητας για κάθε τροχό, αλλά μόνο έναν κινητήρα του συστήματος τετρακίνησης, που μεταδίδει την κίνηση και στους τέσσερις τροχούς. Επομένως, θα λάβουμε την παραδοχή, ότι οι ταχύτητες κάθε τροχού υπακούν στις εξισώσεις (2.82)-(2.85) του κινηματικού μοντέλου της ρομποτικής πλατφόρμας Monstetruck, με βάση την μέτρηση της ταχύτητας που παρέχει ο εν λόγω κινητήρας.

Η γραμμική ταχύτητα του οχήματος προκύπτει από την μέτρηση της ταχύτητας του κινητήρα, ως

$$v_c = \frac{\omega_{motor}}{644 \cdot r} \quad (3.1)$$

ενώ οι γωνίες στρέψης $\delta_l f, \delta_r f, \delta_l r, \delta_r r$ των τροχών, υπολογίζονται από τις γωνίες στρέψεις των σερβοκινητήρων και τις σχέσεις (2.9).

Με βάση τις γωνίες στρέψης των τροχών, μπορούμε να υπολογίσουμε τις γωνίες στρέψης δ_f, δ_r του ισοδύναμου κινηματικού μοντέλου ποδηλάτου τετραδιεύθυνσης, μέσω των τύπων (2.44), (2.45) και, επομένως, την γωνία πλευρικής ολίσθησης β του κέντρου μάζας C του οχήματος, βάσει του τύπου (2.43).

Έπειτα, μέσω της γωνίας πλευρικής ολίσθησης β , των γωνιών στρέψης δ_f, δ_r και της συνισταμένης γραμμικής ταχύτητας του κέντρου μάζας C του οχήματος v_c , μπορούμε να υπολογίσουμε τις συνιστώσες της γραμμική ταχύτητας v_x, v_y , αλλά και την γωνιακή ταχύτητα ω_c του οχήματος, μέσω των τύπων (2.63)-(2.65).

Με βάση τους K. Bohlmann et al. [6], μέσω γεωμετρικής προσέγγισης των κυκλικών τροχιών για μικρά διαστήματα Δt , ως γραμμικές τροχιές, η νέα πόζα του οχήματος, μπορεί να υπολογιστεί, σε σχέση με την προηγούμενη, βάσει των ακόλουθων τύπων.

$$X_i = X_{i-1} + \Delta d \cdot \cos(\Theta_{i-1} + \frac{\Delta\Theta}{2} + \beta_{i-1} + \frac{\Delta\beta}{2}) \quad (3.2)$$

$$Y_i = Y_{i-1} + \Delta d \cdot \sin(\Theta_{i-1} + \frac{\Delta\Theta}{2} + \beta_{i-1} + \frac{\Delta\beta}{2}) \quad (3.3)$$

$$\Theta_i = \Theta_{i-1} + \Delta\Theta \quad (3.4)$$

όπου

$$\Delta d = v_c \cdot \Delta T \quad (3.5)$$

$$\Delta\Theta = \omega_c \cdot \Delta T \quad (3.6)$$

Τέλος, το γενικό πρόβλημα με την οδομετρία έγκειται στην ολοκλήρωση των ταχυτήτων, όπου, λόγω, ολίσθησης, σφαλμάτων μετρήσεων και ατελειών κινηματικού ή δυναμικού μοντέλου, εισάγεται ολοκληρωτικό σφάλμα που αυξάνεται όλο και περισσότερο με το πέρασμα του χρόνου. Το γεγονός αυτό, καθιστά την μέθοδο της οδομετρίας τροχών, ανεπαρκή ως αυτοτελή μέθοδο εντοπισμού θέσης.

Πολλές φορές παράλληλα, με την οδομετρία τροχών χρησιμοποιούνται και αισθητήρες κατεύθυνσης (πυξίδα, IMU), που μπορεί να παρέχουν μετρήσεις της γωνιακής ταχύτητας, γραμμικής επιτάχυνσης και απόλυτου προσανατολισμού ως προς τους άξονες x,y,z.

Στην ρομποτική πλατφόρμα Monstertruck, όπως προαναφέρθηκε στην ενότητα 2.1.3, χρησιμοποιείται μία πυξίδα Compass OS4000, που διαθέτει μαγνητόμετρο και επιταχυνσιόμετρο και άρα παρέχει πληροφορία για τον απόλυτο προσανατολισμό yaw, pitch, roll και την γραμμική επιτάχυνση a_x, a_y, a_z , ως προς τους άξονες x,y,z. Επίσης, οι γραμμικές επιταχύνσεις, μπορούν να ολοκληρωθούν, για τον υπολογισμό της μεταβολής των γραμμικών ταχυτήτων u_x, u_y, u_z του οχήματος, ενώ με δεύτερη ολοκλήρωση, για τον υπολογισμό της μεταβολής της θέσης X, Y, Z. Παρόλα αυτά, οι εν λόγω μετρήσεις, δεν είναι αρκετά αξιόπιστες και είναι επιρρεπείς σε συσσωρευτικά σφάλματα, λόγω ολοκλήρωσης.

Οι μετρήσεις της πυξίδας, μπορούν να χρησιμοποιηθούν, συμπληρωματικά με τις μετρήσεις της οδομετρίας τροχών, με στόχο την εξαγωγή της πλήρους κατάστασης $q = [x, y, z, roll, pitch, yaw]$ του οχήματος και τον πιο αξιόπιστο εντοπισμό θέσης, σε περίπτωση φαινομένων ολίσθησης των τροχών.

3.1.2 Εκτίμηση Κατάστασης με Συνδυαστική Αντίληψη

Για την εκτίμηση της κατάστασης ενός ρομπότ, μπορούν να χρησιμοποιηθούν μία πληθώρα αλγόριθμων που αξιοποιούν διαφορετικούς αισθητήρες, όπως κωδικοποιητές τροχών, πυξίδες, IMU, GPS, κάμερες, σαρωτές λέζερ κ.α., που καθένας, παρόλα αυτά, παρουσιάζει αιβεβαιότητα στις εκτιμήσεις του. Η συνδυαστική αντίληψη (sensor fusion) χρησιμοποιεί τις εκτιμήσεις που παράγονται με βάση τους διάφορους αισθητήρες που χρησιμοποιούνται και παράγει μία βελτιωμένη εκτίμηση της κατάστασης του ρομπότ. Μία δημοφιλής μέθοδος συνδυαστικής αντίληψης αποτελεί η χρήση πιθανοτικών φίλτρων για την εκτίμηση του μέσου διανύσματος κατάστασης και του πίνακα συμμεταβλητότητας.

Τα φίλτρα Kalman αποτελούν μία εξαιρετικά δημοφιλή επιλογή για εκτίμηση κατάστασης σε ρομποτικές εφαρμογές. Στη ρομποτική, τα Φίλτρα Kalman περιλαμβάνουν ένα μοντέλο κίνησης και ένα μοντέλο μετρήσεων, όπου το μοντέλο κίνησης αναπαριστά την σχέση μεταξύ των εντολών κίνησης και της κατάστασης του ρομπότ, ενώ το μοντέλο μετρήσεων αποτελείται από τα δεδομένα που προέρχονται από τους αισθητήρες, όσον αφορά την θέση του ρομπότ. Τα φίλτρα Kalman, χρησιμοποιούν έναν αλγόριθμο πρόβλεψης δύο σταδίων που επαναλαμβάνεται μετά από κάθε χρονικό βήμα και ανανέωση των μετρήσεων. Πρώτον, προβλέπει την εκτίμηση κατάστασης και τον πίνακα συμμεταβλητότητας με βάση τις προηγούμενες καταστάσεις για την εκτίμηση της νέας κατάστασης. Έπειτα, όταν έρθει κάποια μέτρηση από τους αισθητήρες, διορθώνει την προβλεπόμενη εκτίμηση.

Τα παραδοσιακά φίλτρα Kalman μπορούν να χειριστούν μονάχα γραμμικά συστήματα και επειδή τα περισσότερα ρομποτικά συστήματα είναι άκρως μη γραμμικά, η χρήση τους είναι προβληματική. Για αυτό το λόγο αναπτύχθηκαν παραλλαγές των φίλτρων Kalman, όπως τα Εκτεταμένα Φίλτρα Kalman (EKF). Για τον χειρισμό των μη γραμμικών συστημάτων, τα EKF γραμμικοποιούν τους μη γραμμικούς μετασχηματισμούς και αντικαθιστούν τους Ιακωβιανούς Πίνακες με τους γραμμικούς μετασχηματισμούς στα παραδοσιακά Φίλτρα Kalman. Παρόλα αυτά, οι μέθοδοι αυτοί είναι ανεπαρκής για τον εκ νέου εντοπισμό της θέσης του ρομπότ, ενώ, επίσης, είναι επιρρεπείς σε αποκλίσεις σε περίπτωση προβληματικής αρχικοποίησης και επομένως δεν μπορούν να επανέλθουν σε περίπτωση αποτυχίας.

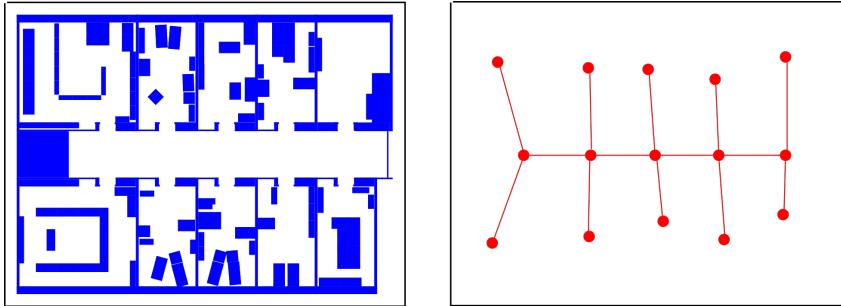
Στην ρομποτική πλατφόρμα Monstertruck χρησιμοποιείται ένα EKF για την πιθανοτική εκτίμηση της κατάστασης του ρομπότ, βάσει των μετρήσεων του συστήματος οδομετρίας των τροχών και των μετρήσεων της πυξίδας που αναφέρθηκαν στην προηγούμενη ενότητα.

3.1.3 Ταυτόχρονη Χαρτογράφηση και Εντοπισμός Θέσης

Στην βιβλιογραφία, συναντάται πληθώρα αλγορίθμων SLAM, ενώ τα τελευταία χρόνια, έχουν κυκλοφορήσει και αρκετές υλοποιήσεις ανοικτού κώδικα. Σημαντική εξέλιξη, επίσης, τα τελευταία χρόνια αποτελεί και η εμφάνιση αλγορίθμων 3D SLAM, που πραγματοποιούν χαρτογράφηση και εντοπισμό θέσης στον χώρο, αντί για το επίπεδο, περίπτωση που παρόλα αυτά δε μας απασχόλησε στα πλαίσια της παρούσας διπλωματικής.

Οι περισσότεροι δισδιάστατοι αλγόριθμοι SLAM χρησιμοποιούν, βασικά, ένα σύνολο μετρήσεων από αισθητήρες απόστασης και δεδομένα οδομετρίας με στόχο την εκτίμηση της θέσης του ρομπότ και παραγωγή ενός χάρτη του περιβάλλοντος. Αυτά επιτυγχάνονται, μέσω τεχνικών, όπως τα Φίλτρα Kalman (Kalman Filters), τα Φίλτρα Σωματιδίων (Particle Filters) και η Αντιστοίχιση Σκαναρισμάτων απόστασης (Scan Matching).

Υπάρχουν δύο βασικοί τύποι χάρτη, που παράγονται από αλγορίθμους SLAM. Ο τοπολογικός χάρτης είναι μία αναπαράσταση του περιβάλλοντος που στοχεύει στην αποτύπωση της συνδετικότητας του περιβάλλοντος και όχι στην παραγωγή ενός γεωμετρικά ακέραιου χάρτη. Αντίθετα, οι Χάρτες Πλέγματος Κατάληψης (OGM) χρησιμοποιούν πίνακες με διακριτά κελιά, για μετρική αναπαράσταση του περιβάλλοντος, όπου κάθε κελί εκφράζεται από έναν αριθμό που δηλώνει την πεποιθηση για την κατάσταση κατάληψης του κελιού, δηλαδή κατειλημμένο, ελεύθερο ή άγνωστο.



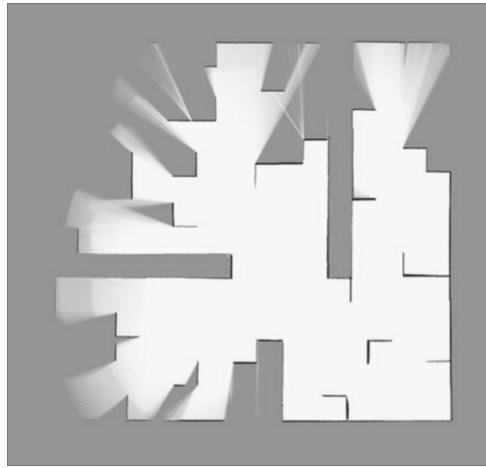
Σχήμα 3.1: Μετρικός 2D χάρτης και αντίστοιχος τοπολογικός χάρτης σε μορφή γράφου [8].

Στην παρούσα διπλωματική εξετάστηκαν δύο αλγόριθμοι 2D SLAM, οι οποίοι χρησιμοποιούν αισθητήρες απόστασης και στην προκειμένη περίπτωση έναν σαρωτή λέιζερ και παράγουν χάρτες πλέγματος κατάληψης (Occupancy Grid Maps). Πρόκειται για τους αλγορίθμους CRSM-SLAM [9] και GMapping [10].

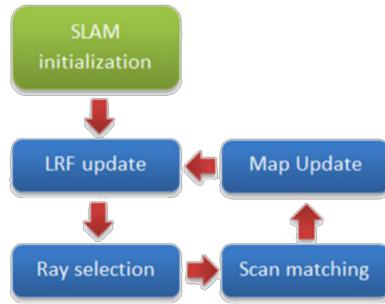
CRSM-SLAM

Ο αλγόριθμος CRSM-SLAM βασίζεται στην τεχνική της αντιστοίχισης σκαναρισμάτων απόστασης, που παρέχει ένας σαρωτής λέιζερ. Με βάση την τεχνική αυτή, η πόζα ενός ρομπότ και ο χάρτης, μπορούν να υπολογιστούν, χρησιμοποιώντας τον δισδιάστατο μετασχηματισμό μεταξύ διαδοχικών σκαναρισμάτων ενός σαρωτή λέιζερ. Ο αλγόριθμος CRSM-SLAM, πραγματοποιεί αντιστοίχιση σκαναρισμάτων, σε σκαναρίσματα που έχουν υποστεί προεπεξεργασία, μέσω μίας μεθόδου επιλογής κρίσιμων, για την επιλογή, ακτίνων (ray selection method). Με αυτόν τον τρόπο, επιτυγχάνεται μείωση της πολυπλοκότητας και του χρόνου που απαιτείται για την διαδικασία αντιστοίχισης. Επίσης, πραγματοποιεί και αντιστοίχιση του τρέχοντος σκαναρίσματος, με τον ολικό χάρτη, που έχει κατασκευασθεί, με στόχο την μείωση των συσσωρευτικών σφαλμάτων.

Ο αλγόριθμος CRSM-SLAM αποτελεί μία αποδοτική και αξιόπιστη υλοποίηση, που μπορεί να χρησιμοποιηθεί σε υπολογιστικά συστήματα, με χαμηλή υπολογιστική ισχύ και απαιτεί μόνο έναν



Σχήμα 3.2: Χάρτης πλέγματος κατάληψης κελιού [9].



Σχήμα 3.3: Διάγραμμα σταδίων του αλγορίθμου CRSM-SLAM [9].

σαρωτή λέιζερ για την παροχή πληροφορίας για το περιβάλλον, για την χαρτογράφηση και τον εντοπισμό θέσης του ρομπότ. Παρόλα αυτά, προϋποθέτει ένα δομημένο περιβάλλον με αρκετά κρίσιμα χαρακτηριστικά (γωνίες, τελειώματα ή ασυνέχειες τοίχων κλπ.), έτσι ώστε να μπορεί να κάνει αξιόπιστη αντιστοίχιση των διαδοχικών σκαναρισμάτων. Επομένως, δεν είναι κατάλληλος για ανοιχτό περιβάλλον, ή μεγάλους ομοιόμορφους διαδρόμους, όπου δεν αρκεί η εμβέλεια του σαρωτή λέιζερ.

GMapping

Ο αλγόριθμος GMapping αποτελεί έναν αλγόριθμο SLAM, που βασίζεται σε Rao-Blackwellized φίλτρα σωματιδίων για την παραγωγή χαρτών πλέγματος κατάληψης. Παράλληλα, χρησιμοποιεί, προσαρμοστικές τεχνικές για την μείωση του αριθμού των σωματιδίων, ενώ λαμβάνει υπόψιν την κίνηση του ρομπότ και τις πιο πρόσφατες μετρήσεις των αισθητήρων απόστασης, για την μείωση της αβεβαιότητας σχετικά με την πόζα του ρομπότ, κατά το στάδιο της πρόβλεψης.

Στην παρούσα διπλωματική, ο αλγόριθμος GMapping, χρησιμοποιήθηκε σε συνδυασμό με σκαναρίσματα από έναν σαρωτή λέιζερ και οδομετρία από ένα εκτεταμένο φίλτρο Kalman, που συνδυάζει πληροφορία οδομετρίας τροχών και πυξίδας. Το δεδομένο σύστημα, θα μπορούσε να επεκταθεί περαιτέρω με προσθήκη οπτικής οδομετρίας από κάμερα, αλλά και πληροφορία από αισθητήρα GPS. Η συμπερίληψη της οδομετρίας, κατά τον εντοπισμό θέσης, σε αντίθεση με τον αλγόριθμο CRSM-SLAM, καταστεί τον αλγόριθμο GMapping κατάλληλο και για εσωτερικούς και εξωτερικούς χώρους, αλλά με το μειονέκτημα της μεγαλύτερης πολυπλοκότητας και ανάγκης για μεγαλύτερη υπολογιστική ισχύ.

3.2 Αυτόνομη Πλοήγηση

Στις προηγούμενες ενότητες ασχολήθηκαμε με την κινηματική ανάλυση, την ρομποτική αντίληψη, την χαρτογράφηση και τον εντοπισμό θέσης, ικανότητες απαραίτητες για την αυτόνομη πλοήγηση σε άγνωστο περιβάλλον, δηλαδή την ικανότητα ενός ρομποτικού οχήματος να μεταβεί από μία αρχική θέση σε μία τελική, αυτόνομα, αποφεύγοντας, ταυτόχρονα εμπόδια και γενικότερα ανεπιθύμητες καταστάσεις.

Η αυτόνομη πλοήγηση ενός ρομποτικού οχήματος σχετίζεται άμεσα με την ευφυΐα ενός ρομπότ, όσον αφορά την λήψη αποφάσεων και την εκτέλεση στόχων που ορίζονται από λειτουργίες υψηλότερου επιπέδου, όπως για παράδειγμα η λειτουργία της επιλογής στόχων (*target selection*). Επομένως, η επάρκεια της ικανότητας αυτόνομης πλοήγησης ενός ρομπότ συνεπάγεται μία εύρωστη ευφυΐα κίνησης, ούτως ώστε με μερική γνώση του περιβάλλοντος και με δεδομένους στόχους να δρα αποδοτικά, αξιόπιστα και με ασφάλεια για την επίτευξη των στόχων αυτών.

Ένας πολύ σημαντικός όρος για τον τομέα της αυτόνομης πλοήγησης αποτελεί ο όρος της *πληρότητας* (*completeness*). Με βάση τον ορισμό της πληρότητας, ένα σύστημα αυτόνομης πλοήγησης είναι πλήρες, εάν μπορεί να βρει λύση στο πρόβλημα επίτευξης ενός στόχου, εάν αυτή υπάρχει. Παρόλα αυτά, συνήθως, στις περισσότερες πρακτικές εφαρμογές, η πληρότητα του συστήματος θυσιάζεται για χάρη της υπολογιστικής πολυπλοκότητας. Επομένως, μας ενδιαφέρει η λύση ενός προβλήματος να μπορεί να βρεθεί σε πραγματικό χρόνο (*real-time*) και να καταναλώνει όσο το δυνατόν λιγότερους υπολογιστικούς πόρους.

Για την αντιμετώπιση του προβλήματος της αυτόνομης πλοήγησης, οι περισσότερες προσεγγίσεις προτείνουν κατανεμημένες αρχιτεκτονικές, που διασπούν το πρόβλημα σε επιμέρους υποπροβλήματα, προς αντιμετώπιση. Με αυτόν τον τρόπο επιτυγχάνεται δραστική μείωση της πολυπλοκότητας, παροχή δυνατότητας πιο αποδοτικής και απλουστευμένης υλοποίησης, ελέγχου και αποσφαλμάτωσης κάθε επιμέρους τμήματος, αλλά και δυνατότητα επαναχρησιμοποίησης τμημάτων σε άλλες προσεγγίσεις.

Η πιο δημοφιλής αρχιτεκτονική αυτόνομης πλοήγησης διασπά το πρόβλημα σε δύο επιμέρους υποπροβλήματα, προβλήματα αντίθετα, αλλά, ταυτόχρονα, συμπληρωματικά [1]. Το πρώτο πρόβλημα προς επίλυση αποτελεί η *κατασκευή μονοπατιού* (*path planning*), η οποία, κατά το πλείστον, χρησιμοποιεί έναν χάρτη του περιβάλλοντος, τη θέση του ρομπότ και τη θέση του τρέχοντα στόχου, με σκοπό την κατασκευή ενός μονοπατιού που επιτρέπει στο ρομπότ την πλοήγηση προς τον στόχο, αποφεύγοντας στατικά εμπόδια. Το δεύτερο πρόβλημα προς επίλυση αποτελεί η *αποφυγή εμποδίων* (*obstacle avoidance*), η οποία χρησιμοποιεί τις πιο πρόσφατες μετρήσεις αισθητήρων ή και ένα τοπικό τμήμα του παραγομένου χάρτη, με στόχο την παραμόρφωση του κατασκευασμένου μονοπατιού για αποφυγή απρόσμενων εμποδίων και καταστάσεων, αλλά και την παραγωγή εφικτών κινηματικά και δυναμικά τροχιών, από το ρομπότ. Επομένως, η κατασκευή μονοπατιού σχετίζεται με μακροπρόθεσμο σχεδιασμό για την επίτευξη του δοσμένου στόχου, ενώ η αποφυγή εμποδίων σχετίζεται με την βραχυπρόθεσμη λήψη αποφάσεων για την ασφαλή πλοήγηση προς τον δοσμένο στόχο, συνδυάζοντας σχεδιασμό και εκτέλεση, βάσει της πιο πρόσφατης αντίληψης για το περιβάλλον.

Στην παρούσα διπλωματική εξετάζονται δύο αρχιτεκτονικές συστήματος αυτόνομης πλοήγησης. Η πρώτη αρχιτεκτονική επεκτείνει την προσέγγιση, που αναφέρθηκε παραπάνω, διασπώντας τη λειτουργία της αποφυγής εμποδίων σε τρία επιμέρους στάδια. Το σύνολο των σταδίων αυτών, αποτελείται από ένα στάδιο παραμόρφωσης μονοπατιού για απομάκρυνση από εμπόδια, στατικά ή δυναμικά, ένα στάδιο μετατροπής του παραμορφωμένου μονοπατιού σε ένα νέο κινηματικά εφικτό μονοπάτι και ένα τελευταίο στάδιο που είναι υπεύθυνο για την διάσχιση του μονοπατιού, μέσω κινηματικά εφικτών εντολών στο χώρο *ταχύτητας-στρέψης* (v, δ_f, δ_r). Αντίθετα, η δεύτερη αρχιτεκτονική που εξετάζεται

χρησιμοποιεί έναν αλγόριθμο κατασκευής κινηματικά εφικτών μονοπατιών με δυναμική ανακατασκευή (dynamic replanning), σε περίπτωση δυναμικών εμποδίων ή γενικά απρόσμενων καταστάσεων, σε συνδυασμό με έναν αλγόριθμο διάσχισης μονοπατιού, αντίστοιχο της πρώτης προσέγγισης.

Η κατασκευή μονοπατιών και η αποφυγή εμποδίων πραγματοποιούνται σε μία αναπαράσταση του περιβάλλοντος, μέσω χαρτών κόστους. Οι χάρτες κόστους (*costmaps*) είναι μία δομή αναπαράστασης του περιβάλλοντος που συντίθεται από ένα σύνολο επιπέδων (*layers*). Στην προκειμένη περίπτωση, χρησιμοποιούνται τρία επίπεδα, το *στατικό επίπεδο static layer* που αναπαριστά τα σχετικά αμετάβλητα τμήματα του χάρτη όπως αυτά παράγονται από έναν αλγόριθμο SLAM, το *επίπεδο εμποδίων obstacle layer* που αποτελεί μία 2D αναπαράσταση των εμποδίων του περιβάλλοντος, όπου τα κελιά των εμποδίων περιγράφονται από ένα *απαγορευτικό κόστος lethal cost* που δηλώνει μη προσπελασμότητα και το *επίπεδο διαστολής inflation layer* που διαδίδει τα απαγορευτικά κόστη ακτινικά με αποσβενόμενο κόστος κατά μία απόσταση που προσδιορίζεται από τον χρήστη κάθετα από τα εμπόδια. Στα κελιά που βρίσκονται σε απόσταση από τα εμπόδια, μικρότερη της ακτίνας εγγεγραμμένου, στο ρομπότ, κύκλου, ανατίθεται ένα υψηλό κόστος, ενώ στα επόμενα το κόστος υπολογίζεται από μία εκθετικά φθίνουσα συνάρτηση, τέτοια ώστε το κόστος μηδενίζεται σε απόσταση ίση με την απόσταση που ορίστηκε από τον χρήστη. Με αυτόν τον τρόπο, το ρομπότ μπορεί να αντιμετωπίζεται σαν σημείο που κινείται στον ελεύθερο χώρο, όπου ως ελεύθερος χώρος ορίζεται το σύνολο των κελιών του χάρτη κόστους με μηδενικό κόστος, ή αρκετά χαμηλό κόστος που προσδιορίζεται από τον χρήστη.

Στην συνέχεια, θα παρουσιαστούν τα αντικείμενα της κατασκευής μονοπατιών, της αποφυγής εμποδίων και της διάσχισης μονοπατιού όπως επίσης και η σχετική βιβλιογραφία, αλλά και οι τελικοί αλγόριθμοι που χρησιμοποιήθηκαν στο σύστημα αυτόνομης πλοήγησης της ρομποτικής πλατφόρμας Monstertruck.

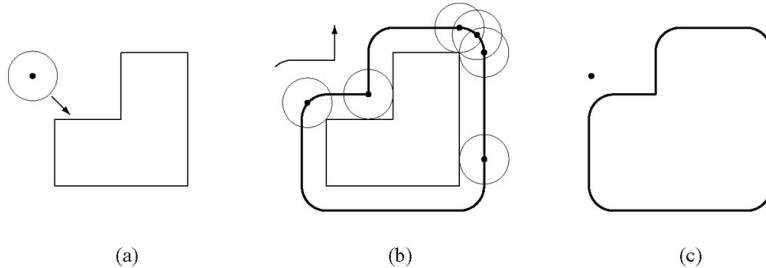
3.2.1 Κατασκευή Μονοπατιού

Το αντικείμενο της κατασκευής μονοπατιών ξεκίνησε με τους βιομηχανικούς ρομποτικούς βραχίονες για την αυτοματοποίηση γραμμών παραγωγής σε εργοστάσια, αλλά από τότε έχει περιλάβει κάθε είδους ρομποτική εφαρμογή, όπως και η κατασκευή μονοπατιών για πλοήγηση ρομπότ στο επίπεδο (2D path planning) ή στο χώρο (3D path planning), πρόβλημα που αποτελεί αρκετά απλούστερο από αυτό των ρομποτικών βραχιόνων, λόγω λιγότερων βαθμών ελευθερίας και άρα λιγότερες διαστάσεις.

Πριν προχωρήσουμε στην κατασκευή μονοπατιών, αυτή καθ' αυτή, θα πρέπει πρώτα να περιγράψουμε την αναπαράσταση του χώρου στον οποίο πραγματοποιείται η εργασία αυτή. Η αναπαράσταση αυτή, ονομάζεται *χώρος καταστάσεων configuration space* C και έχει τόσες διαστάσεις όσες και οι βαθμοί ελευθερίας (DOF) του ρομπότ. Στην προκειμένη περίπτωση, εφόσον έχουμε ένα ρομποτικό όχημα που κινείται στο επίπεδο, μιλάμε για τρεις βαθμούς ελευθερίας (3-DOF), όσον αφορά τις καρτεσιανές συντεταγμένες θέσης (x, y) και τον προσανατολισμό θ του ρομπότ. Ο χώρος αυτός αποτελείται από όλες τις δυνατές καταστάσεις (*configurations*) (x, y, θ) που μπορεί να βρεθεί το ρομπότ στον χώρο. Στις περισσότερες περιπτώσεις ένας χώρος δεν είναι πλήρως ελεύθερος, αλλά περιλαμβάνει εμπόδια. Επομένως, ως ελεύθερος χώρος C_{free} ορίζεται ο χώρος καταστάσεων μείον τον υποχώρο των εμποδίων C_O , δηλαδή $C_{free} = C - C_O$ και αποτελεί το υποσύνολο του χώρου, στο οποίο, το ρομπότ μπορεί να κινηθεί χωρίς να υποστεί συγκρούσεις.

Ο παραπάνω χώρος καταστάσεων C περιορίζεται περαιτέρω, αν αναλογιστεί κανείς, ότι τα περισσότερα ρομπότ χρησιμοποιούν κινηματικά μοντέλα, όπως Differential-Drive, Skid-Steer-Drive, Ackermann ή Four-Wheel Steering, τα οποία παρουσιάζουν μη ολονομικούς περιορισμούς (*nonholonomic constraints*) που περιορίζουν τον χώρο ταχυτήτων $(\dot{x}, \dot{y}, \dot{\theta})$, σε αντίθεση με ολονομικά (*holonomic*) πανκατευθυντικά (*omnidirectional*) ρομπότ. Παρόλα αυτά, η συνήθης προσέγγιση έγκειται στην αγνόηση της μη ολονομικότητας για την κατασκευή μονοπατιών, καθώς τα πιο συνήθη ρομποτικά

κινηματικά μοντέλα Differential-Drive και Skid-Steer-Drive, θεωρούνται ως ψευδο-ολονομικά, καθώς διαθέτουν δυνατότητα επιτόπου στροφής (0-point turn) και μπορούν να ακολουθήσουν ολονομικά μονοπάτια, ενώ ίδια προσέγγιση ακολουθείται και για ρομπότ με Ackermann ή Four-Wheel Steering, αλλά στην συγκεκριμένη περίπτωση, το πρόβλημα της διάσχισης του μονοπατιού λύνεται σε επόμενο στάδιο. Επίσης, στις περισσότερες προσεγγίσεις, για την διευκόλυνση του προβλήματος της κατασκευής μονοπατιών λαμβάνεται η παραδοχή ότι το ρομπότ είναι ένα κινούμενο σημείο στο 2D χώρο (x, y), ενώ αντί για τα πραγματικά εμπόδια, λαμβάνεται μία διασταλμένη εκδοχή αυτών, κατά την ακτίνα ενός κύκλου, στον οποίο εγγράφεται το ίχνος/αποτύπωμα του ρομπότ. Τέλος, όπως είναι φυσικό, έχουν μελετηθεί και αναπτυχθεί και διαφορετικές προσεγγίσεις που λαμβάνουν υπόψιν κινηματικούς και δυναμικούς περιορισμούς στο στάδιο της κατασκευής μονοπατιού, κατάλληλοι για μη ολονομικά οχήματα. Παρακάτω, θα εξεταστούν και οι δύο περιπτώσεις.

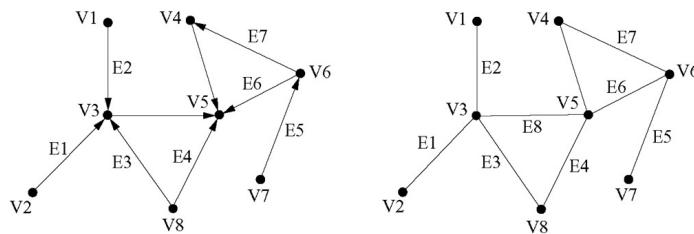


Σχήμα 3.4: Αναπαράσταση ρομπότ ως σημείο και διαστολή εμποδίων, βάσει της ακτίνας του ρομπότ [11].

Όπως αναφέρθηκε παραπάνω, η κατασκευή μονοπατιών λαμβάνει χώρα πάνω σε έναν χάρτη κόστους, ο οποίος παράγεται από τον συνολικό τρέχοντα χάρτη πλέγματος κατάληψης, που έχει παραχθεί από έναν αλγόριθμο SLAM και ονομάζεται ολικός χάρτης κόστους (*global costmap*). Ο ολικός χάρτης κόστους, έπειτα, αντιμετωπίζεται ως γράφος και επομένως, η κατασκευή μονοπατιού μετασχηματίζεται σε αναζήτηση μονοπατιού σε γράφο.

Αναζήτηση Μονοπατιού σε Γράφο

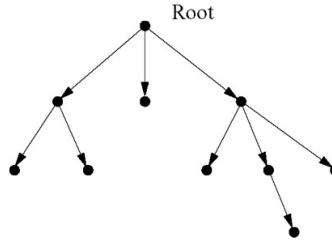
Ως γράφος G ορίζεται μία συλλογή από κόμβους V και ακμές E . Στην κατασκευή μονοπατιών, ένας κόμβος V αναπαριστά μία θέση στον χώρο, ενώ μία ακμή E αναπαριστά την σύνδεση μεταξύ δύο γειτονικών κόμβων, όπου δύο κόμβοι ορίζεται ως γειτονικοί εάν είναι προσπελάσιμοι ο ένας από τον άλλο. Μία ακμή είναι κατευθυνόμενη εάν επιτρέπει την μονόδρομη μετάβαση από έναν κόμβο V_i σε έναν άλλο V_j (E_{ij} ή E_{ji}) ή μη κατευθυνόμενη εάν επιτρέπει αμφίδρομη μετάβαση (E_{ij} και E_{ji}). Επίσης, σε κάθε ακμή E_{ij} ανατίθεται μία τιμή που αναπαριστά το κόστος μετάβασης από τον κόμβο V_i προς έναν γειτονικό κόμβο V_j και ονομάζεται βάρος w_{ij} της ακμής.



Σχήμα 3.5: Κατευθυνόμενος (αριστερά) και μη κατευθυνόμενος (δεξιά) γράφος [11].

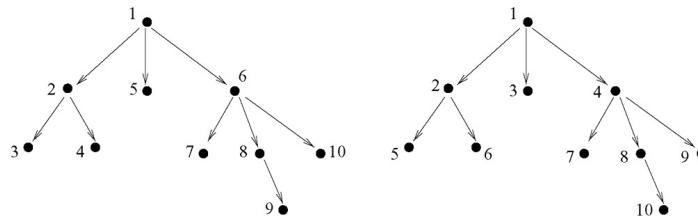
Ως μονοπάτι στον γράφο ορίζεται μία ακολουθία κόμβων V_i , τέτοια ώστε, για κάθε V_i, V_{i+1} υπάρχει μία ακμή $E_{i,i+1}$ που ενώνει τους κόμβους V_i, V_{i+1} . Αν υπάρχει ακμή E_{ij} για κάθε συνδυασμό κόμβων V_i, V_j , τότε ο γράφος είναι συδεδεμένος.

Τα δέντρα αποτελούν μία υποπερίπτωση των γράφων και ορίζονται ως συνδεδεμένοι, καυτευθυντικοί άκυκλοι γράφοι, όπου ένας γράφος θεωρείται άκυκλος εάν δεν περιέχει κυκλικά μονοπάτια, δηλαδή μονοπάτια που αποτελούνται από μία ακολουθία n κόμβων, όπου ο πρώτος κόμβος V_1 ταυτίζεται με τον τελευταίο V_n . Τα δέντρα έχουν έναν αρχικό κόμβο, που ονομάζεται ρίζα και συνδέεται μόνο με εξερχόμενες ακμές (outgoing) και όχι με εισερχόμενες (incoming). Στα δέντρα, επίσης χρησιμοποιείται μία τοπολογία πατέρα - παιδιού, όπου κάθε κόμβος έχει πατέρα (εκτός τη ρίζα) και μπορεί να έχει από μηδέν ή περισσότερα παιδιά. Εάν ένας κόμβος δεν περιλαμβάνει παιδιά, αυτό υποδηλώνει ότι είναι τερματικός κόμβος και ονομάζεται φύλλο (leaf) του δέντρου.



Σχήμα 3.6: Δέντρο: συνδεδεμένος, καυτευθυνόμενος και άκυκλος γράφος [11].

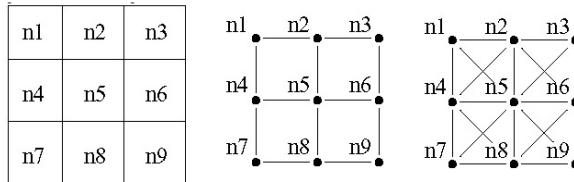
Η αναζήτηση μονοπατιού από έναν αρχικό κόμβο προς ένα κόμβο - στόχο σε ένα δέντρο, πραγματοποιείται με μεθόδους αναζήτησης μονοπατιού. Υπάρχουν δύο βασικές προσεγγίσεις που διακρίνονται, η **αναζήτηση κατά βάθος** (depth-first search) και η **αναζήτηση κατά πλάτος** (breadth-first search). Η αναζήτηση κατά βάθος ξεκινά από την ρίζα του δέντρου, επιλέγει ένα παιδί της ρίζας, έπειτα ένα παιδί του παιδιού της ρίζας και ούτω καθεξής, μέχρις ότου να βρεθεί σε φύλλο ή στον στόχο. Εάν βρεθεί σε φύλλο, τότε ανεβαίνει ένα επίπεδο πάνω και προχωράει στο επόμενο παιδί. Η διαδικασία αυτή επαναλαμβάνεται, μέχρις ότου βρεθεί ο στόχος ή καλυφθούν όλοι οι κόμβοι του δέντρου. Αντίθετα, η αναζήτηση κατά πλάτος, λειτουργεί με την πεποίθηση ότι ο στόχος βρίσκεται κοντά στον αρχικό κόμβο - ρίζα. Έτσι, λοιπόν, ξεκινάει από την ρίζα, επισκέπτεται όλα τα παιδιά της ρίζας και έπειτα όλα τα εγγόνια (παιδιά των παιδιών) της ρίζας και ούτω καθεξής, μέχρις ότου επισκεφθεί τον στόχο, ή μέχρι να καλύψει όλους τους κόμβους του δέντρου.



Σχήμα 3.7: Μέθοδοι αναζήτησης κατά βάθος (αριστερά) και κατά πλάτος (δεξιά) σε δέντρο [11].

Σε ρομποτικές εφαρμογές, συνήθως χρησιμοποιείται ένα πλέγμα (grid), για την αναπαράσταση του χώρου. Η δομή του πλέγματος, μπορεί να θεωρηθεί ως γράφος, εάν λάβουμε ως κόμβους του γράφου, τα pixels του πλέγματος και ως ακμές τις ενώσεις γειτονικών κελιών. Η γειτονικότητα των κελιών, συνήθως ορίζεται με βάση τον ορισμό της συνδετικότητας του γράφου, όπου επιλέγεται συνήθως συνδετικότητα τεσσάρων σημείων ή οκτώ σημείων, όπως παρουσιάζεται στο σχήμα 3.8.

Οι γράφοι πλέγματος, όπως είναι προφανές, δεν αποτελούν δέντρα. Παρόλα αυτά, με κάποιες παραδοχές, γίνεται, δυνατή η χρήση των προαναφερθέντων μεθόδων αναζήτησης κατά βάθος και κατά πλάτος. Συγκεκριμένα, η αναζήτηση κατά βάθος ξεκινάει από τον αρχικό κόμβο, επιλέγει ένα παιδί - κόμβο και μετέπειτα παιδιά - κόμβους με αυξανόμενη απόσταση από τον αρχικό κόμβο, μέχρις ότου βρεθεί σε κόμβο χωρίς παιδιά ή σε κόμβο που έχει ήδη επισκεφθεί. Αντίστοιχα, η αναζήτηση κατά



Σχήμα 3.8: Πλέγμα και γράφοι συνδετικότητας τεσσάρων και οκτώ σημείων (από αριστερά προς τα δεξιά) [11].

πλάτος επισκέπτεται πρώτα τα κόμβους που ισαπέχουν από τον αρχικό κόμβο και έπειτα μεταβαίνει σε επόμενα επίπεδα κόμβων. Τέλος, υπάρχει και η περίπτωση των άπληστων μεθόδων αναζήτησης γράφων (greedy-search methods) που επεκτείνονται σε κόμβους που βρίσκονται κοντινότερα στον κόμβο, που έχει οριστεί ως στόχος.

Αλγόριθμοι Dijkstra και A^*

Οι πιο γνωστοί και δημοφιλείς αλγόριθμοι αναζήτησης μονοπατιού σε γράφο είναι οι αλγόριθμοι Dijkstra και A^* . Ο αλγόριθμος Dijkstra αναπτύχθηκε από τον Edsger W. Dijkstra, το 1956, με στόχο την αναζήτηση του συντομότερου μονοπατιού διάσχισης μεταξύ δύο κόμβων σε έναν γράφο, βάσει αναζήτησης κατά πλάτος. Ο αλγόριθμος A^* αποτελεί μία γενίκευση του αλγορίθμου Dijkstra, που αναπτύχθηκε το 1968 από τους Peter Hart, Nils Nilsson και Bertram Raphael και βασίζεται στην χρήση ευρετικών (heuristics) μεθόδων για την κατεύθυνση του αναζήτησης, με στόχο την βελτιστοποίηση της όλης διαδικασίας αναζήτησης, όσον αφορά την ελαχιστοποίηση του αριθμού των επισκεπτόμενων κόμβων, την ελαχιστοποίηση του χρόνου εκτέλεσης κλπ.

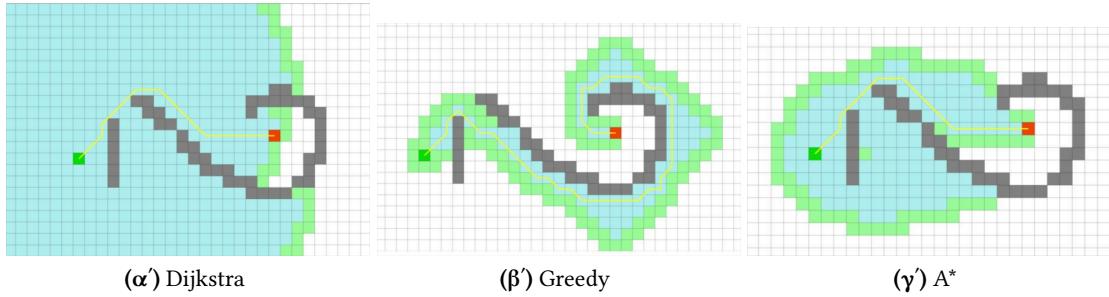
Το μειονέκτημα των μεθόδων αναζήτησης κατά βάθος και κατά πλάτος, έγκειται στο γεγονός ότι δεν χρησιμοποιούν πληροφορία σχετικά με την θέση του κόμβου, που έχει οριστεί ως στόχος, για να κατευθύνουν την αναζήτηση. Το πρόβλημα αυτό λύνεται, από τον αλγόριθμο A^* ο οποίος απαιτεί τον ορισμό μίας ευρετικής μεθόδου για την εκτίμηση του κόστους για την μετάβαση από έναν κόμβο στον δεδομένο στόχο. Η ευρετική αυτή μέθοδος χρησιμοποιείται κατά την επιλογή του νέου κόμβου, βάσει της απόστασης του από τον στόχο. Ο νέος στόχος, επομένως, πρέπει να εκφράζει την υψηλότερη πιθανότητα προσέγγισης του στόχου, με βάση τοπική πληροφορία. Η μέθοδος αυτή, παρόλα αυτά, δεν προσφέρει καμία εγγύηση για την σύγκλιση στο ολικώς βέλτιστο μονοπάτι.

Η απόδοση του αλγορίθμου A^* βασίζεται στην επιλογή μίας καλής ευρετικής μεθόδου και επομένως, καθορίζει την πιθανότητα εύρεσης του βέλτιστου μονοπατιού. Μία ευρετική μέθοδος, θεωρείται καλή εάν είναι αισιόδοξη, δηλαδή εάν επιστρέφει κόστος, μικρότερο ή ίσο με το μήκος του βέλτιστου μονοπατιού για την μετάβαση από τον τρέχον κόμβο στον στόχο. Για παράδειγμα, μία αισιόδοξη ευρετική μέθοδος, που χρησιμοποιείται συχνά, είναι η ευκλείδια απόσταση ή απόσταση Manhattan μεταξύ του τρέχοντος κόμβου και του στόχου.

Αν ορίσουμε ως $h(n)$ την ευρετική συνάρτηση και $g(n)$ την συνάρτηση κόστους μετάβασης από τον αρχικό στον τρέχοντα κόμβο, τότε η συνάρτηση $f(n)$ που δηλώνει την εκτίμηση του κόστους μετάβασης από τον αρχικό κόμβο στον τελικό στόχο, μέσω του κόμβου n ορίζεται ως

$$f(n) = g(n) + h(n) \quad (3.7)$$

Εδώ αξίζει να αναφερθεί, ότι αν λάβουμε $h(n) = 0$, η σχέση (3.7) περιγράφει το κόστος επιλογής νέου κόμβου του αλγορίθμου Dijkstra, ενώ αν λάβουμε $g(n) = 0$, τότε παίρνουμε την αντίστοιχη επιλογή νέου κόμβου του άπληστου αλγορίθμου αναζήτησης.



Σχήμα 3.9: Σύγκριση αλγορίθμων αναζήτησης μονοπατιού σε γράφο πλέγματος με Dijkstra, Greedy και A^* [12].

Οι αλγόριθμοι Dijkstra και A^* που εφαρμόζονται στον χώρο καταστάσεων, όπως παρουσιάστηκαν παραπάνω και με βάση τις παραδοχές που αναφέρθηκαν για την γεωμετρία και κινητικότητα του ρομπότ, αγνοούν πλήρως κινηματικούς και δυναμικούς περιορισμούς, που μπορεί να παρουσιάζει αυτό. Επίσης, αποτελούν μεθόδους ακατάλληλες για δυναμικά περιβάλλοντα, όπου ένα μονοπάτι, μπορεί να καταστεί μη προσπελάσιμο, με βάση τις πιο σύγχρονες πληροφορίες σχετικά με το περιβάλλον.

