



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης Και Παραγωγής

Διπλωματική Εργασία

*Σχεδιασμός κίνησης αυτόνομου ρομποτικού οχήματος
σε περιβάλλον με εμπόδια, με χρήση μεθόδων
τεχνητής νοημοσύνης.*

Εκπόνηση:

Κούτσικα Αγγελική Ντίουβερτσε

Επιβλέπουσα:

Ζαχαρία Παρασκευή
Επικ. Καθηγήτρια

Αιγάλεω, Ιούλιος 2022

Η παρούσα διπλωματική εργασία εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από την Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του τμήματος.

Επιτροπή Αξιολόγησης:

ΕΠΩΝΥΜΟ ΟΝΟΜΑ	ΒΑΘΜΙΑΔΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
ΖΑΧΑΡΙΑ ΠΑΡΑΣΚΕΥΗ	ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ	
ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΗΣ	ΛΕΚΤΟΡΑΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ	
ΧΑΤΖΟΠΟΥΛΟΣ ΑΒΡΑΑΜ	ΛΕΚΤΟΡΑΣ ΕΦΑΡΜΟΓΩΝ	

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Κούτσικα Αγγελική – Ντίουβερτσε του Ευαγγέλου, με αριθμό μητρώου 71447951 φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Η Δηλούσα,

Κούτσικα Αγγελική Ντίουβερτσε



Περίληψη

Η παρούσα διπλωματική εργασία εστιάζει στην κινηματική ανάλυση ενός διαφορικού συστήματος οδήγησης, στον εντοπισμό θέσης και στην κατασκευή μονοπατιών ενός αυτόνομου ρομποτικού οχήματος, το οποίο είναι ικανό να πλοηγηθεί προς μια επιθυμητή τελική θέση, μέσα σε ένα περιβάλλον κλειστού χώρου και με ελεύθερα στατικά εμπόδια.

Στα πλαίσια αυτής της εργασίας, κατασκευάστηκαν το αυτόνομο ρομποτικό όχημα MIRc, στο πρόγραμμα προσομοίωσης CoppeliaSim και οι συναρτήσεις για τη μετακίνησή του, στην πλατφόρμα προγραμματισμού Matlab, με σκοπό την κατασκευή διαδρομών που θα εκτελούν διαμετακομίσεις κιβωτίων μεταξύ των γραμμών παραγωγής. Το ρομποτικό σύστημα του οχήματος καλείται, αρχικά, να κατασκευάσει ένα μονοπάτι από την τρέχουσα θέση του μέχρι μια τελική θέση που θα ορίζεται από τον χρήστη. Το συγκεκριμένο μονοπάτι δημιουργείται με τη χρήση του αλγόριθμου Dijkstra, ο οποίος υπολογίζει τη συντομότερη διαδρομή, αποφεύγοντας ταυτόχρονα τα εμπόδια που βρίσκονται στον εργασιακό χώρο του ρομπότ. Έπειτα, το ρομποτικό όχημα εκτελεί τη διαδρομή χρησιμοποιώντας τη μέθοδο Dynamic Window Approach και πλοηγείται αποφεύγοντας εμπόδια μέχρι να φτάσει στην επιθυμητή τελική θέση.

Λέξεις κλειδιά:

Αυτόνομο ρομποτικό όχημα, σύστημα διαφορικής οδήγησης, εντοπισμός θέσης, κατασκευή μονοπατιού, πλοήγηση, αποφυγή εμποδίων.

Abstract

MOVEMENT PLANNING OF AN AUTONOMOUS ROBOTIC VEHICLE IN AN ENVIRONMENT WITH OBSTACLES USING ARTIFICIAL INTELLIGENCE METHODS

The present thesis focuses on the kinematics of a differential drive system, localization and path planning of an autonomous robotic vehicle which is capable of navigating to a desired target location in an indoor environment with free static obstacles.

As part of this thesis, the MIRc autonomous robotic vehicle was developed in CoppeliaSim simulation program and the functions for its movement were created in Matlab programming platform, in order to accomplish path planning and material handling in-between production lines. Firstly, the robot's system must create a path from its local position to a goal position, which is provided by the user. The algorithm used for path planning is the Dijkstra algorithm, which simultaneously finds the shortest path and avoids obstacles in the robot's workspace. Afterwards, the robotic vehicle navigates towards the desired final position avoiding any obstacle, using the Dynamic Window Approach method.

Keywords:

Autonomous robotic vehicle, differential drive system, localization, path planning, navigation, obstacle avoidance.

Πίνακας Περιεχομένων

Εισαγωγή	9
Περιγραφή του προβλήματος.....	10
Διάρθρωση της Διπλωματικής.....	10
Κεφάλαιο 1 – Ρομποτικό Σύστημα.....	11
1.1 Αυτόνομα Ρομπότ	11
1.1.1 Κατηγοριοποίηση ρομπότ	12
1.1.2 Αρχιτεκτονική αυτόνομων ρομποτικών οχημάτων.....	16
1.2 Σύστημα Μετάδοσης Κίνησης και Ελέγχου	22
1.2.1 Κινηματική ρομποτικού οχήματος διαφορικής οδήγησης	23
1.2.2 Έλεγχος και ευστάθεια	28
Κεφάλαιο 2 – Αυτόνομη Πλοήγηση.....	32
2.1 Σύστημα Πλοήγησης.....	33
2.1.1 Λειτουργικό πλαίσιο.....	34
2.1.2 Αντίληψη χώρου.....	40
2.2 Κατασκευή Μονοπατιού	46
2.2.1 Αλγόριθμοι κατασκευής μονοπατιού	47
2.2.2 Μέθοδοι αποφυγής εμποδίων.....	53
Κεφάλαιο 3 – Λογισμικό Προσομοίωσης.....	56
3.1 CoppeliaSim	57
3.2 Matlab.....	60
Συμπεράσματα	71
Βιβλιογραφία – Αναφορές – Διαδικτυακές Πηγές	72

Λίστα Εικόνων

Εικόνα 1 α) Ρομποτικός βραχίονας σταθερής βάσης [38], β) Πλάγια όψη ρομποτικού βραχίονα σταθερής βάσης [39].	12
Εικόνα 2 α) Ρομποτική σκούπα Roomba j7+, iRobot [40], β) Ρομπότ απολύμανσης Robotsafe, Gizelis Robotics [9].	13
Εικόνα 3 Ανθρωποειδές ρομπότ Atlas, Boston Dynamics [10].	14
Εικόνα 4 Η χρήση των drone στον αγροτικό τομέα για την επισκόπηση εκτάσεων και την παρακολούθηση της συνολικής υγείας μιας καλλιέργειας [41].	15
Εικόνα 5 Μη επανδρωμένο υποβρύχιο όχημα Remus 300, Hydroid [42].	15
Εικόνα 6 α) Διαδικασία διαμετακόμισης προϊόντων με τη χρήση του ρομπότ MiR100, της εταιρείας MiR [35], β) Κλήση ρομποτικού οχήματος από τον χρήστη για την παραλαβή ενός επεξεργασμένου προϊόντος [35].	16
Εικόνα 7 Διάγραμμα συστήματος αυτόνομου ρομποτικού οχήματος [37].	18
Εικόνα 8 Οργάνωση λειτουργικής μονάδας [16].	20
Εικόνα 9 Οργάνωση επιπέδου εκτέλεσης [16].	21
Εικόνα 10 Πρότυπο σχεδιαστή – επιθεωρητή [16].	22
Εικόνα 11 α) Ρομποτικό όχημα Pioneer 3-DX, Mobile Robots [17] β) Πλάγια όψη ρομποτικού οχήματος Pioneer 3-DX [17].	22
Εικόνα 12 Το απόλυτο σύστημα αναφοράς και το σχετικό σύστημα αναφοράς του ρομποτικού οχήματος [20].	23
Εικόνα 13 Περιστροφή ρομποτικού οχήματος διαφορετικής οδήγησης [21].	24
Εικόνα 14 Προσβάσιμοι και μη προσβάσιμοι βαθμοί ελευθερίας ενός ρομποτικού οχήματος διαφορετικής οδήγησης [6].	25
Εικόνα 15 Διανύσματα θέσης και προσανατολισμού ρομποτικού οχήματος [20].	26
Εικόνα 16 Δημιουργία ομαλής διαδρομής ενός ρομποτικού οχήματος [18].	29
Εικόνα 17 Ρομποτικό όχημα MiRc κατά τη διαδικασία πλοήγησής του, στο πρόγραμμα προσομοίωσης CoppeliaSim.	32
Εικόνα 18 Παράδειγμα τοποθέτησης ρομποτικού βραχίονα πάνω σε κινητό βιομηχανικό ρομπότ, Enabled Robotics - ER-FLEX, της εταιρείας MiR [35].	34
Εικόνα 19 α) Χώρος εργασίας του ρομποτικού οχήματος [11], β) Χώρος διάταξης του ρομποτικού συστήματος [11].	36
Εικόνα 20 Εφαρμογή φίλτρου Kalman [18].	38
Εικόνα 21 BIBA Robot, BlueBotics SA [18].	41
Εικόνα 22 Χρήση ενός αισθητήρα απόστασης [6].	42
Εικόνα 23 Αισθητήρας LIDAR URG-04LX, Hokuyo [43].	43
Εικόνα 24 α) Μαγνητικός κωδικοποιητής [44], β) Οπτικός κωδικοποιητής [45].	43
Εικόνα 25 α) Μηχανικό γυροσκόπιο δύο αξόνων [18], β) Οπτικό γυροσκόπιο [46].	45
Εικόνα 26 Μια πιθανή πορεία που μπορεί να ακολουθήσει το ρομπότ [11].	47
Εικόνα 27 Κατευθυνόμενος γράφος με βάρη [11].	49
Εικόνα 28 Ανάπτυξη του αλγόριθμου RRT από 45 μέχρι 2345 επαναλήψεις [18].	50
Εικόνα 29 Παράδειγμα δυναμικού πεδίου που χρησιμοποιείται για την πλοήγηση προς μια τελική θέση αποφεύγοντας και την αποφυγή εμποδίων [47].	51

Εικόνα 30 Παράδειγμα αλγορίθμου RRT [11]	53
Εικόνα 31 Δυναμικό παράθυρο [18].....	55
Εικόνα 32 Γωνία θ της συνάρτησης <i>heading</i> , ω του ρομποτικού οχήματος [34]	55
Εικόνα 33 Χώρος πλοήγησης του ρομποτικού οχήματος MIRc στο περιβάλλον προσομοίωσης CoppeliaSim.....	56
Εικόνα 34 α) Μέρη ρομποτικού οχήματος MIRc, β) Ρομποτικό όχημα MIRc	57
Εικόνα 35 Εμφάνιση ενδεικτικής θέσης για την επιλογή του σημείου 2	58
Εικόνα 36 Σχετικός χάρτης του χώρου	61
Εικόνα 37 Μετακίνηση της θέσης στόχου μέσα στον χώρο προσομοίωσης.....	62
Εικόνα 38 Θέση έναρξης (μπλε) και στόχου (κόκκινο).....	64
Εικόνα 39 α) Διαδικασία έναρξης αναζήτησης μονοπατιού από τη θέση εκκίνησης ως τη θέση στόχου, β) Ολοκλήρωση διαδικασίας αναζήτησης και παρουσίαση λύσης του συντομότερου μονοπατιού.....	64
Εικόνα 40 Χάρτης αποστάσεων αλγόριθμου Dijkstra (Dijkstra's cost map)	66
Εικόνα 41 α) Πλοήγηση του MIRc με τη μέθοδο DWA, β) Παρουσίαση στιγμιότυπου από το πρόγραμμα προσομοίωσης CoppeliaSim, από την άφιξη του ρομποτικού οχήματος στην τελική θέση.....	67
Εικόνα 42 Εύρεση μονοπατιού με τη χρήση του αλγόριθμου αναζήτησης RRT.....	69
Εικόνα 43 Λύση του προβλήματος πλοήγησης με τη χρήση του αλγόριθμου Dijkstra και της μεθόδου DWA σε ένα χαρτογραφημένο περιβάλλον	70
Εικόνα 44 Λύση του προβλήματος πλοήγησης με τη χρήση του αλγόριθμου Dijkstra και της μεθόδου DWA σε ένα χαρτογραφημένο περιβάλλον με εμπόδια.	70

Λίστα Αλγορίθμων

Αλγόριθμος 1 Markov Localization.....	39
Αλγόριθμος 2 Φίλτρο Kalman	39
Αλγόριθμος 3 Dijkstra Algorithm.....	52
Αλγόριθμος 4 RRT (Rapidly exploring Random Tree).....	53

Εισαγωγή

Τα αυτόνομα κινητά ρομπότ αποτελούν ρομπότ που έχουν την ικανότητα να μετακινούνται από ένα σημείο προς ένα άλλο αυτόνομα. Συγκριτικά με τα βιομηχανικά ρομπότ που μπορούν να κινηθούν σε στατικό σημείο, τα κινητά ρομπότ και συγκεκριμένα τα αυτόνομα ρομποτικά οχήματα, που μελετώνται στην παρούσα διπλωματική, έχουν τη δυνατότητα να μετακινούνται ελεύθερα μέσα σε έναν προκαθορισμένο χώρο εργασίας έτσι ώστε να διεκπεραιώσουν τις εργασίες που τους αναθέτονται. Αυτό το χαρακτηριστικό τα καθιστά κατάλληλα για την εκτέλεση ενός μεγάλου αριθμού διαφορετικών εφαρμογών, τόσο σε δομημένους όσο και σε μη δομημένους χώρους.

Σύμφωνα με το Ινστιτούτο Ρομποτικής της Αμερικής το ρομπότ είναι ένας προγραμματιζόμενος και πολυλειτουργικός χωρικός μηχανισμός σχεδιασμένος να μετακινεί υλικά, αντικείμενα, εργαλεία ή εξειδικευμένες συσκευές με κατάλληλες μεταβλητά προγραμματιζόμενες κινήσεις που στοχεύουν στη βελτίωση της απόδοσης μιας σειράς εργασιών [1]. Αντίστοιχα, σύμφωνα με τον Michael Brady η ρομποτική είναι η ευφυής σύνδεση μεταξύ της αντίληψης και της πράξης [2]. Στην παρούσα διπλωματική εργασία χρησιμοποιείται ένας συνδυασμός των δύο ορισμών που αναφέρθηκαν παραπάνω ώστε να γίνει η κατάλληλη προσαρμογή της έννοιας στο αντικείμενο μελέτης, που αφορά το αυτόνομο ρομποτικό όχημα. Συνεπώς, το ρομπότ μπορεί να θεωρηθεί ένα μηχάνημα ικανό να επεξεργαστεί και να εξάγει πληροφορίες από το περιβάλλον του και να τις χρησιμοποιήσει για να αποκτήσει γνώσεις που αφορούν το χώρο, με αποτέλεσμα να κινείται με ασφαλή και ουσιαστικό τρόπο και για σκόπιμο λόγο.

Σήμερα ο κλάδος της ρομποτικής και συγκεκριμένα ο τομέας των αυτόνομων ρομποτικών οχημάτων παρουσιάζει τεράστια εξέλιξη, προσφέροντας πολλαπλές λύσεις σχετικά με τη συνεισφορά των έξυπνων συστημάτων στο εργασιακό περιβάλλον. Κάτω από τον όρο «αυτόνομο» συμπεριλαμβάνεται ένας τεράστιος αριθμός ικανοτήτων με αποτέλεσμα να μπορεί να επιλεγθεί για την εκτέλεση πολλαπλών εφαρμογών. Αυτό σημαίνει πως όσο πιο ανεπτυγμένο είναι ένα ρομποτικό σύστημα τόσο πιο πολύπλοκες εφαρμογές μπορεί να διεκπεραιώσει. Σήμερα η νέα ψηφιακή τεχνολογία του Industry 4.0 απαιτεί ανεπτυγμένα ρομποτικά συστήματα στις βιομηχανίες έτσι ώστε να μπορούν να λειτουργούν πιο παραγωγικά και να βελτιστοποιήσουν τα ποσοστά επιτυχίας [3]. Ως Industry 4.0 θεωρείται ένα αυτόνομο σύστημα που καθοδηγείται από το Διαδίκτυο των Πραγμάτων (*Internet of Things – IoT*), που επιτρέπει την αλληλεπίδραση με άλλες συσκευές, εργαλεία και υπολογιστές. Οι δραστηριότητες του ρομπότ μπορούν να δημιουργούνται και να αυτοματοποιούνται συνδέοντας το ρομποτικό όχημα με έναν κεντρικό υπολογιστή, με μια βάση δεδομένων ή με έναν προγραμματιζόμενο λογικό ελεγκτή (*Programmable Logic Controller – PLC*). Με αυτόν τον τρόπο τα ρομποτικά οχήματα μπορούν να πραγματοποιούν εργασίες χωρίς την ανθρώπινη παρέμβαση, όπως είναι η διαμετακόμιση προϊόντων μεταξύ σημείων, η αποφυγή εμποδίων και η αλληλεπίδραση με άλλα ρομποτικά οχήματα, δημιουργώντας συνθήκες προτεραιότητας κατά τη διάρκεια της πλοήγησης.

Συνεπώς, για την επίτευξη της αυτονομίας ενός ρομποτικού οχήματος είναι απαραίτητο να γίνουν κατανοητές όλες οι βασικές λειτουργίες του συστήματός του, όπως είναι για παράδειγμα ο τρόπος με τον οποίο αντιλαμβάνεται το περιβάλλον του και πραγματοποιεί τη διαδικασία της

πλοήγησης. Με αυτόν τον τρόπο μπορούν στη συνέχεια να αναπτυχθούν επιπλέον δυνατότητες που θα έχουν σκοπό τη λύση πολύπλοκων προβλημάτων που θα κληθεί να λύσει το αυτόνομο ρομποτικό όχημα στον χώρο εργασίας του.

Περιγραφή του προβλήματος

Η παρούσα διπλωματική εργασία εξετάζει το πρόβλημα της πλοήγησης ενός αυτόνομου ρομποτικού οχήματος διαφορετικής οδήγησης, σε ένα περιβάλλον με εμπόδια. Συγκεκριμένα, παρουσιάζεται ο τρόπος με τον οποίο το ρομπότ καλείται να κατασκευάσει μια πορεία πλοήγησης από μια τρέχουσα θέση προς ένα επιθυμητό σημείο άφιξης, χρησιμοποιώντας το σύστημα αντίληψης του και αλγόριθμους κατασκευής μονοπατιών και αποφυγής εμποδίων. Έπειτα, σε δεύτερο στάδιο, καλείται να πλοηγηθεί μέχρι να φτάσει στην τελική θέση-στόχος προσπερνώντας στατικά και δυναμικά εμπόδια, χρησιμοποιώντας μεθόδους αποφυγής εμποδίων. Σκοπός είναι το ρομποτικό όχημα να φτάσει στον προορισμό του, ολοκληρώνοντας έτσι με επιτυχία την εργασία που του έχει ανατεθεί χωρίς να χρειαστεί η παρέμβαση ανθρώπινου χειριστή.

Διάρθρωση της Διπλωματικής

Ένα αυτόνομο ρομποτικό σύστημα αποτελείται από κάποιες βασικές κατηγορίες που συντελούν στην αποτελεσματική λειτουργία του, οι οποίες θα παρουσιαστούν αναλυτικά στα επόμενα κεφάλαια. Συγκεκριμένα:

Στο **Κεφάλαιο 1** παρουσιάζεται το ρομποτικό σύστημα, όσον αφορά τα αυτόνομα ρομπότ. Αρχικά, γίνεται μια κατηγοριοποίηση των διάφορων τύπων αυτόνομων ρομπότ που υπάρχουν σήμερα και έπειτα αναλύονται τα επίπεδα της αρχιτεκτονικής των αυτόνομων ρομποτικών οχημάτων. Ακόμα, στο συγκεκριμένο κεφάλαιο παρουσιάζεται το σύστημα μετάδοσης κίνησης που έχει επιλεγεί για την κατασκευή του ρομποτικού οχήματος MIRc σχετικά με την κινηματική και τον έλεγχο ευστάθειας της κατασκευής.

Στο **Κεφάλαιο 2** παρουσιάζεται η έννοια της αυτόνομης πλοήγησης, δηλαδή το λειτουργικό πλαίσιο ενός συστήματος πλοήγησης και η διαδικασία με την οποία το ρομποτικό όχημα μπορεί να αντιληφθεί το περιβάλλον του. Επίσης, παρουσιάζονται οι αλγόριθμοι κατασκευής μονοπατιού και οι μέθοδοι αποφυγής εμποδίων που χρησιμοποιούνται για την ανάπτυξη του ρομποτικού οχήματος MIRc.

Στο **Κεφάλαιο 3** παρουσιάζονται το λογισμικό προσομοίωσης και τμήματα του κώδικα προγραμματισμού που δημιούργησαν το ρομποτικό όχημα MIRc. Επιπλέον, παρουσιάζεται ένα σενάριο πλοήγησης στο περιβάλλον προσομοίωσης για την εξέταση της συμπεριφοράς του ρομποτικού οχήματος.

Κεφάλαιο 1 – Ρομποτικό Σύστημα

Το 1980 η ρομποτική προσδιορίστηκε ως η επιστήμη που μελετά την ευφυή σύνδεση μεταξύ της εκτέλεσης ενεργειών και της αντίληψης [4]. Λαμβάνοντας υπόψη τον συγκεκριμένο ορισμό η εκτέλεση ενεργειών ενός ρομποτικού συστήματος αποτελείται από ένα σύστημα κίνησης, που είναι υπεύθυνο για τη μετακίνηση του ρομπότ στον χώρο και από συστήματα χειρισμού, όπως βραχίονες, όπου χρησιμοποιώντας τους κατάλληλους ενεργοποιητές προκαλείται η ενεργοποίηση των μηχανικών τμημάτων του ρομπότ. Η αντίληψη αποτελείται από το σύστημα αισθητήρων που παρέχουν πληροφορίες σχετικά με τη λειτουργική κατάσταση του ρομποτικού οχήματος, όπως είναι η διάταξη και η ταχύτητα και με το περιβάλλον του ρομπότ, όπως είναι η εμβέλεια και η όραση. Η ευφυής σύνδεση των δύο τμημάτων πραγματοποιείται μέσα από τον αρχιτεκτονικό σχεδιασμό, τον προγραμματισμό και τον έλεγχο, που βασίζονται στην αντίληψη και τα διαθέσιμα μοντέλα του ρομποτικού οχήματος και του περιβάλλοντός του.

Όλα τα ρομποτικά συστήματα χρησιμοποιούν μια αρχιτεκτονική καθώς διευκολύνει την ανάπτυξη του συστήματος παρέχοντας ωφέλιμους περιορισμούς για την κατασκευή και την λειτουργία του. Αυτό συμβαίνει καθώς οι απαιτήσεις ενός ρομποτικού συστήματος είναι υψηλές, όπως για παράδειγμα η ανάγκη να αλληλοεπιδρά ασύγχρονα και σε πραγματικό χρόνο με ένα άγνωστο και δυναμικό περιβάλλον [5]. Επιπλέον, πολλά ρομποτικά συστήματα χρειάζεται να ανταποκριθούν και να στείλουν ανατροφοδότηση ελέγχου μέσα σε χιλιοστά του δευτερολέπτου για περίπλοκα ζητούμενα. Για τον διαχειρισμό αυτών των απαιτήσεων οι αρχιτεκτονικές περιέχουν ικανότητες για την εκτέλεση ενεργειών σε πραγματικό χρόνο, για τον έλεγχο ενεργοποιητών και αισθητήρων, για την ανίχνευση και την αντίδραση ειδικών περιπτώσεων και για την αντιμετώπιση της αβεβαιότητας.

Παρακάτω παρουσιάζονται οι κατηγορίες των αυτόνομων ρομπότ που υπάρχουν σήμερα και συγκεκριμένα αναλύονται η αρχιτεκτονική και το σύστημα μετάδοσης κίνησης της κατηγορίας που χρησιμοποιήθηκε για το ρομποτικό όχημα MIRc, που αποτελεί το αυτόνομο ρομποτικό όχημα.

1.1 Αυτόνομα Ρομπότ

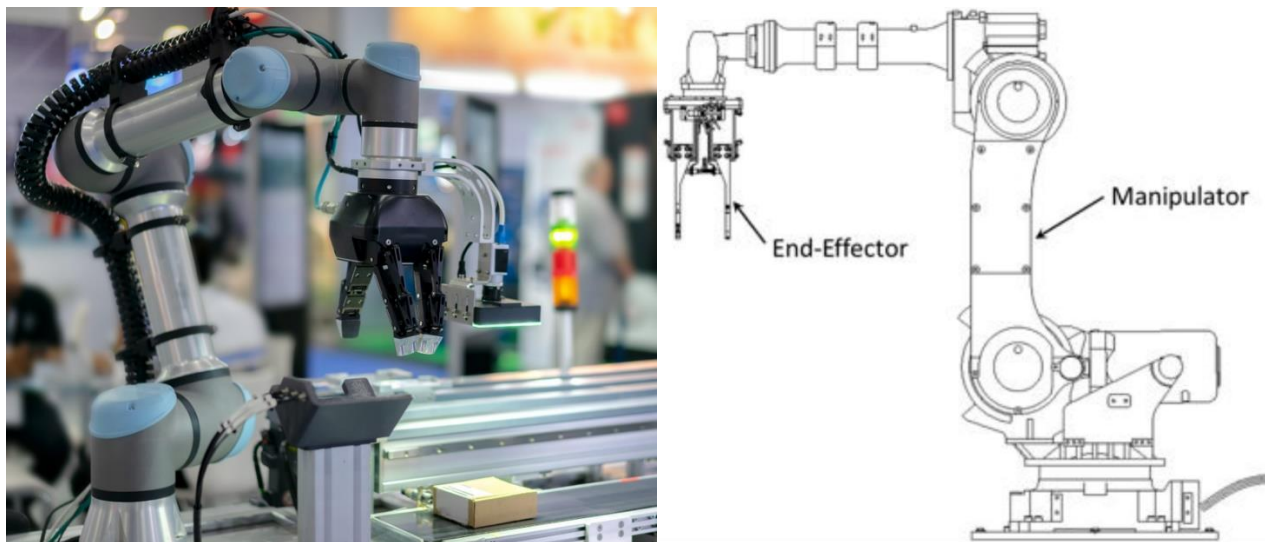
Τα ρομπότ μπορούν να κατηγοριοποιηθούν με βάση το περιβάλλον στο οποίο τίθενται σε λειτουργία και ένας από τους συνηθέστερους τρόπους διαχωρισμού βρίσκεται στον αν αποτελείται από ένα στατικό (*fixed*) ή κινητό (*mobile*) ρομποτικό σύστημα [6]. Τα στατικά ρομποτικά συστήματα είναι συνήθως βιομηχανικοί ρομποτικοί βραχίονες που λειτουργούν σε προκαθορισμένο και ειδικά προσαρμοσμένο περιβάλλον, για τον συγκεκριμένο τύπο ρομπότ. Αντίθετα, τα κινητά ρομποτικά συστήματα προβλέπονται να κινούνται και να πραγματοποιούν εργασίες σε μεγάλους χώρους και έχουν τη δυνατότητα να αντιμετωπίζουν άγνωστες και χρονικά μεταβαλλόμενες καταστάσεις. Το χαρακτηριστικό που διαχωρίζει τα κινητά ρομποτικά οχήματα με τα αυτόνομα κινητά ρομποτικά οχήματα (*Autonomous Mobile Robots – AMRs*) είναι πως τα AMRs δεν εξαρτώνται από χειριστή και μπορούν να λαμβάνουν αποφάσεις και να εκτελούν εργασίες, όπως διαμετακομίσεις υλικών, ενώ ταυτόχρονα πλοηγούνται σε άγνωστους χώρους όπου το περιβάλλον μπορεί να μεταβάλλεται συνεχώς.

1.1.1 Κατηγοριοποίηση ρομπότ

Η κατηγοριοποίηση ενός ρομποτικού συστήματος δεν γίνεται μόνο με βάση τον χώρο στον οποίο θα κινηθεί αλλά και ανάλογα με τα τεχνικά κριτήρια που απαιτούνται για τη σωστή λειτουργία του, όπως η ευελιξία, η αποδοτικότητα και η ευστάθεια [7]. Επομένως, με βάση το σύστημα που χρησιμοποιεί για την κίνησή του μπορεί να χωριστεί σε τέσσερις κατηγορίες, όπως τα στατικά, τα επίγεια, τα εναέρια και τα υποβρύχια ρομπότ, τα οποία αντίστοιχα χωρίζονται σε πολλές υποκατηγορίες, κάποιες από τις οποίες θα αναφερθούν παρακάτω.

Στατικά ρομπότ

Έναν βασικό τύπο στατικού ρομπότ αποτελούν τα βιομηχανικά ρομπότ και συγκεκριμένα οι βιομηχανικοί ρομποτικοί βραχίονες. Τα πρώτα ρομπότ που κατασκευάστηκαν αποτελούν τα βιομηχανικά ρομπότ (*industrial robots*) τα οποία αντικατέστησαν τους εργάτες, προκειμένου να πραγματοποιήσουν τις απλές επαναλαμβανόμενες κινήσεις στη θέση τους [6]. Η τοποθέτησή τους σε γραμμές παραγωγής εργοστασίων έδωσαν τη δυνατότητα να μην απαιτείται η παρουσία ανθρώπων, αφού το ρομπότ μπορεί να εκτελέσει μια σειρά εργασιών σε αντικείμενα που βρίσκονται στον χώρο του. Τα σημερινά βιομηχανικά ρομπότ έχουν μεγαλύτερη ελαστικότητα, καθώς έχουν τη δυνατότητα να αλλάξουν τον προσανατολισμό τους και να τον προσαρμόζουν ανάλογα με τη θέση που βρίσκεται το αντικείμενο που θέλουν να επεξεργαστούν. Η λειτουργία του ρομποτικού βραχίονα καθορίζεται στο ελεύθερο άκρο του, όπου υπάρχει το κατάλληλο, κατά περίπτωση, εργαλείο που ονομάζεται αποληκτήριο επενεργητής (*end-effector*) [8]. Ανάλογα με την εφαρμογή, ο αποληκτήριο επενεργητής μπορεί να είναι αρπάγη, ηλεκτρόδιο συγκόλλησης, ηλεκτρομαγνήτης ή κάποια άλλη διάταξη. Στις Εικόνα 1α και β παρουσιάζεται ένας ρομποτικός βραχίονας σταθερής βάσης, όπου ο αποληκτήριο επενεργητής είναι αρπάγη.



Εικόνα 1 α) Ρομποτικός βραχίονας σταθερής βάσης [38], β) Πλάγια όψη ρομποτικού βραχίονα σταθερής βάσης [39].

Επίγεια ρομπότ

Ο τύπος ρομπότ που θα μελετηθεί διεξοδικά στην παρούσα εργασία αναφορικά με τον σχεδιασμό της κίνησής του στον χώρο, αποτελεί το τροχοφόρο ρομπότ και συγκεκριμένα το αυτόνομο

ρομποτικό όχημα. Το τροχοφόρο ρομπότ αποτελεί ένα συχνά εμφανιζόμενο είδος επίγειου ρομπότ στην καθημερινότητά μας, καθώς η χρήση τροχών είναι πιο απλή, σε αντίθεση με τη χρήση άκρων, για τον σχεδιασμό, την κατασκευή και τον προγραμματισμό της κίνησής του στο επίπεδο [7]. Ένας βασικός σκοπός των επίγειων ρομπότ είναι η λύση προβλημάτων σχετικά με την πλοήγησή τους μέσα στον χώρο χωρίς να είναι απαραίτητη η ανθρώπινη παρέμβαση. Επομένως, κύρια μελέτη των επιστημόνων σχετικά με την κατασκευή ενός ρομποτικού συστήματος ελέγχου αποτελεί η εξέλιξη των αυτόνομων δυνατοτήτων του οχήματος, έτσι ώστε να αποκτήσει μεγαλύτερη ευφυΐα, ευαισθησία και υπευθυνότητα ως προς οποιαδήποτε αλλαγή που μπορεί να συμβεί στο περιβάλλον σε πραγματικό χρόνο και ταυτόχρονα να βρίσκεται σε ετοιμότητα για τη λήψη της βέλτιστης απόφασης σχετικά με την λύση κάποιου απρόοπτου προβλήματος που θα μπορούσε να επιφέρει αυτή η αλλαγή.

Χαρακτηριστικός τύπος επίγειων ρομπότ αποτελούν τα αυτόνομα ρομποτικά οχήματα (*Autonomous Robotic Vehicles*) που έχουν την ικανότητα να κινούνται στον χώρο και να παίρνουν αποφάσεις σε πραγματικό χρόνο. Τα αυτόνομα ρομποτικά οχήματα αξιοποιούν τεχνολογίες όπως αισθητήρες και κάμερες, ώστε να αντλήσουν πληροφορίες από τον χώρο στον οποίο κινούνται και να τις επεξεργαστούν προκειμένου να πάρουν κάποια συγκεκριμένη απόφαση. Μια τέτοια απόφαση μπορεί να αφορά σε κίνηση για την αποφυγή ενός ανθρώπου, την απολύμανση μιας συγκεκριμένης διαδρομής ή το καλωσόρισμα ανθρώπων που εισέρχονται σε έναν χώρο έκθεσης. Ένα παράδειγμα αυτόνομου ρομποτικού οχήματος αποτελεί η σκούπα Roomba j7+ (Εικόνα 2α), της εταιρείας iRobot που καθαρίζει χώρους κάνοντας χαρτογράφηση και δημιουργώντας αυτόνομα τη διαδρομή που πρέπει να ακολουθήσει για να καλύψει όλον τον χώρο. Ένα ακόμα παράδειγμα αποτελεί το αυτόνομο ρομπότ απολύμανσης Robotsafe (Εικόνα 2β), της εταιρείας Gizelis Robotics που έχει κατασκευαστεί για την απολύμανση εταιρικών χώρων, εγκαταστάσεις, βιομηχανίες, αποθήκες, εργοστάσια, κέντρα διανομής, αεροδρόμια και υγειονομικούς χώρους [9].



Εικόνα 2 α) Ρομποτική σκούπα Roomba j7+, iRobot [40], β) Ρομπότ απολύμανσης Robotsafe, Gizelis Robotics [9].

Ένας δεύτερος τύπος επίγειου ρομπότ αποτελεί το ρομποτικό σύστημα που χρησιμοποιεί πόδια ή άκρα (*legged robot*) για τη μετακίνησή του, στον οποίο εντάσσονται τα δίποδα (ανθρωποειδές) και τα τετράποδα ρομπότ. Ο σκοπός των συγκεκριμένων ρομπότ είναι η δημιουργία λύσεων σε προβλήματα που πρέπει να επιλυθούν σε χώρους όπου το έδαφος είναι πιο τραχύ, δυσπρόσιτο από κοινά ρομποτικά οχήματα ή απαιτείται μεγαλύτερος βαθμός κινητικότητας [10]. Τα τροχοφόρα επίγεια ρομποτικά οχήματα παρουσιάζουν περιορισμούς όταν πρόκειται για χώρους που περιλαμβάνουν σκάλες, κενά ή μικρές διαφορές ύψους στο δάπεδο, σε αντίθεση με τα ανθρωποειδές ρομπότ που χαρακτηρίζονται από τη δυναμική και την ισορροπία κατά τη διάρκεια της κίνησης. Η εταιρεία Boston Dynamics αποτελεί ένα βασικό παράδειγμα για την κατασκευή ανθρωποειδών ρομπότ (Εικόνα 3) που έχουν την ικανότητα να διεκπεραιώνουν απαιτητικές και περίπλοκες εργασίες.



