ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός Διαδρομής και Έλεγχος Κίνησης για Ρομποτικό Αυτοκίνητο

Διπλωματική Εργασία

Εμμανουήλ Βαρδάκης

Εξεταστική Επιτροπή: Επ. Καθ. Μιχαήλ Λαγουδάκης (επιβλέπων)
Επ. Καθ. Αντώνιος Δεληγιαννάκης
Αν. Καθ. Σταύρος Μουσταϊζής (Γενικό Τμ. Πολ. Κρήτης)

Χανιά 2009

– Παππού, παππού, θέλω να φύγουμε, πείνασα και δεν έχουμε πράμα να φάμε.
– Παιδί μου έχω δουλειά ακόμα, όμως αν θες εσύ γύρε σιγά-σιγά προς το σπίτι.
– Παππού, δεν ξέρω το δρόμο!!!!
– Ανέβα στον γάιδαρο και αυτός τον ε κατέχει…

Περίληψη

Ένα από τα βασικότερα προβλήματα στο χώρο των αυτόνομων ρομποτικών συστημάτων είναι αυτό της πλοήγησης, το οποίο περιλαμβάνει την χαρτογράφηση του περιβάλλοντα χώρου, τον εντοπισμό της τρέχουσας θέσης μέσα στο χαρτογραφημένο περιβάλλον, το σχεδιασμό ασφαλούς διαδρομής μεταξύ θέσεων, και τέλος τον έλεγχο την κίνησης για τη μετακίνηση κατά μήκος κάποιας διαδρομής. Τα τελευταία χρόνια έχει αναπτυχθεί σημαντική ερευνητική δραστηριότητα γύρω από το πρόβλημα της ρομποτικής πλοήγησης και έχει οδηγήσει σε μια πληθώρα εναλλακτικών προσεγγίσεων για διάφορες περιπτώσεις. Η παρούσα εργασία εστιάζει στην ανάπτυξη ενός συστήματος αυτόνομης πλοήγησης για το τετράτροχο ρομποτικό όχημα του Εργαστηρίου Δομής της Ύλης και Φυσικής Λέιζερ του Πολυτεχνείου Κρήτης, το οποίο έχει δυνατότητες κίνησης αντίστοιχες με αυτές ενός κοινού αυτοκινήτου (έλεγχος της γραμμικής ταχύτητας και της γωνίας στροφής). Στα πλαίσια της εργασίας αυτής αναπτύχθηκαν τα δύο βασικά επίπεδα της πλοήγησης, σχεδιασμός διαδρομής και έλεγχος κίνησης, προσαρμοσμένα στις δυνατότητες των τετράτροχων ρομποτικών συστημάτων υπό μελέτη, θεωρώντας ότι ο χάρτης του περιβάλλοντα χώρου και η θέση του οχήματος σ' αυτόν είναι διαθέσιμα. Συγκεκριμένα, για τον σχεδιασμό διαδρομής χρησιμοποιείται ένας αλγόριθμος πληροφορημένης ευριστικής αναζήτησης ο οποίος παράγει μια διακριτοποιημένη διαδρομή που εγγυάται ασφάλεια (αποφυγή συγκρούσεων με εμπόδια) και βέλτιστο κόστος (ελάχιστο μήκος διαδρομής). Η διαδρομή αυτή μεταβιβάζεται στον ελεγκτή κίνησης ο οποίος αναλαμβάνει να μετακινήσει το όχημα κατά μήκος της διαδρομής θέτοντας κατάλληλες τιμές στην ταχύτητα και στη γωνία στροφής του οχήματος. Η ταχύτητα και η γωνία στροφής αποφασίζονται δυναμικά σε πραγματικό χρόνο με γνώμονα την ακρίβεια παρακολούθησης της διαδρομής και την ομαλότητα της κίνησης λαμβάνοντας υπ' όψιν όλους τους κινηματικούς και δυναμικούς περιορισμούς της κίνησης του οχήματος. Το προτεινόμενο σύστημα πλοήγησης ανταποκρίνεται επιτυχώς σε διάφορα προσομοιωμένα σενάρια που έχουν μελετηθεί στο γραφικό περιβάλλον που αναπτύχθηκε για να χρησιμοποιηθεί ως διασύνδεση μεταξύ του χρήστη και του συστήματος πλοήγησης.

Ευχαριστίες

Πρώτα απ' όλα θα ήθελα να ευχαριστήσω τον κ. Μιχαήλ Λαγουδάκη για τη συνεχή καθοδήγηση και την ουσιαστική βοήθεια σε κάθε στάδιο της εκπόνησης της εργασίας αυτής. Θα ήθελα να ευχαριστήσω επίσης το πρόγραμμα «Φυτώριο Ιδεών» για την ευκαιρία που μου έδωσε να αναπτύξω και να υλοποιήσω το πρώτο στάδιο του συγκεκριμένου ρομποτικού συστήματος. Επίσης, θα ήθελα να ευχαριστήσω το εργαστήριο «Δομής της Ύλης και Φυσικής Λέιζερ» με διευθυντή τον κ. Σταύρο Μουσταϊζή για την ευκαιρία που μου έδωσε, την συνεργασία και τη βοήθεια στο πρώτο στάδιο της ανάπτυξης. Θα ήθελα να ευχαριστήσω επίσης τον κ. Ιωάννη Τωμαδάκη για την καθοδήγηση και τις συμβουλές του, καθώς επίσης και όλους αυτούς που βοήθησαν για την ολοκλήρωση του πρώτου σταδίου ανάπτυξης. Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην Εβίνα Αφεντάκη για την υπομονή και την στήριξη όλων αυτών των χρόνων.

Αφιερωμένο σ' όλους αυτούς, που αναζητούν τον δρόμο τους…

Περιεχόμενα

Περίληψη....3
Ευχαριστίες....5
Περιεχόμενα....7
Λίστα εικόνων....9
Κεφάλαιο 1 Εισαγωγή....11
1.1 Η πλοήγηση στη ζωή μας....11
1.2 Αναλυτική περιγραφή εργασίας....11
1.3 Διάθρωση κειμένου....13
Κεφάλαιο 2 Ρομποτικό μοντέλο....14
2.1 Ιστορική αναδρομή....14
2.2 Βασικές έννοιες....14
2.3 Περιγραφή κινητής μονάδος....15
2.3.1 Σύστημα ελέγχου διεύθυνσης....17
2.3.2 Σύστημα προώθησης....18
2.3.3 Σύστημα πέδησης....19
2.3.4 Σύστημα ενεργοποίησης/απενεργοποίησης....20
2.3.5 Κεραία ανύψωσης εικονολήπτη....21
2.3.6 Ρομποτική κεφαλή στήριξης εικονολήπτη και αισθητήρων....21
2.3.7 Ηλεκτρονικό σύστημα ελέγχου....22
2.3.8 Ελεγκτής κεφαλής και εικονολήπτη....23
2.3.9 Κεντρικό υπολογιστικό σύστημα....24
2.4 Περιγραφή κεντρικού σταθμού μετάδοσης δεδομένων....25
2.5 Διεπαφή χρήστη....26
2.5.1 Έλεγχος ρομποτικού συστήματος....26
2.5.2 Τηλεχειρισμός συστήματος εικονολήπτη....27
2.6 Εφαρμογές....28
Κεφάλαιο 3 Πλοήγηση....29
3.1 Πλοήγηση και περιβάλλον εργασίας....29
3.2 Σχεδιασμός διαδρομής....32
3.2.1 Οδικός χάρτης....32
3.2.2 Γεωμετρική προσέγγιση....32
3.2.3 Προσέγγιση διάσπασης κελιών....33
3.2.4 Ανακλαστική προσέγγιση....33
3.2.5 Εικονικά πεδία δυνάμεων....33
3.2.6 Δυναμικά πεδία....34
3.3 Έλεγχος κίνησης....34
3.3.1 Μηδενικά συστήματα....35
3.3.2 Ημιτονοειδής τροχιά δράσης....35
3.3.3 Επίπεδα συστήματα....35
3.3.4 Βέλτιστος έλεγχος....36
Κεφάλαιο 4 Η δική μας προσέγγιση πλοήγησης....37
4.1 Σχεδιασμός διαδρομής....37
4.1.1 Υλοποίηση A*....40
4.1.2 Αναπαράσταση εμποδίων....42
4.1.3 Ο αλγόριθμος σχεδιασμού διαδρομής....45
4.2 Έλεγχος κίνησης....46

4.2.1 Πολυκριτήριος ελεγκτής κίνησης 47
4.2.2 Αλγόριθμος ελέγχου κίνησης 51
Κεφάλαιο 5 Διεπαφή χρήστη 52
5.1 Περιβάλλον εργασίας 52
5.1.1 Φόρμα ελέγχου 53
5.1.2 Φόρμα συντεταγμένων κίνησης εμποδίων 55
5.1.3 Φόρμα ταχυτήτων – Γωνιών στροφής 56
5.1.4 Φόρμα πλοήγησης 56
Κεφάλαιο 6 Αποτελέσματα προσομοίωσης 59
6.1 Σχεδιασμός διαδρομής 59
6.1.1 Παράδειγμα σχεδιασμού διαδρομής 1 60
6.1.2 Παράδειγμα σχεδιασμού διαδρομής 2 61
6.1.3 Παράδειγμα σχεδιασμού διαδρομής 3 62
6.1.4 Παράδειγμα σχεδιασμού διαδρομής 4 63
6.1.5 Παράδειγμα σχεδιασμού διαδρομής 5 64
6.1.6 Παράδειγμα σχεδιασμού διαδρομής 6 64
6.1.7 Παράδειγμα σχεδιασμού διαδρομής 7 65
6.1.8 Παράδειγμα σχεδιασμού διαδρομής 8 65
6.2 Έλεγχος Κίνησης 66
6.2.1 Παράδειγμα ελέγχου κίνησης 1 66
6.2.2 Παράδειγμα ελέγχου κίνησης 2 67
6.2.3 Παράδειγμα ελέγχου κίνησης 3 68
6.2.4 Παράδειγμα ελέγχου κίνησης 4 69
6.2.5 Παράδειγμα ελέγχου κίνησης 4 71
6.3 Πλοήγηση 72
6.3.1 Παράδειγμα Πλοήγησης 1 72
6.3.2 Παράδειγμα Πλοήγησης 2 74
6.3.3 Παράδειγμα Πλοήγησης 3 75
6.3.4 Παράδειγμα Πλοήγησης 4 76
6.3.5 Παράδειγμα Πλοήγησης 5 77
6.3.6 Παράδειγμα Πλοήγησης 6 78
Κεφάλαιο 7 Επίλογος 79
7.1 Συμπεράσματα 79
7.2 Περαιτέρω ανάπτυξη 79
7.3 Κατακλείδα 80
Βιβλιογραφία 81

Λίστα Εικόνων

Σχήμα 2.3.1 Ρομποτικό μοντέλο - κινητή μονάδα 15
Σχήμα 2.3.2 Διάγραμμα ηλεκτρονικών υποσυστημάτων κινητής μονάδος 17
Σχήμα 2.3.3 Λειτουργικό διάγραμμα συστήματος ελέγχου διεύθυνσης 17
Σχήμα 2.3.4 Ηλεκτρικό σύστημα προσαρμογής ελέγχου διεύθυνσης 18
Σχήμα 2.3.5 Λειτουργικό διάγραμμα συστήματος προώθησης 19
Σχήμα 2.3.6 Ηλεκτρικό σύστημα προσαρμογής ελέγχου προώθησης 19
Σχήμα 2.3.7 Λειτουργικό διάγραμμα συστήματος πέδησης 20
Σχήμα 2.3.8 Τυπωμένο κύκλωμα συστήματος ενεργοποίησης απενεργοποίησης 20
Σχήμα 2.3.9 Μηχανισμός ανύψωσης εικονολήπτη 21
Σχήμα 2.3.10 Ρομποτική κεφαλή στήριξης εικονολήπτη και αισθητηρίων 22
Σχήμα 2.3.11 Ηλεκτρονική κατασκευή συστήματος ελέγχου της κινητής μονάδας 23
Σχήμα 2.3.12 Ηλεκτρονική κατασκευή συστήματος έλεγχου ρομποτικής κεφαλής και αισθητηρίων 23
Σχήμα 2.3.13 Ηλεκτρονική κατασκευή κεντρικού υπολογιστικού συστήματος κινητής μονάδας 24
Σχήμα 2.4.1 Διάγραμμα επιμέρους υπομονάδων κεντρικού σταθμού 25
Σχήμα 2.5.1 Λογισμικό ελέγχου κινητής μονάδας 26
Σχήμα 2.5.2 Λογισμικό ελέγχου ρομποτικής κεφαλής και εικονολήπτη 27
Σχήμα 2.6.1 Ρομποτικό σύστημα, κινητή μονάδα, υποσυστήματα και κεντρικός σταθμός 28
Σχήμα 3.1.1 Επίπεδα ελέγχου 31
Σχήμα 4.1.1 Διακριτοποίηση του χώρου εργασίας του ρομποτικού συστήματος 38
Σχήμα 4.1.2 Οκτώ συνολικά επιτρεπόμενοι προσανατολισμοί 38
Σχήμα 4.1.3 Επιτρεπτές μεταβάσεις στο διακριτοποιημένο χώρο. 39
Σχήμα 4.1.4 Παράδειγμα αναζήτησης του αλγορίθμου A* παρουσία εμποδίου 41
Σχήμα 4.1.5 Αναπαράσταση εμποδίων 42
Σχήμα 4.1.6 Μη επιτρεπτές (επικίνδυνες) μεταβάσεις μέσω εμποδίων 43
Σχήμα 4.1.7 Παράδειγμα αναζήτησης βελτιωμένου αλγορίθμου A* παρουσία εμποδίου 44
Σχήμα 4.1.8 Αναπαράσταση εμποδίων και επίπεδο ασφαλείας Offset 44
Σχήμα 4.1.9 αλγόριθμος σχεδιασμού διαδρομής σε ψευδοκώδικα 45
Σχήμα 4.2.1 θεωρητικό ρομποτικό μοντέλο 46
Σχήμα 4.2.2 Χώρος αναζήτησης δυναμικού παραθύρου 48
Σχήμα 4.2.3 Εκτίμηση τροχιάς ρομποτικού συστήματος και τελική θέση τερματισμού 50
Σχήμα 4.2.4 αλγόριθμος ελέγχου κίνησης σε ψευδοκώδικα 51
Σχήμα 5.1.1Λογισμικό διασύνδεσης χρήστη για προσομοίωση του ρομποτικού συστήματος 52
Σχήμα 5.1.2 Φόρμα ελέγχου 54
Σχήμα 5.1.3 Φόρμα συντεταγμένων κίνησης και εμποδίων 55
Σχήμα 5.1.4 Φόρμα ταχυτήτων - γωνιών στροφής 56
Σχήμα 5.1.5 Φόρμα πλοήγησης 56
Σχήμα 5.1.6 Φόρμα πλοήγησης επιλογής File 57
Σχήμα 5.1.7 Φόρμα πλοήγησης επιλογής View 57
Σχήμα 5.1.8 Φόρμα πλοήγησης επιλογή Run 58
Σχήμα 6.1.1Left: Start(0, 0, 0), Destination (300, 300, X). 60
Σχήμα 6.1.2 Left: Start(0, 0, 90), Destination (0, -300, X). 60
Σχήμα 6.1.3 Left: Start(0, 0, 0), Destination (300, 300, 180). 61
Σχήμα 6.1.4 Left: Start(0, 0, 0), Destination (0, -300,0). 61
Σχήμα 6.1.5 Left: Obstacle_Offset=0, Start(0, 0, 0), Destination (600, 0, X). 62
Σχήμα 6.1.6 Left: Obstacle_Offset=100, Start(0, 0, 0), Destination (600, 0, X). 62
Σχήμα 6.1.7 Left: Obstacle_Offset=20, Start(0, 0, 0), Destination (600, -300, X). 63

Σχήμα 6.1.8 Left: Obstacle_Offset=20, Start(0, 0, 0), Destination (500, 0,, X). 63
Σχήμα 6.1.9 Astar Pivot=100, Obstacle_Offset=100, Start(-1100, -600,-90), Destination (800,700, X).64
Σχήμα 6.1.10 Astar Pivot =100, Obstacle_Offset=50, Start(-1100, -600, 0), Destination (150, 150, X). 64
Σχήμα 6.1.11 Astar Pivot =100, Obstacle_Offset=50, Start(1700, 800, 0), Destination (1700, -800, X) 65
Σχήμα 6.1.12 Astar Pivot =100, Obstacle_Offset=50, Start(1700, 800, 180), Destination (-300, 200, X) 65
Σχήμα 6.2.1 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 0, X) 66
Σχήμα 6.2.2 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 0, X) 66
Σχήμα 6.2.3 L=1 dt=0,01, Start(0, 0, 0), Destination (2, 2, X). 67
Σχήμα 6.2.4 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 2, X) 67
Σχήμα 6.2.5 L=4 (Left), L=10 (Right), dt=0,01, Start(0, 0, 0), Destination (10, 10, X). 68
Σχήμα 6.2.6 L=4(Up), L=10(Down), dt=0,01, Start(0, 0, 0), Destination (10, 10, X). 68
Σχήμα 6.2.7 L=1 (Up Left), L=1.5 (Up Right). 69
Σχήμα 6.2.8 dt=0,01, Start(0, 0, 0), Destination (10, 10, X). 70
Σχήμα 6.3.1 Left: Path Planning, Astar Pivot =50, Obstacle_Offset=50), Start(0, 0, 0), 72
Σχήμα 6.3.2 Motion Controller, για διαφορετικές τιμές delay, Start(0, 0, 0), 73
Σχήμα 6.3.3 Astar Pivot =50, Obstacle_Offset=20), Start(0, 0, 0), 74
Σχήμα 6.3.4 Astar Pivot =50, Obstacle_Offset=20, dt=0,01, Start(0, 0, 0), 75
Σχήμα 6.3.5 Astar Pivot =100, Obstacle_Offset=50, dt=0,01, Start(-1100, -600, -90), 76
Σχήμα 6.3.6 Astar Pivot =100, Obstacle_Offset=50, dt=0,01 Start(3000, 0, 0), 77
Σχήμα 6.3.7 Astar Pivot=100, Obstacle_Offset=50, dt=0,01 Start(3000, 0, 0), 78

Κεφάλαιο 1 Εισαγωγή

1.1 Η πλοήγηση στη ζωή μας

Η εργασία αυτή ασχολείται με τη σχεδίαση και την υλοποίηση ενός συστήματος πλοήγησης αυτόνομων ρομποτικών συστημάτων. Για να εισάγουμε τον αναγνώστη στην έννοια της πλοήγησης, θα δώσουμε ένα παράδειγμα που συναντάμε συχνά στην καθημερινή μας ζωή.

Ας υποθέσουμε ότι είμαστε στο σπίτι μας και θέλουμε να πάμε στο πανεπιστήμιο. Επίσης ότι το αυτοκίνητο μας είναι παρκαρισμένο έξω από αυτό και επιπλέον ότι είναι μια μέρα που δεν έχει κανένα αυτοκίνητο στο δρόμο που οδηγεί στον προορισμό μας.

Η πρώτη σκέψη που κάνουμε είναι ένα πλάνο-σχέδιο πάνω σε ένα χάρτη που η μνήμη μας έχει φροντίσει να αποθηκεύσει, ώστε να βρούμε ποια είναι η πιο σύντομη διαδρομή που πρέπει να ακολουθήσουμε για να φτάσουμε.

Όμως ακόμα δεν έχουμε μπει στο αυτοκίνητο μας, έτσι για να φτάσουμε πραγματικά πρέπει να μπούμε και να το οδηγήσουμε κατά μήκος αυτής της διαδρομής μέχρι να φτάσουμε στο επιθυμητό σημείο, θέτοντας σε λειτουργία τις γνώσεις, τις εμπειρίες, αλλά και το ένστικτό μας προκειμένου να επιτύχουμε το στόχο μας.

Τα βήματα που ακολουθούνται είναι τα εξής:

1. Αποφασίσαμε που θέλαμε να πάμε.
2. Δημιουργήσαμε ένα πλάνο στο μυαλό μας για τη διαδρομή που θα ακολουθήσουμε.
3. Οδηγήσαμε το αυτοκίνητο κατά μήκος αυτής της διαδρομής, λαμβάνοντας υπ' όψιν όλους τους περιορισμούς που θέτει και με βάση τους χειρισμούς που δέχεται.

Το παραπάνω αποτελεί μία διαδικασία πλοήγησης του οχήματος με σκοπό να μεταφέρουμε τον εαυτό μας με ασφάλεια στο πανεπιστήμιο. Αυτό μπορεί να θεωρηθεί ως μία απλή διαδικασία για το μυαλό του ανθρώπου, αλλά είναι αρκετά πολύπλοκη για να υλοποιηθεί και να εφαρμοστεί σε αυτόνομα ρομποτικά συστήματα. Με το ζήτημα αυτό ασχολούνται ολοένα και περισσότεροι ερευνητές και μηχανικοί κατά την τελευταία δεκαετία.

Στις ενότητες που ακολουθούν θα περιγράψουμε αναλυτικά τον τρόπο αντιμετώπισης τέτοιων προβλημάτων.

1.2 Αναλυτική περιγραφή εργασίας

Όπως αναφέρθηκε και πιο πάνω η πλοήγηση ρομποτικών συστημάτων είναι ένα πολύπλοκο πρόβλημα πολλών διαστάσεων και απαιτεί αντιμετώπιση σε κάθε επίπεδο ξεχωριστά.

Η ανάπτυξη και υλοποίηση ενός αξιόπιστου συστήματος ως προς τα ηλεκτρομηχανικά του υποσυστήματα είναι το πρώτο επίπεδο που πρέπει να

ερευνηθεί, προκειμένου να υπάρχει δυνατότητα καθοδήγησης και το ρομποτικό σύστημα να ανταποκρίνεται με ομαλότητα και αξιοπιστία.

Στη συνέχεια απαιτείται η ανάπτυξη και υλοποίηση αλγορίθμων και μεθοδολογιών που να επιτρέπουν στο σύστημα να προσαρμόζεται και να αποφασίζει για τις κινήσεις του, δηλαδή να κινείται στο χώρο χωρίς να παραβιάζει τους κανόνες ασφάλειας του αλλά και του χώρου στον οποίο κινείται.

Η ανάπτυξη των απαραίτητων αλγορίθμων και μεθοδολογιών περιλαμβάνει αρχικά την χαρτογράφηση του περιβάλλοντα χώρου (mapping), ώστε να είναι δυνατή η απεικόνιση του σε μια μορφή που το ρομποτικό σύστημα να μπορεί επεξεργαστεί. Μέσω της χαρτογράφησης και με τη βοήθεια αλγορίθμων εντοπισμού (localization) αλλά και αισθητήρων μπορεί να καθοριστεί η τρέχουσα θέση του ρομπότ μέσα στο χαρτογραφημένο περιβάλλον. Στη συνέχεια με την ανάπτυξη αλγορίθμων σχεδιασμού ασφαλούς διαδρομής (path planning) μεταξύ θέσεων, αποφασίζεται η συγκεκριμένη διαδρομή που θα ακολουθήσει το ρομποτικό σύστημα. Η επιλεγμένη διαδρομή με την βοήθεια του ελεγκτή κίνησης θα επιτρέψει τη μετακίνηση του ρομποτικού συστήματος κατά μήκος κάποιας διαδρομής.

Η παρούσα εργασία εστιάζει στην συμπληρωματική ανάπτυξη του ρομποτικού οχήματος του Εργαστηρίου Δομής της Ύλης και Φυσικής Laser του Πολυτεχνείου Κρήτης, με στόχο την εξέλιξη του σε αυτόνομο ρομποτικό όχημα πλοήγησης. Συγκεκριμένα πρόκειται για ένα τετράτροχο ρομποτικό όχημα με δυνατότητα στροφής των δύο εμπρόσθιών τροχών, δηλαδή έχει δυνατότητες κίνησης αντίστοιχες με αυτές ενός αυτοκινήτου. Το επίπεδο ελέγχου που παρέχεται από το συγκεκριμένο όχημα επιτρέπει καθορισμό της γραμμικής ταχύτητας του, αλλά και της γωνίας στροφής.