Αλγόριθμος SBPL Lattice Planner

Οι Maxim Likhachev και Dave Ferguson [13] προτείνουν έναν αλγόριθμο κατασκευής δυναμικά εφικτών μονοπατιών, μεγάλου μήκους για αυτόνομα οχήματα που κινούνται με υψηλές ταχύτητες (25kph), ο οποίος χρησιμοποιήθηκε στον διαγωνισμό DARPA Urban Challenge, με πολύ θετικά αποτελέσματα. Ο αλγόριθμος, αυτός, βασίζεται σε μία χρονικά άμεση και σταδιακή αναζήτηση (*anytime & incremental search*) σε ένα πολλαπλής-ανάλυσης δικτύωμα δυναμικά εφικτών καταστάσεων (*multi-resolution dynamically feasible lattice state space*).

Το δικτύωμα καταστάσεων (*state lattice*) ορίζεται ως μία διακριτοποίηση του χώρου καταστάσεων σε ένα σετ καταστάσεων και τις αντίστοιχες συνδέσεις μεταξύ αυτών, όπου μία σύνδεση μεταξύ δύο καταστάσεων, δηλώνει ένα εφικτό μονοπάτι. Επομένως, το πρόβλημα κατασκευής μονοπατιών μετασχηματίζεται και πάλι σε αναζήτηση σε γράφο. Σε αντίθεση με γράφους με συνδετικότητα τεσσάρων ή οκτώ σημείων, ο ορισμός της συνδετικότητας καταστάσεων στα δικτυωμάτα καταστάσεων εγγυάται ότι όποια λύση βρεθεί θα είναι κινηματικά και δυναμικά εφικτή, γεγονός που καθιστά τον εν λόγω αλγόριθμο κατάλληλο για μη ολονομικά οχήματα.

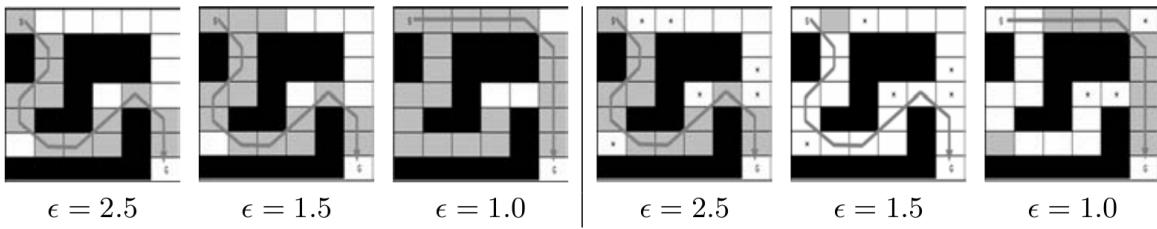
Στην προκειμένη περίπτωση για την αναπαράσταση μίας κατάστασης χρησιμοποιείται το διάνυσμα τεσσάρων μεταβλητών $s = (x, y, \theta, v)$, όπου (x, y) είναι οι καρτεσιανές συντεταγμένες, θ ο προσανατολισμός και v η ταχύτητα για την δεδομένη κατάσταση. Οι συντεταγμένες και ο προσανατολισμός είναι βασικοί για τον έλεγχο της εγκυρότητας μίας πόζας, δηλαδή κατά πόσο είναι εφικτή, ενώ η ταχύτητα που μπορεί να πάρει μόνο μέγιστη κατά απόλυτο τιμή, χρησιμοποιείται για να λαμβάνεται υπόψιν ο χρόνος που απαιτείται για την αντιστροφή της ταχύτητας, σε περιπτώσεις ελιγμών.

Για την κατασκευή των δικτυωμάτων καταστάσεων λαμβάνονται υπόψιν δύο βασικά σημεία. Πρώτον η διακριτοποίηση / δειγματοληψία του χώρου καταστάσεων για την αναπαράσταση των καταστάσεων στο δικτύωμα και δεύτερον η κατασκευή ενός χώρου κινήσεων (*action space / control set*) που ορίζει τις δυναμικά εφικτές συνδέσεις μεταξύ δύο καταστάσεων. Ο χώρος κινήσεων θα πρέπει να είναι αρκετά πλήρης, ώστε να καθιστά δυνατή την κατασκευή όλων των δυνατών μονοπατιών στο δικτύωμα καταστάσεων, μέσω συνδυασμού ακολουθιών των βασικών δυνατών κινήσεων (*motion primitives*).

Για την μείωση του κόστους ελέγχου όλων των συνδυασμών των δυνατών κινήσεων για την κατασκευή ενός μονοπατιού μεταξύ δύο καταστάσεων, ο αλγόριθμος χρησιμοποιεί μία προσέγγιση πολλαπλής ανάλυσης, με βάση την οποία χρησιμοποιείται ένας χώρος ελέγχου υψηλής ανάλυσης για

περιοχές κοντά στην αρχική και τελική κατάσταση και ένας χώρος ελέγχου χαμηλής ανάλυσης για το ενδιάμεσο τμήμα.

Η αναζήτηση στο δικτύωμα καταστάσεων πραγματοποιείται με βάση τις δυνατές κινήσεις και τα κόστη αυτών. Η πιο δημοφιλής μέθοδος αναζήτησης είναι ο αλγόριθμος A^* , ο οποίος είναι αρκετά αποδοτικός και στοχεύει στην εύρεση του βέλτιστου μονοπατιού μεταξύ δύο καταστάσεων. Παρόλα αυτά λόγω του μεγέθους του προβλήματος που καλείται να λύσει, μπορεί να παραβιάζει χρονικούς περιορισμούς και άρα καθίσταται ανεπαρκής. Μία λύση στο πρόβλημα αυτό προσφέρουν παραλλαγές του αλγόριθμου A^* που βασίζονται στην προσέγγιση της χρονικής αμεσότητας (Anytime A^* variants), όπως ο αλγόριθμος ARA^* (Anytime Repairing A^*). Οι αλγόριθμοι, που ανήκουν σ' αυτήν την κατηγορία στοχεύουν στην εύρεση μίας αρχικής άκρως μη βέλτιστης λύσης, αλλά σε πολύ μικρό χρόνο, ενώ έπειτα, ασχολούνται με την συνεχή βελτίωση της λύσης αυτής, όσο επιτρέπεται από τους χρονικούς περιορισμούς.



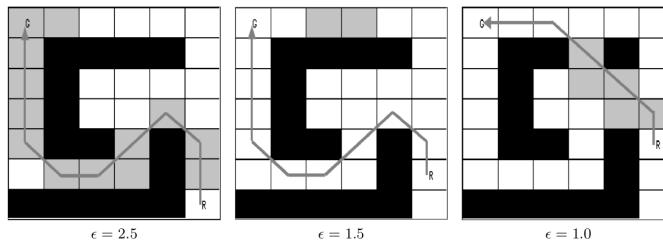
Σχήμα 3.10: Αναζήτηση σε γράφο πλέγματος με A^* (αριστερά) και ARA^* (δεξιά) για φθίνοντα παράγοντα διαστολής ϵ [14].

Ο αλγόριθμος A^* και οι παραλλαγές του δουλεύουν αποτελεσματικότερα, όταν το περιβάλλον εργασίας είναι γνωστό από πριν και δεν μεταβάλλεται, κάτι αρκετά απίθανο στις περισσότερες ρομποτικές εφαρμογές. Ένα ρομπότ, δέχεται συνεχώς δεδομένα από τους αισθητήρες του μπορεί να μεταβάλλουν την αντίληψη του για το περιβάλλον και άρα μία λύση να καθίσταται μη εφικτή. Επίσης, τα περισσότερα περιβάλλοντα είναι δυναμικά, δηλαδή περιλαμβάνουν κινούμενα αντικείμενα, όπως άλλα ρομπότ, οχήματα, ανθρώπους κα., στα οποία θα πρέπει να μπορεί να αντιδράσει το ρομπότ για να αποφύγει πιθανές συγκρούσεις. Το πρόβλημα αυτό θα μπορούσε να λυθεί χρησιμοποιώντας συνεχής ανανέωση του μονοπατιού, λύση που αποτελεί ιδιαίτερα κοστοβόρα και μπορεί να μην είναι χρονικά εφικτή. Επομένως, απαιτείται μία λύση που να στοχεύει στην επιδιόρθωση τμημάτων της υπάρχουσας λύσης και όχι στην αντικατάσταση της από νέα, όπως για παράδειγμα ο αλγόριθμος D^* και οι παραλλαγές του.

Ο αλγόριθμος αναζήτησης που προτείνουν οι Likhachev και Ferguson, είναι ο αλγόριθμος AD^* (Anytime Dynamic A^*). Ο αλγόριθμος AD^* συνδυάζει τα θετικά και των δύο παραπάνω κατηγοριών, παραλλαγών $Anytime A^*$ και D^* αλγορίθμων. Συγκεκριμένα, πραγματοποιεί μη βέλτιστες αναζητήσεις A^* , μέσω διαστολής του κόστους λύσης κατά έναν μειούμενο παράγοντα $\epsilon > 1$, χρησιμοποιώντας παράλληλα πληροφορία από την προηγούμενη αναζήτηση. Επίσης, δανείζεται ιδέες από τον αλγόριθμο D^* και τις παραλλαγές του, βάσει των οποίων, χρησιμοποιεί καινούργιες πληροφορίες, μόνο για τμήματα του χώρου αναζήτησης, που αφορούν την τρέχουσα κατάσταση και αναζήτηση. Αυτό επιτυγχάνεται με ανανέωση του μονοπατιού, μέσω αναζήτησης από την τελική κατάσταση (στόχος) προς τα πίσω μέχρι την τρέχουσα κατάσταση του ρομπότ.

Η απόδοση του αλγορίθμου AD^* , όπως και του αλγορίθμου A^* , βασίζεται απόλυτα στην επιλογή μίας αποδοτικής ευρετικής συνάρτησης $h(n)$ για την κατεύθυνση της αναζήτησης, η οποία μπορεί να μειώσει της απαιτήσεις της αναζήτησης σε χρόνο και μνήμη κατά τουλάχιστον μία τάξη μεγέθους. Μία γενική αισιόδοξη ευρετική συνάρτηση που μπορεί να χρησιμοποιηθεί αποτελεί η βέλτιστη λύση

σε περιβάλλον χωρίς εμπόδια, που μπορεί να υπολογιστεί *offline* και να αποθηκευθεί σε έναν ευρετικό πίνακα για ένα δεδομένο περιβάλλον. Παρόλα αυτά, η λύση αυτή δεν είναι πρακτική για διάφορα περιβάλλοντα. Αντίθετα, προτιμότερη είναι η *online* λύση ενός απλοποιημένου προβλήματος αναζήτησης και η χρήση του αποτελέσματος για την κατεύθυνση του πιο σύνθετου προβλήματος αναζήτησης. Μία τέτοια λύση, αποτελεί η χρήση του αλγορίθμου αναζήτησης Dijkstra για την εύρεση του κόστους μονοπατιού από την κατάσταση του ρομπότ σε κάθε άλλη εφικτή κατάσταση στο περιβάλλον. Η ευρετική συνάρτηση Dijkstra μπορεί, παρόλα αυτά, να υπερεκτιμήσει το κόστος του μονοπατιού, αφού εκτιμάει την μετακίνηση του κέντρου του ρομπότ μόνο και άρα δεν είναι πάντα αισιόδοξη, όπως απαιτείται από τον αλγόριθμο. Για την επίλυση του προβλήματος αυτού, το κόστος κάθε κελιού στο δισδιάστατο πλέγμα (grid) που χρησιμοποιείται για την εξαγωγή του ευρετικού κόστους, ορίζεται ως ο μέσος όρος των κοστών των κελιών που βρίσκονται μέσα σε ένα κύκλο, με κέντρο το κελί που εξετάζεται και ακτίνα, την ακτίνα του κύκλου στον οποίο εγγράφεται το αποτύπωμα (footprint) του ρομπότ. Επομένως, το κόστος της μετάβασης $c(s, s')$ από μία κατάσταση s σε μία κατάσταση s' προκύπτει ως το μήκος του μονοπατιού μετάβασης επί του μέγιστου εκ των (α) μέσος όρος κόστους κελιών κατά την μετάβαση $c(s, s')$ και (β) μέγιστο κόστος κελιών, "τιμωρώντας" έτσι μεταβάσεις υψηλού κόστους (κοντά σε εμπόδια). Κάθε μια από τις δύο ευρετικές συνάρτησεις που αναφέρθηκαν, έχουν πλεονεκτήματα, ανά περίπτωση και γενικά λειτουργούν συμπληρωματικά. Επομένως, επιλέχθηκε ένας συνδυασμός τους, μέσω μίας νέας ευριστικής συνάρτησης $h(n) = \max(h_{fsh}(s), h_{2D}(s))$, όπου $h_{fsh}(s)$ είναι η ευρετική συνάρτηση στον ελεύθερο από εμπόδια χώρο, ενώ η $h_{2D}(s)$ είναι η ευρετική συνάρτηση με βάση τα εμπόδια στο περιβάλλον.



Σχήμα 3.11: Αναζήτηση σε γράφο πλέγματος με AD^* για φθίνοντα παράγοντα διαστολής ϵ [15].

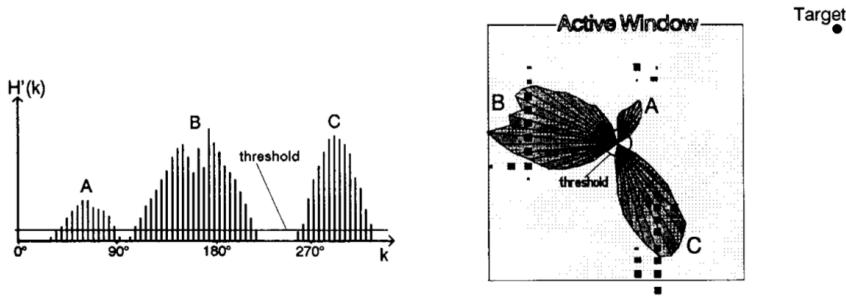
3.2.2 Αποφυγή Εμποδίων

Η λειτουργία ενός αλγορίθμου δυναμικής αποφυγής εμποδίων είναι να χρησιμοποιεί τις πιο πρόσφατες μετρήσεις των αισθητήρων του ρομπότ για την ανανέωση της πεποίθησης του, για την θέση του, τη θέση του τρέχοντα στόχου και την τοπολογία του περιβάλλοντος, έτσι ώστε να μπορεί να προσαρμόζεται σε απρόσμενες καταστάσεις, αλλά ταυτόχρονα να μην αποκλίνει σημαντικά από τον δεδομένο στόχο.

Η αυτόνομη αποφυγή εμποδίων για ρομπότ, αποτελεί ένα εξαιρετικά μελετημένο αντικείμενο, με μεγάλο ερευνητικό ενδιαφέρον και συνεχή παραγωγή νέων μεθόδων και αλγορίθμων. Ένας από τους πιο απλοϊκούς αλγορίθμους αποφυγής εμποδίων είναι ο αλγόριθμος Bug, όπως επίσης και η εξέλιξη του Bug 2, οι οποίοι βασίζονται στην ακολούθηση του συνόρου των εμποδίων καθώς το ρομπότ κινείται προς τον δοσμένο στόχο. Όπως είναι προφανές, η συγκεκριμένη μέθοδος, δεν είναι πρακτική για πιο απαιτητικές εφαρμογές και επομένως έχουν αναπτυχθεί πολλές πιο ευφυείς μέθοδοι, κάποιες από τις οποίες εξετάστηκαν για την παρούσα υλοποίηση και παρουσιάζονται συνοπτικά στη συνέχεια.

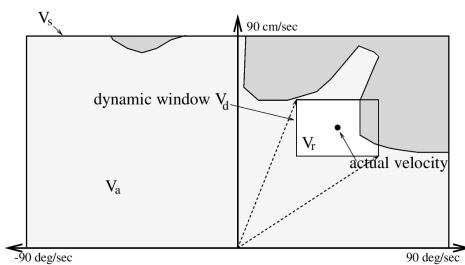
- Ο αλγόριθμος VFH (Vector Field Histogram) που αναπτύχθηκε από τους Borenstein and Koren [16], βασίζεται στην παραγωγή ενός πολικού ιστογράμματος με βάση την πιθανότητα ύπαρξης εμποδίων ανά κατεύθυνση γύρω από το ρομπότ, μέσα σε ένα τοπικό τμήμα του περιβάλλοντος (ενεργό παράθυρο). Το ιστόγραμμα που παράγεται περιλαμβάνει "βουνά" και "κοιλάδες",

που δηλώνουν την ύπαρξη εμποδίων και ελεύθερου χώρου, αντίστοιχα. Τελικά, επιλέγει την κατεύθυνση που παρουσιάζει την μικρότερη υπάρξης εμποδίων, χωρίς παράλληλα να αποκλίνει σημαντικά από τον στόχο. Αργότερα, οι Ulrich and Borenstein επέκτειναν τον αλγόριθμο VFH στις εκδοχές VFH+[17] και VFH*[18], για βελτίωση της ομαλότητας των μονοπατιών και εκμετάλλευση πληροφορίας από μονοπάτια, που παράγονται μέσω του αλγορίθμου A*, βάσει της ολικής διαθέσιμης πληροφορίας για το περιβάλλον.



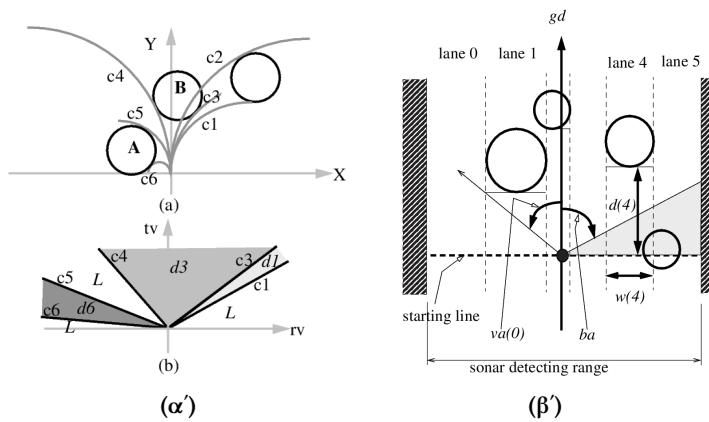
Σχήμα 3.12: Πολικό Ιστόγραμμα Πυκνότητας Εμποδίων βάσει του εμποδίων που βρίσκονται μέσα στο ενεργό παράθυρο. [16].

- Ο αλγόριθμος DWA (Dynamic Window Approach) που αναπτύχθηκε από τους Fox et al. [19] και βασίζεται στην κατασκευή ενός χώρου ταχυτήτων (velocity space). Ο χώρος ταχυτήτων ορίζει τις κινηματικά και δυναμικά εφικτές ταχύτητες (v, ω), με τις οποίες μπορεί να κινηθεί το ρομπότ, λαμβάνοντας υπόψιν τα εμπόδια του περιβάλλοντος. Από τον χώρο ταχυτήτων, επιλέγει σε κάθε χρονικό βήμα, ένα δυναμικό παράθυρο (Dynamic Window), το οποίο ορίζει τις εφικτές ταχύτητες που μπορεί να φτάσει μέσα σε ένα χρονικό βήμα, με βάση την γραμμική και γωνιακή επιτάχυνση και επιβράδυνση. Τελικά, η επιλογή του ζεύγους ταχυτήτων (v, ω), πραγματοποιείται βάσει μίας αντικειμενικής συνάρτησης, που εφαρμόζεται σε όλα τα ζεύγη (v, ω) του δυναμικού παραθύρου. Η αντικειμενική αυτή συνάρτηση, γενικά, προτιμάει τροχιές που μειώνουν το σφάλμα προσανατολισμού του ρομπότ ως προς το στόχο, περιγράφονται από υψηλή ταχύτητα και κρατούν ασφαλή απόσταση από τα εμπόδια του περιβάλλοντος. Ο αλγόριθμος DWA, έχει μελετηθεί και επεκταθεί από αρκετούς ερευνητές, με την εξαιρετικά σημαντική περίπτωση του Global DWA, που αναπτύχθηκε από τους Brock and Khatib [20]. Ο αλγόριθμος Global DWA χρησιμοποιεί ολική πληροφορία (global information) για το περιβάλλον και όχι μόνο τοπική, αποφεύγοντας έτσι το πρόβλημα τοπικών ελαχίστων (local minima), που μπορεί να οδηγεί το ρομπότ σε αδιέξοδες καταστάσεις. Ο αλγόριθμος DWA και ιδιαίτερα η Global εκδοχή του αποτελούν μία εξαιρετικά δημοφιλής επιλογή για ρομποτικές εφαρμογές, λόγω του γεγονότος ότι λαμβάνει υπόψιν κινηματικούς και δυναμικούς περιορισμούς, χρησιμοποιεί ολική πληροφόρηση και μπορεί να κινείται, αποφεύγοντας εμπόδια σε υψηλές ταχύτητες.



Σχήμα 3.13: Ενδεικτικός χώρος ταχυτήτων και δυναμικό παράθυρο του αλγορίθμου DWA [19]. Η γκρίζα περιοχή δηλώνει τις περιοχές που προκαλούν σύγκρουση με εμπόδια.

- Ο αλγόριθμος CVM (Curvature Velocity Method) αναπτύχθηκε από τον Simmons [21]. Ο αλγόριθμος λαμβάνει υπόψιν κινηματικούς και δυναμικούς περιορισμούς, όπως επίσης και περιβαλλοντικούς περιορισμούς (εμπόδια), μέσω ενός χώρου ταχυτήτων (v, ω), αντίστοιχα, όπως και ο αλγόριθμος DWA. Ο αλγόριθμος CVM, όμως, αναπαριστά τα εμπόδια ως κύκλους ακτίνας r και ορίζει μη εφικτές τροχιές $c = \omega/v$, βάσει της ανισότητας $c_{min} < c < c_{max}$, όπου c_{min}, c_{max} είναι οι εφαπτομενικές τροχιές στον κύκλο που αναπαριστά το εμπόδιο. Ο αλγόριθμος CVM, είναι επιρρεπείς στο πρόβλημα τοπικών ελαχίστων, καθώς λαμβάνει υπόψιν μόνο τοπική πληροφορία του χώρου. Μία βελτίωση του αλγορίθμου CVM, αποτελεί ο αλγόριθμος LCM (Lane Curvature Method), που αναπτύχθηκε από τους Ko and Simmons [22]. Σε αντίθεση με τον αλγόριθμο CVM, ο αλγόριθμος LCM χωρίζει τον χώρο σε λωρίδες (lanes) και επιλέγει την λωρίδα με το μεγαλύτερο μήκος και πλάτος, όσον αφορά τα εμπόδια. Αντίστοιχα, με τον αλγόριθμο DWA, οι αλγόριθμοι CVM Και LCM χρησιμοποιούν αντικειμενικές συναρτήσεις για την εύρεση της καλύτερης δυνατής τροχιάς.



Σχήμα 3.14: (α) Καθορισμός ελεύθερων τροχιών, με βάση τον αλγόριθμο CVM [21] και (β) Χωρισμός του χώρου σε λωρίδες για μετάβαση στην πιο ελεύθερη λωρίδα, με βάση τον αλγόριθμο LCM [22].

Οι περισσότεροι αλγόριθμοι αποφυγής εμποδίων, όπως οι παραπάνω, παρότι, έχουν σημαντικά πλεονεκτήματα και αποδοτική συμπεριφορά σε πραγματικές συνθήκες, σχεδιάστηκαν από τους εμπνευστές τους για ολονομικά ή ψευδο-ολονομικά ρομπότ, που μπορούν να ακολουθήσουν μη ομαλές τροχιές ή τροχιές μηδενικής ακτίνας - επί τόπου στροφές (0-point turns). Όμως, τα μη ολονομικά ρομπότ που χρησιμοποιούν κινηματικά μοντέλα Ackermann ή Τετραδιεύθυνσης, παρουσιάζουν κάτωφραγμένη ακτίνα τροχιάς ($|R| > R_{min}$), το οποίο καθιστά τις παραπάνω μεθόδους ανεπαρκή για την περίπτωση που εξετάζεται.

Για την λύση του προβλήματος της αποφυγής εμποδίων για την ρομποτική πλατφόρμα Monstertuck και γενικότερα για ρομπότ με κινηματικό Ackermann ή Τετραδιεύθυνσης, υλοποιήθηκε ένας αλγόριθμος, βάσει του αλγορίθμου Bubble Band που ανέπτυξαν οι Khatib et al. [23]. Παρόλα αυτά πριν παρουσιαστεί ο αλγόριθμος Bubble Band και η προσεγγιστική υλοποίηση του, θα παρουσιάσουμε πρώτα τον αλγόριθμο της ελαστικής ζώνης (*elastic band*) των Quinlan and Khatib [24], όπως επίσης και τα μονοπάτια Reeds-Shepp των J.J. Reeds και L.A. Shepp [25], που είναι απαραίτητες για την κατανόηση του αλγορίθμου Bubble Band.

Ο αλγόριθμος της Ελαστικής Ζώνης

Ο αλγόριθμος της ελαστικής ζώνης αναπτύχθηκε από τους Quinlan and Khatib [24] με στόχο την πλήρωση του χάσματος μεταξύ της κατασκευής μονοπατιού και του ελέγχου του ρομπότ, με έναν ομαλό και ασφαλή τρόπο. Η ιδέα πίσω από τον αλγόριθμο της ελαστικής ζώνης έχει να κάνει με

την ελαστική παραμόρφωση ενός δεδομένου μονοπατιού, βάσει τεχνητών δυνάμεων, με στόχο την αποφυγή εμποδίων και την εξομάλυνση του μονοπατιού.

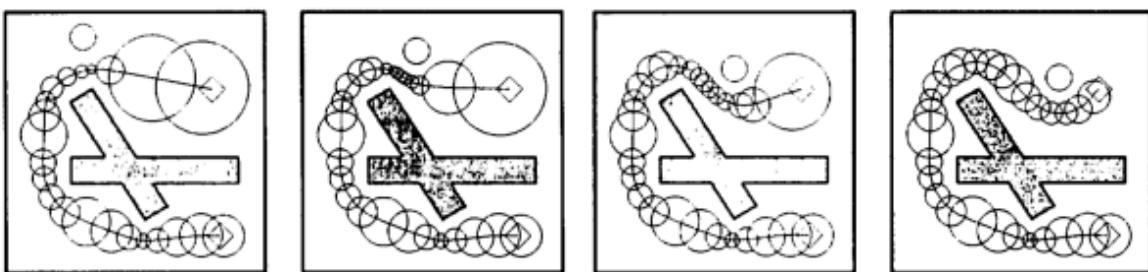
Ο αλγόριθμος της ελαστικής ζώνης αποσκοπεί στην μίμηση της συμπεριφοράς ενός λάστιχου που παραμορφώνεται, εξαιτίας ενός συνόλου δυνάμεων που ασκούνται σ' αυτό. Οι δυνάμεις αυτές είναι γενικά δύο ειδών, εσωτερικές ελκτικές δυνάμεις συστολής και εξωτερικές απωστικές δυνάμεις, που παραμορφώνουν το λάστιχο, μέχρι να φτάσει σε ένα επίπεδο ηρεμίας, δηλαδή μηδενικών συνισταμένων δυνάμεων. Στην περίπτωση, του αλγορίθμου της ελαστικής ζώνης, ορίζονται τεχνητές απωστικές δυνάμεις που τείνουν να απομακρύνουν την ελαστική ζώνη από εμπόδια και ελκτικές δυνάμεις που τείνουν να μειώσουν την διαστολή - τέντωμα της.

Το κλειδί για την υλοποίηση του αλγορίθμου της ελαστικής ζώνης, αποτελεί η φούσκα (bubble). Η φούσκα ορίζεται ως το μέγιστο τοπικό, προσβάσιμο τμήμα του ελεύθερου χώρου, γύρω από μία κατάσταση b . Αν $\rho(b)$ είναι η ελάχιστη ευκλείδια απόσταση του ρομπότ στην κατάσταση b , από το κοντινότερο εμπόδιο, τότε είναι προφανές ότι το ρομπότ μπορεί να κινηθεί για απόσταση $\rho(b)$ προς οποιαδήποτε κατεύθυνση σε μια νέα κατάσταση q , χωρίς να έλθει σε σύγκρουση με κάποιο εμπόδιο. Το μέγεθος της φούσκας είναι ανάλογο της απόστασης $\rho(b)$ του ρομπότ σε μία κατάσταση b από το κοντινότερο εμπόδιο, ενώ ουσιαστικά δηλώνει το σύνολο των δυνατών καταστάσεων q στις οποίες μπορεί να διέλθει το ρομπότ από μία κατάσταση b . Επομένως, η φούσκα μπορεί να οριστεί μαθηματικά ως

$$B(b) = \{q : \|b - q\| < \rho(b)\} \quad (3.8)$$

Η ελαστική ζώνη κατασκευάζεται, αρχικά βάσει του ολικού μονοπατιού που παράγεται από έναν αλγόριθμο κατασκευής μονοπατιών (πχ. A^*), τοποθετώντας μία φούσκα σε κάθε ενδιάμεσο σημείο του μονοπατιού. Επομένως, η ελαστική ζώνη αναπαρίσταται από ένα πεπερασμένο σύνολο από φούσκες. Για να είναι τώρα δυνατή η κατασκευή ενός ασφαλούς μονοπατιού, θα πρέπει οι διαδοχικές φούσκες της ελαστικής ζώνης να επικαλύπτονται κατά ένα ποσοστό, δηλαδή όσο το μονοπάτι βρίσκεται εντός των φουσκών, αυτό είναι ασφαλές.

Η παραμόρφωση του μονοπατιού της ελαστικής ζώνης, πραγματοποιείται ασκώντας τις τεχνητές δυνάμεις που αναφέρθηκαν παραπάνω σε κάθε μία από τις φούσκες που αποτελούν την ελαστική ζώνη, σειριακά πάνω κάτω και με έναν πεπερασμένο αριθμό επαναλήψεων. Η άσκηση των τεχνητών δυνάμεων στις φούσκες έχει ως αποτέλεσμα, αυτές να μετακινούνται. Η μετακίνηση αυτή, παρόλα αυτά, έχει ως αποτέλεσμα διαδοχικές φούσκες να μην επικαλύπτονται επαρκώς ή να υπερκαλύπτονται. Επομένως, για να διατηρηθεί η συνθήκη της μερικής επικάλυψης, θα πρέπει να δημιουργούνται καινούργιες φούσκες για την περίπτωση της μη επαρκούς επικάλυψης και να διαγράφονται φούσκες που υπερκαλύπτονται. Με αυτόν τον τρόπο βελτιώνεται η απόδοση του αλγορίθμου.



Σχήμα 3.15: Παραμόρφωση ελαστικής ζώνης, βάσει ελκτικών δυνάμεων μεταξύ των φουσκών και απωστικών δυνάμεων που ασκούνται από το στατικό και το δυναμικό εμπόδιο [24].

Το μέγεθος και η κατεύθυνση της μετακίνηση μίας φούσκας καθορίζεται από τις τεχνητές ελκτικές δυνάμεις f_c και απωστικές δυνάμεις f_r που ασκούνται σε αυτή και υπολογίζονται από τις από τις ακόλουθες σχέσεις.

$$\mathbf{f}_c = k_c \cdot \left(\frac{\mathbf{b}_{i-1} - \mathbf{b}_i}{\|\mathbf{b}_{i-1} - \mathbf{b}_i\|} + \frac{\mathbf{b}_{i+1} - \mathbf{b}_i}{\|\mathbf{b}_{i+1} - \mathbf{b}_i\|} \right) \quad (3.9)$$

$$\mathbf{f}_r = \begin{cases} k_r \cdot \left(\rho_0 - \rho \frac{\partial \rho}{\partial \mathbf{b}} \right) & \rho < \rho_0 \\ 0 & \rho \geq \rho_0 \end{cases} \quad (3.10)$$

όπου

k_c : κέρδος ελκτικής δύναμης

k_r : κέρδος απωστικής δύναμης

και

$$\frac{\partial \rho}{\partial \mathbf{b}} = \frac{1}{2h} \begin{bmatrix} \rho(b - h_x) - \rho(b + h_x) \\ \rho(b - h_y) - \rho(b + h_y) \end{bmatrix} \quad (3.11)$$

Έπειτα, η μετατόπιση κάθε φούσκας μπορεί να υπολογιστεί χρησιμοποιώντας την προσεγγιστική σχέση

$$\mathbf{b}_{new} = \mathbf{b}_{old} + \alpha \cdot \mathbf{f}_{total} \quad (3.12)$$

όπου α είναι ένα κέρδος που μπορεί να οριστεί ως $\rho(\mathbf{b}_{old})$, βάσει της οποίας η μετατόπιση της φούσκας είναι αναλογική του μεγέθους της. Ουσιαστικά, πρόκειται, δηλαδή, για μία μέθοδο μέγιστης καθόδου για την εύρεση του σημείου ισορροπίας των δυνάμεων.

Τα μονοπάτια Reeds-Shepp

Το 1990 οι J.A. Reeds και L. A. Shepp εκδώσαν μία επιστημονική [25] στην οποία περιγράφουν τα βέλτιστα μονοπάτια για ένα αυτοκίνητο που κινείται και εμπρός και πίσω. Συγκεκριμένα ορίζουν ένα σύνολο μονοπατιών που είναι επαρκές για κάθε συνδυασμό καταστάσεων, βέλτιστο από άποψη μήκους και μικρό σε πλήθος.

Το πρόβλημα που εξετάζεται είναι η κατασκευή βέλτιστου μονοπατιού για ένα αυτοκίνητο που κινείται και μπρος και πίσω με μοναδιαία ταχύτητα και παρουσιάζει άνω φραγμένη γωνία στρέψης των τροχών και άρα κάτω φραγμένη ακτίνα τροχιάς. Το εν λόγω σύστημα περιγράφεται από τις ακόλουθες σχέσεις.

$$\dot{x} = u_1 \cdot \cos \theta \quad (3.13)$$

$$\dot{y} = u_1 \cdot \sin \theta \quad (3.14)$$

$$\dot{\theta} = u_1 \cdot u_2 \quad (3.15)$$

όπου

u_1 : η ταχύτητα με τιμές $\{-1, 1\}$

u_2 : η καμπυλότητα της τροχιάς με τιμές στο διάστημα $[-\tan \phi_{max}, \tan \phi_{max}]$

ϕ : η γωνία στρέψης των μπροστινών τροχών του αυτοκινήτου με τιμές στο διάστημα $[-\phi_{max}, \phi_{max}]$

ϕ_{max} : μέγιστη γωνία στρέψης τροχών με τιμή στο διάστημα $(0, \frac{\pi}{2})$

Ο στόχος, λοιπόν, είναι η ελαχιστοποίηση του μήκους L μονοπατιού μετάβασης από μία κατάσταση q_{start} σε μία κατάσταση q_{goal} και συγκεκριμένα η ελαχιστοποίηση της σχέσης

$$L(\tilde{q}, \tilde{u}) = \int_0^{t_F} \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} dt \quad (3.16)$$

Οι J.A. Reeds και L.A. Shepp απέδειξαν ότι τα βέλτιστα μονοπάτια για την εξεταζόμενη περίπτωση είναι το πολύ σαράντα οχτώ και αναπαρίστανται από ένα αντίστοιχο σύνολο λέξεων. Οι εν λόγω λέξεις συντίθενται από ένα σύνολο συμβόλων βασικών κινήσεων. Οι βασικές κινήσεις (*motion primitives*) περιγράφονται από τα σύμβολα S, L, R , όπου το σύμβολο S δηλώνει ευθύγραμμη κίνηση, το σύμβολο L δηλώνει αριστερή στροφή, ενώ το σύμβολο R δεξιά στροφή. Επίσης, χρησιμοποιείται το σύμβολο "!" για να δηλώσει την αντιστροφή της φοράς κίνησης, αλλά και άνω δείκτης $\{+, -\}$ (πχ. R^+) που δηλώνουν την φορά κίνησης και κάτω δείκτες (πχ. $S_{1.5}, C_{\frac{\pi}{2}}$) που δηλώνουν το μέτρο του κάθε επιμέρους τμήματος - βασικής κίνησης ενός μονοπατιού. Αναπαριστώντας, τώρα τις βασικές κινήσεις στροφής R, L με το σύμβολο C , το βέλτιστο μονοπάτι μπορεί να εκφραστεί βάσει των ακόλουθων βασικών λέξεων, για το πολύ πέντε βασικές κινήσεις.

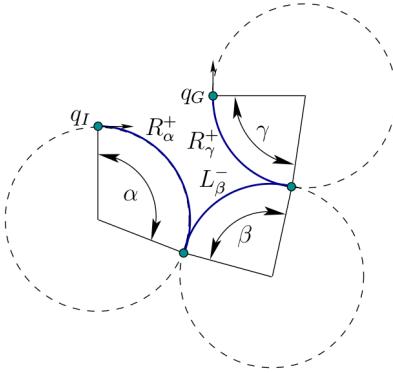
$$\begin{aligned} & \{C|C|C, CC|C, C|CC, CSC, CC_\beta|C_\beta C, C|C_\beta C_\beta|C, \\ & C|C_{\frac{\pi}{2}}SC, CSC_{\frac{\pi}{2}}|C, C|C_{\frac{\pi}{2}}SC_{\frac{\pi}{2}}|C\} \end{aligned} \quad (3.17)$$

Χρησιμοποιώντας τις βασικές λέξεις (3.17) για κάθε δυνατή περίπτωση, προκύπτουν οι σαράντα οχτώ διαφορετικές λέξεις, που αναπαριστούν τα μονοπάτια Reeds-Shepp, όπως φαίνεται στον πίνακα 3.1.

Πίνακας 3.1: Τα 48 είδη μονοπατιών Reeds-Shepp.

Βασική Λέξη	Δυνατές Ακολουθίες Βασικών Κινήσεων
$C C C$	$(L^+R^-L^+), (L^-R^+L^-), (R^+L^-R^+), (R^-L^+R^-)$
$CC C$	$(L^+R^+L^-), (L^-R^-L^+), (R^+L^+R^-), (R^-L^-R^+)$
$C CC$	$(L^+R^-L^-), (L^-R^+L^+), (R^+L^-R^-), (R^-L^+R^+)$
CSC	$(L^+S^+L^+), (L^-S^-L^-), (R^+S^+R^+), (R^-S^-R^-),$ $(L^+S^+R^+), (L^-S^-R^-), (R^+S^+L^+), (R^-S^-L^-)$
$CC_\beta C_\beta C$	$(L^+R_\beta^+L_\beta^-R^-), (L^-R_\beta^+L_\beta^+R^+), (R^+L_\beta^+R_\beta^-L^-), (R^-L_\beta^-R_\beta^+L^+)$
$C C_\beta C_\beta C$	$(L^+R_\beta^-L_\beta^-R^+), (L^-R_\beta^+L_\beta^+R^-), (R^++L_\beta^-R_\beta^-L^+), (R^-L_\beta^+R_\beta^+L^-)$
$C C_{\frac{\pi}{2}}SC$	$(L^+R_{\frac{\pi}{2}}^-S^-R^-), (L^-R_{\frac{\pi}{2}}^+S^+R^+), (R^+L_{\frac{\pi}{2}}^-S^-L^-), (R^-L_{\frac{\pi}{2}}^+S^+L^+),$ $(L^+R_{\frac{\pi}{2}}^-S^-L^-), (L^-R_{\frac{\pi}{2}}^+S^+L^+), (R^+L_{\frac{\pi}{2}}^-S^-R^-), (R^-L_{\frac{\pi}{2}}^+S^+R^+)$
$CSC_{\frac{\pi}{2}} C$	$(L^+S^+L_{\frac{\pi}{2}}^+R^-), (L^-S^-L_{\frac{\pi}{2}}^-R^+), (R^+S^+R_{\frac{\pi}{2}}^+L^-), (R^-S^-R_{\frac{\pi}{2}}^-L^+),$ $(R^+S^+L_{\frac{\pi}{2}}^+R^-), (R^-S^-L_{\frac{\pi}{2}}^-R^+), (L^+S^+R_{\frac{\pi}{2}}^+L^-), (L^-S^-R_{\frac{\pi}{2}}^-L^+)$
$C C_{\frac{\pi}{2}}SC_{\frac{\pi}{2}} C$	$(L^+R_{\frac{\pi}{2}}^-S^-L_{\frac{\pi}{2}}^-R^+), (L^-R_{\frac{\pi}{2}}^+S^+L_{\frac{\pi}{2}}^+R^-), (R^+L_{\frac{\pi}{2}}^-S^-R_{\frac{\pi}{2}}^-L^+),$ $(R^-L_{\frac{\pi}{2}}^+S^+R_{\frac{\pi}{2}}^+L^-)$

Τέλος, οι H.J. Sussmann και G. Tang [27] απέδειξαν ότι οι ακολουθίες $(L^-R^+L^-), (R^-L^+R^-)$ είναι πλεονάζουσας και μπορούν να παραληφθούν και επομένως προκύπτουν τελικά σαράντα έξι δυνατές ακολουθίες, που μπορούν να περιγράψουν τα βέλτιστα μονοπάτια για ένα αυτοκίνητο όχημα που μπορεί να κινείται και μπρος και πίσω με κάτω φραγμένη ακτίνα τροχιάς.



Σχήμα 3.16: Παράδειγμα μονοπατιού Reeds-Shepp $R_\alpha^+ L_\beta^- R_\gamma^+$ [26].

Ο Αλγόριθμος Bubble Band

Οι M. Khatib, H. Jaouni, R. Chatila και J.P. Laumond [23] ανέπτυξαν τον αλγόριθμο Bubble Band με στόχο την δυναμική παραμόρφωση των μονοπατιών μη ολονομικών ρομπότ, ούτως ώστε να προσαρμόζονται σε μεταβολές του περιβάλλοντος, ενώ παράλληλα να υπακούν σε δυναμικούς περιορισμούς του ρομπότ, χωρίς να θυσιάζεται η επίτευξη του τρέχοντα στόχου.

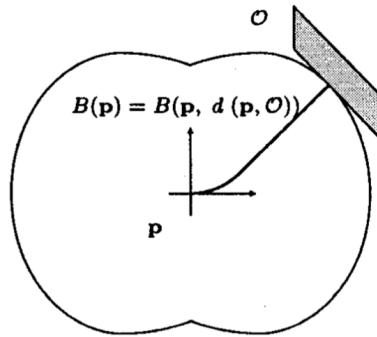
Ο αλγόριθμος Bubble Band, ορίζει την μη ολονομική φούσκα (*nonholonomic bubble*), για ένα μη ολονομικό ρομπότ, ως το μέγιστο τοπικά προσβάσιμο τμήμα του χώρου, γύρω από μία κατάσταση, λαμβάνοντας υπόψιν τυχόν εμπόδια, όπως επίσης και τους μη ολονομικούς περιορισμούς του ρομπότ. Ο αλγόριθμος Bubble Band, χρησιμοποιεί ένα μονοπάτι, που έχει κατασκευασθεί από έναν αλγόριθμο που δεν λαμβάνει υπόψιν μη ολονομικούς περιορισμούς, όπως ο A^* και παραμορφώνει το μονοπάτι αυτό, βάσει της μεθόδου της ελαστικής ζώνης, των S. Quinlan και O. Khatib [24], που αναφέρθηκε παραπάνω.

Σε αντίθεση με την φούσκα που χρησιμοποιεί ο αλγόριθμος της ελαστικής ζώνης, η μη ολονομική φούσκα παράγεται βάσει των αποστάσεων του ρομπότ σε μία κατάσταση q από τα εμπόδια, αλλά και βάσει του κινηματικού και δυναμικού μοντέλου του ρομπότ. Αυτό το επιτυγχάνει χρησιμοποιώντας μία διαφορετική μετρική απόστασης κατά την κατασκευή της φούσκας. Συγκεκριμένα, ενώ ο αλγόριθμος της ελαστικής ζώνης χρησιμοποιεί την ευκλείδια απόσταση για να βρει το κοντινότερο εμπόδιο, ο αλγόριθμος Bubble Band χρησιμοποιεί μία μετρική, που ονομάζεται μη ολονομική απόσταση(*nonholonomic distance*). Η μη ολονομική απόσταση ορίζεται από τους J.P. Laumond και P. Soueres [28] στον χώρο καταστάσεων $R^2 \times S^1$ ως το κοντινότερο μονοπάτι μεταξύ δύο καταστάσεων. Επομένως, μία μη ολονομική φούσκα $B(p)$ κατασκευάζεται βάσει της μικρότερης μη ολονομικής απόστασης μεταξύ της κατάστασης p που οδηγεί σε σύγκρουση.

Με βάση τα παραπάνω η μη ολονομική φούσκα στην κατάσταση p , με ακτίνα r δηλώνει το σύνολο των εφικτών καταστάσεων q στις οποίες μπορεί να μεταβεί το ρομπότ από την κατάσταση p , με την μη ολονομική απόσταση $d(p, q)$ μεταξύ των καταστάσεων p και q να είναι κατά μέτρο μικρότερη από r . Η μαθηματική αναπαράσταση της μη ολονομικής φούσκας προκύπτει ως

$$B(p, r) = \{q \in R^2 \times S^1 \mid d(p, q) < r\} \quad (3.18)$$

Αντίστοιχα με την συνθήκη μερικής επικάλυψης του αλγορίθμου ελαστικής ζώνης που αναφέρθηκε στην ενότητα 3.2.2, ορίζεται και οι συνθήκες επικάλυψης του αλγορίθμου Bubble Band ως η απαίτηση δύο διαδοχικές φούσκες να επικαλύπτονται, αλλά να μην υπερκαλύπτονται. Αυτό σημαίνει ότι για τρεις διαδοχικές φούσκες $B(p_{i-1}, r_{i-1})$, $B(p_i, r_i)$, $B(p_{i+1}, r_{i+1})$ θα πρέπει να ισχύουν οι σχέσεις



Σχήμα 3.17: Μη ολονομική φούσκα [23].