Εικόνα 3 Ανθρωποειδές ρομπότ Atlas, Boston Dynamics [10].

Εναέρια ρομπότ

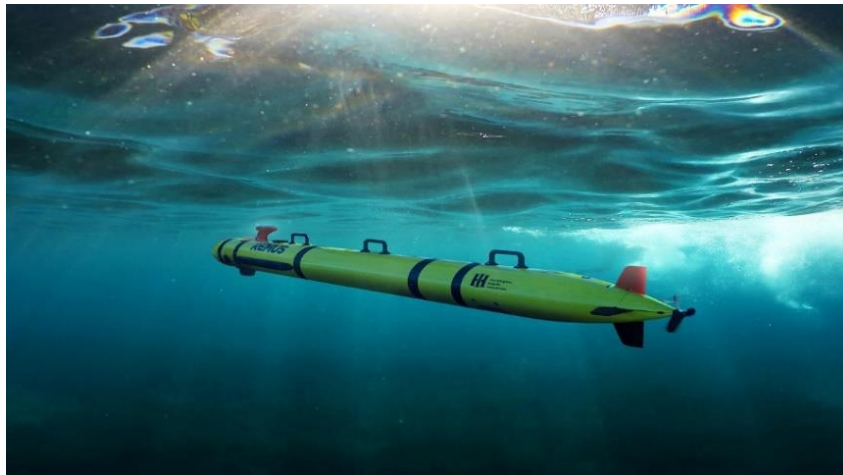
Ο συγκεκριμένος τύπος ρομπότ αποτελείται από κινητά συστήματα (*mobile systems*) που έχουν την ικανότητα να πετάξουν σε εναέριους χώρους [11]. Σε αυτή την κατηγορία εντάσσονται και τα μη επανδρωμένα οχήματα (*Unmanned Aerial Vehicles – UAV*) ή αλλιώς drone, τα οποία αποτελούν αεροσκάφη που κατά την πλοήγησή τους δεν απαιτούν πιλότο. Τα πιο εξελιγμένα drone έχουν την ικανότητα να απογειώνονται και να προσγειώνονται αυτόνομα χωρίς την παρέμβαση του χειριστή τους [7]. Κατά την πρωτοεμφάνιση των UAVs, η χρήση τους προοριζόταν αποκλειστικά για στρατιωτικές εφαρμογές, αλλά στη συνέχεια διευρύνθηκε μετά την αναγνώριση της χρηστικότητάς τους σε διάφορους κλάδους, όπως εκείνον της επιστήμης, της γεωργίας, των logistics, κλπ.



Εικόνα 4 Η χρήση των drone στον αγροτικό τομέα για την επισκόπηση εκτάσεων και την παρακολούθηση της συνολικής υγείας μιας καλλιέργειας [41].

Υποβρύχια ρομπότ

Τα υποβρύχια ρομπότ ή μη επανδρωμένα υποβρύχια οχήματα (*Unmanned Underwater Vehicles – UUV*) έχουν την ικανότητα να κολυμπούν ελεύθερα, χωρίς να υπάρχει χειριστής στο εσωτερικό του οχήματος [12]. Αυτός ο τύπος ρομπότ χρησιμοποιεί τη δική του παροχή ενέργειας την οποία μεταφέρει στο εσωτερικό του οχήματος και επικοινωνεί μόνο με ασύρματη επικοινωνία, γνωστά ως ακουστικά σήματα (*sonar signals*). Η χρήση τους συνήθως αφορά την εξερεύνηση επικίνδυνων και ανεξερεύνητων τμημάτων του ωκεανού και περιλαμβάνουν αισθητήρες για τη μέτρηση των χαρακτηριστικών αυτών των περιοχών, όπως αλμυρότητα, θερμοκρασία και την κατεύθυνση ή την ταχύτητα των ρευμάτων [13].

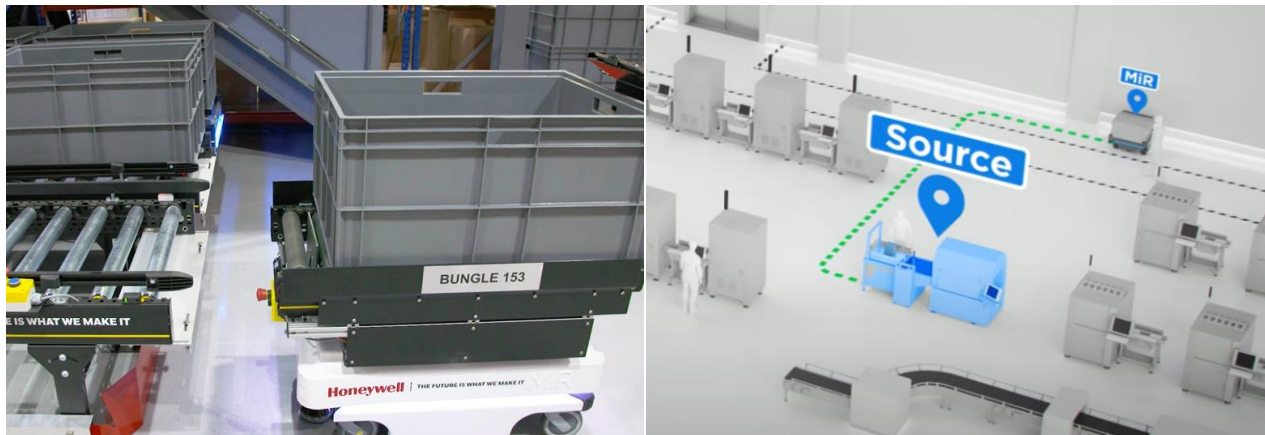


Εικόνα 5 Μη επανδρωμένο υποβρύχιο όχημα Remus 300, Hydroid [42].

1.1.2 Αρχιτεκτονική αυτόνομων ρομποτικών οχημάτων

Ο τύπος ρομπότ που χρησιμοποιείται συχνότερα για τη διαμετακόμιση προϊόντων (Εικόνα 6α) μέσα στον χώρο παραγωγής ή μεταξύ των τμημάτων μιας βιομηχανίας αποτελεί το κινητό βιομηχανικό ρομπότ (*mobile industrial robot*). Ο σκοπός των κινητών βιομηχανικών ρομπότ είναι να αυτοματοποιηθούν επαναλαμβανόμενες, επικίνδυνες και υψηλής αντοχής εργασίες στον χώρο παραγωγής, με αποτέλεσμα να αυτοματοποιηθούν ακόμα και όσες απαιτούν ανθρώπινη ευφυΐα. Σύμφωνα με τον διεθνή οργανισμό τυποποίησης, ISO 8373:2021, ένα βιομηχανικό ρομπότ αποτελεί έναν αυτόματα ελεγχόμενο, επαναπρογραμματιζόμενο πολλαπλών χρήσεων χειριστή (*manipulator*), ο οποίος μπορεί να προγραμματιστεί σε τρεις ή παραπάνω άξονες και μπορεί να είναι τοποθετημένος σε σταθερό σημείο ή σε κινητή πλατφόρμα, με σκοπό τη χρήση του σε εφαρμογές αυτοματισμού σε ένα βιομηχανικό περιβάλλον [14].

Στην παρούσα εργασία μελετάται κυρίως η ικανότητα διαμετακόμισης προϊόντων με τη χρήση βιομηχανικών ρομπότ από ένα σημείο Α προς ένα άλλο σημείο Β. Η ικανότητα του ρομπότ να πλοηγείται αυτόνομα, μεταφέροντας προϊόντα μεταξύ των δύο σημείων, βασίζεται στη χρήση μεθόδων τεχνητής νοημοσύνης, καθώς εξυπηρετούν στην αύξηση της αποδοτικότητας, στην προσαρμογή διάφορων παραλλαγών σε προϊόντα, στην ασφάλεια, στην ελάττωση κόστους, κλπ. Κατά το στάδιο παραγωγής υπάρχει συνεργασία μεταξύ ανθρώπου και ρομπότ, που σημαίνει ότι μόλις ολοκληρωθεί η επεξεργασία ενός προϊόντος ο χρήστης καλεί το ρομποτικό όχημα να μεταφέρει το προϊόν στην επόμενη γραμμή παραγωγής, όπως φαίνεται στην Εικόνα 6β. Αφού το ρομποτικό όχημα φτάσει στη θέση Α, παραλαμβάνει το προϊόν και ξεκινάει τη μεταφορά του προς τον προορισμό του. Κατά τη μεταφορά, το ρομπότ μπορεί να πλοηγηθεί αυτόνομα, αποφεύγοντας εμπόδια, μέχρι να φτάσει στη θέση Β.



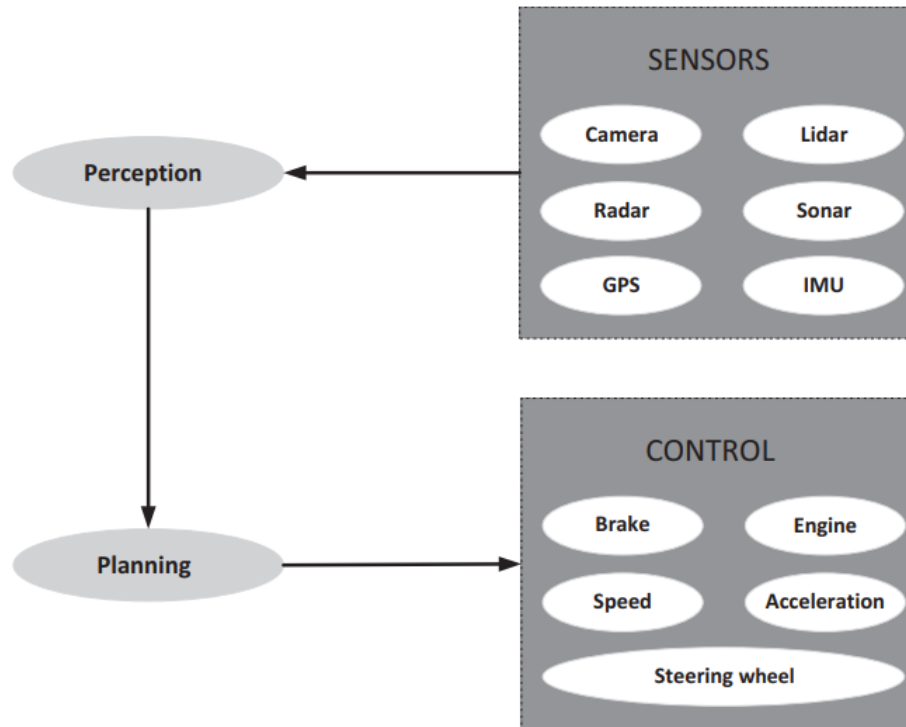
Εικόνα 6 α) Διαδικασία διαμετακόμισης προϊόντων με τη χρήση του ρομπότ MiR100, της εταιρείας MiR [35],
β) Κλήση ρομποτικού οχήματος από τον χρήστη για την παραλαβή ενός επεξεργασμένου προϊόντος [35].

Ένα κριτήριο που επιτρέπει στα βιομηχανικά ρομπότ να προσαρμόζονται εύκολα σε αλλαγές αποτελεί η χρήση αλγορίθμων που εφαρμόζονται σε διαφορετικά σενάρια με μικρές τροποποιήσεις που γίνονται από τους χειριστές των μηχανημάτων. Η ανάγκη για τη χρήση ευφών βιομηχανικών ρομπότ αυξάνεται όταν υπάρχει αβεβαιότητα στον χώρο εργασίας του ρομποτικού οχήματος, η οποία χαρακτηρίζεται από τυχαία τοποθετημένα αντικείμενα, δυναμικά εμπόδια ή την παρουσία ανθρώπων σε αυτόν. Επομένως, το σύστημα ελέγχου που χρησιμοποιεί ο συγκεκριμένος

τύπος ρομπότ ακολουθεί τη θεμελιώδη αρχή του see-think-act, η οποία περιλαμβάνει τέσσερις κύριες λειτουργίες [15]. Πρώτη είναι η λειτουργία της αντίληψης, η οποία αποτελεί το σύστημα που παρέχει στο ρομπότ την ικανότητα να αντιλαμβάνεται, να κατανοεί και να συλλογίζεται και περιλαμβάνει την επεξεργασία, την αναπαράσταση και την ερμηνεία δεδομένων που αντλούνται από τους αισθητήρες. Έπειτα, υπάρχει η λειτουργία της κατανόησης μέσω βάσης δεδομένων, που ενεργοποιεί το σύστημα ελέγχου του ρομπότ, μέσα από την απόκτηση νέων πληροφοριών και γενικεύοντάς τις, ώστε να δημιουργηθούν καινούριες αποστολές με βάση το ιστορικό αποφάσεων. Τρίτη λειτουργία είναι η εκτέλεση εργασιών κάτω από υψηλού επιπέδου οδηγίες και την επίγνωση του περιβάλλοντος, που αποτελεί λόγο που τίθενται γνωστικοί περιορισμοί κατά την εκτέλεση των εργασιών, έτσι ώστε τα ρομπότ να μπορούν να προσδιορίσουν μόνα τους τις καλύτερες πιθανές μελλοντικές ενέργειες προκειμένου να διεκπεραιώσουν την αποστολή που θα τους δοθεί. Η διαδικασία της κατανόησης υποστηρίζεται από τον προσαρμοστικό προγραμματισμό, που αποτελεί την τέταρτη λειτουργία και επιτρέπει την πρόβλεψη γεγονότων και την προετοιμασία της αντίδρασης εκ των προτέρων ώστε να είναι διαχειρίσιμη μια απρόοπτη κατάσταση.

Ένα αυτόνομο ρομποτικό όχημα μπορεί να χωριστεί σε τέσσερα βασικά τμήματα, το σύστημα αισθητήρων, το σύστημα αντίληψης, το σύστημα σχεδιασμού και το σύστημα ελέγχου, όπως φαίνεται στην Εικόνα 7. Το όχημα αντιλαμβάνεται το περιβάλλον στο οποίο βρίσκεται χρησιμοποιώντας αισθητήρες, οι οποίοι συγκροτούν το σύστημα αισθητήρων και αντλούν πληροφορίες σχετικά με τον χώρο, όπως εμπόδια, τοίχους και αδιέξοδα. Οι πληροφορίες αυτές δέχονται, αρχικά, επεξεργασία σε ένα σύστημα αντίληψης και έπειτα μετατρέπονται σε χρήσιμες πληροφορίες. Στη συνέχεια, οι επεξεργασμένες πληροφορίες χρησιμοποιούνται από ένα σύστημα σχεδιασμού για τον προγραμματισμό μελλοντικών ενεργειών που θα ακολουθήσει το ρομπότ κάτω από συγκεκριμένες συνθήκες και για τη δημιουργία μικρής εμβέλειας ή μακροπρόθεσμων σχεδίων. Τέλος, το σύστημα ελέγχου εξασφαλίζει πως το σχέδιο προερχόμενο από το σύστημα σχεδιασμού ακολουθείται από το ρομποτικό όχημα και στέλνει επαναλαμβανόμενες εντολές ελέγχου στο όχημα.

Κατά την κατασκευή ενός ρομποτικού συστήματος καθορίζεται η ικανότητά του να πραγματοποιεί εργασίες και να αντιδρά σε γεγονότα [16]. Επομένως, σχετικά με τη δομή του συστήματος ελέγχου είναι απαραίτητο να προστεθούν οι δυνατότητες για τη λήψη αποφάσεων και για την ικανότητα αντίδρασης σε μη αναμενόμενες καταστάσεις. Αυτό συνεπάγεται πως το ρομποτικό σύστημα θα πρέπει να προγραμματιστεί έτσι ώστε να μπορεί να διαχειριστεί μεταβολές του περιβάλλοντος, οποιαδήποτε χρονική στιγμή ακόμα και κατά τη διάρκεια διεκπεραίωσης εργασιών.



Εικόνα 7 Διάγραμμα συστήματος αυτόνομου ρομποτικού οχήματος [37].

Κατά τη διάρκεια λειτουργίας ενός ρομποτικού οχήματος εκείνο καλείται να εκτελέσει εργασίες τις οποίες συχνά χρειάζεται να επεξεργαστεί κατά τη διάρκεια της εκτέλεσής τους. Αυτές οι τροποποιήσεις οφείλουν να γίνονται σύμφωνα με τις συνθήκες και τους περιορισμούς που έχουν τεθεί κατά την κατασκευή του ρομποτικού συστήματος. Οπότε, για να ικανοποιηθούν οι απαιτήσεις, το σύστημα ελέγχου του αυτόνομου ρομποτικού οχήματος πρέπει να ικανοποιεί τις παρακάτω ιδιότητες:

- **Ευέλικτος προγραμματισμός:** για να θεωρείται ένα ρομπότ αποδοτικό, δεν μπορεί να προγραμματιστεί μόνο για τις συνθήκες που τρέχουν σε ένα συγκεκριμένο περιβάλλον ή για μια μοναδική εργασία αλλά πρέπει να είναι ικανό να διεκπεραιώσει πολλαπλές εργασίες, για τις οποίες δεν θα παρέχονται όλες οι πληροφορίες. Ακόμα, οι συναρτήσεις που δημιουργούνται κατά το στάδιο του προγραμματισμού θα πρέπει να μπορούν να συνδυαστούν εύκολα μεταξύ τους έτσι ώστε η αποστολή να διεκπεραιωθεί και να προκύψει το επιθυμητό αποτέλεσμα.
- **Αυτονομία και προσαρμοστικότητα:** το ρομπότ θα πρέπει να είναι ικανό να εκτελεί τις κατάλληλες ενέργειες προκυμμένον να βελτιώσει το αποτέλεσμα της εργασίας και για να τροποποιήσει τη λειτουργία του αναλογικά με τον βαθμό διεκπεραίωσης της αποστολής και τις οδηγίες εκτέλεσης που του έχουν δοθεί.
- **Αντιδραστικότητα:** το ρομπότ συχνά καλείται να αναλάβει εργασίες που πρέπει να διεκπεραιωθούν μέσα σε συγκεκριμένα χρονικά διαστήματα και ταυτόχρονα να τις εκτελέσει με ορθότητα και αποδοτικότητα, λαμβάνοντας υπόψη και τη δική του ασφάλεια.
- **Σταθερή συμπεριφορά:** οι αντιδράσεις του ρομπότ σε γεγονότα πρέπει να κατευθύνονται από τους στόχους της εργασίας.

- Ανθεκτικότητα: το σύστημα ελέγχου θα πρέπει να είναι σε θέση να εκμεταλλευτεί τον πλεονασμό των λειτουργιών, δηλαδή η ανθεκτικότητα του ρομποτικού συστήματος θα εξασφαλίζει πως ο έλεγχος θα μπορεί να αποκεντρωθεί σε μεγάλο βαθμό.
- Δυνατότητες μάθησης: το ρομπότ θα πρέπει να έχει την επιλογή να αναπτύσσει νέες λειτουργίες και ερμηνείες για μελλοντικές εργασίες, το οποίο μπορεί να γίνει μέσα από τις πληροφορίες που αντλεί το ρομποτικό σύστημα από ήδη διεκπεραιωμένες αποστολές.

Η αρχιτεκτονική του ρομποτικού συστήματος πρέπει να αποτελείται από την κατάλληλη δομή ώστε να υπάρχει αποτελεσματική επικοινωνία μεταξύ των διαδικασιών αξιολόγησης και εκτέλεσης. Επιπρόσθετα, η αξιολόγηση αποτελεί μια στοχοπροσηλωμένη διαδικασία κατά την οποία το ρομπότ παραμένει σε ετοιμότητα σχετικά με εξελίξεις που μπορούν να προκύψουν στο περιβάλλον. Επομένως, αφορά μια διαδικασία που είναι ταυτόχρονα χρονικά οριοθετημένη, εξαρτημένη από τους στόχους και τις απαιτήσεις της εργασίας και υπεύθυνη για τη λήψη αποφάσεων, ώστε να εξασφαλίσει μια γρήγορη ανταπόκριση σε οποιοδήποτε γεγονός λάβει χώρα.

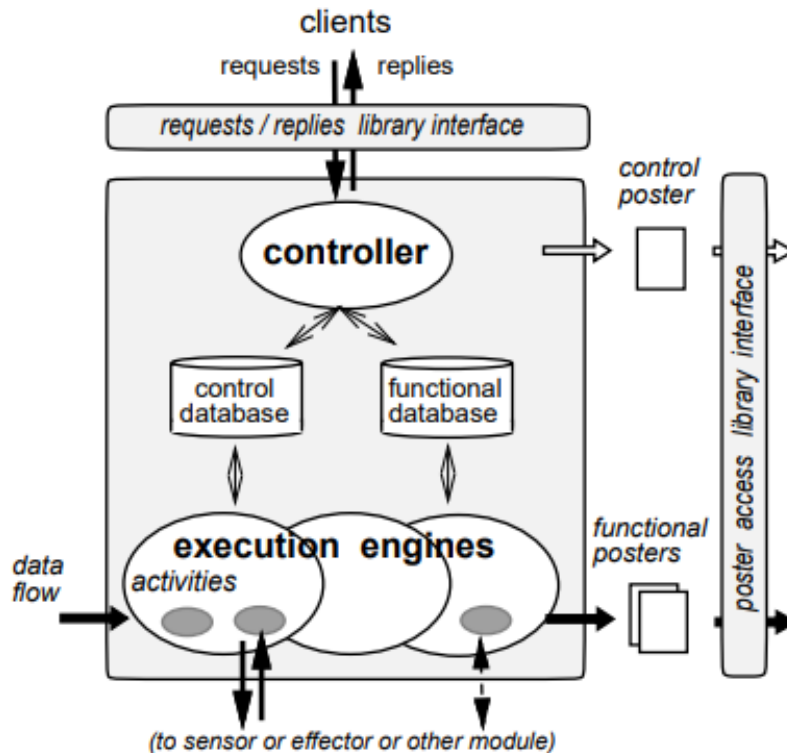
Συγκεκριμένα, η αρχιτεκτονική των αυτόνομων ρομποτικών οχημάτων μπορεί να χωριστεί σε τρία επίπεδα, το επίπεδο λειτουργίας, το επίπεδο εκτέλεσης και το επίπεδο αποφάσεων, με το καθένα από αυτά να έχει διαφορετικούς περιορισμούς έτσι ώστε να μπορούν να διαχειρίζονται διαφορετικά δεδομένα.

Επίπεδο λειτουργίας

Το επίπεδο λειτουργίας (*functional*) αποτελείται από τον συνδυασμό των συναρτήσεων προγραμματισμού και ενός προκαθορισμένου συνόλου εργασιών, όπως την κατασκευή μονοπατιών, τη ρομποτική όραση και τον εντοπισμό θέσης, που προκαλούν μια κατάσταση δράσης – αντίδρασης. Οι συναρτήσεις βρίσκονται μέσα σε μια λειτουργική μονάδα, η οποία είναι ένα τμήμα λογισμικού ικανό να εκτελέσει έναν αριθμό συγκεκριμένων λειτουργιών μέσα από την επεξεργασία εισόδων ενός ρομποτικού συστήματος ή άλλων λειτουργικών μονάδων. Ακόμα, επιτρέπουν στο ρομπότ να κινηθεί, να αντιλαμβάνεται, να κάνει υπολογισμούς διαδρομών, να εκτιμάει τη θέση που έχει συγκριτικά με τον χώρο στον οποίο κινείται, κλπ. Όλες αυτές οι διαδικασίες χρησιμοποιούν αριθμητικά δεδομένα και πραγματοποιούν αλγοριθμικούς υπολογισμούς. Για παράδειγμα, ο εντοπισμός θέσης μπορεί να γίνει με διάφορους τρόπους, με ένα παράδειγμα να αποτελεί ο συνδυασμός της οδομετρίας και της ρομποτικής όρασης.

Μια λειτουργική μονάδα μπορεί να διαβάσει δεδομένα που εξάγονται από άλλες λειτουργικές μονάδες και να εξάγει αποτελέσματα σε δομές δεδομένων, τα οποία έχει επεξεργαστεί βασισμένη σε συγκεκριμένες προδιαγραφές. Οι προδιαγραφές μπορούν να είναι είτε φυσικές είτε λογικές, για παράδειγμα αισθητήρες, τελεστές ή δεδομένα που προέρχονται από χάρτες, εικόνες ή τη θέση του ρομποτικού οχήματος. Ο σκοπός μιας λειτουργικής μονάδας είναι ο έλεγχος των δεδομένων εισόδων, όπως για παράδειγμα ο έλεγχος της εγκυρότητας των αιτούμενων παραμέτρων και των δυνατοτήτων της εφαρμογής των απαιτούμενων ενεργειών. Η κωδικοποίηση μιας λειτουργικής μονάδας γίνεται σύμφωνα με ένα γενικό πρότυπο, το οποίο χαρακτηρίζεται από τον έλεγχο και την εκτέλεση. Υπάρχουν δύο μέρη στην οργάνωση μιας λειτουργικής μονάδας (Εικόνα 8), το πρώτο μέρος αποτελεί τον ελεγκτή (*controller*) και το δεύτερο τους μηχανισμούς εκτέλεσης (*execution engines*). Ο ελεγκτής χειρίζεται τη λειτουργική μονάδα με βάση τα αιτήματα του χειριστή και την τρέχουσα κατάσταση της και στη συνέχεια εφαρμόζεται ως μια μη συγχρονισμένη

διαδικασία που ενεργοποιείται μετά από ένα νέο αίτημα ή μια εσωτερική αλλαγή. Οι μηχανισμοί εκτέλεσης διεξάγουν ενέργειες που απαιτούνται από τον ελεγκτή. Κάθε μηχανισμός είναι μια περιοδική ή μη περιοδική διαδικασία που σχετίζεται με την έξοδο λειτουργικών συναρτήσεων.



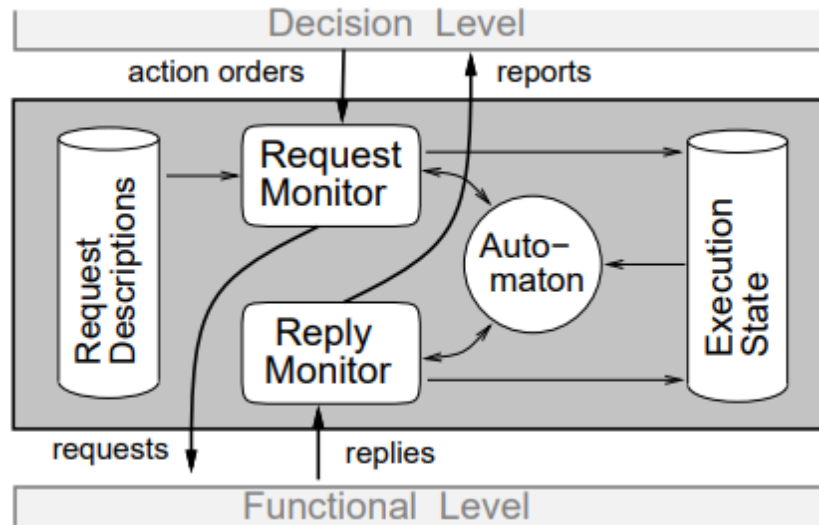
Εικόνα 8 Οργάνωση λειτουργικής μονάδας [16]

Επίπεδο εκτέλεσης

Το επίπεδο εκτέλεσης αποτελεί ένα σύστημα αντίδρασης χωρίς προγραμματιστικές ικανότητες και θεωρείται το επίπεδο σύνδεσης μεταξύ των επιπέδων αποφάσεων και λειτουργίας. Συγκεκριμένα, λαμβάνει από το επίπεδο αποφάσεων τη σειρά με την οποία θα εκτελεστούν οι ενέργειες και επιπλέον επιλέγει, παραμετροποιεί και συγχρονίζει δυναμικά τις συναρτήσεις που έχουν προκύψει από το επίπεδο λειτουργίας.

Η επιλογή και η εγκατάσταση των παραμέτρων εξαρτώνται από το ζητούμενο και από την τρέχουσα κατάσταση του ρομποτικού συστήματος. Η κατάσταση του συστήματος ελέγχεται και συντηρείται από το επίπεδο εκτέλεσης με βάση τις εργασίες που βρίσκονται σε εξέλιξη και τις εξόδους προηγούμενων διαδικασιών. Επίσης, συστηματικά στέλνεται μια αναφορά προς το επίπεδο αποφάσεων σχετικά με τα αποτελέσματα της επεξεργασίας, όπως ο εντοπισμός θέσης του ρομπότ, η μη εύρεση του στόχου και ο υπολογισμός πορείας, έτσι ώστε να ενεργοποιηθεί η κατάσταση παρακολούθησης της διαδικασίας και να γίνει επιλογή της επόμενης ενέργειας. Το συγκεκριμένο επίπεδο ελέγχει τόσο τη ροή ελέγχου όσο και τη ροή δεδομένων του επιπέδου λειτουργίας, το οποίο συμβαίνει συγκεκριμένα χρησιμοποιώντας δύο συστήματα, το σύστημα αίτησης (*request monitor*) και το σύστημα ανταπόκρισης (*reply monitor*), όπως παρουσιάζεται και στην Εικόνα 9. Το σύστημα αίτησης αντιδρά σε αποφάσεις από το υψηλότερο επίπεδο και επιλέγει τα

ικανοποιητικά αιτήματα, σύμφωνα με τη βάση δεδομένων που περιέχει τις περιγραφές των αιτημάτων, τις παραμέτρους εισόδων και την κατανομή των δεδομένων. Το σύστημα ανταπόκρισης ελέγχει τις εξόδους των συναρτήσεων, παραλαμβάνει τα αποτελέσματα του επιπέδου λειτουργίας και επιστρέφει τις αναφορές στο επίπεδο αποφάσεων.



Εικόνα 9 Οργάνωση επιπέδου εκτέλεσης [16]

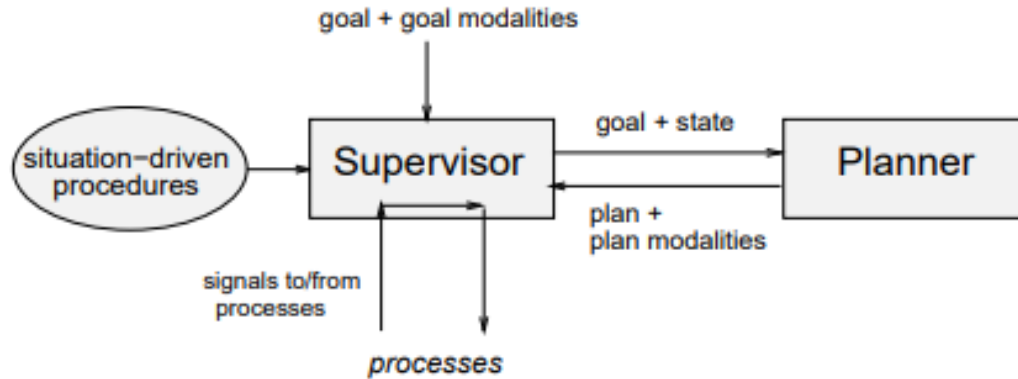
Επίπεδο αποφάσεων

Το επίπεδο αποφάσεων είναι υπεύθυνο για όλες τις διαδικασίες, για τις οποίες απαιτούνται προϋπάρχουσες γνώσεις σχετικά με το ζητούμενο της αποστολής και την περιγραφή της εκτέλεσής τους. Το συγκεκριμένο επίπεδο περιλαμβάνει τις ικανότητες για τον προγραμματισμό, τη δημιουργία και τη λήψη αποφάσεων ενώ ταυτόχρονα παραμένει ενεργοποιημένο το σύστημα αναγνώρισης νέων εισερχόμενων γεγονότων. Επομένως, για να ενεργοποιηθεί η σύνδεση των ικανοτήτων της λήψης αποφάσεων και της αντίδρασης, υπάρχει το πρότυπο του σχεδιαστή (*planner*) – επιθεωρητή (*supervisor*), όπως φαίνεται στην Εικόνα 10, το οποίο παρέχει μια εγγύηση προς τον χρήστη, πως για μια απρόοπτη κατάσταση θα μπορεί να βρεθεί λύση σε ένα συγκεκριμένο χρονικό διάστημα.

Ο σχεδιαστής δημιουργεί μια σειρά ενεργειών που είναι απαραίτητες για τη διεκπεραίωση της αποστολής ή για την επίτευξη ενός στόχου, οι οποίες στη συνέχεια χρησιμοποιούνται ως δεδομένα από τον επιθεωρητή. Επίσης, διαθέτει μια περιγραφή της κατάστασης του περιβάλλοντος και του στόχου, δημιουργώντας έτσι ένα σχέδιο επίλυσης, το οποίο περιλαμβάνει χαρακτηριστικά σχετικά με το κόστος που απαιτείται για την επιτυχία της αποστολής και την ανθεκτικότητά της.

Ο επιθεωρητής αλληλοεπιδρά με τον σχεδιαστή και με τα άλλα επίπεδα, ελέγχει την εκτέλεση της εργασίας και αντιδρά σε εισερχόμενες καταστάσεις. Οι εισερχόμενες καταστάσεις παρουσιάζονται ως ένα σύνολο νέων λειτουργιών, που μπορούν να ανταλλάξουν σήματα με τον επιθεωρητή και αντιστοιχούν σε μελλοντικές ενέργειες του ρομπότ και σε γεγονότα σχετικά με αλλαγές που συμβαίνουν στον χώρο εργασίας του ρομποτικού οχήματος. Οι νέες λειτουργίες ελέγχονται από τον επιθεωρητή για να επιβεβαιώσει πως τηρούνται οι προδιαγραφές που έχουν δοθεί για τη

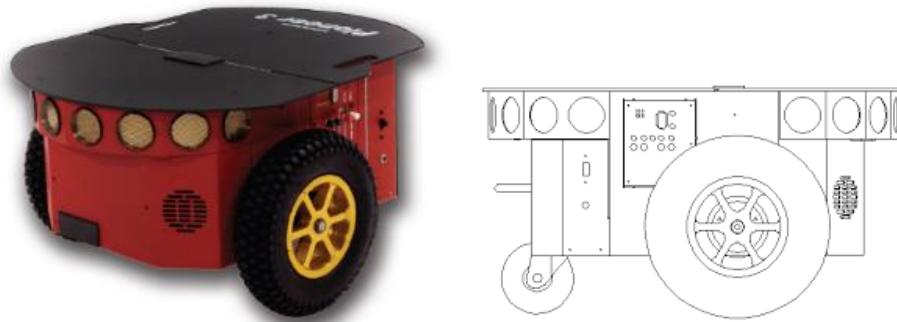
συνολική λειτουργία του ρομποτικού συστήματος. Ακόμα, ο επιθεωρητής είναι υπεύθυνος για την παρακολούθηση της εκτέλεσης του σχεδίου επίλυσης, με αποτέλεσμα να πραγματοποιεί ανιχνεύσεις και αξιολογήσεις στις λειτουργίες του ρομπότ και να λαμβάνει αποφάσεις σχετικά με την αποτελεσματικότητά τους σε πραγματικό χρόνο. Τέλος, για την επιτυχημένη εκτέλεση του σχεδίου επίλυσης, ο επιθεωρητής χρησιμοποιεί αλγορίθμους για τη λήψη αποφάσεων, που μπορούν να εγγυηθούν την εκτέλεση μιας εργασίας σε συγκεκριμένο χρονικό διάστημα και μια βάση δεδομένων, την οποία λαμβάνει υπόψη για να αναγνωρίζει επαναλαμβανόμενα μοτίβα καθώς περιλαμβάνει ένα σύνολο από ζεύγη κατάστασης – αποτελέσματος.