Στα πλαίσια της εργασίας αυτής αναπτύχθηκαν τα δύο βασικά επίπεδα της πλοήγησης, σχεδιασμός διαδρομής και έλεγχος κίνησης, προσαρμοσμένα στις δυνατότητες των τετράτροχων ρομποτικών συστημάτων υπό μελέτη, θεωρώντας ότι ο χάρτης του περιβάλλοντα χώρου και η θέση του οχήματος σ' αυτόν είναι διαθέσιμα. Συγκεκριμένα, για το σχεδιασμό διαδρομής χρησιμοποιείται ένας αλγόριθμος πληροφορημένης ευριστικής αναζήτησης (A*), ο οποίος παράγει μια διακριτο-ποιημένη διαδρομή που εγγυάται ασφάλεια (αποφυγή συγκρούσεων με εμπόδια) και βέλτιστο κόστος (ελάχιστο μήκος διαδρομής). Η διαδρομή αυτή μεταβιβάζεται στον ελεγκτή κίνησης ο οποίος αναλαμβάνει να μετακινήσει το όχημα κατά μήκος της διαδρομής, θέτοντας κατάλληλες τιμές στην ταχύτητα και στη γωνία στροφής του οχήματος. Επιπλέον το ρομποτικό όχημα δεν επιτρέπεται να μεταβάλλει τις τιμές στα στοιχεία ελέγχου από μια χρονική στιγμή στην επόμενη, πάνω από ένα μέγιστο επιτρεπτό όριο, με κύριο κριτήριο να φτάσει όσο το δυνατό πιο κοντά στον επιθυμητό στόχο και να σταματήσει εκεί. Αυτό επιτυγχάνεται με την βοήθεια μιας πολυκριτήριας αντικειμενικής συνάρτησης (objective function) και του δυναμικού παραθύρου (dynamic window), μέσω των οποίων η ταχύτητα και η γωνία στροφής αποφασίζονται δυναμικά σε πραγματικό χρόνο, με γνώμονα την ακρίβεια παρακολούθησης της διαδρομής και την ομαλότητα της κίνησης. Επιπλέον, λαμβάνονται υπ' όψιν όλοι οι κινηματικοί και δυναμικοί περιορισμοί της κίνησης του οχήματος εξασφαλίζοντας μετακίνηση με ομαλότητα και ακρίβεια.

Το προτεινόμενο σύστημα πλοήγησης ανταποκρίνεται επιτυχώς σε διάφορα προσομοιωμένα σενάρια που έχουν μελετηθεί στο γραφικό περιβάλλον που αναπτύχθηκε για να χρησιμοποιηθεί ως διασύνδεση μεταξύ του χρήστη και του συστήματος πλοήγησης.

1.3 Διάθρωση κειμένου

Μπορούμε να κάνουμε μια περιγραφική αναφορά για τη δομή των κεφαλαίων που ακολουθούν στην εργασία αυτή.

Στο κεφάλαιο 2 θα παραθέσουμε τα τεχνικά χαρακτηριστικά και τα υποσυστήματα του ρομποτικού συστήματος στο οποίο στηριχτήκαμε για να αναπτύξουμε τη μεθοδολογία του αλγορίθμου που αφορά τη μελέτη του σχεδιασμού διαδρομής και του ελέγχου κίνησης.

Στο κεφάλαιο 3 γίνεται μια αναφορά στο πρόβλημα της πλοήγησης και στους τρόπους επίλυσής του. Επίσης, παρατίθενται οι βασικές κατηγορίες αντιμετώπισης του προβλήματος αυτού καθώς και σχετικές μελέτες και έρευνες που έχουν πραγματοποιηθεί μέχρι σήμερα από διάφορες ερευνητικές ομάδες.

Στο κεφάλαιο 4 αναπτύσσεται η δική μας προσέγγιση και περιλαμβάνει την ανάπτυξη και υλοποίηση των αλγορίθμων σχεδιασμού διαδρομής και ελέγχου κίνησης.

Στο κεφάλαιο 5 περιγράφεται η υλοποίηση και ο σχεδιασμός της διεπαφής που αναπτύξαμε και υλοποιήσαμε για τη προσομοίωση των αλγόριθμων πλοήγησης.

Στο κεφάλαιο 6 παρατίθενται αναλυτικά τα αποτελέσματα της προσομοίωσης πλοήγησης για διάφορες περιπτώσεις αναζήτησης μονοπατιού και ελέγχου κίνησης.

Στο Κεφάλαιο 7 παρατίθενται τα συμπεράσματα στα οποία καταλήξαμε με την ολοκλήρωση της εργασίας αυτής. Συμπληρώνουμε κάποιες ιδέες για μελλοντική δουλειά που θα μπορούσε να εξελίξει τη συγκεκριμένη εργασία.

Κεφάλαιο 2 Ρομποτικό μοντέλο

2.1 Ιστορική αναδρομή

Η εργασία αυτή εκπονήθηκε για ένα τετράτροχο ρομποτικό όχημα που αναπτύχθηκε στο Εργαστήριο Δομής της Ύλης και Φυσικής Λέιζερ του Γενικού Τμήματος του Πολυτεχνείου Κρήτης, καθώς και όλων των υποσυστημάτων όπως περιγράφονται στην συνέχεια. Το 2005 το εργαστήριο συμμετείχε στην Καινοτόμο Ενέργεια «Φυτώριο Ιδεών Φοιτητών Πανεπιστημίων UNISTEP», στέγασε και ανέπτυξε το ρομποτικό όχημα με στόχο την εφαρμογή του σε «Forecast Fires». Ανέλαβα την τεχνική ευθύνη με σκοπό την ανάπτυξη ενός συστήματος που θα αποτελούνταν από το ρομποτικό όχημα και από ένα κεντρικό σταθμό λήψης/επεξεργασίας δεδομένων και αποστολής σημάτων ελέγχου για τη κίνηση του οχήματος. Το ρομποτικό όχημα θα περιελάμβανε προηγμένα οπτοηλεκτρονικά συστήματα, όπως εικονολήπτες και αισθητήρες θερμοκρασίας, επιπλέον θα βρίσκεται σε συνεχή επικοινωνία με το κεντρικό σταθμό βάσης, στέλνοντάς του δεδομένα από τα οπτοηλεκτρονικά συστήματα, παίρνοντας τις απαραίτητες πληροφορίες του περιβάλλοντος καθώς και τις εντολές καθοδήγησης από τον κεντρικό σταθμό. Το σύστημα (ρομποτικό όχημα και κεντρικός σταθμός) θα μπορούσε να χρησιμοποιηθεί κυρίως για περιφρούρηση και προστασία ευαίσθητων από πυρκαγιές περιοχών, όπως τα δάση.

Το ρομποτικό όχημα που υλοποιήθηκε στα πλαίσια του συγκεκριμένου προγράμματος έχει δυνατότητα μόνο τηλεχειρισμού. Η παρούσα εργασία έχει σκοπό την συμπληρωματική εξέλιξη του αναπτυγμένου αυτού ρομποτικού συστήματος και τον εμπλουτισμό του με δυνατότητες αυτόνομης πλοήγησης.

2.2 Βασικές έννοιες

Με τον όρο ρομπότ (robot) εννοούμε το αυτόνομο σύστημα που έχει τη δυνατότητα να λαμβάνει πληροφορίες από το περιβάλλον του και να προσαρμόζει τις δράσεις του στις αντιλήψεις του, καθώς και στις εντολές του ανωτέρου του.

Μπορούμε να χωρίσουμε τα ρομποτικά συστήματα σε δύο κατηγορίες: τα στατικά και τα κινητά**.** Στατικά είναι τα ρομποτικά συστήματα που έχουν ένα σταθερό σημείο αναφοράς και οι κινήσεις των στοιχείων δράσης τους βασίζονται στην αναφορά αυτή. Κινητά είναι τα ρομποτικά συστήματα που το σημείο αναφοράς τους μεταβάλλεται συνεχώς.

Γενικότερα, τα ρομποτικά συστήματα μπορούν να ελέγξουν και να μεταβάλλουν τις θέσεις των αρθρώσεων τους. Κάθε ανεξάρτητη κίνηση που μπορεί να πραγματοποιήσει ένα ρομπότ με μια από τις συσκευές δράσης του καθορίζει και ένα βαθμό ελευθερίας (DOF- Degree of Freedom).

Για τα κινητά ρομποτικά συστήματα έχουμε δυο κατηγορίες: αυτά που κινούνται με τροχούς και αυτά που κινούνται με άκρα. Επίσης, στα κινητά ρομποτικά συστήματα, οι βαθμοί ελευθερίας δεν είναι απαραίτητα ίδιοι με τις διαστάσεις του χώρου διατάξεων (configurations) που μπορεί να έχει το ρομποτικό σύστημα στο περιβάλλον. Για παράδειγμα, ένα αυτοκίνητο μπορεί να κινηθεί πάνω σε μια επιφάνεια και να προσεγγίσει οποιοδήποτε σημείο (X,Y) με οποιαδήποτε

προσανατολισμό Θ, άρα το μοντέλο του αυτοκινήτου έχει ένα τρισδιάστατο χώρο διατάξεων (X,Y,Θ) αλλά μόνο δύο βαθμούς ελευθερίας αφού μπορεί να πάει μπρος/πίσω και να στρέψει τους δυο από τους τέσσερις τροχούς αριστερά-δεξιά. Τέτοια ρομποτικά συστήματα ονομάζονται μη ολονομικά. Αντίθετα, ένας ρομποτικός βραχίονας έχει ίδια διάσταση διατάξεων με τους βαθμούς ελευθερίας. Τέτοια συστήματα ονομάζονται ολονομικά. Το μοντέλο του αυτοκινήτου είναι μη ολονομικό (nonholonomic), ενώ ο βραχίονας είναι ολονομικός (holonomic).

2.3 Περιγραφή κινητής μονάδος

Η κινητή μονάδα είναι ένα μη επανδρωμένο επίγειο τετράτροχο ρομποτικό όχημα με δυνατότητα στροφής των δύο εμπρόσθιων τροχών (σχήμα 2.3.1). Άρα είναι ένα μη ολονομικό κινητό ρομποτικό σύστημα. Πέρα από την κινητή μονάδα υπάρχει και ένας κεντρικός σταθμός μετάδοσης και επεξεργασίας δεδομένων. Η κινητή μονάδα βρίσκεται σε συνεχή επικοινωνία με το σταθερό κεντρικό σταθμό, ο οποίος ελέγχει την κινητή μονάδα αποστέλλοντας και λαμβάνοντας όλα τα απαραίτητα σήματα ελέγχου (εικόνα, θερμοκρασία, καθοδήγηση, κλπ).

Σχήμα 2.3.1 Ρομποτικό μοντέλο - κινητή μονάδα

Η κινητή μονάδα αποτελείται από τα εξής (σχήμα 2.3.2):

Ηλεκτρικά-ηλεκτρονικά συστήματα και ελεγκτές

1) Μικροεπεξεργαστής ATMEL AT8515. Αυτός ο μικροεπεξεργαστής είναι σχεδιασμένος για να έχει τον έλεγχο της κινητής μονάδας. Ουσιαστικά, ελέγχει όλες τις παραμέτρους που καθορίζουν την πλοήγηση του συστήματος.

2) Ελεγκτές κινητήρων και οδηγοί έλεγχου του συστήματος. Αναφερόμαστε στις απαραίτητες ηλεκτρονικές διατάξεις που επιτρέπουν την καθοδήγηση σε ρεύμα και ισχύ των κινητήρων, καθώς επίσης και των απαραιτήτων διατάξεων που ελέγχουν το σύστημα. Οι εντολές λαμβάνονται από τον μικροεπεξεργαστή AT8515 και οι ελεγκτές καθορίζουν πώς θα λειτουργήσει ο κάθε κινητήρας και κάθε οδηγός έλεγχου του συστήματος.
3) Ηλεκτρικοί βηματικοί (step motor) ή συνεχούς ρεύματος (DC) κινητήρες. Οι κινητήρες αυτοί τροφοδοτούν με τις απαραίτητες ροπές τα αντίστοιχα σημεία ελέγχου που πρέπει να περιστραφούν.

Προηγμένα οπτοηλεκτρονικά συστήματα

Η οπτική παρακολούθηση επιτυγχάνεται με χρήση μιας έγχρωμης βιντεοκάμερας η οποία αποστέλλει τα δεδομένα στο σταθμό δεδομένων μέσω μίας κάρτας PCI πού υποστηρίζει μέχρι τέσσερις τέτοιες κάμερες. Γίνεται χρήση έτοιμου λογισμικού που επιτρέπει τη μεταφορά των σημάτων video μέσω του τοπικού δικτύου.

Σύστημα ασύρματης επικοινωνίας

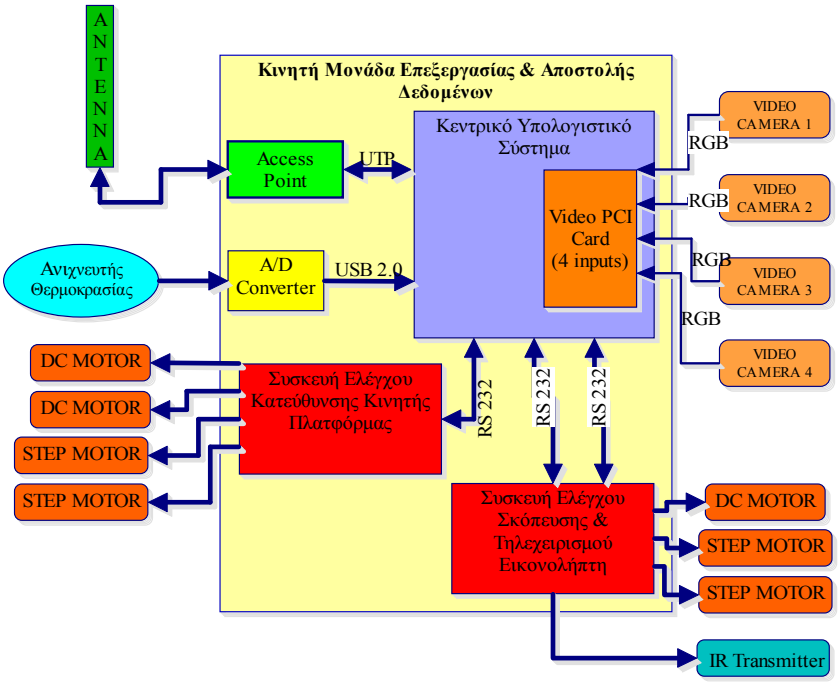
1) Μονάδα ασύρματης (wireless) επικοινωνίας βασισμένη σε τεχνολογία δικτύων Ethernet και συγκεκριμένα στο πρωτόκολλο ασύρματου τοπικού δικτύου (TCP-UDP WiFi 802-11b).
2) Κεραία unipolar με εμβέλεια μέχρι και 5Km.

Εξελιγμένο υπολογιστικό σύστημα

Το σύστημα αυτό αναλαμβάνει όλες τις λειτουργίες των περιφερειακών υποσυστημάτων (A/D, PCI Cards, Controllers κλπ). Το κεντρικό υπολογιστικό σύστημα αναθέτει στα επιμέρους υποσυστήματα τις διεργασίες που θα πραγματοποιηθούν. Ο χρονοπρογραμματισμός και η προτεραιότητα των διεργασιών καθορίζονται από το χρήστη μέσω ανάλογου προγράμματος διασύνδεσης (σε Visual Basic 6) το οποίο επιτρέπει και τη διασύνδεση και επικοινωνία μεταξύ δύο υπολογιστών που είναι συνδεδεμένοι με ασύρματο τοπικό δίκτυο (WirelessEthernet). Η διασύνδεση με τα επιμέρους ηλεκτρονικά υποσυστήματα και κατ' επέκταση με τα μηχανικά μέρη επιτυγχάνεται μέσω των πρωτοκόλλων RS-232 και USB 2.0.

Μηχανικά συστήματα προσαρμογής και ελέγχου

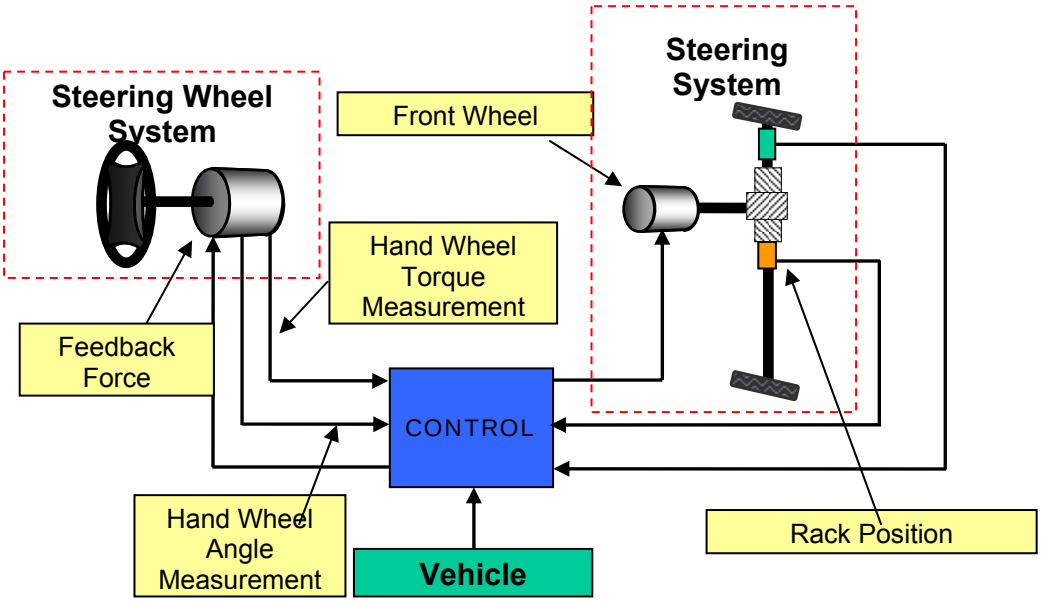
1) Μηχανικά συστήματα στήριξης και μηχανικοί μετατροπείς (μειωτήρες κινητήρων, βάσεις ηλεκτρονικών κατασκευών, βάσεις κινητήρων κλπ.)
2) Κατασκευή ανύψωσης κάμερας στα 4m.
3) Ρομποτική κεφαλή στήριξης εικονολήπτη και αισθητήρων.
4) Εικονολήπτης και αισθητήρες ελέγχου.



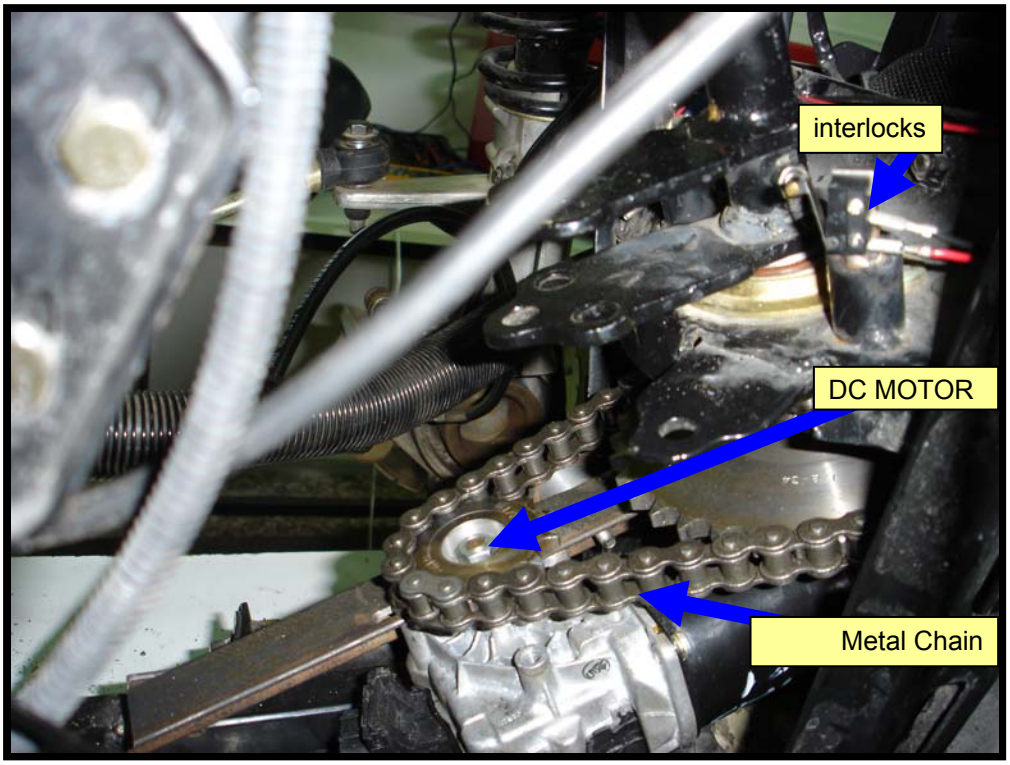
Σχήμα 2.3.2 Διάγραμμα ηλεκτρονικών υποσυστημάτων κινητής μονάδος

2.3.1 Σύστημα ελέγχου διεύθυνσης

Το σύστημα διεύθυνσης αποτελείται από πολλά ανεξάρτητα μηχανικά και ηλεκτρονικά υποσυστήματα που σκοπό έχουν να δώσουν πλήρη έλεγχο της κινητής μονάδας στο χρήστη. Η μέγιστη γωνία στροφής του τιμονιού είναι 45°. Το σύστημα διεύθυνσης φαίνεται στα σχήματα 2.3.3 και 2.3.4.



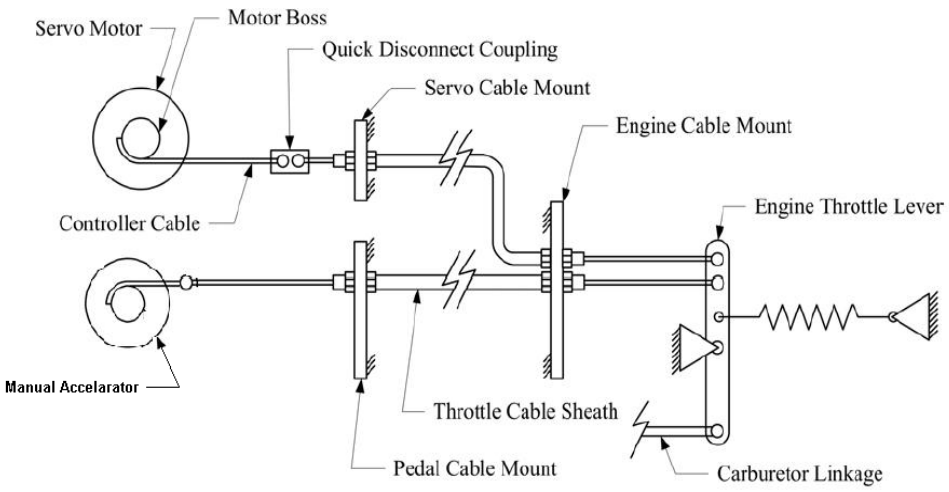
Σχήμα 2.3.3 Λειτουργικό διάγραμμα συστήματος ελέγχου διεύθυνσης



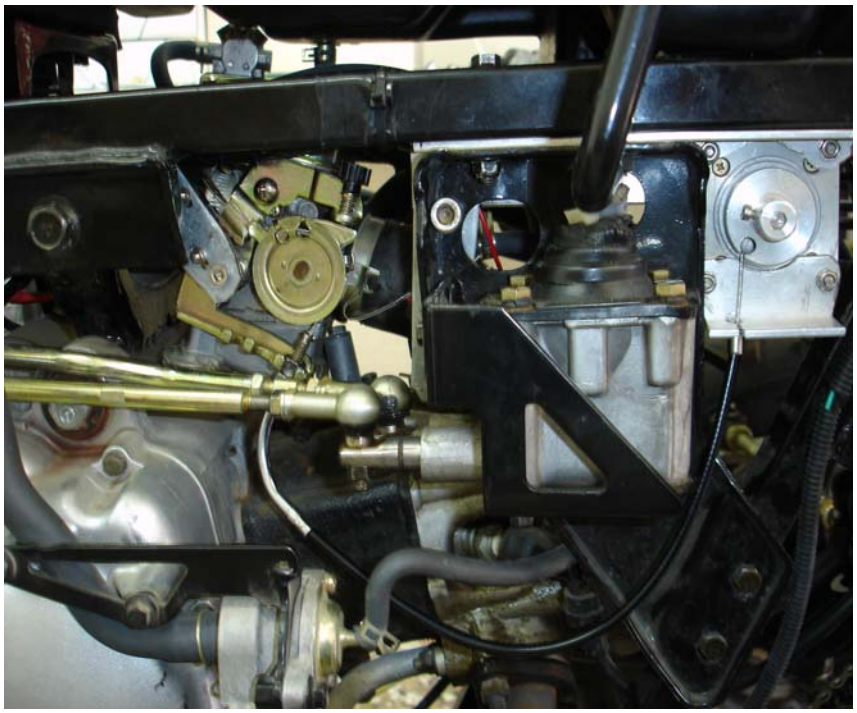
Σχήμα 2.3.4 Ηλεκτρικό σύστημα προσαρμογής ελέγχου διεύθυνσης

2.3.2 Σύστημα προώθησης

Το σύστημα προώθησης επιτρέπει με ελεγχόμενο τρόπο να καθορίζουμε την ταχύτητα του οχήματος σε πραγματικό χρόνο. Η μέγιστη δυνατή ταχύτητα μπορεί να φτάσει μέχρι και τα 70Km/h. Αποτελείται από δύο δομικά μέρη, το μηχανικό και το ηλεκτρονικό υποσύστημα. Στο μηχανικό σύστημα έχουν προσαρμοστεί ηλεκτρικοί βηματικοί κινητήρες που μας δίνουν την δυνατότητα να ελέγχουμε την ταχύτητα του οχήματος με ακρίβεια και ομαλότητα. Για λόγους απλότητας έχουμε θεωρήσει ότι το όχημα μπορεί να κινηθεί μόνο προς τα εμπρός. Στα σχήματα 2.3.5 και 2.3.6 φαίνονται αναλυτικά το σχέδιο και η πραγματική προσαρμογή του μηχανικού υποσυστήματος με τον κινητήρα και το ρομποτικό σύστημα.