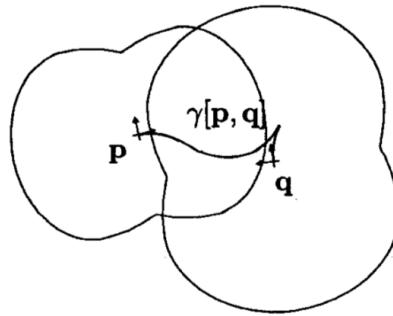
$$d(p_i, p_{i+1}) \leq r_i + r_{i+1} - \epsilon_c \quad (3.19)$$

$$d(p_{i-1}, p_{i+1}) \geq r_{i-1} + r_{i+1} - \epsilon_o \quad (3.20)$$

όπου ϵ_c, ϵ_o θετικές μικρές σταθερές.

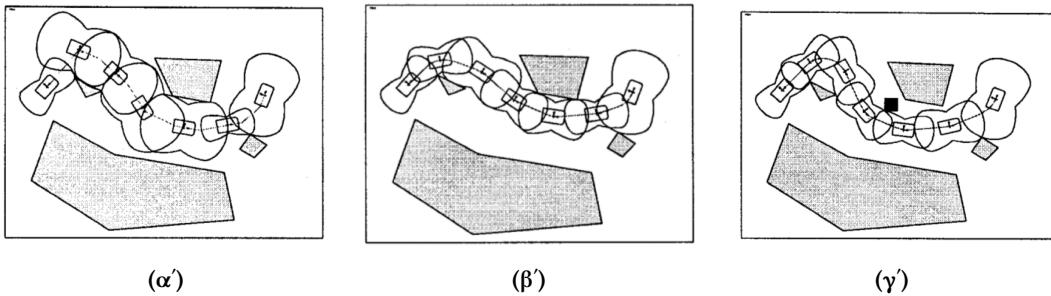
Σε περίπτωση που δεν ικανοποιείται η σχέση (3.19) για δύο διαδοχικές φούσκες, τότε θα πρέπει να δημιουργείται μία ενδιάμεση φούσκα, ενώ σε περίπτωση που δεν ισχύει η σχέση 3.20 τότε θα πρέπει να διαγράφεται η ενδιάμεση φούσκα. Επίσης, για να αποφεύγεται η συνεχής δημιουργία και διαγραφή φουσκών θα πρέπει να επιλέγονται οι σταθερές ϵ_c, ϵ_o τέτοιες ώστε $\epsilon_c < \epsilon_o$.

Σαν αποτέλεσμα, ένα βέλτιστο μονοπάτι Reeds-Shepp μεταξύ των κέντρων δύο διαδοχικών φουσκών $B(p_i, r_i), B(p_{i+1}, r_{i+1})$ θα περιέχεται εξολοκλήρου μέσα στην ένωση τους $B(p_i, r_i) \cup B(p_{i+1}, r_{i+1})$, γεγονός που καθιστά το μονοπάτι ασφαλές, όπως φαίνεται και στο σχήμα 3.18. Επομένως, μπορεί να κατασκευασθεί ένα ασφαλές μονοπάτι, μέσω της εύρεσης επιμέρους βέλτιστων μονοπατιών Reeds-Shepp μεταξύ των κέντρων των φουσκών.



Σχήμα 3.18: Βέλτιστο ασφαλές μονοπάτι μεταξύ των κέντρων δυο μη ολονομικών φουσκών [23].

Ο αλγόριθμος Bubble Band ξεκινάει με ένα ολικό μονοπάτι και το δειγματοληπτεί, λαμβάνοντας ένα σύνολο σημείων κατά μήκους του μονοπατιού. Έπειτα δημιουργεί μία μη ολονομική φούσκα για κάθε ένα από τα επιμέρους σημεία και ξεκινάει την διαδικασία παραμόρφωσης μέσω της άσκησης τεχνητών δυνάμεων, ενώ παράλληλα ελέγχει την συνδετικότητα και αλληλοκάλυψη των φουσκών και ανάλογα με την περίπτωση δημιουργεί ή διαγράφει φούσκες. Όπως και με τον αλγόριθμο της ελαστικής ζώνης οι τεχνητές δυνάμεις που ασκούνται στις φούσκες, χωρίζονται σε εσωτερικές - ελκτικές



Σχήμα 3.19: (α') Δημιουργία ζώνης φουσκών, (β') παραμόρφωση και (γ') αντίδραση σε δυναμικό εμπόδιο [23].

και εξωτερικές - απωστικές δυνάμεις. Οι εσωτερικές δυνάμεις αναπαριστούν την αλληλεπίδραση μεταξύ διαδοχικών φουσκών και τείνουν να τοποθετούν μία φούσκα πάνω στο κοντινότερο μονοπάτι που ενώνει τις διαδοχικές της φούσκες ενώ οι εξωτερικές δυνάμεις δηλώνουν την αλληλεπίδραση μίας φουσκας με τα εμπόδια και τείνουν να απομακρύνουν την φούσκα από τα εμπόδια. Με βάση τα παραπάνω προκύπτει ένα δυναμικό πεδίο από την συνισταμένη δράση των δυναμικών πεδίων που δημιουργούν τις εσωτερικές και εξωτερικές δυνάμεις. Στην συνέχεια παρουσιάζονται οι σχέσεις των δυναμικών πεδίων των εσωτερικών και εξωτερικών δυνάμεις.

$$P_f(p_i) = \frac{K_f}{2} (d(p_i, p_{i+1}) - (r_i + r_{i+1}) + \epsilon_c) \times (d(p_i, p_{i+1}) - (r_i + r_{i+1}) + \epsilon_o) \quad (3.21)$$

$$P_b(p_i) = \frac{K_b}{2} (d(p_i, p_{i-1}) - (r_i + r_{i-1}) + \epsilon_c) \times (d(p_i, p_{i-1}) - (r_i + r_{i-1}) + \epsilon_o) \quad (3.22)$$

$$P_c(p_i) = \frac{K_c}{2} (d^\gamma(p_i))^2 \quad (3.23)$$

όπου

P_f : το δυναμικό πεδίο που μεταξύ των φουσκών $B(p_i), B(p_{i+1})$

P_b : το δυναμικό πεδίο που μεταξύ των φουσκών $B(p_{i-1}), B(p_i)$

P_c : το δυναμικό πεδίο που ασκείται που τείνει να συστάλλει το μονοπάτι

d^γ : η μικρότερη μη ολονομική απόσταση μεταξύ των φουσκών (p_{i-1}) και (p_{i+1})

K_f : κέρδος δύναμης από την επόμενη διαδοχικά φούσκα $B(p_{i+1})$

K_b : κέρδος δύναμης από την προηγούμενη διαδοχικά φούσκα $B(p_{i-1})$

K_c : κέρδος δύναμης συστολής

Αντίστοιχα, τα δυναμικά πεδία των εξωτερικών δυνάμεων από τα εμπόδια [29] υπολογίζονται ως

$$P_r(p_i) = \begin{cases} \frac{K_r}{2} (d_{cs}^O(p_i) - d_c)^2 + \frac{K_\infty}{2} \left(\frac{1}{d_{cs}^O(p_i)} - \frac{1}{d_c} \right)^2 & , \quad d_{cs}^O(p_i) \leq d_c \\ 0 & , \quad \text{αλλιώς} \end{cases} \quad (3.24)$$

όπου

P_r : δυναμικό πεδίο απωστικών δυνάμεων από εμπόδια

K_r, K_∞ : κέρδη απωστικών δυνάμεων

d_{cs}^O : κοντινότερο μονοπάτι Reeds-Shepp από εμπόδια

d_c : άνω κατώφλι απόστασης από τα εμπόδια για περιορισμό του μεγέθους της φούσκας

Τέλος, η εξαγωγή των εσωτερικών και εξωτερικών δυνάμεων μπορεί να γίνει με την εφαρμογή του τελεστή $-\nabla$, στις αντίστοιχες εξισώσεις δυναμικού.

Ο Αλγόριθμος Reeds-Shepp Band

Ο αλγόριθμος Reeds-Shepp Band που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, όπως προαναφέρθηκε, βασίστηκε στον αλγόριθμο Bubble Band, που αναλύθηκε παραπάνω και αποτελεί μία απλοποιημένη προσέγγιση αυτού.

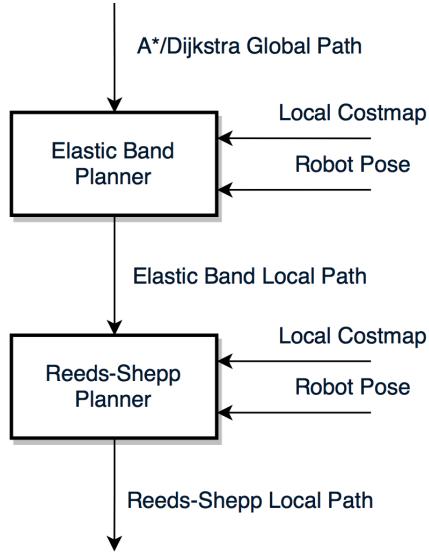
Λόγω της αυξημένης πολυπλοκότητας υπολογισμού της μη ολονομικής απόστασης, σε αντίθεση με την ευκλείδια και συνεπώς και του υπολογισμού των επιμέρους δυνάμεων που παραμορφώνουν το δοσμένο ολικό μονοπάτι, επιλέχθηκε, τελικά, η χρησιμοποίηση του απλούστερου και λιγότερο απαιτητικού αλγορίθμου της ελαστικής ζώνης. Ο αλγόριθμος της ελαστικής ζώνης, όπως ορίστηκε παραμορφώνει δυναμικά το δοσμένο ολικό μονοπάτι, αλλά δεν λαμβάνει υπόψιν του μη ολονομικούς περιορισμούς.

Για την μετατροπή της ελαστικής ζώνης σε ένα δυναμικά και κινηματικά εφικτό μονοπάτι λαμβάνονται το σύνολο των κέντρων των φουσκών της ελαστικής ζώνης και για κάθε δύο διαδοχικά κέντρα παράγεται ένα μονοπάτι Reeds-Shepp. Σε αντίθεση με τον αλγόριθμο Bubble Band, παρόλα αυτά, ένα μονοπάτι Reeds-Shepp που ενώνει τα κέντρα δύο φουσκών της ελαστικής ζώνης, μπορεί να μην περιλαμβάνεται πλήρως μέσα στις φουσκές, με αποτέλεσμα να υπάρχει κίνδυνος σύγκρουσης με εμπόδια, εάν δεν υπάρχει πρόσθετος έλεγχος σύγκρουσης. Επομένως, για την λύση του προβλήματος αυτού πραγματοποιείται έλεγχος σύγκρουσης για κάθε κατάσταση ενός υποψήφιου μονοπατιού Reeds-Shepp, το οποίο γίνεται δεκτό ή απορρίπτεται ανάλογα με το αποτέλεσμα.

Επίσης, με στόχο την μείωση του υπολογιστικού φόρτου, η δημιουργία της ελαστικής ζώνης πραγματοποιείται για ένα τοπικό τμήμα του ολικού μονοπατιού, αλλά παράλληλα ανανεώνεται δυναμικά όσο κινείται το ρομπότ για αποφυγή δυναμικών απρόσμενων καταστάσεων. Επίσης, πραγματοποιείται περαιτέρω μείωση του υπολογιστικού φόρτου, κατά το στάδιο της μετατροπής της ελαστικής ζώνης σε μονοπάτια Reeds-Shepp, επιλέγοντας την κατασκευή μονάχα ενός μονοπατιού Reeds-Shepp μεταξύ του κέντρου της φουσκας που αντιστοιχεί στην τρέχουσα κατάσταση του ρομπότ και του κέντρου της επόμενης διαδοχικά φουσκας στην ελαστική ζώνη, ή σε επόμενη, εάν η απόσταση μεταξύ των δύο κέντρων είναι μικρότερη από ένα κατώφλι.

3.2.3 Διάσχιση Μονοπατιού

Έχοντας κατασκευάσει ένα μονοπάτι για να ακολουθήσει το ρομπότ, με στόχο την μετάβαση από μία αρχική σε μία τελική κατάσταση, το πρόβλημα που απομένει είναι ο ορισμός ενός νόμου ελέγχου που θα παράγει τις κατάλληλες ταχύτητες και θα επιτρέψει στο ρομπότ να κινηθεί με τον αναμενόμενο τρόπο. Πολλές προσεγγίσεις αλγορίθμων αποφυγής εμποδίων, όπως ο αλγόριθμος DWA [19] που αναφέρθηκε στην ενότητα 3.2.2 παράγουν τροχιές στον χώρο των ταχυτήτων, για την ασφαλή ακολούθηση ολικού μονοπατιού, με αποτέλεσμα να λύνονται τρία προβλήματα ταυτόχρονα. Στην προκειμένη περίπτωση, παρόλα αυτά, χρησιμοποιείται η κατασκευή ενός τοπικού, ασφαλούς και κινηματικά δυναμικά εφικτού μονοπατιού στο επίπεδο και επομένως, θα πρέπει να ορισθεί ένας



Σχήμα 3.20: Το διάγραμμα σταδίων παραμόρφωσης ολικού μονοπατιού μέσω αλγορίθμου ελαστικής ζώνης και μονοπατών Reeds-Shepp.

ξεχωριστός νόμος ελέγχου που θα παράγει τις εντολές ελέγχου, δηλαδή τις ταχύτητες με τις οποίες θα πρέπει να κινηθεί το ρομπότ για να ακολουθήσει αποτελεσματικά το εν λόγω μονοπάτι.

Οι αλγόριθμοι ή μέθοδοι διάσχισης μονοπατιού συνήθως χωρίζονται σε δύο κατηγορίες, ανοικτού και κλειστού βρόχου. Οι αλγόριθμοι διάσχισης μονοπατιού, ανοικτού βρόχου, συνήθως, λύνουν το πρόβλημα, μέσω του, εκ των προτέρου διαχωρισμού ενός μονοπατιού σε μία ακολουθία ευθύγραμμων τμημάτων και καμπυλών και τον υπολογισμό των εντολών ταχύτητας για την ακολούθηση της ακολουθίας των τμημάτων. Σαν αποτέλεσμα, δεν λαμβάνουν υπόψιν απρόσμενες καταστάσεις, όπως ολίσθηση, αλλά και ατέλειες κινηματικού και δυναμικού μοντέλου του ρομπότ και επομένως μπορεί να αποκλίνουν σημαντικά από την επιθυμητή συμπεριφορά. Αντίθετα, μία μέθοδος διάσχισης μονοπατιού, κλειστού βρόχου, λειτουργούν δυναμικά, μετασχηματίζοντας το πρόβλημα της διάσχισης μονοπατιού, στο πρόβλημα της επιλογής ενός υπό-στόχου και την παραγωγή εντολών ελέγχου για την ακολούθηση του στόχου αυτού, λαμβάνοντας υπόψιν την τρέχουσα κατάσταση του ρομπότ και την απόκλιση από την επιθυμητή συμπεριφορά, δηλαδή, για παράδειγμα τα σφάλματα θέσης και προσανατολισμού από το τρέχοντα υπό-στόχο. Οι μέθοδοι κλειστού βρόχου, όπως είναι προφανές είναι αποδοτικότεροι και πιο εύρωστοι από τις μεθόδους ανοικτού βρόχου, αλλά απαιτούν την ύπαρξη μεθόδων για την συνεχή εκτίμηση της κατάστασης του ρομπότ και τον υπολογισμό των αποκλίσεων του από την. Στην προκειμένη περίπτωση, εφόσον παρέχεται η εκτίμηση της κατάστασης του ρομπότ επιλέχθηκε να χρησιμοποιηθεί ένας αλγόριθμος διάσχισης μονοπατιού, κλειστού βρόχου.

Το αντικείμενο της διάσχισης μονοπατιού για ρομποτικά οχήματα, έχει ερευνηθεί εκτενώς από τον ερευνητικό κόσμο, αλλά έχει επικεντρωθεί κατά κύριο λόγο σε ολονομικά ή ψευδο-ολονομικά ρομπότ, όπως ρομπότ με κινηματικό μοντέλο omnidirectional, differential drive και skid steer drive. Παράλληλα, υπάρχει και επαρκής έρευνα για την ακολούθηση μονοπατιού για μη ολονομικά ρομπότ, όπως το συμβατικό αυτοκίνητο με κινηματικό Ackermann. Παρόλα αυτά, η περίπτωση του κινηματικού μοντέλου τετραδιεύθυνσης που εξέταζεται δεν έχει μελετηθεί εκτενώς, λόγω της περιορισμένης χρήσης του σε ρομποτικές εφαρμογές, αλλά και στην αυτοκινητοβιομηχανία. Επίσης, ακόμα και οι ερευνητές που έχουν μελετήσει το πρόβλημα της διάσχισης μονοπατιού για ρομποτικά οχήματα με τετραδιεύθυνση, όπως στις δημοσιεύσεις [30], [31], [32], εστιάζονται περισσότερο στην κίνηση με αρνητική/αντίστροφη τετραδιεύθυνση, χωρίς να εκμεταλλεύονται πλήρως τις δυνατότητες του εν λόγω κινηματικού μοντέλου. Ακόμα και όταν μελετάται το πρόβλημα της θετικής τετραδιεύθυνσης, συνήθως χρησιμοποιείται είτε για την αύξηση της σταθερότητας του αυτοκινήτου, είτε για ομαλή αλλαγή

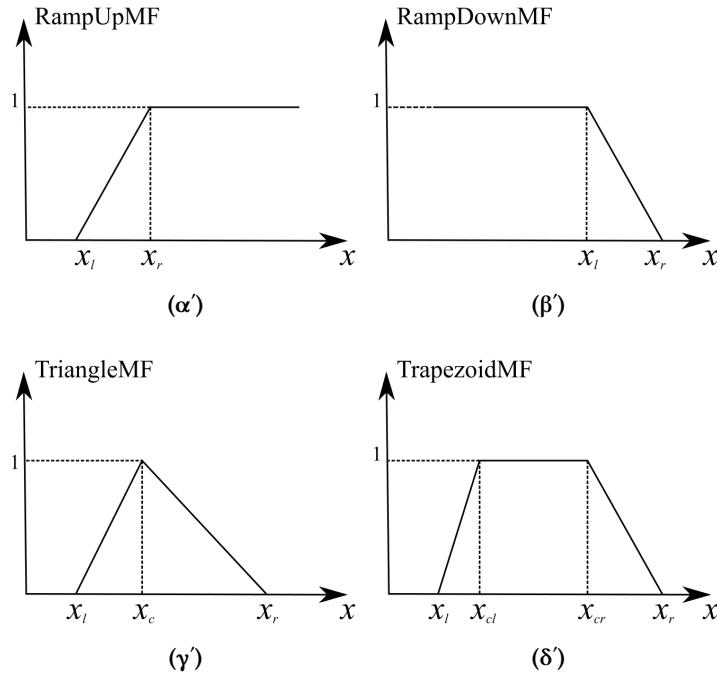
λωρίδων σε υψηλές ταχύτητες, όπου λαμβάνουν χώρα φαινόμενα διαταραχών, περίπτωση που δεν είχε νόημα να εξεταστεί στα πλαίσια της παρούσας εργασίας, καθώς μιλάμε για ρομποτικές εφαρμογές χαμηλών ταχυτήτων.

Μεγάλο ενδιαφέρον παρουσιάζει η πρόταση των Cariou et al. [33] για έναν αλγόριθμο διάσχισης μονοπατιού, μέσω προσαρμοστικού ελέγχου για την αντιστάθμιση φαινομένων ολίσθησης, ο οποίος ουσιαστικά στοχεύει στην μείωση της πλευρικής απόκλισης από το μονοπάτι που ακολουθεί, προσπαθώντας ταυτόχρονα να κρατήσει μικρό σφάλμα προσανατολισμού. Επίσης, εξίσου ενδιαφέρουσα, αν όχι περισσότερο, είναι και η πρόταση των Mendes et al. [34] για έναν αλγόριθμο διάσχισης μονοπατιού με ταυτόχρονη αποφυγή εμποδίων, βασισμένο σε Ασαφή Λογική (Fuzzy Logic), που ανεξαρτητοποιεί τις εντολές στρέψης των μπροστινών από τους πίσω τροχούς, όπου οι πίσω τροχοί χρησιμοποιούνται, όμοια με τον αλγόριθμο των [33] [33] για την αντιστάθμιση της πλευρικής απόκλισης από ένα δοσμένο μονοπάτι.

Με βάση τις δύο μεθόδους που παρουσιάστηκαν παραπάνω, αποφασίστηκε η ανάπτυξη ενός αλγορίθμου διάσχισης μονοπατιού, που θα εκμεταλλεύεται την θετική τετραδιεύθυνση για την αντιστάθμιση της πλευρικής απόκλισης σε περίπτωση ολίσθησης, διαταραχής ή ατελειών του νόμου ελέγχου, ενώ παράλληλα θα χρησιμοποιεί αρνητική τετραδιεύθυνση σε περίπτωση κλειστών στροφών. Επίσης, αποφασίστηκε η ανάπτυξη του αλγορίθμου να γίνει με Ασαφή Λογική, όπως η υλοποίηση των Mendes et al. [34], καθώς η Ασαφής Λογική διευκολύνει σημαντικά την διαδικασία υλοποίησης, ενώ παράλληλα προσφέρει υψηλούς βαθμούς προσαρμοστικότητας και επέκτασης.

Ελεγκτές Ασαφούς Λογικής

Οι μεταβλητές εισόδου ενός ελεγκτή ασαφούς λογικής (FLC), ονομάζονται ασαφής μεταβλητές (fuzzy variables) και συσχετίζονται με ένα σύνολο ασαφών συνόλων (fuzzy set). Κάθε ασαφές σύνολο χαρακτηρίζεται από μία συνάρτηση συμμετοχής (membership variable), που δηλώνει κατά πόσο τις εκατό, μία τιμή της ασαφούς μεταβλητής ανήκει στο εν λόγω ασαφές σύνολο. Για την υλοποίηση του ασαφούς ελεγκτή διάσχισης που παρουσιάζεται πιο κάτω, χρησιμοποιήθηκαν ράμπες, τριγωνικές και τραπεζοειδής συναρτήσεις συμμετοχής, οι οποίες παρουσιάζονται στο σχήμα 3.21 ορίζονται ως



Σχήμα 3.21: Συναρτήσεις συμμετοχής ασαφούς συνόλου, που χρησιμοποιήθηκαν στον ασαφή ελεγκτή διάσχισης μονοπατιού.

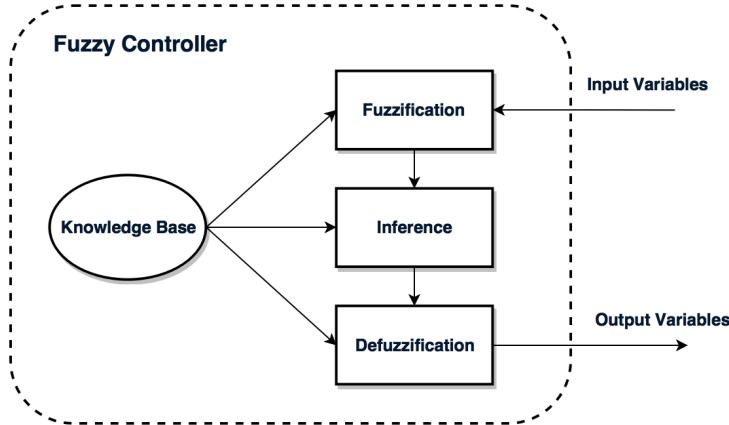
$$RampUpMF(x_l, x_r, x) = \begin{cases} 0, & \text{εάν } x \leq x_l \\ \frac{x-x_l}{x_r-x_l}, & \text{εάν } x_l \leq x \leq x_r \\ 1, & \text{εάν } x \geq x_r \end{cases} \quad (3.25)$$

$$RampDownMF(x_l, x_r, x) = \begin{cases} 1, & \text{εάν } x \leq x_l \\ \frac{x_r-x}{x_r-x_l}, & \text{εάν } x_l \leq x \leq x_r \\ 0, & \text{εάν } x \geq x_r \end{cases} \quad (3.26)$$

$$TriangleMF(x_l, x_c, x_r, x) = \begin{cases} 0, & \text{εάν } x \leq x_l \\ \frac{x-x_l}{x_c-x_l}, & \text{εάν } x_l \leq x \leq x_c \\ \frac{x_r-x}{x_r-x_c}, & \text{εάν } x_c \leq x \leq x_r \\ 0, & \text{εάν } x \geq x_r \end{cases} \quad (3.27)$$

$$TrapezoidMF(x_l, x_{cl}, x_{cr}, x_r, x) = \begin{cases} 0, & \text{εάν } x \leq x_l \\ \frac{x-x_l}{x_{cl}-x_l}, & \text{εάν } x_l \leq x \leq x_{cl} \\ 1, & \text{εάν } x_{cl} \leq x \leq x_{cr} \\ \frac{x_r-x}{x_r-x_{cr}}, & \text{εάν } x_c \leq x \leq x_{cr} \\ 0, & \text{εάν } x \geq x_r \end{cases} \quad (3.28)$$

Η γενική αρχιτεκτονική ενός ελεγκτή ασαφούς λογικής αποτελείται από τέσσερα βασικά τμήματα και τρία στάδια επεξεργασίας, όπως παρουσιάζονται στο σχήμα 3.22.



Σχήμα 3.22: Γενική Αρχιτεκτονική Ασαφούς Ελεγκτή.

Τα επιμέρους τμήματα της γενικής αρχιτεκτονικής του ελεγκτή ασαφούς λογικής περιγράφονται ως ακολούθως.

- Η **Βάση Γνώσης (Knowledge Base)** περιλαμβάνει τις ασαφείς μεταβλητές και τα αντίστοιχα τους ασαφή σύνολα, τα οποία αναπαρίστανται βάσει συναρτήσεων συμμετοχής, όπως επίσης και τους ασαφείς κανόνες, τύπου "if <conditions> then <action>".
- Στο στάδιο της **Ασαφοποίηση (Fuzzification)**, μία κανονική τιμή (crisp value) μίας ασαφούς μεταβλητής εισόδου ασαφοποιείται σε ζεύγη (ασαφές σύνολο, ποσοστό συμμετοχής), μέσω αντιστοίχισης της τιμής με την συνάρτηση συμμετοχής κάθε ασαφούς συνόλου.
- Στο στάδιο του **Συμπερασμού (Inference)**, λαμβάνονται οι ασαφείς είσοδοι, εφαρμόζονται οι ασαφείς κανόνες σε αυτούς και παράγεται μία ασαφής έξοδος.
- Στο στάδιο της **Αποασαφοποίησης (Defuzzification)**, τα ασαφή σύνολα που παράγονται από κάθε κανόνα στο στάδιο του συμπερασμού μετατρέπονται σε κανονικές τιμές, μέσω σταθμισμένου μέσου (weighted average) Takagi-Sugeno:

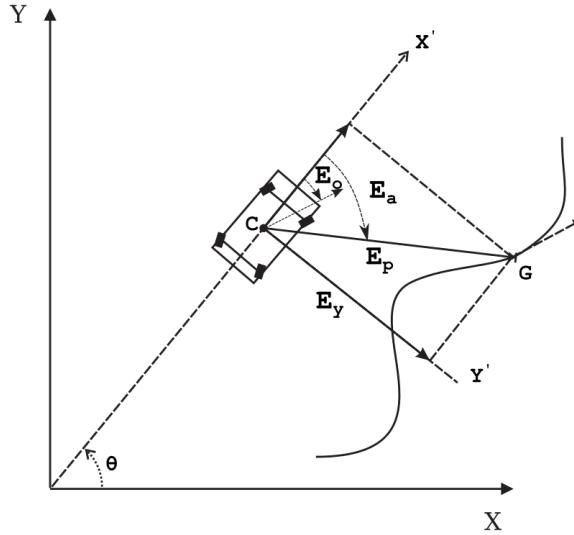
$$WA = \frac{\sum_{i=1}^n \mu(x_i)x_i}{\sum_{i=1}^n \mu(x_i)} \quad (3.29)$$

Ασαφής Ελεγκτής Διάσχισης Μονοπατιού

Όπως, προαναφέρθηκε, στόχος του αλγορίθμου διάσχισης μονοπατιού, κλειστού βρόχου είναι να ακολουθήσει όσο πιο εύρωστα γίνεται μία διαδρομή, μέσω του ορισμού υπό-στόχων κατά μήκος του μονοπατιού. Επομένως, το μονοπάτι προς διάσχιση αναπαρίσταται ως μία ακολουθία προσανατολισμένων σημείων στο επίπεδο. Επομένως, βάσει ενός μονοπατιού $\{p_1, p_2, \dots, p_n\}$ που πρέπει να ακολουθήσει το ρομπότ για να μεταβεί από μία αρχική κατάσταση $q_I = [x_I, y_I, \theta_I]$ σε ένα στόχο - μία τελική κατάσταση $q_F = [x_F, y_F, \theta_F]$, επιλέγεται ένας υπό-στόχος $q_G = [x_G, y_G, \theta_G]$ πάνω στο εν λόγω μονοπάτι. Σαν αποτέλεσμα, το πρόβλημα της διάσχισης μονοπατιού μετασχηματίζεται σε ένα πρόβλημα προσέγγισης διαδοχικών υπό-στόχων, το οποίο συνεπάγεται έναν νόμο ελέγχου που θα τείνει να μειώνει την απόκλιση μεταξύ του ρομπότ και του τρέχοντα υπό-στόχου, μέχρι τελικά να φτάσει στην επιθυμητή τελική κατάσταση.

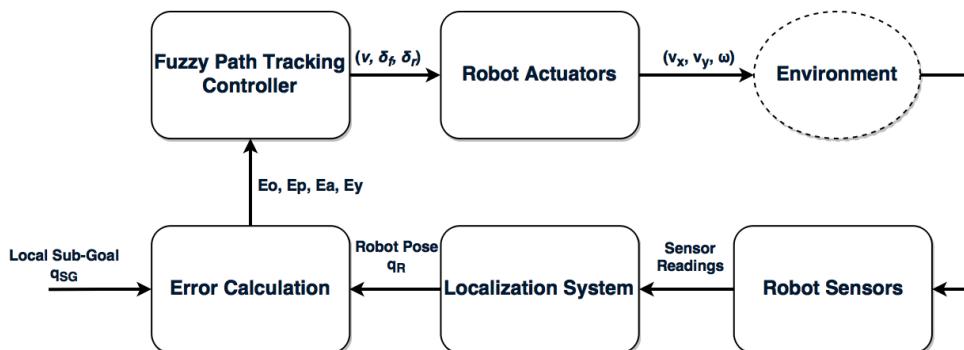
Η απόκλιση μεταξύ δύο καταστάσεων στο επίπεδο, μπορεί να οριστεί με διάφορους τρόπους, μέσω ενός συνόλου σφαλμάτων. Μία συνήθης επιλογή, αποτελεί η ευκλείδια απόσταση και το σφάλμα προσανατολισμού μεταξύ δύο καταστάσεων. Στην προκειμένη περίπτωση, για την ικανοποίηση των στόχων του αλγορίθμου διάσχισης μονοπατιού, επιλέχθηκε το ακόλουθο σύνολο τεσσάρων σφαλμάτων που παρουσιάζεται και παραστατικά στο σχήμα 3.23.

- E_o : Σφάλμα προσανατολισμού μεταξύ κατάστασης ρομπότ και τρέχοντα υπό-στόχου.
- E_p : Σφάλμα απόστασης μεταξύ θέσης ρομπότ και τρέχοντα υπό-στόχου.
- E_a : Σφάλμα γωνιακής απόκλισης μεταξύ θέσης ρομπότ και τρέχοντα υπό-στόχου.
- E_y : Σφάλμα πλευρικής απόκλισης μεταξύ θέσης ρομπότ και τρέχοντα υπό-στόχου.



Σχήμα 3.23: Σφάλματα μεταξύ τρέχουσας κατάστασης ρομπότ και τρέχοντα υπό-στόχου που επιλέχθηκαν για την υλοποίηση του αλγορίθμου διάσχισης μονοπατιού.

Τα σφάλματα E_o, E_p, E_a, E_y χρησιμοποιούνται ως είσοδοι στον ελεγκτή ασαφούς λογικής που σχεδιάστηκε για την ακολούθηση μονοπατιού. Αντίστοιχα, ως έξοδοι του ελεγκτή ορίζονται οι γωνίες δ_f, δ_r στρέψης των μπροστά και πίσω τροχών και η ταχύτητα v του οχήματος, όπως παρουσιάζεται στο σχήμα 3.24

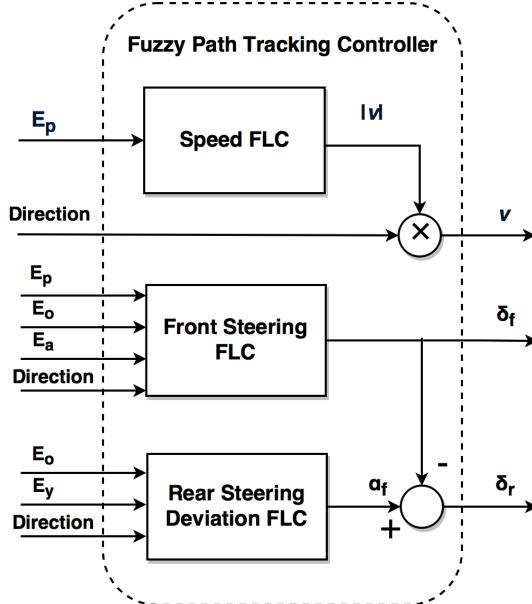


Σχήμα 3.24: Βρόχος ελέγχου λειτουργίας διάσχισης μονοπατιού.

Επίσης, για την μείωση του απαιτούμενου αριθμού ασαφών κανόνων του προβλήματος ορίζεται η μεταβλητή εισόδου, *Direction*, που δηλώνει την φορά κίνησης συναρτήσει του σφάλματος γωνίας απόκλισης E_a ως

$$\text{Direction} = \begin{cases} -1, & \text{εάν } |E_a| > 120^\circ \\ 1, & \text{εάν } |E_a| < 120^\circ \end{cases} \quad (3.30)$$

Ο ελεγκτής διάσχισης μονοπατιού αποτελείται από τρεις επιμέρους ανεξάρτητους ελεγκτές, κάθε ένας υπεύθυνος για τον καθορισμό μίας από τις τρεις εξόδους δ_f , δ_r , v του συστήματος. Συγκεκριμένα, οι ελεγκτές παρουσιάζονται στο σχήμα 3.25 και ορίζονται ως ακολούθως.



Σχήμα 3.25: Ο ελεγκτής διάσχισης μονοπατιού.

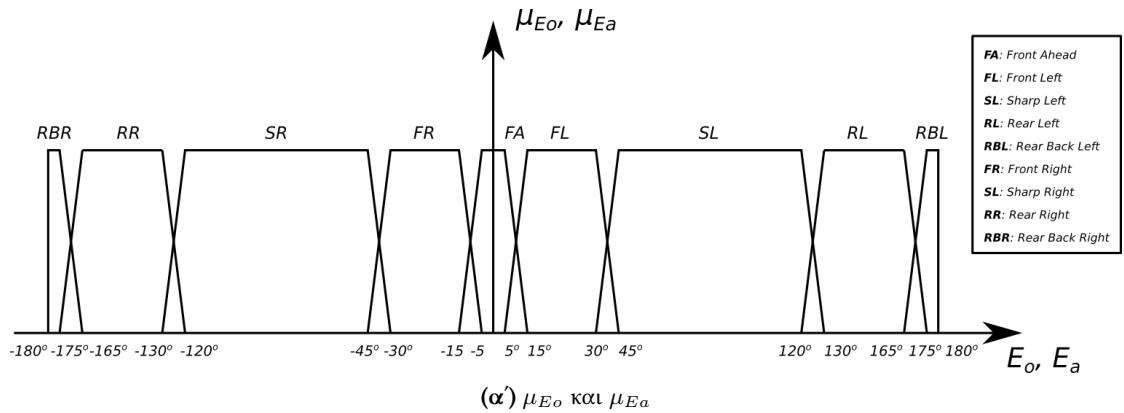
- Ο ασαφής ελεγκτής ταχύτητας (Speed Fuzzy Logic Controller) παίρνει ως είσοδο το σφάλμα θέσης E_p και την φορά κίνησης $Direction$ και παρέχει ως έξοδο την ταχύτητα v κίνησης του ρομπότ.
- Ο ασαφής ελεγκτής μπροστινοδιεύθυνσης (Front Steering Fuzzy Logic Controller) παίρνει ως είσοδο τα σφάλματα E_o , E_a , E_p και την φορά κίνησης $Direction$ και παράγει ως έξοδος την γωνία στρέψης δ_f των μπροστινών τροχών.
- Ο ασαφής ελεγκτής απόκλισης πισωδιεύθυνσης (Rear Steering Deviation Fuzzy Logic Controller) παίρνει ως είσοδο τα σφάλματα E_o , E_y και την φορά κίνησης $Direction$ και παράγει την επιθυμητή γωνία απόκλισης της γωνίας στρέψης των πίσω τροχών από την αντίθετη γωνία στρέψης των μπροστινών τροχών. Επομένως, η γωνία στρέψης των πίσω τροχών προκύπτει ως

$$\delta_r = -\delta_f + \alpha_r \quad (3.31)$$

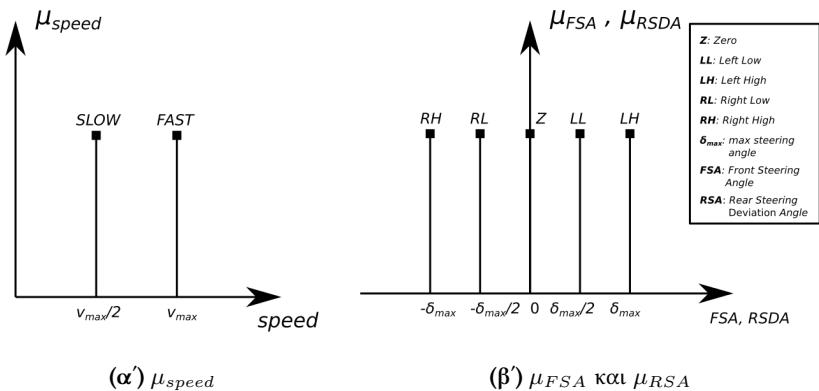
Στη συνέχεια για κάθε μία από τις εισόδους του ελεγκτή ορίζονται ένα σύνολο ασαφών συνόλων και οι αντίστοιχες συναρτήσεις συμμετοχής, όπως παρουσιάζεται στο σχήμα 3.26.

Αντίστοιχα, ορίζονται και τα ασαφή σύνολα των εξόδων του ελεγκτή διάσχισης μονοπατιού, όπως παρουσιάζεται στο σχήμα 3.27.

Έχοντας ορίσει τις μεταβλητές εισόδου, εξόδου και τα αντίστοιχα ασαφή σύνολα και συναρτήσεις συμμετοχής, το μόνο που μένει είναι ο ορισμός των ασαφών κανόνων της βάση γνώσης του ελεγκτή. Η επιλογή των ασαφών κανόνων των τριών επιμέρους ελεγκτών του ελεγκτή διάσχισης μονοπατιού, πραγματοποιήθηκε βάσει της επιθυμητής συμπεριφοράς του ελεγκτή ανά ένα σύνολο περιπτώσεων. Οι περιπτώσεις που πρέπει ικανοποιούν οι ασαφείς κανόνες έχουν ως εξής:



Σχήμα 3.26: Οι συναρτήσεις συμμετοχής των μεταβλητών εισόδου του ελεγκτή διάσχισης μονοπατιού.



Σχήμα 3.27: Οι συναρτήσεις συμμετοχής των μεταβλητών εξόδου του ελεγκτή διάσχισης μονοπατιού.

- Η ταχύτητα θα πρέπει να η μέγιστη δυνατή, όταν το όχημα βρίσκεται μακριά από τον στόχο και να ελαττώνεται όταν πλησιάζει σ' αυτόν.
- Στην περίπτωση που το όχημα βρίσκεται μακριά από τον στόχο θα πρέπει η γωνία στρέψης των μπροστινών τροχών να στρέφεται ανάλογα με το μέγεθος και την κατεύθυνση του σφάλματος γωνιακής απόκλισης. Γι αυτήν την περίπτωση δεν μας ενδιαφέρει, τόσο το σφάλμα προσανατολισμού, όσο η προσέγγιση του στόχου.
- Στην περίπτωση που το όχημα βρίσκεται κοντά στον στόχο, αντιθέτως, δεν μας ενδιαφέρει η προσέγγιση του στόχου, μιας και είμαστε είδη κοντά, αλλά η διόρθωση του σφάλματος προσανατολισμού.

4. Στην περίπτωση που το σφάλμα προσανατολισμού είναι μικρό και η πλευρική απόκλιση δεν είναι εξαιρετικά μεγάλη, θα πρέπει η απόκλιση της γωνίας στρέψης των πίσω τροχών να είναι μεγάλη, ούτως ώστε, το όχημα να κινείται με θετική τετραδιεύθυνση, κρατώντας το σφάλμα προσανατολισμού μικρό. Η περίπτωση αυτή είναι χρήσιμη για την περίπτωση ολίσθησης ή για διόρθωση μικρής πλευρικής απόκλισης.
5. Σε περίπτωση που το σφάλμα προσανατολισμού είναι μικρό, αλλά το σφάλμα πλευρικής απόκλισης είναι πολύ μεγάλο τότε θα πρέπει η απόκλιση της γωνίας στρέψης των πίσω τροχών να είναι μηδενική, ούτως ώστε, το όχημα να κινείται με αρνητική τετραδιεύθυνση, με στόχο την προσέγγιση του μονοπατιού, διορθώνοντας το σφάλμα γωνιακής απόκλισης και πλευρικής απόκλισης, χωρίς να λαμβάνεται υπόψιν το σφάλμα προσανατολισμού.
6. Ο ελεγκτής θα πρέπει να μπορεί να ικανοποιεί τις παραπάνω περιπτώσεις και για τις δύο δυνατές φορές κίνησης, εμπρός και πίσω.

Πίνακας 3.2: Οι ασαφείς κανόνες του ελεγκτή ταχύτητας.

Speed Fuzzy Rules	if Ep is CLOSE then Speed is SLOW if Ep is FAR then Speed is FAST
-------------------	--

Πίνακας 3.3: Οι ασαφείς κανόνες του ελεγκτή της γωνίας στρέψης των μπροστινών τροχών.

FSA Fuzzy Rules	if Ep is FAR and Ea is RBL then FSA is Z if Ep is FAR and Ea is RL then FSA is LH if Ep is FAR and Ea is SL then FSA is LH if Ep is FAR and Ea is FL then FSA is LL if Ep is FAR and Ea is FA then FSA is Z if Ep is FAR and Ea is FR then FSA is RL if Ep is FAR and Ea is SR then FSA is RH if Ep is FAR and Ea is RR then FSA is RH if Ep is FAR and Ea is RBR then FSA is Z if Ep is CLOSE and Direction is FW and Eo is SL then FSA is LH if Ep is CLOSE and Direction is FW and Eo is FL then FSA is LL if Ep is CLOSE and Direction is FW and Eo is FA then FSA is Z if Ep is CLOSE and Direction is FW and Eo is FR then FSA is RL if Ep is CLOSE and Direction is FW and Eo is SR then FSA is RH if Ep is CLOSE and Direction is BW and Eo is SL then FSA is RH if Ep is CLOSE and Direction is BW and Eo is FL then FSA is RL if Ep is CLOSE and Direction is BW and Eo is FR then FSA is LL if Ep is CLOSE and Direction is BW and Eo is SR then FSA is LH
-----------------	--

Πίνακας 3.4: Οι ασαφείς κανόνες του ελεγκτή της απόκλισης της γωνίας στρέψης των πίσω τροχών.

RSDA Fuzzy Rules	if Eo is not FA and Eo is not FR and Eo is not FL then RSDA is Z if Direction is FW and Eo is FA and Ey is BP then RSDA is Z if Direction is FW and Eo is FL and Ey is BP then RSDA is Z if Direction is FW and Eo is FR and Ey is BP then RSDA is Z if Direction is FW and Eo is FA and Ey is SP then RSDA is LH if Direction is FW and Eo is FL and Ey is SP then RSDA is LH if Direction is FW and Eo is FR and Ey is SP then RSDA is LH if Direction is FW and Eo is FA and Ey is Z then RSDA is Z if Direction is FW and Eo is FL and Ey is Z then RSDA is Z if Direction is FW and Eo is FR and Ey is Z then RSDA is Z if Direction is FW and Eo is FA and Ey is SN then RSDA is RH if Direction is FW and Eo is FL and Ey is SN then RSDA is RH if Direction is FW and Eo is FR and Ey is SN then RSDA is RH if Direction is FW and Eo is FA and Ey is BN then RSDA is Z if Direction is FW and Eo is FL and Ey is BN then RSDA is Z if Direction is FW and Eo is FR and Ey is BN then RSDA is Z if Direction is BW and Eo is FA and Ey is BP then RSDA is Z if Direction is BW and Eo is FL and Ey is BP then RSDA is Z if Direction is BW and Eo is FR and Ey is BP then RSDA is Z if Direction is BW and Eo is FA and Ey is SP then RSDA is LH if Direction is BW and Eo is FL and Ey is SP then RSDA is LH if Direction is BW and Eo is FR and Ey is SP then RSDA is LH if Direction is BW and Eo is FA and Ey is Z then RSDA is Z if Direction is BW and Eo is FL and Ey is Z then RSDA is Z if Direction is BW and Eo is FR and Ey is Z then RSDA is Z if Direction is BW and Eo is FA and Ey is SN then RSDA is RH if Direction is BW and Eo is FL and Ey is SN then RSDA is RH if Direction is BW and Eo is FR and Ey is SN then RSDA is RH if Direction is BW and Eo is FA and Ey is BN then RSDA is Z if Direction is BW and Eo is FL and Ey is BN then RSDA is Z if Direction is BW and Eo is FR and Ey is BN then RSDA is Z
------------------	--

Κεφάλαιο 4

Αρχιτεκτονική Συστήματος

Στο παρόν κεφάλαιο θα παρουσιαστεί η αρχιτεκτονική του συστήματος που υλοποιήθηκε, το λογισμικό και τα επιμέρους εργαλεία χρησιμοποιήθηκαν, με στόχο την αυτόνομη ή μη λειτουργία της ρομποτικής πλατφόρμας Monstertruck. Επίσης, παρουσιάζονται και τα περιβάλλοντα προσομοίωσης του ρομπότ στις δύο και στις τρεις διαστάσεις, που χρησιμοποιήθηκαν κατά την υλοποίηση και την αποσφαλμάτωση του συστήματος.