Εικόνα 10 Πρότυπο σχεδιαστή – επιθεωρητή [16]

1.2 Σύστημα Μετάδοσης Κίνησης και Ελέγχου

Για το ρομποτικό όχημα που κατασκευάστηκε στο πρόγραμμα προσομοίωσης χρησιμοποιήθηκαν τα κινηματικά χαρακτηριστικά του μοντέλου Pioneer 3-DX (Εικόνα 11α και β), της εταιρείας Mobile Robots, το οποίο διαθέτει δύο τροχούς με δύο προγραμματιζόμενους κινητήρες διαφορετικού μηχανισμού και έναν επιπλέον ροδάκι (*castor wheel*) που εξυπηρετεί στην ισορροπία του οχήματος [17].



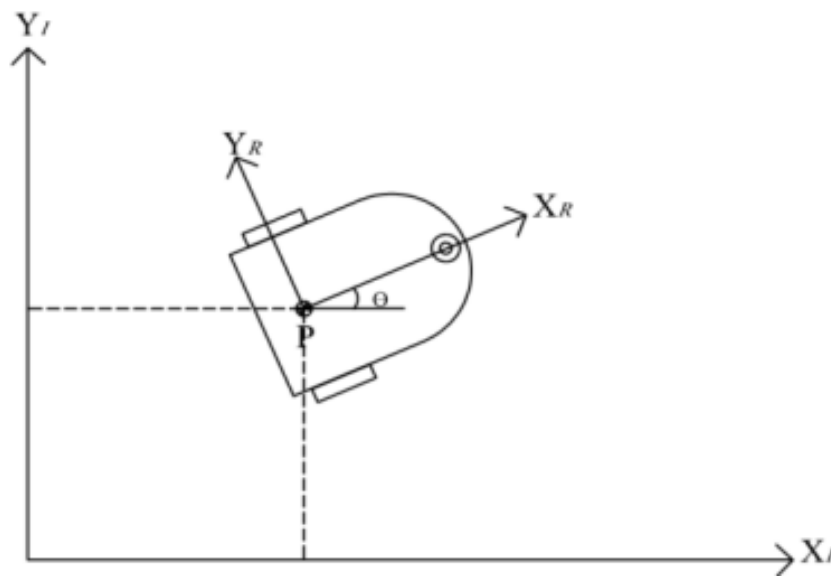
Εικόνα 11 α) Ρομποτικό όχημα Pioneer 3-DX, Mobile Robots [17] β) Πλάγια όψη ρομποτικού οχήματος Pioneer 3-DX [17].

Η μηχανική κατασκευή ενός ρομποτικού οχήματος πρέπει να ελέγχεται με τέτοιο τρόπο ώστε να εκτελεί ενέργειες, πραγματοποιώντας τους στόχους του ζητήματος. Ένα σύστημα ελέγχου εμπεριέχει τρία διαφορετικά τμήματα, το τμήμα αντίληψης, το τμήμα επεξεργασίας, το γνωστικό τμήμα και το τμήμα εκτέλεσης ενεργειών [7]. Με βάση τις πληροφορίες που λαμβάνονται από

τους αισθητήρες και τον στόχο του ζητούμενου που έχει δοθεί στο ρομπότ, το γνωστικό τμήμα και το τμήμα ελέγχου πρέπει να αποφασίσουν τη διαδικασία που θα ακολουθήσουν και ποιες ενέργειες θα εκτελεστούν έτσι ώστε να διεκπεραιωθεί το ζητούμενο.

1.2.1 Κινηματική ρομποτικού οχήματος διαφορικής οδήγησης

Η διαδικασία κατανόησης των κινήσεων ενός ρομπότ ξεκινάει με τη διαδικασία επεξήγησης της συνεισφοράς που έχει κάθε τροχός στην κίνηση ολόκληρου του ρομποτικού συστήματος [18]. Η θέση ενός ρομποτικού οχήματος μπορεί να δοθεί από δύο συστήματα συντεταγμένων, το απόλυτο σύστημα αναφοράς (*global reference frame*), που αποτελεί το σύστημα συντεταγμένων του περιβάλλοντος μέσα στο οποίο κινείται το ρομποτικό όχημα και το σχετικό σύστημα αναφοράς (*local reference frame*), που αποτελεί το σύστημα συντεταγμένων του ρομπότ [8]. Επομένως, το πρώτο βήμα αποτελεί η περιγραφή της θέσης και του προσανατολισμού ενός ρομπότ στο επίπεδο, δημιουργώντας μια σχέση μεταξύ του απόλυτου συστήματος αναφοράς του επιπέδου και ενός σχετικού συστήματος αναφοράς. Όπως παρατηρείται στην Εικόνα 12, οι άξονες X_I και Y_I προσδιορίζουν μια τυχαία θέση στο επίπεδο, ενός απόλυτου συστήματος αναφοράς με αρχή των αξόνων $O: \{X_I, Y_I\}$ [18].



Εικόνα 12 Το απόλυτο σύστημα αναφοράς και το σχετικό σύστημα αναφοράς του ρομποτικού οχήματος [20].

Μέσα στο απόλυτο σύστημα αναφοράς δίνεται η θέση $P[x \ y \ z]$ του ρομποτικού οχήματος που εκφράζεται στο καρτεσιανό σύστημα συντεταγμένων και προσδιορίζεται από τις συντεταγμένες x , y και τη γωνία θ , η οποία υπολογίζεται από τη γωνία που σχηματίζεται μεταξύ του απόλυτου και του σχετικού συστήματος αναφοράς. Αξιοποιώντας αυτά τα τρία στοιχεία δίνεται η πόζα του ρομποτικού οχήματος, για την οποία χρησιμοποιείται η μεταβλητή ξ_I :

$$\xi_I = [x \ y \ \theta]^T$$

Για να περιγραφεί η θέση του ρομποτικού οχήματος σχετικά με τις κινήσεις των εξαρτημάτων είναι απαραίτητο να καταγραφούν οι κινήσεις συγκριτικά με τους άξονες του απόλυτου συστήματος αναφοράς και συγκριτικά με τους άξονες του σχετικού συστήματος αναφοράς του

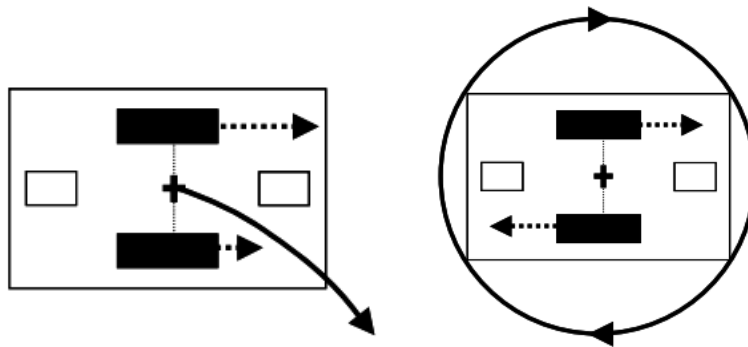
ρομποτικού οχήματος. Αυτή η καταγραφή γίνεται χρησιμοποιώντας τον πίνακα περιστροφής $R(\theta)$ και μπορεί να χρησιμοποιηθεί για να καταγράψει τις κινήσεις στο απόλυτο σύστημα αναφοράς $\{X_I, Y_I\}$ σε σχέση με το σχετικό σύστημα αναφοράς $\{X_R, Y_R\}$. Αυτή η διαδικασία δίνεται από τη σχέση $\xi_R = R(\theta)\xi_I$, καθώς ο υπολογισμός αυτής της διαδικασίας εξαρτάται από την τιμή της γωνίας θ . Επομένως, για μια τιμή ταχύτητας $(\dot{x}, \dot{y}, \dot{\theta})$ στο απόλυτο σύστημα αναφοράς μπορεί να υπολογιστεί η κίνηση σε σχέση με τους τοπικούς άξονες X_R και Y_R του ρομπότ:

$$\xi_R = R(\theta)\xi_I = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

Διαφορική οδήγηση

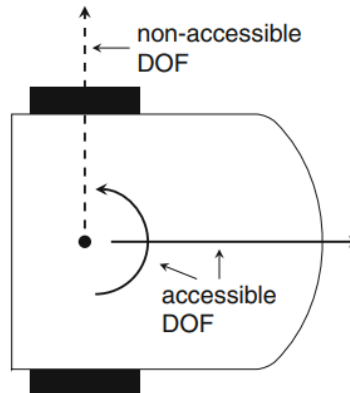
Ένα αυτόνομο ρομποτικό όχημα που χρησιμοποιεί το διαφορικό σύστημα οδήγησης (*differential drive*) σημαίνει πως αποτελείται από δύο τροχούς που κινούνται με τη χρήση δύο ανεξάρτητων μεταξύ τους κινητήρων [6]. Προκειμένου να υπάρχει ισορροπία στην κατασκευή είναι απαραίτητο να υπάρχουν ένα ή δύο επιπλέον βοηθητικά ροδάκια, ανάλογα με το σημείο που έχουν τοποθετηθεί οι τροχοί. Στο MIRc ρομποτικό όχημα επιλέχθηκε να τοποθετηθούν δύο βοηθητικές ρόδες, ένα στο μπροστινό μέρος και ένα στο πίσω μέρος της βάσης της κατασκευής.

Ο κάθε τροχός έχει από έναν ενεργοποιητή, που επιτρέπει στο ρομπότ να κινηθεί κατά μήκος του ίδιου άξονα ευθεία και όπισθεν, εφαρμόζοντας διαφορετική ισχύς. Η τιμή ισχύος στους κινητήρες (ενεργοποιητές) μπορεί να κυμαίνεται μεταξύ μιας μέγιστης και μιας ελάχιστης τιμής. Συγκεκριμένα, η μέγιστη τιμή χαρακτηρίζει τη μέγιστη δυνατή ισχύς που δίνεται στον κινητήρα, για την ευθεία κίνηση, η ελάχιστη τιμή χαρακτηρίζει αντίστοιχα την μέγιστη ισχύς, για την όπισθεν κίνηση και η ενδιάμεση, που συνήθως αποτελεί την τιμή 0, χαρακτηρίζει τη μηδενική ισχύς, που σημαίνει πως αν και στους δύο τροχούς δοθεί μηδενική τιμή το ρομποτικό όχημα είναι σταματημένο. Συνεπώς, η ευθεία κίνηση επιτυγχάνεται όταν και οι δύο τροχοί περιστρέφονται με την ίδια τιμή, η δεξιά στροφή επιτυγχάνεται όταν ο αριστερός τροχός περιστρέφεται με υψηλότερη τιμή από τον δεξί τροχό και η αριστερή στροφή όταν ο δεξιός τροχός περιστρέφεται με υψηλότερη τιμή από τον αριστερό τροχό. Ακόμα, ο συγκεκριμένος τύπος ρομπότ μπορεί να περιστραφεί γύρω από το κέντρο του όταν ο ένας τροχός περιστραφεί σε αντίθετη κατεύθυνση από τον άλλον και με ίδια τιμή, όπως φαίνεται στην Εικόνα 13.



Εικόνα 13 Περιστροφή ρομποτικού οχήματος διαφορικής οδήγησης [21].

Ένα ρομπότ διαφορικής οδήγησης έχει τρεις βαθμούς ελευθερίας αλλά μόνο δύο ενεργοποιητές. Ο ένας ενεργοποιητής αφορά τον κινητήριο τροχό που εφαρμόζεται κατά μήκος του ρομποτικού οχήματος και ενεργοποιεί την ευθεία και όπισθεν κίνηση και ο δεύτερος ενεργοποιητής αφορά τον τροχό περιστροφής και επιτρέπει τον προσανατολισμό, όπως φαίνεται στην Εικόνα 14. Το ρομπότ δεν μπορεί να περιστραφεί γύρω από τον κατακόρυφο άξονα ή πλαγίως, επομένως έχει μόνο έναν βαθμό κινητικότητας, $\delta_m = 1$. Ένα ακόμα χαρακτηριστικό των ρομποτικών οχημάτων διαφορικής οδήγησης είναι πως εκφράζονται από μη-ολονομικούς περιορισμούς (*non-holonomic constraints*), που εκφράζεται από τη σχέση $\dot{x} \sin \theta - \dot{y} \cos \theta = 0$. Αυτό οφείλεται στο ότι ο αριθμός των βαθμών κινητικότητας είναι μικρότερος από τον αριθμό των βαθμών ελευθερίας με αποτέλεσμα να δημιουργούνται περιορισμοί κατά την κίνηση του ρομποτικού οχήματος, όπως για παράδειγμα ο περιορισμός που υπάρχει λόγω της μη περιστροφής των τροχών κατά τον άξονα z. Ο συγκεκριμένος περιορισμός έχει ως αποτέλεσμα το όχημα να πρέπει να εφαρμόσει πολλαπλούς ή περίπλοκους ελιγμούς έτσι ώστε να στρίψει μέσα σε έναν στενό διάδρομο ή να παρκάρει παράλληλα. Συνεπώς, οι περιορισμοί που υπάρχουν σε ένα κινηματικό μοντέλο διαφορικής οδήγησης αντικατοπτρίζουν την περιορισμένη ευελιξία ενός ρομποτικού οχήματος που συνήθως οφείλεται στις επιτρεπτές τιμές ή και τα όρια που θέτονται για την ταχύτητα και την επιτάχυνση του ρομπότ [19]. Οι περιορισμοί αυτοί εκφράζονται μέσω κινηματικών και δυναμικών μοντέλων και προτείνεται να υπολογίζονται κατά τη διαδικασία του προγραμματισμού κίνησης (*motion planning*).

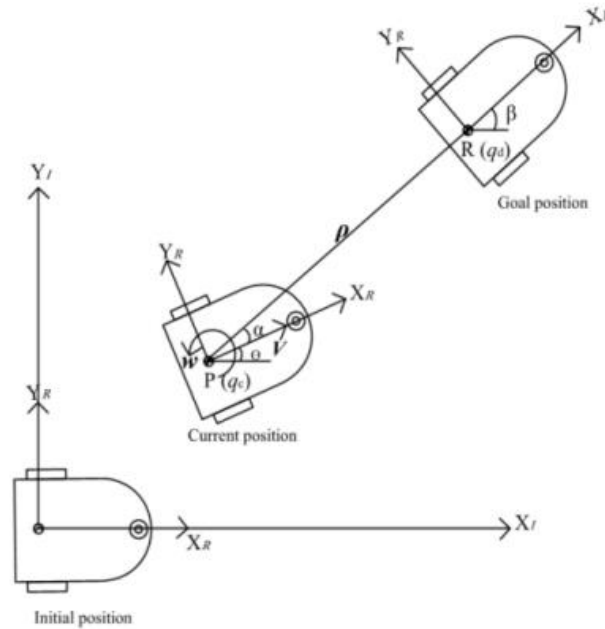


Εικόνα 14 Προσβάσιμοι και μη προσβάσιμοι βαθμοί ελευθερίας ενός ρομποτικού οχήματος διαφορικής οδήγησης [6].

Κινηματικές εξισώσεις

Η κινηματική αποτελεί την επιστήμη που μελετά την κίνηση και τη συμπεριφορά μηχανικών συστημάτων χωρίς να λαμβάνει υπόψη τις δυνάμεις που την προκαλούν [6]. Επομένως, αναζητείται η θέση, η ταχύτητα, η επιτάχυνση και κάθε παράγωγος μεταβλητής που μπορεί να περιγράψει μια θέση. Για να μπορεί να κατασκευαστεί το κατάλληλο ρομποτικό σύστημα που θα εκτελεί με ακρίβεια τις ενέργειες που το αναθέτονται πρέπει να κατανοηθεί η μηχανική συμπεριφορά του ρομπότ. Για να γίνει αυτό χρειάζεται να μελετηθούν οι κινηματικές εξισώσεις, συνεπώς, θεωρούμε ένα ρομποτικό όχημα διαφορικής οδήγησης, το οποίο βρίσκεται σε μια τυχαία θέση $P(q_c) = [x_c, y_c, \theta_c]$ που βρίσκεται σε μη μηδενική απόσταση από τη θέση στόχος $R(q_d) =$

$[x_d, y_d, \beta]$ και η οποία ορίζεται με βάση το απόλυτο σύστημα αναφοράς, όπως φαίνεται στην Εικόνα 15 [20].



Εικόνα 15 Διανύσματα θέσης και προσανατολισμού ρομποτικού οχήματος [20].

Το σύστημα συντεταγμένων του ρομπότ αποτελείται από τον συνδυασμό της γραμμικής ταχύτητας v και της γωνιακής ταχύτητας ω , οι οποίες διαμορφώνουν τις παρακάτω εξισώσεις:

$$\dot{x} = v \cdot \cos\theta$$

$$\dot{y} = v \cdot \sin\theta$$

$$\dot{\theta} = \omega$$

όπου, τα $v \cdot \cos\theta$ και $v \cdot \sin\theta$ αποτελούν τις συνιστώσες της γραμμικής ταχύτητας v , που βρίσκονται κατά μήκος των αξόνων X και Y και οι συντεταγμένες x, y και η γωνία περιστροφής θ υπολογίζονται με βάση το σχετικό σύστημα αναφοράς. Παρομοίως, μπορεί να παρουσιαστεί και η θέση του ρομπότ σε πολικές συντεταγμένες περιλαμβάνοντας την απόσταση σφάλματος, με $\rho > 0$, σύμφωνα με τις παρακάτω σχέσεις:

$$\dot{\rho} = -v \cdot \cos(\beta - \theta) = -v \cdot \cos\alpha$$

$$\dot{\beta} = v \frac{\sin\alpha}{\rho}$$

$$\dot{\theta} = \omega$$

Οπότε, προκύπτει ότι η γωνία $\alpha = \beta - \theta$, είναι η γωνία ανάμεσα στο σχετικό σύστημα αναφοράς του ρομποτικού οχήματος $[X_R, Y_R]$ και το απόλυτο σύστημα αναφοράς και το διάνυσμα $\rho = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$ είναι η απόσταση μεταξύ των δύο συστημάτων συντεταγμένων.

Οπότε, προκύπτουν οι παρακάτω σχέσεις, οι οποίες εφαρμόζονται για τον έλεγχο ανατροφοδότησης κατά το στάδιο των ελιγμών του ρομπότ.

$$\dot{\rho} = -v \cdot \cos\alpha$$

$$\dot{\alpha} = -\omega + v \frac{\sin\alpha}{\rho}$$

$$\dot{\beta} = v \frac{\sin\alpha}{\rho}$$

Κινηματικό μοντέλο

Για να υπολογιστεί η τρέχουσα πορεία που ακολουθεί το ρομποτικό όχημα είναι απαραίτητο να παρακολουθούνται συνεχώς οι κωδικοποιητές των δύο τροχών και για να γίνει αυτό πρέπει να προσδιοριστούν οι τιμές s_L και s_R σε μέτρα, που αποτελούν τις αποστάσεις που το όχημα έχει πλοηγηθεί λόγω του αριστερού και του δεξιού τροχού, αντίστοιχα [21]. Οπότε, διαιρώντας τον υπολογισμένο αριθμό ελέγχων με τον αριθμό των ελέγχων ανά περιστροφή και έπειτα πολλαπλασιάζοντας το αποτέλεσμα με την περίμετρο των τροχών, δίνονται οι τιμές για τις αποστάσεις πλοήγησης ανά τροχό.

$$s_L = 2\pi \cdot r \cdot \text{ticks}_L / \text{ticks_per_rev}$$

$$s_R = 2\pi \cdot r \cdot \text{ticks}_R / \text{ticks_per_rev}$$

όπου, γνωστά δεδομένα θεωρούνται η τιμή r που ορίζεται η ακτίνα των τροχών, η τιμή d που ορίζεται η απόσταση μεταξύ των τροχών, η τιμή ticks_per_rev που ορίζεται ο αριθμός των ελέγχων ενός κωδικοποιητή για μια ολόκληρη περιστροφή, η τιμή ticks_L που ορίζεται ο αριθμός των ελέγχων κατά τη μέτρηση του αριστερού κωδικοποιητή και η τιμή ticks_R που ορίζεται ο αριθμός των ελέγχων κατά τη μέτρηση του δεξιού κωδικοποιητή. Συνεπώς η συνολική απόσταση που έχει διανύσει το ρομποτικό όχημα υπολογίζεται από τη σχέση: $s = \frac{(s_L + s_R)}{2}$.

Θεωρώντας πως το ρομποτικό όχημα ακολουθεί μια κυκλική πορεία οι αποστάσεις s_L και s_R μπορούν να οριστούν ως ένα μέρος ενός κύκλου πολλαπλασιασμένο με την ακτίνα περιστροφής του κάθε τροχού αντίστοιχα. Εάν η ακτίνα περιστροφής του κέντρου του οχήματος ορίζεται από τη μεταβλητή c , τότε κατά την αριστερή περιστροφή η ακτίνα περιστροφής του δεξιού τροχού υπολογίζεται από τη σχέση $c + \frac{d}{2}$, ενώ η ακτίνα περιστροφής του αριστερού τροχού υπολογίζεται από τη σχέση $c - \frac{d}{2}$.

$$s_R = \varphi \cdot \left(c + \frac{d}{2}\right)$$

$$s_L = \varphi \cdot \left(c - \frac{d}{2}\right)$$

Με την αφαίρεση των δύο παραπάνω σχέσεων απαλείφεται η μεταβλητή c και υπολογίζοντας το αποτέλεσμα ως προς φ προκύπτει η σχέση:

$$\varphi = \frac{(s_R - s_L)}{d}$$

Χρησιμοποιώντας τις ταχύτητες $v_{L,R}$ των τροχών στη θέση των αποστάσεων $s_{L,R}$ και την περιστροφή των τροχών ανά δευτερόλεπτο $\dot{\theta}_{L,R}$ με ακτίνες r για τον αριστερό και δεξί τροχό, προκύπτει ότι:

$$v_R = 2\pi r \cdot \dot{\theta}_R$$

$$v_L = 2\pi r \cdot \dot{\theta}_L$$

Από τις παραπάνω σχέσεις προκύπτει η ευθεία κινηματική (forward kinematics), η οποία μπορεί να προβλέψει την ολική ταχύτητα του ρομποτικού οχήματος στο απόλυτο σύστημα αναφοράς.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = 2\pi r \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{d} & \frac{1}{d} \end{bmatrix} \begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix}$$

όπου, v ορίζεται η γραμμική ταχύτητα του ρομποτικού οχήματος, ω ορίζεται η γωνιακή ταχύτητα και $\dot{\theta}_{L,R}$ ορίζεται η ταχύτητα περιστροφής ανά δευτερόλεπτο, κάθε τροχού ξεχωριστά.

Η κινηματική αποτελεί την επιστήμη που μελετά την κίνηση χωρίς να λαμβάνει υπόψη τις δυνάμεις που την προκαλούν [6]. Επομένως, αναζητείται η θέση, η ταχύτητα, η επιτάχυνση και κάθε παράγωγος μεταβλητής που μπορεί να περιγράψει μια θέση. Συγκεκριμένα, για τον προσδιορισμό της κίνησης ενός ρομποτικού οχήματος, όπου είναι γνωστή η γεωμετρία και η ταχύτητα των τροχών, χρησιμοποιείται η ευθεία κινηματική [18]. Η αντίστροφη κινηματική προκύπτει λαμβάνοντας υπόψη την παραπάνω μεθοδολογία και λύνοντας ως προς τις ταχύτητες των τροχών, ξεχωριστά. Επιπλέον, δίνονται οι τιμές για την απαιτούμενη ταχύτητα, δηλαδή για την επιθυμητή γραμμική και την περιστροφική κίνηση. Ο πίνακας της αντίστροφης κινηματικής προκύπτει αναστρέφοντας τον πίνακα της ευθείας κινηματικής, δίνοντας τον παρακάτω πίνακα:

$$\begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix} = \frac{1}{2\pi r} \begin{bmatrix} 1 & -\frac{d}{2} \\ 1 & \frac{d}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

1.2.2 Έλεγχος και ευστάθεια

Ένα σημαντικό χαρακτηριστικό που χρειάζεται να μελετηθεί κατά τη σχεδίαση ενός συστήματος αποτελεί η απόλυτη και σχετική ευστάθειά του. Δηλαδή κατά πόσο μπορεί να ισορροπήσει σε συγκεκριμένα σημεία λειτουργίας και πόσο ευσταθές είναι κάτω από συγκεκριμένες λειτουργίες. Ο σκοπός ενός κινηματικού ελεγκτή (*kinematic controller*) είναι να ακολουθήσει μια διαδρομή που περιγράφεται από τη θέση του ή και από τις ταχύτητές του, ως μια χρονική συνάρτηση [18]. Αυτό γίνεται συνήθως διαιρώντας το σύνολο της διαδρομής σε μικρότερα τμήματα τα οποία έχουν ξεκάθαρο σχήμα, δηλαδή αποτελούν μια ευθεία ή μια κυκλική τροχιά. Συνεπώς, το πρόβλημα ελέγχου αφορά τη δημιουργία ενός ομαλού μονοπατιού, που αποτελείται από μικρότερα τμήματα και που έχει σκοπό να οδηγήσει το ρομποτικό όχημα από την αρχική του θέση προς την

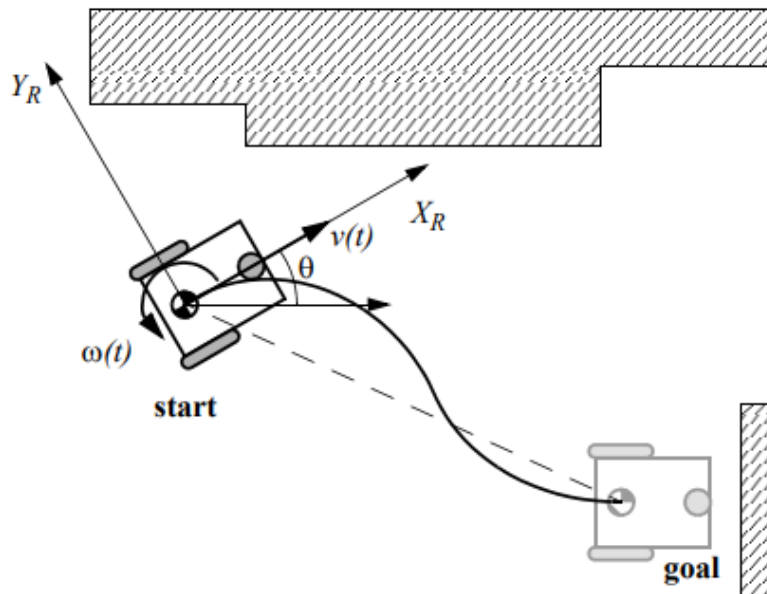
επιθυμητή τελική θέση. Η συγκεκριμένη προσέγγιση μπορεί να θεωρηθεί ως ένα σύστημα ελέγχου κίνησης ανοιχτού βρόχου καθώς η θέση του ρομπότ δεν ανατροφοδοτείται από τον έλεγχο της ταχύτητας ή της θέσης του.

Έλεγχος ανατροφοδότησης δεδομένων

Μια συνηθισμένη προσέγγιση σε ένα σύστημα ελέγχου κίνησης αποτελεί η χρήση ενός ηλεκτρικής ανατροφοδότησης καταστάσεων, ο οποίος κατά τη διαδικασία της κατασκευής ενός μονοπατιού τοποθετεί ενδιάμεσους στόχους πάνω στην επιθυμητή διαδρομή. Η λύση στο πρόβλημα σταθεροποίησης της ανατροφοδότησης του ελέγχου ενός ρομποτικού οχήματος διαφορικής οδήγησης είναι η αναζήτηση ενός πίνακα ελέγχου K . Οπότε, έστω μια τυχαία διάταξη ενός ρομποτικού οχήματος και μια προκαθορισμένη τελική διάταξη, όπως φαίνεται στην Εικόνα 16. Επιπλέον, η πόζα σφάλματος δίνεται στο σχετικό σύστημα αναφοράς του ρομποτικού οχήματος $\{X_R, Y_R, \theta\}$ και αποτελεί το διάνυσμα $e = {}^R[x, y, \theta]^T$, όπου x, y και θ αποτελούν τις συντεταγμένες της τελικής θέσης του ρομποτικού οχήματος. Επιπλέον, ο πίνακας ελέγχου K έχει τέτοια μορφή ώστε κατά την σχεδίαση να βρεθεί η κατάλληλη σχέση ελέγχου $u = [v(t) \ \omega(t)]$ έτσι ώστε το ρομπότ να φτάσει στον επιθυμητό στόχο μέσα σε ένα πεπερασμένο χρονικό διάστημα και το σφάλμα e να τείνει προς το μηδέν, $\lim_{t \rightarrow \infty} e(t) = 0$.

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

όπου, $K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$ με $k_{ij} = k(t, e)$



Εικόνα 16 Δημιουργία ομαλής διαδρομής ενός ρομποτικού οχήματος [18].

Η κινηματική ενός ρομποτικού οχήματος διαφορικής οδήγησης περιγράφεται στο απόλυτο σύστημα αναφοράς $\{X_I, Y_I, \theta\}$ από τη σχέση:

$${}^I \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

όπου, \dot{x} και \dot{y} είναι οι γραμμικές ταχύτητες προς την κατεύθυνση των X_I και Y_I του απόλυτου συστήματος αναφοράς. Οπότε, έστω ότι η μεταβλητή α υποδηλώνει τη γωνία ανάμεσα στον άξονα X_R του σχετικού συστήματος αναφοράς και το διάνυσμα \hat{x} , το οποίο συνδέει το κέντρο των τροχών με την τελική θέση-στόχος του ρομποτικού οχήματος. Εάν η μεταβλητή $\alpha \in I_1$, ανήκει σε διάστημα τέτοιο ώστε $I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, τότε λαμβάνεται υπόψη η μετατροπή των συντεταγμένων σε πολικές συντεταγμένες χρησιμοποιώντας τις σχέσεις:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\theta + \text{atan2}(\Delta y, \Delta x)$$

$$\beta = -\theta - \alpha$$

οι οποίες δημιουργούν ένα καινούριο σύστημα, με καινούριες πολικές συντεταγμένες και χρησιμοποιείται ο πίνακας:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\frac{\cos\alpha}{\rho} & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Όπου, ρ ορίζεται η απόσταση ανάμεσα στο κέντρο των τροχών του ρομποτικού οχήματος και την τελική θέση και θ ορίζεται η γωνία ανάμεσα στον άξονα X_R του συστήματος αναφοράς του ρομποτικού οχήματος και στον άξονα X_I σε συνδυασμό με την τελική θέση. Αντίθετα, εάν η μεταβλητή $\alpha \in I_2$, ανήκει σε διάστημα τέτοιο ώστε $I_2 = \left(-\pi, -\frac{\pi}{2}\right] \cup \left[\frac{\pi}{2}, \pi\right)$, τότε θέτοντας $v = -v$, αποκτάται ένα σύστημα της μορφής:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \frac{\cos\alpha}{\rho} & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Έλεγχος σταθερότητας

Λαμβάνοντας υπόψη τα σήματα ελέγχου v και ω με τη μορφή $v = K_\rho \cdot \rho$ και $\omega = K_\alpha \cdot \alpha + K_\beta \cdot \beta$, σχεδιάζεται η πορεία που θα ακολουθήσει το ρομπότ από την τρέχουσα θέση του ($\rho_0, \alpha_0, \beta_0$) προς την τελική του θέση και προκύπτει ένα σύστημα κλειστού βρόχου .

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -K_\rho \cdot \rho \cdot \cos\alpha \\ K_\rho \cdot \sin\alpha - K_\alpha \cdot \alpha - K_\beta \cdot \beta \\ -K_\rho \cdot \sin\alpha \end{bmatrix}$$

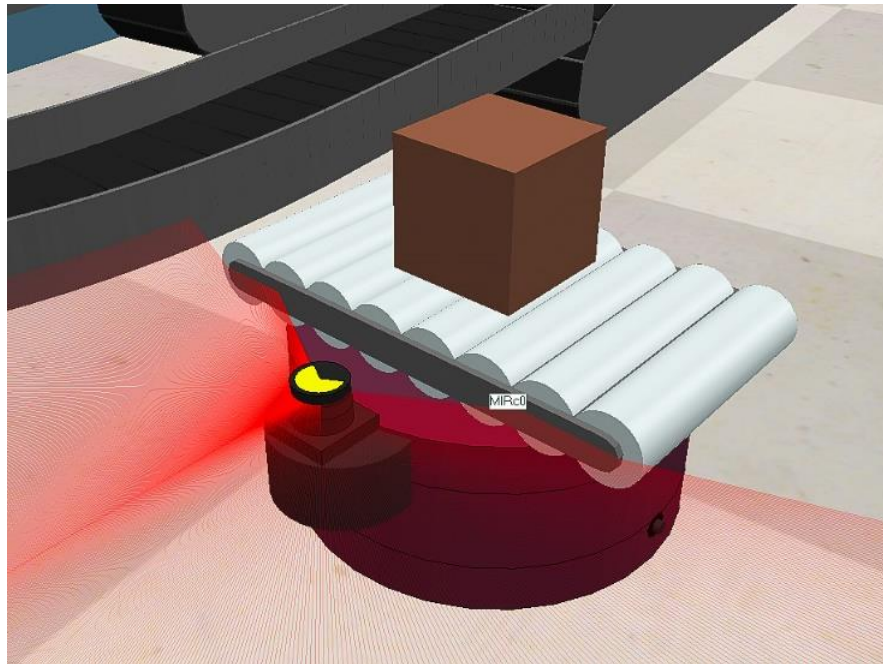
Ακόμα, το παραπάνω σύστημα ελέγχου κλειστού βρόχου παρουσιάζει τοπική ευστάθεια εάν $K_\rho > 0$, $K_\beta < 0$ και $K_\alpha - K_\rho > 0$. Το οποίο αποδεικνύεται όταν για τη θέση ισορροπίας, όπου $\cos x = 1$ και $\sin x = x$ η συνάρτηση παίρνει τη μορφή:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -K_\rho & 0 & 0 \\ 0 & -(K_\alpha - K_\rho) & -K_\beta \\ 0 & -K_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} = A \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}$$

όπου, η συνάρτηση είναι σταθερή εάν οι ιδιοτιμές του πίνακα A έχουν αρνητικό πραγματικό μέρος. Συνεπώς, το χαρακτηριστικό πολυώνυμο που προκύπτει από τη λύση του πίνακα A είναι $(\lambda + K_\rho)(\lambda^2 + \lambda(K_\alpha - K_\rho) - K_\rho \cdot K_\beta)$ και όλες οι ρίζες έχουν αρνητικά πραγματικά μέρη εάν $K_\rho > 0$, $K_\beta < 0$ και $K_\alpha - K_\rho > 0$.