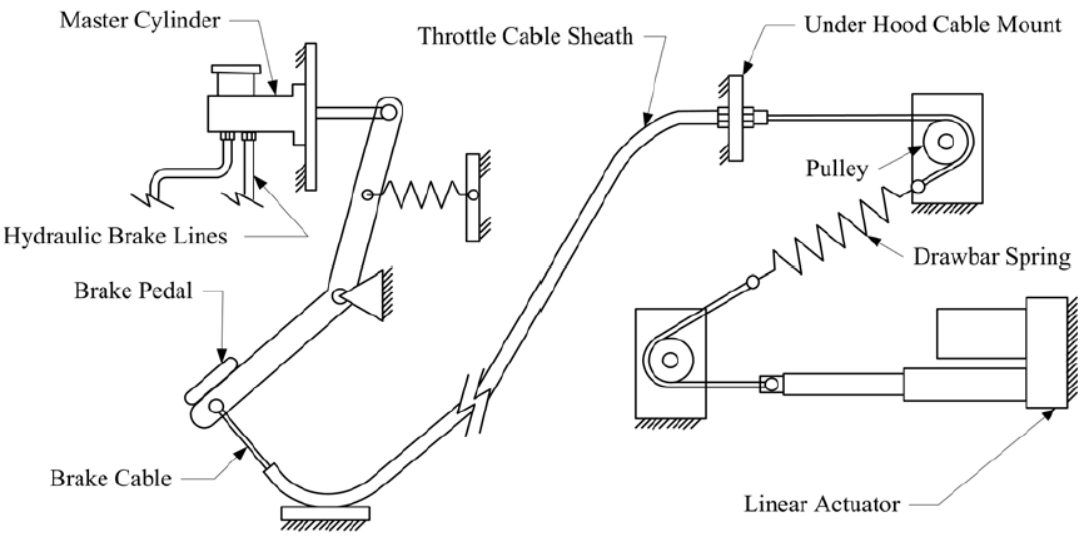
Σχήμα 2.3.5 Λειτουργικό διάγραμμα συστήματος προώθησης



Σχήμα 2.3.6 Ηλεκτρικό σύστημα προσαρμογής ελέγχου προώθησης

2.3.3 Σύστημα πέδησης

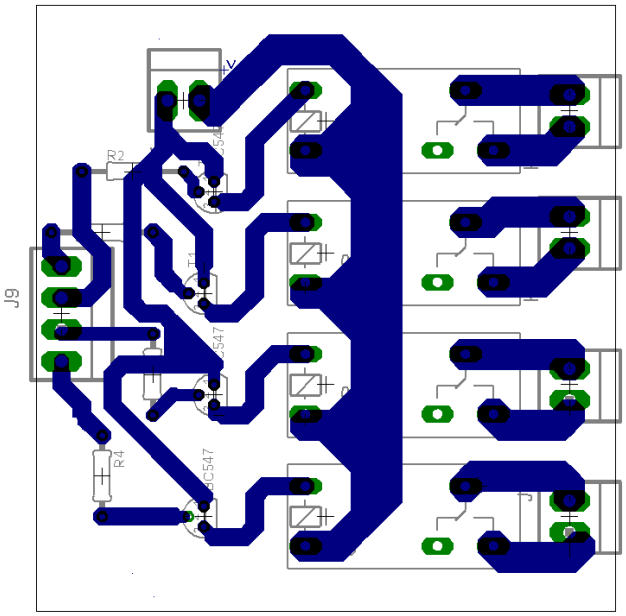
Το σύστημα πέδησης επιτρέπει με ελεγχόμενο τρόπο να καθορίζουμε την επιβράδυνση του οχήματος σε πραγματικό χρόνο. Αποτελείται από ένα προσαρμογέα που συνδέεται στο μηχανικό σύστημα πέδησης και μας επιτρέπει την συνδεσμολογία με έναν κινητήρα, ο οποίος ελέγχει το ποσοστό πέδησης που απαιτείται από την εφαρμογή. Το σύστημα πέδησης φαίνεται στο σχήμα 2.3.7.



Σχήμα 2.3.7 Λειτουργικό διάγραμμα συστήματος πέδησης

2.3.4 Σύστημα ενεργοποίησης/απενεργοποίησης

Όπως είναι φανερό εκτός από τις βασικές λειτουργίες της συμπεριφοράς της πλατφόρμας εν κινήσει χρειάζονται υλοποίηση και κάποια πολύ απλά αλλά και πολύ σημαντικά στοιχεία. Συγκεκριμένα, αναφερόμαστε στην ενεργοποίηση και απενεργοποίηση της μηχανής (διακόπτης) με ηλεκτρονικό τρόπο καθώς και την αυτόματη ανάφλεξη του κινητήρα (μίζα). Το σύστημα αυτό αποτελείται από μια ηλεκτρονική κατασκευή που στόχο έχει να ενεργοποιεί και να απενεργοποιεί το ρομποτικό σύστημα (με τη βοήθεια ηλεκτρομηχανικών διακοπτών relay) και επιπλέον να θέτει σε λειτουργία τον κινητήρα του. Στο σχήμα 2.3.8 φαίνεται το τυπωμένο κύκλωμα του εν λόγω ηλεκτρονικού κυκλώματος.

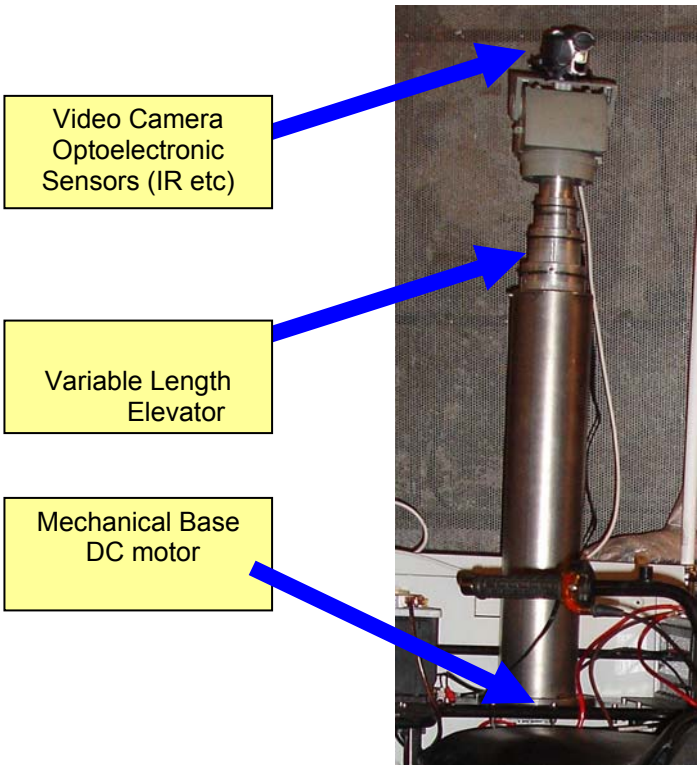


Σχήμα 2.3.8 Τυπωμένο κύκλωμα συστήματος ενεργοποίησης απενεργοποίησης

2.3.5 Κεραία ανύψωσης εικονολήπτη

Η κεραία ανύψωσης είναι ένας μηχανικός ανελκυστήρας κυκλικής διατομής με εσωτερικό ατέρμονα κοχλία αντίστροφου σπειρώματος. Η συγκεκριμένη σχεδίαση δίνει την δυνατότητα ανύψωσης σχετικά μεγάλων βαρών με μικρό όγκο μηχανισμών ανύψωσης και μικρή ιπποδύναμη οδήγησης. Επίσης, αυτός ο τρόπος κατασκευής εξασφαλίζει μια σχετικά καλή σταθερότητα και ευθύγραμμη κίνηση κατά την ανύψωσή του. Η πλήρης ανάπτυξη πρέπει να γίνει εν στάσει του οχήματος, γιατί όπως θα ήταν αναμενόμενο αν γίνει εν κινήσει θα υπάρχει μεγάλη πιθανότητα αστάθειας.

Η κατασκευή αυτή πραγματοποιήθηκε με την βοήθεια του συνεργάτη του εργαστηρίου Αποστολάκη Εμμανουήλ και παρέχει την επιλογή να αλλάξουμε το οπτικό μας πεδίο όταν αυτό για κάποιο λόγο είναι περιορισμένο. Επίσης, δίπλα στον εικονολήπτη προσαρμόζονται και διάφοροι αισθητήρες που βελτιώνουν πολύ περισσότερο την εικόνα με γεωγραφικά-θερμικά χαρακτηριστικά του χώρου στον οποίο βρίσκεται το όχημα. Η κεραία φαίνεται στο σχήμα 2.3.9.

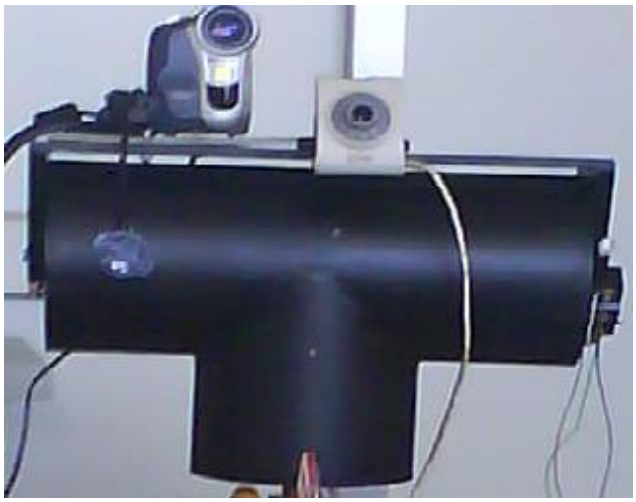


Σχήμα 2.3.9 Μηχανισμός ανύψωσης εικονολήπτη

2.3.6 Ρομποτική κεφαλή στήριξης εικονολήπτη και αισθητήρων

Η ρομποτική κεφαλή (σχήμα 2.3.10) είναι υπεύθυνη για την περιστροφή του εικονολήπτη και των αισθητήρων ως προς το οριζόντιο και το κατακόρυφο επίπεδο, έτσι ώστε να μας δίνεται η δυνατότητα να έχουμε πιο σφαιρική εικόνα για το περιβάλλον του ρομποτικού μας συστήματος. Το σύστημα αυτό καθοδηγείται από ένα ηλεκτρονικό σύστημα έλεγχου κινητήρων. Ο εικονολήπτης μεταφέρει το οπτικό του σήμα μέσω του τοπικού δικτύου στον κεντρικό σταθμό. Ο χρήστης έχει την

δυνατότητα απομακρυσμένου ελέγχου του εικονολήπτη καθώς και επιπλέον λειτουργίες, όπως εγγραφή και λήψη εικόνων, zoom, κλπ. Για να επιτευχθεί αυτός ο έλεγχος χρησιμοποιούμε ένα σύστημα αποκωδικοποίησης υπερύθρων. Τα δύο παραπάνω υποσυστήματα δοκιμάστηκαν ότι λειτουργούν σε πραγματικές συνθήκες και μεταφέρουν σε πραγματικό χρόνο εικόνα και δεδομένα σε συνεργασία με το ρομποτικό όχημα.



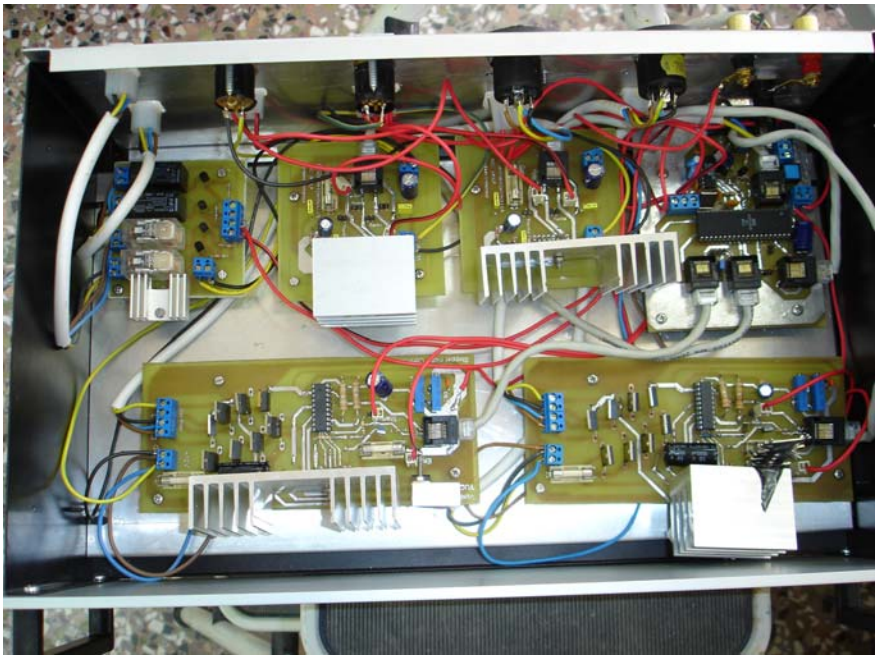
Σχήμα 2.3.10 Ρομποτική κεφαλή στήριξης εικονολήπτη και αισθητηρίων

2.3.7 Ηλεκτρονικό σύστημα ελέγχου

Το σύστημα ελέγχου είναι υπεύθυνο για να μεταφέρει τα σήματα ελέγχου σε κάθε ένα από τα παραπάνω υποσυστήματα (σύστημα αλλαγής διεύθυνσης, προώθησης, πέδησης, κλπ) τροφοδοτώντας τα με τις αντίστοιχες τιμές τάσεων και ρεύματος, έτσι ώστε οι αντίστοιχοι σερβομηχανισμοί και διακόπτες να κινηθούν σύμφωνα με τις εντολές του χρήστη.

Το σύστημα αυτό αποτελείται καταρχήν από το κύκλωμα του μικροεπεξεργαστή AT90S8515. Το ηλεκτρονικό κύκλωμα αυτό είναι εφοδιασμένο με το κατάλληλο λογισμικό που του επιτρέπει καταρχάς την επικοινωνία με το κεντρικό υπολογιστικό σύστημα (κατ' επέκταση και με τον χρήστη) και επιπλέον την κωδικοποίηση των αντίστοιχων σημάτων ώστε να γίνει εφικτός ο έλεγχος των επιμέρους ελεγκτών, κινητήρων, κλπ.

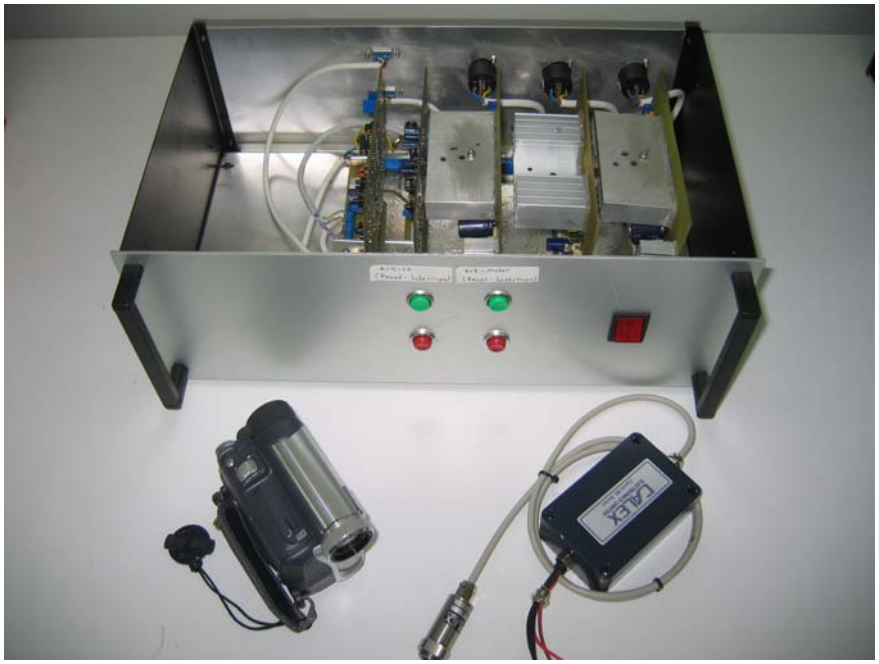
Περιλαμβάνει επίσης τα επιμέρους υποσυστήματα ελεγκτών, δηλαδή ελεγκτές κινητήρων, διακόπτες, κλπ. Κάθε ένα από αυτά τα υποσυστήματα καθορίζουν το πότε, αν, ή/και πώς θα λειτουργήσει ο κάθε σερβοκινητήρας. Τα κυκλώματα αυτά ελέγχονται όπως είπαμε από το μικροϋπολογιστικό σύστημα και στόχο έχουν να τροφοδοτήσουν τους αντίστοιχους σερβομηχανισμούς και διακόπτες με τις επιθυμητές τιμές. Στο σχήμα 2.3.11 φαίνεται αναλυτικά το συνολικό ηλεκτρονικό σύστημα ελέγχου.



Σχήμα 2.3.11 Ηλεκτρονική κατασκευή συστήματος ελέγχου της κινητής μονάδας

2.3.8 Ελεγκτής κεφαλής και εικονολήπτη

Το σύστημα αυτό (σχήμα 2.3.12) είναι υπεύθυνο για τον έλεγχο και την ακριβή καθοδήγηση της ρομποτικής κεφαλής του ρομποτικού μας συστήματος, αναπτύχθηκε κατασκευάστηκε και δοκιμάστηκε στο εν λόγω εργαστήριο στα πλαίσια διπλωματικής εργασίας [1]. Επιπλέον έχει αναπτυχτεί ένας εξελιγμένος αποκωδικοποιητής υπερύθρων που στόχο έχει τον απομακρυσμένο έλεγχο της βιντεοκάμερας (zoom, εγγραφή, αποθήκευση εικόνων).



Σχήμα 2.3.12 Ηλεκτρονική κατασκευή συστήματος έλεγχου ρομποτικής κεφαλής και αισθητηρίων

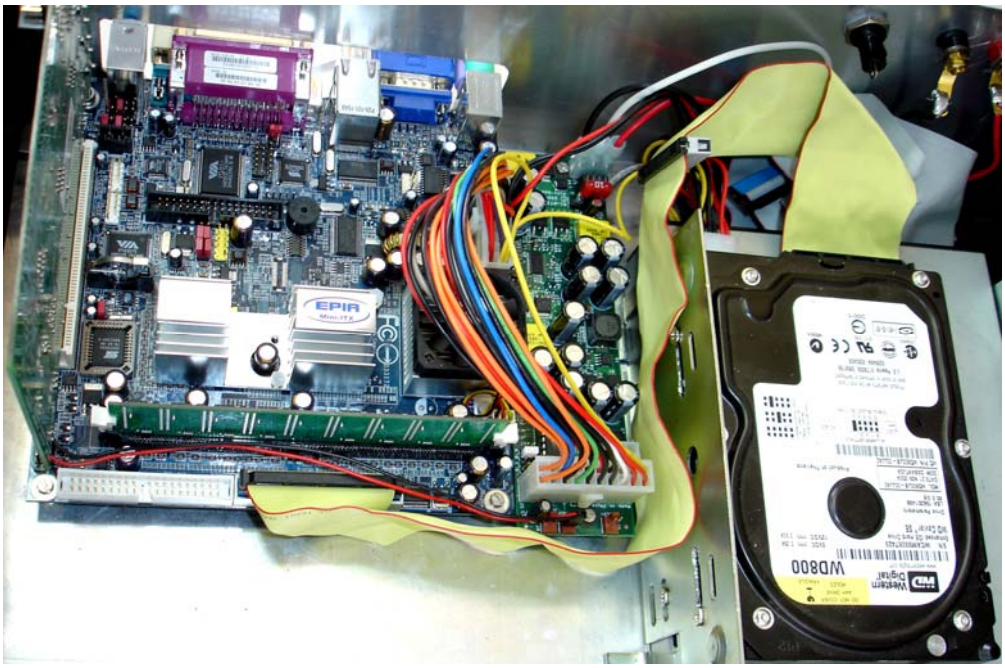
2.3.9 Κεντρικό υπολογιστικό σύστημα

Πρόκειται για ένα υπολογιστικό σύστημα χαμηλής κατανάλωσης και μικρού όγκου, ειδικά σχεδιασμένο για την κάλυψη εφαρμογών, όπου ο χώρος, η ταχύτητα και η κατανάλωση είναι κρίσιμης σημασίας.

Αναλυτικότερα, ο υπολογιστής αποτελείται από:

- ✓ Επεξεργαστή: 1 GHz Celeron
- ✓ Μητρική Κάρτα: EPIR mini - ITX
- ✓ Σκληρό Δίσκο: 80 GB
- ✓ Μνήμη RAM: DDR 400FSB 512 MB
- ✓ Κάρτα Video PCI: SPECTRA 8 (MULTI – INPUT VIDEO CAPTURE CARD)
- ✓ CD – ROM: 32x
- ✓ Λειτουργικό Σύστημα: Windows 2000
- ✓ Κατανάλωση 70Watt

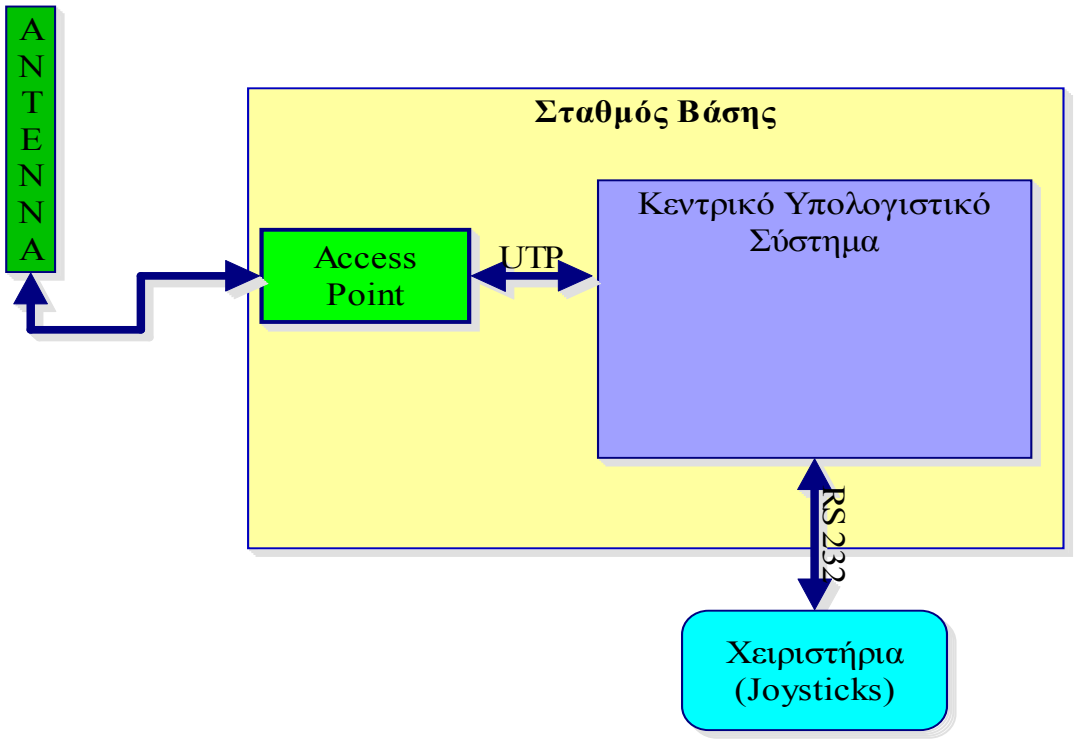
Το σύστημα αυτό (σχήμα 2.3.13) βρίσκεται σε συνεχή επαφή με τον κεντρικό σταθμό δεδομένων (μέσω ασύρματου τοπικού δικτύου). Είναι συνδεμένο επίσης με όλα τα υπόλοιπα υποσυστήματα τις κινητής μονάδας, έτσι ώστε να μπορεί μεταφέρει όλες τις πληροφορίες ελέγχου από και προς τον χρήστη. Τέλος, με την εγκατάσταση του κατάλληλου λογισμικού πραγματοποιεί τη μεταφορά του οπτικού σήματος του εικονολήπτη στον κεντρικό σταθμό σε πραγματικό χρόνο.