4.1 Robotic Operating System

Οι περισσότερες ρομποτικές εφαρμογές απαιτούν ένα εξαιρετικά σύνθετο λογισμικό για να μπορούν να εκτελέσουν βασικές λειτουργίες, λογισμικό που μεγεθύνεται όσο εξελίσσεται το αντικείμενο της ρομποτικής και αυξάνουν οι απαιτήσεις της. Παράλληλα, κάθε ρομποτικό σύστημα παρουσιάζει ένα βαθμό ιδιαίτερότητας, λόγω διαφορετικού hardware, λογισμικού και σκοπού λειτουργίας, γεγονός που καθιστά την επαναχρησιμοποίηση κώδικα εξαιρετικά δύσκολο εγχείρημα. Επίσης, λόγω του μεγάλου μεγέθους του λογισμικού και των απαιτήσεων ταυτόχρονης λειτουργίας και επικοινωνίας μεταξύ των επιμέρους τμημάτων, από το επίπεδο των οδηγών (drivers) των επιμέρους συσκευών του ρομποτικού συστήματος μέχρι τα επίπεδα της αντίληψης, επεξεργασίας και λήψης αποφάσεων, καθίσταται απαραίτητη η ανάγκη για μία μέθοδο υψηλής κλίμακας ενσωμάτωσης λογισμικού.

Για την επίλυση των παραπάνω προβλημάτων και όχι μόνο, οι Quigley et al. [35] ανέπτυξαν το ROS (Robot Operating System)¹. Το ROS αποτελεί ένα ευέλικτο framework ρομποτικής που προσφέρει ένα σύνολο εργαλείων και βιβλιοθηκών λογισμικού, ενώ παράλληλα επιβάλλει ένα σύνολο αυστηρών συμβάσεων, με στόχο την απλοποίηση της ανάπτυξης σύνθετων και εύρωστων εφαρμογών για πληθώρα ρομποτικών συστημάτων. Επίσης, το ROS σχεδιάστηκε από την αρχή με απώτερο στόχο την συνεργασία του κόσμου της ρομποτικής για την κατανεμημένη ανάπτυξη λογισμικού ανοικτού κώδικα για ρομποτικές εφαρμογές. Το ROS χαρακτηρίζεται, γενικά, ως

- **Peer-to-Peer**, καθώς ένα σύστημα βασισμένο στο ROS αποτελείται από ένα σύνολο διεργασιών που τρέχουν σε έναν ή περισσότερους hosts που συνδέονται μεταξύ τους μέσω μίας τοπολογίας Peer-to-Peer και μπορούν να επικοινωνούν μεταξύ τους διαμέσου της διεργασίας του ROS Master.
- **Πολύγλωσσο**, καθώς επιτρέπει την χρήση διαφορετικών γλωσσών προγραμματισμού (C++, python κλπ) για την ανάπτυξη εφαρμογών, που μπορούν να επικοινωνούν μεταξύ τους, μέσω μίας αυστηρά ορισμένης διεπαφής.
- **Βασισμένο σε εργαλεία**, καθώς χρησιμοποιείται μεγάλος αριθμός μικρών εργαλείων για την ανάπτυξη και λειτουργία των επιμέρους τμημάτων του συστήματος.
- **Λεπτό**, από την άποψη ότι η ανάπτυξη των αλγορίθμων πραγματοποιείται σε ανεξάρτητες βιβλιοθήκες, οι οποίες έπειτα μπορούν να χρησιμοποιηθούν σε ένα σύστημα βασισμένο στο ROS,

¹ROS Tutorials: www.wiki.ros.org/ROS/StartGuide, Mastering ROS for Robotics Programming [36]

μέσω μικρών προγραμμάτων που εξάγουν την λειτουργικότητα των εν λόγω βιβλιοθηκών μέσω του συστήματος διεπαφών του ROS.

- **Ανοικτού Κώδικα,** ιδιότητα απαραίτητη και κρίσιμη για την κατανόηση της λειτουργίας του, την επέκταση του, αλλά και την ανάπτυξη και αποσφαλμάτωση του λογισμικού ρομποτικών εφαρμογών σε όλα τα επίπεδα.

Ένα σύστημα βασισμένο στο ROS, αποτελείται από ένα σύνολο διεργασιών που ονομάζονται κόμβοι (*nodes*) και επικοινωνούν μεταξύ τους, θυμίζοντας τοπολογικά έναν γράφο. Το ROS προσφέρει μία διεπαφή διάδοσης μηνυμάτων μεταξύ των κόμβων και γι αυτό το λόγο αναφέρεται συχνά ως middleware. Συγκεκριμένα, Η επικοινωνία μεταξύ των κόμβων επιτυγχάνεται μέσω των μηνυμάτων (*ROS Messages*) που αποτελούν ένα σύνολο αυστηρά ορισμένων δομών δεδομένων γραμμένων σε γλώσσα message IDL(Interface Description Language). Τα μηνύματα υποστηρίζουν βασικούς τύπους δεδομένων, όπως *int*, *float* κοκ, όπως επίσης και πιο σύνθετες δομές που συντίθενται από τους βασικούς τύπους δεδομένων ή άλλες σύνθετες δομές. Ένας κόμβος, λοιπόν μπορεί να εκδίδει ένα μήνυμα σε ένα *topic*, το οποίο αποτελεί ουσιαστικά ένα *string*, ενώ ένας άλλος κόμβος μπορεί να λάβει το μήνυμα ακούγοντας στο αντίστοιχο *topic*, επιτρέποντας με αυτόν τον τρόπο την επικοινωνία μεταξύ των δύο κόμβων. Ένας κόμβος, μπορεί, παράλληλα να εκδίδει μηνύματα σε διαφορετικά *topic* και να ακούει μηνύματα από άλλα *topic*, χωρίς κάποιο περιορισμό, ενώ αντίστοιχα, ένα *topic* μπορεί να λαμβάνει μηνύματα από πολλούς κόμβους ταυτόχρονα ή να ακούνε σ' αυτό πολλοί κόμβοι ταυτόχρονα, χωρίς ο ένας να γνωρίζει την ύπαρξη του άλλου.

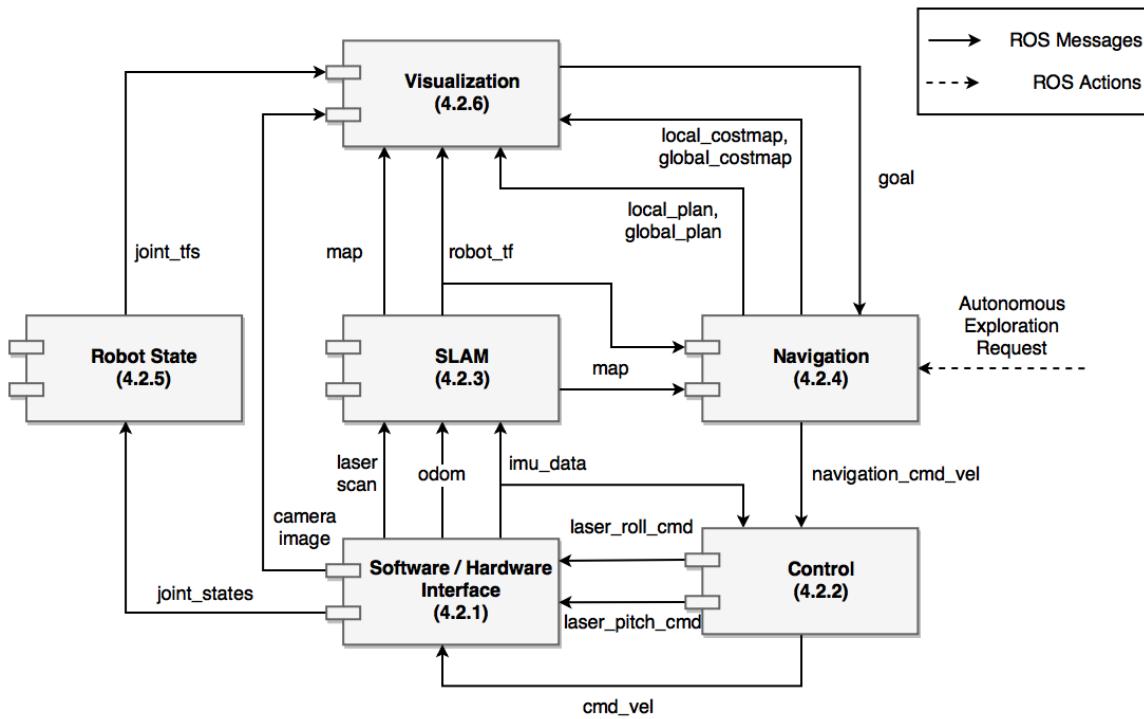
Τα μηνύματα, παρόλα αυτά, δεν αποτελούν την βέλτιστη μέθοδο για σύγχρονες δοσοληψίες μεταξύ δύο κόμβων. Γι αυτό αναπτύχθηκαν και οι υπηρεσίες (*services*), οι οποίες είναι όμοιες με *web services* και αποτελούνται από δύο μηνύματα, ένα για το αίτημα (*request*) της υπηρεσίας και ένα για την απόκριση (*response*), όπου αντίθετα με τα μηνύματα, μόνο ένας κόμβος - διακομιστής (*server*) μπορεί να παρέχει μία συγκεκριμένη υπηρεσία, αλλά πολλοί κόμβοι - πελάτες (*clients*) μπορούν να την αιτηθούν. Σε περιπτώσεις, όμως, που μία υπηρεσία απαιτεί μεγάλο χρόνο και απαιτείται να παρέχεται κάποια μορφή ανάδρασης (*feedback*) για την πρόοδο του αιτήματος ή να παρέχεται δυνατότητα διακοπής αυτής, αντί για υπηρεσίες χρησιμοποιούνται οι δράσεις (*actions*).

Η επικοινωνία μεταξύ δύο ή περισσότερων κόμβων καθίσταται δυνατή από τον κόμβο *ROS Master*. Ο κόμβος *ROS Master* παρέχει υπηρεσίες ονοματολογίας και κατοχύρωσης στους υπόλοιπους κόμβους του συστήματος και είναι αυτός που αντιστοιχίζει τους εκδότες (*publishers*) και συνδρομητές (*subscribers*) μηνυμάτων, αλλά και διακομιστές και πελάτες υπηρεσιών ή δράσεων. Επιτρέπει, δηλαδή σε ξεχωριστούς κόμβους να βρουν ο ένας τον άλλο, ενώ έπειτα, αυτοί επικοινωνούν μεταξύ τους χωρίς διαμεσολάβηση. Ο κόμβος *ROS Master*, πέρα από τις παραπάνω λειτουργίες, παρέχει, επίσης και τον διακομιστή παραμέτρων (*Parameter Server*), ο οποίος αποτελεί ένα "λεξικό" πολλών μεταβλητών προσπελάσιμων από όλους τους κόμβους που είναι συνδεδεμένοι στον ίδιο *ROS Master*.

Επιπρόσθετα, μία πολύ σημαντική βιβλιοθήκη του ROS, αποτελεί η βιβλιοθήκη *tf*. Η βιβλιοθήκη *tf* σχεδιάστηκε με στόχο την παροχή ενός τυποποιημένου τρόπου αποθήκευσης και ανανέωσης των διαφόρων πλαισίων συντεταγμένων του ρομπότ και του περιβάλλοντος, αλλά και παροχή δυνατότητας μετασχηματισμού συντεταγμένων από το ένα πλαίσιο στο άλλο, χωρίς να απαιτείται γνώση όλων των πλαισίων του συστήματος.

4.2 Αρχιτεκτονική Λογισμικού Ρομποτικής Πλατφόρμας Monstertruck

Η υλοποίηση του λογισμικού της ρομποτικής πλατφόρμας *Monstertruck* βασίστηκε στο framework του ROS που παρουσιάστηκε στην προηγούμενη ενότητα και χωρίζεται σε έξι βασικά τμήματα, όπως παρουσιάζεται στο σχήμα 4.1.



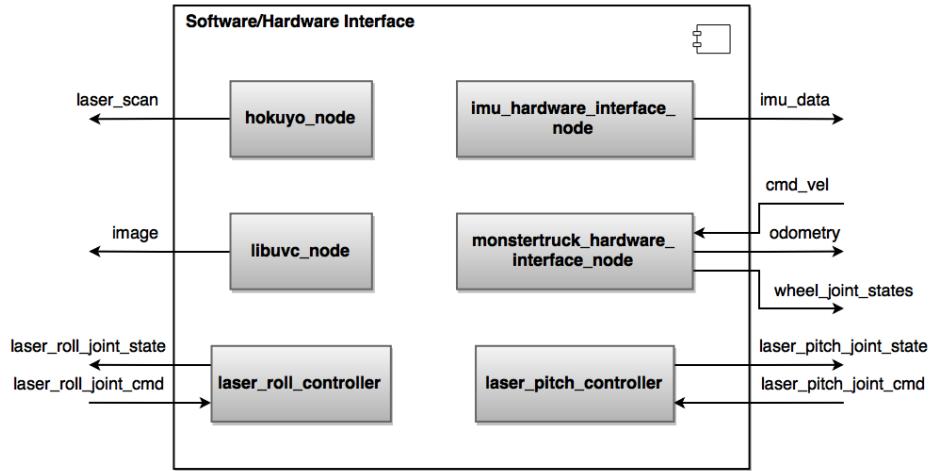
Σχήμα 4.1: Το διάγραμμα τμημάτων του λογισμικού της ρομποτικής πλατφόρμας Monstertruck.

Κάθε τμήμα του λογισμικού που παρουσιάζεται στο σχήμα 4.1 αποτελείται από ένα σύνολο κόμβων, συμβατών με το ROS και υπεύθυνων για τις λειτουργίες που απαιτούνται από το αντίστοιχο τμήμα, όπως παρουσιάζεται στη συνέχεια.

4.2.1 Software/Hardware Interface

Το τμήμα του *Software/Hardware Interface* αποτελεί ένα επίπεδο διεπαφής των επιμέρους περιφερειακών συσκευών του ρομπότ με το framework του ROS. Η δομή, η οποία χρησιμοποιείται στο λογισμικό της ομάδας P.A.N.D.O.R.A για έναν κόμβο *Software/Hardware Interface* αποτελείται βασικά από τον driver της συσκευής, συνήθως ανεξάρτητο από το ROS, που επιτρέπει την επικοινωνία μεταξύ του υπολογιστή και της συσκευής, μέσω ενός πρωτοκόλλου σειριακής επικοινωνίας, έναν wrapper που αποτελεί την διεπαφή μεταξύ του driver και του ROS και τέλος έναν ROS Controller, ο οποίος αποτελεί ένα plugin που μπορεί να φορτώνεται ή να σταματάει δυναμικά και είναι υπεύθυνο για την έκδοση και την λήψη μετρήσεων και εντολών αντίστοιχα, ως μηνύματα του ROS. Στην συνέχεια παρουσιάζεται το σύνολο των κόμβων του επιπέδου *Software/Hardware Interface*.

- Ο κόμβος **hokuyo_node** [37] είναι υπεύθυνος για την διεπαφή των σαρωτή λέιζερ Hokuyo URG-04LX, με το framework του ROS, εκδίδοντας μηνύματα ROS με τις σαρώσεις του αισθητήρα, όπως επίσης και τις παραμέτρους του (πχ. εμβέλεια, εύρος γωνίας σάρωσης κλπ).
- Οι κόμβοι **laser_roll_controller** και **laser_pitch_controller** είναι **dynamixel_controllers** [38] και είναι υπεύθυνοι για την διεπαφή των έξυπνων σερβοκινητήρων Dynamixel, που χρησιμοποιούνται για την σταθεροποίηση του σαρωτή λέιζερ στο οριζόντιο επίπεδο.
- Ο κόμβος **libuvc_node** του πακέτου **libuvc_camera** [39] είναι υπεύθυνος για την διεπαφή της κάμερας Logitech C905, λαμβάνοντας εικόνες-frames από αυτήν και έπειτα μεταδίδοντας τις μέσω μηνυμάτων ROS.



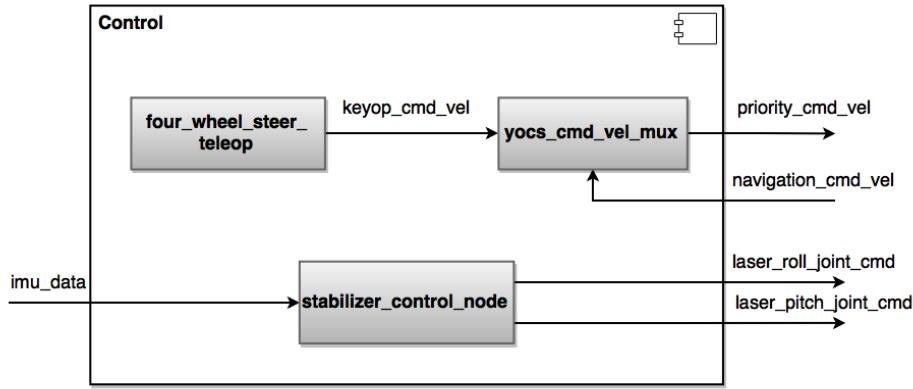
Σχήμα 4.2: Το διάγραμμα των κόμβων του τμήματος Software/Hardware Interface της ρομποτικής πλατφόρμας Monstertruck.

- Ο κόμβος `imu.hardware_interface` του πακέτου `pandora_imu.hardware_interface` είναι υπεύθυνος για την διεπαφή μεταξύ της πυξίδας Compass OS4000 και του ROS, και χρησιμοποιείται για την παροχή του προσανατολισμού του ρομπότ ως προς τους άξονες x , y , z , δηλαδή τις μετρήσεις roll, pitch, yaw, όπως επίσης και για τις γραμμικές επιταχύνσεις a_x , a_y , a_z , του ρομπότ ως προς τους τρεις άξονες.
- Ο κόμβος `monstertruck.hardware_interface` του πακέτου `pandora_monstertruck.hardware_interface`, που γράφτηκε στα πλαίσια της παρούσας διπλωματικής εργασίας είναι υπεύθυνος για την διεπαφή του λογισμικού με τον ελεγκτή EPOS 24/1 του κινητήρα του συστήματος τετρακίνησης και με τον ελεγκτή Pololu Micro Maestro των σερβοκινητήρων του συστήματος τετραδιεύθυνσης της ρομποτικής πλατφόρμας Monstertruck. Επομένως, λαμβάνει μηνύματα-εντολές κίνησης του οχήματος, τις μετατρέπει σε εντολές κίνησης του κινητήρα και των σερβοκινητήρων, βάσει της ανάλυσης που προηγήθηκε στην ενότητα 2.3.3 και έπειτα τις προωθεί σ' αυτούς. Παράλληλα, ειδίδει μηνύματα σχετικά με τις μετρήσεις ταχύτητας και θέσης των τροχών.

4.2.2 Control

Το τμήμα *Control* περιλαμβάνει κόμβους υπεύθυνους για τον μερικό ή ολικό έλεγχο των κινούμενων μερών της ρομποτικής πλατφόρμας Monstertruck. Συγκεκριμένα, περιλαμβάνει τους ακόλουθους κόμβους, όπως παρουσιάζονται και στο σχήμα 4.3.

- Ο κόμβος `stabilizer_control_node` του πακέτου `pandora_stabilizer` είναι υπεύθυνος για τον έλεγχο θέσης των έξυπνων σερβοκινητήρων Dynamixel, που χρησιμοποιούνται για την σταθεροποίηση του σαρωτή λέιζερ στο οριζόντιο επίπεδο. Συγκεκριμένα, λαμβάνει μηνύματα με πληροφορία για τις κλίσεις roll, pitch του οχήματος, από την πυξίδα και έπειτα στέλνει μηνύματα-εντολές με αντίστροφες κλίσης roll, pitch στους έξυπνους σερβοκινητήρες της ρομποτικής πλατφόρμας Monstertruck με στόχο την αντιστάθμιση των κλίσεων και την διατήρηση του σαρωτή λέιζερ στο οριζόντιο επίπεδο. Η εν λόγω αντιστάθμιση κλίσης είναι απαραίτητη, καθώς ένας αλγόριθμος χαρτογράφησης 2D, όπως αυτοί που εξετάζονται μπορεί να παρερμηνεύσει τις σαρώσεις και να παράγει έναν εσφαλμένο χάρτη.
- Για τον τηλεχειρισμό της ρομποτικής πλατφόρμας Monstertruck, αναπτύχθηκε ο κόμβος `teleop`, ο οποίος επιτρέπει τον έλεγχο του οχήματος με βάση το



Σχήμα 4.3: Το διάγραμμα των κόμβων του τμήματος Control της ρομποτικής πλατφόρμας Monstertruck.

κινηματικό τετραδιεύθυνσης 2.3.2, δίνοντας την δυνατότητα κίνησης και με θετική και με αρνητική τετραδιεύθυνση. Οι δύο παραπάνω κόμβοι τηλεχειρισμού εκδίδουν μηνύματα εντολών, τα οποία λαμβάνονται από τον κόμβο monstertruck_hardware_interface, ο οποίος τις εκτελεί.

- Ο κόμβος `yocs_cmd_vel_mux` [40] χρησιμοποιήθηκε σαν διαμεσολαβητής μεταξύ εντολής και εκτέλεσης για την προτεραιοποίηση των πηγών εντολών ταχύτητας. Συγκεκριμένα, ένα αυτόνομο ρομποτικό σύστημα όπως η ρομποτική πλατφόρμα Monstertruck μπορεί να λαμβάνει εντολές ταχύτητας από διάφορες πηγές ταυτόχρονα, όπως ο κόμβος αυτόνομης πλοήγησης, ο κόμβος τηλεχειρισμού, κάποιος κόμβος ελέγχου ασφαλείας κοκ, με αποτέλεσμα να δημιουργείται σύγχυση του ρομπότ και η συμπεριφορά του να καθίσταται απρόβλεπτη. Γι αυτό το λόγο χρησιμοποιείται ο κόμβος `yocs_cmd_vel_mux` ο οποίος αποτελεί ουσιαστικά έναν πολυπλέκτη εντολών ταχυτήτων με βάση την προτεραιότητα των κόμβων από τους οποίους προέρχονται οι εντολές. Στην προκειμένη περίπτωση παρέχεται υψηλότερη προτεραιότητα για τον κόμβο τηλεχειρισμού από τον κόμβο αυτόνομης πλοήγησης, ούτως ώστε να μπορεί να παρακαμφθεί η αυτόνομη λειτουργία σε περίπτωση κινδύνου ή ανεπιθύμητης συμπεριφοράς.

4.2.3 SLAM

Στην ενότητα 3.1 παρουσιάστηκαν οι αλγόριθμοι SLAM και οι μέθοδοι εκτίμησης κατάστασης που χρησιμοποιήθηκαν στην ρομποτική πλατφόρμα Monstertruck. Στην παρούσα ενότητα, παρουσιάζονται δύο διαφορετικά τμήματα που μπορούν να χρησιμοποιηθούν για την χαρτογράφηση και την εκτίμηση της θέσης του ρομπότ. Πριν προχωρήσουμε, παρόλα αυτά, στα εν λόγω τμήματα, είναι πολύ σημαντικό για την κατανόηση τους, να ορισθούν τα ακόλουθα πλαίσια αναφοράς (*reference frames*), όσον αφορά την παρούσα εφαρμογή.

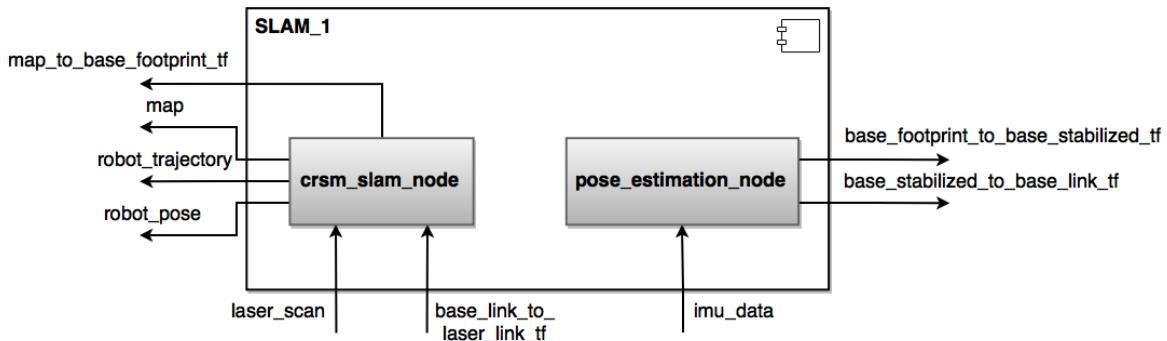
- world:** Είναι η ρίζα στο δέντρο των πλαισίων αναφοράς και αποτελεί έναν αμετάβλητο πλαίσιο αναφοράς, που στην προκειμένη περίπτωση επιλέγεται βάσει της αρχική πόζας του ρομπότ.
- map:** Είναι ένα σταθερό πλαίσιο αναφοράς ως προς το πλαίσιο world και αναπαριστά την αρχή των αξόνων του χάρτη που παράγεται από έναν αλγόριθμο SLAM. Η πόζα του ρομπότ δεν ολισθαίνει σημαντικά με τον χρόνο, με βάση το πλαίσιο αυτό, αλλά μπορεί να μεταβάλλεται σπασμωδικά.
- odom:** Είναι ένα σταθερό πλαίσιο αναφοράς ως προς το πλαίσιο map. Είναι ακατάλληλο πλαίσιο αναφοράς για μακροπρόθεσμη καθολική αναφορά, επειδή η πόζα του ρομπότ στο πλαίσιο αυτό μπορεί να ολισθαίνει χωρίς όρια, με την πάροδο του χρόνου, αλλά, παρόλα αυτά, εγγυάται συνεχή και ομαλή μεταβολή της πόζας, χωρίς διακριτές μεταβολές, σε αντίθεση με το πλαίσιο αναφοράς map.

- **base_footprint:** Αποτελεί το πλαίσιο αναφοράς του αποτυπώματος του ρομπότ στο επίπεδο, όσον αφορά το πλαίσιο αναφοράς map ή odom.
- **base_link:** Αποτελεί το πλαίσιο αναφοράς του ρομπότ, λαμβάνοντας υπόψιν την 6D κατάσταση ($x, y, z, yaw, pitch, roll$) του, όσον αφορά το πλαίσιο αναφοράς base_footprint.
- **laser_link:** Αποτελεί το πλαίσιο αναφοράς του σαρωτή λέιζερ.

Στην συνέχεια παρουσιάζονται τα δύο διαφορετικά συστήματα χαρτογράφησης και εκτίμησης κατάστασης που εξετάστηκαν στα πλαίσια της παρούσας διπλωματικής εργασίας.

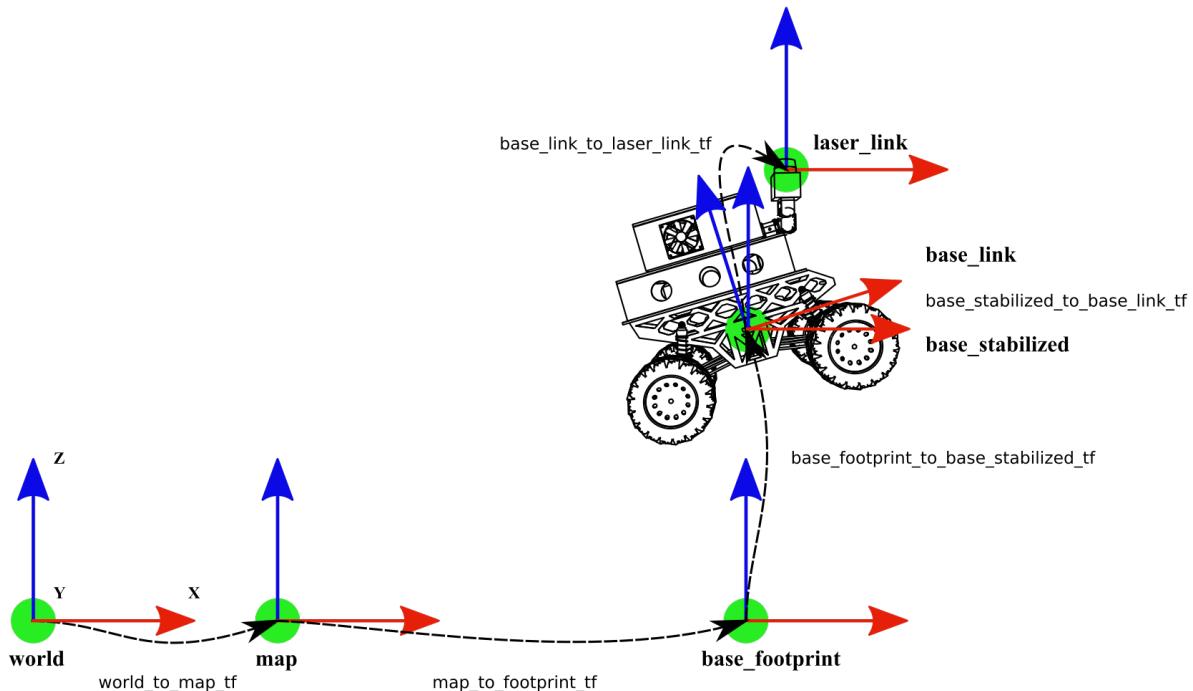
CRSM-SLAM

Το πρώτο σύστημα χαρτογράφησης και εκτίμησης κατάστασης που εξετάστηκε αποτελείται από δύο κόμβους, όπως παρουσιάζεται στο σχήμα 4.4.



Σχήμα 4.4: Διάγραμμα κόμβων του τμήματος χαρτογράφησης και εκτίμησης κατάστασης, με τον αλγόριθμο CRSM-SLAM.

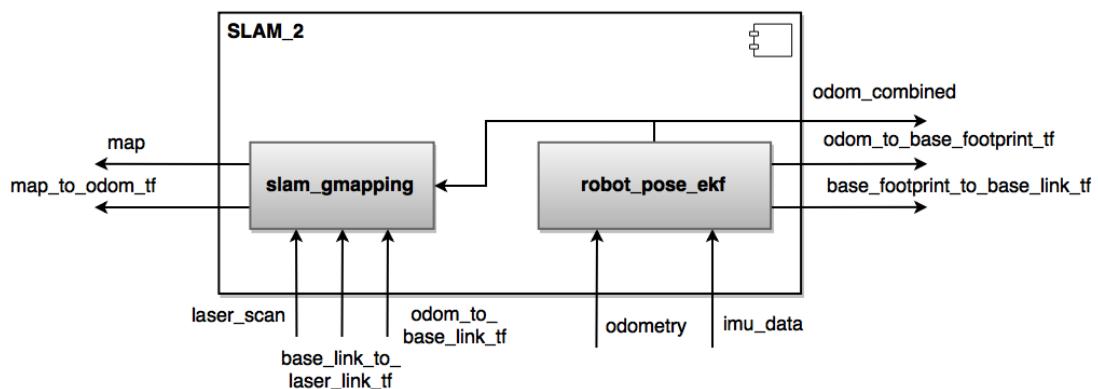
- Ο κόμβος **crsm_slam_node** του πακέτου **crsm_slam** [41] είναι υπεύθυνος για την χαρτογράφηση του περιβάλλοντος του ρομπότ και του ταυτόχρονου εντοπισμού της πόζας του, ως προς τον χάρτη που παράγει. Για την λειτουργία του, ο κόμβος αυτός, απαιτεί μηνύματα τύπου *sensor_msgs/LaserScan* με σαρώσεις του περιβάλλοντος από έναν σαρωτή λέιζερ, όπως ο Hokuyo URG-04LX που χρησιμοποιείται στην ρομποτική πλατφόρμα Monstertruck, όπως επίσης και τον μετασχηματισμό μεταξύ των πλαισίων αναφοράς *base_link* και *laser_link*. Με βάση αυτές τις πληροφορίες παράγει έναν 2D χάρτη πλέγματος κατάληψης κελιού (OGM) του περιβάλλοντος, ενώ παράλληλα εκδίδει μέσω μηνυμάτων την πόζα (x, y, θ) του ρομπότ, την τροχιά που έχει ακολουθήσει από τη στιγμή που ξεκίνησε και τον μετασχηματισμό μεταξύ των πλαισίων αναφοράς *map* και *base_footprint*.
- Ο κόμβος **pose_estimation_node** του πακέτου **pandora_pose_estimation** είναι υπεύθυνος για την έκδοση του μετασχηματισμού μεταξύ των πλαισίων αναφοράς *base_footprint* και *base_link*, όπου λαμβάνονται υπόψιν οι κλίσεις *roll*, *pitch* του ρομπότ, τις οποίες λαμβάνει από τον κόμβο του αισθητήρα Compass OS4000. Επομένως, συμπληρώνεται η κατάσταση του ρομπότ και από τις κλίσεις *roll*, *pitch* του ρομπότ, έχοντας ως αποτέλεσμα την εκτίμηση της 5D κατάστασης ($x, y, \theta, roll, pitch$) του ρομπότ, χωρίς παρόλα αυτά, να λαμβάνεται υπόψιν η θέση του ρομπότ ως προς τον κατακόρυφο άξονα *z*, λόγω έλλειψης κάποιου αλγόριθμου που να την υπολογίζει με εύρωστο τρόπο.



Σχήμα 4.5: Πλαισία αναφοράς, βάσει του συστήματος χαρτογράφησης και εκτίμησης κατάστασης με τον αλγόριθμο CRSM-SLAM.

GMapping και State Estimation με EKF

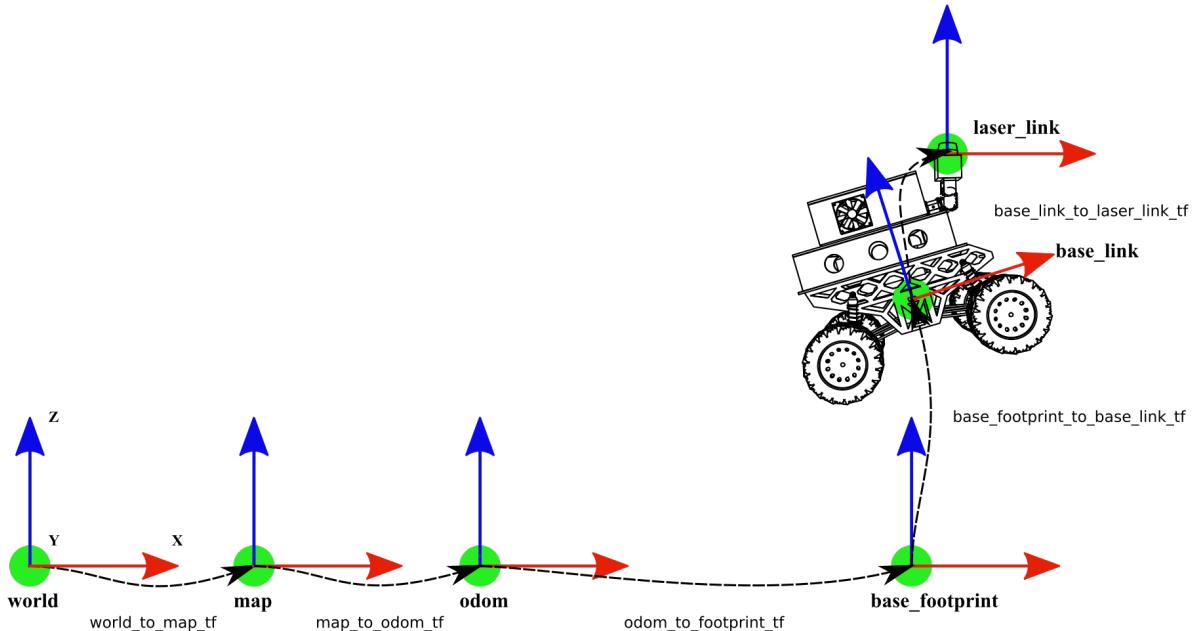
Το δεύτερο σύστημα χαρτογράφησης και εκτίμησης κατάστασης που εξετάστηκε αποτελείται, επίσης από δύο κόμβους, όπως παρουσιάζεται στο σχήμα 4.6.



Σχήμα 4.6: Διάγραμμα κόμβων του τμήματος χαρτογράφησης και εκτίμησης κατάστασης, με τον αλγόριθμο GMapping και με EKF.

- Ο κόμβος `slam_gmapping` του πακέτου `gmapping` [42] είναι μία υλοποίηση του αλγορίθμου Gmapping που αναφέρθηκε στην ενότητα 3.1.3, συμβατή με το ROS. Ο κόμβος αυτός, λαμβάνει σαρώσεις λέιζερ του περιβάλλοντος, την οδομετρία του οχήματος, όπως επίσης και τους μετασχηματισμούς μεταξύ των πλαισίων αναφοράς `base_link`, `laser_link` και `odom`, `base_link` και δημιουργεί τον χάρτη του περιβάλλοντος, όπως επίσης και τον μετασχηματισμό μεταξύ των πλαισίων αναφοράς `map` και `odom`.

- Ο κόμβος **robot_pose_ekf** του πακέτου **robot_pose_ekf** [43] χρησιμοποιεί ένα EKF με ένα 6D μοντέλο (3D θέσης και 3D προσανατολισμός) για να συνδυάσει τις μετρήσεις της οδομετρίας των τροχών και της πυξίδας, ενώ δίνει την δυνατότητα συμπερίληψης και μετρήσεων οπτικής οδομετρίας, με απότερο στόχο την παραγωγή μίας πιο εύρωστης αντίληψης για την 6D κατάσταση του ρομπότ, σε σύγκριση με εκτιμήσεις από μεμονωμένες πηγές πληροφορίας. Στην προκειμένη περίπτωση, λαμβάνει, μέσω μηνυμάτων ROS, την οδομετρία των τροχών και τις μετρήσεις της πυξίδας και εκδίδει την συνδυασμένη οδομετρία και τους μετασχηματισμούς μεταξύ των πλαισίων αναφοράς odom, base_footprint και base_link, base_link.



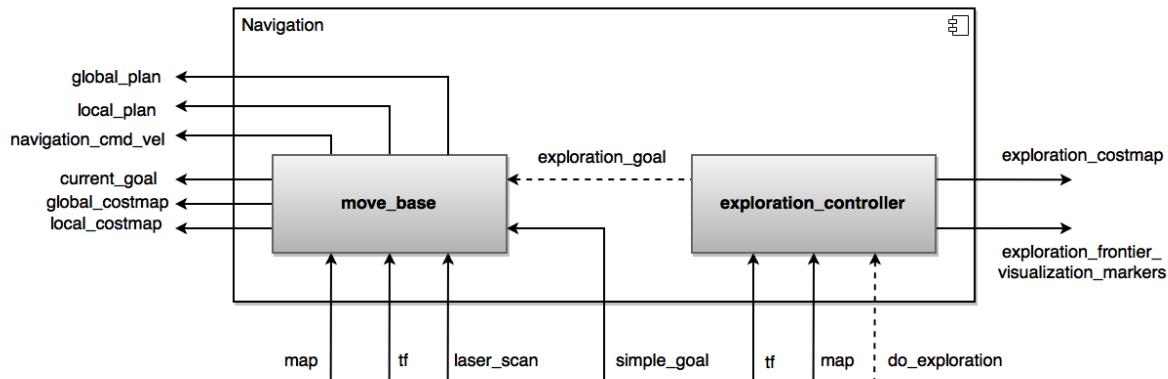
Σχήμα 4.7: Πλαισία αναφοράς, βάσει του συστήματος χαρτογράφησης και εκτίμησης κατάστασης με τον αλγόριθμο Gmapping και με EKF.

4.2.4 Σύστημα Αυτόνομης Πλοήγησης

Το σύστημα αυτόνομης πλοήγησης της ρομποτικής πλατφόρμας Monstertruck, όπως παρουσιάζεται στο σχήμα 4.8, αποτελείται, βασικά, από τον κόμβο **move_base** [44], ο οποίος είναι υπεύθυνος για την πλοήγηση του οχήματος, με στόχο την αυτόνομη μετάβαση του από μία αρχική θέση σε μία τελική, αλλά και έναν προαιρετικό κόμβο **exploration_controller** του πακέτου **pandora_explorer** ο οποίος χρησιμοποιείται για την εξερεύνηση του περιβάλλοντος, μέσω της παραγωγής διαδοχικών στόχων, σε αντίθεση με την επιλογή στόχων από τον χειριστή/επόπτη του ρομπότ.

Ο κόμβος **exploration_controller**, χρησιμοποιεί έναν αλγόριθμο επιλογής στόχων [45], με σκοπό την πλήρη εξερεύνηση του περιβάλλοντος. Χρησιμοποιεί σαν είσοδο την τρέχουσα πόζα του ρομπότ και μία αναπαράσταση του περιβάλλοντος, στη προκειμένη περίπτωση τον χάρτη πλέγματος κατάληψης, βάσει των οποίων αναζητά μέτωπα προς εξερεύνηση. Ως μέτωπα εξερεύνησης επιλέγονται τα σύνορα μεταξύ εξερευνημένου και ανεξερεύνητου χώρου, ενώ η επιλογή του τρέχοντος μετώπου εξερεύνησης πραγματοποιείται, λαμβάνοντας υπόψιν διάφορες συναρτήσεις κόστους μετρικών, όπως το μέγεθος του συνόρου, το μήκος του μονοπατιού για μετάβαση στο σύνορο αυτό, την γωνιακή απόκλιση μεταξύ του ρομπότ και του συνόρου και την συχνότητα επιλογής αυτού.

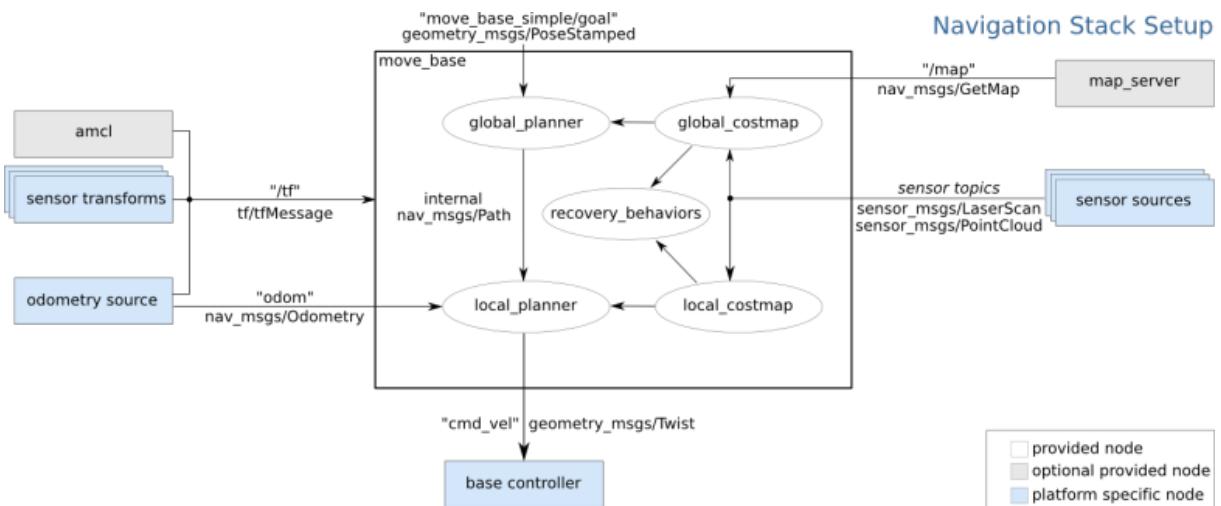
Ο κόμβος **move_base** αποτελεί έναν κόμβο ο οποίος, δοσμένου ενός στόχου, προσπαθεί να πλοηγήσει το ρομπότ προς τον στόχο αυτόν. Όπως παρουσιάζεται στο σχήμα 4.9, αυτό το πετυχαίνει



Σχήμα 4.8: Το τμήμα αυτόνομης πλοιάργησης της ρομποτικής πλατφόρμας Monstertruck.

συνδυάζοντας έναν *global_planner* για την κατασκευή ενός ολικού μονοπατιού που βρίσκεται εξ ολοκλήρου στον ελεύθερο χώρο και συνδέει το ρομπότ με τον στόχο και έναν *local_planner* ο οποίος προσπαθεί να ακολουθήσει το ολικό μονοπάτι, αποφεύγοντας ταυτόχρονα τυχόν δυναμικά εμπόδια και γενικότερα απρόβλεπτες καταστάσεις, λαμβάνοντας υπόψιν κινηματικούς και δυναμικούς περιορισμούς του ρομπότ. Ο κόμβος *move_base*, επίσης, δημιουργεί και διαχειρίζεται δύο 2D χάρτες κόστους (*costmap_2d*), τον *global_costmap*, που αποτελεί μία αναπαράσταση του ολικού τρέχοντος χάρτη, και χρησιμοποιείται από τον *global_planner* για την κατασκευή του ολικού μονοπατιού (*global_plan*), όπως επίσης και τον *local_costmap*, που αναπαριστά μία τοπική αναπαράσταση του τρέχοντος χάρτη, με κέντρο το ρομπότ, το οποίο χρησιμοποιείται από τον *local_planner*, για την κατασκευή του τοπικού μονοπατιού (*local_plan*). Επίσης, ο κόμβος *move_base* υποστηρίζει ρουτίνες επαναφοράς (*recovery behaviors*) σε περίπτωση που ο *global_planner* ή ο *local_planner* αποτύχουν να βρουν μία εφικτή λύση.

Το μεγάλο πλεονέκτημα του κόμβου *move_base* αποτελεί η ευελιξία του και η προσαρμοστικότητα του, όσον αφορά την δυνατότητα χρήσης διαφορετικών αλγορίθμων για κατασκευή μονοπατιών, αποφυγή εμποδίων, επαναφορά από αποτυχία κοκ, μέσω προσθέτων (plugins), που αποτελεί και έναν τους λόγους που επιλέχθηκε για την υλοποίηση του συστήματος αυτόνομης πλοιάργησης της ρομποτικής πλατφόρμας Monstertruck.



Σχήμα 4.9: Ο κόμβος *move_base* με τις βασικές απαιτούμενες διεπαφές [44].

Στην συνέχεια, παρουσιάζονται πιο αναλυτικά οι 2D χάρτες κόστους, οι ρουτίνες επαναφοράς, όπως επίσης και τα δύο συστήματα αυτόνομης πλοιάργησης που υλοποιήθηκαν.