Κεφάλαιο 2 – Αυτόνομη Πλοήγηση

Στο παρόν κεφάλαιο παρουσιάζονται οι μέθοδοι εντοπισμού θέσης και οι τεχνικές για την κατασκευή μονοπατιών και την αποφυγή εμποδίων, που χρησιμοποιούνται στο ρομποτικού συστήμα του MIRc. Το ρομποτικό όχημα MIRc (Εικόνα 17) καλείται να πλοηγηθεί μέσα σε ένα βιομηχανικό περιβάλλον με σκοπό την διαμετακόμιση προϊόντων προς, μέσα από και από παραγωγικές διαδικασίες που εκτελούνται στις γραμμές παραγωγής ενός εργοστασίου. Σήμερα κατασκευάζονται βιομηχανικά ρομποτικά οχήματα που κινούνται σε έναν χώρο εργασίας χρησιμοποιώντας προγραμματιζόμενους μικροελεγκτές, οι οποίοι προσφέρουν έναν μεγαλύτερο βαθμό ευφυίας, καθώς μπορούν να αποθηκεύσουν πληροφορίες διαδρομών, να λάβουν αποφάσεις και να συμμετέχουν στη ρύθμιση της κυκλοφορίας του συνόλου των αυτόνομων ρομποτικών οχημάτων που κινούνται στο βιομηχανικό περιβάλλον. Επομένως, με τη χρήση ασύρματων συστημάτων οδηγίων, τα οποία χρησιμοποιούν αισθητήρες laser ή αδρανειακά συστήματα πλοήγησης, τα ρομπότ μπορούν να κινούνται αυτόνομα χωρίς να απαιτείται η ύπαρξη μιας πορείας (μαύρη γραμμή) στο δάπεδο.



Εικόνα 17 Ρομποτικό όχημα MIRc κατά τη διαδικασία πλοήγησής του, στο πρόγραμμα προσομοίωσης CoppeliaSim.

Η αυτόνομη πλοήγηση είναι ένα πλεονέκτημα για τα ρομποτικά οχήματα καθώς εξυπηρετεί στην ανεξαρτητοποίηση του οχήματος και στον περιορισμό της ανάγκης για ανθρώπινη παρέμβαση, το οποίο φυσικά προϋποθέτει την επίλυση πολλών ζητούμενων και προβλημάτων, όπως είναι για παράδειγμα η διαδικασία κατασκευής μονοπατιών. Το συγκεκριμένο πρόβλημα που σχετίζεται με τη δημιουργία μιας πορείας πλοήγησης, αφορά την εύρεση της βέλτιστης σειράς ενεργειών που έχει ως σκοπό το ρομποτικό όχημα να φτάσει από την τρέχουσα θέση του σε μια επιθυμητή τελική θέση. Μια σειρά ενεργειών έρχεται στη μορφή μιας διαδρομής ή αλλιώς μονοπατιού και χρησιμοποιείται για να κατευθύνει το ρομπότ προς την τελική του διάταξη. Παρόλο που μπορούν να σχηματιστούν πολλές πιθανές διαδρομές μέσα στον χώρο εργασίας του

ρομποτικού οχήματος, με τη χρήση αλγορίθμων κατασκευής μονοπατιών επιτυγχάνεται η εύρεση της καλύτερης διαδρομής ή μιας αποδεκτής προσέγγισης προς αυτή που θα φέρει το βέλτιστο δυνατό αποτέλεσμα. Επομένως, για να επιτευχθεί η αυτόνομη πλοήγηση (*navigation*) ενός ρομπότ είναι απαραίτητη η επιτυχής λειτουργία τεσσάρων κατηγοριών. Η πρώτη κατηγορία αποτελεί την αντίληψη (*perception*) του ρομποτικού οχήματος, όπου το ρομπότ καλείται να ερμηνεύσει τους αισθητήρες που βρίσκονται στο σύστημά του ώστε να εξαχθούν χρήσιμα δεδομένα. Η δεύτερη κατηγορία αποτελεί τον εντοπισμό θέσης (*localization*) του ρομποτικού οχήματος, όπου το ρομπότ πρέπει να προσδιορίσει με ακρίβεια τη θέση του μέσα στον χώρο. Η τρίτη κατηγορία αφορά τη γνωστική λειτουργία (*cognition*) του ρομποτικού συστήματος κατά την οποία ένα ρομπότ αποκτά την ικανότητα να αποφασίζει με ποιον τρόπο πρέπει να ενεργήσει προκειμένου να διεκπεραιώσει τον στόχο του. Τέλος, η τέταρτη κατηγορία αποτελεί τον έλεγχο της κίνησης του ρομπότ (*motion control*), όπου το ρομποτικό σύστημα πρέπει να προσαρμόσει τη λειτουργία των κινητήρων ώστε να προκληθούν οι κατάλληλες κινήσεις στους τροχούς και να ακολουθήσει την επιθυμητή διαδρομή.

2.1 Σύστημα Πλοήγησης

Όπως προαναφέρθηκε, στην παρούσα εργασία ο τύπος του ρομποτικού συστήματος που μελετάται χρησιμοποιείται για τη διαμετακόμιση προϊόντων που έχουν μεγάλο βάρος ή μικρό μέγεθος ή η εταιρεία θέλει να εστιάσει στην επίτευξη υψηλότερης παραγωγικότητας και ακρίβειας. Κατά τη διαδικασία της διαμετακόμισης ένα κινητό βιομηχανικό ρομπότ καλείται να δημιουργήσει μια διαδρομή από ένα σημείο Α προς ένα σημείο Β. Εάν ένα ρομποτικό όχημα κατά την πλοήγησή του είναι ικανό να αντιληφθεί τον χώρο στον οποίο κινείται χωρίς ανθρώπινη παρέμβαση και ταυτόχρονα να αποφύγει εμπόδια, μπορεί να θεωρηθεί αυτόνομο [22]. Μια διαδρομή μπορεί να δημιουργηθεί αυτόνομα μέσα από τους αισθητήρες, τις κάμερες και τους χάρτες ενός ρομποτικού συστήματος, επιτρέποντας τη μεταφορά προϊόντων χωρίς να απαιτούνται άλλοι μέθοδοι για τη μετακίνηση των οχημάτων, όπως συστήματα τηλεκατεύθυνσης ή μέσω ιχνηλάτησης (*Autonomous Ground Vehicles – AGV*).

Η δημιουργία ενός συστήματος πλοήγησης, για ένα αυτόνομο ρομποτικό όχημα, απαιτεί την ύπαρξη ορισμένων συντελεστών, όπως την πλοήγηση σύμφωνα με μια σειρά ενεργειών, τον προγραμματισμό και τη δρομολόγηση, τη διαχείριση κυκλοφορίας, τη μεταφορά φορτίου και τη διαχείριση του συστήματος. Η πλοήγηση σύμφωνα με μια σειρά ενεργειών επιτρέπει στο ρομπότ να ακολουθήσει μια διαδρομή. Ο προγραμματισμός και η δρομολόγηση δίνουν την ικανότητα στο ρομποτικό όχημα να λαμβάνει αποφάσεις κατά τη πλοήγησή του με σκοπό να επιλέξει τη βέλτιστη διαδρομή προς έναν συγκεκριμένο στόχο [23]. Ο σκοπός του προγραμματισμού είναι να παρακολουθήσει την εξέλιξη της πορείας μιας ομάδας ευφυών πρακτόρων, στην οποία έχει ανατεθεί η διεκπεραίωση του στόχου κάτω από συγκεκριμένους περιορισμούς, όπως χρονικά όρια, προτεραιότητες, κλπ. Έπειτα, κατά τη δρομολόγηση γίνεται η εύρεση της κατάλληλης πορείας, σύμφωνα με τις προϋποθέσεις που έχουν τεθεί, όπως αναζήτηση της διαδρομής με τη μικρότερη απόσταση, της διαδρομής με τον συντομότερο χρόνο διεκπεραίωσης ή της διαδρομής με το ελάχιστο κόστος. Κατά τη διαδικασία επιλογής της καταλληλότερης διαδρομής, αρχικά, εντοπίζεται αν υπάρχει μονοπάτι που μπορεί να οδηγήσει το ρομπότ από την αρχική προς την τελική του θέση και στη συνέχεια, εάν στον χώρο εργασίας υπάρχουν περισσότερα από ένα ρομποτικά οχήματα,

επιλέγεται η διαδρομή στην οποία δεν υπάρχει κυκλοφοριακή συμφόρηση μεταξύ ρομποτικών οχημάτων, δεν προβλέπεται σύγκρουση με εμπόδιο και δεν προκύπτει αδιέξοδο. Η διαχείριση της κυκλοφορίας είναι ένα σύστημα ή μια ικανότητα του οχήματος να αποφεύγει συγκρούσεις με άλλα ρομποτικά οχήματα [24]. Η μεταφορά φορτίου αφορά τη διαδικασία παραλαβής και παράδοσης ενός φορτίου που πραγματοποιεί το αυτόνομο ρομποτικό όχημα και μπορεί να είναι απλή ή ενταγμένη σε άλλα υποσυστήματα, όπως για παράδειγμα η τοποθέτηση ενός ρομποτικού βραχίονα στο πάνω μέρος του οχήματος (Εικόνα 18) για την τοποθέτηση φορτίων από μια γραμμή παραγωγής πάνω στο ρομποτικό όχημα και αντιστρόφως. Η διαχείριση του συστήματος αποτελεί τη μέθοδο ελέγχου του ρομποτικού συστήματος και χρησιμοποιείται για να δίνει εντολές στο σύστημα.



Εικόνα 18 Παράδειγμα τοποθέτησης ρομποτικού βραχίονα πάνω σε κινητό βιομηχανικό ρομπότ, Enabled Robotics - ER-FLEX, της εταιρείας MiR [35].

2.1.1 Λειτουργικό πλαίσιο

Το λειτουργικό πλαίσιο της παρούσας εργασίας βασίζεται στην προσέγγιση της κατασκευής ενός ευέλικτου αυτόνομου ρομποτικού οχήματος που είναι ικανό να κινηθεί μεταξύ διαφορετικών θέσεων στον χώρο εργασίας, λαμβάνοντας υπόψη πιθανές αλλαγές στο περιβάλλον. Για να μπορεί το ρομποτικό σύστημα να ικανοποιήσει την παραπάνω προσέγγιση εφαρμόζεται ένα σύνολο τεχνικών, που χωρίζονται σε τμήματα με βάση τις λειτουργικές ανάγκες του προβλήματος που το ρομπότ καλείται να επιλύσει. Η συγκεκριμένη προσέγγιση διευκολύνει την εύρεση του σημείου σύνδεσης μεταξύ των διαφορετικών λειτουργιών και μπορεί να χρησιμοποιηθεί για την ανάπτυξη οποιουδήποτε ρομποτικού συστήματος που χρησιμοποιεί αισθητήρες απόστασης laser και οδομετρίας. Τα τμήματα οργανώνονται σε τρία επίπεδα, το χαμηλό, το μεσαίο και το υψηλό. Το χαμηλό επίπεδο προσφέρει τη σύνδεση με το λειτουργικό σύστημα του ρομπότ. Το μεσαίο επίπεδο αποτελείται από τα τμήματα της αντίληψης και του ελέγχου, τα οποία εμφανίζουν, αντίστοιχα, μια συνεχή παρουσίαση της κατάστασης του περιβάλλοντος και παρέχουν έλεγχο σχετικά με αντιδράσεις που μπορεί να έχει το ρομπότ σε απρόσμενα γεγονότα. Το υψηλό επίπεδο

αποτελείται από το τμήμα πλοήγησης, που ασχολείται με την παρουσίαση των αντικειμένων στον χώρο εργασίας και από το τμήμα προγραμματισμού, που ασχολείται με τη λήψη αποφάσεων που βασίζονται σε εξωτερικές πληροφορίες. Το τμήμα της αντίληψης αποτελείται από δύο διαδικασίες, το σύστημα δημιουργίας μοντέλων, για σχετικά εμπόδια μέσα στο περιβάλλον και τον εντοπισμό θέσης μέσω φίλτρων. Η πρώτη διαδικασία υπολογίζει και συνδέει τα σχετικά μοντέλα με τα εμπόδια που εντοπίζονται από τον αισθητήρα απόστασης laser και η δεύτερη δημιουργεί εκτιμήσεις μέσω των φίλτρων εντοπισμού θέσης, χρησιμοποιώντας τα δεδομένα που αντλούνται από τους αισθητήρες οδομετρίας. Το τμήμα ελέγχου σχετίζεται με τις αποφάσεις που λαμβάνει το ρομποτικό σύστημα σχετικά με τις ενέργειες που πρέπει να εκτελεστούν ανάλογα με το ζήτημα που προκύπτει, όπου τρία κλασικά ζητήματα αποτελούν η αποφυγή εμποδίων, η παρακολούθηση διεκπεραίωσης ενός μονοπατιού και η πρόσδεση σε μια βάση, για παράδειγμα μια θέση παραλαβής φορτίου ή μια βάση υποδοχής για τη φόρτιση της μπαταρίας του ρομποτικού οχήματος. Το τμήμα πλοήγησης αποτελείται από δύο διαδικασίες, την κατασκευή χάρτη και την κατασκευή μονοπατιού. Στην πρώτη διαδικασία τοποθετείται η τρέχουσα θέση του ρομποτικού οχήματος και η αντίληψή του σχετικά με τον χώρο σε έναν ασαφή χάρτη. Έπειτα, χρησιμοποιώντας τον αλγόριθμο αναζήτησης μονοπατιού, που στην παρούσα διπλωματική εργασία αποτελεί τον αλγόριθμο Dijkstra, υπολογίζεται η βέλτιστη διαδρομή προς την τελική θέση. Το τμήμα προγραμματισμού εφαρμόζει μια διαδικασία κατά την οποία λαμβάνει ένα ζητούμενο από το σύστημα διαχείρισης και δημιουργεί ένα πρόγραμμα που λαμβάνει υπόψη την τρέχουσα θέση του ρομποτικού οχήματος και έναν χάρτη του περιβάλλοντος. Το πρόγραμμα αποθηκεύεται σε έναν πεπερασμένο μηχανισμό δημιουργίας ενεργειών, όπου κάθε ενέργεια αποτελεί μέρος μιας εργασίας.

Χώρος καταστάσεων

Η κατασκευή μιας διαδρομής λαμβάνει υπόψη τη συγκρότηση του ρομποτικού οχήματος, δηλαδή το σχήμα του και τις διαστάσεις του και τον χώρο εργασίας του. Για την παρουσίαση των παραπάνω χρησιμοποιείται ένας χώρος καταστάσεων C (*configuration space*), μέσα στον οποίο η πραγματική μορφή που έχει το ρομπότ σε καρτεσιανές συντεταγμένες απλοποιείται και αναπαρίσταται συνήθως με τη μορφή μιας κουκίδας μέσα στον χάρτη [19]. Αντίστοιχα, το ίδιο συμβαίνει και με το χώρο εργασίας που αποκτά μια απλούστερη μορφή στον χώρο καταστάσεων.

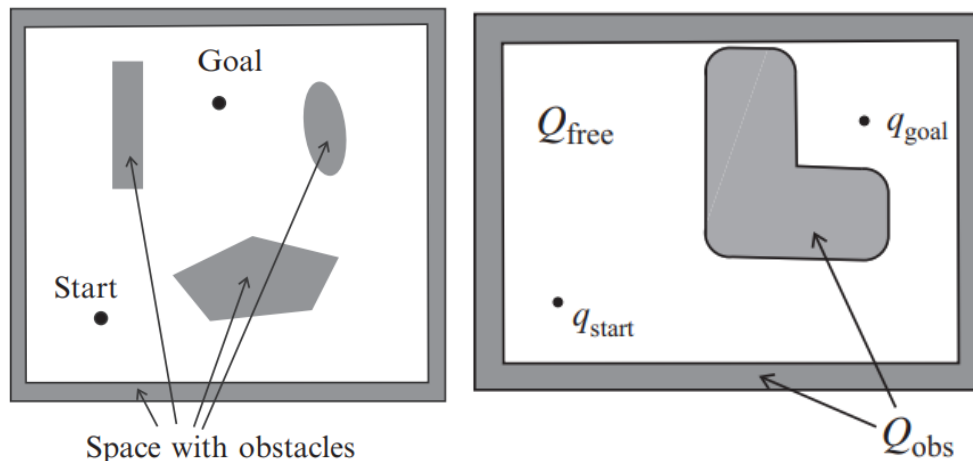
Ο χώρος εργασίας αποτελεί τον χώρο στον οποίο το ρομποτικό όχημα μπορεί να κινηθεί και περιλαμβάνει τον ελεύθερο χώρο (*free space*) και τον χώρο που είναι κατειλημμένος από εμπόδια [11]. Τα εμπόδια μπορούν να είναι στατικά ή δυναμικά, που σημαίνει πως εάν σε έναν χώρο εργασίας υπάρχουν κινητά εμπόδια, δημιουργείται ένα δυναμικό περιβάλλον (*dynamic environment*), ενώ εάν σε αυτόν τον χώρο δεν συμβαίνουν αλλαγές με την πάροδο του χρόνου, δημιουργείται ένα στατικό περιβάλλον (*static environment*). Στον ελεύθερο χώρο ορίζονται οι διατάξεις των ρομπότ, που αποτελούν το σύνολο των παραμέτρων που ορίζουν τη θέση και τον προσανατολισμό τους στον χώρο ή τις γωνίες των συνδέσμων τους. Οπότε, η αρχική και η επιθυμητή τελική θέση του ρομποτικού οχήματος εντοπίζονται στον ελεύθερο χώρο, όπως φαίνεται στην Εικόνα 19α. Επίσης, ο αριθμός των παραμέτρων στη διάταξη του ρομπότ ισοδυναμεί με τους βαθμούς ελευθερίας του ρομποτικού οχήματος.

Η διάταξη μιας θέσης σε έναν χώρο χωρίζεται σε n κατευθύνσεις με τον πίνακα που περιγράφει τη θέση να ορίζεται ως $q = [q_1, \dots, q_n]^T$. Η θέση q αποτελεί ένα σημείο σε στον χώρο με n

κατευθύνσεις που ονομάζεται χώρος διάταξης Q και παρουσιάζει όλες τις πιθανές διατάξεις του ρομποτικού συστήματος με βάση το κινηματικό του μοντέλο. Το μέρος του χώρου διάταξης που περιλαμβάνει εμπόδια O_i , ορίζεται ως $Q_{obst} = \cup_i O_i$. Επομένως, το ελεύθερο μέρος του χώρου χωρίς εμπόδια ορίζεται από τη σχέση:

$$Q_{free} = Q - Q_{obst}$$

Οπότε στον συντελεστή Q_{free} περιλαμβάνεται ο χώρος στον οποίο το ρομποτικό σύστημα μπορεί να προγραμματίσει τις κινήσεις του από τη θέση έναρξης στην θέση άφιξης, όπως φαίνεται στην Εικόνα 19β.



Εικόνα 19 α) Χώρος εργασίας του ρομποτικού οχήματος [11], β) Χώρος διάταξης του ρομποτικού συστήματος [11].

Πιθανολογικός εντοπισμός θέσης

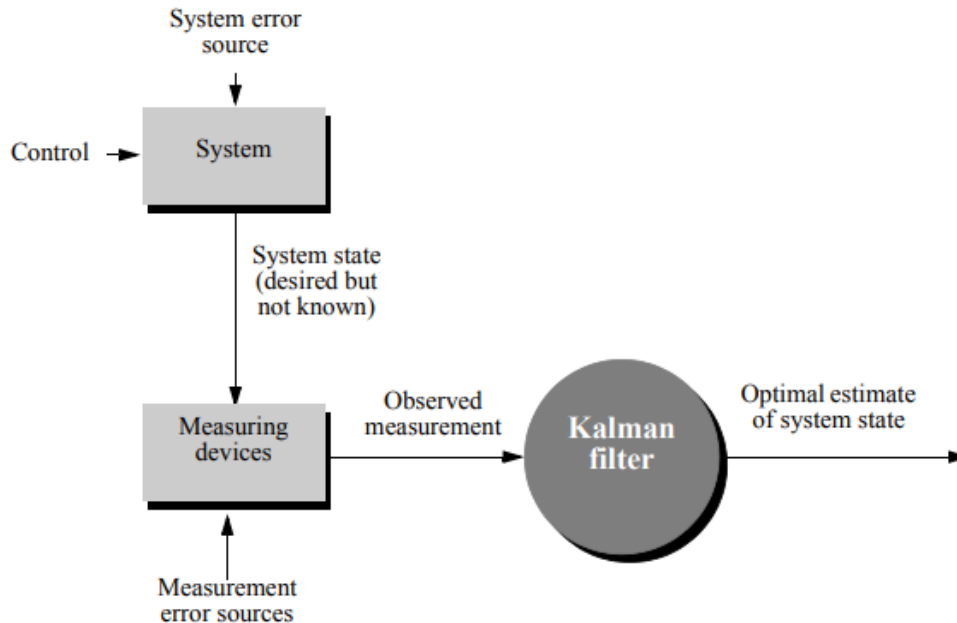
Όταν ένα ρομποτικό όχημα πλοηγείται σε ένα περιβάλλον με μερική ή καμία πληροφορία σχετικά με τη θέση των εμποδίων, πρέπει να έχει αισθητήρες που θα παρέχουν στο σύστημα επαρκείς πληροφορίες για να καλύψουν τη συγκεκριμένη έλλειψη δεδομένων. Κάθε ρομποτικό σύστημα, βέβαια, παρουσιάζει σφάλματα και αβεβαιότητες καθώς επηρεάζονται σε κάποιον βαθμό από ορισμένους παράγοντες [19], όπως:

- Το περιβάλλον μέσα στο οποίο κινείται το ρομποτικό όχημα μπορεί να παρουσιάσει υψηλή αστάθεια και αλλαγές μέσα στον χρόνο.
- Οι αισθητήρες έχουν κάποια όρια στις δυνατότητες αντίληψης και οι μετρήσεις που λαμβάνουν συγχέονται από περιβαλλοντικούς θορύβους και από σφάλματα του ρομποτικού συστήματος.
- Οι τελεστές των κινητήρων συνήθως δεν παρέχουν ακρίβεια και συγχέονται από εξωτερικούς θορύβους.
- Οποιοδήποτε μοντέλο παρέχει μη ολοκληρωμένη περιγραφή διαδικασιών και γεγονότων που εισάγουν σφάλματα στη διαθέσιμη εικόνα του ρομπότ και του περιβάλλοντός του.
- Οι ενσωματωμένοι μικροϋπολογιστές που βρίσκονται πάνω στο ρομποτικό σύστημα έχουν περιορισμένες δυνατότητες και οι αλγόριθμοι χρησιμοποιούνται κατά προσέγγιση με αποτέλεσμα να υπάρχει ανακρίβεια στους υπολογισμούς.

Η αντιμετώπιση αυτών των ανακριβειών μπορεί να γίνει με διάφορους τρόπους. Ο τρόπος που επιλέχθηκε να παρουσιαστεί αποτελεί εκείνον των πιθανολογικών μεθόδων δημιουργίας αποφάσεων, όπως είναι το φίλτρο Markov. Η συγκεκριμένη προσέγγιση εφαρμόζεται για την πλοήγηση των ρομποτικών οχημάτων σε ένα στοχαστικό περιβάλλον, λαμβάνοντας υπόψη κάθε αβεβαιότητα, η οποία μπορεί να διαφέρει ανάλογα με τον τύπο του ρομπότ και τις ικανότητές του. Ο προσδιορισμός της διαδικασίας δημιουργίας αποφάσεων προσδιορίζεται ως εξής: Έστω ένα πεπερασμένο ή μετρήσιμο σύνολο πιθανών διατάξεων ενός ρομποτικού ευφυούς πράκτορα (*agent*) $S = \{s_1, s_2, s_3, \dots\}$ και ένα σύνολο από πιθανές του ενέργειες $A = \{a_1, a_2, a_3, \dots\}$. Επιπρόσθετα, έστω ένα σύνολο από πιθανά αποτελέσματα μετρήσεων του περιβάλλοντος, προερχόμενα από τον ευφυή πράκτορα, όπως είναι για παράδειγμα ο αριθμός των ελέγχων που γίνονται στον χώρο εργασίας για εμπόδια σε απόσταση δύο μέτρων, $M = \{m_1, m_2, m_3, \dots\}$. Από τα παραπάνω προκύπτει ένα γενικό πλαίσιο διαδικασιών, που εκτελούν οι ευφυείς πράκτορες, σε διακριτό χρόνο $t = 0, 1, 2, \dots$ και αποτελείται από πέντε βασικές εντολές. Η πρώτη εντολή αφορά την ανάθεση διακριτού χρόνου t σε $t = 0$ και η έναρξη της αρχικής διάταξης $s(0) = S$. Έπειτα η δεύτερη εντολή αφορά τις διαδικασίες που εκτελούνται όταν το ρομποτικό όχημα βρίσκεται στη διάταξη $s(t) = S$. Όσο το ρομπότ βρίσκεται σε αυτή τη διάταξη, αρχικά, παρατηρεί το περιβάλλον και λαμβάνει τα αποτελέσματα των μετρήσεων $m(t) = M$ και στη συνέχεια, επιλέγει μια ενέργεια $a(t) = A$, με βάση τη διάταξη $s(t)$ και λαμβάνει το αποτέλεσμα της μέτρησης $m(t) = M$. Στην τρίτη εντολή εφαρμόζεται η ενέργεια $a(t)$ και λαμβάνεται μια καινούρια διάταξη $s(t + 1) = S$. Στην τέταρτη εντολή γίνεται ανάθεση μιας καινούριας τιμής διακριτού χρόνου t σε $t = t + 1$. Τέλος, στην πέμπτη εντολή το πρόγραμμα επιστρέφει στην εντολή 2. Φυσικά αν για τη χρονική στιγμή t η επιλεγμένη ενέργεια $a(t)$ είναι ο τερματισμός τότε η διάταξη $s(t + 1)$ σταματά την εκτέλεση του προγράμματος και ο ευφυής πράκτορας ολοκληρώνει την αποστολή που το ανατέθηκε.

Η μέθοδος πλοήγησης βασίζεται στη χρήση γνωστού χάρτη που υπάρχει αποθηκευμένος στη μνήμη του ρομποτικού συστήματος, με αποτέλεσμα κατά τη διαδικασία του εντοπισμού θέσης, το ρομπότ να προσπαθεί να προσδιορίσει τη θέση του συλλέγοντας δεδομένα από τους αισθητήρες. Στην παρούσα εργασία οι διατάξεις της αρχικής και της τελικής θέσης είναι γνωστές πριν την έναρξη της κίνησης του ρομποτικού οχήματος, επομένως, αρχικά κατασκευάζεται ένα μονοπάτι αξιοποιώντας τα διαθέσιμα δεδομένα και έπειτα με βάση τις καινούριες πληροφορίες που προέρχονται από τους αισθητήρες, το μονοπάτι τροποποιείται ώστε να αποφευχθούν πιθανές συγκρούσεις. Οπότε, για να δημιουργηθεί μια επόμενη θέση είναι απαραίτητο να χρησιμοποιηθεί η μέθοδος πιθανοτήτων, καθώς με αυτόν τον τρόπο οι μετρήσεις των αισθητήρων συνδυάζονται για να υπολογιστούν τα στατικά και δυναμικά χαρακτηριστικά του χώρου. Προκειμένου να εξασφαλιστεί το μικρότερο δυνατό σφάλμα το ρομποτικό όχημα πρέπει να προσδιορίζει τη θέση του ανάλογα με τον χάρτη του χώρου που διαθέτει στο σύστημά του [18]. Για τον εντοπισμό της θέσης του ρομποτικού οχήματος χρησιμοποιούνται εξωδεκτικοί αισθητήρες, όπως αισθητήρες υπερήχων, λέιζερ ή κάμερες, δημιουργώντας μια αντίληψη του χώρου. Επομένως, οι πληροφορίες των αισθητήρων οδομετρίας και των εξωδεκτικών αισθητήρων μπορούν να συνδυαστούν για τον εντοπισμό της θέσης πετυχαίνοντας όσο το δυνατόν λιγότερα σφάλματα συγκριτικά με τον χάρτη. Ύστερα η διαδικασία της ενημέρωσης που βασίζεται στους ιδιοδεκτικούς και στους εξωδεκτικούς αισθητήρες χωρίζεται σε δύο στάδια. Πρώτο στάδιο αποτελεί η ενημέρωση της πρόβλεψης (*prediction update*), όπου το ρομπότ χρησιμοποιεί ιδιοδεκτικούς αισθητήρες για να προσδιορίσει

τις ρυθμίσεις και παραμέτρους του συστήματός του. Σε αυτό το στάδιο το ποσοστό αβεβαιότητας συγκριτικά με τη θέση του ρομποτικού οχήματος αυξάνεται εξαιτίας του οδομετρικού σφάλματος ανά χρονική στιγμή. Το δεύτερο στάδιο αποτελεί η ενημέρωση αντίληψης (*perception update*) ή αλλιώς στάδιο διόρθωσης, όπου το ρομπότ χρησιμοποιεί τους εξωδεκτικούς αισθητήρες για να διορθώσει τη θέση που προσδιορίστηκε κατά το στάδιο της πρόβλεψης. Σε αυτό το στάδιο το ποσοστό αβεβαιότητας μειώνεται και για τη δημιουργία μιας πιο αξιόπιστης κατάστασης γίνεται χρήση πιθανοτικών μεθόδων, όπως ο εντοπισμός θέσης Markov (Markov localization) και ο εντοπισμός θέσης με φίλτρα Kalman (Εικόνα 20).



Εικόνα 20 Εφαρμογή φίλτρου Kalman [18]

Ο εντοπισμός θέσης Markov επιτρέπει το ρομποτικό όχημα να ξεκινήσει από οποιαδήποτε άγνωστη θέση και να ανακάμψει σε συγκεκριμένες καταστάσεις, καθώς το ρομπότ έχει την ικανότητα να ανιχνεύσει πολλαπλές και διαφορετικές πιθανές θέσεις. Συγκεκριμένα, υπολογίζει τη πιθανότητα σε καθεμία από τις θέσεις, με τη θέση που έχει τη μεγαλύτερη πιθανότητα να θεωρείται εκείνη όπου βρίσκεται το ρομποτικό όχημα [25]. Η πιθανότητα αποτελεί τη συνάρτηση:

$$bel(x_t) = f(bel(x_{t-1}), u_t, z_t, m),$$

όπου,

x : αποτελεί το σύνολο των πιθανών θέσεων.

$bel(x_t)$: αποτελεί την πιθανότητα ότι το ρομπότ βρίσκεται στη θέση x τη χρονική στιγμή t .

$bel(x_{t-1})$: αποτελεί τη πιθανότητα ότι το ρομπότ ήταν στη θέση x_{t-1} τη χρονική στιγμή $t-1$.

u_t : αποτελεί το σύνολο διαδικασιών ελέγχου ή τι κινήσεις πρέπει να εκτελέσει το ρομπότ τη χρονική στιγμή t .

z_t : αποτελεί το σύνολο μετρήσεων ή τι παρατηρεί το ρομπότ τη χρονική στιγμή t .

m : αποτελεί τον χάρτη.

Markov Localization Algorithm($bel(x_{t-1}), u_t, z_t, m$)	
1	For all x_t do
2	$bel(x_t) = \int p(x_t u_t, x_{t-1}, m) \cdot bel(x_{t-1}) dx_{t-1}$
3	$bel(x_t) = \eta p(z_t x_t, m) \cdot bel(x_t)$
4	end for
5	return $bel(x_t)$

Αλγόριθμος 1 Markov Localization [48].

Ο εντοπισμός θέσης με φίλτρα Kalman ανιχνεύει το ρομποτικό όχημα από μια γνωστή αρχική θέση και αποτελεί μια πιο ακριβής και αποδοτική μέθοδο. Το φίλτρο Kalman μπορεί να προσδιορίσει τη θέση X μιας διαδικασίας που πραγματοποιείται σε διακριτό χρόνο [26]. Η διαδικασία μπορεί να περιγραφεί από μια γραμμική στοχαστική διαφορική εξίσωση σε μια αναπαράσταση κατάστασης – χώρου:

$$X_k = F_{k-1}X_{k-1} + G_{k-1}U_{k-1} + u_{k-1}$$

όπου, F αποτελεί έναν δυναμικό πίνακα μετατροπής, G αποτελεί το διάνυσμα εισόδου και U αποτελεί το διάνυσμα ελέγχου της εισόδου. Επίσης, οι αισθητήρες κατά τη διάρκεια της διαδικασίας παρέχουν μετρήσεις $Z_k = H_kX_k + e_k$, όπου H αποτελεί το μοντέλο παρακολούθησης. Οι μεταβλητές u και e συμβολίζουν το θόρυβο και τη μέτρηση του θορύβου, αντίστοιχα.

Το φίλτρο Kalman προσδιορίζει μια γνωστική εκτίμηση χρησιμοποιώντας μια προηγούμενη εκτίμηση και μια σταθμισμένη διαφορά ανάμεσα στην καινούρια μέτρηση και σε αυτήν που έχει προβλεφθεί.

$$\hat{x}_k = \hat{x}_k^- + K(y_k + H\hat{x}_k^-)$$

όπου, το βάρος K επιλέγεται έτσι ώστε η συνδιακύμανση της κατάστασης να είναι ελαχιστοποιημένη.

Kalman filter Algorithm	
1	Initialize:
2	$\hat{X}_{1 0} = E(x_0), P_{1 0} = \text{Cov}(x_0)$
3	Loop:
4	Predict:
5	$\hat{X}_{k k-1} = A_{k-1}\hat{X}_{k-1 k-1} + B_{k-1}U_{k-1}$
6	$P_{k k-1} = A_{k-1}P_{k-1 k-1}A_{k-1}^T + Q_{k-1}$
7	Observe and correct:
8	$K_k = P_{k k-1}H_k^T(H_kP_{k k-1}H_k^T + R_k)^{-1}$
9	$\hat{X}_{k k} = \hat{X}_{k k-1} + K_k(Y_k - H_k\hat{X}_{k k-1})$
10	$P_{k k} = P_{k k-1} - K_kH_kP_{k k-1}$

Αλγόριθμος 2 Φίλτρο Kalman [26].