Σχήμα 2.3.13 Ηλεκτρονική κατασκευή κεντρικού υπολογιστικού συστήματος κινητής μονάδας

2.4 Περιγραφή κεντρικού σταθμού μετάδοσης δεδομένων

Ο κεντρικός σταθμός μετάδοσης δεδομένων (σχήμα 2.4.1) αποτελείται από τα παρακάτω:
1) Από ένα ή περισσότερους υπολογιστές που δέχονται τα παραπάνω δεδομένα.
2) Από ένα δίαυλο επικοινωνίας από και προς την κινητή μονάδα.
3) Από τον πομποδέκτη του ασύρματου τοπικού δικτύου
4) Από το αντίστοιχο λογισμικό διασύνδεσης που επιτρέπει στο χρήστη να έχει εικόνα και να ελέγχει ανά πάσα χρονική στιγμή τη θέση, το περιβάλλον δράσης και τα πιθανά προβλήματα που μπορεί να προκύψουν, ώστε να μπορεί να λαμβάνει σωστές αποφάσεις για τις επόμενες κινήσεις του ρομποτικού συστήματος.



Σχήμα 2.4.1 Διάγραμμα επιμέρους υπομονάδων κεντρικού σταθμού

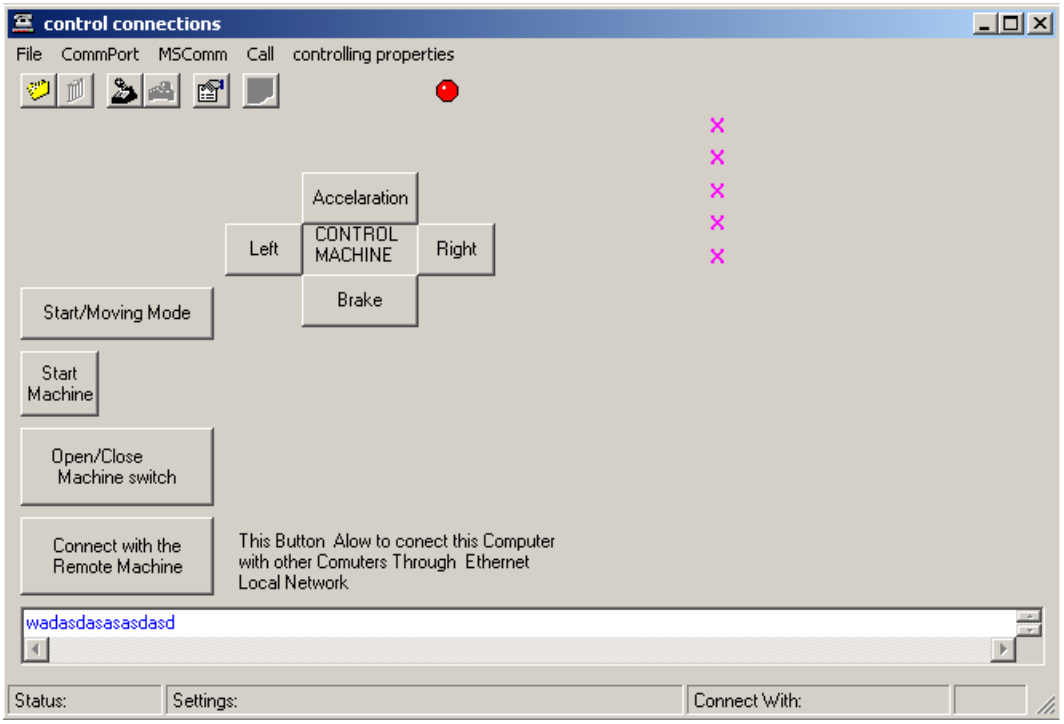
2.5 Διεπαφή χρήστη

Η εφαρμογή ελέγχου του ρομποτικού συστήματος έχει υλοποιηθεί για να μπορέσουμε να ελέγξουμε τις παραμέτρους του ρομποτικού μας συστήματος με εύκολο και γρήγορο τρόπο, με στόχο τον πλήρη απομακρυσμένο έλεγχο του ρομποτικού συστήματος από έναν χρήστη. Η ανάπτυξη και η υλοποίηση όλου του προγραμματισμού πραγματοποιήθηκε σε Visual Studio 6 Enterprise Version.

2.5.1 Έλεγχος ρομποτικού συστήματος

Η παρακάτω φόρμα εργασίας (σχήμα 2.5.1) χρησιμοποιείται για την επικοινωνία του χρήστη με το ρομποτικό μας σύστημα. Το λογισμικό αυτό είναι ενεργό τόσο στο ρομποτικό μας σύστημα όσο και στον κεντρικό σταθμό όπου βρίσκεται ο χρήστης. Ο χρήστης έχει την δυνατότητα να ενεργοποιήσει ή να απενεργοποιήσει το ρομποτικό σύστημα. Μπορεί επίσης να μεταβάλλει την ταχύτητα και τη γωνία διεύθυνσης και τέλος μπορεί να χρησιμοποιήσει τη μονάδα πέδησης όταν απαιτηθεί, για να σταματήσει το ρομποτικό μας σύστημα. Στο παρακάτω σχήμα φαίνεται η συγκεκριμένη υλοποίηση που αναπτύξαμε για τη συγκεκριμένη εφαρμογή απομακρυσμένου ελέγχου του ρομποτικού μας συστήματος.

Ο χρήστης μπορεί να κινήσει το ρομποτικό όχημα είτε με τα ελεγκτήρια εργασίας που φαίνονται παρακάτω, είτε κατευθείαν αν χρησιμοποιήσει το παρακάτω τερματικό κειμένου, ή ακόμα και μέσω χειριστηρίου joystick.

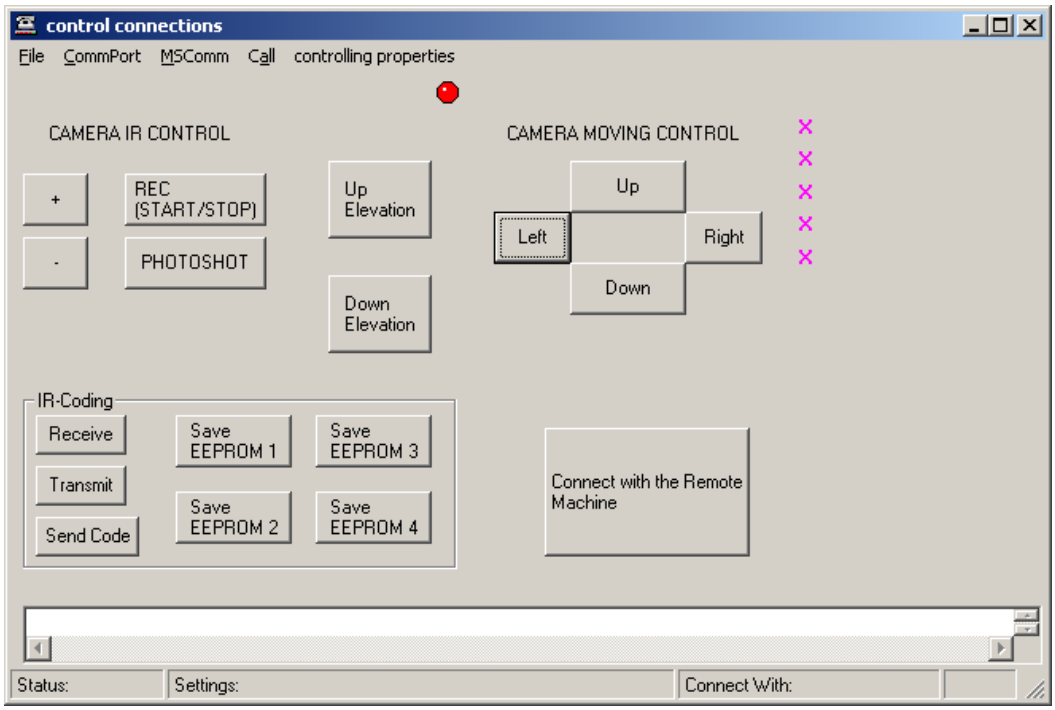


Σχήμα 2.5.1 Λογισμικό ελέγχου κινητής μονάδας

2.5.2 Τηλεχειρισμός συστήματος εικονολήπτη

Αντίστοιχα με τον έλεγχο του ρομποτικού συστήματος έχει υλοποιηθεί λογισμικό ελέγχου του εικονολήπτη και του συστήματος κεφαλής όπου είναι εγκατεστημένο.

Το σύστημα αυτό που ελέγχει δύο κινητήρες της κεφαλής καθώς και τον κινητήρα της κεραίας ανύψωσης. Επίσης ο εικονολήπτης ελέγχεται με την βοήθεια ενός συστήματος αποκωδικοποίησης υπερύθρων. Στο σχήμα 2.5.2 φαίνεται η συγκεκριμένη υλοποίηση.



Σχήμα 2.5.2 Λογισμικό ελέγχου ρομποτικής κεφαλής και εικονολήπτη

2.6 Εφαρμογές

Έχοντας μελετήσει τα επιμέρους υποσυστήματα του ρομποτικού μας συστήματος μπορούμε να δούμε κάποιες εφαρμογές στις οποίες θα ήταν χρήσιμη η συγκεκριμένη υλοποίηση.

Σχήμα 2.6.1 Ρομποτικό σύστημα, κινητή μονάδα, υποσυστήματα και κεντρικός σταθμός

Το εν λόγω ρομποτικό σύστημα μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εφαρμογή απαιτεί χρήση απομακρυσμένου ελέγχου, καταγραφή και επεξεργασία δεδομένων του χώρου στον οποίο εργάζεται για περαιτέρω μελέτη και ανάλυση. Συγκεκριμένα, μπορεί να βρει εφαρμογή στη περιφρούρηση συνόρων και γενικότερα δύσβατων περιοχών, όπου ο άνθρωπος είναι επικίνδυνο να βρίσκεται, καθώς επίσης σε αεροδρόμια, λιμάνια και δασικές εκτάσεις.

Επιπλέον, με την εγκατάσταση πρόσθετων αισθητήρων και ανιχνευτικών συστημάτων, όπως ανιχνευτές laser ή sonar καθώς και με την αντίστοιχη εφαρμογή των απαραίτητων αλγορίθμων μπορεί να χρησιμοποιηθεί στην καταγραφή και χαρτογράφηση περιοχών, δίνοντας επιπρόσθετα την δυνατότητα στο ρομποτικό όχημα να πλοηγηθεί με ακρίβεια και ασφάλεια προς οποιαδήποτε κατεύθυνση.

Κεφάλαιο 3 Πλοήγηση

Με τον όρο πλοήγηση εννοούμε την ασφαλή μεταφορά ενός οποιουδήποτε κινητού μέσου από ένα σημείο σε ένα άλλο. Για την επίτευξη αυτού του στόχου χρησιμοποιείται ένας ειδικός γι' αυτήν τη δουλειά, ο πλοηγός. Ο πλοηγός είναι αυτός που θα αναλάβει να φέρει εις πέρας αυτό το έργο. Στη ναυτιλία ο πλοηγός είναι υπαρκτό πρόσωπο και σε κάθε απόπλου και κατάπλου ενός πλοίου επιτάσσεται να καθορίσει από που και με ποιο τρόπο θα αποπλεύσει ή θα καταπλεύσει ένα πλοίο.

Στην ρομποτική το ρόλο του πλοηγού τον αναλαμβάνει κάποιο λογισμικό, το οποίο λαμβάνει υπόψη του τα χαρακτηριστικά του χώρου όπου έχει υποβληθεί το ρομπότ καθώς επίσης και τα χαρακτηριστικά του ιδίου του ρομπότ, ώστε να μπορεί να σχεδιάσει την διαδρομή που θα ακολουθήσει και να επιλέξει τις κατάλληλες ενέργειες ελέγχου.

3.1 Πλοήγηση και περιβάλλον εργασίας

Όπως αναφέραμε η ανάπτυξη μεθόδων πλοήγησης ρομποτικών συστημάτων δεν είναι δυνατό να αντιμετωπιστεί με την εφαρμογή μιας συγκεκριμένης μεθόδου και τεχνικής. Αντίθετα απαιτείται η ανάπτυξη και εφαρμογή μεθόδων που αναλαμβάνουν να αντιμετωπίσουν ξεχωριστά τα επιμέρους προβλήματα που πρέπει να αντιμετωπιστούν. Ταυτόχρονα όμως είναι αναγκαίο να συνεργάζονται μεταξύ τους μιας και οι αποφάσεις της μίας μεθόδου καθορίζουν τις δράσεις της άλλης, αντιμετωπίζοντας έτσι το συνολικό πρόβλημα και κατ' επέκταση την πλοήγηση του ρομποτικού συστήματος. Γενικότερα, το ρομπότ θα πρέπει να απαντήσει σε μια σειρά από ερωτήματα, για να μπορέσει να πλοηγηθεί επιτυχώς σε ένα τυχαίο και άγνωστο περιβάλλον.

Οι ερωτήσεις που πρέπει να απαντηθούν είναι οι εξής:

1. Τι πρέπει να γνωρίζω;
2. Που βρίσκομαι;
3. Από που θα πάω;
4. Πως θα πάω;

Κάθε μία από τις παραπάνω ερωτήσεις αποτελεί ένα ξεχωριστό πεδίο έρευνας και μελέτης στα ρομποτικά συστήματα. Έτσι για να είναι δυνατή η απάντηση τους, πρέπει να προβούμε στην περιγραφή συγκεκριμένων τεχνικών που πρέπει να ακολουθηθούν για να επιτευχθεί το επιθυμητό αποτέλεσμα. Αναλυτικά, με βάση τα παραπάνω ερωτήματα έχουμε:

- Χαρτογράφηση (mapping): Ουσιαστικά, με τη τεχνική αυτή δημιουργείται ένας χάρτης στον οποίο κωδικοποιείται και αποθηκεύεται όλη η απαραίτητη πληροφορία που απαιτείται για την αναπαράστασή του περιβάλλοντα χώρου (εμπόδια, όρια κίνησης κλπ) του ρομποτικού συστήματος.
- Εντοπισμός (localization): Το ρομπότ πρέπει να εφοδιαστεί με τους απαραίτητους αλγόριθμους, αλλά και αισθητήρια όργανα ώστε να μπορεί ανά πάσα χρονική στιγμή να προσδιορίζει την τρέχουσα θέση του μέσα στο χάρτη.

- Σχεδιασμός διαδρομής (path planning): Γνωρίζοντας τη θέση του ρομπότ μέσα στο χάρτη και με τη βοήθεια του κατάλληλου αλγορίθμου, πρέπει να επιλέξει από ένα σύνολο (ασφαλών, άνευ εμποδίων) δυνατών διαδρομών τη συντομότερη, με γνώμονα τα κριτήρια που επιβάλει ο εν λόγω αλγόριθμος.
- Έλεγχος κίνησης (motion control): Έχοντας αποφασίσει για τη διαδρομή που θα ακολουθήσει το ρομπότ, θα πρέπει με τη βοήθεια της κατάλληλης μεθόδου, να αποφασίσει για τις τιμές των παραμέτρων των ελεγχόμενων βαθμών ελευθερίας του ρομποτικού συστήματος, καθοδηγώντας το ανά πάσα χρονική στιγμή προς τον επιθυμητό στόχο κατά μήκος της σχεδιασμένης διαδρομής.

Περιβάλλον εργασίας

Τα ρομποτικά συστήματα σχεδιάζονται για να λειτουργήσουν σε φυσικά περιβάλλοντα εργασίας. Ως περιβάλλον εργασίας (work space/environment) ορίζεται το πραγματικό φυσικό πεδίο στο οποίο το ρομποτικό σύστημα έχει κληθεί για να εκτελέσει τις επιθυμητές εργασίες. Είναι φανερό ότι η θέση του στον περιβάλλοντα χώρο σε σχέση με ένα σημείο αναφοράς, απαιτείται να καθοριστεί με ακρίβεια. Ο χώρος διατάξεων (configuration space), είναι ο χώρος όλων των πιθανών διατάξεων του ρομπότ μέσα στο περιβάλλον. Αυτό που έχει σημασία εδώ είναι η αντιστοίχηση του χώρου διατάξεων με το χώρο των βαθμών ελευθερίας. Πολλές φορές οι δύο αυτοί χώροι έχουν διαφορετική διάσταση με αποτέλεσμα η τοποθέτηση του ρομπότ σε μια επιθυμητή διάταξη να απαιτεί πολύπλοκο έλεγχο.

Σφαιρική και τοπική αναζήτηση

Η πλοήγηση των ρομποτικών συστημάτων μπορεί να χωριστεί σε δύο βασικές κατηγορίες: στη σφαιρική πλοήγηση (global navigation) και στην τοπική πλοήγηση (local navigation).

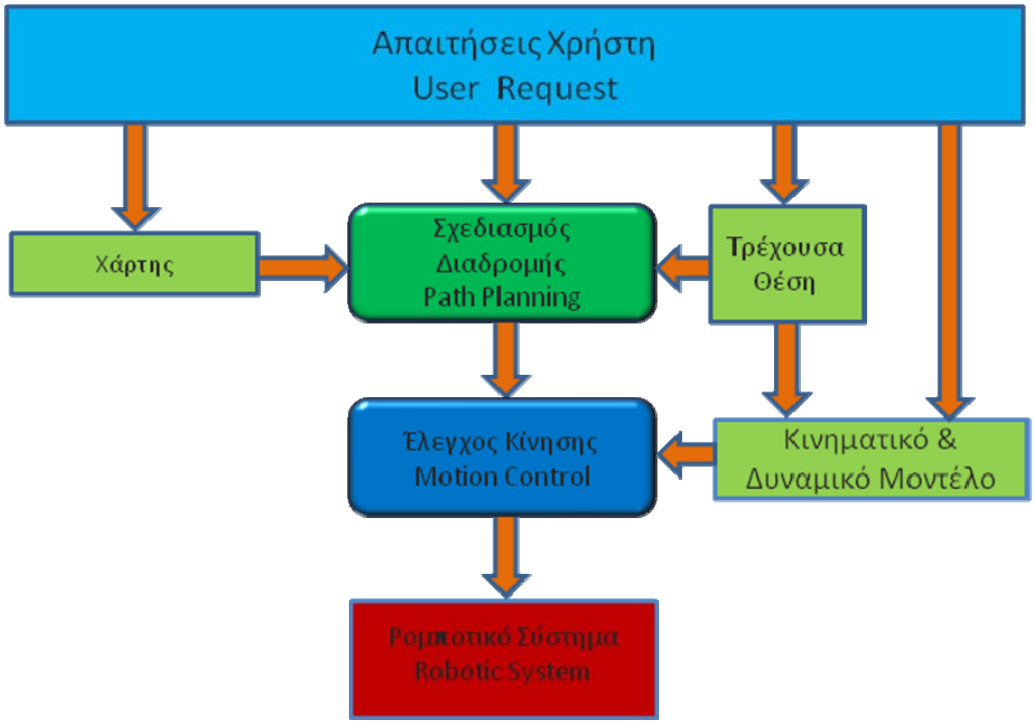
Στην πρώτη κατηγορία πρέπει να συγκεντρώσουμε πληροφορίες για το χώρο εργασίας. Η χαρτογράφηση είτε πραγματοποιείται με τη βοήθεια εξωτερικών αισθητήρων στον χώρο, είτε από τον ίδιο τον χρήστη. Στην συνέχεια καταχωρούνται σε έναν χάρτη (map) όπου με βάση αυτόν το ρομπότ θα αποφασίσει τη διαδρομή που πρέπει να ακολουθήσει. Η περίπτωση αυτή γενικότερα θεωρεί το περιβάλλον κίνησης του ρομποτικού μας συστήματος από ένα ανώτερο επίπεδο, κατά συνέπεια εάν υπάρχει λύση για τον επιθυμητό στόχο τότε το ρομποτικό σύστημα θα μπορεί να τη βρει.

Απ' την άλλη, στην τοπική πλοήγηση το ρομπότ λαμβάνει πληροφορίες για το περιβάλλον του μόνο από τα αισθητήρια όργανα που διαθέτει. Κατά συνέπεια το εύρος της πληροφορίας και η ακρίβεια των δεδομένων του περιβάλλοντος περιορίζεται στην ακρίβεια και το εύρος των αισθητήριων οργάνων που έχουν εγκατασταθεί στο ίδιο το ρομποτικό μας σύστημα. Έτσι δεν είναι δυνατό πάντα να υπολογιστεί κατάλληλη διαδρομή για ένα απομακρυσμένο στόχο.

Επίπεδα Ελέγχου

Στην εργασία αυτή θα ασχοληθούμε με το σχεδιασμό διαδρομής και τον έλεγχο κίνησης. Θεωρούμε ότι ο χάρτης καταχώρησης των δεδομένων του περιβάλλοντος και η θέση του ρομποτικού συστήματος σ' αυτόν είναι γνωστά. Τα δεδομένα για το χώρο εργασίας τα δημιουργεί ο χρήστης και τα εισάγει στη μνήμη του ρομπότ. Κατά

συνέπεια, ο τρόπος υλοποίησης μπορεί να διαιρεθεί σε τέσσερα επίπεδα, όπως φαίνεται στο σχήμα 3.1.1.



Σχήμα 3.1.1 Επίπεδα ελέγχου

Τα επίπεδα ελέγχου εκτελούν τις παρακάτω λειτουργίες:

- Απαιτήσεις χρήστη: Ο χρήστης αποφασίζει για το χάρτη, τη θέση εκκίνησης/τερματισμού, καθώς επίσης και για τις παραμέτρους που απαιτούνται να καθοριστούν σε κάθε επίπεδο της πλοήγησης.

- Σχεδιασμός διαδρομής: Το σύστημα αποφασίζει, λαμβάνοντας όλες τις απαραίτητες παραμέτρους που καθορίζονται από το χρήστη, την τελική διαδρομή που θα ακολουθήσει.

- Ελεγκτής κίνησης: Με βάση το μονοπάτι που θα αποφασιστεί από το επίπεδο σχεδιασμού της διαδρομής, ο ελεγκτής κίνησης θα παράγει τις ακριβείς τιμές των συστημάτων ελέγχου του ρομπότ ανά πάσα χρονική στιγμή, ικανοποιώντας τους περιορισμούς που θέτουν το κινηματικό και δυναμικό μοντέλο του οχήματος.

- Ρομποτικό σύστημα: Οι τιμές που παράγονται από τον ελεγκτή κίνησης δίνονται στους αντίστοιχους ηλεκτρονικούς μετατροπείς της μηχανής για να κινήσει τους αντίστοιχους σερβομηχανισμούς και κατ' επέκταση το ίδιο το ρομπότ.