2D Χάρτες Κόστους

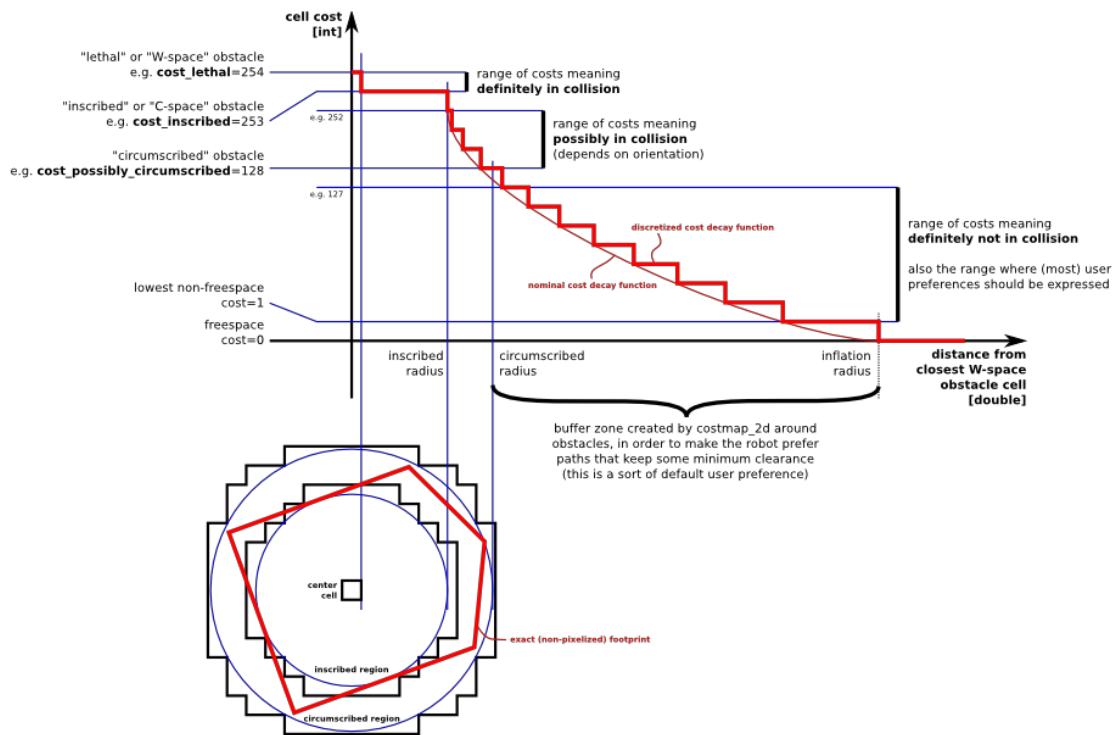
Ο κόμβος move_base, όπως προαναφέρθηκε χρησιμοποιεί 2D χάρτες κόστους [46], οι οποίοι παράγονται, βάσει του χάρτη πλέγματος κατάληψης κελιού ενός αλγορίθμου SLAM και συντίθενται από έναν αριθμό επιπέδων. Στην προκειμένη περίπτωση, χρησιμοποιούνται τρία επίπεδα, όπου το *στατικό επίπεδο (static layer)* αναπαριστά τα αμετάβλητα τμήματα ενός χάρτη κόστους, όπως αυτά παράγονται από έναν αλγόριθμο SLAM, το *επίπεδο των εμποδίων (obstacle layer)* αναπαριστά τα εμπόδια όπως αυτά αναγνωρίζονται από μετρήσεις του σαρωτή λέιζερ και το *επίπεδο διαστολής (inflation layer)* που μεταδίδει τα κόστη ακτινικά των εμποδίων, όπως αναφέρθηκε και στην ενότητα 3.2, προσπαθώντας να μετατρέψει τον χάρτη κόστους στο χώρο εφικτών καταστάσεων του ρομπότ. Στο σχήμα 4.10 παρουσιάζονται τα διάφορα κόστη ενός χάρτη κόστους, τα οποία ορίζονται ακολούθως, αν λάβουμε το κόστος ως έναν μη προσημασμένο ακέραιο με 8bit.

- **cost_unknown(255):** Είναι το κόστος των κελιών, για τα οποία δεν είναι γνωστή η κατάσταση κατάληψης.
- **cost_lethal(254):** Είναι το κόστος των κελιών που ανήκουν σε εμπόδια.
- **cost_inscribed(253):** Είναι το κόστος των κελιών στα οποία το ρομπότ έρχεται σε σύγκρουση και έχουν απόσταση από τα κελιά των εμποδίων, ίση με την ακτίνα του κύκλου που εγγράφεται στο αποτύπωμα του ρομπότ.
- **cost_possibly_circumscribed(128):** Είναι το κόστος των κελιών που η απόσταση τους από τα κελιά των εμποδίων ισούται με την ακτίνα του περιγεγραμμένου, στο αποτύπωμα του ρομπότ, κύκλου. Κόστη που ανήκουν στο διάστημα [128, 252] δηλώνουν πιθανή σύγκρουση που καθορίζεται από τον προσανατολισμό ρομπότ.
- **cost_safe ∈ [128, 0):** Κόστος κελιών που μπορούν να προσπελαστούν από το ρομπότ χωρίς κίνδυνο για σύγκρουση. Η απόσταση τους από τα κελιά των εμποδίων μπορεί να οριστεί από τον χρήστη.
- **cost_freespace (0):** Είναι το κόστος των ελεύθερων κελιών που δεν εισάγουν κανένα περιορισμό προσπέλασης από το ρομπότ και θα πρέπει γενικά να προτιμάται, κατά την κατασκευή μονοπατιών και την πλοιήγηση του ρομπότ.

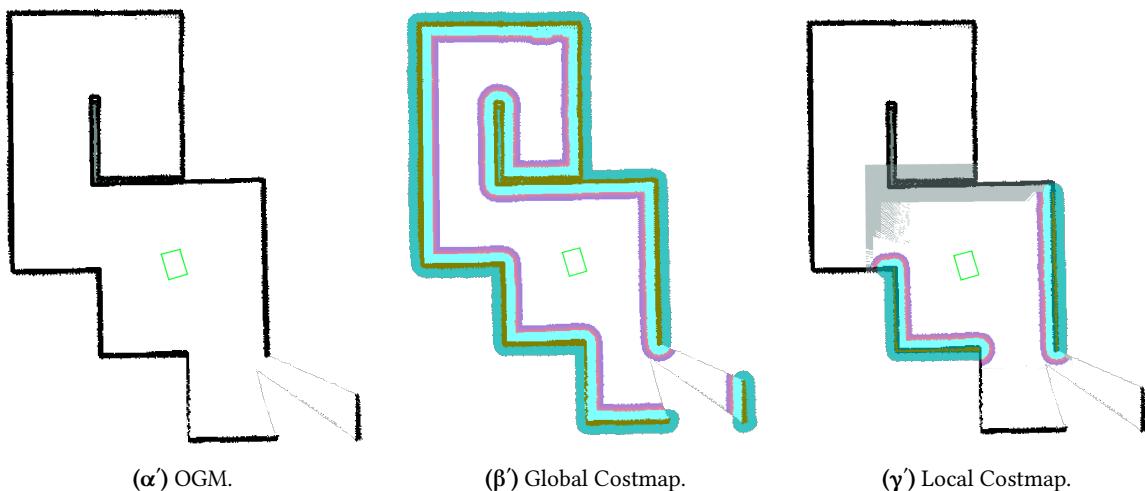
Ρουτίνες Επαναφοράς

Ένα πολύ σημαντικό τμήμα του κόμβου move_base αποτελεί μία ακολουθία ρουτινών επαναφοράς που εκτελούνται σε περίπτωση που το ρομπότ έχει κολλήσει, δηλαδή έχει αποτύχει ο global_planner ή ο local_planner να παράξουν κάποιο μονοπάτι ή τροχιά για την επίτευξη του τρέχοντα στόχου. Οι ρουτίνες επαναφοράς στοχεύουν στον καθαρισμό του χώρου περιμετρικά του ρομπότ, έτσι ώστε να μπορεί να ξεκολλήσει. Αρχικά, λοιπόν, πραγματοποιείται καθαρισμός των εμποδίων έξω από μία περιοχή γύρω από το ρομπότ, που ορίζεται αυθαίρετα. Έπειτα, ακολουθεί μία επιτόπου περιστροφή του ρομπότ με στόχο τον καθαρισμό τυχόν εσφαλμένων εμποδίων ή άγνωστου χώρου που εμφανίζονται στον χάρτη κόστους. Αν η ρουτίνα αυτή αποτύχει, τότε επαναλαμβάνεται η διαδικασία, αλλά για την ελάχιστη δυνατή περιοχή που επιτρέπει την επί τόπου περιστροφή του ρομπότ. Εάν, παρόλα αυτά, και αυτή η ρουτίνα αποτύχει, τότε ο στόχος ορίζεται σαν μη εφικτός.

Όπως είναι προφανές, η παραπάνω ακολουθία ρουτινών επαναφοράς δεν είναι εφικτές για ένα μη ολονομικό όχημα, που δεν μπορεί να πραγματοποιήσει επί τόπου στροφές, όπως για παράδειγμα η υλοποιημένη ρομποτική πλατφόρμα Monstertruck. Επομένως, επιλέχθηκε η ανάπτυξη μίας ρουτίνας επαναφοράς που προσπαθεί να ξεφύγει από ανεπιθύμητες καταστάσεις, πραγματοποιώντας κινηματικά εφικτούς ελιγμούς.



Σχήμα 4.10: Τα κόστη του επιπέδου διαστολής του 2D χάρτη κόστους [46].



Σχήμα 4.11: Ο χάρτης πλέγματος κατάληψης κελιού που παράχθηκε από τον αλγόριθμο CRSMSLAM, για το περιβάλλον roboscup του προσωμοιωτή STDR και οι αντίστοιχοι χάρτες κόστους.

Η κυριότερη αιτία αποτυχίας κατασκευής ολικού ή τοπικού μονοπατιού, αποτελεί συνήθως η πολύ κοντινή απόσταση του ρομπότ σε εμπόδια. Για το λόγο αυτό, αναπτύχθηκε η ρουτίνα επαναφοράς `car_maneuver_recovery` που προσπαθεί μέσω κινηματικά εφικτών ελιγμών να απομακρύνει το ρομπότ από εμπόδια. Η εν λόγω ρουτίνα επαναφοράς υποστηρίζει κινήσεις, βάσει κινηματικού Ackermann, θετικής τετραδιεύθυνσης και αρνητικής τετραδιεύθυνσης και λειτουργεί όπως παρουσιάζεται στον Αλγόριθμο 1.

Algorithm 1 Αλγόριθμος Ρουτίνας Επαναφοράς car_maneuver_recovery

Input: frontSideCost, rearSideCost, leftSideCost, rightSideCost

Output: speed, frontSteeringAngle, rearSteeringAngle

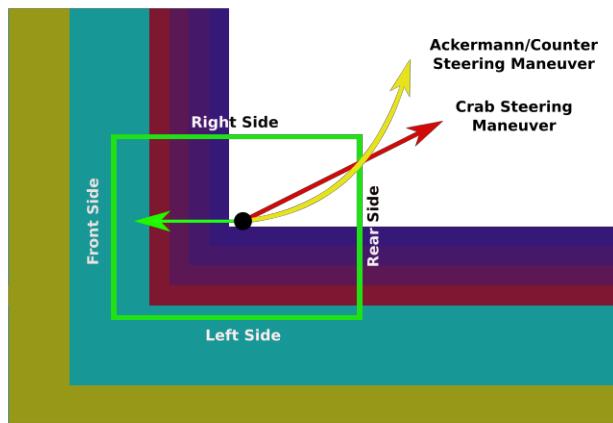
```

1: front ← frontSideCost < cost_circumscribed
2: rear ← rearSideCost < cost_circumscribed
3: left ← leftSideCost < cost_circumscribed
4: right ← rightSideCost < cost_circumscribed
5: if front && right then
6:   if frontSideCost > rearSideCost then
7:     speed ← recoverySpeed
8:   else
9:     speed ← -recoverySpeed
10:  end if
11: else
12:   speed ← (front - rear) × recoverySpeed
13: end if
14: frontSteeringAngle ← (left-right) × recoverySteeringAngle
15: if fourWheelSteering then
16:   if crabSteering then
17:     rearSteeringAngle ← frontSteeringAngle
18:   else
19:     rearSteeringAngle ← - frontSteeringAngle
20:   end if
21: else
22:   rearSteeringAngle ← 0
23: end if

```

▷ Four Wheel Steering
▷ Positive Four Wheel Steering
▷ Negative Four Wheel Steering
▷ Ackermann Steering

όπου τα κόστη frontSideCost, rearSideCost, leftSideCost και rightSideCost υπολογίζονται ως ο μέσος όρος των κοστών της αντίστοιχης πλευράς του αποτυπώματος του ρομπότ στον τοπικό χάρτη κόστους και οι μεταβλητές fourWheelSteering και crabSteering δηλώνουν την προτίμηση του χρήστη για τον ελιγμό που θα πραγματοποιεί η ρουτίνα επαναφοράς.



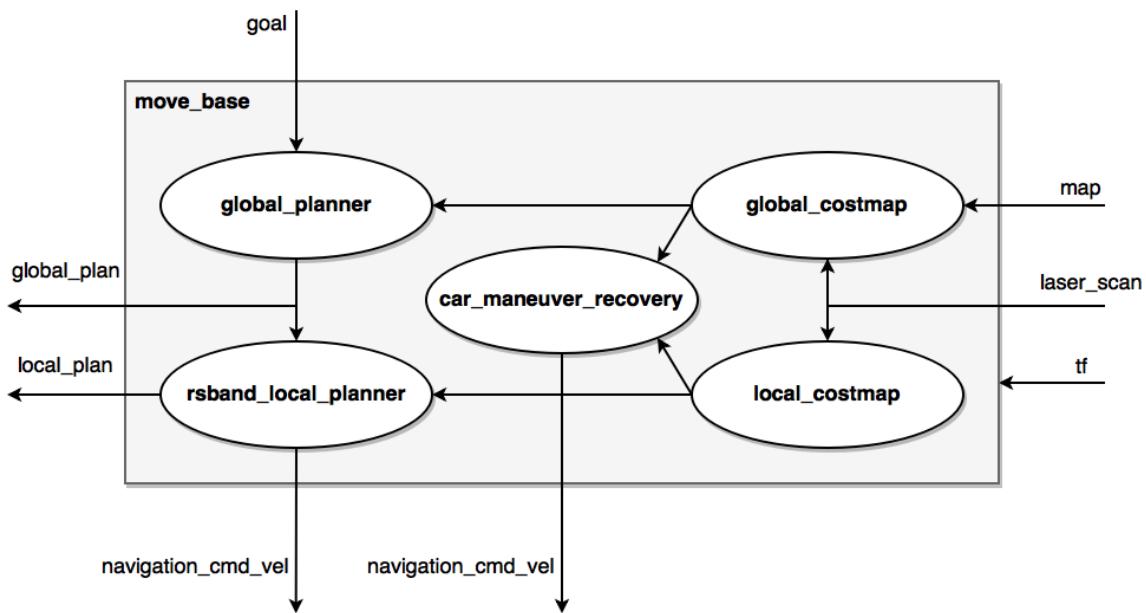
Σχήμα 4.12: Παράδειγμα εκτέλεσης ρουτίνας επαναφοράς του αλγορίθμου car_maneuver_recovery.

Στην συνέχεια, παρουσιάζονται τα δύο συστήματα αυτόνομης πλοήγησης που χρησιμοποιήθηκαν, με βάση τον κόμβο move_base, τα οποία διαφοροποιούνται όσον αφορά την αρχιτεκτονική και

την συνολική προσέγγιση επίλυσης του προβλήματος, μέσω της επιλογής διαφορετικών ζευγών αλγορίθμων local_planner και global_planner.

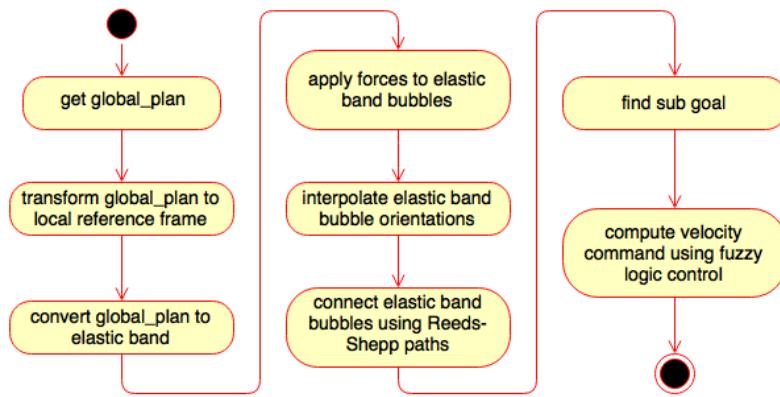
Σύστημα Αυτόνομης Πλοήγησης με Δυναμική Παραμόρφωση Μονοπατιού

Το πρώτο σύστημα αυτόνομης πλοήγησης που υλοποιήθηκε με βάση τον κόμβο move_base, αποτελείται από έναν global planner ο οποίος κατασκευάζει ένα στατικό ολικό μονοπάτι στον ελεύθερο χώρο του ολικού χάρτη κόστους (global costmap), βάσει των αλγορίθμων A* ή Dijkstra και από έναν local planner που παίρνει το ολικό μονοπάτι και παραμορφώνει το τμήμα αυτού που ανήκει στον τοπικό χάρτη κόστους με στόχο την παραγωγή ενός τοπικού, κινηματικά εφικτού και ασφαλούς μονοπατιού, βάσει του αλγορίθμου Reeds-Sheep Band, που παρουσιάστηκε στην ενότητα 3.2.2.



Σχήμα 4.13: Ο κόμβος move_base με τα plugins του συστήματος αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού.

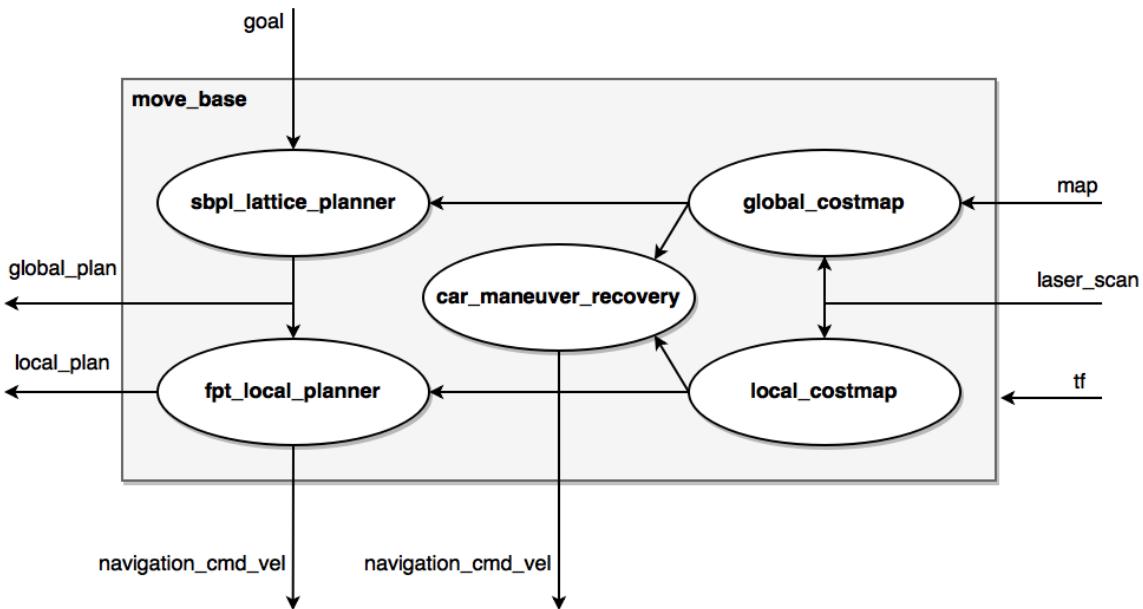
- Το plugin **global_planner** του Lu [47] είναι ένα plugin του κόμβου move_base, το οποίο υλοποιεί τους αλγορίθμους αναζήτησης μονοπατιού A* και Dijkstra που αναφέρθηκαν στην ενότητα 3.2.1, με στόχο την εύρεση ενός μονοπατιού που βρίσκεται εξ ολοκλήρου στον ελεύθερο χώρο του ολικού χάρτη κόστους και ενώνει την αρχική θέση του ρομπότ με τον στόχο.
- Το plugin **rsband_local_planner**, που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, υλοποιεί τον αλγόριθμο Reeds-Sheep Band, που παρουσιάστηκε στην ενότητα 3.2.2, σε συνδυασμό με τον ελεγκτή διάσχισης μονοπατιού ασαφούς λογικής που παρουσιάστηκε στην ενότητα 3.2.3. Για την υλοποίηση του εν λόγω αλγορίθμου χρησιμοποιήθηκε η υλοποίηση του αλγορίθμου ελαστικής ζώνης των Connnette et al. [48] και η υλοποίηση του χώρου καταστάσεων Reeds-Shepp (Reeds-Shepp State Space) του Moll [49] της βιβλιοθήκης OMPL (Open Motion Planning Library) [50]. Επίσης, για την ανάπτυξη του αλγορίθμου διάσχισης μονοπατιού, που βασίζεται σε ασαφή λογική, χρησιμοποιήθηκε η βιβλιοθήκη ελέγχου ασαφούς λογικής fuzzylite [51] του Rada-Vilela. Στο σχήμα 4.14 παρουσιάζεται η βασική λειτουργικότητα του plugin rsband_local_planner, μέσω ενός προσεγγιστικού διαγράμματος δραστηριοτήτων.



Σχήμα 4.14: Το διάγραμμα δραστηριοτήτων του plugin rsband_local_planner.

Σύστημα Αυτόνομης Πλοήγησης με Δυναμική Επανακατασκευή Μονοπατιού

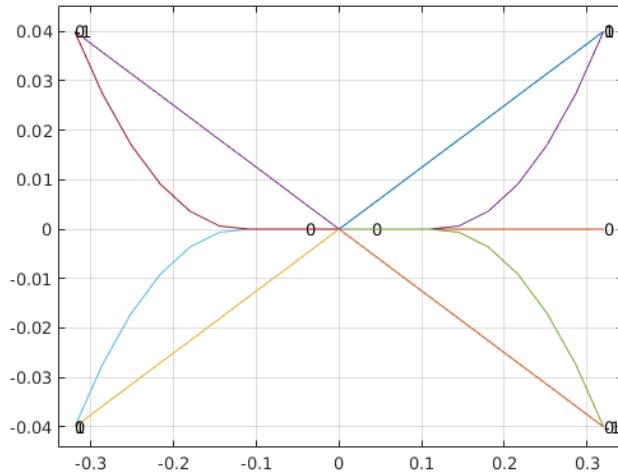
Το δεύτερο σύστημα αυτόνομης πλοήγησης που υλοποιήθηκε, με βάση τον κόμβο move_base χρησιμοποιεί έναν global planner για την κατασκευή ενός ολικού κινηματικά εφικτού μονοπατιού με δυναμική ανακατασκευή, αλλά και έναν local planner που απλά χρησιμοποιείται για την διάσχιση του ολικού μονοπατιού.



Σχήμα 4.15: Ο κόμβος move_base με τα plugins του συστήματος αυτόνομης πλοήγησης με δυναμική ανακατασκευή μονοπατιού.

- Για την κατασκευή του ολικού μονοπατιού του συστήματος αυτόνομης πλοήγησης με δυναμική ανακατασκευή μονοπατιού χρησιμοποιείται το plugin sbpl_lattice_planner, το οποίο αναπτύχθηκε από τον Phillips[52] και βασίζεται στην βιλβιοθήκη SBPL [53] του Search-Based Planning Lab. Όπως, παρουσιάστηκε και στην ενότητα 3.2.1, ο αλγόριθμος παράγει ένα μονοπάτι που αποτελείται από τον συνδυασμό ενός συνόλου βασικών κινήσεων (motion primitives), οι

οποίες αποτελούν μικρές κινηματικά εφικτές κινήσεις. Η κατασκευή του μονοπατιού, πραγματοποιείται μέσω των αλγορίθμων *ARA** ή *AD** στον χώρο (x, y, θ) και επομένως λαμβάνει υπόψιν τον προσανατολισμό του ρομπότ, σε αντίθεση με το σύστημα αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού, όπου η αναζήτηση γίνεται στον χώρο (x, y) .

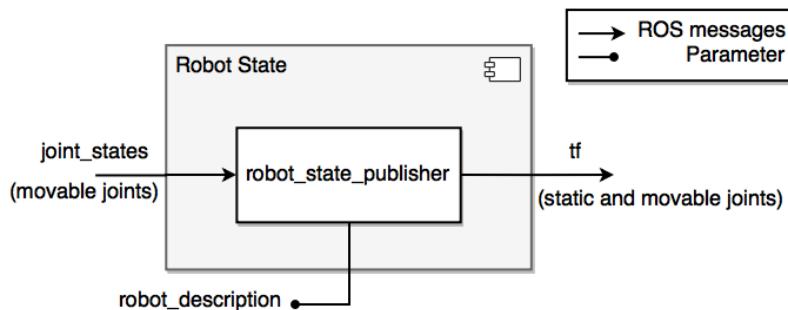


Σχήμα 4.16: Οι βασικές κινήσεις που ορίστηκαν για χρήση στον `sbl_lattice_planner`, για την κατασκευή μονοπατιών της ρομποτικής πλατφόρμας Monstertruck.

- Αφού κατασκευαστεί το κινηματικά εφικτό ολικό μονοπάτι, με βάση τον παραπάνω αλγόριθμο, χρησιμοποιείται το plugin `fpt_local_planner`, ως ο `local_planner` του κόμβου `move_base`, ο οποίος υλοποιεί τον ασαφή ελεγκτή διάσχισης μονοπατιού που παρουσιάστηκε στην ενότητα 3.2.3. Ο ασαφής ελεγκτής δεν πραγματοποιεί αποφυγή εμποδίων, αλλά απλή διάσχιση μονοπατιού και επομένως, η αποφυγή εμποδίων επιτυγχάνεται μέσω της δυναμικής ανακατασκευής του ολικού μονοπατιού.

4.2.5 Robot State

Το τμήμα του Robot State αποτελείται από τον κόμβο `robot_state_publisher` [54], ο οποίος χρησιμοποιεί το κινηματικό μοντέλο του ρομπότ, το οποίο φορτώνεται από τον διακομιστή παραμέτρων (parameter server) και βάσει των θέσεων των κινούμενων αρθρώσεων του ρομπότ υπολογίζει και εκδίδει τους μετασχηματισμούς μεταξύ όλων των αρθρώσεων, στατικών και δυναμικών. Για την λειτουργία του, απαιτεί την περιγραφή του μοντέλου του ρομπότ σε μορφή URDF (Unified Robot Description Format), όπως επίσης και κόμβους που εκδίδουν τις θέσεις των κινούμενων αρθρώσεων του.



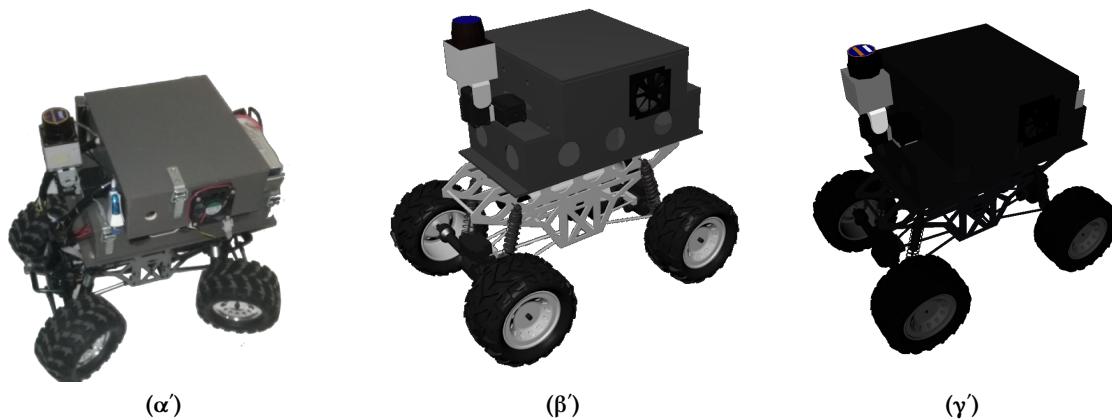
Σχήμα 4.17: Το τμήμα Robot State της ρομποτικής πλατφόρμας Monstertruck.

Μέσω του URDF είναι δυνατόν να μοντελοποιηθεί ένα ρομπότ σαν ένα σύνολο άκαμπτων συνδέσμων και αρθρώσεων μεταξύ των συνδέσμων αυτών, σε τοπολογία με δενδρική δομή που απαγορεύει κλειστούς βρόχους. Παρόλα αυτά, προσφέρει δυνατότητες για κινηματική και δυναμική περιγραφή του ρομπότ, οπτική αναπαράσταση των συνδέσμων του, όπως επίσης και περιγραφή του μοντέλου σύγκρουσης αυτών.

Στα πλαίσια της παρούσας εργασίας, αναπτύχθηκε ένα λεπτομερές αλλά προσεγγιστικό οπτικά και διαστασιολογικά μοντέλο της ρομποτικής πλατφόρμας Monstertruck σε μορφή URDF. Αρχικά, πραγματοποιήθηκε η ανάπτυξη ενός τρισδιάστατου στατικού μοντέλου, μέσω του προγράμματος *Blender*² και έπειτα, βάσει αυτού αναπτύχθηκε το αντίστοιχο μοντέλο σε μορφή URDF, όπου συμπεριλήφθηκαν και οι αρθρώσεις του ρομπότ. Συνολικά ορίστηκαν δεκαοκτώ κινούμενες αρθρώσεις, όπως αυτές παρουσιάζονται στον πίνακα 4.1.

Πίνακας 4.1: Οι αρθρώσεις του μοντέλου URDF της ρομποτικής πλατφόρμας Monstertruck.

Αρθρώσεις	Τύπος	Όρια	Πλήθος
άρθρωση ώθησης τροχού	περιστροφική	-	4
άρθρωση στρέψης τροχού	περιστροφική	[−25°, 25°]	4
ανάρτηση τροχού	πρισματική	[−0.1, 0.3]	4
εικονική άρθρωσης άξονα τροχών	περιστροφική	[−10°, 10°]	2
εικονική ανάρτησης άξονα τροχών	πρισματική	[−0.1, 0.3]	2
άρθρωση roll σαρωτή λέιζερ	περιστροφική	[−30°, 30°]	1
άρθρωση pitch σαρωτή λέιζερ	περιστροφική	[−30°, 30°]	1

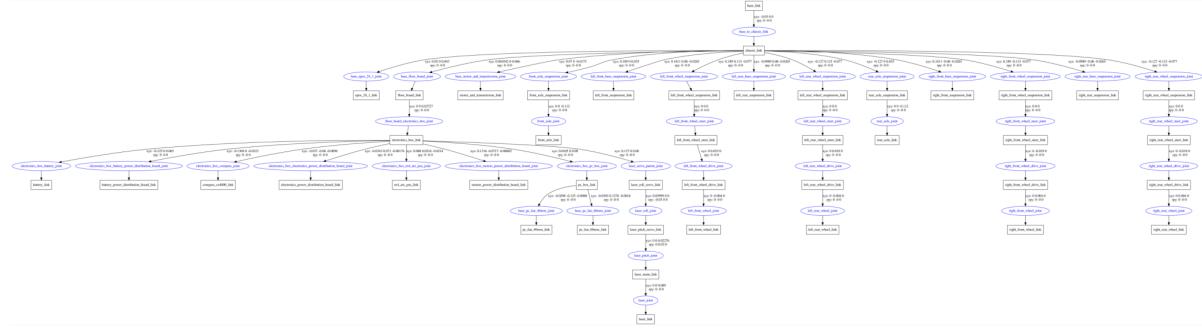


Σχήμα 4.18: Η ρομποτική πλατφόρμα Monstertruck σε (α') φωτογραφία, (β') 3D μοντέλο και (γ') URDF.

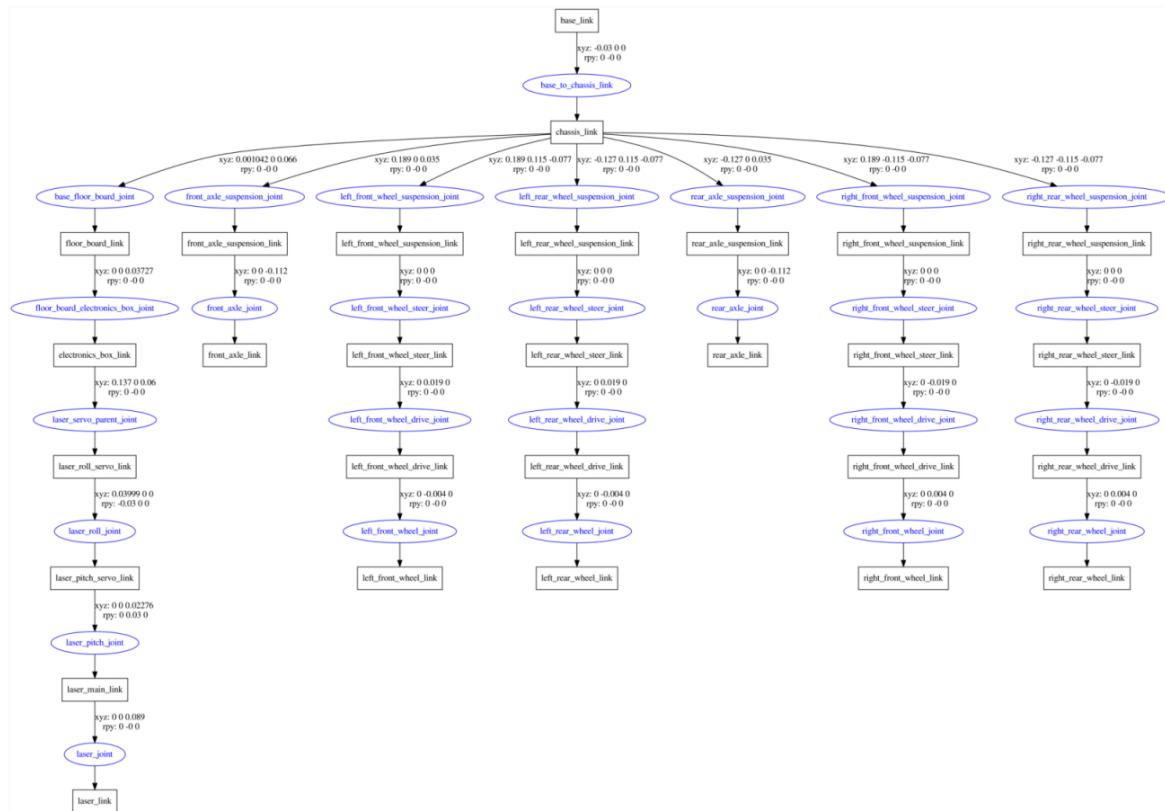
Η θέση και η ταχύτητα των αρθρώσεων ώθησης και στρέψης των τροχών, όπως επίσης και των αρθρώσεων σταθεροποίησης του σαρωτή λέιζερ παρέχονται από τους αντίστοιχους κόμβους του τμήματος Software/Hardware Interface 4.2.1. Παρόλα αυτά, οι αρθρώσεις των αναρτήσεων και των αξόνων των μπροστινών και πίσω τροχών, μοντελοποιήθηκαν για την πιο ρεαλιστική προσομοίωση του ρομπότ, αλλά δεν έχουν καμία χρησιμότητα στο πραγματικό ρομπότ, καθώς δεν υπάρχει κάποιος τρόπος μέτρησης της θέσης τους. Για την προσομοίωση, παρόλα αυτά, η μοντελοποίηση τους πραγματοποιήθηκε με κάποιες προσεγγίσεις. Συγκεκριμένα, η μοντελοποίηση των αναρτήσεων του ρομπότ προσεγγίστηκε μέσω πρισματικών αρθρώσεων με πεπερασμένα όρια δυνάμεων επιβολής της προκαθορισμένης θέσης τους, με αποτέλεσμα, όταν ασκούνται μεγαλύτερες δυνάμεις να ισορροπούν

²www.blender.org

σε διαφορετική θέση. Επίσης, εφόσον το μοντέλο ορίζεται σε δενδρική δομή, που απαγορεύει βρόχους, η κίνηση του μπροστινού και του πίσω αξόνα των τροχών προσεγγίσθηκε χρησιμοποιώντας πρισματικές αρθρώσεις για τις κατακόρυφες κινήσεις των αξόνων σε συνδυασμό με περιστροφικές αρθρώσεις για την περιστροφή τους, συναρτήσει της κίνησης των αναρτήσεων των τροχών.



Σχήμα 4.19: Πλήρης δενδρική δομή των συνδέσμων και αρθρώσεων της ρομποτικής πλατφόρμας Monstertruck.

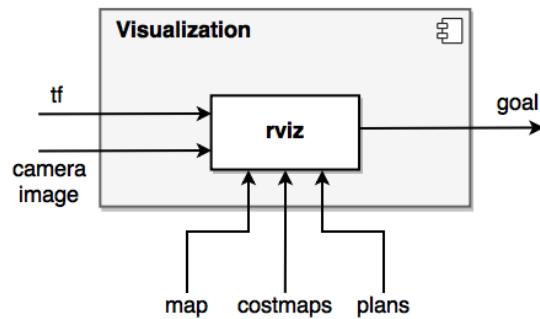


Σχήμα 4.20: Μερική δενδρική δομή των σημαντικότερων συνδέσμων και αρθρώσεων της ρομποτικής πλατφόρμας Monstertruck.

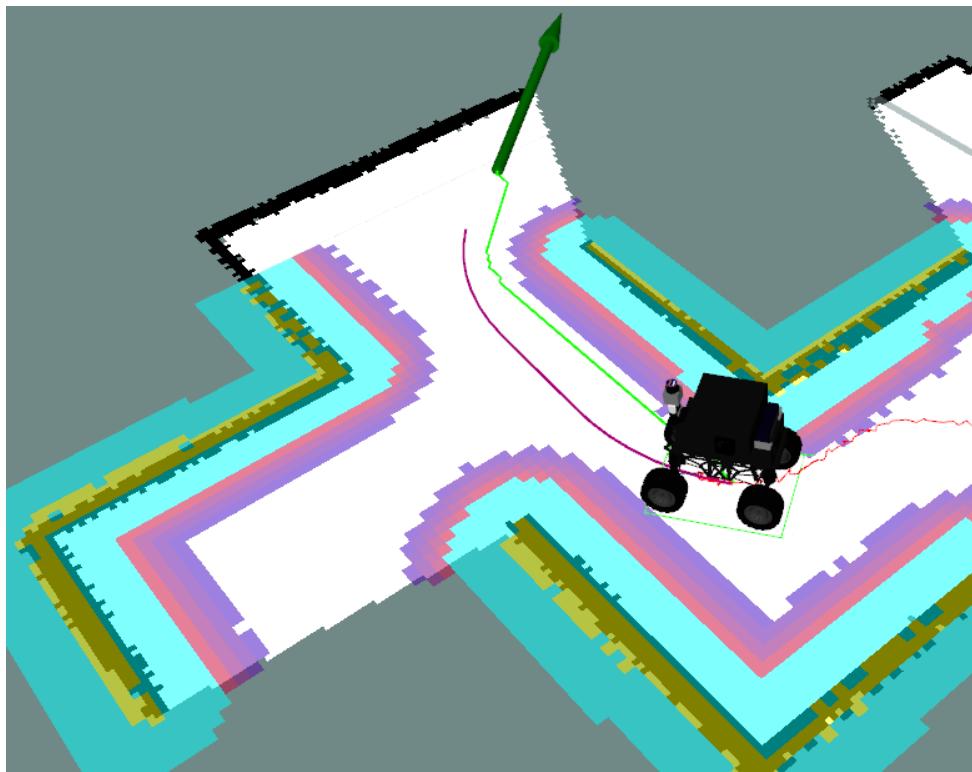
4.2.6 Visualization

Το τμήμα Visualization της ρομποτικής πλατφόρμας Monstertruck αποτελείται από τον κόμβο rviz [55], ο οποίος αποτελεί ένα εργαλείο τρισδιάστατης απεικόνισης για το ROS, με στόχο την απεικόνιση δεδομένων του ρομπότ, όπως μετρήσεις αισθητήρων, χάρτες, το μοντέλο URDF του ρομπότ,

ολικά και τοπικά μονοπάτια, στόχους, τροχιά του ρομπότ, εικόνα από κάμερα κοκ, με αποτέλεσμα την δυνατότητα οπτικής επίβλεψης και εντοπισμού προβλημάτων κατά την ανάπτυξη του λογισμικού του ρομπότ, αλλά και κατά την λειτουργία αυτού.



Σχήμα 4.21: Το τμήμα Visualization της ρομποτικής πλατφόρμας Monstertruck.



Σχήμα 4.22: Περιβάλλον τρισδιάστατης απεικόνισης από τον κόμβο rviz, όπου παρουσιάζεται το μοντέλο του ρομπότ, ο χάρτης, οι χάρτες κόστους, ο στόχος (πράσινο βέλος) το ολικό μονοπάτι(πράσινο), το τοπικό μονοπάτι(μωβ) και η τρέχουσα τροχιά που έχει διασχιστεί(κόκκινο).

4.3 Εργαλεία Προσομοίωσης

Η ανάπτυξη αλγορίθμων και εφαρμογών για ρομποτικά συστήματα αποτελεί μία ιδιαίτερα σύνθετη, χρονοβόρα και απαιτητική διαδικασία που καθίσταται ακόμα πιο δύσκολη εάν υπάρχει έλλειψη των απαραίτητων εργαλείων και εξοπλισμού ή εάν χρησιμοποιούνται φυσικά συστήματα, τα οποία απαιτούν συνεχή παρακολούθηση και συντήρηση, τα οποία παράλληλα είναι επιρρεπείς σε ζημιές. Για παράδειγμα, η ανάπτυξη εφαρμογών αυτόνομης πλοϊγησης για ρομποτικά οχήματα, μπορεί να

αποτελέσει αρκετά επικίνδυνη διαδικασία, για το ίδιο το ρομπότ και τον εξοπλισμό του, αλλά και για το φυσικό περιβάλλον του, σε περίπτωση εσφαλμένης συμπεριφοράς των αλγορίθμων, περιορίζοντας έτσι σημαντικά τα περιθώρια δυνατών δοκιμών, πάνω σε αυτό. Επίσης, η ανάπτυξη των εφαρμογών απαιτεί συνεχείς δοκιμές πάνω στο ρομπότ και επομένως προϋποθέτει την διαρκή φυσική παρουσία και πλήρη ετοιμότητα αυτού. Τα παραπάνω προβλήματα λύνονται με την χρήση προσομοιωτών ρομποτικών συστημάτων, μέσω των οποίων είναι δυνατή η μοντελοποίηση ενός ή περισσότερων ρομποτικών συστημάτων και η προσομοίωση αυτών, σε ένα τεχνητό περιβάλλον το οποίο μπορεί να αποτελεί μία επαρκής προσέγγιση του πραγματικού περιβάλλοντος, βάσει της εκάστοτε εφαρμογής.

Κατά την ανάπτυξη της ρομποτικής πλατφόρμας Monstertruck, στα πλαίσια της παρούσας διπλωματικής εργασίας, χρησιμοποιήθηκαν δύο προσομοιωτές ρομποτικών συστημάτων, για προσομοίωση του ρομπότ σε δύο και τρεις διαστάσεις, ανάλογα με τις απαιτήσεις του εκάστοτε αντικειμένου, που εξεταζόταν την κάθε στιγμή. Συγκεκριμένα, χρησιμοποιήθηκε ο δισδιάστατος προσομοιωτής STDR [56] για την προσομοίωση του ρομπότ κατά την ανάπτυξη του συστήματος αυτόνομης πλοήγησης, όσον αφορά την κατασκευή ολικών μονοπατιών, την αποφυγή στατικών εμποδίων, όπως επίσης και την διάσχιση μονοπατιού. Παράλληλα, χρησιμοποιήθηκε ο τρισδιάστατος προσομοιωτής GAZEBO [57] κατά την ανάπτυξη του κινηματικού μοντέλου του ρομπότ, όπως επίσης και για την εξέταση της συμπεριφοράς - απόκρισης του συστήματος αυτόνομης πλοήγησης και γενικότερα του συνολικού συστήματος λογισμικού της ρομποτικής πλατφόρμας Monstertruck στις τρεις διαστάσεις, σε προσεγγιστικά πραγματικές συνθήκες, όπως για παράδειγμα ανώμαλο έδαφος. Στην συνέχεια ακολουθεί μία πιο αναλυτική περιγραφή των δύο προσομοιωτών, όπως επίσης και η ενσωμάτωση της ρομποτικής πλατφόρμας Monstertruck στα αντίστοιχα περιβάλλοντα.