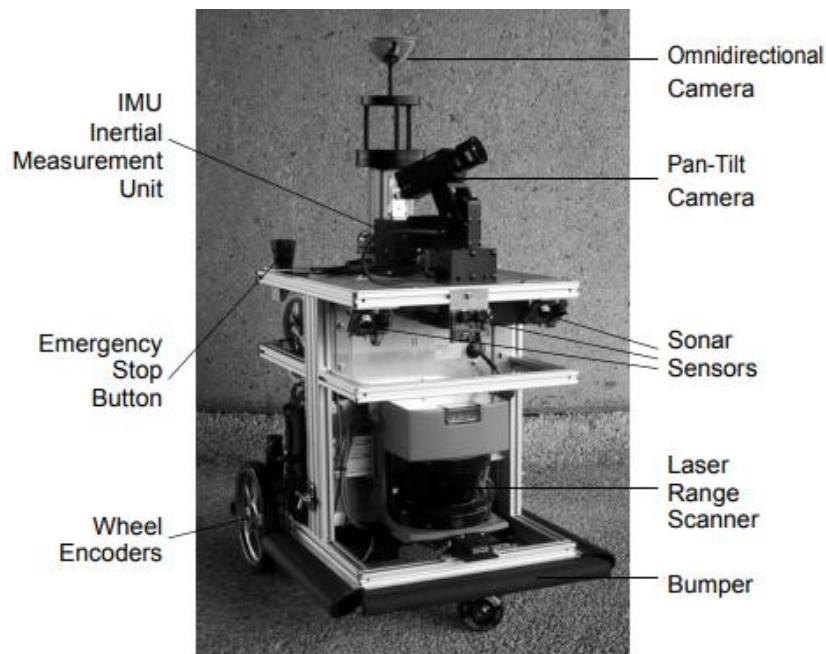
2.1.2 Αντίληψη χώρου

Κατά τη διαδικασία του εντοπισμού θέσης των αυτόνομων ρομποτικών οχημάτων χρησιμοποιούνται οι αισθητήρες του ρομποτικού συστήματος για δύο βασικούς σκοπούς: για τη δημιουργία ενός σχετικού χάρτη από τις καταγεγραμμένες μετρήσεις και για την ενεργοποίηση της διαδικασίας προσδιορισμού της πόζας του ρομπότ μέσα από τις πληροφορίες των αισθητήρων που σχετίζονται με τον χάρτη. Ο όρος αισθητήρας χρησιμοποιείται για την περιγραφή ενός στοιχείου που δημιουργεί ένα σήμα, το οποίο είναι σχετικό με την ποσότητα που καταμετρείται [27]. Ο όρος μορφομετατροπέας (*transducer*) χρησιμοποιείται συχνά στη θέση του όρου «αισθητήρας» και ορίζεται ως ένα στοιχείο που όταν ανιχνεύσει κάποια φυσική μεταβολή, προκαλεί κάποια αλλαγή.

Κάποιοι αισθητήρες συνδυάζονται με το υποσύστημα επεξεργασίας σήματος που υπάρχει στην ίδια συσκευή και, συχνά, πρόκειται για ολοκληρωμένους αισθητήρες (*integrated sensors*), οι οποίοι για να μπορέσουν να χρησιμοποιηθούν απαιτείται περαιτέρω επεξεργασία δεδομένων. Στην περίπτωση όπου ένας αισθητήρας συνδυαστεί με ένα σύστημα επεξεργασίας σήματος και έναν επεξεργαστή (*microprocessor*) στην ίδια συσκευή, προκύπτει ένας ευφυής αισθητήρας. Η διάταξη του ευφυή αισθητήρα είναι ικανή να πραγματοποιήσει λειτουργίες όπως την αντιστάθμιση τυχαίων σφαλμάτων, την προσαρμογή στις περιβαλλοντικές μεταβολές, την πραγματοποίηση ενός αυτόματου υπολογισμού για την ακρίβεια μιας μέτρησης και την αυτοδιάγνωση σφαλμάτων. Επομένως, οι αισθητήρες είναι ένα εργαλείο που καθίσταται υποχρεωτικό σε ένα ρομποτικό σύστημα, καθώς ενεργοποιεί τη διαδικασία συλλογής, την αποκωδικοποίηση και την επεξεργασία πληροφοριών σχετικά με το περιβάλλον του [18]. Οι πληροφορίες αυτές μπορούν να αφορούν το εξωτερικό περιβάλλον του συστήματος, όπως είναι η θερμοκρασία, η κατάσταση φωτισμού ή η απόσταση από ένα εμπόδιο. Διαφορετικά, μπορούν να αφορούν το εσωτερικό περιβάλλον ενός συστήματος, που σχετίζεται με την τροφοδοσία του κινητήρα, την ταχύτητα περιστροφής των τροχών και την τάση της μπαταρίας. Ένα ρομποτικό σύστημα, ανάλογα με το σκοπό της εργασίας του, μπορεί να χρησιμοποιεί περισσότερους από έναν αισθητήρα, όπως φαίνεται στην Εικόνα 21, όπου παρουσιάζεται το ρομπότ BIBA, της εταιρείας BlueBotics SA. Το συγκεκριμένο ρομπότ αποτελείται από συνολικά επτά είδη αισθητήρων: μια αδρανειακή μονάδα πλοήγησης (*Inertial Measurement Unit – IMU*), κωδικοποιητές θέσεις τροχών (*wheel encoders*), μια πανκατευθυντική (*omnidirectional*) κάμερα, μια περιστρεφόμενη (*pan-tilt*) κάμερα, αισθητήρες υπερήχων (*sonar*), λέιζερ απόστασης (*laser range*) και προφυλακτήρες (*bumper*).

Οι αισθητήρες κατατάσσονται σε δύο βασικές λειτουργικές κατηγορίες: στους ιδιοδεκτικούς ή εξωδεκτικούς αισθητήρες και στους ενεργούς ή παθητικούς αισθητήρες. Οι ιδιοδεκτικοί αισθητήρες μετρούν τις εσωτερικές τιμές ενός συστήματος, όπως την ταχύτητα του κινητήρα, το ωφέλιμο φορτίο των τροχών, τις γωνίες ενός ρομποτικού βραχίονα και την τάση της μπαταρίας. Αντίθετα, οι εξωδεκτικοί αισθητήρες αντλούν πληροφορίες από το περιβάλλον του ρομπότ, όπως μετρήσεις αποστάσεων, ένταση φωτεινότητας και ένταση ήχων. Επομένως, οι εξωδεκτικοί αισθητήρες αξιοποιούνται από το ρομπότ ώστε να αποσπάσει στοιχεία και χαρακτηριστικά από το περιβάλλον στο οποίο κινείται. Οι ιδιοδεκτικοί αισθητήρες τοποθετούνται συνήθως σε συνδυασμό με εσωτερικούς τελεστές, καθώς ο συνδυασμός των αποκρίσεών τους έχει τη δυνατότητα να αλλάξει την εσωτερική κατάσταση του συστήματος και την επιστροφή μιας κατάστασης πλησιέστερης στην επιθυμητή [19]. Ένας παθητικός αισθητήρας παίρνει μετρήσεις από το εξωτερικό περιβάλλον ενός ρομπότ [18]. Παραδείγματα παθητικών αισθητήρων αποτελούν τα μικρόφωνα και οι CCD

(*Charged Coupled Device*) ή CMOS (*Complementary Metal-Oxide Semiconductor*) κάμερες. Αντίθετα, οι ενεργοί αισθητήρες εκπέμπουν ενέργεια προς το περιβάλλον και στη συνέχεια μετρούν την αντίδραση που προκύπτει σε αυτό. Επειδή οι ενεργοί αισθητήρες μπορούν να διαχειριστούν περισσότερες ελεγχόμενες αλληλεπιδράσεις με το περιβάλλον, συχνά επιτυγχάνουν καλύτερη απόδοση. Μια αδυναμία που έχει ένας ενεργός αισθητήρας είναι οι παρεμβολές που μπορεί να δεχτεί μεταξύ των σημάτων που προορίζονται για εκείνον και των σημάτων που είναι πέρα από τον έλεγχό του. Αυτό συμβαίνει εξαιτίας των σημάτων που εκπέμπονται από άλλα κοντινά ρομπότ ή παρόμοιων αισθητήρων που βρίσκονται στο ίδιο ρομπότ, επηρεάζοντας έτσι τα αποτελέσματα των μετρήσεων, με αποτέλεσμα να δημιουργούνται σφάλματα. Παραδείγματα ενεργών αισθητήρων είναι οι αισθητήρες υπερήχων και οι αισθητήρες απόστασης laser.

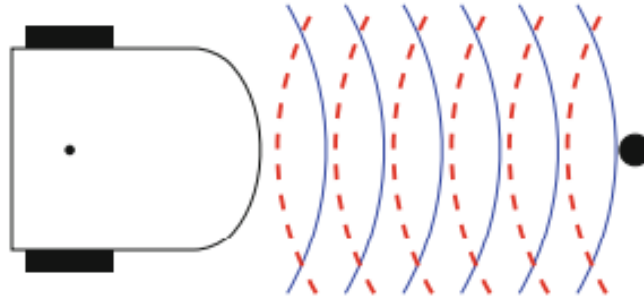


Εικόνα 21 BIBA Robot, BlueBotics SA [18].

Αισθητήρες απόστασης

Οι αισθητήρες απόστασης (*distance sensors*) έχουν την ικανότητα να εντοπίσουν αντικείμενα που βρίσκονται σε κοντινή απόσταση με το ρομπότ. Οι συγκεκριμένοι αισθητήρες είναι συνήθως ενεργοί, καθώς μεταδίδουν ένα σήμα, λαμβάνοντας στη συνέχεια την ανάκλασή του, εάν υπάρχει, από το αντικείμενο, όπως φαίνεται στην Εικόνα 22[6]. Ένας τρόπος να υπολογιστεί η απόσταση είναι να μετρηθεί το χρονικό διάστημα που περνάει από τη στιγμή που στάλθηκε το σήμα μέχρι τη στιγμή που λήφθηκε, χρησιμοποιώντας τη σχέση $s = \frac{1}{2}vt$, όπου s αποτελεί την απόσταση, v η ταχύτητα του σήματος και t το χρονικό διάστημα που πέρασε από την αποστολή μέχρι τη λήψη του σήματος. Για την κατασκευή των συγκεκριμένων αισθητήρων υπάρχουν πολλές τεχνολογίες που μπορούν να χρησιμοποιηθούν, όπως είναι οι υπέρυθρες ακτινοβολίες, οι υπέρηχοι ή το laser [6]. Η χρήση τους στα ρομποτικά οχήματα οφείλεται σε μια σημαντική ικανότητα που έχουν, η οποία αποτελεί τη μέτρηση της απόστασης από το ρομπότ μέχρι το αντικείμενο και τον υπολογισμό της γωνίας ανάμεσα στο μπροστινό μέρος του ρομποτικού οχήματος και το αντικείμενο. Για

τον εντοπισμό και την αποφυγή εμποδίων τα περισσότερα ρομποτικά συστήματα βασίζονται κυρίως στους αισθητήρες απόστασης, όπως επίσης και για διαδικασίες εντοπισμού θέσης ή τη μοντελοποίηση ρομποτικών οχημάτων [18].



Εικόνα 22 Χρήση ενός αισθητήρα απόστασης [6].

Οι αισθητήρες υπέρυθρων είναι απλές συσκευές που χρησιμοποιούν τις ακτίνες για να εντοπίσουν την παρουσία ενός αντικειμένου στον χώρο, το οποίο πετυχαίνουν μετρώντας την ένταση της ανάκλασης των ακτινών. Πολλές φορές οι μετρήσεις περιέχουν αβεβαιότητες, καθώς η ένταση της ανάκλασης συχνά εξαρτάται και από τα χαρακτηριστικά του αντικειμένου, όπως για παράδειγμα ένα μαύρο αντικείμενο ανακλά την ακτίνα με λιγότερη ένταση φωτός σε σύγκριση με ένα άσπρο αντικείμενο που έχει τοποθετηθεί στο ίδιο σημείο. Αντίστοιχα, οι αισθητήρες υπέρυθρων (*ultrasonic sensors*) εκπέμπουν ηχητικά κύματα σε πακέτα και στη συνέχεια μετρούν τον χρόνο που χρειάζεται για να κάνουν αντανάκλαση σε κάποιο αντικείμενο στον χώρο του ρομπότ και έπειτα να επιστρέψει πάλι πίσω στον δέκτη [18]. Επίσης, σε έναν αισθητήρα laser το πλάτος της ακτίνας είναι αρκετά μικρό με αποτέλεσμα να χρειάζονται περισσότεροι από έναν αισθητήρα για να εντοπίσουν αντικείμενα, τα οποία μπορούν να βρίσκονται σε οποιαδήποτε γωνία [6]. Οι συγκεκριμένοι αισθητήρες εκπέμπουν μια προκαθορισμένη εμβέλεια φωτός μέσα στον χώρο που κινείται το ρομποτικό όχημα και στη συνέχεια μετρούν τον χρόνο που χρειάζεται το φως για να φτάσει στο αντικείμενο και να επιστρέψει. Στην παρούσα εργασία ο αισθητήρας που χρησιμοποιείται για τον εντοπισμό εμποδίων και την χαρτογράφηση είναι ο LiDAR (*Light Detection And Ranging*) και παρουσιάζεται στην Εικόνα 23. Ο συγκεκριμένος αισθητήρας χρησιμοποιεί ηλεκτρομαγνητικά κύματα μέσα στα οπτικά και υπέρυθρα μήκη κύματος [28]. Είναι ένας ενεργός αισθητήρας, οπότε στέλνει ένα ηλεκτρομαγνητικό κύμα και λαμβάνει την ανάκλαση του σήματος μόλις επιστρέψει. Στην πιο απλοϊκή τους μορφή οι αισθητήρες LiDAR τοποθετούνται μέσα σε συσκευές μέτρησης αποστάσεων και χρησιμοποιούνται για τη μέτρηση της απόστασης ως ένα σύνολο σημείων. Σε αυτή την περίπτωση οι αισθητήρες έχουν κατεύθυνση προς έναν συγκεκριμένο στόχο, επομένως εφόσον η επεξεργασία γίνεται προς μια κατεύθυνση αποτελούν μονοδιάστατοι ή 1D αισθητήρες. Σε μια δεύτερη περίπτωση, όπου η ακτίνα μετακινείται ή περιστρέφεται, παρέχοντας μια ένδειξη της απόστασης και της γωνίας, τα αποτελέσματα δίνονται σε δύο διαστάσεις. Οπότε οι συγκεκριμένοι αισθητήρες είναι γνωστοί ως 2D σαρωτές laser ή 2D αισθητήρες LiDAR και εντοπίζουν υπολογισμένες τιμές σε διαδοχική σειρά που συνήθως έχουν το ίδιο χρονικό κενό ανάμεσα στις μετρήσεις. Ακόμα, μπορούν να λειτουργήσουν και σε τρεις διαστάσεις εάν είναι τοποθετημένοι σε μια περιστρεφόμενη βάση, το οποίο παρέχει πληροφορίες για την απόσταση και τη θέση κατά μήκος του άξονα X όπως και τη θέση κατά μήκος του άξονα Y και Z.

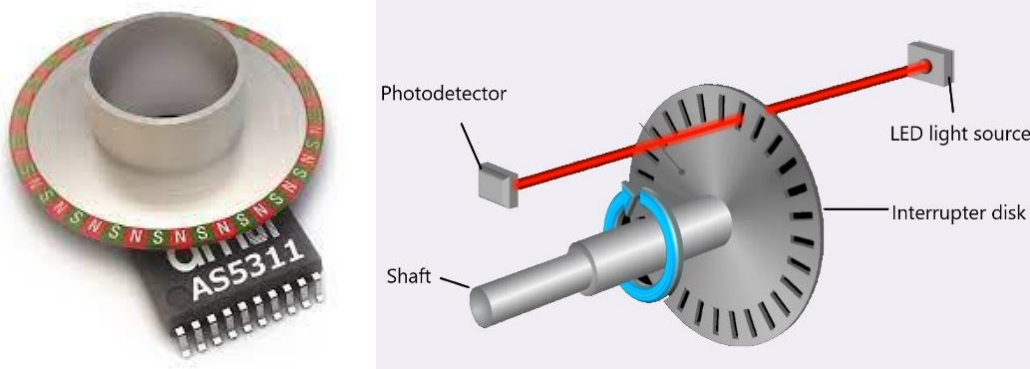
Παρόμοιες πληροφορίες μπορούν να ληφθούν και για διαφορετικές παραμέτρους μέσα στον χώρο εάν τοποθετηθούν πολλαπλά συστήματα αποστολής και λήψης σημάτων σε διαφορετικές οριζόντιες γωνίες καθώς ο αισθητήρας κινείται πάνω στο ρομποτικό όχημα. Ο συγκεκριμένος αισθητήρας είναι γνωστός ως σαρωτής πολλαπλών επιπέδων.



Εικόνα 23 Αισθητήρας LIDAR URG-04LX, Hokuyo [43].

Αισθητήρες τροχών και κινητήρα

Οι αισθητήρες τροχών και κινητήρα αποτελούν συσκευές που χρησιμοποιούνται για τον υπολογισμό της θέσης ή της ταχύτητας των τροχών και ανήκουν στους ιδιοδεκτικούς αισθητήρες [18]. Μια κατηγορία αυτών των αισθητήρων αποτελούν οι κωδικοποιητές (*encoders*), που είναι ηλεκτρομηχανικές συσκευές και μετατρέπουν τη γραμμική ή την γωνιακή κίνηση ενός ρομποτικού οχήματος σε αναλογικό ή σε ψηφιακό σήμα [19]. Υπάρχουν πολλές τεχνικές για την κατασκευή ενός κωδικοποιητή από τις οποίες υπάρχουν δύο που χρησιμοποιούνται συχνότερα, οι μαγνητικοί κωδικοποιητές και οι οπτικοί κωδικοποιητές. Οι μαγνητικοί κωδικοποιητές χρησιμοποιούν το φαινόμενο του Hall και έναν περιστρεφόμενο δίσκο τοποθετημένο στον άξονα του κινητήρα, ο οποίος είναι πλαισιωμένος περιμετρικά με μαγνήτες, όπως φαίνεται στην Εικόνα 24α [21]. Ο οπτικός κωδικοποιητής (*optical encoder*) αποτελείται από έναν δίσκο με σκουρόχρωμες και ανοιχτόχρωμες περιοχές και ζεύγη φωτοδεκτών και φωτοεκπομπών που επεξεργάζονται το μοτίβο του δίσκου κατά την κίνηση του ρομποτικού οχήματος, υπολογίζοντας τη θέση που βρίσκεται ο κινητήρας, όπως φαίνεται στην Εικόνα 24β [29].



Εικόνα 24 α) Μαγνητικός κωδικοποιητής [44], β) Οπτικός κωδικοποιητής [45].

Ένα παράδειγμα αποτελεί ο αισθητήρας οδομετρίας που καταγράφει την απόσταση που διένυσε το ρομποτικό όχημα από μια συγκεκριμένη θέση εκκίνησης [19]. Συγκεκριμένα, η οδομετρία αποτελεί μια θεμελιώδης μέθοδο που χρησιμοποιείται από τα ρομπότ για την πλοήγηση του οχήματος. Θεωρείται πως αποτελεί μια μορφή εύρεσης θέσης, καθώς το ρομποτικό σύστημα πρέπει να υπολογίσει τη θέση του μέσα στο περιβάλλον που κινείται [6]. Η διαδικασία αυτή ονομάζεται *deduced reckoning (dead reckoning)* και χρησιμοποιεί την οδομετρία για να υπολογίσει την κίνηση του ρομποτικού οχήματος σε συνδυασμό τους κωδικοποιητές που εφαρμόζουν προσεγγίσεις των κινηματικών μοντέλων $[t_k, t_{k+1}]$ [20]. Στην περίπτωση που είναι γνωστή η διάταξη του ρομποτικού οχήματος $q_k = [x_k \ y_k \ \theta_k]$ και οι ταχύτητες εισόδου v_k και ω_k σε διακριτό χρόνο t_k , τότε χρησιμοποιώντας τη μέθοδο Euler προκύπτουν οι σχέσεις:

$$x_{k+1} = x_k + v_k \cdot T_s \cdot \cos\theta_k$$

$$y_{k+1} = y_k + v_k \cdot T_s \cdot \sin\theta_k$$

$$\theta_{k+1} = \theta_k + \omega_k \cdot T_s$$

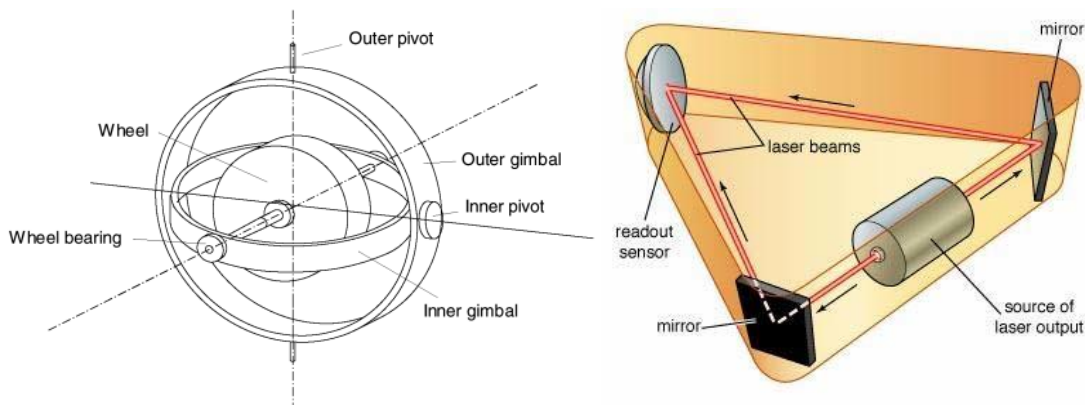
όπου, $v_k \cdot T_s = \Delta s$ και $\omega_k \cdot T_s = \Delta\theta$ και $T_s = t_{k+1} - t_k$. Η αλλαγή της τρέχουσας διάταξης του ρομπότ βασίζεται στα δεδομένα που λαμβάνει το ρομποτικό σύστημα από τον αισθητήρα οδομετρίας, οπότε για να υπολογιστεί η διάταξη για τη χρονική στιγμή t_k θεωρούνται οι μεταβλητές $\Delta\varphi_R$ και $\Delta\varphi_L$, οι οποίες ορίζουν τον αριθμό περιστροφών κάθε τροχού, ξεχωριστά, στο χρονικό διάστημα T_s . Οπότε η γραμμική και γωνιακή μετατόπιση του ρομπότ δίνεται από τις σχέσεις $\Delta s = \frac{r}{2}(\Delta\varphi_R + \Delta\varphi_L)$ και $\Delta\theta = \frac{r}{d}(\Delta\varphi_R - \Delta\varphi_L)$, όπου r είναι η ακτίνα του τροχού και d η αξονική απόσταση ανάμεσα στους τροχούς. Οπότε, η θέση μπορεί να υπολογιστεί ενσωματώνοντας σε μια αρχική γνωστή θέση τις μετατοπίσεις για τη χρονική στιγμή t_k , δίνοντας την παρακάτω διάταξη:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix}$$

Αισθητήρες κατεύθυνσης

Οι αισθητήρες κατεύθυνσης (*heading sensors*) χρησιμοποιούνται για τον υπολογισμό του προσανατολισμού και της κλίσης ενός ρομποτικού οχήματος, όπως επίσης και σε συνδυασμό με κατάλληλες μετρήσεις ταχύτητας για τον προσδιορισμό της θέσης του [18]. Οι συγκεκριμένοι αισθητήρες μπορούν να είναι ιδιοδεκτικοί ή εξωδεκτικοί. Ένα παράδειγμα ιδιοδεκτικού αισθητήρα αποτελεί η πυξίδα (*compass*), η οποία χρησιμοποιείται για τη μέτρηση του απόλυτου προσανατολισμού, με βάση το μαγνητικό πεδίο της γης. Αντίθετα, ένας εξωδεκτικός αισθητήρας μπορεί να είναι το γυροσκόπιο (*gyroscope*), που χρησιμοποιείται για να διατηρεί τον προσανατολισμό του ρομποτικού οχήματος σύμφωνα με ένα πλαίσιο αναφοράς. Τα γυροσκόπια μπορούν να χωριστούν σε δύο κατηγορίες, τα μηχανικά γυροσκόπια και τα οπτικά γυροσκόπια. Τα κλασικά γυροσκόπια έχουν μηχανικούς περιστρεφόμενους δίσκους που είναι τοποθετημένοι σε μια σταθερή βάση, έτσι ώστε ο άξονας περιστροφής να παραμένει σταθερός, όπως φαίνεται στην Εικόνα 25α [6]. Τα οπτικά γυροσκόπια βασίζονται στην αποστολή δύο ακτινών laser σε αντίθετες κατευθύνσεις γύρω από μια κυκλική ή τριγωνική διαδρομή, όπως φαίνεται στην Εικόνα 25β. Στην περίπτωση που το

γυροσκόπιο περιστρέφεται τότε το η διαδρομή που ακολουθεί η μια ακτίνα θα είναι μακρύτερη από τη διαδρομή που ακολουθεί η δεύτερη ακτίνα. Η διαφορά είναι ανάλογη με την γωνιακή ταχύτητα και μπορεί να μετρηθεί και να μεταφερθεί προς το σύστημα πλοήγησης.



Εικόνα 25 α) Μηχανικό γυροσκόπιο δύο αξόνων [18], β) Οπτικό γυροσκόπιο [46].

Παγκόσμιο σύστημα εντοπισμού – GPS

Το σύστημα εντοπισμού GPS (*Global Positioning System*) είναι ένα δορυφορικό σύστημα, το οποίο με τη βοήθεια ενός επίγειου δέκτη GPS, προσδιορίζει με σχετική ακρίβεια την αντίστοιχη γεωγραφική του θέση [29]. Το GPS χωρίζεται σε τρία τμήματα, το τμήμα διαστήματος, το τμήμα ελέγχου και το τμήμα του χειριστή [26]. Το τμήμα του διαστήματος αποτελείται από 24 δορυφόρους, χωρισμένους σε έξι κυκλικές τροχιές, με τέσσερις δορυφόρους σε κάθε τροχιά. Οι δορυφόροι περιστρέφονται στην τροχιά μεσαίου ύψους (*Medium Earth Orbit – MEO*), έχουν κλίση 55° από τον ισημερινό και μια περίοδος περιστροφής διαρκεί δώδεκα ώρες. Ο τρόπος τοποθέτησής τους εξασφαλίζει πως τουλάχιστον τέσσερις δορυφόροι θα είναι ορατοί από οποιοδήποτε σημείο της γης, οποιαδήποτε στιγμή. Το τμήμα ελέγχου αποτελεί ένα παγκόσμιο δίκτυο από επίγειες εγκαταστάσεις και έχει στόχο τον έλεγχο και τη συντήρηση του συστήματος. Το συγκεκριμένο τμήμα αποτελείται από σταθμούς παρακολούθησης, επίγειες κεραίες και έναν σταθμό ελέγχου. Οι σταθμοί παρακολούθησης ακολουθούν τους δορυφόρους, συλλέγουν σήματα GPS και προωθούν τις πληροφορίες στον σταθμό ελέγχου. Οι επίγειες κεραίες επικοινωνούν με τους δορυφόρους μέσα από τη ζώνη δορυφορικών συχνοτήτων S με σκοπό να στείλει εντολές, να μεταφέρει δεδομένα πλοήγησης και κώδικες προγραμματισμού. Ο σταθμός ελέγχου ελέγχει τους δορυφόρους και συλλέγει δεδομένα από τους σταθμούς παρακολούθησης και τις επίγειες κεραίες. Τέλος, το τμήμα του χειριστή αποτελείται από τους δέκτες των σημάτων GPS οι οποίοι δέχονται κωδικοποιημένα σήματα με σκοπό να προσδιορίσουν τη θέση, την ταχύτητα και τον χρόνο.

Μηχανική όραση

Η ορατότητα παρέχει έναν μεγάλο αριθμό πληροφοριών σχετικά με το περιβάλλον του ρομπότ και ενεργοποιεί την αλληλεπίδραση ευφών συστημάτων μέσα στο δυναμικό περιβάλλον. Η ανάλυση των εικόνων και η επεξεργασία τους αποτελούν δύο κατηγορίες γνωστές ως μηχανική όραση (*computer vision*) και επεξεργασία εικόνας [18].

Η ψηφιακή κάμερα αποτελεί ένα εργαλείο που χρησιμοποιείται ευρέως στην ρομποτική, καθώς προσφέρει λεπτομερείς πληροφορίες σχετικά με το περιβάλλον στο οποίο κινείται ένα ρομπότ [6]. Οι συγκεκριμένες κάμερες χρησιμοποιούν μια συσκευή συζευγμένου φορτίου (*charge-coupled*) που ανιχνεύει κύματα φωτός και επιστρέφει έναν πίνακα από pixel. Προκειμένου να διεκπεραιωθούν οι εφαρμογές των ρομποτικών οχημάτων, η ανάλυση των εικόνων πρέπει να είναι αρκετά υψηλή ώστε να μπορεί να εντοπίζει αντικείμενα από μια συγκεκριμένη απόσταση [21]. Υπάρχουν τρεις τρόποι που χρησιμοποιούνται γενικά για τη λήψη δεδομένων από κάμερες. Στον πρώτο τρόπο η εφαρμογή αιτείται ένα στιγμιότυπο από το πρόγραμμα οδήγησης (*driver*) και μπλοκάρει την εκτέλεση του επεξεργαστή CPU (*Central Processing Unit*). Έπειτα το πρόγραμμα οδήγησης περιμένει το επόμενο ολοκληρωμένο στιγμιότυπο από την κάμερα και στη συνέχεια το λαμβάνει. Μόλις το στιγμιότυπο διαβαστεί πλήρως, τα δεδομένα περνούν μέσα στην εφαρμογή. Σε αυτό το σημείο το πρόγραμμα οδήγησης χρειάζεται να περιμένει να ληφθεί το καινούριο στιγμιότυπο, που σημαίνει ότι η εφαρμογή αναμένει την εύρεση του νέου στιγμιότυπου και να διαβαστεί το παρόν στιγμιότυπο από το πρόγραμμα οδήγησης. Στον δεύτερο τρόπο το πρόγραμμα οδήγησης λαμβάνει ένα στιγμιότυπο από την κάμερα και την αποθηκεύει σε μια ενδιάμεση μνήμη. Στη συνέχεια, μόλις η εφαρμογή αιτηθεί ένα στιγμιότυπο, ο δείκτης της ενδιάμεσης μνήμης περνάει το στιγμιότυπο στην εφαρμογή. Τέλος, στον τρίτο τρόπο το πρόγραμμα οδήγησης λαμβάνει κάθε στιγμιότυπο της κάμερας και το αποθηκεύει σε μια ενδιάμεση μνήμη. Όταν ένα στιγμιότυπο διαβαστεί πλήρως ένα σήμα στέλνεται στην εφαρμογή και γίνεται επεξεργασία των δεδομένων του στιγμιότυπου.

2.2 Κατασκευή Μονοπατιού

Όπως ήδη έχει αναφερθεί η ρομποτική είναι κλάδος της τεχνολογίας που ασχολείται με ρομπότ, τα οποία είναι προγραμματιζόμενες μηχανές που μπορούν να διεξάγουν εργασίες ημιαυτόνομα ή πλήρως αυτόνομα [30]. Αντίθετα, η τεχνητή νοημοσύνη (*Artificial Intelligence – AI*) είναι ένας κλάδος της πληροφορικής και αφορά την ανάπτυξη υπολογιστικών συστημάτων που έχουν τη δυνατότητα να πραγματοποιήσουν εργασίες, για τις οποίες διαφορετικά θα ήταν απαραίτητη η ανθρώπινη νοημοσύνη. Ένας από τους κύριους στόχους ενός ρομποτικού οχήματος είναι η ικανότητά του να διεκπεραιώνει αποστολές σε επικίνδυνους και γεμάτο εμπόδια χώρους. Για την κατασκευή ενός αυτόνομου ρομποτικού οχήματος θα πρέπει να δημιουργηθεί ένα σύστημα το οποίο αντιλαμβάνεται τον χώρο, αντιδρά σε μη προβλεπόμενες καταστάσεις και επαναπρογραμματίζεται δυναμικά με σκοπό να πετύχει τον στόχο του.

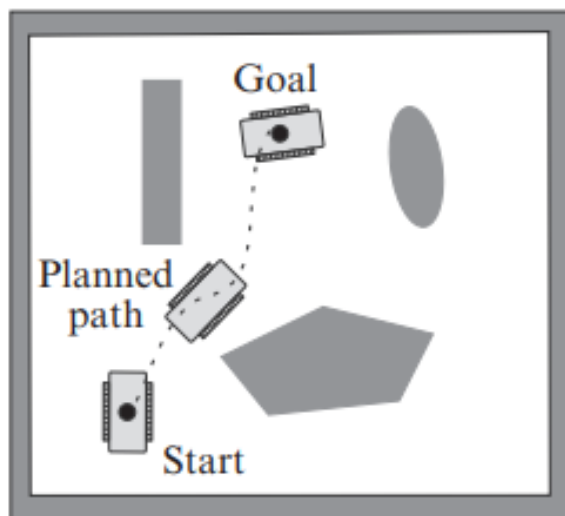
Στην παρούσα διπλωματική εργασία το πρόβλημα που καλείται να λύσει το ρομποτικό σύστημα είναι το πρόβλημα της αυτόνομης πλοήγησης. Δηλαδή, καλείται να δημιουργήσει ένα μονοπάτι χρησιμοποιώντας αλγόριθμους αναζήτησης μονοπατιού και στη συνέχεια, χρησιμοποιώντας τη μέθοδο του δυναμικού παραθύρου, να πραγματοποιήσει τη διαδρομή προς την τελική θέση αποφεύγοντας οποιοδήποτε εμπόδιο κατά την πλοήγησή του. Η κατασκευή ενός μονοπατιού (*path planning*) από μια θέση A σε μια θέση B σε συνδυασμό με την ταυτόχρονη αποφυγή εμποδίων και την ανταπόκριση σε αλλαγές που μπορούν να προκύψουν στον χώρο, αποτελούν καταστάσεις τις οποίες κάθε ρομποτικό όχημα πρέπει να εκτελέσει ώστε να θεωρηθεί αυτόνομο [11].