3.2 Σχεδιασμός διαδρομής

Όπως γνωρίζουμε ένα μεγάλο μέρος της επιστήμης υπολογιστών ασχολείται με μεθόδους αναζήτησης. Οι μέθοδοι που ακολουθούνται ποικίλλουν ανάλογα με την περίπτωση και την εφαρμογή. Το πρόβλημα της αναζήτησης της διαδρομής που θα πρέπει να ακολουθήσει το ρομπότ δεν είναι ένα απλό υπολογιστικό πρόβλημα. Για να γίνει δυνατή η σχεδίαση του μονοπατιού που μπορεί να ακολουθηθεί, χρησιμοποιούνται τεχνικές που στόχο έχουν να ελαχιστοποιήσουν το χώρο, έτσι ώστε να είναι δυνατή η εξέτασή του. Για την επίλυση του συγκεκριμένου προβλήματος υπάρχει ανεξάντλητη ποσότητα πληροφορίας, μιας και από τη δεκαετία του '80 έχουν δοθεί κάποιες προσεγγίσεις επίλυσης. Στο κείμενο που ακολουθεί παρουσιάζονται κάποιες βασικές μέθοδοι σχεδιασμού διαδρομής.

3.2.1 Οδικός χάρτης

Η μέθοδος αυτή στόχο έχει να ελαχιστοποιήσει τις διαστάσεις του χώρου εργασίας του ρομπότ. Αυτό επιτυγχάνεται με τη διασύνδεση του ελεύθερου χώρου κίνησης του ρομπότ με μονοδιάστατες καμπύλες. Η κατασκευή αυτή ονομάζεται οδικός χάρτης (roadmap) και το αποτέλεσμα αυτής της μεθόδου δίνει ένα ολοκληρωμένο μονοπάτι με τη μορφή γράφου που αποτελείται από τρία επιμέρους μονοπάτια:

1. Ένα αρχικό μονοπάτι διασυνδέει το αρχικό σημείο που βρίσκεται το ρομποτικό σύστημα $\mathbf{q_{start}}$ με το κοντινότερο σ' αυτό σημείο του οδικού χάρτη $\mathbf{q'}$.
2. Το βασικό μονοπάτι το οποίο ξεκινάει από το $\mathbf{q'}$ του οδικού χάρτη και φθάνει στο σημείο $\mathbf{q''}$ του οδικού χάρτη.
3. Το τελικό μονοπάτι που συνδέει το σημείο τερματισμού $\mathbf{q_{end}}$ με το κοντινότερο σ' αυτό, σημείο του οδικού χάρτη $\mathbf{q''}$.

Διαδεδομένες τεχνικές αυτής της μεθόδου είναι τα διαγράμματα Voronoi και οι γράφοι ορατότητας (visibility graphs). Σ' αυτές τις περιπτώσεις το πρόβλημα είναι η κατασκευή του χάρτη διαδρομής (υπολογιστικά δύσκολο) και ο τρόπος που μπορούν να τροποποιηθούν για εφαρμογή σε δυναμικά περιβάλλοντα [2].

3.2.2 Γεωμετρική προσέγγιση

Αυτή η προσέγγιση μοντελοποιεί το περιβάλλον εργασίας, αλλά και το ίδιο το ρομπότ κάνοντας χρήση γεωμετρικών δομών (τετράγωνα, κύκλους, κλπ) [3]. Με τη βοήθεια των δομών αυτών η μέθοδος εκτελεί τις απαραίτητες περιστροφές και μετατοπίσεις των συγκεκριμένων γεωμετρικών αντικειμένων, ώστε το ρομποτικό σύστημα να οδηγηθεί στον επιθυμητό προορισμό. Ένα πολύ γνωστό πρόβλημα αυτής της κατηγορίας είναι το πρόβλημα του «piano movers», το οποίο εξετάζει τις περιστροφές που πρέπει να γίνουν σε ένα πιάνο για να μετακινηθεί έξω από ένα δωμάτιο μέσω μιας στενής πόρτας. Οι μέθοδοι αυτοί εξαιτίας του κόστους υπολογισμού, σπάνια εφαρμόζονται σε πραγματικά ρομποτικά συστήματα, ωστόσο έχουν μελετηθεί σε βάθος [4][5].

3.2.3 Προσέγγιση διάσπασης κελιών

Η μέθοδος αυτή διακριτοποιεί τον χάρτη κίνησης του ρομπότ σε ένα μεγάλο αριθμό από μικρότερες περιοχές που τις καλούμε κελιά (shells). Συγκεκριμένα, κάθε δυνατή κίνηση του ρομπότ από το ένα κελί στο γειτονικό του μπορεί να πραγματοποιηθεί με βάση τα χαρακτηριστικά και τις δυνατότητες κίνησης του ρομπότ. Η συνολική διαδρομή που πρέπει να ακολουθήσει το ρομπότ για να μεταβεί μέσω των κελιών αυτών σε ένα επιθυμητό σημείο, αναπαριστάται σαν ένας μη κατευθυνόμενος γράφος. Το μέγεθος των κελιών μπορεί να μεταβάλλεται ανάλογα με τις απαιτήσεις και ο υπολογισμός του μονοπατιού μπορεί να πραγματοποιηθεί από οποιοδήποτε αλγόριθμο αναζήτησης σε γράφο. Αυτή η κατηγορία έχει πιθανώς ερευνηθεί πολύ περισσότερο από οποιαδήποτε άλλη, μιας και η πολυπλοκότητά της σχετίζεται άμεσα με τον αλγόριθμο αναζήτησης που μπορεί να χρησιμοποιηθεί [6][7][8][9][10]. Λόγω της γενικότητας αυτών των τεχνικών, επιλέξαμε να χρησιμοποιήσουμε μια μέθοδο διακριτοποίησης και μια τεχνική αναζήτησης για το σχεδιασμό διαδρομής στο σύστημά μας.

3.2.4 Ανακλαστική προσέγγιση

Αυτή η μέθοδος εστιάζει σε τεχνικές που στόχο έχουν την αντιμετώπιση του υπολογιστικού κόστους, δηλαδή την κατασκευή γρήγορων λύσεων σε εφαρμογές πραγματικού χρόνου. Η ιδέα είναι να συνδεθεί η αντίληψη απευθείας με τη δράση. Τέτοια σύνδεση μπορεί να πραγματοποιηθεί με τη μορφή συμπεριφορών που απεικονίζουν αντιλήψεις και δράσεις και να υλοποιηθεί με μια απλή δομή, π.χ πεπερασμένα αυτόματα [11][12]. Παρόλο που οι δυνατότητες της μεθόδου αυτής φαίνονται περιορισμένες, ο συνδυασμός πολλών κανόνων συμπεριφοράς παράλληλα, θα έχει ως αποτέλεσμα η συνολική συμπεριφορά του ρομποτικού συστήματος να είναι ένας συνδυασμός των επιμέρους συμπεριφορών. Άλλες τεχνικές ανακλαστικών ελεγκτών επιτυγχάνονται με τη χρήση ασαφούς λογικής (fuzzy logic) και νευρωνικών δικτύων (neural networks) [13][14][15][16][17]. Μια γρήγορη μηχανή ασαφούς λογικής σε συνδυασμό με ένα άρτια εκπαιδευμένο νευρωνικό δίκτυο μπορούν να δώσουν άμεση και ομαλή ανταπόκριση στη συμπεριφορά του ρομπότ, ιδιαίτερα σε γρήγορες απότομες μεταβολές στο περιβάλλον εργασίας. Τα τελευταία χρόνια η τεχνική του μοντέλου των συμπεριφορών γίνεται ολοένα και πιο δημοφιλής, μιας και έχει δώσει μια εντελώς διαφορετική οπτική γωνία στην κατασκευή έξυπνων ρομποτικών συστημάτων. Ωστόσο, οι δυνατότητες αυτών των τεχνικών είναι περιορισμένες, διότι οι αλληλεπιδράσεις μεταξύ συμπεριφορών ελέγχονται δύσκολα.

3.2.5 Εικονικά πεδία δυνάμεων

Η μέθοδος αυτή καθορίζει τον τρόπο που δρουν κάποια εικονικά πεδία δυνάμεων πάνω στο χώρο εργασίας του ρομπότ, έτσι ώστε το αποτέλεσμα της δράσης αυτής να είναι ένας συνδυασμός δυνάμεων πάνω στο ρομπότ που το οδηγούν στο επιθυμητό σημείο. Γενικότερα, υποθέτουμε ότι το σημείο που θέλουμε να φτάσουμε εφαρμόζει στο ρομπότ μια ελκτική δύναμη, ενώ τα εμπόδια μια απωθητική. Αθροίζοντας τα επιμέρους διανύσματα όλων των δυνάμεων, θα οδηγηθεί το ρομπότ προς το συνολικά

παραγόμενο διάνυσμα. Το αποτέλεσμα είναι να μετακινήσουμε το ρομπότ προς τον στόχο και παράλληλα να αποφύγουμε τα εμπόδια. Δυστυχώς, σε κάποιες περιπτώσεις το άθροισμα του συνδυασμού των δυνάμεων αλληλοαναιρείται, κατά συνέπεια το ρομποτικό σύστημα οδηγείται σε αδιέξοδο. Αναλυτικά παραδείγματα της τεχνικής αυτής έχουν μελετηθεί από τους Oussama Khatib [18] και Johann Borenstein [19].

3.2.6 Δυναμικά πεδία

Με τη μέθοδο αυτή η πλοήγηση μπορεί να περιγραφεί σαν μια διαδικασία που ακολουθεί το μέγιστο άνυσμα σε κάποιο ιδεατό πεδίο δυναμικού μέσα στο περιβάλλον του ρομπότ. Η φύση, τα ζώα ή ακόμα και ο άνθρωπος εφαρμόζουν αντίστοιχες τεχνικές για να προσανατολιστούν. Για παράδειγμα, όπως ένας άνθρωπος μπορεί να κατευθυνθεί από ένα ήχο και να τον ακολουθήσει καθώς ενισχύεται μέχρι να προσδιορίσει την πηγή του, έτσι αντίστοιχα κάποια ζώα κατευθύνονται με βάση τα ένστικτα και τα αισθητήρια όργανα που διαθέτουν προς κάποιο ερέθισμα (νερό, φαγητό, ήχους, κλπ). Στα ρομποτικά συστήματα η μέθοδος αυτή υπολογίζει ένα άνυσμα με τη βοήθεια των αισθητήρων του ρομπότ και με βάση το ερέθισμα που δέχεται. Έτσι, το ρομπότ απλά ακολουθεί κάθε χρονική στιγμή το μέγιστο παραγόμενο άνυσμα το οποίο εντέλει το οδηγεί στο στόχο. Τα δυναμικά πεδία επιλύουν ένα μεγαλύτερο εύρος του προβλήματος της πλοήγησης σε σχέση με τις άλλες μεθόδους. Ο τρόπος αυτός λαμβάνει χώρα στο συνολικό περιβάλλον του ρομπότ και το αποτέλεσμα που προκύπτει είναι ένα διάνυσμα κατεύθυνσης σε κάθε θέση του χώρου εργασίας. Η τεχνική αυτή είναι πιθανόν από τις πιο δημοφιλείς στην πλοήγηση ρομποτικών συστημάτων [20][21][22].

3.3 Έλεγχος κίνησης

Όπως αναφέραμε, η κίνηση των ρομποτικών συστημάτων περιλαμβάνει το σχεδιασμό διαδρομής σε υψηλότερο επίπεδο και τον έλεγχο κίνησης σε χαμηλότερο. Η μέθοδος σχεδιασμού διαδρομής ασχολείται με την αναζήτηση της διαδρομής που πρέπει να ακολουθήσει το σύστημα και ο ελεγκτής κίνησης με τον τρόπο που θα ακολουθήσει το ρομπότ τη διαδρομή αυτή. Έτσι ο υπολογισμός του μονοπατιού θα καθοριστεί με βάση τα εμπόδια και τον χώρο κίνησης του συστήματος, δίδοντας σαν αποτέλεσμα το συντομότερο μονοπάτι ελεύθερο εμποδίων. Ο ελεγκτής κίνησης θα λάβει αυτά τα σημεία και θα παράγει τις απαραίτητες τιμές για τα συστήματα ελέγχου του ρομπότ. Ο έλεγχος κίνησης του ρομπότ πρέπει να γίνεται με κατάλληλο τρόπο ώστε οι παράμετροι ελέγχου σε κάθε χρονική στιγμή να καθορίζονται με την καλύτερη δυνατή ακρίβεια. Οι κινηματικές εξισώσεις του ρομπότ είναι αυτές που καθορίζουν την επομένη θέση-διεύθυνση του ρομπότ σε κάθε χρονική στιγμή. Οι τεχνικές που ακολουθούνται είναι ποικίλλες και διαφοροποιούνται ανάλογα με την εφαρμογή και το ρομπότ για το οποίο προορίζονται. Γενικότερα, καμία μέθοδος δεν μπορεί να μας εγγυηθεί ότι το ρομπότ θα καταφέρει να φτάσει ακριβώς στον στόχο. Ας δούμε μερικές προσεγγίσεις που ακολουθούνται με έμφαση στα κινητά μη ολονομικά ρομποτικά συστήματα.

3.3.1 Μηδενικά συστήματα

Από τις πρώτες μεθόδους που μελετήθηκαν στο σχεδιασμό μη ολονομικών συστημάτων ελέγχου κίνησης, είναι αυτή των μηδενικών συστημάτων (Nilpotent systems). Θεωρώντας σύστημα STLC (Small Time Local Controllability) και με τη βοήθεια της Lie Algebra παράγουμε μια βάση $\boldsymbol{B}$ (Philip Hall Basis) [30]. Επιπλέον αν υποθέσουμε ότι το μονοπάτι $\boldsymbol{\gamma}$ είναι γνωστό και εκφράσουμε τα διανύσματα $\tan(\dot{\gamma}(t))$ ως γραμμικό συνδυασμό των διανυσμάτων στην $\boldsymbol{B(\gamma(t))}$ τότε οι συντελεστές που προκύπτουν ελέγχουν το ρομποτικό σύστημα, ώστε να κινηθεί κατά μήκος του μονοπατιού $\boldsymbol{\gamma}$. Οι εμπνευστές αυτής της μεθόδου Lafferiere και Syssman [23][24][25], θεώρησαν ότι αν ένα σύστημα είναι μηδενικό, τότε η μέθοδος αυτή μπορεί να υπολογίσει τιμές για τις παραμέτρους του ρομπότ που θα οδηγήσουν ακριβώς πάνω στον στόχο. Αλλιώς η μέθοδος οδηγείται σε μια προσεγγιστική λύση που θα δώσει ως αποτέλεσμα μια λύση πολύ κοντά στον στόχο [26]. Βέβαια με τη χρήση αναδράσεων στο σύστημα μπορούμε να δημιουργήσουμε συνθήκες ώστε να μπορεί ένα μη μηδενικό σύστημα να μετατραπεί σε μηδενικό [27].

3.3.2 Ημιτονοειδής τροχιά δράσης

Η μέθοδος της ημιτονοειδούς τροχιάς δράσης (Sinusoidal Action Trajectories) στηρίζεται στην αρχική δουλειά του Brockett και στην περαιτέρω ανάλυση από τους Murray και Sastry [28][29]. Η μέθοδος παράγει άμεσα, ανεξάρτητα διανυσματικά πεδία και τα παραγόμενα διανύσματα εφαρμόζονται σε διάφορα επίπεδα του συστήματος. Έτσι, με την κατασκευή της βάσης $\boldsymbol{B}$ (Philip Hall Basis) και λαμβάνοντας υπόψη τη γραμμική εξάρτηση των προηγούμενων αποφάσεων, κανένα στοιχείο των προηγούμενων διανυσματικών πεδίων δε χρειάζεται να διαγραφεί. Η προσέγγιση αυτή εφαρμόζει καμπύλες ημιτόνου ή σχετικές συχνότητες σε κάποιες από τις μεταβλητές ελέγχου [30].

3.3.3 Επίπεδα συστήματα

Ο έλεγχος κίνησης στην περίπτωση αυτή πραγματοποιείται, εάν καθοριστεί ο τρόπος που θα οριστεί ένα επίπεδο σύστημα (flat system). Ως επίπεδο σύστημα ορίζεται, ένα σύστημα που αποτελείται από ένα πεπερασμένο σε αριθμό διαφορικά ανεξάρτητων μεταβλητών $\mathbf{y_i}$, που εμφανίζονται σαν διαφορικές συναρτήσεις του συστήματος μεταβλητών (μεταβλητές κατάστασης και εισόδων) και από ένα πεπερασμένο αριθμό των παραγώγων τους. Κάθε μεταβλητή συστήματος είναι από μόνη της μια συνάρτηση των μεταβλητών $\mathbf{y_i}$ και των παραγώγων τους. Οι μεταβλητές $\mathbf{y_i}$ καλούνται ως γραμμικές έξοδοι του συστήματος. Η μέθοδος αυτή έχει μελετηθεί από τους Fliess, Levine, Marine και Rouchon [31][32].

3.3.4 Βέλτιστος έλεγχος

Οι μέθοδοι του βέλτιστου ελέγχου είναι από τις πρώτες μεθοδολογίες που έχουν μελετηθεί και εφαρμοστεί στη σχεδίαση και ανάπτυξη ελεγκτών, ιδιαίτερα για τα κινητά ρομποτικά μοντέλα. Ο τρόπος ελέγχου των παραμέτρων του ρομποτικού συστήματος καθορίζεται μέσω μιας αντικειμενικής συνάρτησης, η οποία επιλέγοντας τους κατάλληλους συνδυασμούς τιμών των μεταβλητών ελέγχου, ελαχιστοποιείται. Μια τέτοια μέθοδο έχουν μελετήσει οι J. Latombe και J. Laumond [33][34]. Οι μέθοδοι αυτοί έχουν εκτενέστατα μελετηθεί κάτω από διαφορετικές συνθήκες και με διαφορετικούς τρόπους. Γενικά, αυτό που έχει σημασία είναι η μαθηματική συνάρτηση πού πρέπει να επιλεχθεί για να μπορεί το ρομπότ σε κάθε χρονική στιγμή να λαμβάνει σωστές τιμές ελέγχου.

Μια εφαρμογή της συγκεκριμένης τεχνικής, κάνοντας χρήση της μεθόδου του δυναμικού παραθύρου στο ρομποτικό μοντέλο του μονόκυκλου, δίνεται από τον Δρ. Μιχαήλ Λαγουδάκη [35]. Σε αυτή στηριχτήκαμε για να υλοποιήσουμε τη δική μας εκδοχή για το ρομποτικό αυτοκίνητο.

Κεφάλαιο 4 Η δική μας προσέγγιση πλοήγησης

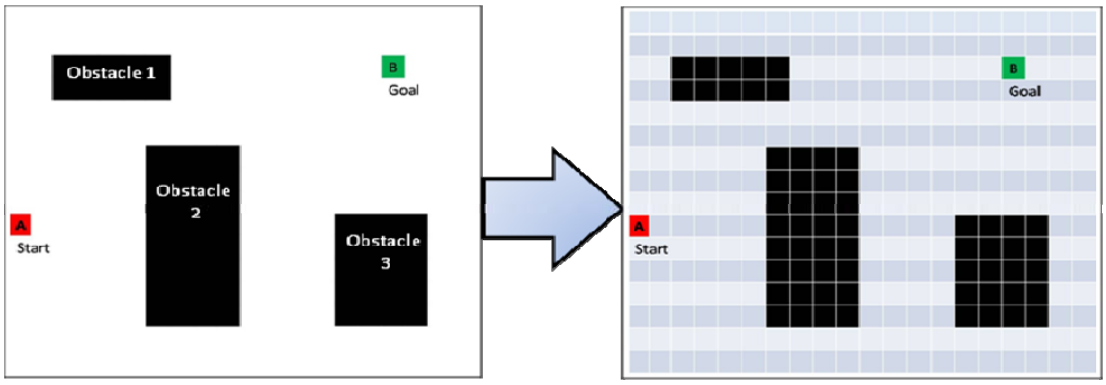
Όπως αναφέραμε στο κεφάλαιο 2 το μοντέλο του ρομπότ για το οποίο θέλουμε να επιλύσουμε το πρόβλημα της πλοήγησης είναι ένα όχημα με τέσσερις τροχούς και δυνατότητα αλλαγής πορείας μόνο από τις μπροστινές ροδές (όπως ένα αυτοκίνητο). Γνωρίζουμε ότι το ρομποτικό μοντέλο του αυτοκινήτου είναι στην κατηγορία των μη ολονομικών ρομπότ και οι βαθμοί ελευθερίας που μπορούμε να ελέγξουμε είναι δύο, η διεύθυνση των μπροστινών τροχών και η γραμμική ταχύτητα του οχήματος (ενότητα 2.2).

Με βάση τα χαρακτηριστικά κίνησης του ρομποτικού αυτοκινήτου και αντιμετωπίζοντας τις μεθόδους σχεδίασης διαδρομής και του ελεγκτή κίνησης σε διαφορετικά επίπεδα, ένας αλγόριθμος σχεδίασης της διαδρομής (Path Planner) A* αποφασίζει για την βέλτιστη διαδρομή που θα πρέπει να ακολουθήσει το ρομποτικό σύστημα για να προσεγγίσει τον επιθυμητό στόχο. Οφείλει να λάβει βεβαία υπόψη του όλα τα πιθανά εμπόδια που μπορεί να υπάρχουν στην διαδρομή του, εξασφαλίζοντας ότι η τελική διαδρομή θα είναι πρώτον χωρίς εμπόδια και δεύτερον η συντομότερη δυνατή. Στην συνέχεια ο ελεγκτής κίνησης (motion controller) με τη βοήθεια μιας αντικειμενικής συνάρτησης (objective function) και του δυναμικού παραθύρου (dynamic window) αποφασίζει ανά πασά χρονική στιγμή, τις απαραίτητες τιμές για τα συστήματα ελέγχου (γωνία στροφής, γραμμική ταχύτητα) του ρομποτικού συστήματος.

4.1 Σχεδιασμός διαδρομής

Στο κεφάλαιο 3 αναλύσαμε την προσέγγιση διάσπασης κελιών. Η μεθοδολογία που ακολουθείται συνήθως σ' αυτές τις περιπτώσεις είναι ο δυναμικός προγραμματισμός και η πληροφορημένη αναζήτηση. Στις παραγράφους που ακολουθούν θα αναλύσουμε τον τρόπο λειτουργίας ενός πολύ γνωστού και ταυτόχρονα αποδοτικού αλγορίθμου που καταφέρνει να επιλύει τέτοια προβλήματα με σχετικά μικρή πολυπλοκότητα, του A* (Astar). Ο A* ή Αλγόριθμος Αναζήτησης με βάση το βέλτιστο κόστος (Best-First Search for Optimal Paths) είναι από τους πιο αποδοτικούς αλγόριθμους σ' αυτού του είδους τα προβλήματα [10][37]. Ας περιγράψουμε λοιπόν τα στάδια της λειτουργίας του.

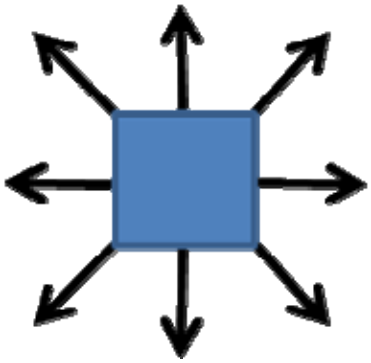
Αυτό που μας ενδιαφέρει είναι να μεταφέρουμε το ρομποτικό σύστημα μας από ένα σημείο A (σημείο εκκίνησης) σε ένα σημείο B (σημείο τερματισμού), λαμβάνοντας υπόψη όλα τα χαρακτηριστικά του περιβάλλοντος. Αρχικά πρέπει να διαλέξουμε με ποιο τρόπο θα διακριτοποιήσουμε το χώρο επιλέγοντας σε κάθε βήμα του αλγόριθμου το εύρος κίνησης που θέλουμε. Στο σχήμα 4.1.1 φαίνεται ένα στιγμιότυπο χώρου εργασίας του ρομποτικού συστήματος πριν και μετά τη διακριτοποίηση.