4.3.1 STDR

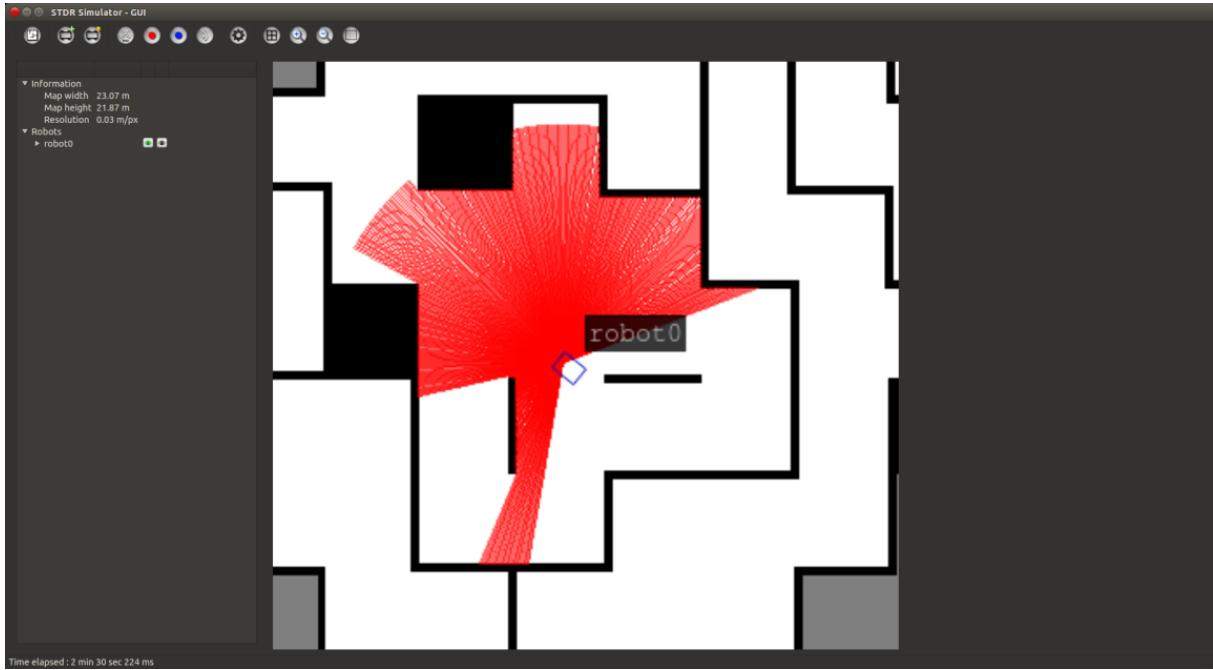
Ο προσομοιωτής STDR (Simple Two Dimensional Robot Simulator) αποτελεί έναν απλό, ελαφρύ δισδιάστατο προσομοιωτή, με δυνατότητα ταυτόχρονης προσομοίωσης πολλαπλών ρομπότ. Είναι πλήρως συμβατός με το ROS και στοχεύει στην παροχή μεγάλου πλήθους δυνατοτήτων για ρομποτικές εφαρμογές, χωρίς, παρόλα αυτά, να προσφέρει την πιο ρεαλιστική αναπαράσταση των συνθηκών λειτουργίας του ρομπότ. Ο προσομοιωτής STDR υποστηρίζει ρομπότ με κυκλικό και πολυγωνικό σχήμα, με differential, skid-steer ή omnidirectional κινηματικό μοντέλο. Επίσης, υποστηρίζει μία πληθώρα από αισθητήρες, όπως για παράδειγμα, σαρωτές λέιζερ, αισθητήρες υπερήχων, θερμικούς αισθητήρες, αισθητήρες μέτρησης του διοξειδίου του άνθρακα στο περιβάλλον κ.α.

Για την προσομοίωση της ρομποτικής πλατφόρμας Monstertruck με τον προσομοιωτή STDR ορίστηκε ένα ρομποτικό μοντέλο με ορθογωνικό αποτύπωμα με τις διαστάσεις του αποτυπώματος του πραγματικού ρομπότ. Όσον αφορά τους αισθητήρες του μοντέλου, χρησιμοποιήθηκε μόνο ένας σαρωτής λέιζερ με αντίστοιχες προδιαγραφές με τον σαρωτή λέιζερ Hokuyo URG-04LX (πίνακας 2.2), που χρησιμοποιείται στο πραγματικό ρομπότ.

Εφόσον, ο προσομοιωτής STDR δεν υποστηρίζει το κινηματικό μοντέλο τετραδιεύθυνσης που χρησιμοποιεί η ρομποτική πλατφόρμα Monstertruck, χρησιμοποιήθηκε προσεγγιστικά το omnidirectional κινηματικό μοντέλο, σε συνδυασμό με έναν κόμβο που φιλτράρει τις εντολές ταχυτήτων και επιτρέπει μόνο αυτές που είναι εφικτές για το κινηματικό μοντέλο τετραδιεύθυνσης, προσεγγίζοντας έτσι το μοντέλο της ρομποτικής πλατφόρμας Monstertruck.

4.3.2 Gazebo

Το Gazebo αποτελεί έναν τρισδιάστατο προσομοιωτή ρομποτικών συστημάτων, με δυνατότητες ταυτόχρονης προσομοίωσης πολλαπλών ρομπότ, σε εσωτερικά και εξωτερικά δυναμικά περιβάλλοντα μέσω ανεπτυγμένων μηχανών φυσικής. Προσφέρει μία μεγάλη γκάμα ρομποτικών μοντέλων, μοντέλων αισθητήρων και περιβαλλόντων, αλλά και δυνατότητες εύκολης ανάπτυξης και επέκτασης

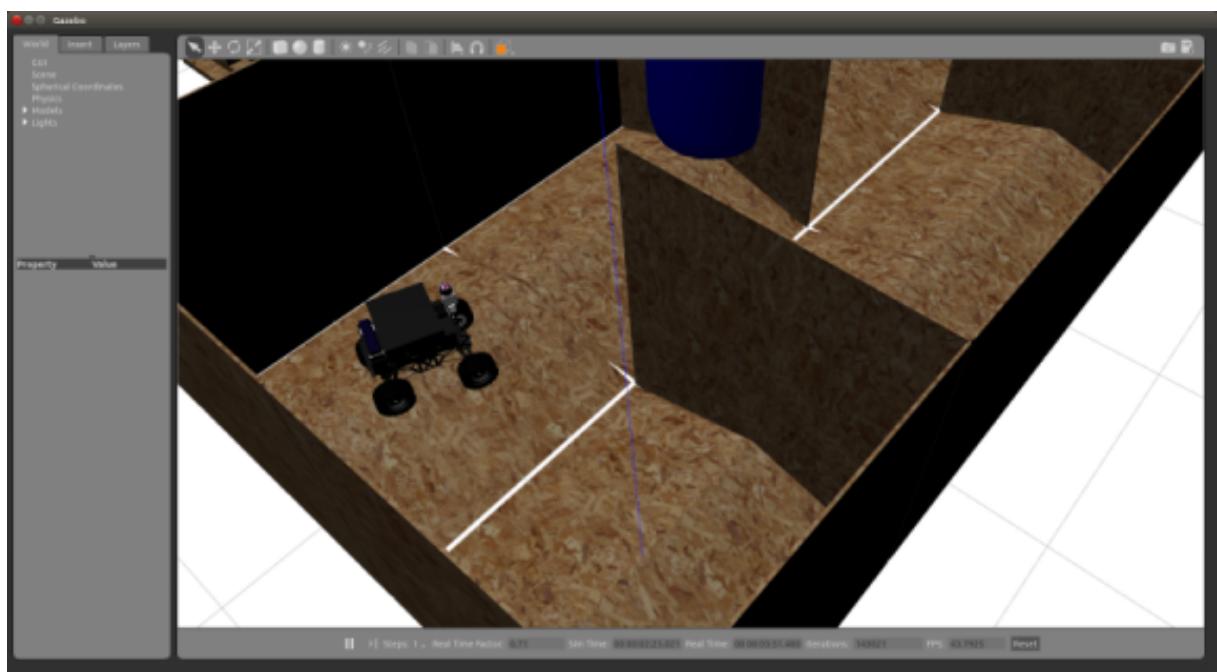


Σχήμα 4.23: Περιβάλλον προσομοίωσης της ρομποτικής πλατφόρμας Monstertruck με τον προσομοιωτή STDR.

νέων. Σε αντίθεση με δισδιάστατους προσομοιωτές, όπως o STDR, το Gazebo προσφέρει ένα ιδιαίτερα ρεαλιστικό περιβάλλον για την προσομοίωση ρομποτικών συστημάτων σε συνθήκες αρκετά κοντά στην πραγματικότητα. Συγκεκριμένα, δίνει την δυνατότητα για προσομοίωση ρομποτικών συστημάτων υπό συνθήκες τριβών και ολίσθησης, σε ποικιλόμορφα περιβάλλοντα και αποτελεί ιδανική λύση για ταχεία και ασφαλή δοκιμή αλγορίθμων, όπως επίσης και για ανάπτυξη και προτυποποίηση νέων ρομποτικών συστημάτων.

Στα πλαίσια της παρούσας διπλωματικής εργασίας, ο προσομοιωτής Gazebo χρησιμοποιήθηκε, αρχικά, για την ανάπτυξη και επαλήθευση του κινηματικού μοντέλου της ρομποτικής πλατφόρμας Monstertruck, υπό συνθήκες τριβών και ολίσθησης και έπειτα για την προσομοίωση του πλήρους ρομποτικού συστήματος σε ένα τρισδιάστατο περιβάλλον, με συνθήκες αντίστοιχες, με αυτές για τις οποίες προορίστηκε το πραγματικό ρομπότ, αποφεύγοντας έτσι να τεθεί σε κίνδυνο το φυσικό ρομπότ, πριν επαληθευτεί η εύρωση συμπεριφορά του συστήματος.

Για την προσομοίωση της ρομποτικής πλατφόρμας Monstertruck χρησιμοποιήθηκε το μοντέλο URDF που αναφέρθηκε στην ενότητα 4.2.5 σε συνδυασμό με ένα σύνολο από plugins για την προσομοίωση της κίνησης των αρθρώσεων του και την λειτουργία των αισθητήρων του, δηλαδή του σαρωτή λέιζερ και της πυξίδας, βάσει των προδιαγραφών του πραγματικού ρομπότ. Επίσης, χρησιμοποιήθηκε και ένα plugin που εκδίδει την ιδανική οδομετρία του ρομπότ για σύγκριση με το μη ιδανικό σύστημα οδομετρίας που αναπτύχθηκε.



Σχήμα 4.24: Περιβάλλον προσομοίωσης της ρομποτικής πλατφόρμας Monstertruck με τον προσομοιωτή Gazebo.

Κεφάλαιο 5

Πειράματα και Αποτελέσματα

Στο παρόν κεφάλαιο παρουσιάζονται τα πειράματα και τα αποτελέσματα που πραγματοποιήθηκαν για την επαλήθευση και την αξιολόγηση της συμπεριφοράς των υποσυστημάτων της ρομποτικής πλατφόρμας Monstertruck, σε σχέση με την επιθυμητή. Παράλληλα, πραγματοποιείται και σύγκριση μεταξύ των διαφορετικών εκδοχών ορισμένων υποσυστημάτων, όπως αυτά παρουσιάστηκαν στο κεφάλαιο 4.

Η εκτέλεση των πειραμάτων πραγματοποιήθηκε, μέσω προσομοίωσης σε δύο διαστάσεις με τον προσομοιωτή STDR, σε τρεις διαστάσεις με τον προσομοιωτή Gazebo, όπως επίσης και στο φυσικό ρομπότ που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής. Τα εν λόγω πειράματα εκτελέστηκαν σε επίπεδο προσομοίωσης σε έναν υπολογιστή με επεξεργαστή Intel Core i5 2.4GHz, 6GB RAM και κάρτα γραφικών NVIDIA GeForce GT330M, ενώ σε επίπεδο φυσικού ρομπότ σε έναν υπολογιστή ODROID-XU4 με προδιαγραφές, όπως αυτές παρουσιάστηκαν στον πίνακα 2.1, όπου και τα δύο υπολογιστικά συστήματα έτρεχαν το λειτουργικό σύστημα Ubuntu 14.04, σε συνδυασμό με το αντίστοιχο συμβατό meta-λειτουργικό σύστημα ROS Indigo Igloo.

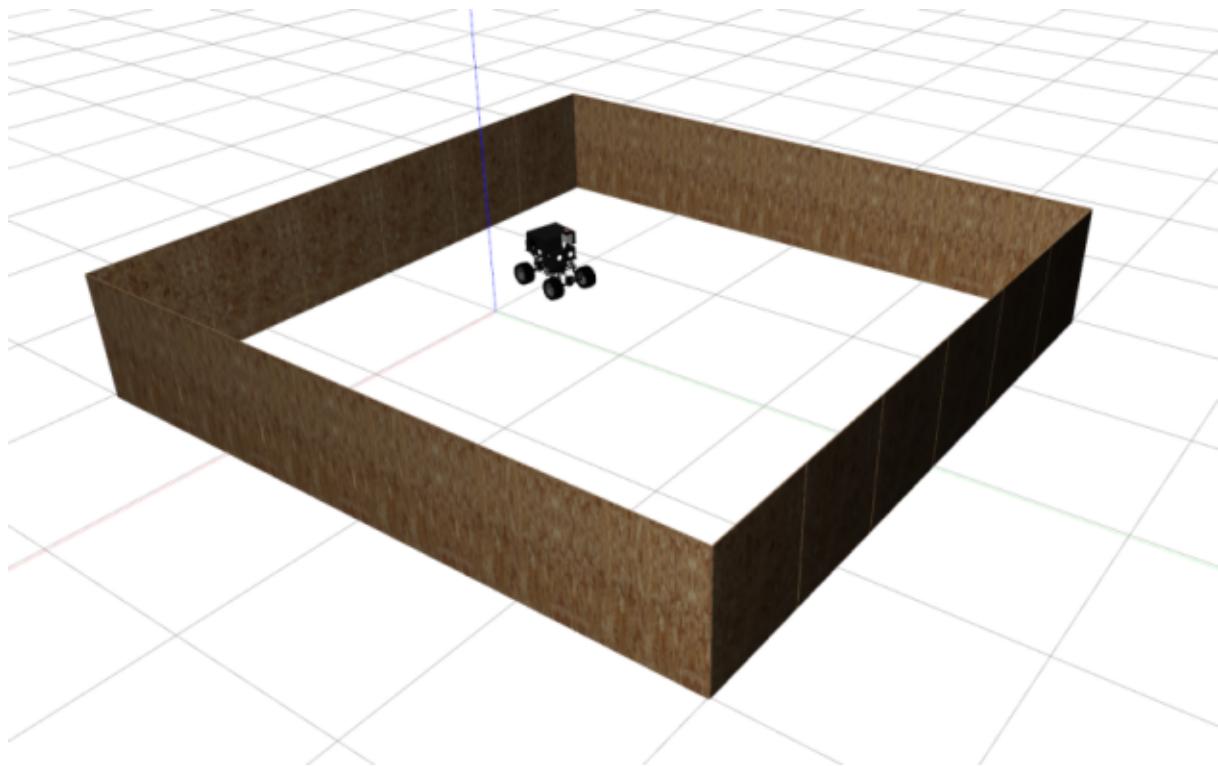
Στην συνέχεια παρουσιάζονται τα πειράματα και τα αποτελέσματα που πραγματοποιήθηκαν για την επαλήθευση και αξιολόγηση του κινηματικού μοντέλου, των συστημάτων χαρτογράφησης και εντοπισμού θέσης, των συστημάτων κατασκευής μονοπατιού, αποφυγής εμποδίων και διάσχισης μονοπατιού, ενώ τελικά γίνεται και μία αξιολόγηση της αυτόνομης συμπεριφοράς του ρομπότ, αλλά και της απόδοσης του υπολογιστή του ρομπότ, όσον αφορά τους πόρους που δεσμεύονται από κάθε διεργασία ξεχωριστά και όλων μαζί.

5.1 Πειράματα Κινηματικού Μοντέλου

Η αξιολόγηση του κινηματικού μοντέλου που αναπτύχθηκε για την ρομποτική πλατφόρμα Monstertruck, πραγματοποιήθηκε, αρχικά, στον προσομοιωτή Gazebo και έπειτα στο φυσικό ρομπότ, μέσα από μία σειρά πειραμάτων με στόχο την σύγκριση της ιδανικής με την πραγματική συμπεριφορά του ρομπότ.

Για την εκτέλεση των πειραμάτων, κρίθηκε σκόπιμο να χρησιμοποιηθούν χώροι, αρκετά μικροί, ούτως ώστε να επιτρέπουν την χαρτογράφηση και τον εντοπισμό θέσης του ρομπότ, βάσει της εμβέλειας του σαρωτή λέιζερ Hokuyo URG-04LX, ενώ παράλληλα να είναι αρκετά μεγάλοι, ώστε να επιτρέπουν την εκτέλεση των επιθυμητών τροχιών από το ρομπότ. Επίσης, κρίθηκε σκόπιμο να χρησιμοποιηθούν χώροι με επίπεδο και ομαλό έδαφος, για την πιο αξιόπιστη εκτέλεση των πειραμάτων, χωρίς την εισαγωγή διαταραχών, λόγω ανώμαλου εδάφους ή ολίσθησης σε κεκλιμένο επίπεδο. Συγκεκριμένα, για την εκτέλεση των πειραμάτων στον προσομοιωτή Gazebo αναπτύχθηκε το περιβάλλον *simple_room* που αποτελείται από ένα απλό δωμάτιο $4m \times 4m \times 1.5m$ με τέσσερις τοίχους και επίπεδο και ομαλό έδαφος, όπως αυτό παρουσιάζεται στην εικόνα 5.1, ενώ τα πειράματα στο φυσικό ρομπότ πραγματοποιήθηκαν στην πίστα πειραμάτων του εργαστηρίου της ομάδας P.A.N.D.O.R.A, όπως παρουσιάζεται στην εικόνα ??.

TODO: Insert Lab Circuit Photo



Σχήμα 5.1: Το περιβάλλον simple_room που χρησιμοποιήθηκε για την εκτέλεση πειραμάτων αξιολόγησης του κινηματικού μοντέλου στον προσομοιωτή Gazebo.

Για την εκτέλεση των πειραμάτων χρησιμοποιήθηκε το σύστημα χαρτογράφησης και εντοπισμού θέσης με τον αλγόριθμο CRSM-SLAM, ενώ ο έλεγχος του ρομπότ έγινε μέσω τηλεχειρισμού. Για την αξιολόγηση της συμπεριφοράς του ρομπότ, χρησιμοποιήθηκαν οι εντολές κίνησης που δόθηκαν στο ρομπότ, ως η επιθυμητή συμπεριφορά και η πραγματική τροχιά του ρομπότ, που εκδίδει ο αλγόριθμος CRSM-SLAM, ως η πραγματική συμπεριφορά.

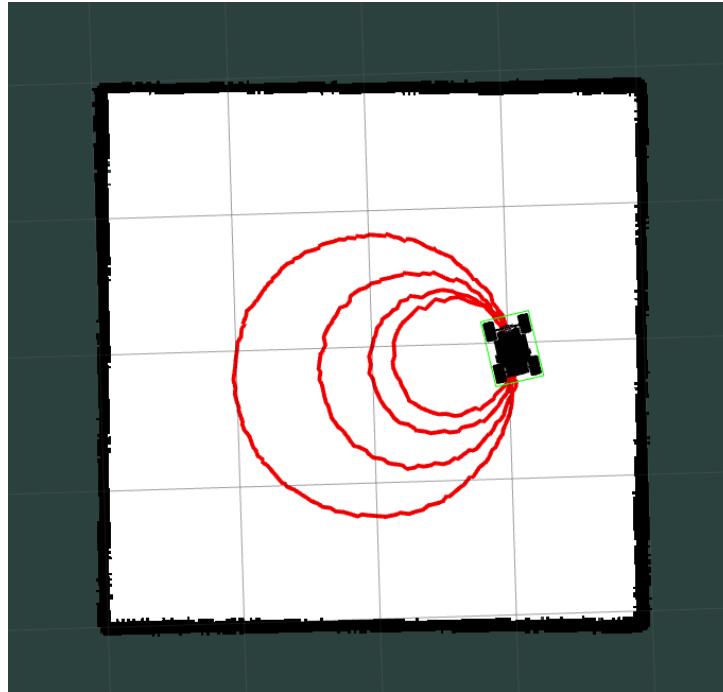
Συνολικά, πραγματοποιήθηκαν δύο είδη πειραμάτων για την αξιολόγηση του κινηματικού μοντέλου και συγκεκριμένα για την αξιολόγηση της ακτίνας τροχιάς κατά την εκτέλεση κυκλικών τροχιών με αρνητική τετραδιεύθυνση και για την αξιολόγηση της πλευρικής γωνίας ολίσθησης κατά την εκτέλεση διαγώνιων ευθύγραμμων τροχιών με θετική τετραδιεύθυνση, όπως αυτά παρουσιάζονται στην συνέχεια.

5.1.1 Τροχιές Αρνητικής Τετραδιεύθυνσης

Για την αξιολόγηση της αρνητικής τετραδιεύθυνσης πραγματοποιήθηκε μία σειρά πειραμάτων, στην οποία, δόθηκαν εντολές στο ρομπότ για εκτέλεση κυκλικών τροχιών, για διάφορες δυνατές γωνίες στρέψης των τροχών, για τις οποίες, έπειτα υπολογίστηκαν οι ιδανικές ακτίνες τροχιάς R_{ideal} και οι πραγματικές R_{real} που εκτελέστηκαν από το ρομπότ, όπως αυτές παρουσιάζονται στον πίνακα 5.1 και στο σχήμα 5.2. Με βάση τον πίνακα αυτόν, παρατηρείται μία μικρή απόκλιση των πραγματικών τροχιών που εκτελεί το ρομπότ σε σύγκριση με τις ιδανικές, κάτι που, παρόλα αυτά, είναι δικαιολογημένο, λόγω της μη ιδανικότητας του κινηματικού μοντέλου τετραδιεύθυνσης της ρομποτικής πλατφόρμας Monstertruck, όπως επίσης και την επιρροή φαινομένων τριβών και ολίσθησης στους τροχούς.

Πίνακας 5.1: Αποτελέσματα πειράματος κινηματικού σε τροχιές αρνητικής τετραδιεύθυνσης.

$\delta_f, -\delta_r$	R_{ideal}	R_{real}	ϵ_R
0.4	0.378	0.377	0.001
0.32	0.483	0.536	0.053
0.24	0.654	0.722	0.068
0.16	0.991	1.048	0.058



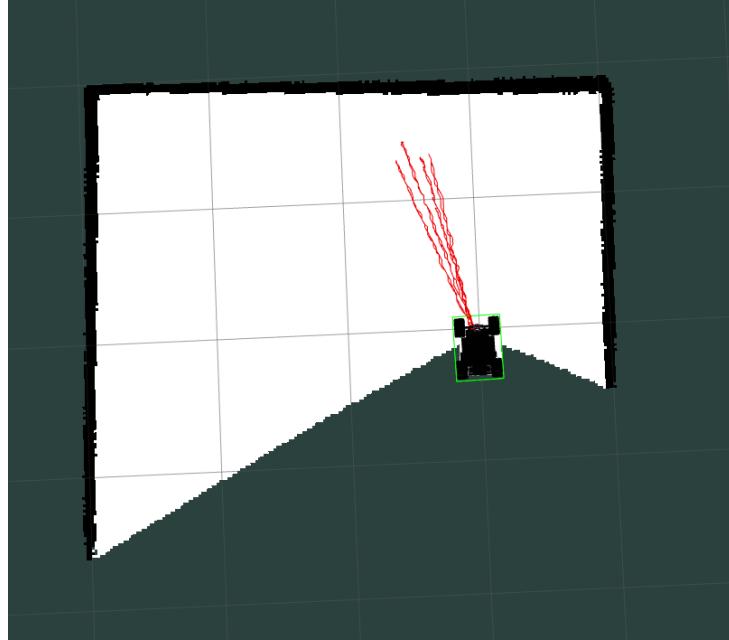
Σχήμα 5.2: Τροχιές που παράχθηκαν κατά τα πειράματα αξιολόγησης του κινηματικού μοντέλου σε τροχιές αρνητικής τετραδιεύθυνσης στο περιβάλλον simple_rooms.

5.1.2 Τροχιές Θετικής Τετραδιεύθυνσης

Για την αξιολόγηση της θετικής τετραδιεύθυνσης πραγματοποιήθηκε μία σειρά πειραμάτων, στην οποία, δόθηκαν εντολές στο ρομπότ για εκτέλεση διαγώνιων τροχιών, για διάφορες δυνατές γωνίες στρέψης των τροχών δ_f, δ_r , τέτοιες ώστε $\delta_f = \delta_r$, ενώ, έπειτα, υπολογίστηκε η ιδανική γωνία πλευρικής ολίσθησης β_{ideal} , όπως επίσης και η γωνία πλευρικής ολίσθησης β_{real} των πραγματικών τροχιών που εκτελέστηκαν από το ρομπότ και οι οποίες παρουσιάζονται στον πίνακα 5.2 και στο σχήμα 5.3. Παρατηρείται, γενικά, πολύ μικρό ύψως αμελητέο σφάλμα της πραγματικής συμπεριφοράς του ρομπότ σε σύγκριση με την ιδανική και επομένως, αυτή, κρίνεται πλήρως αποδεκτή.

Πίνακας 5.2: Αποτελέσματα πειράματος κινηματικού σε τροχιές θετικής τετραδιεύθυνσης.

δ_f, δ_r	β_{ideal}	β_{real}	ϵ_β
0.4	0.4	0.398	0.002
0.32	0.32	0.326	0.006
0.24	0.24	0.232	0.008
0.16	0.16	0.151	0.009



Σχήμα 5.3: Τροχιές που παράχθηκαν κατά τα πειράματα αξιολόγησης του κινηματικού μοντέλου σε τροχιές θετικής τετραδιεύθυνσης στο περιβάλλον simple_rooms.

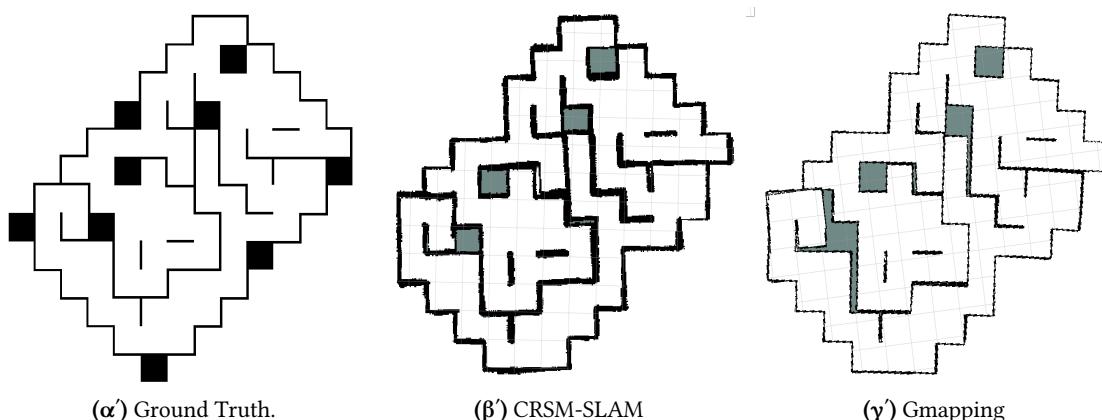
5.2 Πειράματα Χαρτογράφησης

Στην παρούσα ενότητα γίνεται μία αξιολόγηση των συστημάτων χαρτογράφησης και υπολογισμού θέσης, όπως αυτά παρουσιάσθηκαν στην ενότητα 4.2.3, μέσω σύγκρισης των δύο επιμέρους συστημάτων, αλλά και μέσω σύγκρισης με τον πραγματικό χάρτη, εφόσον αυτός παρέχεται. Τα πειράματα πραγματοποιήθηκαν στον 2D προσομοιωτή STDR, στον 3D προσομοιωτή Gazebo, αλλά και στο φυσικό ρομπότ, με χρήση των συστημάτων χαρτογράφησης και εντοπισμού θέσης των αλγορίθμων CRSM-SLAM και Gmapping.

Για την χαρτογράφηση του περιβάλλοντος, η ρομποτική πλατφόρμα Monstertruck χρησιμοποιεί ως βασικό εργαλείο τον σαρωτή λέιζερ Hokuyo URG-04LX, ο οποίος έχει εμβέλεια $4m$ και επομένως τα περιβάλλοντα των πειραμάτων επιλέχθηκαν αναλόγως. Συγκεκριμένα, επιλέχθηκαν περιβάλλοντα με εσωτερικούς κλειστούς χώρους, με αρκετά στενούς διαδρόμους και δωμάτια με επαρκή ποικιλομορφία για την όσο το δυνατόν πιο αξιόπιστη χαρτογράφηση. Επίσης, λόγω της περιορισμένης υπολογιστικής ισχύς του υπολογιστή ODROID-XU4, της ρομποτικής πλατφόρμας Monstertruck, επιλέχθηκε ανάλυση χαρτογράφησης ίση με $0.04m$ και γενικότερα ειδική παραμετροποίηση των αλγορίθμων SLAM, έτσι ώστε να επιτευχθεί ταυτόχρονα αύξηση της συχνότητας ανανέωσης του χάρτη και μείωση του υπολογιστικού φόρτου, με εμφανή επίδραση, παρόλα αυτά, στην ποιότητα του.

5.2.1 Χαρτογράφηση στον 2D Προσομοιωτή STDR

Για τα πειράματα στον προσομοιωτή STDR, επιλέχθηκε ο χάρτης *robocup* που παρέχεται από τον προσομοιωτή, λόγω της πολυπλοκότητας του και της ομοιότητας του με τα περιβάλλοντα για τα οποία προορίζεται η ρομποτική πλατφόρμα Monstertruck. Παρόλα αυτά, μετά από αρχικά πειράματα, διαπιστώθηκε ότι ο χάρτης είχε κάποιους μεγάλους ομοιόμορφους διαδρόμους που δημιουργούσαν πρόβλημα στην χαρτογράφηση, με τον αλγόριθμο CRSM-SLAM, λόγω αδυναμίας εντοπισμού της ακριβής θέσης, εξαιτίας του περιορισμένου εύρους του σαρωτή λέιζερ και άρα της εσφαλμένης αντιστοίχισης σκαναρισμάτων. Γι αυτό το λόγο, κρίθηκε σκόπιμο να μειωθεί η ανάλυση του χάρτη από 0.03 σε 0.02 με στόχο την μείωση της κλίμακας του. Στο σχήμα 5.4 παρουσιάζεται ο χάρτης *robocup* και οι αντίστοιχοι χάρτες που κατασκεύασαν οι αλγόριθμοι CRSM-SLAM και Gmapping.



Σχήμα 5.4: Χάρτες του περιβάλλοντος robocup.

Βάσει του σχήματος 5.4, παρατηρείται, ότι τα δύο συστήματα χαρτογράφησης που εξετάστηκαν, παράγουν μία αναπαράσταση του περιβάλλοντος robocup, με αρκετές ατέλειες, αλλά χωρίς να υπάρχει κάποιο εξαιρετικά μεγάλο σφάλμα που να επιδρά σημαντικά στην ακεραιότητα του χάρτη και επομένως κρίνονται ικανοποιητικά για τα πλαίσια της παρούσας εργασίας.

5.2.2 Χαρτογράφηση στον 3D προσομοιωτή Gazebo

Για τα πειράματα χαρτογράφησης στον προσομοιωτή Gazebo, επιλέχθηκε ο χάρτης της αρένας του διαγωνισμού RoboCup Rescue 2013, σε δύο εκδοχές, με ομαλό και ανώμαλο έδαφος για την εξέταση της συμπεριφοράς του συστήματος, υπό όλες τις δυνατές συνθήκες για τις οποίες προορίζεται να λειτουργήσει αυτό.

Για την διεξαγωγή των πειραμάτων, το ρομπότ τέθηκε σε λειτουργία αυτόματης εξερεύνησης χωρίς κάποια χειροκίνητη παρέμβαση, για την κάλυψη όλης της αρένας για τις περιπτώσεις ομαλού και ανώμαλου εδάφους, με στόχο την πλήρη χαρτογράφηση αυτών, μέσω των αλγορίθμων CRSM-SLAM και Gmapping.

Χαρτογράφηση της Αρένας RoboCup Rescue 2013 με Ομαλό Έδαφος

Για την χαρτογράφηση, σε ομαλό έδαφος, η αρένα RoboCup Rescue 2013 προσαρμόστηκε κατάλληλα για να επιτρέπει την πλήρη προσπέλαση της από το ρομπότ, μέσω της αφαίρεσης ανυψωμένων τμημάτων, όπως ράμπες, σκάλες κοκ. Στο σχήμα 5.6 παρουσιάζεται η προσαρμοσμένη αρένα RoboCup Rescue 2013, με ομαλό έδαφος, όπως επίσης και οι αντίστοιχοι χάρτες που κατασκευάστηκαν από τους αλγορίθμους CRSM-SLAM και Gmapping.

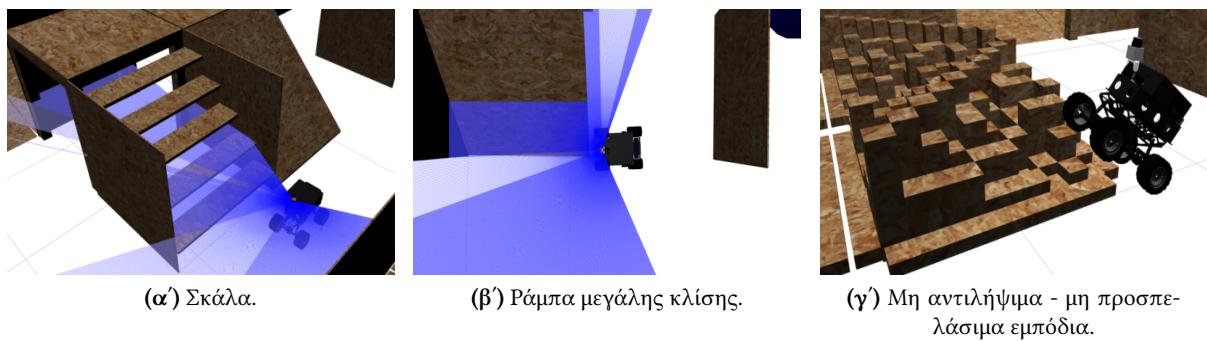
Συγκρίνοντας, τους παραγόμενους χάρτες με το πραγματικό περιβάλλον παρατηρούνται κάποια σφάλματα συστροφής μεταξύ των οριζόντιων διαδρόμων της αρένας, που παρόλα αυτά, δεν επιδρούν σημαντικά στην ακεραιότητα του χάρτη. Τα δύο συστήματα χαρτογράφησης κατασκεύασαν επαρκείς αναπαραστάσεις της αρένας, με ελαφρώς καλύτερη αυτή του αλγορίθμου Gmapping, εις βάρος, παρόλα αυτά του υπολογιστικού φόρτου και των απαιτήσεων αυτού.

Χαρτογράφηση της Αρένας RoboCup Rescue 2013 με Ανώμαλο Έδαφος

Για την περίπτωση της αρένας με ανώμαλο έδαφος, χρησιμοποιήθηκε αρχικά η ακέραιη εκδοχή της αρένας RoboCup Rescue 2013, αλλά παρατηρήθηκε ότι περιείχε τμήματα τα οποία δεν ήταν δυνατόν να γίνουν αντιληπτά, ενώ παράλληλα δεν ήταν ούτε προσπελάσιμα από το ρομπότ. Επομένως, κρίθηκε σκόπιμο, η εν λόγω αρένα να προσαρμοστεί, ούτως ώστε να αφαιρεθούν τα τμήματα αυτά.

Συγκεκριμένα, αφαιρέθηκαν από την αρένα τα ακόλουθα τμήματα, όπως αυτά παρουσιάζονται ακολούθως και στο σχήμα 5.5:

- Μία σκάλα, η οποία είχε κενό μεταξύ των σκαλοπατιών, με αποτέλεσμα να περνάνε διαμέσου αυτής οι ακτίνες του σαρωτή λέιζερ και άρα αυτήν να γίνεται αντιληπτή εσφαλμένα ως ελεύθερος χώρος.
- Μία ράμπα μεγάλης κλίσης, της οποίας το κατώτερο τμήμα δεν αναγνωρίζοταν ως εμπόδιο, με κίνδυνο σύγκρουσης με το ρομπότ.
- Ένα σύνολο τμημάτων με υψηλά και άνισα κομμάτια που προεξέιχαν σε αρκετά χαμηλό ύψος, ώστε να μην χτυπάνε οι ακτίνες του λέιζερ σ' αυτά, αλλά παράλληλα αρκετά ψηλά, ώστε να προκαλούν την προσκόλληση ή ακόμα και ανατροπή του ρομπότ.



Σχήμα 5.5: Τα τμήματα που αφαιρέθηκαν από την αρένα με το ανώμαλο έδαφος, λόγω της αδυναμίας εντοπισμού και προσπελασμότητας από την ρομποτική πλατφόρμα Monstertruck.

Αφού αφαιρεθούν τα παραπάνω εμπόδια από τον χάρτη, το ανώμαλο έδαφος προσεγγίζεται, πλέον, με ξύλινες ράμπες με κλίση περίπου 15° . Η αξιόπιστη χαρτογράφηση περιβάλλοντος, αυτού του τύπου, καθίσταται δυνατή από την αντιστάθμιση κλίσης roll και pitch του σαρωτή λέιζερ, μέσω του μηχανισμού σταθεροποίησης αυτού, σε συνδυασμό με την πυξίδα που παρέχει τις μετρήσεις των κλίσεων roll και pitch του ρομπότ. Στα σχήματα 5.7 που ακολουθούν, παρουσιάζεται η προσαρμοσμένη αρένα και οι αντίστοιχοι χάρτες.

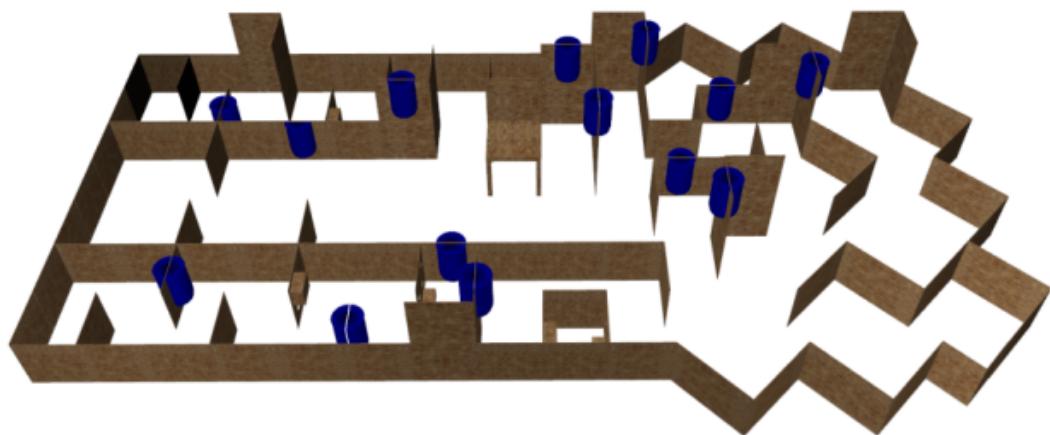
Οι χάρτες που κατασκεύασαν τα δύο συστήματα SLAM για την αρένα με το ανώμαλο έδαφος παρουσιάζουν εξαιρετικά μεγάλη ομοιότητα με τους αντίστοιχους χάρτες της αρένας με το ομαλό έδαφος, γεγονός που αποδεικνύει την ευρωστία του συστήματος σε όλες τις πιθανές συνθήκες λειτουργίας.

5.2.3 Χαρτογράφηση σε Πραγματικό Περιβάλλον με τη Ρομποτική Πλατφόρμα Monstertruck

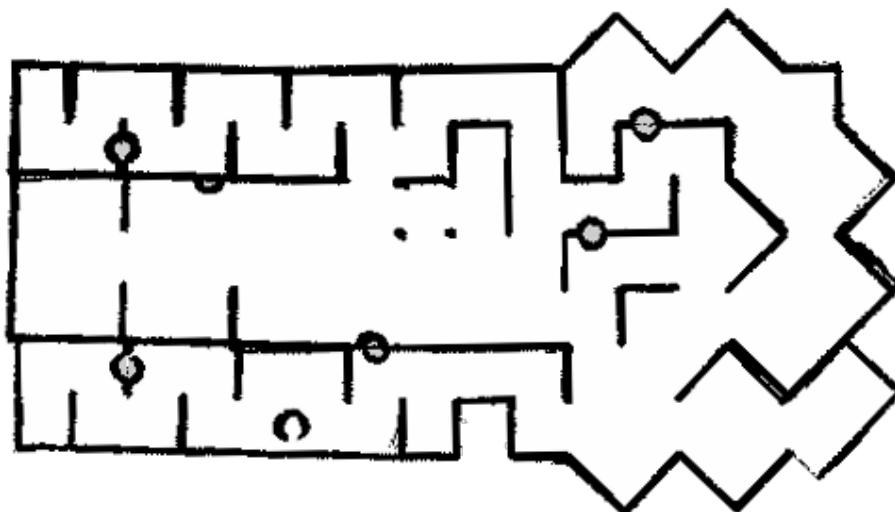
Πέρα από τα πειράματα χαρτογράφησης σε επίπεδο προσομοίωσης, αντίστοιχα πειράματα πραγματοποιήθηκαν για τη φυσική ρομποτική πλατφόρμα Monstertruck. Για την εκτέλεση των πειραμάτων με το φυσικό ρομπότ, επιλέχθηκε ο χώρος του Εργαστηρίου Αρχιτεκτονικής και Υπολογιστών, του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης. Ο χώρος αυτός αποτελεί ένα δωμάτιο με τρεις σειρές γραφείων υπολογιστών με καρέκλες και περιλαμβάνει εξ ολοκλήρου ομαλό έδαφος. Για την αύξηση της πολυπλοκότητας του εν λόγω χώρου, παρόλα αυτά, προστέθηκαν και ξύλινες ράμπες για την επαλήθευση των αποτελεσμάτων που λήφθηκαν κατά την προσομοίωση, υπό όλες τις δυνατές συνθήκες. Στο σχήμα ?? παρουσιάζεται ο

χώρος του εργαστηρίου, όπως επίσης και οι αντίστοιχοι χάρτες που παράχθηκαν, βάσει αυτού, μέσω των αλγορίθμων CRSM-SLAM και Gmapping.

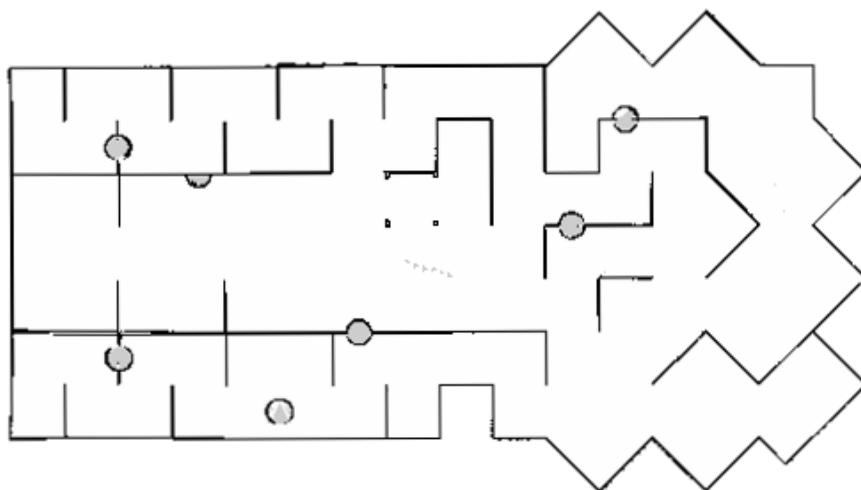
TODO: Προσθήκη φωτογραφίας εργαστηρίου αρχιτεκτονικής υπολογιστών και αντίστοιχων χαρτών.



(α') Αρένα RoboCup Rescue 2013 με ομαλό έδαφος.

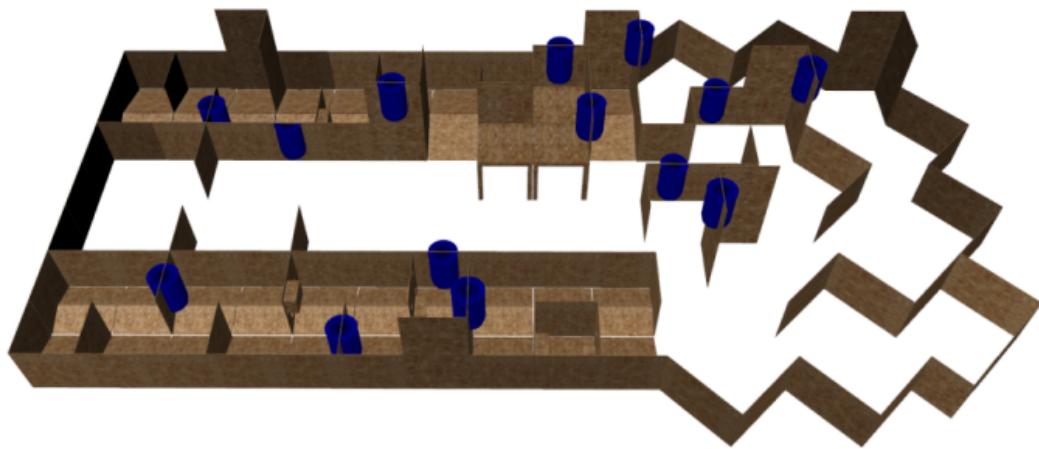


(β') Χάρτης μέσω CRSIM-SLAM.

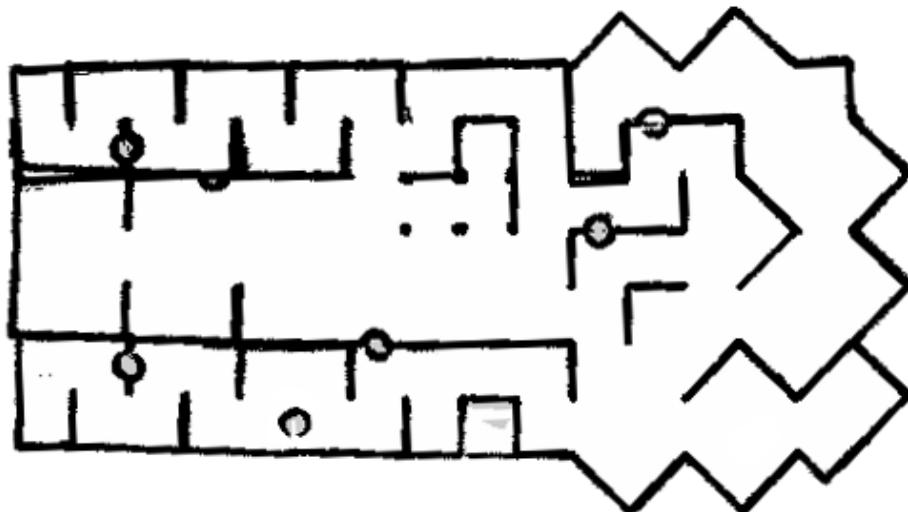


(γ') Χάρτης μέσω Gmapping.