2.2.1 Αλγόριθμοι κατασκευής μονοπατιού

Η κατασκευή μονοπατιού αποτελεί μια από τις βασικές τεχνικές για την καθοδήγηση ενός ρομποτικού οχήματος σε ένα στατικό ή δυναμικό περιβάλλον [31]. Χρησιμοποιείται για την επίλυση προβλημάτων σε διάφορα πεδία, όπως ο προγραμματισμός μιας απλής διαδρομής ή η επιλογή μιας σειράς ενεργειών που είναι κατάλληλες για να οδηγήσουν το ρομποτικό όχημα σε έναν συγκεκριμένο στόχο [11]. Η διαδικασία της κατασκευής μονοπατιού μπορεί να χρησιμοποιηθεί σε γνωστό ή σε μερικώς γνωστό περιβάλλον αλλά και σε εντελώς άγνωστους χώρους, στους οποίους χρησιμοποιούνται πληροφορίες από τους αισθητήρες, ώστε να προσδιοριστεί η επιθυμητή κίνηση για το ρομποτικό όχημα.

Με βάση το εύρος στόχων που μπορούν να υπάρξουν στην κατασκευή μονοπατιού, μπορεί να χωριστεί σε σχετική κατασκευή μονοπατιού (*local path planning*) και σε απόλυτη κατασκευή μονοπατιού (*global path planning*). Η σχετική κατασκευή μονοπατιού αφορά στα μερικώς ή ολικά άγνωστα δεδομένα του χώρου που κινείται το ρομποτικό όχημα, τα οποία συλλέγονται από τους αισθητήρες σε πραγματικό χρόνο για τον προσδιορισμό της κατανομής των κοντινών εμποδίων. Επιπλέον, επιλέγεται η βέλτιστη διαδρομή από το σημείο έναρξης μέχρι το επιθυμητό σημείο άφιξης, οπότε η σχετική κατασκευή μονοπατιού μπορεί να ονομαστεί και δυναμικός σχεδιασμός (*dynamic planning*). Η απόλυτη κατασκευή μονοπατιού, γνωστή και ως στατικός σχεδιασμός (*static planning*) αφορά σε όλα τα γεωγραφικά δεδομένα και την κατασκευή εφικτών μονοπατιών με βάση αυτά.

Ο σκοπός της κατασκευής μονοπατιού αποτελεί την εύρεση μιας συνεχόμενης πορείας που μπορεί να οδηγήσει το ρομποτικό όχημα από την αρχική έως την τελική του διάταξη. Για να μπορέσει το ρομπότ να ολοκληρώσει την εργασία είναι σημαντικό η πορεία του να βρίσκεται μέσα στον χώρο διάταξης και συγκεκριμένα μέσα στον ελεύθερο χώρο, όπως φαίνεται στην Εικόνα 26. Επίσης, κατά την κατασκευή του μονοπατιού το ρομποτικό σύστημα χρησιμοποιεί έναν γνωστό χάρτη του χώρου, ο οποίος βρίσκεται αποθηκευμένος στη μνήμη του ρομπότ.



Εικόνα 26 Μια πιθανή πορεία που μπορεί να ακολουθήσει το ρομπότ [11]

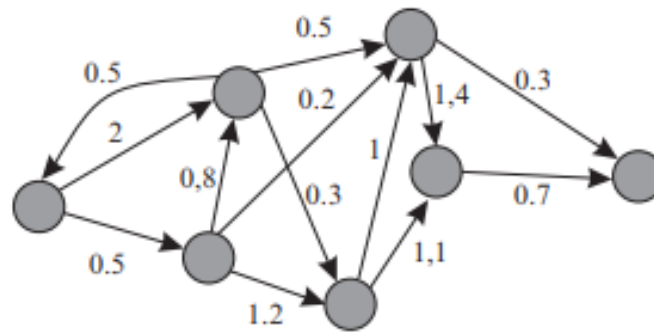
Η θέση ή η διάταξη του ρομποτικού οχήματος δίνει την πιθανή πόζα του μέσα στον χώρο και μπορεί να παρουσιαστεί ως ένα σημείο μέσα στον χώρο διάταξης, το οποίο θα περιλαμβάνει όλες τις πιθανές θέσεις του ρομποτικού οχήματος. Επομένως, ένα ρομποτικό όχημα μπορεί να κινηθεί από μια θέση σε μια άλλη, εφαρμόζοντας μια σειρά από διαφορετικές ενέργειες, οι οποίες μπορούν να δημιουργήσουν μια κατάλληλη πορεία που θα οδηγήσει το ρομποτικό όχημα από την αρχική διάταξη, μέσα από ενδιάμεσες θέσεις, στην τελική διάταξη. Οι ενέργειες που θα επιλεγθούν σε κάθε θέση και η επόμενη θέση που θα προκύψει εξαρτάται από τον αλγόριθμο που θα χρησιμοποιηθεί για την κατασκευή του μονοπατιού και τα κριτήρια που εφαρμόζονται. Ο αλγόριθμος επιλέγει την αμέσως επόμενη κατάλληλη θέση ανάμεσα σε ένα σύνολο πιθανών θέσεων, στις οποίες το ρομπότ μπορεί να οδηγηθεί από τη θέση που βρίσκεται. Η επιλογή της κατάλληλης θέσης γίνεται με βάση τα λειτουργικά κριτήρια, που συνήθως αφορούν σε μετρήσεις αποστάσεων, όπως η μικρότερη Ευκλείδεια απόσταση προς το επιθυμητό σημείο άφιξης.

Ανάμεσα στο σημείο εκκίνησης και το σημείο άφιξης μπορούν να προκύψουν μια ή περισσότερες διαδρομές ή ακόμα και καμία διαδρομή, εάν τα εμπόδια μέσα στον χώρο δεν το επιτρέψουν. Συνήθως, υπάρχουν περισσότερες πιθανές διαδρομές που μπορεί να ακολουθήσει το ρομποτικό όχημα, οι οποίες φιλτράρονται και στη συνέχεια ελαττώνονται αριθμητικά, εφαρμόζοντας ορισμένα κριτήρια εύρεσης της βέλτιστης διαδρομής. Τέτοια κριτήρια μπορούν να είναι σχετικά με το μήκος του μονοπατιού, όπως μικρότερο μήκος διαδρομής, με τη χρονική διάρκεια διεκπεραίωσης της διαδρομής, όπως συντομότερος χρόνος διεκπεραίωσης, με την απόσταση οχήματος από τα εμπόδια, όπως μέγιστη δυνατή απόσταση από τα εμπόδια, με την ομαλότητα του μονοπατιού, όπως ομαλό μονοπάτι χωρίς απότομους ελιγμούς και με κινητικούς περιορισμούς, όπως μη-ολονομικοί περιορισμοί.

Προκειμένου να γίνει η κατασκευή του μονοπατιού είναι απαραίτητο ο χώρος στον οποίο κινείται το ρομποτικό όχημα να μετατραπεί σε έναν διακριτό χάρτη, ώστε να μπορεί να γίνει κατανοητός από τον αλγόριθμο αναζήτησης μονοπατιού που θα επιλεγθεί. Στην παρούσα εργασία θα μελετηθούν δύο βασικές κατηγορίες μεθόδων αναζήτησης μονοπατιών, η αναζήτηση μονοπατιού σε γράφο και η αναζήτηση μονοπατιού με τη χρήση δυναμικού πεδίου.

Αναζήτηση μονοπατιού σε γράφο

Ένας χώρος διάταξης περιλαμβάνει στον ελεύθερο χώρο του όλες τις πιθανές διατάξεις ενός ρομποτικού συστήματος και τον χώρο που καταλαμβάνουν τα εμπόδια. Κατά την αναζήτηση μονοπατιού σε γράφο (*graph search*) παρουσιάζεται ένα υποσύνολο διατάξεων, το οποίο περιλαμβάνει την αρχική και τελική διάταξη του ρομποτικού οχήματος, τον επιθυμητό αριθμό ενδιάμεσων διατάξεων και τις μεταβάσεις που μπορούν να προκύψουν μεταξύ τους [32]. Οι θέσεις παρουσιάζονται από κύκλους, που ονομάζονται κόμβοι και οι συνδέσεις μεταξύ των γειτονικών κόμβων δίνονται από γραμμές, που ονομάζονται ακμές. Οι ακμές παρουσιάζουν την κίνηση που απαιτείται ώστε το ρομποτικό σύστημα να κινηθεί μεταξύ των θέσεων. Στην Εικόνα 27 παρουσιάζεται ένα παράδειγμα κατευθυνόμενου γράφου στον οποίο έχουν προστεθεί βάρη και οι πιθανές κατευθύνσεις στις οποίες μπορεί να μεταβεί. Τα βάρη χρησιμοποιούνται για να υπολογιστούν οι ενέργειες που απαιτούνται για την εκτέλεση και την μετάβαση μεταξύ των θέσεων.



Εικόνα 27 Κατευθυνόμενος γράφος με βάρη [11]

Στην παρούσα εργασία παρουσιάζονται δύο τύποι αναζήτησης σε γράφο, η ντετερμινιστική (*deterministic*) και η τυχαία (*randomized*) αναζήτηση σε γράφο. Ο στόχος της ντετερμινιστικής αναζήτησης είναι η εύρεση του βέλτιστου μονοπατιού ανάμεσα στην αρχική και τελική διάταξη, πληρώντας τα κριτήρια βελτιστοποίησης που έχουν τεθεί. Ο ντετερμινιστικός αλγόριθμος που χρησιμοποιείται στην παρούσα εργασία είναι ο αλγόριθμος Dijkstra. Αντίθετα, ο στόχος της τυχαίας αναζήτησης είναι η εύρεση μονοπατιού σε ταχύτατο χρόνο, χωρίς να τίθεται ως προϋπόθεση η βέλτιστη διαδρομή. Ο αλγόριθμος τυχαίας αναζήτησης που χρησιμοποιείται στην παρούσα εργασία αποτελεί ο RRT (*Rapidly exploring Random Tree*).

Κατά την ντετερμινιστική αναζήτηση οι αλγόριθμοι αναζήτησης ξεκινούν τη λειτουργία τους ελέγχοντας εάν ο κόμβος της αρχικής θέσης συμπίπτει με τον κόμβο της τελικής θέσης [32]. Εάν η υπόθεση είναι λανθασμένη η αναζήτηση επεκτείνεται σε άλλους κόμβους που υπάρχουν γύρω του, με αποτέλεσμα η διαδικασία να επαναλαμβάνεται μέχρι ο κόμβος που θα επιλεγεί να συμπίπτει με τον κόμβο που περιγράφει την τελική θέση. Η επιλογή των κόμβων κατά τη διάρκεια της αναζήτησης εξαρτάται από τον αλγόριθμο που χρησιμοποιείται κάθε φορά και η διαδικασία αναζήτησης ολοκληρώνεται μέχρι να βρεθεί η επιθυμητή λύση ή μέχρι ο αλγόριθμος να περάσει από όλους τους κόμβους του γράφου. Κατά τη διαδικασία αναζήτησης σε έναν γράφο δημιουργούνται λίστες κόμβων τους οποίους έχει ήδη επισκεφτεί ο αλγόριθμος με σκοπό να μην ελεγχθούν ξανά. Για παράδειγμα, οι κόμβοι που μπορούν να αναπτύξουν μια καινούρια πορεία αναζήτησης διατηρούνται σε μια ανοιχτή λίστα και οι κόμβοι που έχουν ελεγχθεί και δεν έχουν πιθανότητα να επιλεγούν για την δημιουργία μονοπατιού προστίθενται σε μια κλειστή λίστα. Οι αλγόριθμοι αναζήτησης σε γράφο μπορούν να χωριστούν σε πληροφορημένους (*informed*) ή μη πληροφορημένους (*non informed*) αλγόριθμους. Οι μη πληροφορημένοι αλγόριθμοι δεν χρησιμοποιούν δεδομένα που βρίσκονται εκτός της περιγραφής του προβλήματος, επομένως αναζητούν συστηματικά τον γράφο και δεν διακρίνουν κόμβους ανάμεσα στους λιγότερο υποσχόμενους για την εύρεσης λύσης. Οι πληροφορημένοι αλγόριθμοι κάνουν ευρετική αναζήτηση (*heuristic search*) ή αλλιώς πληροφορημένη αναζήτηση και συμπεριλαμβάνουν επιπρόσθετες πληροφορίες για τους κόμβους, με αποτέλεσμα να επιλέγονται περισσότερο υποσχόμενοι κόμβοι και η αναζήτηση προς την τελική λύση να είναι πιο αποδοτική.

Κατά την τυχαία αναζήτηση μια μέθοδος αναζήτησης μονοπατιού αποτελεί η κατασκευή μονοπατιού με δειγματοληπτικούς αλγόριθμους (*Sampling-Based Path-Planning*) κατά την οποία

γίνεται δειγματοληψία τυχαίων σημείων για τη διάταξη του ρομποτικού οχήματος στα οποία στη συνέχεια γίνεται έλεγχος εάν ανήκουν στον ελεύθερο χώρο. Από το σύνολο αυτών των σημείων και των συνδέσεων μεταξύ τους γίνεται αναζήτηση μιας διαδρομής μεταξύ του σημείου εκκίνησης και του επιθυμητού σημείου άφιξης. Η μέθοδος δειγματοληψίας δεν απαιτεί υπολογισμούς του ελεύθερου χώρου διάταξης Q_{free} , καθώς τα αυθαίρετα σημεία χρησιμοποιούνται για την απεικόνιση του χώρου διάταξης και ανεξάρτητα από τη γεωμετρία του χώρου, προκύπτει μια λύση στην αναζήτηση του μονοπατιού για μεγάλο εύρος προβλημάτων. Η συγκεκριμένη μέθοδος μπορεί να χωριστεί στην κατηγορία που εξειδικεύεται στην αναζήτηση ενός μονοπατιού και σε εκείνη που μπορεί να χρησιμοποιηθεί για περισσότερες από μια αναζητήσεις. Η πρώτη κατηγορία αφορά την αναζήτηση μονοπατιού, από ένα μοναδικό σημείο εκκίνησης προς ένα μοναδικό σημείο άφιξης, στον συντομότερο δυνατό χρόνο. Οπότε, ο αλγόριθμος επικεντρώνεται μόνο στα σημεία του χώρου, όπου οι πιθανότητες να βρεθεί λύση είναι υψηλότερες με αποτέλεσμα να προστίθενται διαρκώς νέα σημεία και συνδέσεις στον γράφο μέχρι να βρεθεί η λύση. Στη δεύτερη κατηγορία εκτελείται πρώτα η συνολική διαδικασία της παρουσίασης ολόκληρου του ελευθέρου χώρου σε μη κατευθυνόμενο γράφο, με αποτέλεσμα να μπορεί να βρεθεί λύση για την κατασκευή του μονοπατιού για ένα αυθόρμητο ζευγάρι σημείου εκκίνησης και άφιξης.



Εικόνα 28 Ανάπτυξη του αλγόριθμου RRT από 45 μέχρι 2345 επαναλήψεις [18].

Δυναμικό πεδίο

Η μέθοδος του δυναμικού πεδίου (*potential field*) για την αναζήτηση μονοπατιού περιγράφει έναν χώρο με δυναμικό πεδίο όπου η θέση άφιξης έχει χαμηλό δυναμικό πεδίο, το οποίο αυξάνεται όσο η απόσταση μεγαλώνει συγκριτικά με τον στόχο και μεγιστοποιείται στα σημεία όπου υπάρχουν εμπόδια [11]. Το δυναμικό πεδίο εκφράζεται ως σύνολο του ελκτικού πεδίου $U_{attr}(q)$, λόγω της θέσης άφιξης και του απωθητικού πεδίου $U_{rep}(q)$, λόγω των εμποδίων:

$$U(q) = U_{attr}(q) + U_{rep}(q)$$

Το σημείο άφιξης αποτελεί το απόλυτο ελάχιστο του δυναμικού πεδίου και το ελκτικό δυναμικό πεδίο $U_{attr}(q)$ μπορεί να θεωρηθεί ανάλογο με την τετραγωνική ρίζα της Ευκλείδειας απόστασης

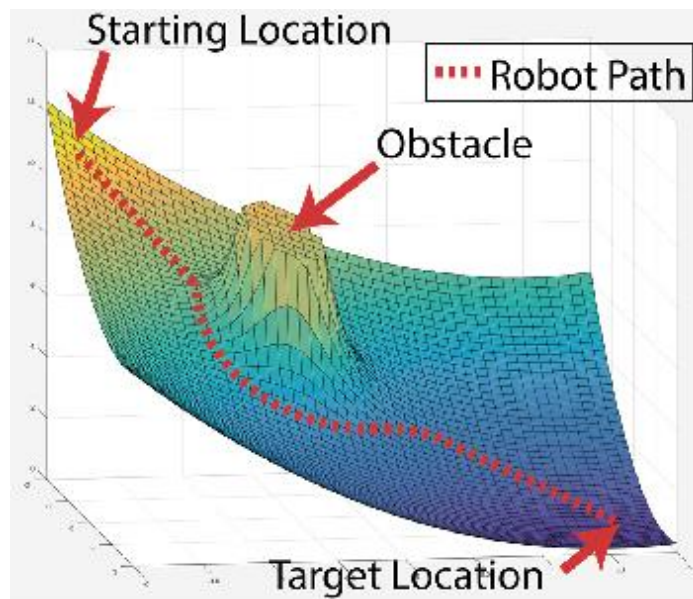
προς το σημείο άφιξης: $D(q, q_{goal}) = \sqrt{(x - x_{goal})^2 + (y - y_{goal})^2}$, με αποτέλεσμα το δυναμικό ελκτικό πεδίο $U_{attr}(q)$ να εκφράζεται ως:

$$U_{attr}(q) = k_{attr} \frac{1}{2} D^2(q, q_{goal})$$

όπου, k_{attr} θεωρείται μια θετική μεταβλητή. Επίσης, το απωθητικό δυναμικό πεδίο $U_{rep}(q)$ πρέπει να έχει υψηλή τιμή κοντά σε εμπόδια και χαμηλή τιμή όσο η απόσταση από τα εμπόδια αυξάνεται $D(q, q_{obst})$. Η τιμή του απωθητικού δυναμικού πεδίου πρέπει να μηδενίζεται όταν η τιμή $D(q, q_{obst})$ είναι μεγαλύτερη από την τιμή κάποιου ορίου D_0 , οπότε το δυναμικό απωθητικό πεδίο $U_{rep}(q)$ να εκφράζεται ως:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{D(q, q_{obst})} - \frac{1}{D_0} \right)^2, & D(q) \leq D_0 \\ 0, & D(q) > D_0 \end{cases}$$

όπου k_{rep} είναι μια θετική μεταβλητή και η τιμή $D(q, q_{obst})$ αποτελεί την απόσταση προς το κοντινότερο σημείο του πλησιέστερου εμποδίου. Επομένως, χρησιμοποιώντας τη μέθοδο του δυναμικού πεδίου για την παρουσίαση του χώρου το ρομποτικό όχημα μπορεί να φτάσει στον στόχο ακολουθώντας τη χαμηλή τιμή του δυναμικού πεδίου, η οποία υπολογίζεται από τη γνωστή θέση του ρομποτικού οχήματος.



Εικόνα 29 Παράδειγμα δυναμικού πεδίου που χρησιμοποιείται για την πλοήγηση προς μια τελική θέση αποφεύγοντας και την αποφυγή εμποδίων [47].

Αλγόριθμος Dijkstra

Ο αλγόριθμος Dijkstra είναι ένας μη πληροφορημένος ντετερμινιστικός αλγόριθμος που βρίσκει το συντομότερο μονοπάτι από τον αρχικό κόμβο προς όλους τους κόμβους του γράφου με αποτέλεσμα να προκύπτει και το συντομότερο δέντρο για το μονοπάτι [32]. Ο αλγόριθμος

Dijkstra επινοήθηκε από τον Ολλανδό επιστήμονα Edsger W. Dijkstra το 1956 και πρόκειται για έναν αλγόριθμο που επιλύει προβλήματα εύρεσης συντομότερων διαδρομών (*Single-Source Shortest Path*) σε έναν κατευθυνόμενο γράφο με μη αρνητικά βάρη στις ακμές του [33]. Με τη χρήση του αλγόριθμου Dijkstra ολόκληρος ο χώρος μετατρέπεται σε ένα δίκτυο μέσα στο οποίο οι κόμβοι του δικτύου αναπαριστούν διατάξεις του χώρου εργασίας και οι συνδέσεις αναπαριστούν την κίνηση από τον έναν κόμβο προς τον επόμενο [19]. Συνεπώς, κάθε σύνδεση σχετίζεται με μια μη αρνητική τιμή και αναπαριστά το κόστος της μετακίνησης από τον έναν κόμβο στον άλλον. Η δημιουργία του μονοπατιού γίνεται με τη συλλογή των μετακινήσεων από τον αρχικό μέχρι τον τελικό κόμβο και το κόστος της διαδρομής είναι το άθροισμα από όλα τα κόστη της διαδρομής.

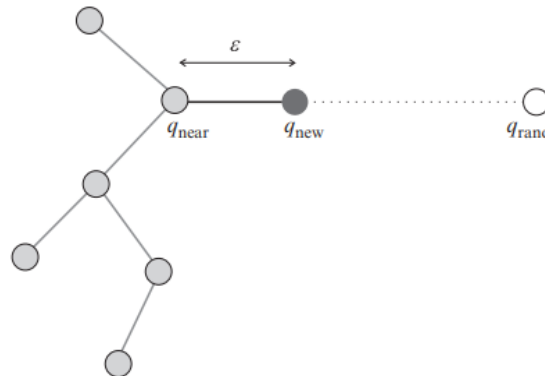
Ο αλγόριθμος επιτυγχάνει την εύρεση της συντομότερης διαδρομής από την αρχική θέση προς την τελική θέση υπολογίζοντας τα κόστη της διαδρομής ανάμεσα στον αρχικό κόμβο και κάθε κόμβου που επιλέγεται [32]. Επομένως, το συνολικό κόστος του μονοπατιού υπολογίζεται από το κόστος που έχει προκύψει στον προηγούμενο κόμβο και το κόστος για τη μετάβαση ανάμεσα στον προηγούμενο και τον επιλεγμένο κόμβο. Κατά την έναρξη της λειτουργίας του αλγόριθμου η ανοιχτή λίστα κόμβων περιέχει μόνο τον κόμβο έναρξης, ο οποίος έχει μηδενικό κόστος διαδρομής και δεν έχει καμία σύνδεση με προηγούμενο κόμβο. Στη συνέχεια, ακολουθεί μια συγκεκριμένη σειρά πέντε βημάτων, όπου στο πρώτο βήμα γίνεται η επιλογή του πρώτου κόμβου της ανοιχτής λίστας. Η ανοιχτή λίστα είναι ταξινομημένη με βάση τους κόμβους που έχουν επιλεγθεί, με αύξουσα σειρά, κατά την οποία ο πρώτος κόμβος έχει και το μικρότερο κόστος διαδρομής. Το δεύτερο βήμα υπολογίζει το συνολικό κόστος διαδρομής προς όλους τους κόμβους που μπορεί να φτάσει ο επιλεγμένος κόμβος. Στο τρίτο βήμα αποθηκεύονται οι υπολογισμοί και οι κατάλληλες συνδέσεις προς τον επιλεγμένο κόμβο, για όσους κόμβους δεν έχει προηγηθεί αποθήκευση των πληροφοριών. Στο τέταρτο βήμα γίνεται σύγκριση ανάμεσα στα κόστη διαδρομής που έχουν προκύψει και στις προηγούμενες επαναλήψεις και διατηρείται το μικρότερο κόστος διαδρομής μαζί με τη σύνδεσή του. Τέλος, στο πέμπτο βήμα γίνεται η ταξινόμηση με βάση το χαμηλότερο κόστος διαδρομής των κόμβων που προστίθενται, έτσι ώστε ο κόμβος με το χαμηλότερο κόστος διαδρομής να μπορεί να βρεθεί πιο γρήγορα. Έπειτα, ο επιλεγμένος κόμβος προστίθεται στην κλειστή λίστα. Ο αλγόριθμος τερματίζει όταν η ανοιχτή λίστα αδειάσει και τα αποτελέσματα αποτελούν τις συντομότερες διαδρομές από τον αρχικό κόμβο προς όλους τους υπόλοιπους κόμβους. Στην περίπτωση που απαιτείται η συντομότερη διαδρομή προς μια μόνο τελική θέση τότε ο αλγόριθμος τερματίζει μόλις ο τελικός κόμβος προστεθεί στην κλειστή λίστα.

Dijkstra(G, w, s)	
1	Initialize-single-source(G, s)
2	S = \emptyset
3	Q = G.V
4	while Q \neq 0
5	u = Extract-min(Q)
6	S = S \cup {u}
7	For each vertex v \in G.Adj[u]
8	Relax(u,v,w)

Αλγόριθμος 3 Dijkstra Algorithm [49].

Αλγόριθμος RRT

Ο αλγόριθμος RRT αποτελεί μια τυχαία μέθοδο αναζήτησης μονοπατιού από μια γνωστή αρχική θέση προς μια τελική θέση [11]. Σε κάθε επανάληψη του αλγορίθμου προστίθεται μια νέα σύνδεση προς την κατεύθυνση της αυθαίρετης θέσης που προστέθηκε στον γράφο προς τη θέση που ήδη προϋπήρχε σε αυτόν. Στην πρώτη επανάληψη του αλγορίθμου η αρχική διάταξη q_i παρουσιάζει το δέντρο στον γράφο. Οπότε, σε κάθε επόμενη επανάληψη μια διάταξη q_{rand} επιλέγεται τυχαία και αναζητείται ο κοντινότερος κόμβος q_{near} από τον υπάρχοντα γράφο. Στην κατεύθυνση από τον κόμβο q_{near} προς τον κόμβο q_{rand} υπολογίζεται ένας νέος κόμβος q_{new} σε μια προκαθορισμένη απόσταση ϵ και εάν ο νέος κόμβος και η σύνδεση του προς τον κόμβο q_{near} βρίσκονται μέσα στον ελεύθερο χώρο, τότε ο νέος κόμβος με τη σύνδεσή του προτίθενται στον γράφο, όπως φαίνεται στο παράδειγμα της Εικόνα 30. Μόλις ικανοποιηθεί το κριτήριο για τον τερματισμό του αλγορίθμου, επιλέγεται η τελική θέση και γίνεται έλεγχος για τη δυνατότητα σύνδεσής του στον γράφο. Ο συγκεκριμένος αλγόριθμος έχει μόνο δύο παραμέτρους, το μήκος της απόστασης ϵ και ο επιθυμητός αριθμός επαναλήψεων που καθορίζουν την κατάσταση τερματισμού του αλγορίθμου.



Εικόνα 30 Παράδειγμα αλγορίθμου RRT [11]

RRT Algorithm(q_{start} , q_{goal} , ϵ , n)	
1	G.initialize(q_{start})
2	for $i=1$ to n do
3	$q_{rand} = \text{Sample}()$
4	$q_{near} = \text{near}(q_{rand}, G)$
5	$q_{new} = \text{steer}(q_{rand}, q_{near}, \epsilon)$
6	G.add_nodes(q_{new})
7	G.add_edge(q_{new}, q_{near})
8	if $q_{new}=q_{goal}$
9	Success()

Αλγόριθμος 4 RRT (Rapidly exploring Random Tree) [50].

2.2.2 Μέθοδοι αποφυγής εμποδίων

Η πλοήγηση ενός ρομποτικού οχήματος σχετίζεται με την ικανότητα του ρομποτικού οχήματος να κινείται στον χώρο ώστε να φτάσει στην επιθυμητή τελική θέση-στόχος χωρίς να συγκρουστεί με κάποιο εμπόδιο. Η πλοήγηση του ρομπότ από την τρέχουσα θέση του προς την

επιθυμητή τελική θέση περιέχει τον υπολογισμό της διαδρομής. Η διαδικασία αυτή απαιτεί την ύπαρξη ενός χάρτη, της τρέχουσας θέσης του ρομπότ και της τελικής θέσης. Οπότε, είναι απαραίτητο να χρησιμοποιηθούν οι κατάλληλοι αλγόριθμοι αποφυγής εμποδίων έτσι ώστε να αποφευχθούν οι συγκρούσεις και να μπορεί το ρομποτικό σύστημα να επαναπροσδιορίσει τη διαδρομή ή να σταματήσει πριν τη σύγκρουση.

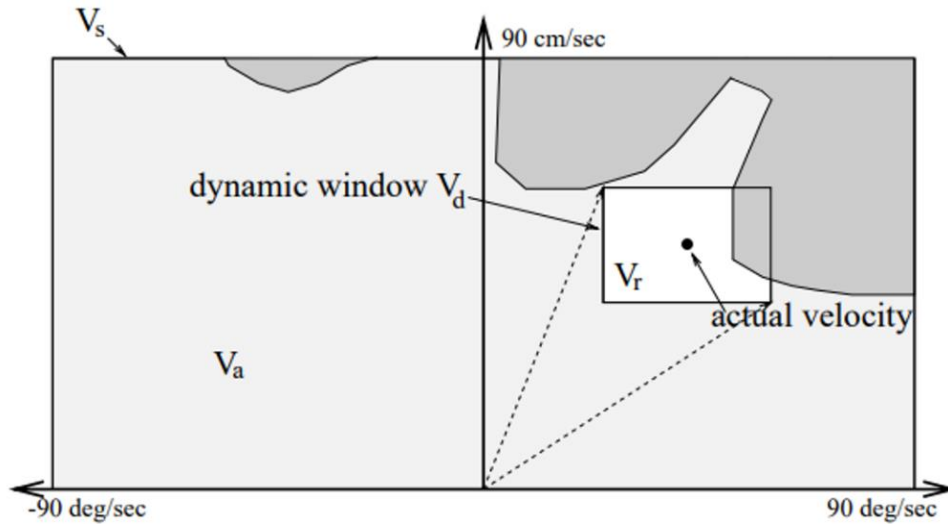
Η διαδικασία της αποφυγής εμποδίων εστιάζει στην αλλαγή πορείας του ρομποτικού οχήματος ύστερα από πληροφορίες που δέχεται από τους αισθητήρες του κατά τη διάρκεια της πλοήγησης [18]. Συνεπώς, το αποτέλεσμα της κίνησης του ρομπότ αποτελεί τον συνδυασμό των συναρτήσεων των τρεχουσών ή τελευταίων μετρήσεων των αισθητήρων και της τελικής διάταξης ή των σχετικών θέσεων μέχρι την τελική διάταξη. Η μέθοδος που χρησιμοποιείται και παρουσιάζεται στην παρούσα διπλωματική εργασία εξαρτάται από την ύπαρξη ενός χάρτη και τη γνώση σχετικά με την ακριβή θέση του ρομποτικού οχήματος σε αυτόν. Οι αλγόριθμοι αποφυγής εμποδίων είναι απαραίτητοι για την πλοήγηση των ρομποτικών οχημάτων και συνδυάζονται με τη διαδικασία κατασκευής μονοπατιών που αποτελεί τη βασική προϋπόθεση για τον σχεδιασμό τους. Οι μέθοδοι που χρησιμοποιούνται για την αποφυγή εμποδίων για τα ρομποτικά οχήματα μπορούν να χωριστούν, γενικά, σε δύο κατηγορίες, τις καθολικές (*global*) και τις σχετικές (*local*). Οι καθολικές τεχνικές, όπως είναι η δημιουργία ενός χάρτη διαδρομής και οι μέθοδοι δυναμικού πεδίου, θεωρούν πως το μοντέλο του χώρου στον οποίο κινείται το ρομποτικό όχημα είναι διαθέσιμο [34]. Αντιθέτως, οι σχετικές τεχνικές χρησιμοποιούν μικρότερα κομμάτια του χώρου για να ελέγξουν το ρομποτικό όχημα.

Μια τεχνική αποφυγής εμποδίων που λαμβάνει υπόψη τις κινηματικές παραμέτρους του ρομποτικού οχήματος αποτελεί η μέθοδος δυναμικού παραθύρου (*Dynamic Window Approach – DWA*), όπου η αναζήτηση για τον έλεγχο του ρομπότ διεξάγεται απευθείας στον χώρο ταχυτήτων. Επομένως, η δυναμική του ρομποτικού οχήματος ενσωματώνεται στη μέθοδο μειώνοντας τον χώρο αναζήτησης στις ταχύτητες που θεωρούνται προσεγγίσιμες με βάση τις δυναμικές παραμέτρους. Η μέθοδος του δυναμικού παραθύρου θεωρεί πως το ρομπότ κινείται μόνο καμπυλωτά και σε κυκλικές πορείες και κάθε μια από αυτές τις καμπύλες προσδιορίζεται από ένα ζεύγος ταχυτήτων (v, ω), όπου v αποτελεί τη μεταφορική ταχύτητα και ω τη γωνιακή ταχύτητα. Για τη δημιουργία μιας διαδρομής προς την επιθυμητή τελική θέση, κάθε χρονική στιγμή το ρομποτικό σύστημα πρέπει να προσδιορίζει ένα ζεύγος ταχυτήτων έτσι ώστε να μην συγκρουστεί με κάποιο εμπόδιο. Τα εμπόδια που υπάρχουν στον χώρο επιβάλλουν τη δημιουργία περιορισμών στις ταχύτητες και όσες έχουν την ικανότητα να σταματήσουν το ρομποτικό όχημα πριν συγκρουστούν με αυτά θεωρούνται αποδεκτές ταχύτητες. Ο χώρος αναζήτησης μειώνεται στο δυναμικό παράθυρο (*dynamic window*), με σκοπό να ληφθούν υπόψη και οι επιταχύνσεις που μπορούν να προκύψουν και περιέχονται σε αυτόν μόνο οι ταχύτητες που μπορούν να προκύψουν μέσα στο επόμενο χρονικό διάστημα, όπως φαίνεται στην Εικόνα 31. Εάν t είναι ο χρόνος που εφαρμόζονται οι επιταχύνσεις \dot{v} και $\dot{\omega}$, τότε το δυναμικό παράθυρο ορίζεται ως:

$$V_d = \{(v, \omega) \mid v \in [v_\alpha - \dot{v} \cdot t, v_\alpha + \dot{v} \cdot t], \omega \in [\omega_\alpha - \dot{\omega} \cdot t, \omega_\alpha + \dot{\omega} \cdot t]\}$$

Έπειτα, μπορεί να προσδιοριστεί ο χώρος αναζήτησης V_r μέσα στο δυναμικό παράθυρο:

$V_r = V_s \cap V_\alpha \cap V_d$, όπου V_s αποτελεί ο χώρος των πιθανών ταχυτήτων και V_α ένα ζεύγος ταχυτήτων (v, ω), που επιτρέπει στο ρομπότ να σταματήσει χωρίς να συγκρουστεί με κάποιο εμπόδιο.