Σχήμα 4.1.1 Διακριτοποίηση του χώρου εργασίας του ρομποτικού συστήματος

Αφού διακριτοποιήσουμε το χώρο με ένα πλέγμα από μικρά κελιά μεγέθους **d**, μπορούμε να προχωρήσουμε στην εξερεύνηση του, έχοντας ως στόχο το τελικό σημείο τερματισμού, στο οποίο θα προσπαθήσει ό αλγόριθμος να φτάσει δημιουργώντας τη συντομότερη δυνατή διαδρομή λαμβάνοντας επιπλέον υπόψη τα πιθανά εμπόδια. Ο αλγόριθμος θα εξετάζει πιθανά κελιά σε κάθε βήμα ελέγχου με βάση το ελάχιστο κόστος. Σε κάθε κελί που εξετάζεται δημιουργείται ένας κόμβος του οποίου το κόστος υπολογίζεται με βάση τα κριτήρια του A*.

Επιπλέον, με βάση τα κινηματικά χαρακτηριστικά του ρομποτικού συστήματος αλλά και τη διακριτοποίηση που επιθυμούμε, ορίζουμε το μέγιστο αριθμό προσανατολισμών που επιτρέπεται να έχει κάποιος κόμβος. Έτσι, για το ρομποτικό μοντέλο του αυτοκίνητου, όπου η γωνία στροφής του τιμονιού μπορεί να πάρει τιμές από -45° έως 45°, θεωρούμε διακριτοποίηση ως προς την κατεύθυνση που ο κάθε κόμβος είναι δυνατό να βρεθεί ανά 45°. Έτσι, ο συνολικός αριθμός των διαφορετικών προσανατολισμών που επιτρέπεται είναι οκτώ (σχήμα 4.1.2).

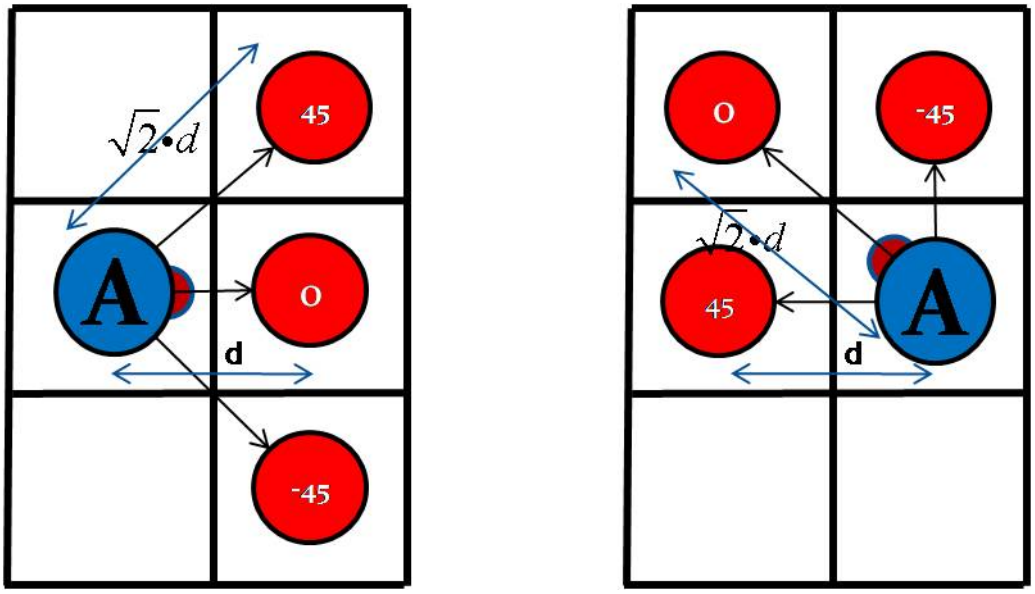


Σχήμα 4.1.2 Οκτώ συνολικά επιτρεπόμενοι προσανατολισμοί

Ωστόσο, από κάθε κόμβο ο αλγόριθμος επιτρέπει μόνο τρείς δυνατές μεταβάσεις εκ των οκτώ δυνατών επιλογών προσανατολισμού. Έτσι, από το κόμβο στον οποίο βρίσκεται, εξετάζει μόνο τους τρείς πιθανούς επόμενους κόμβους στους οποίους το ρομπότ επιτρέπεται να μετακινηθεί. Ξεκινώντας από τον κόμβο που αντιστοιχεί στην αρχική θέση του ρομποτικού συστήματος (ρίζα) δημιουργείται ένα δέντρο αναζήτησης, επεκτείνοντας το από καινούριους κόμβους που δεν έχουν ακόμα εξεταστεί.

Θεωρώντας σαν επιτρεπόμενη κίνηση μόνο προς τα εμπρός (δεν υποστηρίζεται κίνηση προς τα πίσω από το εν λόγω ρομποτικό σύστημα), οι πιθανοί επόμενοι

κόμβοι θα είναι τα τρία γειτονικά κελιά: ευθεία, αριστερά και δεξιά ως προς τον τρέχοντα προσανατολισμό του οχήματος (-45° ,0 ,45°) όπως φαίνεται και στο σχήμα 4.1.3.



Σχήμα 4.1.3 Επιτρεπτές μεταβάσεις στο διακριτοποιημένο χώρο.

Στην συνέχεια ο αλγόριθμος θα εξαιρέσει από την λίστα των πιθανών προς αναζήτηση επόμενων κόμβων, τον κόμβο που βρίσκεται την συγκεκριμένη χρονική στιγμή. Θα επιλέξει έναν από τους κόμβους του δέντρου αναζήτησης που δεν έχει εξετάσει και θα δημιουργήσει τους πιθανούς επομένους κόμβους. Ο τρόπος με τον οποίο θα επιλέξει από την λίστα των κόμβων γίνεται με βάση το εκτιμώμενο συνολικό κόστος. Οπότε ο κόμβος με το ελάχιστο εκτιμώμενο κόστος θα είναι εκείνος που θα αναζητήσουμε, στη συνέχεια.

Το εκτιμώμενο συνολικό κόστος κάθε υποψήφιου κόμβου **n** υπολογίζεται ως εξής:

$$\mathrm{Cost}(n) = g(n) + h(n)$$

$$g(n) = g(n') + c(n', n)$$

$$h(n) = \sqrt{(x_{destination} - x^{n}_{current})^2 + (y_{destination} - y^{n}_{current})^2}$$

g(n΄): Είναι το πραγματικό κόστος κίνησης που υπολογίζεται αθροίζοντας τα επιμέρους κόστη **c(n, n΄)** από το αρχικό σημείο μέχρι το σημείο πού βρίσκεται τη συγκεκριμένη χρονική στιγμή.

c(n΄, n): Είναι το ελάχιστο κόστος κίνησης του ρομπότ από τον πάτερα κόμβο **n΄** προς τον τρέχοντα κόμβο **n**. Η τιμή του ισούται με το μήκος που διανύει το ρομπότ για να εκτελέσει ένα βήμα από τον κόμβο **n** στον **n΄**. Συγκεκριμένα, για οριζόντια ή κάθετη μετάβαση, διανύει απόσταση **d** (το μέγεθος του κελιού) άρα **c(n΄, n) = d** ενώ $\mathbf{c(n',n) = \sqrt{2} \bullet d}$ όταν κινείται διαγώνια (σχήμα 4.1.3).

h(n): Είναι η πιο αισιόδοξη εκτίμηση του εναπομείναντος κόστους το οποίο υπολογίζεται με βάση την Ευκλίδεια απόσταση μεταξύ του σημείου τερματισμού και του κόμβου **n** στον οποίο βρίσκεται τη συγκεκριμένη χρονική στιγμή

Σε κάθε κόμβο που εξετάζουμε επαναλαμβάνουμε τα παραπάνω βήματα για την δημιουργία των επόμενων πιθανών προς εξέταση κόμβων, μέχρι η Ευκλίδεια απόσταση του σημείου που θέλουμε να φτάσουμε σε σχέση με το σημείο που βρισκόμαστε να είναι μικρότερη του κόστους μετάβασης. Συγκεκριμένα πρέπει $\mathbf{h < 0,75 \bullet c(n,n')}$. Όταν συμβεί αυτό τότε ο αλγόριθμος θα τερματίσει την αναζήτηση, θεωρώντας ότι έφτασε στο στόχο. Βέβαια για να βρούμε το τελικό μονοπάτι που είναι και το βέλτιστο ως προς την συνάρτηση κόστους πρέπει να διατρέξουμε ανάποδα το δέντρο αναζήτησης από το στόχο έως τη ρίζα και να το αποθηκεύσουμε.

4.1.1 Υλοποίηση A*

Όπως αναφέραμε στην προηγούμενη ενότητα, ο αλγόριθμος A* θα προσπαθήσει να βρει το βέλτιστο μονοπάτι εξασφαλίζοντας μια διαδρομή χωρίς εμπόδια. Σε κάθε κόμβο ο αλγόριθμος μπορεί να δημιουργήσει τρεις επομένους κόμβους. Έναν κόμβο στην ευθεία του προηγούμενου, ένα κόμβο δεξιά του προηγούμενου κατά -45° και έναν κόμβο αριστερά του προηγούμενου κατά 45°.

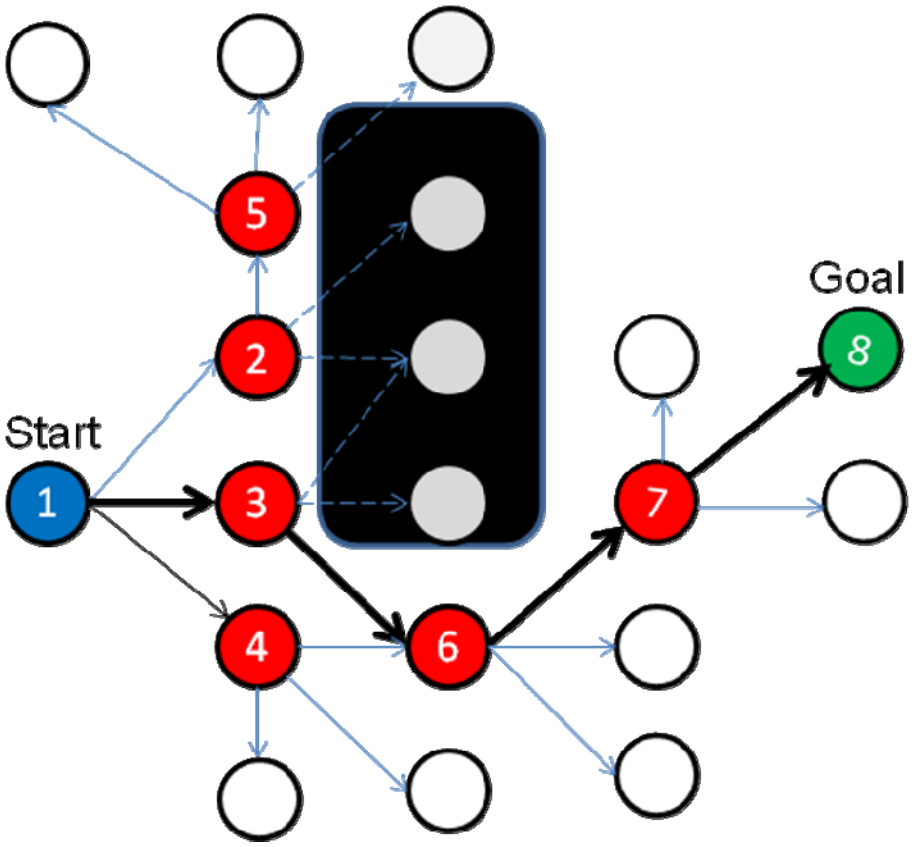
Για την αποθήκευση των δεδομένων που εξετάζουμε ανά πάσα στιγμή έχουμε φροντίσει να υλοποιήσουμε κλάσεις έτσι ώστε να μπορούμε να διαχωρίσουμε τα δεδομένα και να μπορούμε να έχουμε πρόσβαση σε αυτά, με εύκολο και γρήγορο τρόπο.

Έχουμε υλοποιήσει ένα δέντρο αναζήτησης που από κάθε κόμβο μπορεί να προχωρήσει με τρεις τρόπους. Έτσι έχουμε τρεις επιλογές για κίνηση σε επόμενο κόμβο: ευθεία, δεξιά -45° και αριστερά 45°. Η αναζήτηση προχωράει σύμφωνα με το κόστος για να ικανοποιήσουμε της απαιτήσεις του A*. Αυτό σημαίνει ότι ο επόμενος κόμβος που θα εξετάσουμε πρέπει να έχει το ελάχιστο κόστος από όλους τους κόμβους που έχουμε δημιουργήσει. Χρειαζόμαστε έναν γρήγορο και εύκολο τρόπο για να διατρέχουμε τους υπαρκτούς κόμβους και να βρίσκουμε αυτόν με το ελάχιστο κόστος. Έτσι υλοποιήσαμε μια στοίβα και συγκεκριμένα μια ελάχιστη στοίβα (Binary MinHeap) με την μορφή πίνακα. Στην ρίζα ή στο πρώτο στοιχείο της στοίβας είναι αποθηκευμένος ο κόμβος με το μικρότερο κόστος (MinHeap). Όλα τα στοιχεία κάτω από την ρίζα έχουν κόστος μεγαλύτερο από αυτό της ρίζας. Αυτό συμβαίνει για οποιοδήποτε στοιχείο του MinHeap, δηλαδή όλα τα στοιχεία που βρίσκονται σε χαμηλότερο επίπεδο στο ίδιο υποδένδρο από ένα συγκεκριμένο στοιχείο έχουν κόστη μεγαλύτερα από αυτό.

Η εισαγωγή στο MinHeap γίνεται με τέτοιο τρόπο ώστε να τηρούνται οι παραπάνω κανόνες, το ίδιο και η διαγραφή. Πιο συγκεκριμένα για να εισάγουμε ένα στοιχείο το τοποθετούμε αρχικά σαν το τελευταίο φύλλο του δέντρου. Στην συνέχεια κάνουμε εναλλαγές με κάθε στοιχείο που βρίσκεται πιο πάνω από το εισερχόμενο στοιχείο, εφόσον το πιο πάνω στοιχείο έχει μεγαλύτερο κόστος, αλλιώς το στοιχείο που εισάγαμε το αφήνουμε εκεί που είναι. Έτσι εξασφαλίζεται ότι κάθε κόμβος του MinHeap έχει μικρότερο κόστος από τα παιδιά του. Όμοια στην διαγραφή πάντα χρειαζόμαστε τον κόμβο με το μικρότερο κόστος για να τον εξετάσουμε στο επόμενο βήμα του A*. Έτσι όταν τραβάμε τον κόμβο αυτό από τη ρίζα του MinHeap, αυτή

μένει κενή, συνεπώς παίρνουμε το τελευταίο στοιχείο του MinHeap και το τοποθετούμε στην ρίζα. Κατόπιν κάνουμε τις απαραίτητες εναλλαγές με τα στοιχεία παιδιά που έχουν το μικρότερο κόστος από τους κόμβους που βρίσκονται σε παρακάτω επίπεδο, έτσι ώστε να μπει το στοιχείο στην θέση του και να διατηρηθούν οι κανόνες που επιβάλλονται.

Αφού λοιπόν ο αλγόριθμος καταλήξει στον επιθυμητό κόμβο τερματισμού χρειάζεται ένας τρόπος για να βρούμε το βέλτιστο μονοπάτι που ακολούθησε για να φτάσει εκεί. Για τον λόγο αυτό έχουμε φροντίσει να δημιουργήσουμε έναν σύνδεσμο του κάθε κόμβου με τον πατέρα του στο δέντρο αναζήτησης. Διατρέχοντας λοιπόν ανάποδα το μονοπάτι θα βρούμε το βέλτιστο μονοπάτι που μας οδήγησε στον επιθυμητό κόμβο. Στο σχήμα 4.1.4 φαίνεται ένα διάγραμμα δημιουργίας κάθε κόμβου μέχρι και την επίτευξη του τελικού μονοπατιού παρουσία εμποδίου.



Σχήμα 4.1.4 Παράδειγμα αναζήτησης του αλγορίθμου A* παρουσία εμποδίου

Για την επιτάχυνση της απόδοσης του αλγορίθμου μας ιδιαίτερα για δύσκολες περιπτώσεις χρειαζόμαστε άλλη μια κλάση όπου θα κρατάμε τους κόμβους που έχουμε εξετάσει σε προηγούμενα βήματα. Έτσι κάθε κόμβος που ελέγχεται, αποθηκεύεται σε ένα άλλο πίνακα για να μπορούμε να εξετάζουμε κάθε φορά αν ο προς εξέταση κόμβος έχει ήδη εξεταστεί. Αν ναι, τότε δεν ξαναδημιουργούμε τα παιδιά του αλλά εξετάζουμε τον αμέσως επόμενο σε κόστος κόμβο. Μ' αυτό τον τρόπο αποτρέπουμε την επαναδημιουργία κόμβων που έχουν ίδιες συντεταγμένες και διεύθυνση και το κόστος τους είναι μεγαλύτερο ή και ίσο με αυτού που έχουμε εξετάσει προηγουμένως.

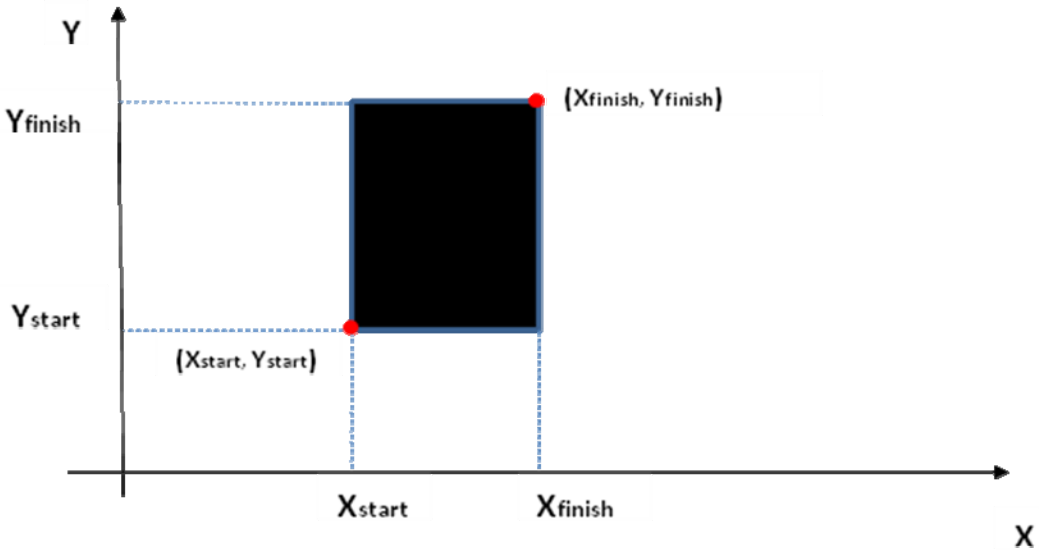
Ο αλγόριθμος λοιπόν θα ψάξει να βρει την βέλτιστη διαδρομή για να φτάσει στον στόχο, Αυτό σημαίνει ότι μπορεί το ρομπότ να φτάσει στον στόχο με

οποιαδήποτε διεύθυνση. Για τον λόγο αυτό έχει υλοποιηθεί επιπλέον λειτουργικότητα που εξασφαλίζει στο σημείο τερματισμού το σύστημα να τερματίζει με συγκεκριμένη διεύθυνση. Τέλος, έχουν αναπτυχθεί μέθοδοι που εξασφαλίζουν τον ασφαλή τερματισμό του αλγορίθμου σε ειδικές περιπτώσεις όπου ο αλγόριθμος δεν καταφέρνει να τερματίσει φυσιολογικά. Τέτοιες περιπτώσεις, για παράδειγμα, είναι αν το ρομπότ είναι περικυκλωμένο από εμπόδια και ο στόχος είναι έξω από αυτό ή ο στόχος είναι περικυκλωμένος από εμπόδια κατά συνέπεια ο αλγόριθμος δεν θα καταφέρει πότε να φτάσει στο στόχο.

4.1.2 Αναπαράσταση εμποδίων

Χωρίς την παρουσία ανεπιθύμητων- επικίνδυνων για το ρομπότ σημείων τα πράγματα είναι σχετικά απλά. Όταν όμως στην σχεδίαση πρέπει να συμπεριλάβουμε και την αναπαράσταση των εμποδίων που μπορεί να βρεθούν στη διαδρομή του ρομπότ, τότε τα πράγματα δυσκολεύουν τόσο ως προς την ταχύτητα του αλγόριθμου όσο και προς την αναπαράσταση των εμποδίων. Ο τρόπος που μπορούν τα εμπόδια να αναπαρασταθούν διαφέρει ανά περίπτωση και εφαρμογή. Αυτό που χρειάζεται είναι ένας τρόπος να διαπιστώνουμε, αν οι συντεταγμένες κάθε κόμβου που εξετάζουμε περιέχουν εμπόδια. Έτσι δημιουργούμε μια κλάση όπου αποθηκεύουμε τις συντεταγμένες εμποδίων σε έναν πίνακα εμποδίων. Τον πίνακα αυτόν τον εξετάζουμε σε κάθε βήμα αναζήτησης του A* και αν οι πιθανοί επόμενοι κόμβοι είναι εμπόδια σταματάμε την αναζήτηση των κόμβων αυτών δηλαδή δεν δημιουργούμε απογόνους κόμβους για τις συντεταγμένες εμποδίων.

Συγκεκριμένα για να μπορέσουμε να αναπαραστήσουμε τα δεδομένα εμποδίων θεωρούμε ότι θα έχουν την μορφή τετράπλευρου (σχήμα 4.1.5).

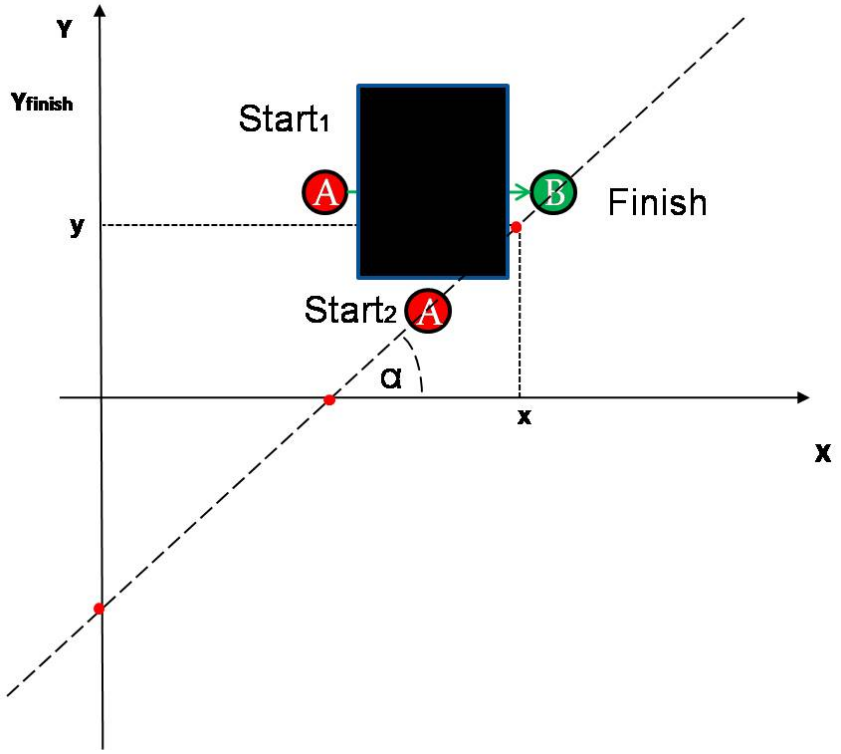


Σχήμα 4.1.5 Αναπαράσταση εμποδίων

Έτσι δημιουργούμε μιας κλάση μέσω της οποίας αποθηκεύονται οι αρχικές (x_{start},y_{start}) και τελικές (x_{finish},y_{finish}) συντεταγμένες και επιπλέον πρέπει να ικανοποιούν την παρακάτω ανισότητα:

$$O(x, y) = (x_{start}, y_{start}) \leq (x, y) \leq (x_{finish}, y_{finish})$$

Οποιοδήποτε ζευγάρι συντεταγμένων (x,y) ικανοποιεί την παραπάνω σχέση θεωρείται εμπόδιο. Πρέπει να αναφέρουμε επίσης ότι στην περίπτωση που οι διαστάσεις του εμποδίου είναι πολύ μικρότερες από τις διαστάσεις του βήματος του A*, τότε το λογισμικό δεν θα καταφέρει να αναγνωρίσει το εμπόδιο. Για τον λόγο αυτό δεν εξετάζουμε μόνο αν ο επόμενος κόμβος βρίσκεται σε σημείο όπου υπάρχει εμπόδιο, αλλά αν στην ευθεία που ενώνει τον κόμβο που βρισκόμαστε και τον προς εξέταση κόμβο βρίσκεται κάποιο εμπόδιο. Δηλαδή, εξετάζουμε αναλυτικά κάθε σημείο της ευθείας που ενώνει τον κόμβο που βρισκόμαστε και τον εξεταζόμενο κόμβο. Στο σχήμα 4.1.6 βλέπουμε δυο τέτοιες περιπτώσεις όπου ο προς αναζήτηση κόμβος B δεν βρίσκεται πάνω σε συντεταγμένες εμποδίων, όμως στην ευθεία που ενώνει τους κόμβους A, B υπάρχουν σημεία που είναι, έτσι με τη βοήθεια της εξίσωσης ευθείας μπορούμε να διαπιστώσουμε αν υπάρχουν τέτοια σημεία.