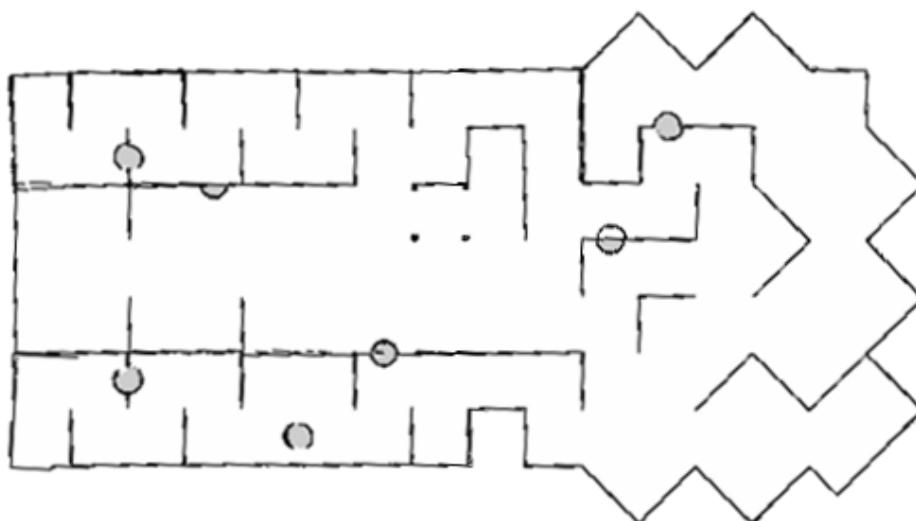
Σχήμα 5.6: Η αρένα του διαγωνισμού RoboCup Rescue 2013 με ομαλό έδαφος και οι αντίστοιχοι παραγόμενοι χάρτες.



(α') Αρένα RoboCup Rescue 2013 με ανώμαλο έδαφος.



(β') Χάρτης μέσω CRSIM-SLAM.



(γ') Χάρτης μέσω Gmapping.

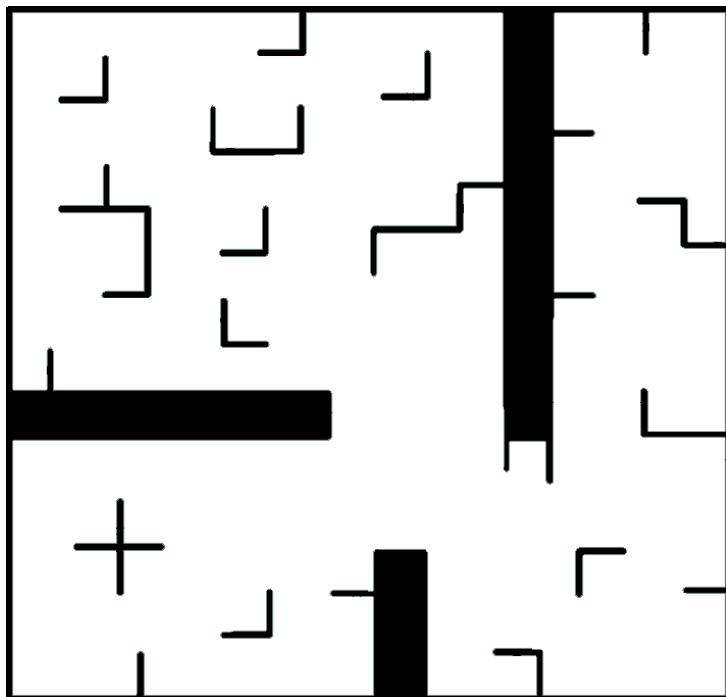
Σχήμα 5.7: Η αρένα του διαγωνισμού RoboCup Rescue 2013 με ομαλό έδαφος και οι αντίστοιχοι παραγόμενοι χάρτες.

5.3 Πειράματα Αξιολόγησης Συστημάτων Αυτόνομης Πλοήγησης

Στην παρούσα ενότητα παρουσιάζονται πειράματα σχετικά με τα επιμέρους τμήματα των συστημάτων αυτόνομης πλοήγησης της ρομποτικής πλατφόρμας Monstertruck, όπως επίσης και μία σύγκριση μεταξύ αυτών, ενώ τελικά παρουσιάζεται και η συνολική συμπεριφορά των δύο συστημάτων. Συγκεκριμένα, παρουσιάζεται η συμπεριφορά των αλγορίθμων κατασκευής μονοπατιών, του αλγορίθμου παραμόρφωσης μονοπατιού Reeds-Shepp Band, ο αλγόριθμος διάσχισης μονοπατιού που βασίζεται σε ασαφή λογική, ενώ τέλος γίνεται και μία σύγκριση μεταξύ του συστήματος αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού και του συστήματος αυτόνομης πλοήγησης με δυναμική ανακατασκευή μονοπατιού.

5.3.1 Κατασκευή Μονοπατιών

Τα πειράματα κατασκευής μονοπατιού πραγματοποιήθηκαν στον 2D προσομοιωτή STDR, χωρίς την χρήση κάποιου αλγορίθμου SLAM, αλλά χρησιμοποιώντας το ground truth του χάρτη του περιβάλλοντος και την τέλεια οδομετρία που παρέχει ο προσομοιωτής, για εντοπισμό της θέσης του ρομπότ. Για την διεξαγωγή των πειραμάτων, επιλέχθηκε το περιβάλλον sparse_obstacles που παρέχει ο προσομοιωτής STDR και το οποίο παρουσιάζεται στο σχήμα 5.8.



Σχήμα 5.8: Το περιβάλλον sparse_obstacles του προσομοιωτή STDR, που χρησιμοποιήθηκε για τα πειράματα κατασκευής μονοπατιού.

Για την περίπτωση των αλγορίθμων Dijkstra και A^* που υλοποιεί το plugin global_planner, ορίζονται δύο μετρικές αξιολόγησης, όπου, η πρώτη μετρική είναι ο χρόνος T που απαιτεί ο αλγόριθμος για την κατασκευή του μονοπατιού, ενώ η δεύτερη μετρική είναι το συνολικό μήκος s του μονοπατιού. Παράλληλα, μπορεί να γίνει και μία ποιοτική αξιολόγηση των αλγορίθμων Dijkstra και A^* , βάσει του καλυπτόμενου χώρου αναζήτησης.

Στην περίπτωση των αλγορίθμων ARA* και AD* που υλοποιεί το plugin sbpl_global_planner, δεν είναι δυνατή η χρήση των ίδιων μετρικών με τους Dijkstra και A^* , μιας και οι αλγόριθμοι αυτοί, χρησιμοποιούν σταθερό χρόνο T για την κατασκευή ενός άκρως μη βέλτιστου μονοπατιού, ενώ παράλληλα συνεχίζουν την διαρκή βελτιστοποίηση αυτού, μειώνοντας σταδιακά τον παράγοντα διαστολής ϵ και

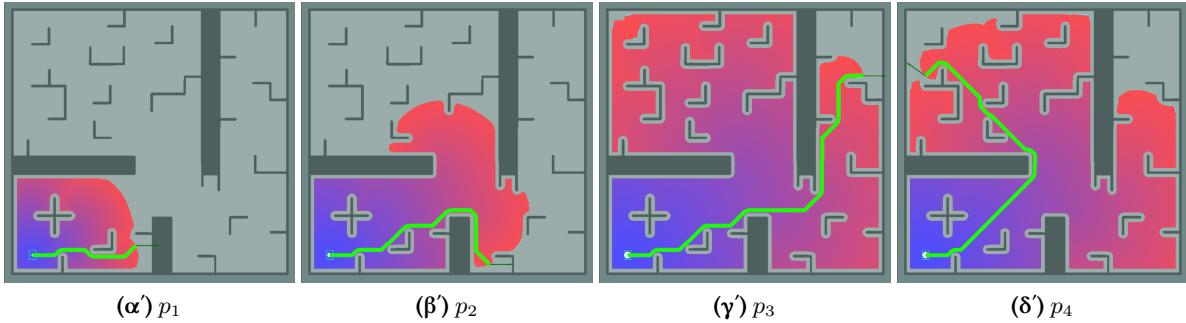
βελτιστοποιώντας το μονοπάτι αυτό για τον τρέχοντα παράγοντα ϵ σε διάστημα T . Επομένως, ορίζονται δύο νέες μετρικές, το μήκος s_{init} του αρχικού μη βέλτιστου μονοπατιού και το τελικό μήκος s_{final} του τελικού μονοπατιού. Επίσης, αναφέρεται ότι επιλέχθηκε, χρόνος κατασκευής μονοπατιού $T = 0.1s$ και αρχικός παράγοντας διαστολής $\epsilon_{init} = 3$.

Για την εξέταση και σύγκριση της συμπεριφοράς των παραπάνω αλγορίθμων, ορίστηκαν πέντε πόζες p_i , $i = 1, \dots, 4$, του περιβάλλοντος `sparse_obstacles`, ως στόχοι, για την κατασκευή μονοπατιών, από την αρχική πόζα $p_0 = (1.5, 1.5, 90^\circ)$ του ρομπότ, όπως αυτές παρουσιάζονται στον πίνακα 5.3.

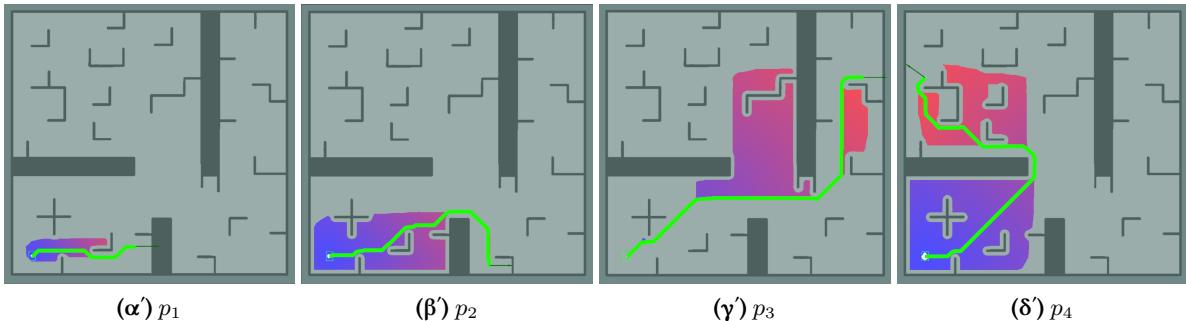
Πίνακας 5.3: Οι πόζες - στόχοι των πειραμάτων κατασκευής μονοπατιών στο περιβάλλον `sparse_obstacles`.

	x[m]	y[m]	θ [μοίρες]
p_1	7	2	0
p_2	10	1	0
p_3	14	11	0
p_4	1.5	11	45°

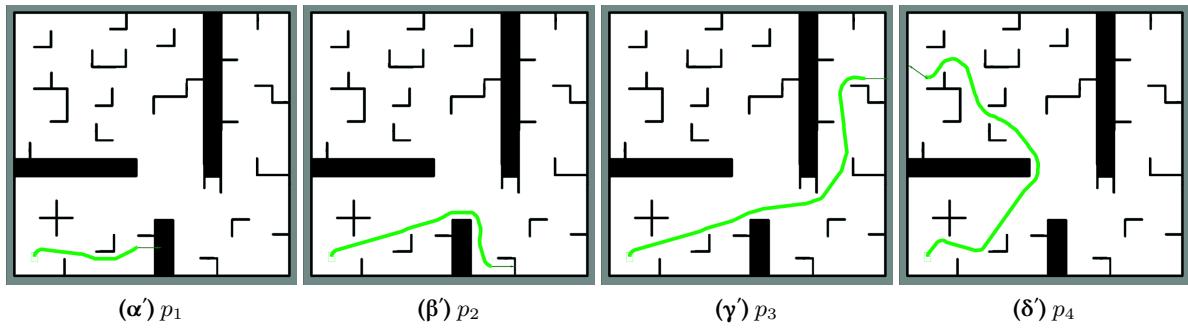
Στην συνέχεια παρουσιάζονται στα σχήματα 5.9-5.12 τα τελικά μονοπάτια που κατασκευάζουν οι αλγόριθμοι κατασκευής μονοπατιών για την μετάβαση από την πόζα p_0 στις πόζες p_i , $i = 1, \dots, 4$, ενώ έπειτα, στον πίνακα 5.4 με τα αποτελέσματα των μετρικών για τα εν λόγω μονοπάτια. Επιπλέον, στα σχήματα 5.13 και 5.12 παρουσιάζονται ενδεικτικές ακολουθίες βελτιστοποίησης μονοπατιού, μέσω των αλγορίθμων ARA* και AD*, για φθίνον παράγοντα διαστολής ϵ από $\epsilon_{init} = 3$ έως $\epsilon_{final} = 1$.



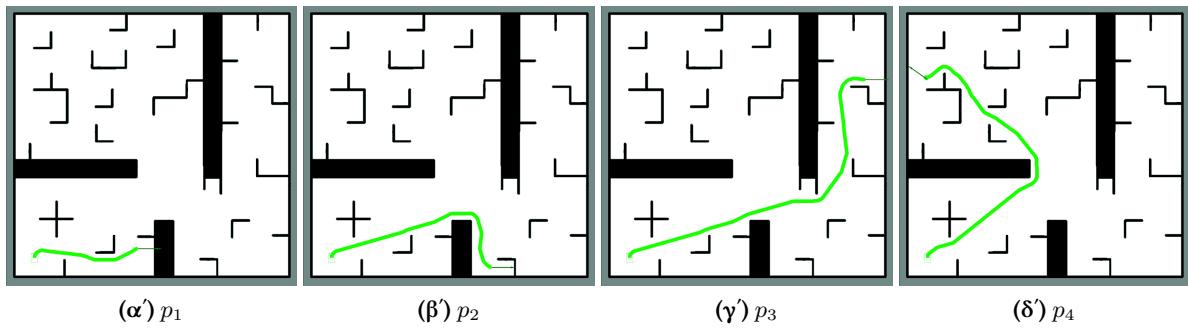
Σχήμα 5.9: Κατασκευή μονοπατιών με τον αλγόριθμο Dijkstra, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.



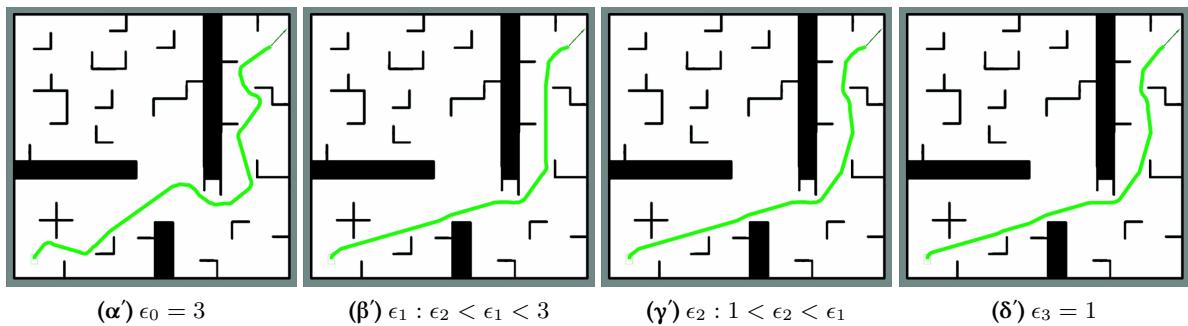
Σχήμα 5.10: Κατασκευή μονοπατιών με τον αλγόριθμο A*, για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.



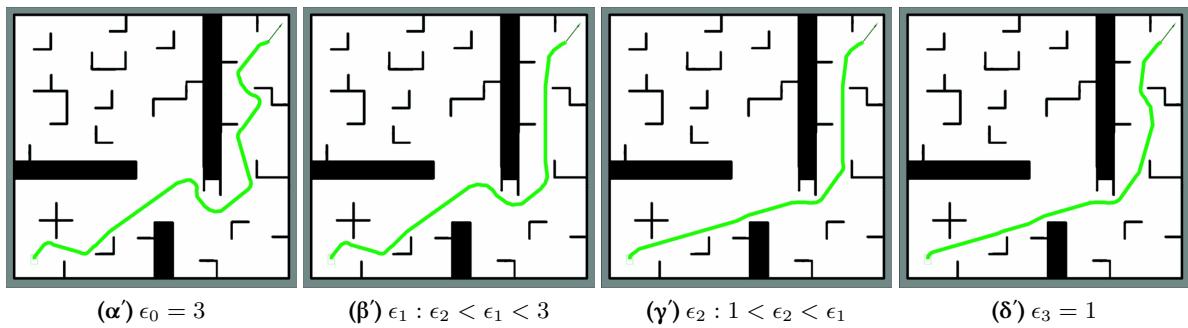
Σχήμα 5.11: Κατασκευή μονοπατιών με τον αλγόριθμο ARA^* , για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.



Σχήμα 5.12: Κατασκευή μονοπατιών με τον αλγόριθμο AD^* , για τις πόζες-στόχους p_i , $i = 1, \dots, 4$.



Σχήμα 5.13: Διαδοχική βελτιστοποίηση μονοπατιού με τον αλγόριθμο ARA^* .



Σχήμα 5.14: Διαδοχική βελτιστοποίηση μονοπατιού με τον αλγόριθμο AD^* .

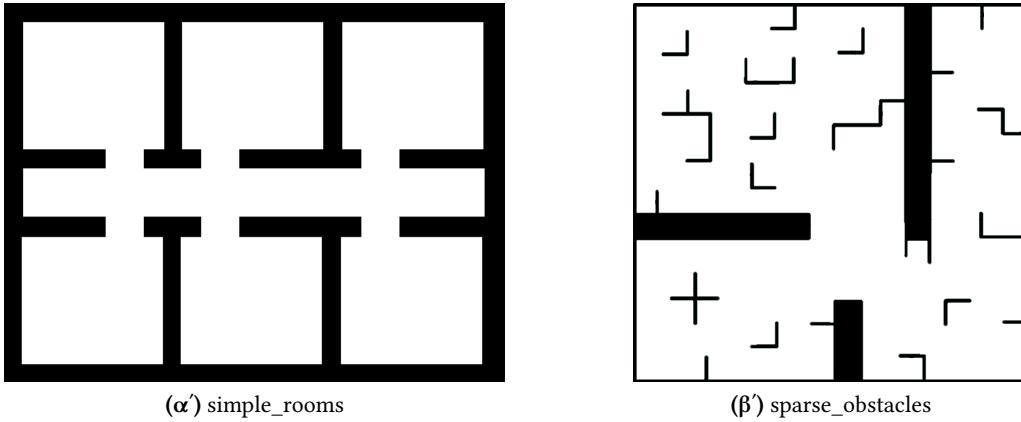
Πίνακας 5.4: Αποτελέσματα των πειραμάτων κατασκευής μονοπατιών.

	Dijkstra		A*		ARA*		AD*		
	$T[s]$	$s[m]$	$T[s]$	$s[m]$	$s_{init}[m]$	$s_{final}[m]$	$s_{init}[m]$	$s_{final}[m]$	
Στόχοι	p_1	0.035	6.02	0.027	6.02	6.68	5.91	7.45	5.92
	p_2	0.060	11.78	0.060	11.78	13.56	11.554	12.62	11.25
	p_3	0.110	19.34	0.090	19.34	19.85	18.73	23.26	18.66
	p_4	0.110	17.69	0.110	17.77	21.36	17.25	20.42	17.57

Με βάσει τα παραπάνω αποτελέσματα κρίνεται ότι οι αλγόριθμοι Dijkstra και A* έχουν παραπλήσια συμπεριφορά, με τον αλγόριθμο Dijkstra να βρίσκει πάντα την βέλτιστη λύση, αλλά με τον αλγόριθμο A* να βρίσκει λύση σε μικρότερο χρόνο. Στην προκειμένη περίπτωση, εφόσον δεν υπάρχει σημαντικός περιορισμός στον χρόνο κατασκευής του μονοπατιού, μπορεί να χρησιμοποιηθεί οποιοσδήποτε από τους δύο αλγορίθμους, με προτίμηση στον αλγόριθμο A*, λόγω του μικρότερου υπολογιστικού φόρτου, λόγω της πιο στοχευμένης αναζήτησης. Αντίστοιχα, όσον αφορά τους αλγορίθμους ARA* και AD* παρατηρείται, επίσης παραπλήσια στατική συμπεριφορά, και επομένως μπορεί να γίνει επιλογή μεταξύ οποιουδήποτε από τους δύο. Φυσικά, το πλεονέκτημα του AD*, έναντι του ARA* έγκειται στην δυναμική προσαρμογή του μονοπατιού σε περίπτωση που έχει αλλάξει επαρκώς το περιβάλλον.

5.3.2 Παραμόρφωσης Μονοπατιού με τον Αλγορίθμου Reeds-Shepp Band

Τα πειράματα παραμόρφωσης μονοπατιού με τον αλγόριθμο Reeds-Shepp Band πραγματοποιήθηκαν εξ ολοκλήρου στον 2D προσομοιωτή STDR, σε αραιό και πυκνό περιβάλλον, όσον αφορά τα εμπόδια. Συγκεκριμένα, χρησιμοποιήθηκαν τα περιβάλλοντα *simple_rooms* και *sparse_obstacles* που παρέχονται από τον προσομοιωτή STDR και τα οποία παρουσιάζονται στο σχήμα 5.15. Επίσης, επιλέχθηκε να χρησιμοποιηθεί και πάλι ο έτοιμος χάρτης με ιδανικό εντοπισμό θέσης, χωρίς να εμπλέκεται το πρόβλημα της χαρτογράφησης και του εντοπισμού θέσης στην προκειμένη περίπτωση, ενώ, παράλληλα χρησιμοποιήθηκε σαρωτής λέιζερ με οπτικό πεδίο 360° , αντί για 240° για την απαλούφη άγνωστων τμημάτων στην νεκρό εύρος γωνιών του ρομπότ, του τοπικού χάρτη κόστους.



Σχήμα 5.15: Τα περιβάλλοντα του προσομοιωτή STDR που χρησιμοποιήθηκαν για τα πειράματα του αλγορίθμου Reeds-Shepp Band.

Στόχος των πειραμάτων, της ενότητας αυτής, είναι η παρατήρηση της στατικής συμπεριφοράς του αλγορίθμου. Δηλαδή, το ρομπότ τοποθετείται σε προκαθορισμένες πόζες, χωρίς να κινείται και του δίνονται στόχοι χειροκίνητα, για την παραμόρφωση του ολικού μονοπατιού σε ελαστική ζώνη και την κατασκευή του αντίστοιχου τοπικού μονοπατιού, βάσει μονοπατιών Reeds-Shepp, όπου το

ολικό μονοπάτι παράγεται μέσω του αλγορίθμου A* που υλοποιεί το plugin global_planner. Επίσης, επιλέχθηκε να γίνει μετατροπή ολόκληρης της ελαστικής ζώνης σε ζώνη Reeds-Shepp μέσω ένωσης των ενδιάμεσων σημείων αυτής με μονοπάτια Reeds-Shepp και όχι μόνο των δύο πρώτων, για την καλύτερη αποτύπωση της συμπεριφοράς του αλγορίθμου.

Τα πρώτα πειράματα πραγματοποιήθηκαν στον ελεύθερο χώρο ενός δωματίου του περιβάλλοντος simple_rooms, με στόχο την παρατήρηση της συμπεριφοράς του αλγορίθμου στον ελεύθερο χώρο, χωρίς την παρεμβολή εμποδίων. Επιλέχθηκαν συνολικά τέσσερις τυχαίες διαφορετικές πόζες ως στόχοι που δόθηκαν στο ρομπότ για την εξέταση της συμπεριφοράς του αλγορίθμου υπό διαφορετικές αντιπροσωπευτικές πιθανές περιπτώσεις. Τα αποτελέσματα των πειραμάτων αυτών, παρουσιάζονται στο σχήμα 5.16.



Σχήμα 5.16: Πειράματα παραμόρφωσης μονοπατιού με τον αλγόριθμο Reeds-Shepp Band σε ελεύθερο χώρο στο περιβάλλον simple_rooms.

Η δεύτερη κατηγορία πειραμάτων πραγματοποιήθηκε όμοια με την πρώτη, αλλά σε περιβάλλον πυκνό από εμπόδια, για την εξέταση της συμπεριφοράς του αλγορίθμου Reeds-Shepp Band, υπό την παρουσία εμποδίων. Τα αποτελέσματα παρουσιάζονται στο σχήμα 5.17.



Σχήμα 5.17: Πειράματα παραμόρφωσης μονοπατιού με τον αλγόριθμο Reeds-Shepp Band, υπό την επίδραση εμποδίων, στο περιβάλλον sparse_obstacles.

5.3.3 Πειράματα Διάσχισης Μονοπατιού

Στην παρούσα ενότητα παρουσιάζονται τα πειράματα που πραγματοποιήθηκαν για την αξιολόγηση του αλγορίθμου διάσχισης μονοπατιού σε τρεις περιπτώσεις, υπό διαφορετικές συνθήκες. Το πρώτο πείραμα και πιο απλό πραγματοποιήθηκε σε μία πίστα με ομαλό έδαφος και διαδοχικά εμπόδια για την εξέταση της συμπεριφορά του αλγορίθμου, χωρίς την επίδραση φαινομένων ολίσθησης. Το δεύτερο πείραμα πραγματοποιήθηκε σε αντίστοιχη πίστα, αλλά με την προσθήκη ενός συνόλου από ράμπες για την προσέγγιση ανώμαλου εδάφους, με στόχο την εξέταση της συμπεριφοράς του αλγορίθμου υπό την επίδραση φαινομένων ολίσθησης. Τέλος, το τρίτο και τελευταίο πείραμα πραγματοποιήθηκε σε μία πίστα η οποία αποτελείται από έναν μακρόστενο διάδρομο με ομοιόμορφα κεκλιμένο επίπεδο για την εξέταση της συμπεριφοράς του αλγορίθμου υπό την επίδραση κάθετης πλευρικής ολίσθησης.

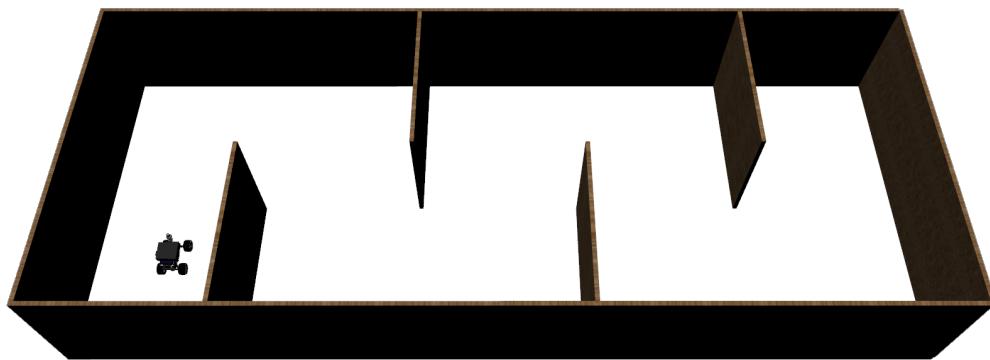
Για την πραγματοποίηση των πειραμάτων, χρησιμοποιήθηκε ο 3D προσομοιωτής Gazebo για την προσομοίωση των παραπάνω περιβαλλόντων. Επίσης, χρησιμοποιήθηκε το σύστημα χαρτογράφησης και εντοπισμού θέσης με τον αλγόριθμο SLAM και το σύστημα αυτόνομης πλοιήγησης με δυναμική ανακατασκευή μονοπατιού, αλλά με σταθερό ολικό μονοπάτι σε συνδυασμό με τον αλγόριθμο

διάσχισης μονοπατιού με ασαφή λογική.

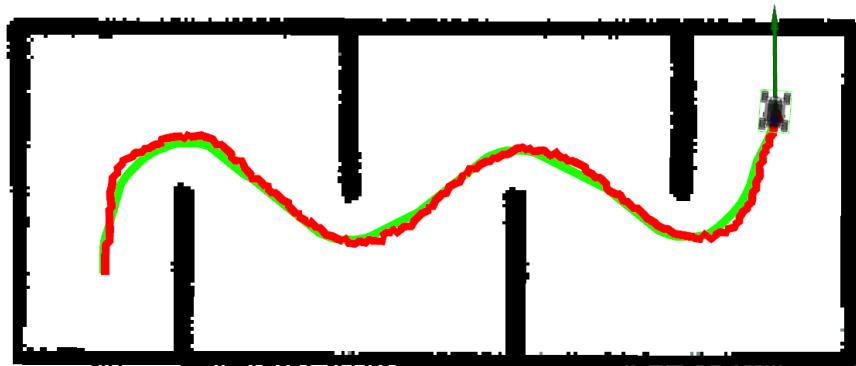
Για την αξιολόγηση των πειραμάτων θα χρησιμοποιηθούν ως μετρικές, η ελάχιστη απόσταση $d_R P$ του ρομπότ από το μονοπάτι, όπως επίσης και το σφάλμα προσανατολισμού E_o μεταξύ της πόζας του ρομπότ και της πόζας του τρέχοντος υπό-στόχου, πάνω στο μονοπάτι. Παράλληλα, θα παρουσιαστούν και τα σφάλματα θέσης E_p , απόκλισης γωνίας E_a , πλευρικής απόκλισης E_y , όπως επίσης και οι γωνίες στρέψης των τροχών καθ' όλη τη διάρκεια των πειραμάτων.

Διάσχιση Μονοπατιού σε Ομαλό Έδαφος

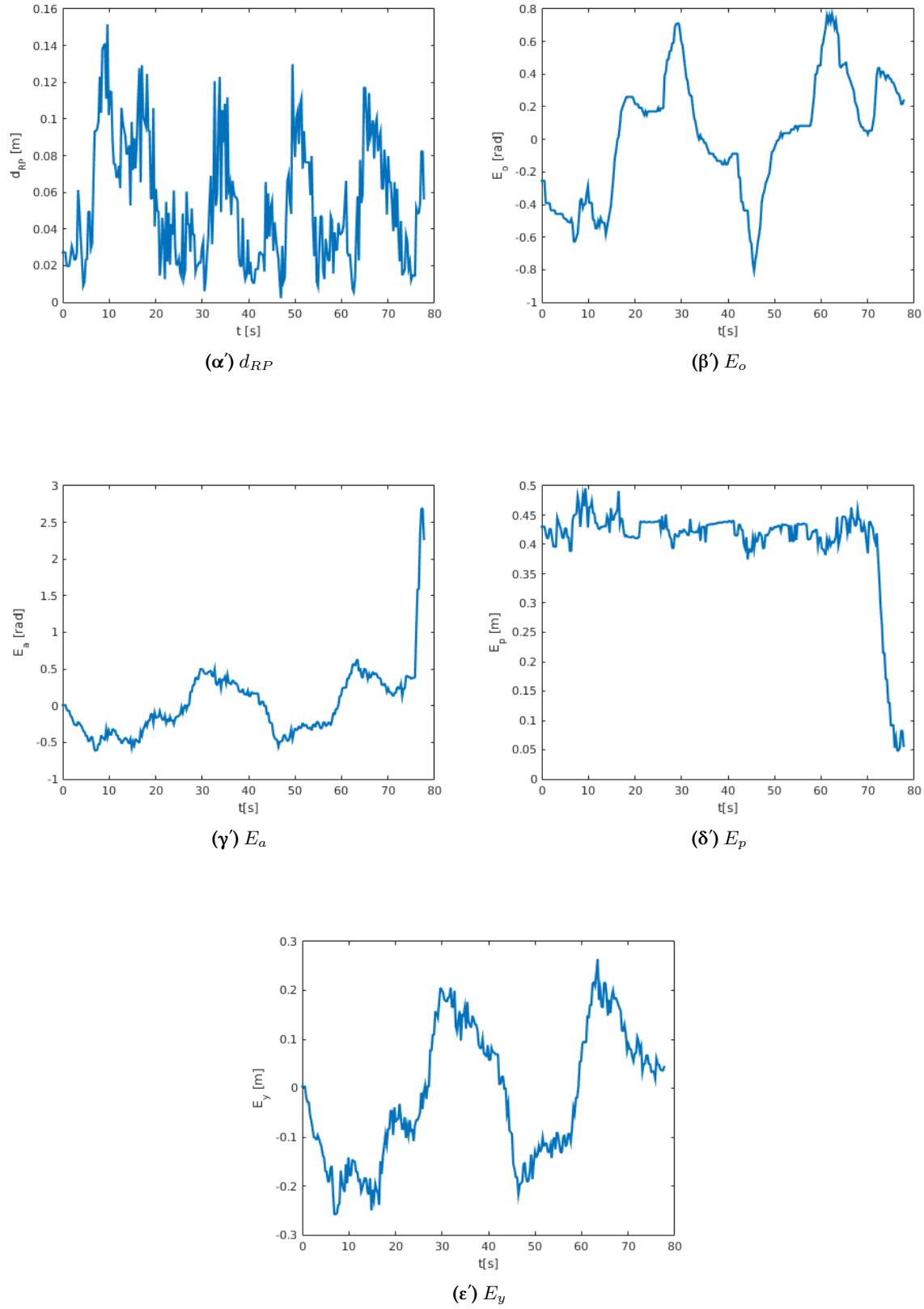
Για το πείραμα διάσχισης μονοπατιού σε ομαλό έδαφος σχεδιάστηκε και χρησιμοποιήθηκε το περιβάλλον *zic_zac_room_no_ramps* που παρουσιάζεται στο σχήμα 5.18. Για την πραγματοποίηση του εν λόγω πειράματος, εφόσον ο χάρτης του περιβάλλοντος δεν είναι εκ των προτέρων γνωστός, πραγματοποιήθηκε μία αρχική εξερεύνηση για την δημιουργία του πλήρους χάρτη του περιβάλλοντος και έπειτα πραγματοποιήθηκε το πείραμα για την μετάβαση από την μία πλευρά του χάρτη στην άλλη, με στόχο την παρακολούθηση της συμπεριφοράς του αλγορίθμου διάσχισης μονοπατιού. Στο σχήμα 5.19 παρουσιάζονται το ολικό μονοπάτι που ακολούθησε το ρομπότ, όπως επίσης και η τροχιά που πραγματοποίησε για την επίτευξη του δεδομένου στόχου, ενώ στο σχήμα 5.20 παρουσιάζονται τα σχετικά σφάλματα που προέκυψαν κατά την διάρκεια του πειράματος. Τέλος, στο σχήμα 5.21 παρουσιάζονται και οι γωνίες στρέψης των μπροστινών και πίσω τροχών του οχήματος καθ' όλη τη διάρκεια του πειράματος.



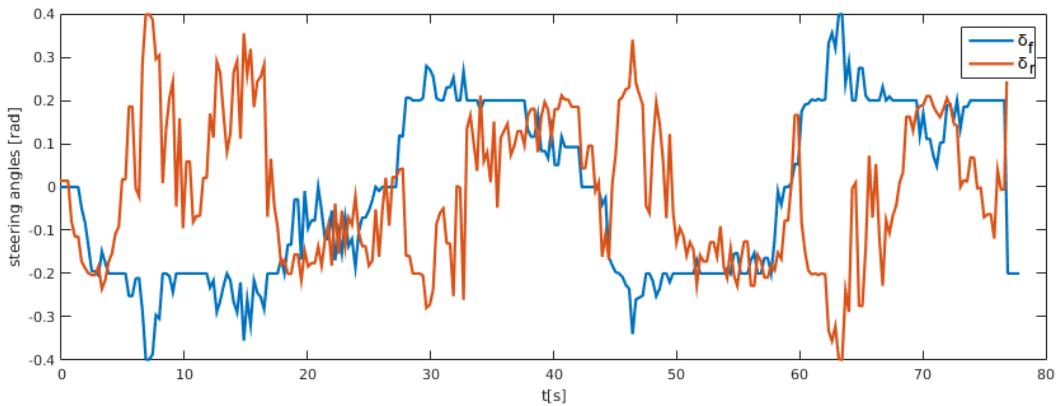
Σχήμα 5.18: Το περιβάλλον *zic_zac_room_no_ramps*



Σχήμα 5.19: Ο χάρτης του περιβάλλοντος *zic_zac_room_no_ramps*, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).



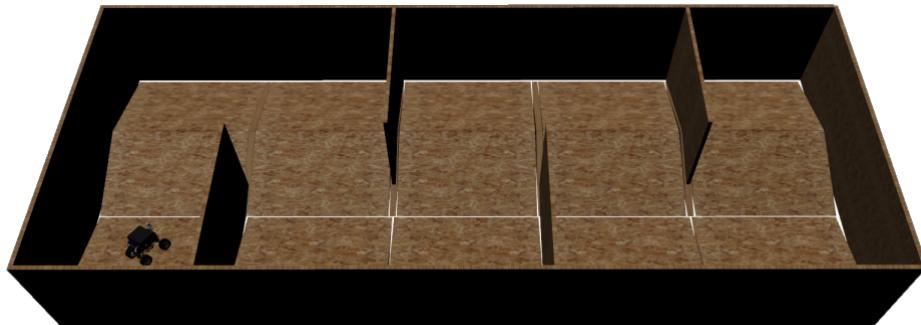
Σχήμα 5.20: Διαγράμματα σφαλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον *zic_zac_room_no_ramps*.



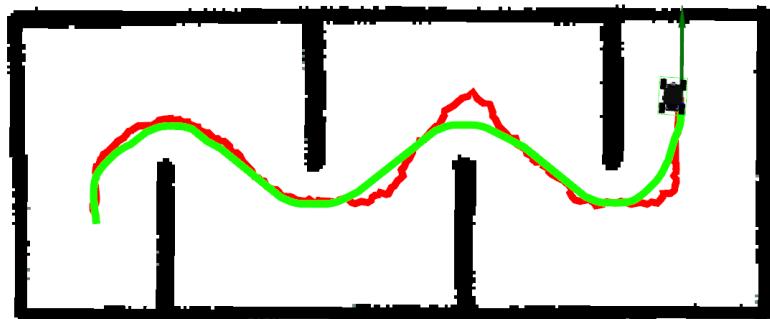
Σχήμα 5.21: Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον *zic_zac_room_no_ramps*.

Διάσχιση Μονοπατιού σε Ανώμαλο Έδαφος

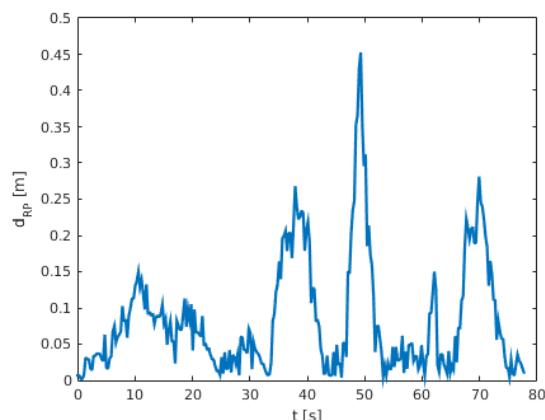
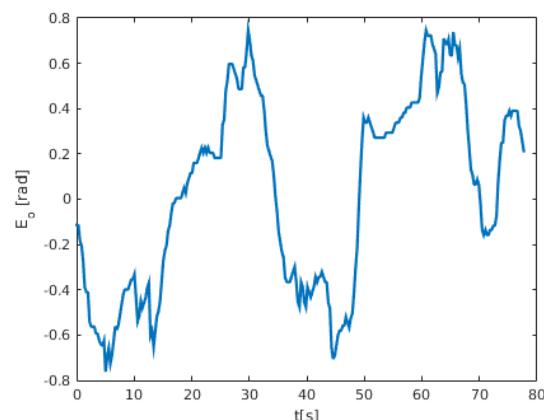
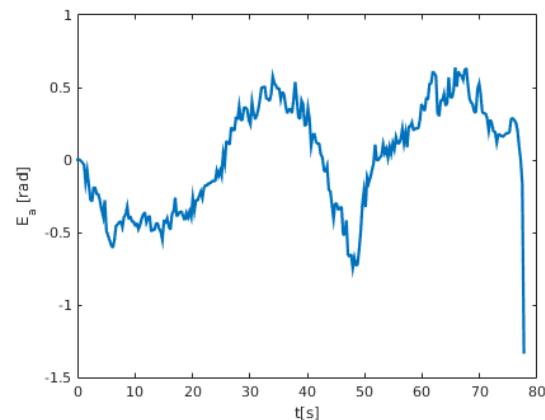
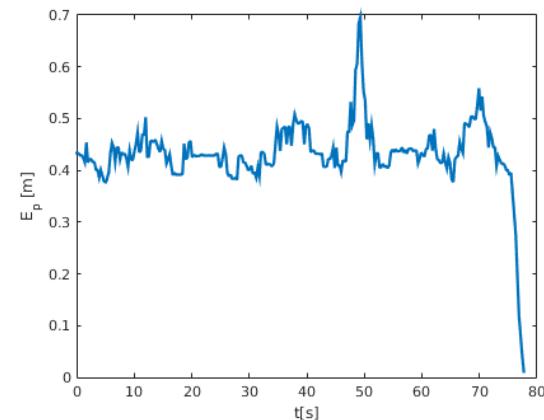
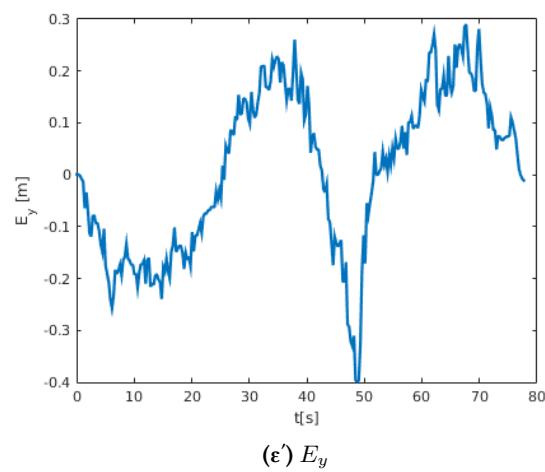
Για το πείραμα διάσχισης μονοπατιού σε ανώμαλο έδαφος σχεδιάστηκε και χρησιμοποιήθηκε το περιβάλλον *zic_zac_room*, όπως αυτό παρουσιάζεται στο σχήμα 5.22, ενώ το πείραμα εκτελέστηκε όμοια με το προηγούμενο. Στο σχήμα 5.23 παρουσιάζεται το ολικό μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ, όπως επίσης και η τροχιά που πραγματοποίησε για την επίτευξη του δεδομένου στόχου, ενώ στο σχήμα 5.24 παρουσιάζονται τα αντίστοιχα σφάλματα που προέκυψαν κατά την διάρκεια του πειράματος. Τέλος, στο σχήμα 5.25 παρουσιάζονται και οι γωνίες στρέψης των μπροστινών και πίσω τροχών του οχήματος καθ' όλη τη διάρκεια του πειράματος.



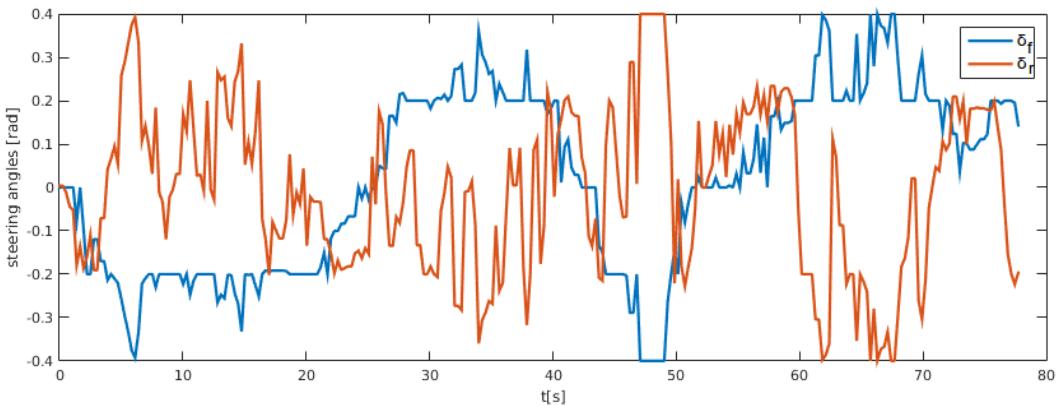
Σχήμα 5.22: Το περιβάλλον *zic_zac_room*.



Σχήμα 5.23: Ο χάρτης του περιβάλλοντος *zic_zac_room*, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).

(α') d_{RP} (β') E_o (γ') E_a (δ') E_p (ε') E_y

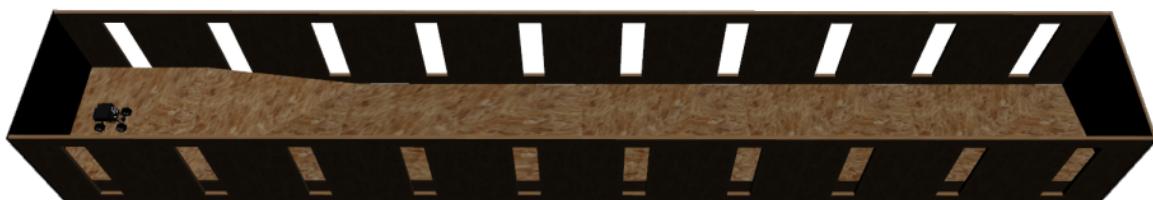
Σχήμα 5.24: Διαγράμματα σφωλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον *zac_zac_room*.



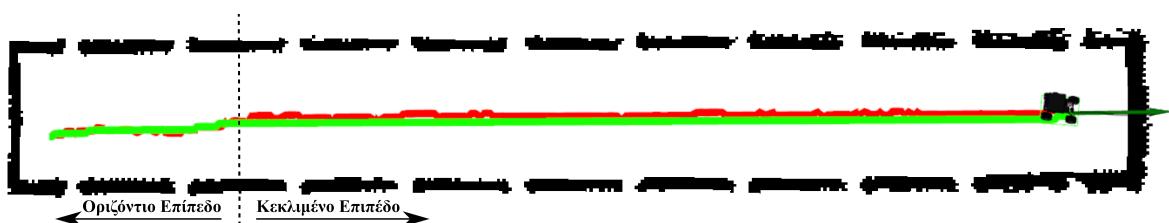
Σχήμα 5.25: Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον *zic_zac_room*.