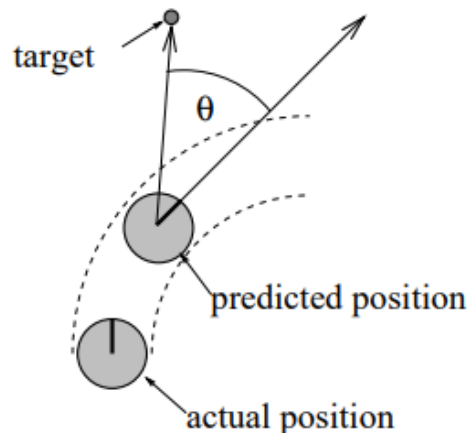


Εικόνα 31 Δυναμικό παράθυρο [18]

Μετά από τον προσδιορισμό του χώρου αναζήτησης ένα ζεύγος ταχυτήτων επιλέγεται από αυτόν με σκοπό να συμπεριληφθούν τα κριτήρια της προόδου (*heading*), της απόστασης (*distance*) και της ταχύτητας (*velocity*), τα οποία υπολογίζονται από την αντικειμενική συνάρτηση:

$$G(v, \omega) = \sigma(a \cdot heading(v, \omega) + b \cdot velocity(v, \omega) + c \cdot dist(v, \omega))$$

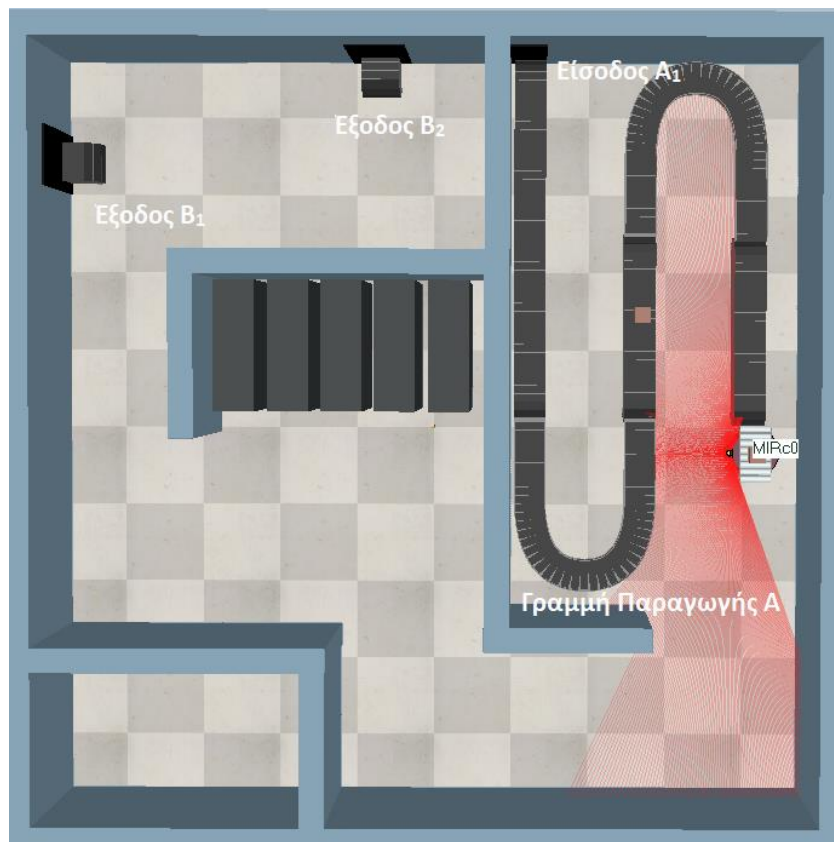
Η συνάρτηση της προόδου $heading(v, \omega)$ μετράει την ευθύγραμμη απόσταση του ρομποτικού οχήματος προς τον στόχο και δίνεται από τη σχέση $180 - \theta$, όπου θ είναι η γωνία του στόχου σε σχέση την κατεύθυνση της πορείας του ρομποτικού οχήματος, όπως φαίνεται και στην Εικόνα 32. Δεδομένου ότι η κατεύθυνση αλλάζει διαφοροποιώντας τις ταχύτητες, η γωνία θ υπολογίζεται για διατάξεις του ρομπότ που έχουν προβλεφθεί. Για τον προσδιορισμό της προβλεπόμενης διάταξης θεωρείται πως το ρομπότ κινείται με την επιλεγμένη ταχύτητα κατά τη διάρκεια του επόμενου χρονικού διαστήματος. Έπειτα, η συνάρτηση της απόστασης $dist(v, \omega)$ παρουσιάζει την απόσταση προς το κοντινότερο εμπόδιο που διασταυρώνεται με την καμπύλη της διαδρομής που ακολουθεί το ρομπότ. Τέλος, η συνάρτηση της ταχύτητας $velocity(v, \omega)$ χρησιμοποιείται για να εκτιμηθεί η πρόοδος του ρομποτικού οχήματος στην αντίστοιχη διαδρομή που ακολουθεί.



Εικόνα 32 Γωνία θ της συνάρτησης $heading(v, \omega)$ του ρομποτικού οχήματος [34]

Κεφάλαιο 3 – Λογισμικό Προσομοίωσης

Στο Κεφάλαιο 3 παρουσιάζονται τα πειράματα που πραγματοποιήθηκαν στο ρομποτικό όχημα MIRc, που σχεδιάστηκε με τη βοήθεια του προγράμματος προσομοίωσης ρομπότ CoppeliaSim, προκειμένου να εξεταστούν τα αποτελέσματα ενός πιθανού σεναρίου πλοήγησης. Το βασικό σενάριο πραγματοποιείται σε ένα διαμορφωμένο βιομηχανικό χώρο, όπως παρουσιάζεται στην Εικόνα 33, που αποτελείται από μια γραμμή παραγωγής A, τον χώρο πλοήγησης, ένα ανοιχτό δωμάτιο, ένα κλειστό δωμάτιο, μια είσοδο A₁ και δύο εξόδους B₁ και B₂, που οδηγούν σε μια γραμμή παραγωγής B. Η αποστολή που καλείται να διεκπεραιώσει το ρομποτικό όχημα αποτελεί τη διαμετακόμιση ενός κιβωτίου από τη γραμμή παραγωγής A προς τη γραμμή παραγωγής B και η έξοδος που επιλέγεται συγκεκριμένα να τοποθετηθεί το κιβώτιο αποτελεί η B₂. Έπειτα ο έλεγχος και ο προγραμματισμός της πλοήγησης του ρομπότ γίνεται μέσα από το πρόγραμμα Matlab.

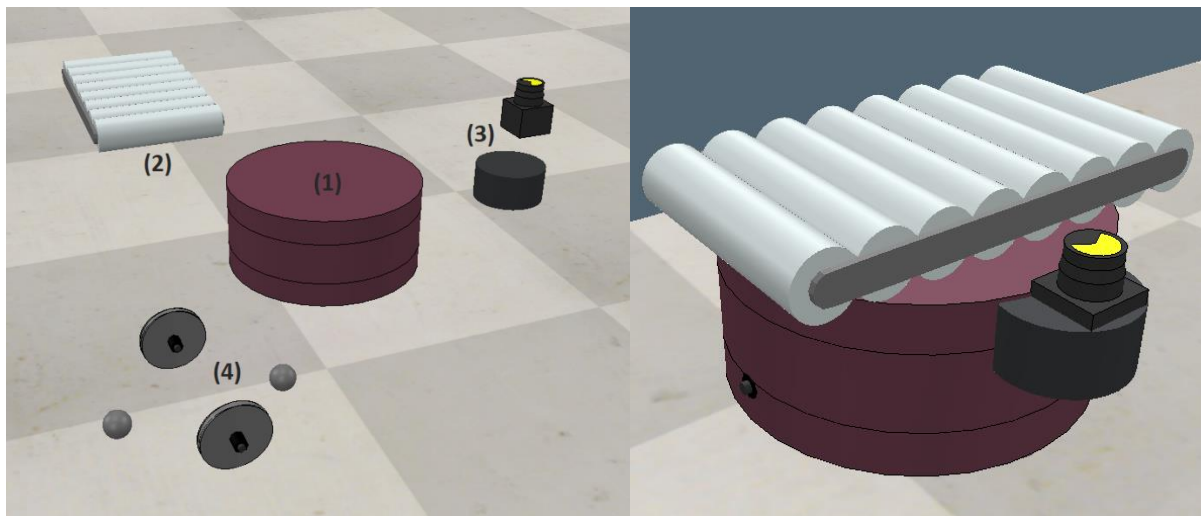


Εικόνα 33 Χώρος πλοήγησης του ρομποτικού οχήματος MIRc στο περιβάλλον προσομοίωσης CoppeliaSim.

Το σενάριο που ακολουθείται αποτελείται από τρία βήματα. Το πρώτο βήμα αφορά την παραλαβή του κιβωτίου, το οποίο τελείωσε το στάδιο επεξεργασίας του στην γραμμή παραγωγής A, από το σημείο 1 (θέση εκκίνησης MIRc). Το δεύτερο βήμα αποτελεί την εύρεση της συντομότερης διαδρομής για την πλοήγηση του ρομπότ, από το σημείο 1 μέχρι το σημείο 2 που βρίσκεται μπροστά στην έξοδο B₂ (θέση στόχος MIRc), λαμβάνοντας υπόψη όλα τα στατικά εμπόδια. Τέλος, το τρίτο βήμα αποτελεί η εκτέλεση της διαδρομής και η αποφυγή απρόοπτων δυναμικών εμποδίων μέχρι την άφιξη του ρομποτικού οχήματος στο σημείο 2.

3.1 CoppeliaSim

Το CoppeliaSim, όπως προαναφέρθηκε αποτελεί ένα πρόγραμμα προσομοίωσης ρομπότ, το οποίο για την παρούσα διπλωματική εργασία, χρησιμοποιήθηκε για τον σχεδιασμό του ρομποτικού οχήματος MIRc και του χώρου εργασίας του. Το ρομποτικό όχημα αποτελείται από τέσσερα βασικά μέρη, όπως φαίνεται στην Εικόνα 34α. Το πρώτο μέρος αποτελεί το κυλινδρικό πλαίσιο (1) και δημιουργήθηκε με παραμέτρους δυναμικού σώματος, ικανό να ανιχνευτεί από άλλα ρομποτικά συστήματα ή ευφυείς πράκτορες, μετρήσιμο και με τη δυνατότητα να συγκρουστεί με εμπόδια. Το δεύτερο μέρος αποτελεί έναν ιμάντα μεταφοράς (2) που χρησιμοποιείται για τη διαδικασία παραλαβής και παράδοσης ενός κιβωτίου από μια γραμμή παραγωγής προς μια άλλη και τοποθετείται στο πάνω μέρος του πλαισίου. Το τρίτο μέρος αποτελεί ο αισθητήρας απόστασης laser και η βάση του (3) που χρησιμοποιούνται για την ανίχνευση εμποδίων κατά τη διαδικασία πλοήγησης από τη θέση 1 στη θέση 2 και τοποθετείται στο μπροστινό μέρος του πλαισίου. Το τέταρτο μέρος αποτελεί το σύστημα κίνησης (4) που χωρίζεται στους δύο κινητήρες και τους δύο τροχούς και στα δύο castor wheels. Το σύστημα κίνησης τοποθετείται στο κάτω μέρος του πλαισίου. Συνδυάζοντας αυτά τα τέσσερα τμήματα προκύπτει η συνολική κατασκευή του ρομποτικού οχήματος MIRc, όπως φαίνεται στην Εικόνα 34β.



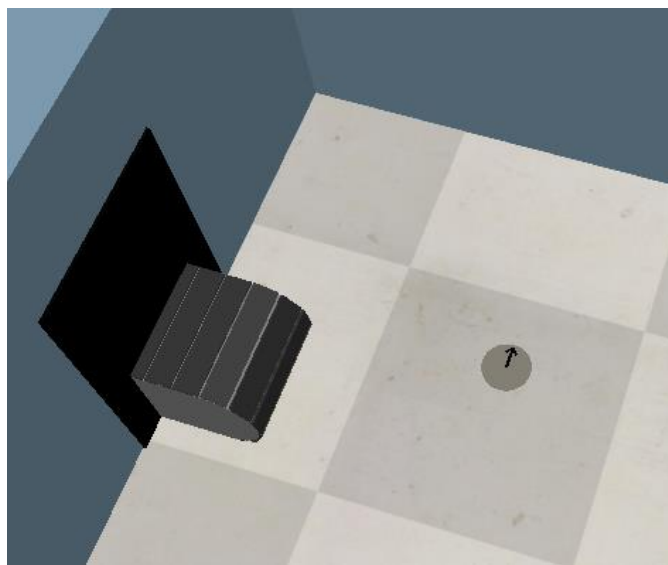
Εικόνα 34 α) Μέρη ρομποτικού οχήματος MIRc, β) Ρομποτικό όχημα MIRc

Η δημιουργία ετικετών για οποιοδήποτε αντικείμενο μέσα στον χώρο προσομοίωσης δημιουργείται στο πρόγραμμα CoppeliaSim, το οποίο χρησιμοποιεί τη γλώσσα προγραμματισμού Lua. Για να μπορέσουν τα δύο προγράμματα να λειτουργήσουν ταυτόχρονα ένα από τα πρώτα στάδια προγραμματισμού, αποτελεί ο ορισμός των ετικετών για τα εξαρτήματα που βρίσκονται πάνω στο MIRc και η δημιουργία μεταβλητών για τις διάφορους παραμέτρους, όπως μια ενδεικτική αρχική τιμή για την πόζα του ρομπότ στο σημείο 2, η διάμετρος του πλαισίου, η απόσταση μεταξύ των τροχών και η διάμετρος τους ή τα δεδομένα από τον αισθητήρα απόστασης laser. Για παράδειγμα, για να σταλθούν δεδομένα σχετικά με τη διάμετρο του πλαισίου δημιουργείται πρώτα μια ετικέτα για το ρομπότ χρησιμοποιώντας την εντολή `sim.getObjectHandle(όνομα_αντικειμένου)`, η οποία δίνει την ετικέτα `RobotHandle` στο ρομπότ. Έπειτα, ορίζονται οι παράμετροι για το πλαίσιο χρησιμοποιώντας την εντολή `sim.getFloatParameter(ετικέτα_αντικειμένου, τιμή)`, η

οποία χρησιμοποιείται δύο φορές για να δώσει την ακριβή θέση του πλαισίου μέσα στον χώρο. Η τιμή 15 χρησιμοποιείται για να προσδιοριστεί συγκεκριμένα η ελάχιστη τιμή στον άξονα x σχετικά με το πλαίσιο του αντικειμένου, που στην προκειμένη περίπτωση είναι το πλαίσιο του ρομποτικού οχήματος. Αντίστοιχα, η τιμή 18 χρησιμοποιείται για τον προσδιορισμό της μέγιστης τιμής στον άξονα x για το πλαίσιο του MIRc. Μετά τον υπολογισμό της διαμέτρου και την προσθήκη του αποτελέσματος σε μια μεταβλητή, το αποτέλεσμα δημοσιεύεται στο πρόγραμμα Matlab χρησιμοποιώντας την εντολή `sim.setFloatSignal('όνομα_δημοσίευσης'..signalSuffix, μεταβλητή_αποτελέσματος)`.

```
-- Set Robot's handle:  
RobotHandle=sim.getObjectHandle('MIRc')  
  
-- Get Robot's diameter:  
res,minX=sim.getObjectFloatParameter(RobotHandle,15)  
res,maxX=sim.getObjectFloatParameter(RobotHandle,18)  
RobotDiameter=maxX-minX  
  
-- "Publish" the diameter:  
sim.setFloatSignal ('Robot_bodyDiameter'..signalSuffix, RobotDiameter)
```

Ένα ακόμα παράδειγμα αποτελεί ο ορισμός ενδεικτικών τιμών για την πόζα του ρομποτικού οχήματος στη θέση 2. Κατά την έναρξη λειτουργίας τους προγράμματος εμφανίζεται στο περιβάλλον προσομοίωσης μια ενδεικτική θέση σχετικά με τις συντεταγμένες και τον προσανατολισμό του ρομποτικού οχήματος κατά την άφιξή του στο σημείο 2. Φυσικά δίνεται η δυνατότητα επιλογής του σημείου και η μετακίνησή του σε διαφορετική θέση κατά τη διάρκεια της προσομοίωσης, όπως για παράδειγμα μπροστά στην έξοδο 1 ή σε κάποια άλλη θέση μέσα στον χώρο πλοήγησης. Για το σενάριο που παρουσιάζεται στην συγκεκριμένη εργασία το ενδεικτικό σημείο δίνεται μπροστά στην έξοδο 2, όπως φαίνεται στην Εικόνα 35.



Εικόνα 35 Εμφάνιση ενδεικτικής θέσης για την επιλογή του σημείου 2

Επομένως, για την εμφάνιση της πόζας είναι απαραίτητο να δημιουργηθεί αρχικά η ετικέτα που θα προσδιορίζει τη θέση 2, επαναλαμβάνοντας την διαδικασία, όπως προηγουμένως. Οπότε δημιουργείται η ετικέτα *TargetHandle* για την τελική θέση του ρομπότ. Για τη συνέχεια, είναι σημαντικό να προστεθεί πως το αντικείμενο που παρουσιάζει τη πόζα για τη θέση 2 δημιουργείται έχοντας ως γονέα το ρομποτικό όχημα και οι συντεταγμένες του {x, y, z} συμπίπτουν με αυτές που έχει το ρομποτικό όχημα στην αρχική του θέση. Επομένως, για να μπορέσει να αποκτήσει τις δικές του συντεταγμένες θα πρέπει να αποσυνδεθεί από τον τρέχοντα γονέα, το οποίο γίνεται μέσω της εντολής *sim.setObjectParent(ετικέτα_αντικείμενου, ετικέτα_γονέα, τιμή)*. Η τιμή *-1* στην ετικέτα που ορίζει τον γονέα χρησιμοποιείται για να δηλώσει πως το συγκεκριμένο αντικείμενο είναι μεμονωμένο, χωρίς γονέα. Επίσης, η τιμή *true* χρησιμοποιείται με σκοπό να διατηρηθεί η τιμή της συντεταγμένης z. Έπειτα, γίνεται η αποθήκευση των αρχικών συντεταγμένων σε έναν πίνακα τριών θέσεων h, χρησιμοποιώντας την εντολή *sim.getObjectPosition(ετικέτα_αντικείμενου, τιμή)*, όπου η τιμή *-1* δηλώνει την απόλυτη θέση του αντικείμενου. Ο ορισμός της καινούριας ενδεικτικής θέσης του ρομποτικού οχήματος γίνεται με τη χρήση της εντολής *sim.setObjectPosition(ετικέτα_αντικείμενου, τιμή, συντεταγμένες)*, όπου η τιμή *-1* δηλώνει την απόλυτη θέση και οι συντεταγμένες ορίζονται ως {1.75, 0.155, h[19]}. Τέλος, μπορεί να οριστεί η πόζα του σημείου 2 ορίζοντας αρχικά σε μια μεταβλητή p τις συντεταγμένες του αντικείμενου και σε μια μεταβλητή o τις γωνίες Euler {alpha, beta, gamma} του αντικείμενου, χρησιμοποιώντας την εντολή *sim.getObjectOrientation(ετικέτα_αντικείμενου, τιμή)*, όπου η τιμή *-1* δηλώνει τον απόλυτο προσανατολισμό. Η πόζα ορίζεται σε μια μεταβλητή *TargetPose* η οποία αποτελείται από έναν πίνακα τριών θέσεων {x, y, gamma}. Για τη δημοσίευση της πόζας στο Matlab πρέπει πρώτα να τοποθετηθεί ο πίνακας που υπάρχει στη μεταβλητή *TargetPose*, σε μια μεταβλητή string, που γίνεται μέσω της εντολής *sim.packFloatTable(όνομα_μεταβλητής)* και στη συνέχεια, η μεταβλητή αυτή δημοσιεύεται μέσω της εντολής *sim.setStringSignal(όνομα_δημοσίευσης'..signalSuffix, όνομα_μεταβλητής)*.

```
-- Set Position 1 & 2 handles:
InitialPositionHandle=sim.getObjectHandle('InitialPosition')
TargetHandle=sim.getObjectHandle('Target')

-- Set indicative coordinates:
sim.setObjectParent(TargetHandle,-1,true)
h=sim.getObjectPosition(TargetHandle,-1)
sim.setObjectPosition(TargetHandle,-1,{1.75,0.155,h[19]})

-- Get the absolute target's pose:
p=sim.getObjectPosition(TargetHandle,-1)
o=sim.getObjectOrientation(TargetHandle,-1)
TargetPose={p[1],p[2],o[19]} -- {x, y, gamma}

-- "Publish" the target's pose:
packed=sim.packFloatTable(TargetPose)
sim.setStringSignal('Robot_targetPose'..signalSuffix,packed)
```

3.2 Matlab

Το Matlab αποτελεί μια πλατφόρμα προγραμματισμού, η οποία χρησιμοποιεί τη γλώσσα προγραμματισμού Matlab και χρησιμοποιείται για τον συνολικό σχεδιασμό της κίνησης του ρομποτικού συστήματος της παρούσας εργασίας. Η σχεδίαση του συστήματος αρχίζει με τη δημιουργία κατάλληλων συνδέσεων μεταξύ των δύο προγραμμάτων, Matlab και CoppeliaSim και τελειώνει με τον σχεδιασμό μιας ολοκληρωμένης λύσης για την αυτόνομη πλοήγηση του MIRc από ένα σημείο A προς ένα σημείο B και με την παρουσίασή της στο περιβάλλον προσομοίωσης. Ο σκοπός στη χρήση του Matlab αποτελεί τη δημιουργία ενός συστήματος που θα διεκπεραιώσει το σενάριο που επινοήθηκε και προαναφέρθηκε στην ενότητα 3.1. Συγκεκριμένα, το προγραμματιστικό κομμάτι της εφαρμογής είναι υπεύθυνο για την πλοήγηση του ρομπότ και την κατασκευή ενός αυτόνομου ρομποτικού οχήματος, καθώς συνδυάζει τους συντελεστές που παρέχουν κίνηση και τις μεθόδους τεχνητής νοημοσύνης που προσφέρουν την αυτονομία στο σύστημα. Παρακάτω παρατίθεται αναλυτικά η επεξήγηση του βασικού κώδικα προγραμματισμού, με τον οποίο κατασκευάζεται η διαδρομή πλοήγησης, του ρομποτικού οχήματος MIRc. Ο κώδικας χωρίζεται σε τρία μέρη, προκειμένου να δοθούν οι απαραίτητες διευκρινήσεις σχετικά με τις συναρτήσεις που χρησιμοποιήθηκαν.

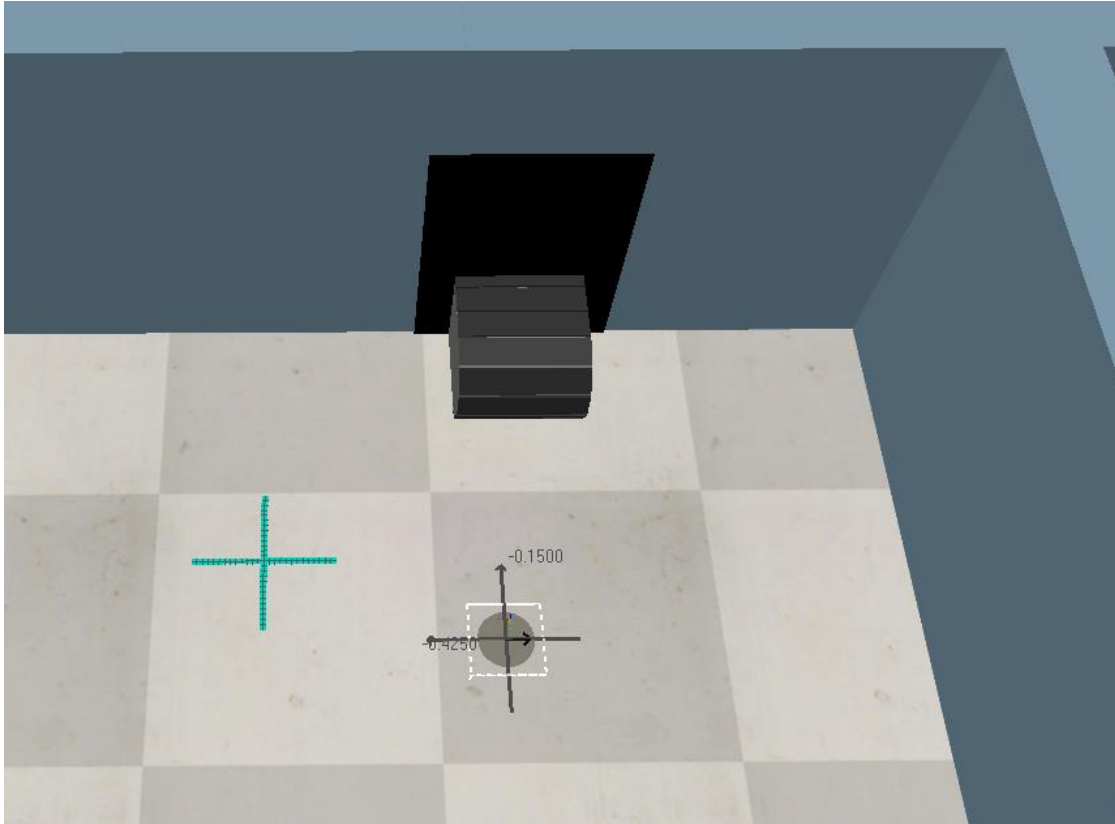
Μέρος 1^ο – Δημιουργία σύνδεσης:

Ξεκινώντας με τον προγραμματισμό του ρομποτικού συστήματος το πρώτο μέρος αποτελεί η σύνδεση του προγράμματος Matlab με το CoppeliaSim, το οποίο απαιτεί μια σειρά από εντολές συναρτήσεων. Η συνάρτηση *function [connection] = simulation_setup()* προσθέτει τις κατάλληλες βιβλιοθήκες, ανάλογα με το λειτουργικό σύστημα του υπολογιστή, δηλαδή αν τρέχει σε λογισμικό windows, mac ή linux. Έπειτα, δημιουργείται η σύνδεση των δύο λογισμικών μέσω της συνάρτησης *function [connection] = simulation_openConnection(connection, robotNb)*, η οποία ανοίγει την πόρτα 19997, που αφορά συγκεκριμένα την επικοινωνία ανάμεσα στο πρόγραμμα CoppeliaSim και τις εξωτερικές εφαρμογές. Οι συνδέσεις αυτές γίνονται μέσω API (Application Programming Interface), γνωστή ως Διασύνδεση Προγραμματισμού Εφαρμογών που παρέχει το λειτουργικό σύστημα και τις βιβλιοθήκες, προκειμένου να επιτραπεί η ανταλλαγή δεδομένων μεταξύ των προγραμμάτων [37].

```
% Initialize connection with CoppeliaSim:
connection = simulation_setup();
connection = simulation_openConnection(connection, 0);
```

Μέρος 2^ο – Εισαγωγή δεδομένων:

Αφού ολοκληρωθούν οι αρχικές ρυθμίσεις γίνεται η εισαγωγή δεδομένων από το CoppeliaSim στο Matlab. Αυτά τα δεδομένα προωθούνται από το πρόγραμμα CoppeliaSim και λαμβάνονται στο Matlab ύστερα από αίτηση μέσω remoteApi. Οι αιτήσεις γίνονται μέσα από τη χρήση συναρτήσεων που παραθέτουν τις κατάλληλες εντολές με σκοπό την ανταλλαγή συγκεκριμένων δεδομένων. Μέσα από τη συνάρτηση *function [bodyDiameter, wheelDiameter, interWheelDist, scannerPose] = robot_initialize(connection)* ανταλλάσσονται κάποια βασικά δεδομένα, όπως οι



Εικόνα 37 Μετακίνηση της θέσης στόχου μέσα στον χώρο προσομοίωσης

Απαραίτητα δεδομένα για τη διεκπεραίωση της αναζήτησης μονοπατιών αποτελούν οι πόζες των θέσεων 1 και 2 αφού οριστούν στο περιβάλλον προσομοίωσης. Μέσα από τη συνάρτηση **function [x, y, gamma] = get_TargetPose(connection)**, λαμβάνονται δεδομένα σχετικά με την πόζα του σημείου 2 και στη συνέχεια επιστρέφονται οι τιμές x και y, που είναι η τετμημένη και η τεταγμένη της θέσης 2, αντίστοιχα. Επιπλέον, η τιμή gamma αποτελεί την περιστροφή του σημείου γύρω από τον άξονα z, με μονάδα μέτρησης rad. Με τον ίδιο τρόπο, μέσα από τη συνάρτηση **function [x, y, gamma] = get_RobotPose(connection)**, λαμβάνονται δεδομένα σχετικά με την πόζα του σημείου 1 και έπειτα επιστρέφονται οι τιμές x, y και gamma.

Τέλος, για να μπορεί να πλοηγηθεί το ρομποτικό όχημα είναι απαραίτητος και ο απόλυτος χάρτης (*global map*), για τον οποίο η ανταλλαγή δεδομένων γίνεται μέσα από τη συνάρτηση **function [map] = createMap(origin, resolution, data)**. Η συγκεκριμένη συνάρτηση λαμβάνει δεδομένα για τον χάρτη από το CoppeliaSim σχετικά με την αρχή των αξόνων (*origin*), που ορίζεται η θέση της πάνω δεξιάς γωνίας του χάρτη και έχει μονάδα μέτρησης το μέτρο, την ανάλυση (*resolution*), που είναι το μέγεθος ενός κελιού (*cell*) σε μέτρα και τα δεδομένα (*data*) που αντλούνται από τον αποθηκευμένο χάρτη που βρίσκεται στο ρομποτικό σύστημα. Στη συνέχεια, η συνάρτηση επιστρέφει τον χάρτη στο βασικό πρόγραμμα, ο οποίος δίνεται σε καρτεσιανές συντεταγμένες και με ανάλυση μέτρο/κελί.


```

% Get data from CoppeliaSim:
robot_initialize(connection);
mapSensor = getMap(connection);
imshow(mapSensor);

% Get poses from CoppeliaSim:
set_TargetVisible(connection, true);
[goalPosition.x, goalPosition.y, dummy] = get_TargetPose(connection);
[robotState.x, robotState.y, robotState.heading] = get_RobotPose(connection);

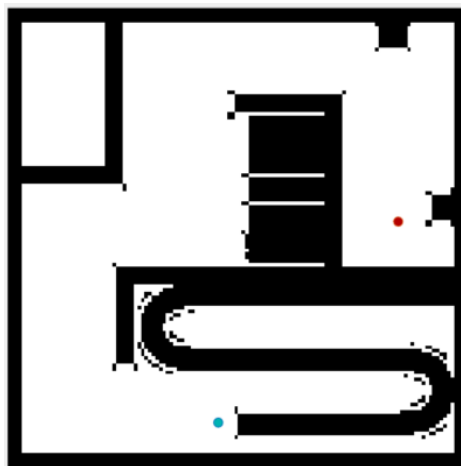
% Get global map:
globalMap = createMap(mapOrigin, mapResolution, globalMapImgSubsaped);

```

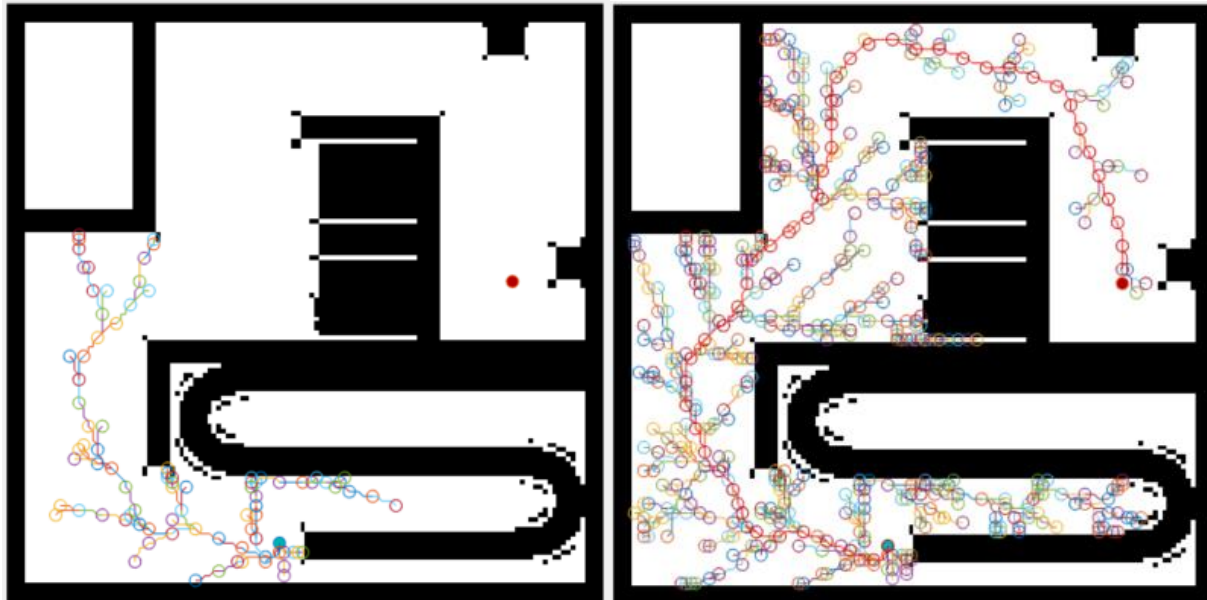
Μέρος 3^ο – Αυτόνομη πλοήγηση:

Το τρίτο μέρος του κώδικα προγραμματισμού αποτελείται από τη διαδικασία προγραμματισμού της αυτόνομης πλοήγησης που θα οδηγήσει το MIRC από τη θέση 1 στη θέση 2 αποφεύγοντας στατικά και δυναμικά εμπόδια. Για την διεκπεραίωση του σεναρίου χρησιμοποιείται ο αλγόριθμος Dijkstra σε συνδυασμό με τη μέθοδο του δυναμικού παραθύρου.

Με σκοπό την επίδειξη μιας ακόμα μεθόδου για την αναζήτηση μονοπατιών επιλέχθηκε η παρουσίαση του αλγορίθμου RRT, ο οποίος διαμορφώνει τη διαδρομή μέσα από τη συνάρτηση *function [nodes, parents, solution] = rrt(map, goalIdx, parameters, startIdx)*. Η συγκεκριμένη συνάρτηση λαμβάνει δεδομένα σχετικά με τον απόλυτο χάρτη (map), τη θέση 2 πάνω στον απόλυτο χάρτη (goalIdx), τις παραμέτρους κατά την αναζήτηση μονοπατιού (parameters), όπως το μήκος μεταξύ κάθε κόμβου και τη τρέχουσα θέση (θέση 1) πάνω στον απόλυτο χάρτη (startIdx). Στη συνέχεια, αφού τρέξει ο αλγόριθμος επιστρέφονται πίσω στον βασικό κώδικα πληροφορίες αναφορικά με κόμβους (nodes), γονείς (parents) και τη λύση (solution) που βρέθηκε από τον RRT αλγόριθμο. Η διαδικασία αναζήτησης μονοπατιού επιστρέφεται από τη συνάρτηση σε μορφή εικόνας, όπως φαίνεται στις Εικόνα 38 και Εικόνα 39α,β.