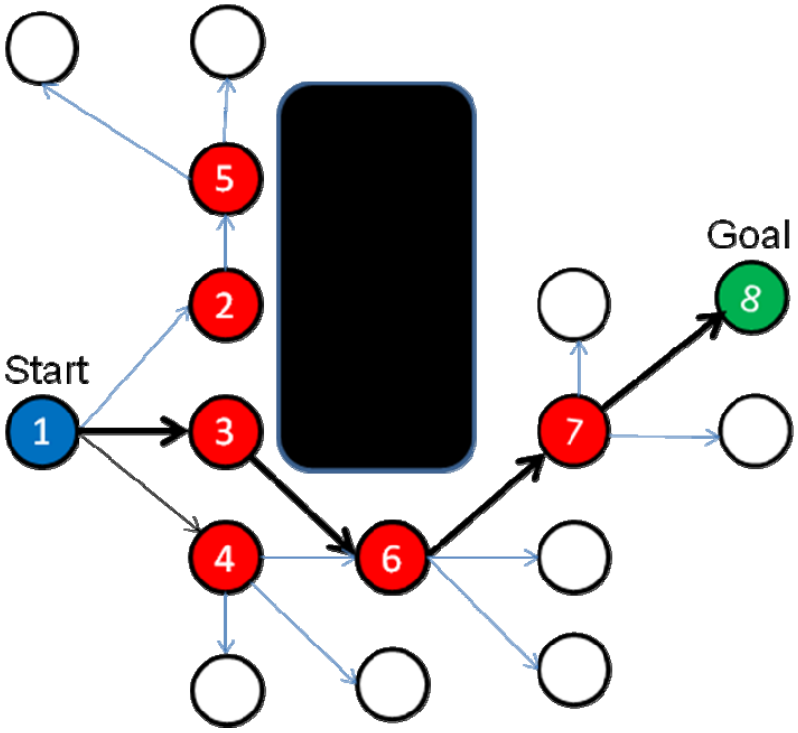
Σχήμα 4.1.6 Μη επιτρεπτές (επικίνδυνες) μεταβάσεις μέσω εμποδίων

$$y = \alpha x + \beta$$

$$\alpha = \frac{y_{finish} - y_{start}}{x_{finish} - x_{start}}$$

$$\beta = y_{start} - \alpha x_{start}$$

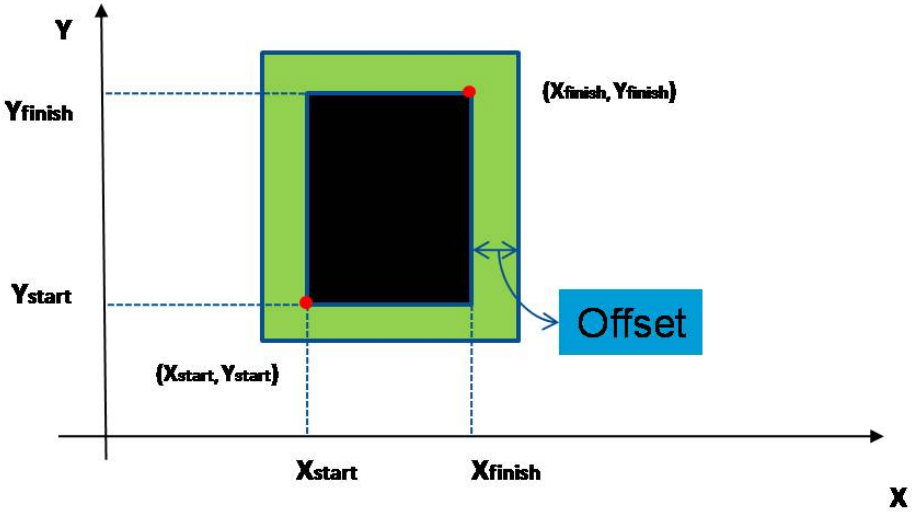
Σύμφωνα με τις παραπάνω εξισώσεις, υπολογίζουμε σε κάθε περίπτωση την εξίσωση ευθείας από τις αρχικές ($\mathbf{x_{start}}$,$\mathbf{y_{start}}$) και τελικές ($\mathbf{x_{finish}}$,$\mathbf{y_{finish}}$) συντεταγμένες, επιπλέον δίνοντας τιμές από $\mathbf{x_{start}}$ έως $\mathbf{x_{finish}}$ στο **x**. Θα λάβουμε τιμές για το **y** (από την εξίσωση ευθείας). Όλα τα ζευγάρια **x, y** τα εξετάζουμε αν είναι συντεταγμένες εμποδίων, σύμφωνα με την σχέση **O(x, y).**



Σχήμα 4.1.7 Παράδειγμα αναζήτησης βελτιωμένου αλγορίθμου A* παρουσία εμποδίου

Είναι αναγκαίο να αναφερθεί ότι κόμβοι με συντεταγμένες εμποδίων δεν επιτρέπεται να δημιουργούνται. Το σχήμα 4.1.7 περιγραφεί μια απλή περίπτωση λειτουργικότητας όπως το υλοποιήσαμε. Παρατηρούμε λοιπόν ότι οι κόμβοι που θα έπεφταν πάνω στο εμπόδιο δεν δημιουργούνται καθόλου από τον αλγόριθμο. Αυτό το κάνουμε για να βελτιώσουμε την ταχύτητα αλλά και τη χρήση των πόρων του υπολογιστικού συστήματος.

Τέλος, για να μπορέσουμε να καθορίσουμε την απόσταση ασφαλείας που θα θέλαμε το ρομπότ να διατηρήσει στην κίνηση του από το εμπόδιο ορίζουμε μια παράμετρο offset (σχήμα 4.1.8).



Σχήμα 4.1.8 Αναπαράσταση εμποδίων και επίπεδο ασφαλείας Offset

Η παράμετρος αυτή εισάγεται ως επιπλέον παράμετρος στον τρόπο που ο αλγόριθμος A* θα αποφασίζει για το αν οι συγκεκριμένες συντεταγμένες που αναζητούμε είναι συντεταγμένες εμποδίων η όχι. Αυτό επιτυγχάνεται αν επιπλέον για τα εμπόδια ορίσουμε τις παρακάτω σχέσεις:

$$x_{start} - offset \leq x \leq x_{finish} + offset$$

$$y_{start} - offset \leq y \leq y_{finish} + offset$$

Είναι φανερό λοιπόν ότι όσο μικρότερη είναι η τιμή του offset τόσο πιο κοντά επιτρέπουμε να περνάει το ρομποτικό μας σύστημα από τα εμπόδια. Έτσι μπορούμε με έναν επιπλέον τρόπο να καθορίσουμε το πόσο ασφαλής είναι ή όχι η διέλευση από συγκεκριμένη απόσταση από το εμπόδιο.

4.1.3 Ο αλγόριθμος σχεδιασμού διαδρομής

Αναλυτικά ο αλγόριθμος σε ψευδοκώδικα είναι ο εξής:

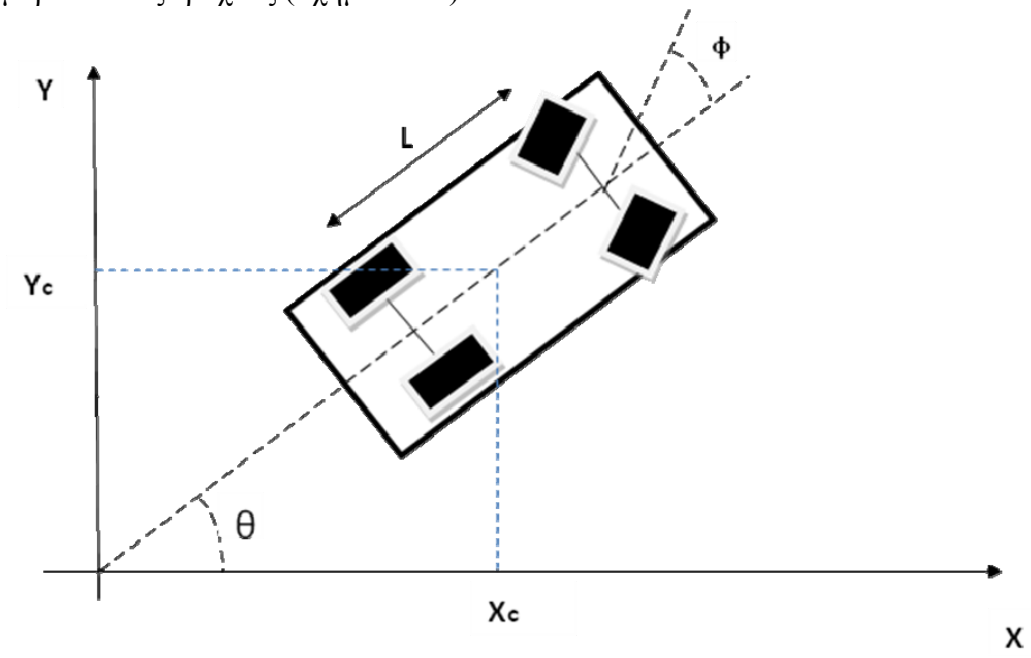
```
Do
        If node!=checked then
            For F=-45 to 45 Step:45
                Compute Xnext,Ynext,Cost, Orientation
                Compute g,h, Cost

                If Xnext,Ynext Aren't Obstacle then
                    Insert to Graph
                    Insert to Heap
                End if

            Next F

            Insert to checked
        End if

        Close node
        Node=MinHeap
        Remove MinHeap

        If Node is Destination Then
            Trace Path
            Return Path
        End if
Loop
```

Σχήμα 4.1.9 αλγόριθμος σχεδιασμού διαδρομής σε ψευδοκώδικα

4.2 Έλεγχος κίνησης

Όπως αναφέραμε η κίνηση των ρομποτικών συστημάτων περιλαμβάνει το σχεδιασμό διαδρομής σε υψηλότερο επίπεδο και τον ελεγκτή κίνησης σε χαμηλότερο.

Οι κινηματικές εξισώσεις κίνησης του ρομπότ είναι αυτές που καθορίζουν την επομένη θέση-διεύθυνση του ρομπότ στην επομένη χρονική στιγμή ανάλογα με τις επιλογές ελέγχου. Στην περίπτωσή μας το ρομποτικό σύστημα που θα μελετήσουμε είναι όχημα με τέσσερις τροχούς με δυνατότητα αλλαγής πορείας μόνο από τους μπροστινούς τροχούς (σχήμα 4.2.1).



Σχήμα 4.2.1 θεωρητικό ρομποτικό μοντέλο

Όπως αναφέραμε το μοντέλο του αυτοκινήτου είναι στην κατηγορία των μη ολονομικών ρομπότ. Άρα οι βαθμοί ελευθερίας που μπορούμε να ελέγξουμε είναι δύο: η διεύθυνση των εμπρόσθιων τροχών και η ταχύτητα του οχήματος. Όποτε, αν θεωρήσουμε ότι το ρομποτικό όχημα μας κινείται πάνω σε ένα καρτεσιανό σύστημα συντεταγμένων με σταθερή ταχύτητα υ και σταθερή γωνία στροφής φ τότε τα χαρακτηριστικά που καθορίζουν πλήρως τη θέση του είναι (X,Y,Θ) και η κίνηση του περιγράφεται από τις παρακάτω κινηματικές εξισώσεις:

$$\frac{dx}{dt} = \upsilon\cos\theta$$

$$\frac{dy}{dt} = \upsilon\sin\theta$$

$$\frac{d\theta}{dt} = \upsilon\frac{\tan\varphi}{L}$$

Όπου L το μήκος του οχήματος.

Αν ολοκληρώσουμε ως προς t τότε θα έχουμε:

$$x(t) = \upsilon t\cos\theta$$

$$y(t) = \upsilon t\sin\theta$$

$$\theta(t) = \upsilon \frac{\tan\varphi}{L} t$$

4.2.1 Πολυκριτήριος ελεγκτής κίνησης

Για να αντιμετωπίσουμε το πρόβλημα του έλεγχου κίνησης του ρομπότ πιο δραστικά χρησιμοποιούμε μια τεχνική που βασίζεται στη διακριτή βελτιστοποίηση μιας πολυκριτήριας αντικειμενικής συνάρτησης (multi-objective function), η οποία καταφέρνει να εξισορροπεί ομαλότητα και ακρίβεια.

Ομαλότητα - δυναμικό παράθυρο

Η μέθοδος του δυναμικού παράθυρου είναι γνωστή από τις αρχές τις δεκαετίας του 90 [38][39]. Αυτό που η μέθοδος απαιτεί είναι να διακριτοποιήσουμε το χώρο των παραμέτρων ελέγχου του ρομπότ. Επιπλέον, και με βάση το δυναμικό μοντέλο του ρομπότ να ορίσουμε τους περιορισμούς στον έλεγχο εισόδου [35]. Τέτοιοι περιορισμοί περιλαμβάνουν τα άνω όρια των ταχυτήτων, αντιστοιχών επιταχύνσεων-επιβραδύνσεων και της γωνίας στροφής του τιμονιού. Θέτουμε ως $\boldsymbol{\upsilon_{max}}$ τη μέγιστη ταχύτητα που μπορεί να κινηθεί το ρομπότ και ως $\dot{\upsilon}_{max}$ τη μεγίστη επιτάχυνση-επιβράδυνση. Επίσης, αν θεωρήσουμε ότι το ρομπότ κινείται με ταχύτητα $\boldsymbol{\upsilon_c}$ και γωνιά στρέψης $\boldsymbol{\varphi_c}$, τότε την επομένη χρονική στιγμή **dt** το δυναμικό παράθυρο στο χώρο των ταχυτήτων και των γωνιών (σχήμα 4.2.2) ορίζεται συμφώνα με τις παρακάτω ανισότητες:

$$0 \leq \upsilon \leq \upsilon_{max}$$

$$\varphi_{min} \leq \varphi \leq \varphi_{max}$$

$$\upsilon_c - \dot{\upsilon}_{max} \bullet dt \leq \upsilon \leq \upsilon_c + \dot{\upsilon}_{max} \bullet dt$$

$$\varphi_c - \varphi_{step} \leq \varphi \leq \varphi_c + \varphi_{step}$$

Όπου:

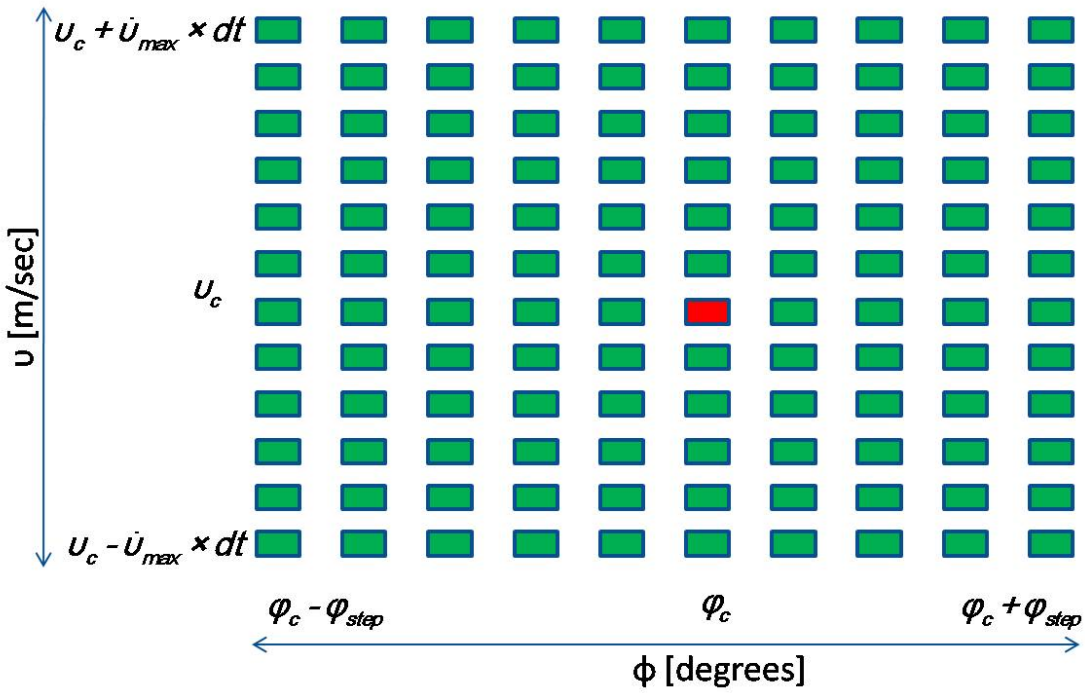
$\boldsymbol{\upsilon_{max}}$ **,** $\boldsymbol{\varphi_{max}}$:Μέγιστη δυνατή ταχύτητα και γωνία στροφής που μπορεί να δεχθεί το ρομποτικό σύστημα.

dt: Χρονικό διάστημα ελέγχου.

$\boldsymbol{\upsilon_c}$, $\boldsymbol{\varphi_c}$: Τρέχουσες τιμές ελέγχου.

$\boldsymbol{\varphi_{step}}$: Μέγιστη μεταβολή γωνίας στροφής σε ένα κύκλο ελέγχου dt.

$\dot{\upsilon}_{\mathbf{max}} \bullet \mathbf{dt}$: Μέγιστη επιτρεπόμενη μεταβολή της ταχύτητας με βάση την μέγιστη επιβράδυνση/επιταχυνση που μπορεί να δεχθεί το ρομποτικό όχημα, σε έναν κύκλο ελέγχου dt.



Σχήμα 4.2.2 Χώρος αναζήτησης δυναμικού παραθύρου

Έτσι, με βάση τα δυναμικά χαρακτηριστικά του ρομπότ αποφασίζεται το εύρος για τις τιμές των ζευγαριών (υ, φ) που επιτρέπονται σε διάρκεια χρόνου dt, για δεδομένες τιμές γραμμικής ταχύτητας και γωνίας στροφής (σχήμα 4.2.2). Το εύρος αυτό αποτελεί το μέγεθος του δυναμικού παραθύρου. Η διακριτοποίηση στις τιμές των ζευγαριών (υ, φ) αποφασίζεται ανάλογα με την ακρίβεια που απαιτείται σε κάθε περίπτωση. Έτσι δίνοντας διακριτές τιμές σε κάθε συνδυασμό των ζευγαριών (υ, φ) εντός του δυναμικού παραθύρου, παράγουμε ένα δισδιάστατο διακριτό πλέγμα. Ένα από τα παραπάνω ζευγάρια τιμών του πλέγματος αυτού είναι εκείνο που θα χρησιμοποιηθεί εν τέλει από το ρομποτικό σύστημα στο επόμενο βήμα ελέγχου.

Μ' αυτό τον τρόπο επιτυγχάνουμε ομαλότητα στον τρόπο ελέγχου των παραμέτρων του ρομποτικού μας συστήματος, ακολουθώντας μια τροχιά χωρίς να παραβιάζουμε τους κανόνες της ομαλής λειτουργίας του συστήματός μας.

Ακρίβεια - Υπολογισμός τροχιάς

Έστω ότι οι συντεταγμένες του στόχου που θέλουμε να προσεγγίσει το ρομπότ βρίσκονται στην θέση ($\mathbf{x_d}$,$\mathbf{y_d}$). Το ερώτημα που γεννιέται είναι τι ταχύτητα θα έχει το ρομποτικό μας σύστημα όταν θα φτάσει στις συγκεκριμένες συντεταγμένες;

Όπως είναι προφανές δεν πρέπει να επιτρέψουμε στο ρομπότ να φτάσει με μη μηδενική ταχύτητα στο επιθυμητό σημείο, γιατί αυτό θα σήμαινε ότι δεν θα μπορούσε να σταματήσει στις επιθυμητές συντεταγμένες. Οπότε πρέπει να εφαρμόσουμε μια μεθοδολογία που να μας επιτρέπει να υπολογίζουμε με ακρίβεια ποια πρέπει να είναι ανά πάσα στιγμή η ταχύτητα για να καταφέρουμε να φτάσουμε στο στόχο με μηδενική ταχύτητα [35].

Για κάθε ζεύγος τιμών ταχυτήτων $\boldsymbol{\upsilon_c}$ και γωνιάς $\boldsymbol{\varphi_c}$ σε μια δεδομένη χρονική στιγμή $\mathbf{t_c}$ που έχουν επιλεγεί από το δυναμικό παράθυρο (Dynamic Window), θεωρούμε ως ελάχιστη απόσταση φρεναρίσματος MBD (Minimum Braking Distance) την απαιτούμενη ελάχιστη απόσταση που χρειάζεται το ρομπότ να ταξιδέψει από την

χρονική στιγμή $\mathbf{t_c}$ μέχρι να σταματήσει εντελώς. Επίσης θεωρούμε ως ελάχιστο χρόνο φρεναρίσματος **MBT** (Minimum Braking Time) τον χρόνο που χρειάζεται για να σταματήσει το ρομπότ με μεγίστη επιβράδυνση $\dot{\upsilon}_{max}$ από την ταχύτητα $\boldsymbol{\upsilon_c}$ διανύοντας μια απόσταση ίση με την **MBD**.

$$MBT = \frac{\upsilon_c}{\dot{\upsilon}_{max}}$$

$$\upsilon_{av} = \frac{1}{2}\upsilon_c$$

Όπου $\boldsymbol{\upsilon_{av}}$ η μέση ταχύτητα στο διάστημα MBT.