Διάσχιση Μονοπατιού σε Κεκλιμένο Επίπεδο

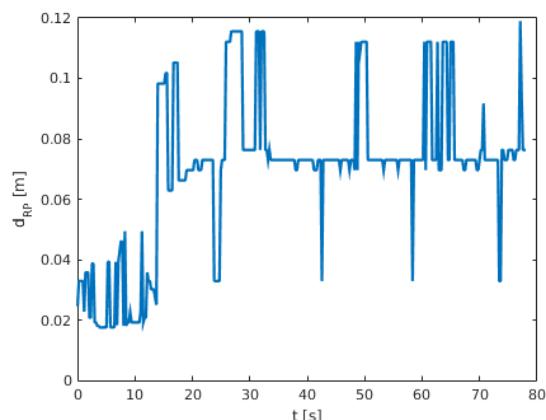
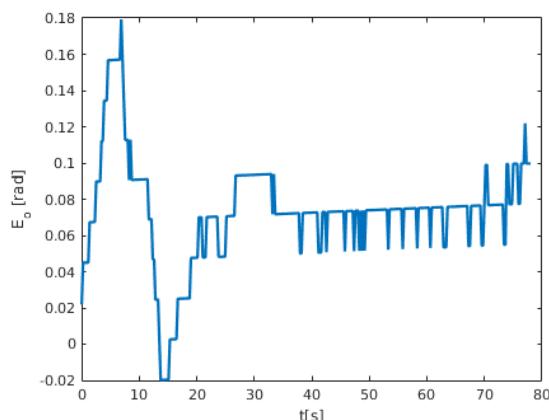
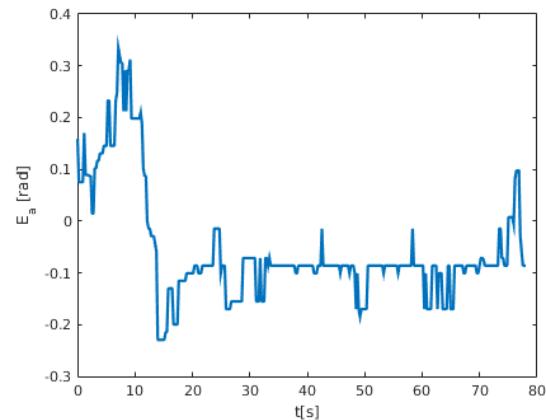
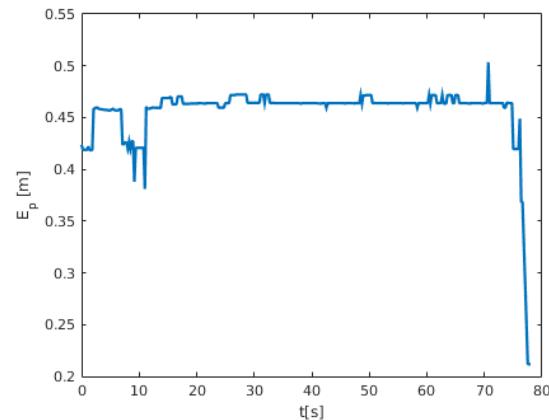
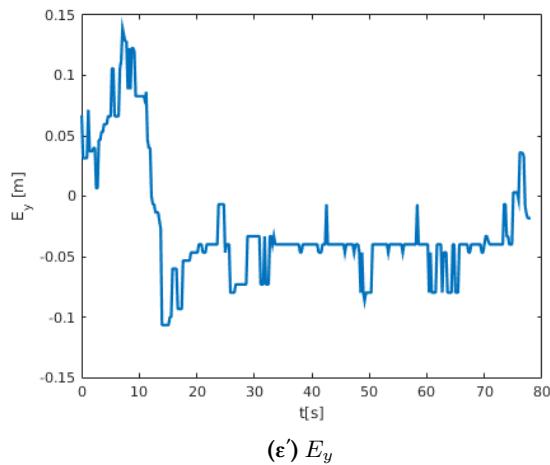
Για το πείραμα διάσχισης μονοπατιού σε κεκλιμένο επίπεδο σχεδιάστηκε και χρησιμοποιήθηκε το περιβάλλον *slope_room*, το οποίο αποτελείται από ένα μακρόστενο δωμάτιο με έδαφος που αποτελείται από ένα μικρό επίπεδο τμήμα στην αρχική θέση του ρομπότ και έπειτα έναν μακρύ διάδρομο με κλίση περίπου 15° . Το εν λόγω πείραμα εμπεριέχει την κίνηση κάθετα στον κεκλιμένο διάδρομο υπό την επίδραση ολίσθησης. Λόγω του μεγάλου μήκους του διαδρόμου, αλλά και την ομοιομορφία του περιβάλλοντος που δεν επιτρέπουν την αξιόπιστη χαρτογράφηση αυτού, επιλέχθηκε να αυξηθεί η εμβέλεια του σαρωτή λέιζερ στα 20m, έτσι ώστε να μπορεί να εντοπίσει το τέρμα της πίστας και να μπορεί έτσι το ρομπότ να αναγνωρίζει την θέση του αξιόπιστα. Στην συνέχεια, στο σχήμα ?? παρουσιάζεται το ολικό μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ, όπως επίσης και η τροχιά που πραγματοποίησε για την επίτευξη του δεδομένου στόχου, ενώ στο σχήμα 5.28 παρουσιάζονται τα σχετικά σφάλματα που προέκυψαν κατά την διάρκεια του πειράματος. Τέλος, στο σχήμα 5.29 παρουσιάζονται και οι γωνίες στρέψης των μπροστινών και πίσω τροχών του οχήματος καθ' όλη τη διάρκεια του πειράματος.



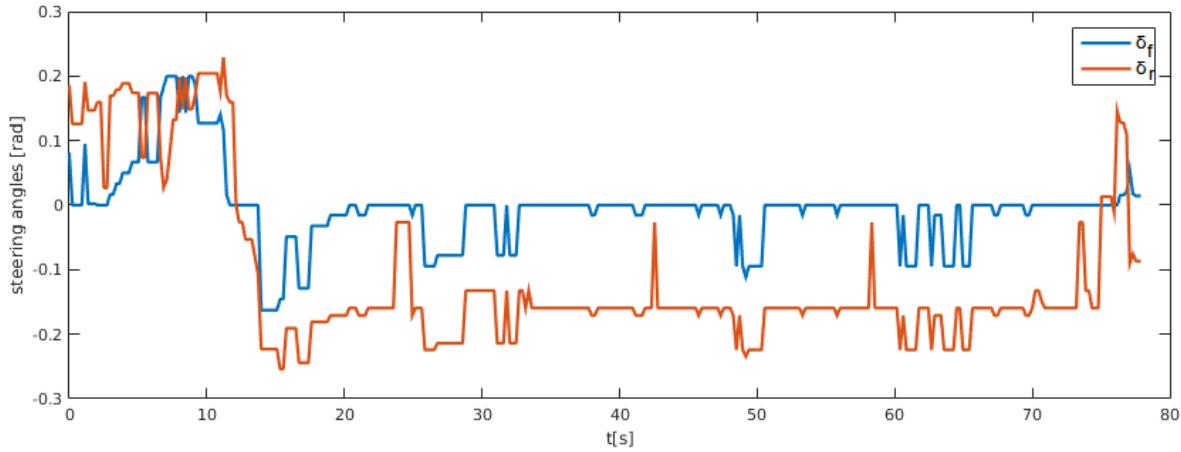
Σχήμα 5.26: Το περιβάλλον *slope_room*.



Σχήμα 5.27: Ο χάρτης του περιβάλλοντος *slope_room*, το μονοπάτι που κλήθηκε να ακολουθήσει το ρομπότ (πράσινο) και η τροχιά που διάνυσε (κόκκινο).

(α') d_{RP} (β') E_o (γ') E_a (δ') E_p (ε') E_y

Σχήμα 5.28: Διαγράμματα σφαλμάτων πειράματος διάσχισης μονοπατιού στο περιβάλλον *slope_room*.



Σχήμα 5.29: Διάγραμμα των γωνιών στρέψης των μπροστινών και πίσω τροχών κατά την πραγματοποίηση του πειράματος διάσχισης μονοπατιού στο περιβάλλον *slope_room*.

5.3.4 Πειράματα Αυτόνομης Πλοϊγησης με Δυναμική Παραμόρφωση και Δυναμική Ανακατασκευή Μονοπατιού

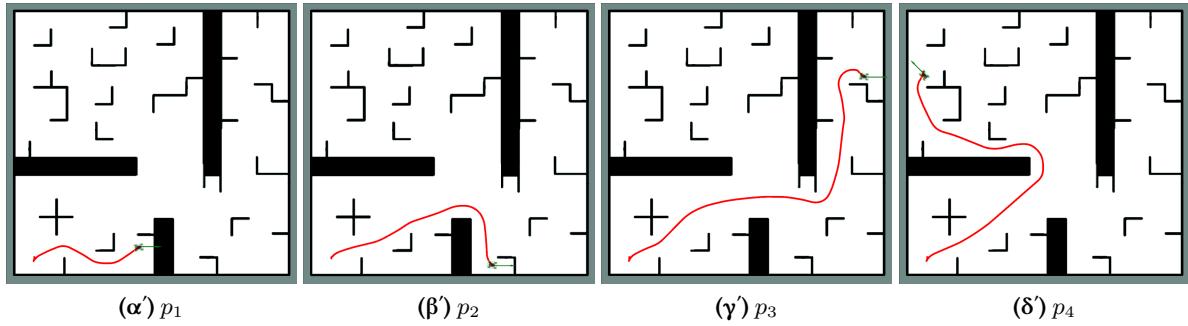
Στην παρούσα ενότητα παρουσιάζεται μία σύγκριση των δύο εξεταζόμενων συστημάτων αυτόνομης πλοϊγησης με δυναμική παραμόρφωση μονοπατιού και δυναμική ανακατασκευή μονοπατιού, μέσω ορισμού στόχων και σύγκριση της απόκρισης κάθε συστήματος. Συγκεκριμένα, για την διεξαγωγή την πειραμάτων χρησιμοποιήθηκε ο προσομοιωτής STDR και το περιβάλλον *sparse_obstacles* που χρησιμοποιήθηκε και σε προηγούμενα πειράματα, ενώ ως στόχοι ορίστηκαν οι πόζες που παρουσιάστηκαν στον πίνακα 5.3 και χρησιμοποιήθηκαν στο πειράματα κατασκευής μονοπατιών της ενότητας 5.3.1.

Το ρομπότ τοποθετήθηκε στην ίδια πόζα $p_0 = [1.5, 1.5, 90^\circ]$, με τα πειράματα κατασκευής μονοπατιών και καταγράφηκε η τροχιά που διάνυσε για κάθε έναν από τους τέσσερις στόχους και για τα δύο εξεταζόμενα συστήματα αυτόνομης πλοϊγησης. Για την αξιολόγηση των πειραμάτων ορίστηκαν οι μετρικές T_G και s_G που δηλώνουν τον χρόνο που απαιτείται για την μετάβαση προς έναν δεδομένο στόχο και το μήκος της τροχιάς που διανύθηκε, αντίστοιχα. Στο σχήμα 5.30 παρουσιάζεται οι τροχιές που διάνυσε το ρομπότ για την μετάβαση από την αρχική του θέση προς κάθε έναν από τους τέσσερις στόχους, μέσω του συστήματος αυτόνομης πλοϊγησης με δυναμικής παραμόρφωση μονοπατιού, ενώ στο σχήμα 5.31 παρουσιάζονται οι αντίστοιχες τροχιές, βάσει του συστήματος δυναμικής ανακατασκευής μονοπατιού. Τέλος, στον πίνακα ?? παρουσιάζονται τα αποτελέσματα των παραπάνω πειραμάτων για κάθε ένα από τα εν λόγω συστήματα αυτόνομης πλοϊγησης.

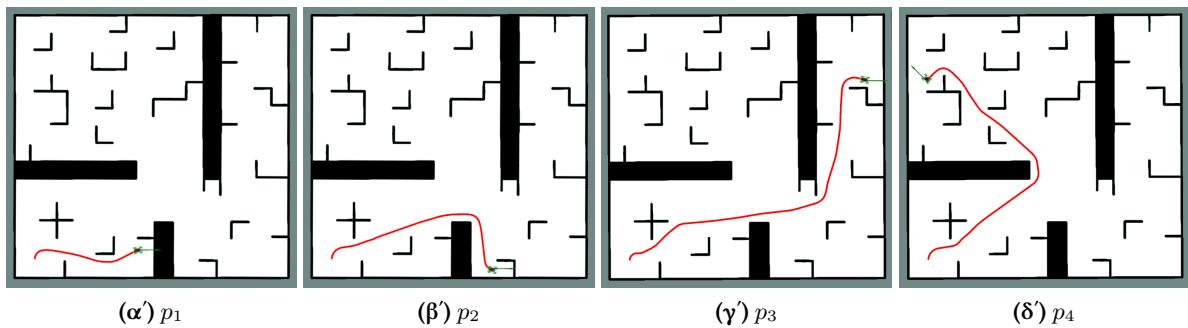
Πίνακας 5.5: Αποτελέσματα των πειραμάτων αυτόνομης πλοϊγησης με δυναμική παραμόρφωση μονοπατιού (DPM) και δυναμική ανακατασκευή μονοπατιού (DPR).

	DPM		DPR		
	$T_G[s]$	$s_G[m]$	$T_G[s]$	$s_G[m]$	
Στόχοι	p_1	38.80	6.93	35.2	6.06
	p_2	65.00	12.23	63.20	11.6
	p_3	113.20	20.76	97.08	18.87
	p_4	102.20	19.06	98.20	17.57

Βάσει του πίνακα 5.5, προκύπτει ότι το σύστημα αυτόνομης πλοϊγησης με δυναμική ανακατασκευή έφτασε στον επιθυμητό στόχο σε μικρότερο χρόνο και διάνυσε μικρότερη απόσταση από το



Σχήμα 5.30: Αυτόνομη πλοιόγηση με δυναμική παραμόρφωση μονοπατιού, για μετάβαση στις πόζες-στόχους p_i , $i = 1, \dots, 4$.



Σχήμα 5.31: Αυτόνομη πλοιόγηση με δυναμική ανακατασκευή μονοπατιού, για μετάβαση στις πόζες-στόχους p_i , $i = 1, \dots, 4$.

σύστημα αυτόνομης πλοιόγησης με δυναμική παραμόρφωση μονοπατιού. Το γεγονός αυτό, εξηγείται, αν παρατηρήσει κανείς ότι το σύστημα με δυναμική παραμόρφωση μονοπατιού κινήθηκε σε πιο ομαλή, αλλά και ασφαλέστερη τροχιά, δηλαδή πιο μακριά από τα εμπόδια απ' ότι το πρώτο. Παράλληλα, αξίζει να σημειωθεί ότι το όχημα κινήθηκε με αρκετά χαμηλή ταχύτητα ($0.2m/s$), γεγονός που επέτρεψε την αξιόπιστη λειτουργία του συστήματος αυτόνομης πλοιόγησης με δυναμική ανακατασκευή μονοπατιού, μέσω ανακατασκευής σε επαρκή συχνότητα. Αντίθετα, το σύστημα αυτόνομης πλοιόγησης με δυναμική παραμόρφωση μονοπατιού, έχει δοκιμαστεί και με υψηλότερες ταχύτητες ($0.4 - 0.5m/s$) με αντίστοιχα αξιόπιστη συμπεριφορά.

5.4 Αξιολόγηση Απόδοσης Υπολογιστή Odroid-XU4

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία παρουσιάστηκε η ανάπτυξη ενός αυτόνομου ρομποτικού οχήματος με τετραδιεύθυνση και τετρακίνηση, τόσο ως προς το υλικό, αλλά όσο και ως προς το λογισμικό που απαιτήθηκε για την επίτευξη της αυτόνομης λειτουργίας αυτού. Η ρομποτική πλατφόρμα Monstertruck που αναπτύχθηκε αποτελεί ένα ρομποτικό όχημα, ικανό να εξερευνήσει, αυτόνομα, ένα άγνωστο περιβάλλον, να το χαρτογραφήσει και να πλοηγηθεί αποτελεσματικά μέσα σε αυτό.

Στο Κεφάλαιο 2 παρουσιάστηκε ο εξοπλισμός, μηχανολικός, ηλεκτρολογικός και ηλεκτρονικός, που εγκαταστάθηκε στην ρομποτική πλατφόρμα Monstertruck, όπως επίσης και η σκοπιμότητα του, για τη επίτευξη της παραπάνω λειτουργικότητας. Έπειτα, έγινε μία κινηματική ανάλυση για την εξαγωγή των σχέσεων μεταξύ των συστημάτων μετάδοσης κίνησης από τους κινητήρες και σερβοκινητήρες του οχήματος στους τροχούς, όσον αφορά την κίνηση ώθησης και στρέψης αυτών. Επίσης, παρουσιάστηκε η κινηματική ανάλυση του οχήματος, μέσω προσαρμογής του ιδανικού κινηματικού μοντέλου τετραδιεύθυνσης, όσον αφορά την κίνηση του στο επίπεδο, όπου παράχθηκε μία σχέση μεταξύ της επιθυμητής τροχιάς, με την αντίστοιχη απαιτούμενη κίνηση των τροχών. Βάσει των πειραμάτων που πραγματοποιήθηκαν για την αξιολόγηση του κινηματικού μοντέλου της ρομποτικής πλατφόρμας Monstertruck, συμπεραίνεται ότι λόγω του μη ιδανικού μηχανισμού στρέψης των τροχών, που έχει σαν αποτέλεσμα την ολισθαίνουσα κίνηση των τροχών, το κινηματικό μοντέλο παρουσιάζει μία μικρή απόκλιση από την ιδανική συμπεριφορά, όσον αφορά την κίνηση σε τροχιές αρνητικής τετραδιεύθυνσης. Αντιθέτως για την περίπτωση των τροχιών θετικής τετραδιεύθυνσης η απόκλιση από την ιδανική συμπεριφορά είναι αμελητέο έως μηδενική, λόγω της προσεγγιστικά παράλληλης στρέψης των τροχών του οχήματος, γεγονός που καθιστά την ρομποτική πλατφόρμα Monstertruck, καταλληλότερη για τροχιές αρνητικής τετραδιεύθυνσης, από ένα όχημα με ιδανικό μοντέλο τετραδιεύθυνσης.

Στο Κεφάλαιο 3 παρουσιάστηκαν οι αλγόριθμοι εκτίμησης κατάστασης, χαρτογράφησης και εντοπισμού θέσης, που χρησιμοποιήθηκαν στην ρομποτική πλατφόρμα Monstertruck, όπως επίσης και οι αλγόριθμοι που αναπτύχθηκαν και χρησιμοποιήθηκαν για την επίτευξη της αυτόνομης πλοιόγησης αυτής σε ένα άγνωστο περιβάλλον.

Όσον αφορά την ταυτόχρονη χαρτοφράφηση και εντοπισμό θέσης, χρησιμοποιήθηκαν οι αλγόριθμοι CRSM-SLAM και Gmapping. Ο αλγόριθμος CRSM-SLAM αποδείχθηκε πολύ εύκολος στην χρήση και παραμετροποίηση, με πολύ μικρές απαιτήσεις σε υπολογιστικούς πόρους και απαιτούμενες πηγές δεδομένων. Συγκεκριμένα, ο αλγόριθμος απαιτεί μόνο σαρώσεις του περιβάλλοντος μέσω ενός σαρωτή λειζερ, για την χαρτογράφηση και τον εντοπισμό θέσης, μέσω αντιστοίχισης των διαδοχικών σαρώσεων. Το γεγονός αυτό τον καθιστά κατάλληλο για κλειστούς χώρους και εύρωστο όσον αφορά φαινόμενα ολίσθησης, αλλά και σε περίπτωση που δεν υπάρχει άλλη πηγή πληροφορίας για τον εντοπισμό θέσης, αλλά παράλληλα ακατάλληλο για την περίπτωση ομοιόμορφου περιβάλλοντος διαστάσεων μεγαλύτερων της εμβέλειας του σαρωτή λειζερ. Αντιθέτως, ο αλγόριθμος Gmapping αποτελεί μία αρκετά πιο απαιτητική λύση, όσον αφορά την παραμετροποίηση, τους υπολογιστικούς πόρους και τις πηγές δεδομένων, μιας και χρησιμοποιεί σαρώσεις του περιβάλλοντος αλλά και την

εκτίμηση της κατάστασης του ρομπότ, βάσει ενός συστήματος οδομετρίας. Επομένως, παράγει μία πιο αξιόπιστη αναπαράσταση του περιβάλλοντος και εκτίμηση της κατάστασης του ρομπότ, ενώ παράλληλα επιτρέπει την χαρτογράφηση σε περιβάλλον, οι διαστάσεις του οποίου ξεπερνούν την εμβέλεια του λέιζερ. Παρόλα αυτά, η αξιοπιστία του αλγορίθμου Gmapping εξαρτάται σε μεγάλο βαθμό από την αξιοπιστία του συστήματος οδομετρίας και άρα είναι επιρρεπείς σε σφάλματα λόγω μη ιδανικού συστήματος οδομετρίας, αλλά και σε περίπτωση φαινομένων ολίσθησης. Για την αξιολόγηση της συμπεριφοράς των δύο αλγορίθμων χαρτογράφησης χρησιμοποιήθηκαν περιβάλλοντα συμβατά και με τους δύο αλγορίθμους, βάσει των οποίων κατασκευάστηκαν παραπλήσιοι χάρτες για την αναπαράσταση αυτών, που δεν ήταν δίχως ατέλειες, αλλά κρίθηκαν επαρκείς για τις απαιτήσεις του προβλήματος που εξετάζει η διπλωματικής εργασίας.

Για την αυτόνομη πλοήγηση της ρομποτικής πλατφόρμας Monstertruck παρουσιάστηκαν δύο συστήματα που επιλύουν το πρόβλημα της αυτόνομης πλοήγησης ενός οχήματος με κινηματικό μοντέλο τετραδιεύθυνσης. Το πρώτο σύστημα, παράγει ένα ολικό μονοπάτι, στον ελεύθερο χώρο, μεταξύ της αρχικής θέσης του ρομπότ και του δεδομένου στόχου και εφαρμόζει μία ακολουθία από στάδια επεξεργασίας για την παραμόρφωση του ολικού μονοπατιού σε ένα τοπικό ασφαλές και κινηματικά εφικτό μονοπάτι που μπορεί να ακολουθήσει το ρομπότ. Το δεύτερο σύστημα παράγει ένα αρχικό ολικό μονοπάτι, το οποίο έπειτα, ανακατασκευάζει συνεχώς, δυναμικά, όσο κινείται το ρομπότ. Για την διάσχιση του παραμορφωμένου ολικού μονοπατιού στην πρώτη περίπτωση, ή του ανακατασκευαζόμενου ολικού μονοπατιού στην δεύτερη περίπτωση, παρουσιάστηκε ένας αλγόριθμος διάσχισης μονοπατιού, βασισμένος σε ασαφή λογική και κατάλληλος για οχήματα με κινηματικό μοντέλο τετραδιεύθυνσης ή Ackermann. Βάσει των πειραμάτων που πραγματοποιήθηκαν στο Κεφάλαιο 5, για την αξιολόγηση και σύγκριση των δύο συστημάτων αυτόνομης πλοήγησης, παρατηρήθηκε επιτυχής πλοήγηση και από τα δύο συστήματα με το δεύτερο να πετυχαίνει μικρότερο χρόνο πλοήγησης και μήκος τροχιάς για την επίτευξη ενός δεδομένου στόχου, αλλά με το δεύτερο να εκτελεί πιο ασφαλείς τροχιές μακριά από τα εμπόδια. Επίσης, το σύστημα με δυναμική παραμόρφωση μονοπατιού παρουσίασε μεγαλύτερη ευρωστία, με την επίτευξη όλων των δυνατών στόχων και αποφυγή συγκρούσεων, αλλά αρκετά ασταθές τοπικό μονοπάτι σε περιπτώσεις που απαιτείται ελιγμός αναστροφής σε πολύ πυκνό από εμπόδια περιβάλλον. Αντιθέτως, το σύστημα με δυναμική ανακατασκευή μονοπατιού, παράγει, δυναμικά, ένα ντετερμινιστικό κινηματικά εφικτό μονοπάτι, αλλά σε πολύ χαμηλή συχνότητα, με αποτέλεσμα, η μέθοδος να μην μπορεί να χρησιμοποιηθεί για υψηλές ταχύτητες, ενώ παράλληλα η απουσία πρόσθετου μηχανισμού αποφυγής εμποδίων στο στάδιο, μεταξύ της κατασκευής μονοπατιού και της διάσχισης αυτού, με υψηλότερη συχνότητα από την κατασκευή μονοπατιού καθιστά το σύστημα αυτό επιρρεπές σε συγκρούσεις, ιδιαίτερα σε περιπτώσεις, που καθυστερεί η ανακατασκευή του ολικού μονοπατιού, λόγω απαίτησης πολύπλοκου ελιγμού σε περιβάλλον πυκνό με εμπόδια.

Όσον αφορά τον αλγόριθμο διάσχισης μονοπατιού με ασαφή λογική που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας, παρατηρήθηκε ότι η επιθυμητή συμπεριφορά του αλγορίθμου επιτεύχθηκε σε μεγάλο βαθμό, με αυτόν να ακολουθεί αρκετά πιστά τα δεδομένα μονοπάτια, ενώ παράλληλα επαναφέρει το ρομπότ στο επιθυμητό μονοπάτι σε περίπτωση σημαντικής απόκλισης, όπως για παράδειγμα σε περίπτωση ολίσθησης, όπως παρουσιάστηκε στα αντίστοιχα πειράματα στην ενότητα 5.3.3. Το μόνο αρνητικό του αλγορίθμου είναι ότι δεν περιλαμβάνει κάποια μέθοδο αναγνώρισης περιπτώσεων σημαντικής απόκλισης από το δοσμένο μονοπάτι, γεγονός που θέτει σε κίνδυνο σύγκρουσης το ρομπότ, ιδιαίτερα σε περιπτώσεις χαμηλής συχνότητας ανανέωσης του μονοπατιού, ενώ παράλληλα δεν διαθέτει ούτε μηχανισμό εντοπισμού επικείμενης σύγκρουσης.

Τέλος, βάσει των παραπάνω, εξάγεται το συμπέρασμα ότι ο στόχος της παρούσας διπλωματικής, για την ανάπτυξη ενός αυτόνομου ρομποτικού οχήματος με κινηματικό μοντέλο τετραδιεύθυνσης και δυνατότητες χαρτογράφησης, εκτίμησης κατάστασης και αυτόνομης πλοήγησης σε άγνωστο περιβάλλον, επιτεύχθηκε σε σημαντικό βαθμό. Παρόλα αυτά, η υλοποίηση του συστήματος της ρομποτικής πλατφόρμας Monstertruck, αφήνει πολλές δυνατότητες βελτίωσης, αλλά ακόμα και επέκτασης

αυτής. Στην ακόλουθη ενότητα παρουσιάζονται ένα σύνολο ιδεών και προτάσεων για βελτίωση της υπάρχουσας υλοποίησης αλλά και επέκταση αυτής με πρόσθετη λειτουργικότητα.

6.2 Μελλοντική Εργασία

Η ρομποτική πλατφόρμα Monstertruck που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας μπορεί να βελτιωθεί βάσει των ακόλουθων προτάσεων, ξεκινώντας από την μηχανολογική κατασκευή έως και καταλήγοντας στο λογισμικό:

- Βελτίωση της μηχανολογικής κατασκευής του ρομποτικού οχήματος με στόχο την πλοήγηση σε περιβάλλοντα, με υψηλότερη δυσκολία προσπελασμότητας, μέσω χρήσης μεγαλύτερων τροχών, μείωση του κέντρου βάρους του οχήματος και αντικατάσταση των πλαστικών τμημάτων του συστήματος μετάδοσης κίνησης, με μεταλλικά.
- Βελτίωση του κινηματικού μοντέλου του οχήματος, μέσω κατασκευής μηχανισμού στρέψης των τροχών που προσεγγίζει την ιδανική συνθήκη τετραδιεύθυνσης 2.42, ή μέσω κατασκευής μηχανισμού ανεξάρτητης στρέψης όλων των τροχών, κάτι που παράλληλα θα προσφέρει και δυνατότητα επιτόπου στροφής, μέσω στρέψης των τροχών κατά τρόπο που να έχει σαν αποτέλεσμα το κέντρο στιγμιαίας περιστροφής (ICR) του ρομπότ να βρίσκεται στο γεωμετρικό κέντρο του.
- Αύξηση της μέγιστης ταχύτητας του οχήματος, χρησιμοποιώντας έναν κινητήρα με υψηλότερη ταχύτητα περιστροφής, ή τροποποιώντας κατάλληλα το σύστημα τετρακίνησης, για μείωση του αντίστοιχου λόγου μετάδοσης.
- Προσθήκη κωδικοποιητών στους τροχούς του οχήματος για την βελτίωση της εκτίμησης της κατάστασης του ρομπότ, που παράγει το σύστημα οδομετρίας του.
- Προσθήκη αισθητήρα βάθους με στόχο τον εντοπισμό εμποδίων που δεν γίνονται αντιληπτά από τον σαρωτή λέιζερ, αλλά και γενικότερα εντοπισμό μη προσπελάσιμων τμημάτων του χώρου, όπως ράμπες μεγάλης κλίσης, σκάλες, γκρεμούς κλπ.
- Χρήση οπτικής οδομετρίας, μέσω κάμερας, για πιο εύρωστη εκτίμηση κατάστασης, αλλά και συμπλήρωση εκτίμησης πλήρους 6-DOF κατάστασης $[x, y, z, yaw, pitch, roll]$, με την προσθήκη εκτίμησης κατακόρυφης μετατόπισης.
- Προσθήκη λειτουργικότητας ρομποτικής όρασης για αναζήτηση σημείων ενδιαφέροντος, ανάλογα με την εφαρμογή.
- Αύξηση της υπολογιστικής ισχύος του ρομπότ, μέσω χρήσης ενός πιο ισχυρού υπολογιστή, ή ενός κατανεμημένου συστήματος υπολογιστών, ανάλογης ισχύος με τον υπολογιστή ODROID-XU4, όπου κάθε ένας θα είναι υπεύθυνος για ένα τμήμα της συνολικής λειτουργικότητας του ρομπότ, ενώ παράλληλα θα μπορούν να επικοινωνούν μεταξύ τους για τον συνδυασμό της παραγόμενης πληροφορίας. Αυτή η πρόταση κρίνεται απαραίτητη για είναι δυνατή η προσθήκη νέων αισθητήρων και λειτουργιών, καθώς, βάσει της υπάρχουσας λειτουργικότητας του ρομπότ και της υπάρχουσας υπολογιστικής ισχύος, δεν είναι δυνατή τέτοια επέκταση του συστήματος, χωρίς σοβαρές συνέπειες για την απόδοση του συστήματος.
- Βελτίωση των δυνατοτήτων χαρτογράφησης και εντοπισμού θέσης με εγκατάσταση σαρωτή λέιζερ με μεγαλύτερη εμβέλεια, από τον υπάρχον αισθητήρα Hokuyo URG-04LX, ή και μεγαλύτερο οπτικό πεδίο, όπως για παράδειγμα ο αισθητήρας RPLIDAR που έχει εμβέλεια 6m και οπτικό πεδίο 360°.
- Επίσης, μπορούν να χρησιμοποιηθούν και αισθητήρες απόστασης (sonar/ir range sensors) για την μέτρηση των αποστάσεων στην νεκρή γωνία του υπάρχοντος σαρωτή λέιζερ, με αποτέλεσμα τον καλύτερο εντοπισμό εμποδίων πίσω από το ρομπότ, λειτουργία εξαιρετικής σημασίας,

ειδικά κατά την εκκίνηση της λειτουργίας του που δεν γνωρίζει, εάν ο χώρος πίσω του είναι ελεύθερος ή κατειλημμένος και άρα δεν μπορεί να τον χρησιμοποιήσει για κάποιον ελιγμό, σε περίπτωση που ο χώρος μπροστά του είναι κατειλημμένος.

- Βελτίωση της εκτίμησης κατάστασης του ρομπότ, μέσω ανάπτυξης εύρωστου συστήματος εκτίμησης κατάστασης για ρομποτικά οχήματα με τετραδιεύθυνση, βάσει Εκτεταμένων Φίλτρων Kalman, που συνδυάζουν πληροφορία από το σύστημα οδομετρίας του ρομπότ, από αισθητήρα IMU ή πυξίδα και από οπτική οδομετρία με κάμερα.
- Βελτίωση της συμπεριφοράς του συστήματος αυτόνομης πλοήγησης με δυναμική παραμόρφωση μονοπατιού, μέσω αντικατάστασης των έτοιμων υλοποιήσεων των αλγορίθμων ελαστικής ζώνης και κατασκευής μονοπατιών, που χρησιμοποιήθηκαν για την υλοποίηση του αλγορίθμου Reeds-Shepp Band. Συγκεκριμένα, προτείνεται η υλοποίηση του αλγορίθμου Bubble Band των Khatib et al. [23], όπως επίσης και η υλοποίηση ενός αλγορίθμου κατασκευής μονοπατιών Reeds-Shepp, με ντετερμινιστικό τρόπο και εξέταση της δυνατότητας παραλληλοποίησης των δύο. Επίσης, θα μπορούσε να ληφθεί υπόψιν και η μελέτη των Sanan et al. [58] για την επέκταση του συνόλου των τύπων των μονοπατιών με μονοπάτια θετικής τετραδιεύθυνσης. Παράλληλα, το σύστημα αυτόνομης πλοήγησης θα βελτιωνόταν σημαντικά, και θα επέτρεπε την κίνηση σε αρκετά υψηλότερες ταχύτητες, μέσω αύξησης της συχνότητας λειτουργίας του αλγορίθμου διάσχισης μονοπατιού, ανεξάρτητα από την λειτουργία παραμόρφωσης μονοπατιού.
- Βελτίωση του αλγορίθμου διάσχισης μονοπατιού με προσθήκη λειτουργίας εντοπισμού επικείμενης σύγκρουσης, βάσει των πιο πρόσφατων μετρήσεων των αισθητήρων και αντίδραση μέσω μείωσης της ταχύτητας ή ακόμα και φρεναρίσματος, όπως παρουσιάζεται στην μελέτη των Mendes et al. [34].
- Εξέταση εναλλακτικών μεθόδων διάσχισης μονοπατιού μέσω προσαρμοστικού ελέγχου [33], ή Machine Learning, όπως Reinforcement Learning [59], ή υβριδικές προσεγγίσεις με Ασαφή Λογική και Νευρωνικά Δίκτυα [60].
- Βελτίωση του συστήματος αυτόνομης πλοήγησης με δυναμική ανακατασκευή μονοπατιού, μέσω αντικατάστασης του αλγορίθμου διάσχισης μονοπατιού, με έναν αλγόριθμο που θα μπορεί να πραγματοποιεί τοπική αποφυγή εμποδίων, ενώ παράλληλα ακολουθεί το ολικό μονοπάτι, όσο πιο πιστά γίνεται, λύνοντας έτσι το πρόβλημα κινδύνου σύγκρουσης, λόγω της χαμηλής συχνότητας ανακατασκευής του ολικού μονοπατιού, επιτρέποντας, παράλληλα, την δυνατότητα κίνησης με υψηλότερες ταχύτητες.

Βιβλιογραφία

- [1] R. Siegwart and I. R. Nourbakhsh. “Introduction to Autonomous Mobile Robots”. In: Cambridge MIT Press, 2004.
- [2] K. Berns and E. von Puttkamer. “Autonomous Land Vehicles”. In: Vieweg+Teubner, 2009. Chap. 3: Sensors.
- [3] *EPOS 24/1 Positioning Controller Documentation: Getting Started*. maxon motor. July 2007.
- [4] R. N. Jazar. “Vehicle Dynamics: Theory and Applications”. In: Springer Science and Business Media, LLC, 2009. Chap. 7: Steering Dynamics.
- [5] K. N. Spentzas, I. Alkhazali, and M. Demic. “Kinematics of four-wheel-steering vehicles”. In: *Springer - Verlag* (2001).
- [6] K. Bohlmann, H. Marks, and A. Zell. “Automated odometry self-calibration for car-like robots with four-wheel-steering”. In: *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*. Nov. 2012, pp. 168–173. doi: [10.1109/ROSE.2012.6402609](https://doi.org/10.1109/ROSE.2012.6402609).
- [7] D. Wang and F. Qi. “Trajectory planning for a four-wheel-steering vehicle”. In: *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. Vol. 4. 2001, 3320–3325 vol.4. doi: [10.1109/ROBOT.2001.933130](https://doi.org/10.1109/ROBOT.2001.933130).
- [8] S. Thrun, W. Burgard, and D. Fox. “Probabilistic Robotics”. In: The MIT Press, 2005. Chap. 7: Mobile Robot Localization.
- [9] E. Tsardoulias and L. Petrou. “Critical Rays Scan Match SLAM”. In: *Journal of Intelligent and Robotic Systems* (2013).
- [10] G. Grisetti, C. Stachniss, and W. Burgard. “Improved Techniques for Grid Mapping with Rao - Blackwellized Particle Filters”. In: *IEEE Transactions on Robotics* 23 (2007), pp. 34–46.
- [11] H. Choset et al. *Principles of Robot Motion: Theory, Algorithms and Implementations*. The MIT Press, 2005.
- [12] E. Tsardoulias. “Ενοποιημένο σύστημα ενός ή πολλαπλών ρομπότ για την πλοϊγηση και εξερεύνηση σε αστικό περιβάλλον με σκοπό τον εντοπισμό ανθρώπινης ζωής (USAR)”. PhD thesis. Electrical and Computer Engineering, Aristotle University of Thessaloniki, 2013.
- [13] M. Likhachev and Dave Ferguson. “Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles”. In: *The International Journal of Robotics Research* 28 (2009), pp. 933–945.
- [14] Maxim Likhachev, Geoff Gordon, and Sebastian Thrun. “ARA*: Anytime A* with Provable Bounds on Sub-Optimality”. In: *IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 16: PROCEEDINGS OF THE 2003 CONFERENCE (NIPS-03)*. MIT Press, 2004.
- [15] Maxim Likhachev et al. “Anytime Dynamic A*: An Anytime, Replanning Algorithm”. In: *In ICAPS*. 2005, pp. 262–271.
- [16] J. Borenstein and Y. Koren. “The vector field histogram-fast obstacle avoidance for mobile robots”. In: *IEEE Transactions on Robotics and Automation* 7.3 (June 1991), pp. 278–288. ISSN: 1042-296X. doi: [10.1109/70.88137](https://doi.org/10.1109/70.88137).
- [17] I. Ulrich and J. Borenstein. “VFH+: reliable obstacle avoidance for fast mobile robots”. In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 2. May 1998, 1572–1577 vol.2. doi: [10.1109/ROBOT.1998.677362](https://doi.org/10.1109/ROBOT.1998.677362).

- [18] I. Ulrich and J. Borenstein. “VFH*: local obstacle avoidance with look-ahead verification”. In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. Vol. 3. 2000, 2505–2511 vol.3. doi: [10.1109/ROBOT.2000.846405](https://doi.org/10.1109/ROBOT.2000.846405).
- [19] D. Fox, W. Burgard, and S. Thrun. “The dynamic window approach to collision avoidance”. In: *IEEE Robotics Automation Magazine* 4.1 (Mar. 1997), pp. 23–33. ISSN: 1070-9932. doi: [10.1109/100.580977](https://doi.org/10.1109/100.580977).
- [20] O. Brock and O. Khatib. “High-speed navigation using the global dynamic window approach”. In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. Vol. 1. 1999, 341–346 vol.1. doi: [10.1109/ROBOT.1999.770002](https://doi.org/10.1109/ROBOT.1999.770002).
- [21] R. Simmons. “The curvature-velocity method for local obstacle avoidance”. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*. Vol. 4. Apr. 1996, 3375–3382 vol.4. doi: [10.1109/ROBOT.1996.511023](https://doi.org/10.1109/ROBOT.1996.511023).
- [22] Nak Yong Ko and R. G. Simmons. “The lane-curvature method for local obstacle avoidance”. In: *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*. Vol. 3. Oct. 1998, 1615–1621 vol.3. doi: [10.1109/IROS.1998.724829](https://doi.org/10.1109/IROS.1998.724829).
- [23] M. Khatib et al. “Dynamic path modification for car-like nonholonomic mobile robots”. In: *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*. Vol. 4. Apr. 1997, 2920–2925 vol.4. doi: [10.1109/ROBOT.1997.606730](https://doi.org/10.1109/ROBOT.1997.606730).
- [24] S. Quinlan and O. Khatib. “Elastic bands: connecting path planning and control”. In: *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. May 1993, 802–807 vol.2. doi: [10.1109/ROBOT.1993.291936](https://doi.org/10.1109/ROBOT.1993.291936).
- [25] J. A. Reeds and L. A. Shepp. “Optimal paths for a car that goes both forwards and backwards.” In: *Pacific J. Math.* 145.2 (1990), pp. 367–393. URL: <http://projecteuclid.org/euclid.pjm/1102645450>.
- [26] Steven M. LaValle. *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.
- [27] H.J. Sussmann and G. Tang. *Shortest Paths For The Reeds-Shepp Car: A Worked Out Example Of The Use Of Geometric Techniques In Nonlinear Optimal Control*. Tech. rep. Rutgers University, Department of Mathematics, 1991.
- [28] J. P. Laumond and P. Soueres. “Metric induced by the shortest paths for a car-like mobile robot”. In: *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*. Vol. 2. July 1993, 1299–1304 vol.2. doi: [10.1109/IROS.1993.583756](https://doi.org/10.1109/IROS.1993.583756).
- [29] M. Khatib. “Sensor-based motion control for mobile robots”. PhD thesis. Toulouse, France: Laboratoire d'Automatique et d'Analyse des Systemes, LAAS-CNRS, 1996.
- [30] Bogdan Dumitrascu and Adrian Filipescu. “Sliding Mode Control of Lateral Motion for Four Driving-Steering Wheels Autonomous Vehicle”. In: *Annals of the University of Craiova* 7 (2010), pp. 20–25.
- [31] S. A. Tchenderli-Braham and F. Hamerlain. “Trajectory tracking with a hybrid control applied to a bi-steerable car”. In: *Systems and Computer Science (ICSCS), 2013 2nd International Conference on*. Aug. 2013, pp. 252–257. doi: [10.1109/IcConSCS.2013.6632056](https://doi.org/10.1109/IcConSCS.2013.6632056).
- [32] O. Azouaoui et al. “Fuzzy motion-based control for a bi-steerable mobile robot navigation”. In: *Mechatronics and its Applications, 2009. ISMA '09. 6th International Symposium on*. Mar. 2009, pp. 1–6. doi: [10.1109/ISMA.2009.5164850](https://doi.org/10.1109/ISMA.2009.5164850).
- [33] C. Cariou et al. “Adaptive control of four-wheel-steering off-road mobile robots: Application to path tracking and heading control in presence of sliding”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2008, pp. 1759–1764. doi: [10.1109/IROS.2008.4650645](https://doi.org/10.1109/IROS.2008.4650645).

- [34] A. Mendes, L. C. Bento, and U. Nunes. “Path-tracking controller with an anti-collision behaviour of a bi-steerable cybernetic car”. In: *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*. Vol. 1. Sept. 2003, 613–619 vol.1. DOI: [10.1109/ETFA.2003.1247763](https://doi.org/10.1109/ETFA.2003.1247763).
- [35] M. Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software*. 2009.
- [36] J. Lentini. *Mastering ROS for Robotics Programming*. Packt Publishing Ltd, 2015.
- [37] B. P. Gerkey, J. Leibs, and B. Gassend. *hokuyo_node*. www.wiki.ros.org/hokuyo_node.
- [38] A. Rebguns, C. Jorgensen, and C. Slutter. *dynamixel_controllers*. www.wiki.ros.org/dynamixel_controllers.
- [39] K. Tossell. *libuvc_camera*. www.wiki.ros.org/libuvc_camera.
- [40] J. S. Simon. *yocs_cmd_vel_mux*. www.wiki.ros.org/yocs_cmd_vel_mux.
- [41] E. Tsardoulias. *crsm_slam*. www.wiki.ros.org/crsm_slam.
- [42] B. Gerkey. *gmapping*. www.wiki.ros.org/gmapping.
- [43] W. Meeussen. *robot_pose_ekf*. www.wiki.ros.org/robot_pose_ekf.
- [44] E. Marder-Eppstein. *move_base*. www.wiki.ros.org/move_base.
- [45] C. Zalidis. “Augmenting perception for unmanned ground vehicles for efficient exploration and navigation in rough terrains”. MA thesis. Aristotle University of Thessaloniki, Dept. of Electrical and Computer Engineering, 2016.
- [46] E. Marder-Eppstein, D. V. Lu, and D. Hershberger. *costmap_2d*. www.wiki.ros.org/.
- [47] D. V. Lu. *global_planner*. www.wiki.ros.org/global_planner.
- [48] Christian Connette, Bhaskara Marthi, and Piyush Khandelwal. *eband_local_planner*. www.wiki.ros.org/eband_local_planner.
- [49] Mark Moll. *OMPL: Geometric Planning for car-like Vehicles, reeds_shepp_state_space*. www.ompl.kavrakilab.org/2012/03/18/geometric-planning-for-car-like-vehicles.html.
- [50] Kavraki Lab. *Open Motion Planning Library*. www.ompl.kavrakilab.org.
- [51] Juan Rada-Vilela. *fuzzylogic: a fuzzy logic control library*. www.fuzzylogic.com.
- [52] Michael Phillips. *sbpl_lattice_planner*. www.wiki.ros.org/sbpl_lattice_planner.
- [53] Search-Based Planning Lab. *Search-based planning library (SBPL)*. www.wiki.ros.org/sbpl.
- [54] W. Meeussen. *robot_state_publisher*. www.wiki.ros.org/.
- [55] D. Hershberger, D. Gossow, and J. Faust. *rviz: 3d Visualization tool for ROS*. www.wiki.ros.org/rviz.
- [56] Manos Tsardoulias, Chris Zalidis, and Aris Thallas. *stdr_simulator*. www.wiki.ros.org/stdr_simulator.
- [57] Open Source Robotics Foundation (OSRF). *gazebo*. www.gazebosim.org.
- [58] S. Sanan et al. “Extension of reeds and shepp paths to a robot with front and rear wheel steer”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. May 2006, pp. 3730–3735. DOI: [10.1109/ROBOT.2006.1642272](https://doi.org/10.1109/ROBOT.2006.1642272).
- [59] J. Baltes and Y. Lin. “Path-Tracking Control of Non-holonomic Car-Like Robot with Reinforcement Learning”. In: *RoboCup-99: Robot Soccer World Cup III*. London, UK, UK: Springer-Verlag, 2000, pp. 162–173. ISBN: 3-540-41043-0. URL: <http://dl.acm.org/citation.cfm?id=646584.698544>.

- [60] O. Azouaoui et al. “Neural Networks and Fuzzy Logic navigation approach for a bi-steerable mobile robot”. In: *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*. Nov. 2011, pp. 44–49. doi: [10.1109/URAI.2011.6145930](https://doi.org/10.1109/URAI.2011.6145930).