Εικόνα 38 Θέση έναρξης (μπλε) και στόχου (κόκκινο)



Εικόνα 39 α) Διαδικασία έναρξης αναζήτησης μονοπατιού από τη θέση εκκίνησης ως τη θέση στόχος, β) Ολοκλήρωση διαδικασίας αναζήτησης και παρουσίαση λύσης του συντομότερου μονοπατιού

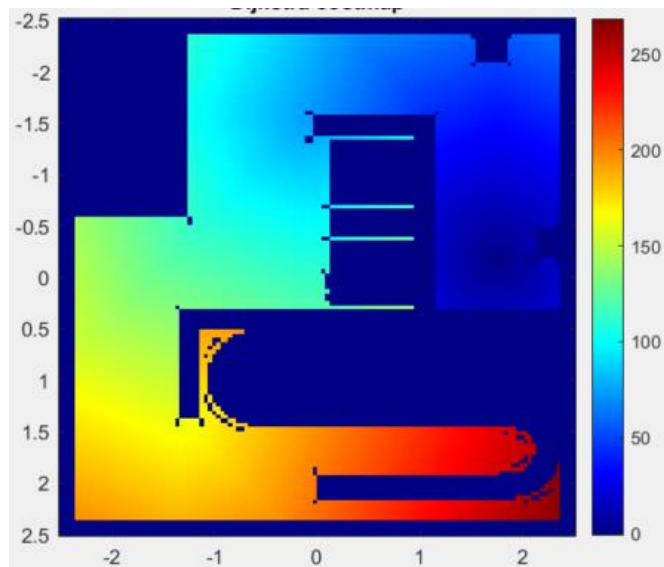
Μετά την παρουσίαση του αλγόριθμου RRT ακολουθεί ο προγραμματισμός της αυτόνομης πλοήγησης του ρομποτικού οχήματος. Το πρώτο βήμα της διαδικασίας αποτελεί ο καθορισμός των παραμέτρων για τον αλγόριθμο Dijkstra και τη μέθοδο DWA. Οι παράμετροι αποτελούν βασικά στοιχεία για τον έλεγχο της πλοήγησης του ρομπότ, καθώς αποτελούν τα αρχικά στοιχεία που δέχεται το ρομποτικό σύστημα. Στη συγκεκριμένη εργασία χρησιμοποιούνται οι παρακάτω παράμετροι:

- Ο αριθμός των δειγμάτων των γραμμικών και γωνιακών ταχυτήτων που αντλούνται κατά την πλοήγηση του ρομποτικού οχήματος.
- Οι συντελεστές που ορίζονται για τη δημιουργία της συνάρτησης κόστους σχετικά με τα βάρη της κατεύθυνσης του οχήματος, την ταχύτητα και την απόσταση από τον στόχο.
- Οι μέγιστες τιμές για τη γραμμική και γωνιακή ταχύτητα και γραμμική και γωνιακή επιτάχυνση.
- Η απόσταση από τον στόχο μόλις το ρομποτικό όχημα ξεκινάει να επιβραδύνει.
- Η τιμή ενός επιπρόσθετου κενού διαστήματος που ορίζεται γύρω από το πλαίσιο του ρομποτικού οχήματος για λόγους ασφαλείας.
- Η συνολική ακτίνα του ρομποτικού οχήματος.

Σε συνδυασμό με τις τιμές των παραμέτρων είναι απαραίτητος ο υπολογισμός ορισμένων επιπρόσθετων μεγεθών του ρομποτικού οχήματος. Μέσα από τη συνάρτηση *function* `[bodyDiameter] = get_BodyDiameter(connection)`, λαμβάνονται δεδομένα σχετικά με διάμετρο του πλαισίου του ρομποτικού οχήματος και στη συνέχεια αυτή η τιμή επιστρέφεται στον βασικό κώδικα και αποθηκεύεται στην μεταβλητή *bodyDiameter*. Μέσα από τη συνάρτηση *function* `[wheelDiameter] = get_WheelDiameter(connection)`, λαμβάνονται δεδομένα σχετικά με τη διάμετρο των τροχών του ρομπότ, τα οποία επιστρέφονται στον βασικό κώδικα και αποθηκεύονται στην μεταβλητή *wheelDiameter*. Τέλος, μέσα από τη συνάρτηση *function* `[interWheelDist] =`

get_InterWheelDistance(connection), λαμβάνονται δεδομένα σχετικά με την εσωτερική διάμετρο των τροχών του ρομποτικού οχήματος και στη συνέχεια η τιμή επιστρέφεται στον βασικό κώδικα και αποθηκεύεται στη μεταβλητή *interWheelDistance*.

Το δεύτερο βήμα της διαδικασίας είναι ο προγραμματισμός της κατασκευής του μονοπατιού με τη χρήση του αλγόριθμου Dijkstra. Για να τρέξει ο αλγόριθμος χρησιμοποιείται η συνάρτηση **function [costs, costGradientDirection, path] = dijkstra(map, goalIdx, parameters, startIdx)**, η οποία λαμβάνει τα αντίστοιχα δεδομένα με τον αλγόριθμο RRT, δηλαδή τον απόλυτο χάρτη (map), τη θέση 2 πάνω στον απόλυτο χάρτη (goalIdx), τις παραμέτρους κατά την αναζήτηση μονοπατιού (parameters) και την τρέχουσα θέση (θέση 1) πάνω στον απόλυτο χάρτη (startIdx). Έπειτα, αφού τρέξει ο αλγόριθμος επιστρέφονται πίσω στον βασικό κώδικα πληροφορίες σχετικά με τα κόστη (costs) κάθε σημείου μέσα στον απόλυτο χάρτη, τα συνολικά κόστη (costGradientDirection) από κάθε σημείο του χάρτη μέχρι την τελική θέση και το μονοπάτι με το μικρότερο συνολικό κόστος. Αξιοποιώντας αυτές τις πληροφορίες, το πρόγραμμα δημιουργεί έναν χάρτη αποστάσεων (Dijkstra cost map), τον οποίο εμφανίζει σε ένα δεύτερο παράθυρο του προγράμματος και στον οποίο παρουσιάζεται, με χρωματική διακύμανση, η απόσταση οποιουδήποτε σημείου του χάρτη σε σχέση με τη θέση εκκίνησης. Συγκεκριμένα, η κάτω αριστερά γωνία, όπως φαίνεται στην Εικόνα 40, χρωματίζεται με σκούρο κόκκινο, καθώς έχει την μεγαλύτερη απόσταση από τον στόχο, ενώ το σημείο του στόχου χρωματίζεται με βαθύ μπλε για να χαρακτηρίσει την περιοχή, ως θέση με απόσταση 0 από τον στόχο. Ακόμα, όλα τα εμπόδια που βρίσκονται στον χώρο χρωματίζονται με διαφορετικό χρώμα προκειμένου να χαρακτηριστούν ως μη προσεγγίσιμα σημεία για το ρομπότ.

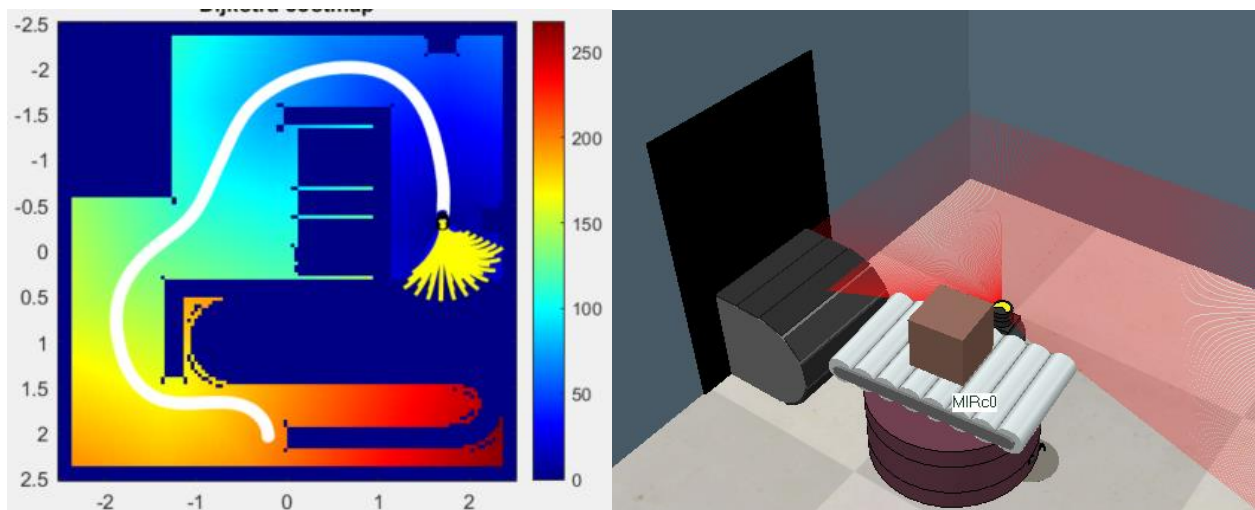


Εικόνα 40 Χάρτης αποστάσεων αλγόριθμου Dijkstra (Dijkstra's cost map)

Το τρίτο βήμα της διαδικασίας αποτελεί η εφαρμογή του DWA, που προαπαιτεί τον καθορισμό των ταχυτήτων. Μέσα από τη συνάρτηση **function [leftVelRadPerSec, rightVelRadPerSec] = get_WheelSpeeds(connection)**, λαμβάνονται δεδομένα σχετικά με τις ταχύτητες των τροχών και στη συνέχεια επιστρέφονται στο βασικό πρόγραμμα, οι τιμές των ταχυτήτων του αριστερού (wheelSpeedLeftRadPerSec) και του δεξιού τροχού (wheelSpeedRightRadPerSec), με μονάδες

μέτρησης rad/s. Έπειτα, για να τρέξει η μέθοδος DWA καλείται η συνάρτηση **function** `[vSolution, omegaSolution, debug] = DynamicWindowApproach(robotState, goalPosition, localMap, parameters, globalGradientMap)`, η οποία λαμβάνει δεδομένα σχετικά με την κατάσταση (robotState) του ρομποτικού οχήματος, όπως είναι οι συντεταγμένες x, y, ο προσανατολισμός και η γραμμική και γωνιακή ταχύτητα. Επίσης, δέχεται δεδομένα για τις συντεταγμένες της θέσης 2 (goalPosition), τον σχετικό χάρτη (localMap) του χώρου πλοήγησης, τις προκαθορισμένες παραμέτρους και τον απόλυτο χάρτη αποστάσεων (globalGradientMap) που χρησιμοποιείται για την πλοήγηση. Στη συνέχεια, η συνάρτηση επιστρέφει πληροφορίες σχετικά με τη γραμμική και τη γωνιακή ταχύτητα και δίνει τη δυνατότητα ελέγχου μέσω της εξόδου debug για την αξιοπιστία των δεδομένων που εισάγονται στον αλγόριθμο.

Μέσα στη συγκεκριμένη συνάρτηση υπάρχει και η συνάρτηση **function** `trajectory = computeTrajectory(robotState, vel, omega, dt, nSteps)`, η οποία υπολογίζει μια μοναδική πορεία με n βήματα (nSteps), ανάλογα με τα δείγματα των γραμμικών (vel) και γωνιακών (omega) ταχυτήτων που εισάγονται σε αυτή. Το μονοπάτι που κατασκευάζεται επιστρέφεται στη συνάρτηση DWA με τη μορφή `trajectory = [x', y', heading', v', w', s']`; Η συγκεκριμένη έξοδος χρησιμοποιείται ως είσοδος για μια ακόμα συνάρτηση που ακολουθεί. Μέσω της συνάρτησης **function** `[headingTerm, distTerm, velTerm] = scoreTrajectory(trajectory, nStepsBrake, goalPosition, maps, distToGoal, goalBrakingDistance, parameters)` υπολογίζονται τα κόστη του μονοπατιού που δημιουργούνται καθώς εκτελείται η πλοήγηση. Σε περίπτωση που ανιχνευτεί κάποιο δυναμικό εμπόδιο η συγκεκριμένη συνάρτηση επιστρέφει συνεχώς τιμές σχετικά με τον προσανατολισμό, την απόσταση και την ταχύτητα του ρομπότ στη συνάρτηση DWA, η οποία επιλέγει εάν θα διατηρήσει το τελευταίο μονοπάτι ή θα χρειαστεί να υπολογιστεί καινούριο για την αποφυγή σύγκρουσης. Κατά τη διάρκεια της πλοήγησης το μονοπάτι που κατασκευάζει το ρομποτικό όχημα αρχίζει να σχηματίζεται στον χάρτη αποστάσεων μέχρι να φτάσει στο τελικό σημείο, όπως φαίνεται στην Εικόνα 41α. Μόλις το MIRc φτάσει στο σημείο 2 έχει ολοκληρώσει τη διαμετακόμιση του κιβωτίου και μπορεί να το περάσει στη γραμμή παραγωγής της εξόδου B₂.



Εικόνα 41 α) Πλοήγηση του MIRc με τη μέθοδο DWA, β) Παρουσίαση στιγμιότυπου από το πρόγραμμα προσομοίωσης CoppeliaSim, από την άφιξη του ρομποτικού οχήματος στην τελική θέση.

```
% RRT map:
[ ~, ~, solution] = rrt(map, goal, parameters, start);

% Define parameters for Dijkstra and DWA:
parameters.nVelSamples = 11; %number of linear velocity samples
parameters.nOmegaSamples = 11; %number of angular velocity samples
parameters.headingScoring = 0.1; %weighting factor for heading term in
%cost function
parameters.velocityScoring = 0.8; %weighting factor for velocity term
%in cost function
parameters.obstacleDistanceScoring = 0.3; %weighting factor for
%distance term in cost function
parameters.maxVel = 0.2; %maximum linear velocity of the robot [m/s]
parameters.maxOmega = pi; %maximum angular velocity of the robot
%[rad/s]
parameters.maxAcc = 1.0; %maximum linear acceleration of the robot
%[m/s2]
parameters.maxOmegaDot = pi; %maximum angular acceleration of the
% robot [rad/s2]
parameters.goalBrakingDistance = 0.5; %distance from goal at which
%robot starts to brake [m]
parameters.safetyMargin = 0.1; %safety margin due to delays and
%inaccuracies when modeling the dynamic constraints
bodyDiameter = get_BodyDiameter(connection);
parameters.robotRadius = bodyDiameter/2.0 + parameters.safetyMargin;
% radius of the robot's circular footprint [m]
wheelDiameter = get_WheelDiameter(connection);
wheelRadius = wheelDiameter/2.0;
interWheelDistance = get_InterWheelDistance(connection);

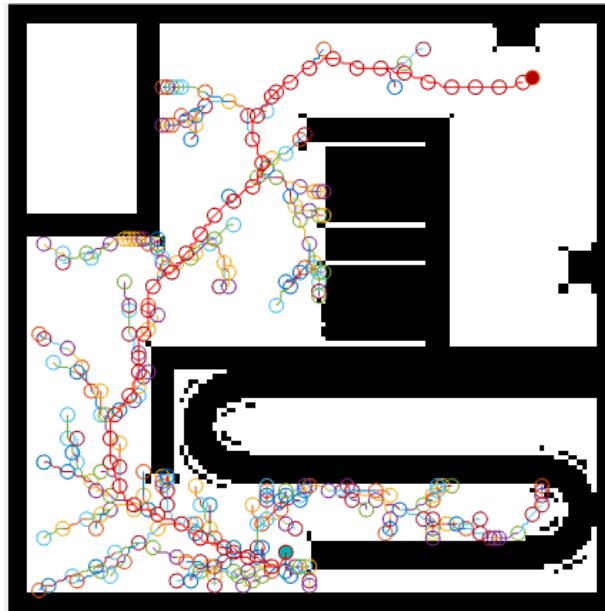
% Compute Dijkstra distance map:
[costs, costGradientDirection] = dijkstra(globalMap.data, goalIdx, pa-
rameters);

% Get the current v/w of the robot from CoppeliaSim:
[wheelSpeedLeftRadPerSec, wheelSpeedRightRadPerSec] = get_Wheel-
Speeds(connection);

% CaLL DWA:
[v, omega, debug] = DynamicWindowApproach(robotState, goalPosition,
localMap, parameters, costGradientDirectionMap);
```

Παραλλαγή αρχικού σεναρίου

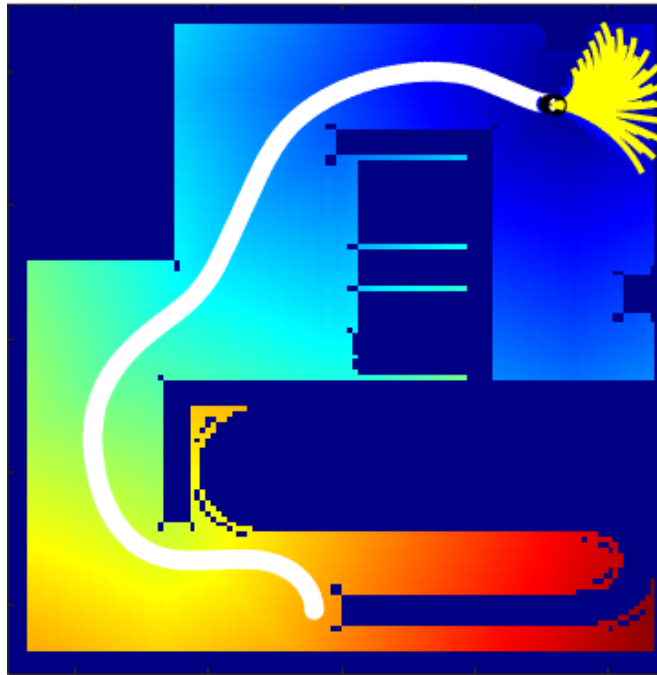
Έστω ότι θέλουμε να δούμε την περίπτωση στην οποία το ρομποτικό όχημα διαμετακομίζει ένα κιβώτιο προς την έξοδο B₁. Σε αυτήν την περίπτωση ο χρήστης επιλέγει το καινούριο τελικό σημείο και το τοποθετεί μπροστά στην έξοδο B₁. Επιπλέον, εξετάζεται και η περίπτωση στην οποία ο χώρος εργασίας περιλαμβάνει και άλλα εμπόδια τα οποία το ρομπότ καλείται να προσπεράσει δημιουργώντας ένα διαφορετικό μονοπάτι κατά την πλοήγησή του. Για την αναζήτηση μονοπατιού χρησιμοποιούνται οι αλγόριθμοι αναζήτησης, RRT και Dijkstra, με τον δεύτερο να επιλέγεται για την πλοήγηση του οχήματος στο πρόγραμμα προσομοίωσης. Αρχικά γίνεται η αναζήτηση της βέλτιστης διαδρομής σύμφωνα με τους κανόνες των δύο αλγορίθμων. Επομένως, στην Εικόνα 42 παρουσιάζεται η λύση του αλγορίθμου αναζήτησης RRT. Έπειτα, στην Εικόνα 43 παρουσιάζεται το αποτέλεσμα της πλοήγησης του ρομποτικού οχήματος MIRC, που έγινε αρχικά με την εύρεση της βέλτιστης διαδρομής, σύμφωνα με τον αλγόριθμο Dijkstra και στη συνέχεια χρησιμοποιώντας τα κόστη και τις πληροφορίες που υπολογίστηκαν σχετικά με αυτή, ώστε να πραγματοποιηθεί η πλοήγηση, χρησιμοποιώντας τη μέθοδο DWA. Η πρώτη περίπτωση, στην οποία δεν υπάρχουν επιπλέον εμπόδια στον χαρτογραφημένο χάρτη παρουσιάζεται στην Εικόνα 43, ενώ η δεύτερη περίπτωση που αφορά την πλοήγηση σε χώρο εργασίας με επιπλέον εμπόδια, παρουσιάζεται στην Εικόνα 44.



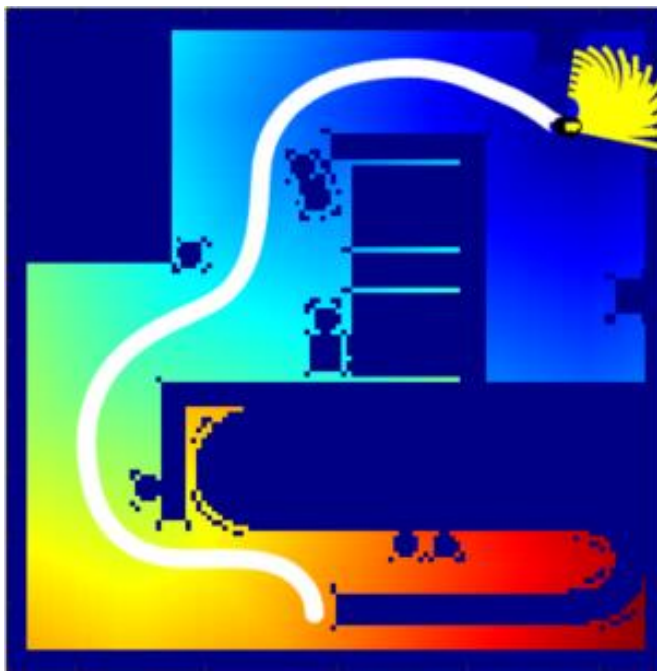
Εικόνα 42 Εύρεση μονοπατιού με τη χρήση του αλγορίθμου αναζήτησης RRT

Παρόλο που και οι δύο αλγόριθμοι χρησιμοποιούνται ευρέως για ρομποτικές εφαρμογές, με βάση την εφαρμογή που μελετάται στην παρούσα διπλωματική εργασία και σύμφωνα με τα αποτελέσματα που προκύπτουν από το λογισμικό προσομοίωσης, παρατηρείται μια βασική διαφορά μεταξύ των δύο αλγορίθμων που συντελεί στην απόρριψη μιας εκ των δύο για την πραγματοποίηση μιας αποδοτικότερης εφαρμογής. Συγκεκριμένα, ενώ ο αλγόριθμος RRT εντοπίζει μια πραγματική λύση, η χρήση της δειγματοληψίας κοντά σε εμπόδια και η εύρεση μη ομαλών μονοπατιών καθιστά τον συγκεκριμένο αλγόριθμο λιγότερο αποδοτικό για την επιλογή του σε πραγματικές εφαρμογές. Συνεπώς, η επιλογή του αλγορίθμου Dijkstra σχετίζεται με την ικανότητά του να

δημιουργήσει μια βέλτιστη διαδρομή σύμφωνα με τις ιδιαιτερότητες και τις απαιτήσεις της συγκεκριμένης εφαρμογής, καθώς υπολογίζει τα κόστη μεταξύ όλων των σημείων του χάρτη, συνυπολογίζοντας τις αποστάσεις μεταξύ του ρομποτικού οχήματος και των εμποδίων.



Εικόνα 43 Λύση του προβλήματος πλοήγησης με τη χρήση του αλγόριθμου Dijkstra και της μεθόδου DWA σε ένα χαρτογραφημένο περιβάλλον



Εικόνα 44 Λύση του προβλήματος πλοήγησης με τη χρήση του αλγόριθμου Dijkstra και της μεθόδου DWA σε ένα χαρτογραφημένο περιβάλλον με εμπόδια που προέκυψαν κατά την πλοήγηση.

Συμπεράσματα

Την τελευταία δεκαετία τα αυτόνομα ρομποτικά οχήματα έχουν αναπτυχθεί τεχνολογικά σε τεράστιο βαθμό, αποκτώντας έτσι την ικανότητα να εκτελούν αποτελεσματικά εφαρμογές σε πολλά διαφορετικά πεδία. Ένας βασικός παράγοντας αποτελεί η αξιοποίηση των μεθόδων της τεχνητής νοημοσύνης που συντελούν στην αύξηση της παραγωγικότητας σε διάφορους λειτουργικούς τομείς ενός ρομποτικού συστήματος, αλλά κυρίως στα χαρακτηριστικά που αφορούν την πλοήγησή του. Η ανάπτυξη ενός ρομποτικού συστήματος στο οποίο υπάρχει άριστη συνεργασία μεταξύ των τμημάτων, όπως το σύστημα αντίληψης, το σύστημα επεξεργασίας, το σύστημα κατανόησης και το σύστημα εκτέλεσης ενεργειών, εξυπηρετεί στην εύρεση μιας αποτελεσματικής λύσης. Η βασική προϋπόθεση, όμως, για την επιτυχημένη αυτόνομη πλοήγηση του ρομποτικού οχήματος, αποτελεί ο αλγόριθμος που θα χρησιμοποιηθεί για την κατασκευή του μονοπατιού. Σήμερα υπάρχουν πολλοί αλγόριθμοι που χρησιμοποιούνται για τη δημιουργία μιας διαδρομής, οι οποίοι μπορούν να χωριστούν σε απόλυτους και σχετικούς αλγόριθμους και να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου και για την εύρεση λύσεων σε πολύπλοκα προβλήματα. Φυσικά, ο κάθε αλγόριθμος έχει πλεονεκτήματα και μειονεκτήματα και σύμφωνα με την πειραματική διαδικασία που πραγματοποιήθηκε για την εργασία, είναι φανερό πως παρόλο που υπάρχει μεγάλος αριθμός αλγορίθμων είναι σημαντικό να χρησιμοποιηθεί μια μέθοδος που θα ταιριάζει με το συνολικό σενάριο και τον σκοπό για τον οποίο κατασκευάζεται ένα ρομπότ. Επομένως, για την ανάπτυξη και την κατασκευή ενός ρομπότ, με ρόλο τη διαμετακόμιση προϊόντων μέσα σε έναν βιομηχανικό περιβάλλον, είναι απαραίτητο η αρχιτεκτονική και το σύστημα πλοήγησής του να σχεδιαστούν λαμβάνοντας υπόψη πως σε αυτόν τον χώρο υπάρχει μεγάλη πιθανότητα να κινούνται άνθρωποι ή και άλλα ρομποτικά οχήματα, καθώς θα διαφέρει από ένα ρομπότ το οποίο κινείται μέσα σε ένα χώρο εργασίας στο οποίο δεν θα υπάρχουν μεταβολές με το πέρασ του χρόνου.

Βιβλιογραφία – Αναφορές – Διαδικτυακές Πηγές

- [1] Κουμπούλης Φ. Ν. & Μέρτζιος Β. Γ. (2002). *Εισαγωγή στη Ρομποτική*. Εκδόσεις Παπασωτηρίου.
- [2] Brady, M. (1985). Artificial intelligence and robotics. *Artificial Intelligence*, 26(1), 79-121.
- [3] Mohd J., Abid H., Ravi P. S., Rajiv S. (2021). Substantial capabilities of robotics in enhancing industry 4.0 implementation. *Cognitive Robotics*, 1(7), 58-75.
- [4] Siciliano B. & Khatib Ous. (2008). Introduction. Στο *Springer Handbook of Robotics*. (pp. 1-4).
- [5] Kortenkamp D. & Simmons R. (2008). Part A Robotics Foundations: Robotic Systems Architectures and Programming. Στο Siciliano B. & Khatib Ous. *Springer Handbook of Robotics*. (pp. 187-206).
- [6] Ben-Ari M. & Mondada F. (2018). *Elements of Robotics*. Springer.
- [7] Rubio F., Valero F., Llopis-Albert C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2).
- [8] Craig J. (2016). *Introduction to Robotics. Mechanics and Control*. 3rd ed. Pearson.
- [9] ROBOTS SAFE™ DRYFOG. Ανακτήθηκε από: <https://grobotics.eu/disinfection-robot>.
- [10] FAQs ABOUT BOSTON DYNAMICS. Ανακτήθηκε από: <https://www.bostondynamics.com/about#Q1>.
- [11] Klančar G., Zdešar A., Blažič S., Àkrjanc I. (2017). *Wheeled Mobile Robotics. From Fundamentals Towards Autonomous Systems*. Elsevier.
- [12] Gianluca A., Thor I. F., Dana R. Y. (2008). Part F Field and Service Robotics: Underwater Robotics. Στο Siciliano B. & Khatib Ous. *Springer Handbook of Robotics*. (pp. 987-1008).
- [13] Lubofsky E. (2019). Underwater robots swarm the ocean Researchers tackle a big problem for tiny ocean robots. Ανακτήθηκε από: <https://www.whoi.edu/news-insights/content/underwater-robots-swarm-the-ocean/>.
- [14] ISO 8373:2021(en) Robotics – Vocabulary. Ανακτήθηκε από: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en:sec:3.6>.
- [15] Janis A. & Modris G. (2022). Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Applied Sciences*, 12(2), 937.
- [16] Alami R., Chatila R., Fleury S., Ghallab M., Ingrand F., (1998). An architecture for autonomy. *International Journal of Robotics Research*, 17(4), 315-337.
- [17] PIONEER 3-DX. Ανακτήθηκε από: <http://cbslab.kocaeli.edu.tr/RRC-Lab/pioneer.php>.

- [18] Siegwart R., Nourbakhsh I. R., Scaramuzza D. (2011). *Introduction to Autonomous Mobile Robots*. 2nd ed. The MIT Press.
- [19] Kagan E., Shvalb N., Ben-Gal I. (2019). *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming*. Wiley.
- [20] Sandeep K. M. & Jharna M. (2014). Kinematics, Localization and Control of Differential Drive Mobile Robot. *Global Journal of Research In Engineering*.
- [21] Bräunl T. (2008). *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*. 3rd ed. Springer.
- [22] Barros Ít. & Nascimento T. P. (2021). Robotic Mobile Fulfillment Systems: A survey on recent developments and research opportunities. *Robotics and Autonomous Systems*, 137(7). DOI: 10.1016/j.robot.2021.103729.
- [23] Qiu L., Hsu W., Huang S., Wang H. (2002). Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research*, 40(3), 745-60.
- [24] Hoff E. B. & Sarker B. R. (1998). An overview of path design and dispatching methods for automated guided vehicles. *Integrated Manufacturing Systems*, 9(5), 296-307.
- [25] Murphy R. (2019). *Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. 2nd ed. The MIT Press.
- [26] Matsson F. (2018). *Sensor fusion for positioning of an autonomous vehicle. Design and implementation of an unscented Kalman filter* (Degree project in mathematics). KTH Royal Institute of technology. School of engineering sciences. Sweden.
- [27] Wolton W. (2021). *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*. 3rd ed. Pearson.
- [28] Weber H. (2018). LiDAR sensor functionality and variants. Ανακτήθηκε από: https://cdn.sick.com/media/docs/3/63/963/whitepaper_lidar_en_im0079963.pdf
- [29] Κωστόπουλος Σ. (2021). *Αρχές και μοντελοποίηση ασύρματης διάδοσης. Εφαρμογές στη σχεδίαση ασύρματων τηλεπικοινωνιακών συστημάτων*. Εκδόσεις Τζιόλα.
- [30] Omankwu, O. C., Nwagu, C. K., Inyama, H. (2017). Robotics and Artificial Intelligence: Differences and Similarities. *International Journal of Computer Science and Information Security*, 15(12).
- [31] Liu L., Lin J., Yao J., He D., Zheng J., Huang J., Shi P. (2021). Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach. *Wireless Communications & Mobile Computing*, 1-12.
- [32] Βλαχάβας Ι., Κεφαλάς Π., Βασιλειάδης Ν., Κόκκορας Φ., Σακελλαρίου Η. (2020). Τεχνητή Νοημοσύνη. Εκδόσεις Πανεπιστημίου Μακεδονίας.

- [33] Cormen T., Leiserson C., Rivest R., Stein C. 2009. *Introduction to Algorithms*. The MIT Press.
- [34] Dieter F., Wolfram B., Sebastian T. (1997). The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23-33.
- [35] MiR: Mobile industrial Robots. Ανακτήθηκε από: <https://www.mobile-industrial-robots.com/>
- [36] Roland S., Margarita Chli, Nicholas L. (2021). *Autonomous Mobile Robots - Spring 2021* [Powerpoint slides]. Ανακτήθηκε από ETHzürich, Swiss Federal Institute of Technology in Zürich, Online: https://asl.ethz.ch/education/lectures/autonomous_mobile_robots/spring-2021.html
- [37] Galar D., Kumar U., Seneviratne D. (2020). *Robots, Drones, UAVs and UGVs for Operation and Maintenance*. CRC Press.
- [38] Ρομποτικός βραχίονας σταθερής βάσης [Εικόνα 1α]. Ανακτήθηκε από: <https://www.howtorobot.com/expert-insight/robot-end-effectors>
- [39] Πλάγια όψη ρομποτικού βραχίονα σταθερής βάσης [Εικόνα 1β]. Ανακτήθηκε από: <https://www.osha.gov/otm/section-4-safety-hazards/chapter-4>
- [40] Ρομποτική σκούπα Roomba j7+, iRobot [Εικόνα 2α]. Ανακτήθηκε από: <https://www.irobot.gr/produkt/roomba-j7-2/prodiagraphes/>
- [41] Η χρήση των drone στον αγροτικό τομέα για την επισκόπηση εκτάσεων και την παρακολούθηση της συνολικής υγείας μιας καλλιέργειας [Εικόνα 4]. Ανακτήθηκε από: <https://isam.education/en/precision-agriculture-agricultural-drone/>
- [42] Μη επανδρωμένο υποβρύχιο όχημα Remus 300, Hydroid [Εικόνα 5]. Ανακτήθηκε από: <https://www.naval-technology.com/projects/remus-300-unmanned-underwater-vehicle-uuv/>
- [43] Αισθητήρας LIDAR URG-04LX, Hokuyo [Εικόνα 23]. Ανακτήθηκε από: <https://hokuyo-usa.com/products/lidar-obstacle-detection/urg-04lx>
- [44] Μαγνητικός κωδικοποιητής [Εικόνα 24α]. Ανακτήθηκε από: https://ams.com/documents/20143/36005/AS5311_DS000200_2-00.pdf
- [45] Οπτικός κωδικοποιητής [Εικόνα 24β]. Ανακτήθηκε από: <https://control.com/technical-articles/how-to-troubleshoot-encoders/>
- [46] Οπτικό γυροσκόπιο [Εικόνα 25β]. Ανακτήθηκε από: https://www.researchgate.net/publication/237111429_Qualification_of_fibre-optic_gyroscopes_for_civil_engineering_applications
- [47] Παράδειγμα δυναμικού πεδίου που χρησιμοποιείται για την πλοήγηση προς μια τελική θέση αποφεύγοντας και την αποφυγή εμποδίων [Εικόνα 26]. Ανακτήθηκε από:

<https://www.researchgate.net/publication/294645105> Dynamic Target Tracking and Obstacle Avoidance using a Drone

[48] Markov Localization [Αλγόριθμος 1]. Ανακτήθηκε από:

<https://slideplayer.com/slide/8524638/>

[49] Dijkstra Algorithm [Αλγόριθμος 3]. Ανακτήθηκε από: <https://slidetodoc.com/singlesource-shortest-paths-chapter-24-cormen-leiserson-riveststein/>

[50] RRT (Rapidly exploring Random Tree) [Αλγόριθμος 4]. Ανακτήθηκε από: <https://www.researchgate.net/publication/259680822> On the Performance of Sampling-Based Optimal Motion Planners