Σε κάθε βήμα ελέγχου διάρκειας dt, το δυναμικό παράθυρο δίνει τιμές στα ζευγάρια παραμέτρων (υ, φ) άρα οι πιθανές επόμενες συντεταγμένες x_{k+1}, y_{k+1}, θ_{k+1} θα δίνονται από τις παρακάτω εξισώσεις:

$$\theta_{k+1}(\upsilon, \varphi, dt) = \theta_k + \upsilon \frac{\tan\varphi}{L} dt$$

$$x_{k+1}(\upsilon, \varphi, dt) = x_k + \upsilon\cos\theta_{k+1} dt$$

$$y_{k+1}(\upsilon, \varphi, dt) = y_k + \upsilon\sin\theta_{k+1} dt$$

Η ελάχιστη απόσταση φρεναρίσματος δίνεται από τις παρακάτω εξισώσεις:

$$MBD_x = \upsilon_{av} MBT\cos(\theta_{k+1} + \upsilon_{av} \frac{\tan\varphi}{L} MBT)$$

$$MBD_y = \upsilon_{av} MBT\sin(\theta_{k+1} + \upsilon_{av} \frac{\tan\varphi}{L} MBT)$$

Όπου $\mathbf{MBD_x}$, $\mathbf{MBD_y}$ η εκτίμηση της απόστασης που προβλέπουμε ότι θα διανύσει το ρομποτικό όχημα στους άξονες x, y αντίστοιχα από την προς εξέταση θέση $(\mathbf{x_{k+1}}, \mathbf{y_{k+1}}, \boldsymbol{\theta_{k+1}})$ (με ταχύτητα **υ** και γωνία **φ**) έως ότου σταματήσει. Για τη πρόβλεψη αυτή υποθέσαμε ότι το όχημα θα επιβραδύνει με μέγιστη επιβράδυνση από την ταχύτητα **υ** μέχρι να μηδενιστεί. Έτσι για τη συνολική κίνηση υποθέτουμε ότι το όχημα θα κινείται για χρόνο MBT και με ταχύτητα $\boldsymbol{\upsilon_{av}}$.
Μπορούμε να εκτιμήσουμε τη τελική θέση και διεύθυνση όπου το ρομπότ θα τερματίσει με μηδενική ταχύτητα, με βάση την εκτίμηση αυτή το όχημα θα σταματήσει στις συντεταγμένες $(\mathbf{F_x}, \mathbf{F_y})$ και με διεύθυνση $\mathbf{F_\theta}$, οι οποίες υπολογίζονται ως εξής:

$$F_x = x_{k+1} + MBD_x$$

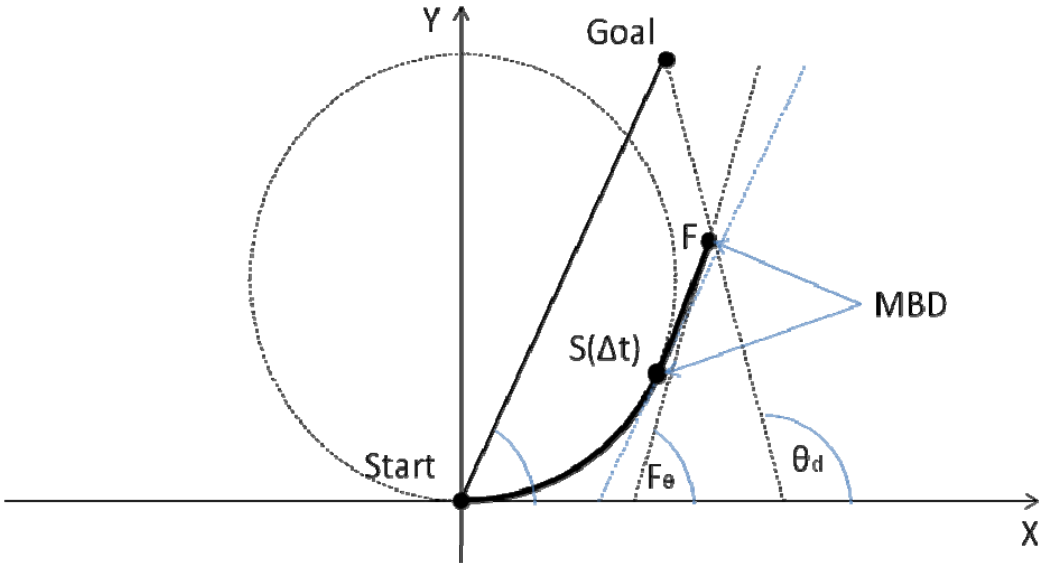
$$F_y = y_{k+1} + MBD_y$$

$$F_\theta = \theta_{k+1} + \upsilon_{av} \frac{\tan\varphi}{L} MBT$$

Ο αλγόριθμος υποθέτει ότι για κάθε ζεύγος τιμών **(υ, φ)** το ρομποτικό όχημα θα φτάσει στην θέση $(\mathbf{F_x}, \mathbf{F_y}, \mathbf{F_\theta})$ με, μηδενική ταχύτητα.

Αντικειμενική Συνάρτηση Κόστους

Έχοντας υπολογίσει όλα τα παραπάνω σε κάθε βήμα εκτέλεσης το μόνο που μένει είναι για κάθε ζευγάρι τιμών να υπολογίζουμε και την αντικειμενική συνάρτηση κόστους **F**. Έτσι το ζευγάρι τιμών που ελαχιστοποιεί τη συνάρτηση αυτή θα είναι και το επιθυμητό επόμενο ζευγάρι. Κατά συνέπεια και σύμφωνα με τις παρακάτω επιμέρους συναρτήσεις η $\mathbf{F_1}$ είναι η συνάρτηση αυτή που μας δείχνει πόσο κοντά έχουμε φτάσει στον στόχο. Η $\mathbf{F_2}$ μας δείχνει πόσο κοντά είναι η διεύθυνση $\mathbf{F_\theta}$ του ρομπότ μας σε σχέση με την διεύθυνση $\boldsymbol{\theta_d}$, όπου $\boldsymbol{\theta_d}$ είναι η γωνιά που σχηματίζεται από την προβολή της ευθείας του στόχου και του σημείου που εκτιμούμε ότι θα σταματήσει το ρομπότ (σχήμα 4.2.3).



Σχήμα 4.2.3 Εκτίμηση τροχιάς ρομποτικού συστήματος και τελική θέση τερματισμού

Αναλυτικά έχουμε:

$$F_1 = (F_x - x_{destination})^2 + (F_y - y_{destination})^2$$

$$F_2 = \left(F_\theta - \theta_d\right)^2$$

$$\theta_d = \mathrm{Arctan2}\left(y_{destination} - F_y, x_{destination} - F_x\right)$$

$$F = W_1 \bullet F_1 + W_2 \bullet F_2 ,$$

Τα W1 και W2 είναι τα εμπειρικά βάρη σημαντικότητας κάθε επιμέρους συνάρτησης ξεχωριστά. Οι τιμές των W1 και W2 καθορίστηκαν κατόπιν πειραματισμού. Τα καλύτερα αποτελέσματα λαμβάνονται για W1=W2=1. Για την περίπτωση μας η συνάρτηση **F** είναι η αντικειμενική συνάρτηση που πρέπει να ελαχιστοποιηθεί.

Για να συνοψίσουμε, κάθε βήμα του ελεγκτή κίνησης περιλαμβάνει τις ακόλουθες ενέργειες:

1) Καθορίζουμε το δυναμικό παράθυρο.
2) Δημιουργούμε δισδιάστατο δυναμικό παράθυρο.
3) Δίνουμε τιμές στις συναρτήσεις κόστους για κάθε ζεύγος τιμών του πλέγματος.
4) Διαλέγουμε το ζεύγος τιμών για τις οποίες η συνάρτηση κόστους **F** ελαχιστοποιείται.
5) Χρησιμοποιούμε αυτό το ζεύγος τιμών στο επόμενο βήμα.

4.2.2 Αλγόριθμος ελέγχου κίνησης

```
Do
    Φcurrent min=Φcurrent - Φstep
    Φcurrent max=Φcurrent + Φstep

    Ucurrent min=Ucurrent - Ustep
    Ucurrent max=Ucurrent + Ustep

    Φcurrent min = max(Φmin, Φcurrent min,)
    Φcurrent max= min(Φmax, Φcurrent max)

    Ucurrent min = max(0, Ucurrent min)
    Ucurrent max= min(Umax, Ucurrent max)

    For φ=Φcurrent min to Φcurrent max With Resolution Step
       For υ=Ucurrent min to U current max With Resolution Step

          Calculate Objective Function with (υ,φ)
          Pick pair (υ,φ) that minimizes objective Function

       End for
    End for

    Apply (υ,φ) best

    If Destinations is Reached exit
Loop
```

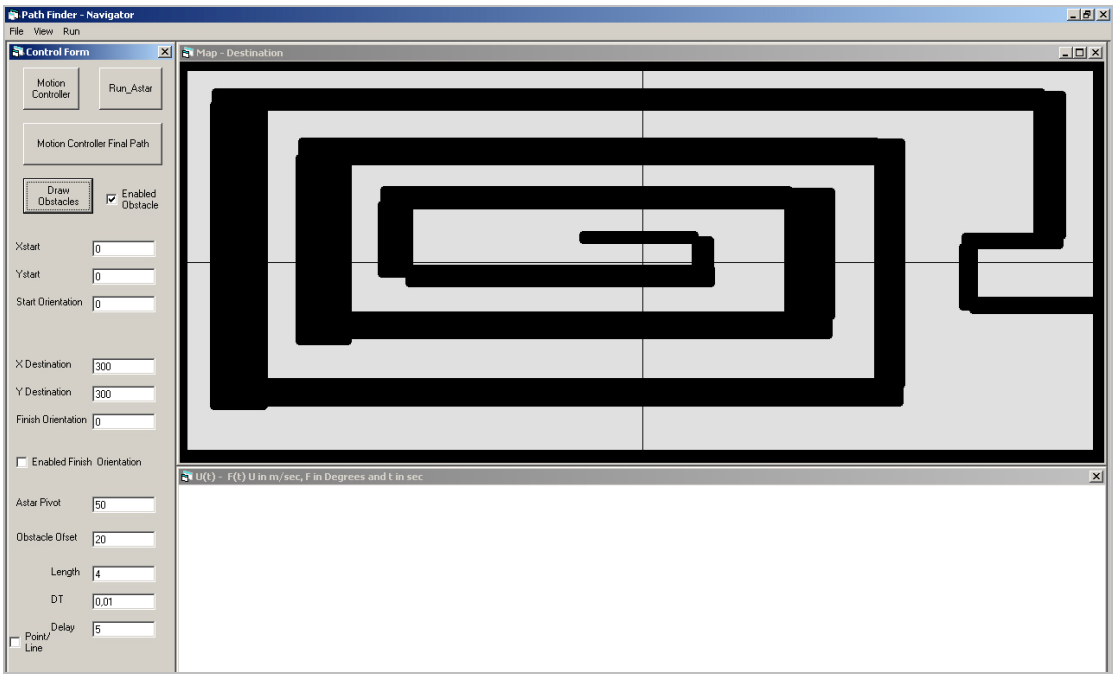
Σχήμα 4.2.4 αλγόριθμος ελέγχου κίνησης σε ψευδοκώδικα

Κεφάλαιο 5 Διεπαφή χρήστη

Στο κεφάλαιο αυτό θα περιγράψουμε τη λειτουργία, τη λειτουργικότητα και τις δυνατότητες του γραφικού περιβάλλοντος εργασίας. Η ανάπτυξη και η υλοποίηση όλου του προγραμματισμού πραγματοποιήθηκε σε Visual Studio 6 Enterprise Version στο λειτουργικό σύστημα Windows Vista.

5.1 Περιβάλλον εργασίας

Το περιβάλλον εργασίας έχει σχεδιαστεί έτσι ώστε να καλύψει όσο το δυνατόν περισσότερες ανάγκες στην προσομοίωση της πλοήγησης του συγκεκριμένου ρομποτικού συστήματος που μελετάμε. Επιπλέον, να διευκολύνει τον χρήστη ως προς το σχεδιασμό και το σενάριο στο οποίο θα ήθελε να υποβάλλει το ρομποτικό του σύστημα, δοκιμάζοντας όλες τις πιθανές εκδοχές πλοήγησης και τη συμπεριφορά του συστήματος σε κάθε περίπτωση.



Σχήμα 5.1.1Λογισμικό διασύνδεσης χρήστη για προσομοίωση του ρομποτικού συστήματος

Όπως βλέπουμε και στο σχήμα 5.1.1 το λογισμικό διασύνδεσης αποτελείται από τέσσερις βασικές φόρμες εργασίας.

Οι φόρμες που υλοποιήθηκαν είναι:

- **Φόρμα Ελέγχου (Control Form)**: Βρίσκεται στα αριστερά του γενικού περιβάλλοντος εργασίας. Μέσω αυτής της φόρμας δίνονται οι βασικές εντολές που θέλουμε να εκτελέσει το ρομποτικό μας σύστημα, ως προς το είδος της προσομοίωσης που θέλουμε να δοκιμάσουμε. Επίσης, έχουμε την δυνατότητα να εισάγουμε όλες τις απαραίτητες παραμέτρους για κάθε είδος προσομοίωσης που θέλουμε να μελετήσουμε.

- **Φόρμα Συντεταγμένων κίνησης - Εμποδίων (Map-Destination):** Η δεύτερη φόρμα εργασίας, είναι μια απεικόνιση του περιβάλλοντος του ρομποτικού μας συστήματος. Η φόρμα αυτή μας δίνει την δυνατότητα να μελετήσουμε την κίνηση του ρομποτικού μας συστήματος σε σχέση με τις συντεταγμένες που έχουν δοθεί. Επίσης, μπορούμε να δούμε και να σχεδιάσουμε τα εμπόδια στο χώρο και να δούμε την θέση του οχήματος σε σχέση με αυτά.

- **Φόρμα Μεταβλητών Ελέγχου (U(t), φ(t))**: Η συγκεκριμένη φόρμα ενεργοποιείται μόνο στην περίπτωση που έχουμε χρησιμοποιήσει τον ελεγκτή κίνησης. Σκοπό έχει να μας δείχνει αναλυτικά την ταχύτητα και τη γωνία στροφής που θα πρέπει να έχει το ρομποτικό μας σύστημα κάθε χρονική στιγμή για να ακολουθήσει το μονοπάτι που του έχει ζητηθεί.

- **Φόρμα Πλοήγησης (Navigation Form):** Η φόρμα αυτή περικλείει όλες τις προηγούμενες, ο βασικός της σκοπός είναι η υποστήριξη βασικών εργασιών που θέλουμε να εκτελέσουμε, όπως άνοιγμα, αποθήκευση, εκτέλεση, κλπ.

5.1.1 Φόρμα ελέγχου

Σ' αυτή τη φόρμα εργασίας (σχήμα 5.1.2) γίνεται ο βασικός έλεγχος και η παραμετροποίηση του συστήματος για να ξεκινήσουμε την προσομοίωση της πλοήγησης.

Αναλυτικά έχουμε:
Motion Controller: Πατώντας το κουμπί αυτό ο προσομοιωτής θα τρέξει τον αλγόριθμο του ελεγκτή κίνησης για τις επιθυμητές συντεταγμένες.

Path Planner: Πατώντας το κουμπί αυτό ο προσομοιωτής θα τρέξει τον αλγόριθμο σχεδιασμού διαδρομής για τις επιθυμητές συντεταγμένες. Αν για κάποιο λόγο η διαδρομή δεν είναι δυνατόν να δημιουργηθεί, ο αλγόριθμος θα τερματίσει χωρίς το σχεδιασμό κάποιας διαδρομής. Τέτοιες περιπτώσεις μπορεί να υπάρξουν αν το επιθυμητό σημείο που έχουμε επιλέξει δεν είναι δυνατό να προσεγγισθεί, ή το σημείο εκκίνησης περιβάλλεται από εμπόδια, κλπ.

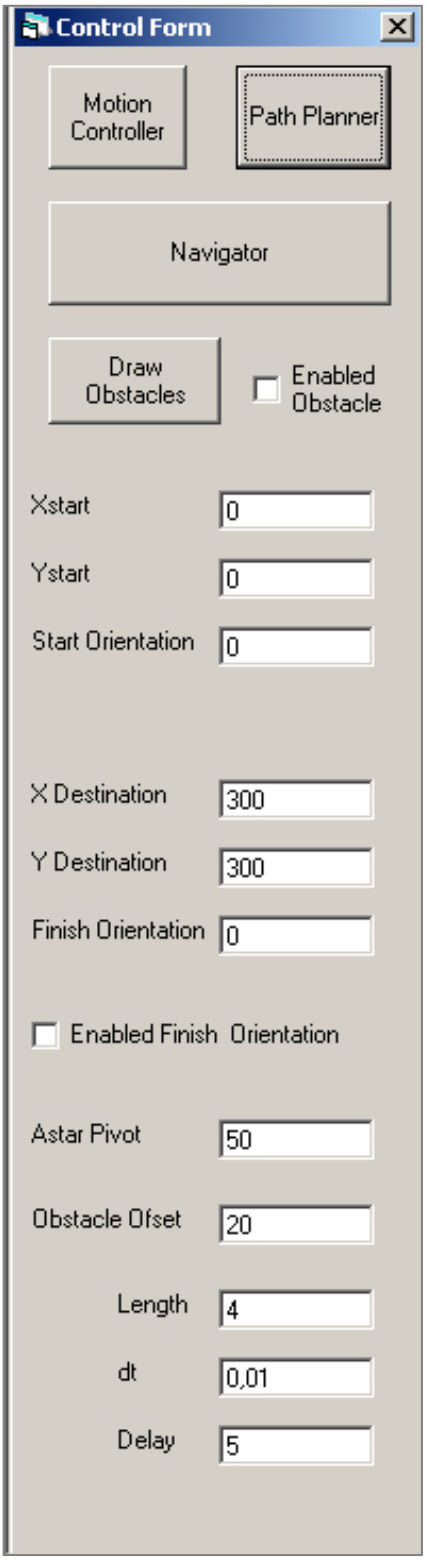
Navigator: Πατώντας το κουμπί αυτό ο προσομοιωτής θα τρέξει πρώτα τον αλγόριθμο σχεδίασης διαδρομής για τις επιθυμητές συντεταγμένες και έπειτα για κάθε βήμα του μονοπατιού που βρέθηκε, θα τρέξει τον ελεγκτή κίνησης μέχρι να φτάσει το όχημα στις επιθυμητές συντεταγμένες.

Draw Obstacles: Με την επιλογή αυτή ενεργοποιούμε το σχεδιασμό των εμποδίων, εάν βέβαια έχουμε ενεργοποιήσει την επιλογή **Enabled Obstacles.**

Στα επόμενα πεδία της φόρμας θέτουμε τις τιμές των παραμέτρων του ρομποτικού συστήματος. Οι τιμές αυτές θα χρησιμοποιηθούν στην επόμενη προσομοίωση.

Xstart, Ystart, StartOientation: Θέτουμε τις συντεταγμένες και τον προσανατολισμό της θέσης εκκίνησης του ρομποτικού συστήματος (X_{start}, Y_{start}, Θ_{start}).

Xdestination, Ydestination, Finish Orientation: Θέτουμε τις συντεταγμένες και τον προσανατολισμό της θέσης τερματισμού του ρομποτικού συστήματος (X_{end}, Y_{end}, Θ_{end}), η διεύθυνση τερματισμού λαμβάνετε υπόψη μόνο εφόσον έχει επιλεχθεί η παράμετρος **Enabled Finish Orientation**.



Σχήμα 5.1.2 Φόρμα ελέγχου

Enabled Finish Orientation: Με την επιλογή αυτού του πεδίου, ο αλγόριθμος θα προσπαθήσει να φτάσει στον στόχο με συγκεκριμένη διεύθυνση, δηλαδή υπολογίζοντας βέλτιστη διαδρομή για τις επιθυμητές συντεταγμένες, αλλά με κριτήριο την επιλεγμένη τελική διεύθυνση. Η διεύθυνση αυτή είναι αυτή που έχουμε εισάγει στο πεδίο **Finish Orientation.** Εάν δεν είναι δυνατό να φτάσει με την επιθυμητή διεύθυνση, ο αλγόριθμος θα τερματίσει υπολογίζοντας την συντομότερη διαδρομή, ανεξάρτητα της διεύθυνσης τερματισμού.

Astar Pivot: το πεδίο αυτό αφορά μόνο τον αλγόριθμο A* και μέσω αυτού καθορίζουμε το μέγεθος κάθε κελιού στη διακριτοποίηση για αναζήτηση.

Obstacle Offset: Η παράμετρος αυτή καθορίζει την απόσταση ασφαλείας που το ρομποτικό σύστημα κρατάει σε σχέση με τα εμπόδια.

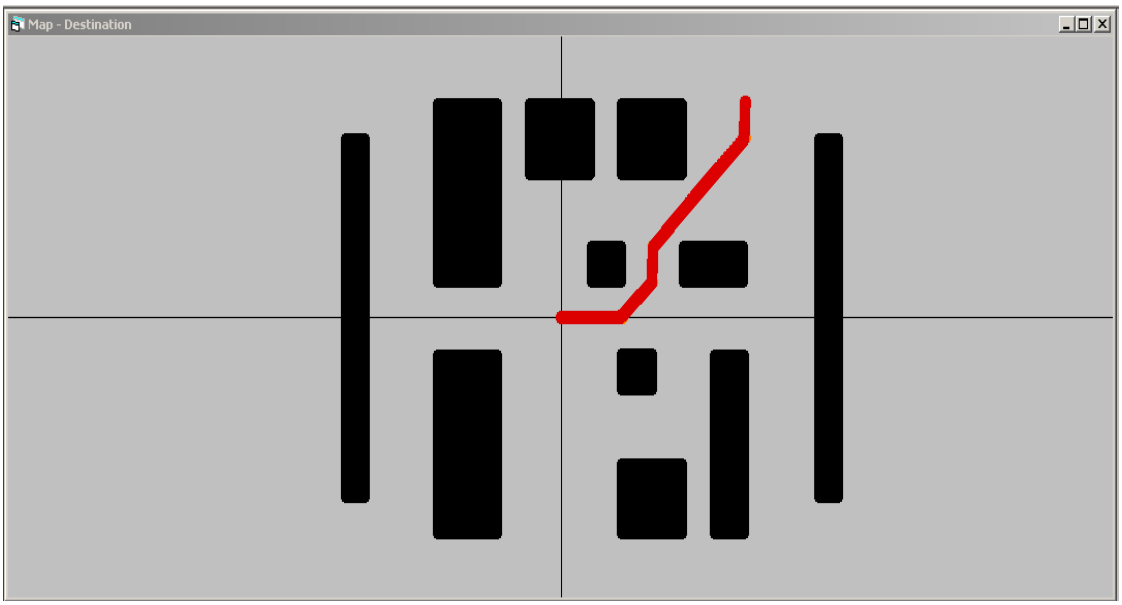
Length: Ορίζουμε το μήκος του ρομποτικού μας συστήματος. Η παράμετρος αυτή επηρεάζει τον ελεγκτή κίνησης**.**

dt: Ορίζουμε το χρονικό βήμα/διάστημα ελέγχου για τον ελεγκτή κίνησης.

Delay: Ο αλγόριθμος σχεδιασμού διαδρομής θα υπολογίσει διακριτούς κόμβους που θα οδηγήσουν στον στόχο. Ο ελεγκτής κίνησης, όπως έχουμε αναφέρει, θα λάβει αυτούς τους κόμβους και θα τους ακολουθήσει τον ένα μετά τον άλλο, στην περίπτωση που θέλουμε να ακολουθήσουμε το συνολικό μονοπάτι από την αρχή ως το τέλος. Ο ελεγκτής κίνησης θα πρέπει να μην επιβραδύνει στην προσέγγιση κάθε ενδιάμεσου κόμβου και να επιταχύνει ξανά για να φτάσει στον επόμενο. Έτσι με την παράμετρο Delay ορίζουμε πόσο πριν φτάσει στον κάθε κόμβο να θεωρήσει ότι τον έχει φτάσει και να προχωρήσει στον επόμενο.

Αυτό επαναλαμβάνεται μέχρι και τον τελευταίο κόμβο, στον οποίο τελικά θα σταματήσει.

5.1.2 Φόρμα συντεταγμένων κίνησης εμποδίων

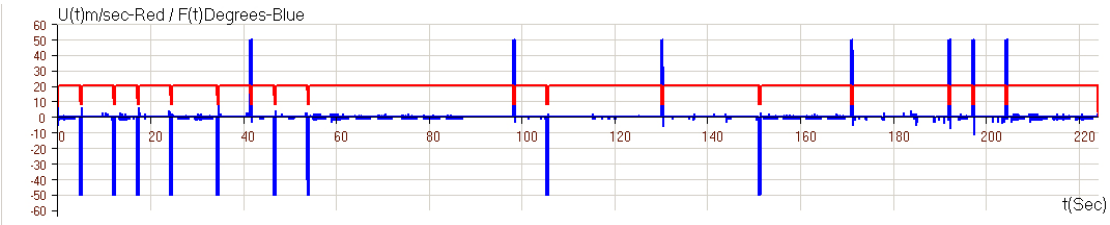


Σχήμα 5.1.3 Φόρμα συντεταγμένων κίνησης και εμποδίων

Η φόρμα αυτή (σχήμα 5.1.3) έχει σαν στόχο να καλύψει τις παρακάτω λειτουργίες:

- Σχεδίαση μονοπατιών του αλγόριθμου σχεδιασμού διαδρομής A*. Συγκεκριμένα μπορούμε να διακρίνουμε για κάθε πιθανό σενάριο πλοήγησης, το διακριτό μονοπάτι που παράγει ο A* λαμβάνοντας υπόψη τα εμπόδια του χώρου του ρομποτικού συστήματος.
- Σχεδίαση του συνεχούς μονοπατιού αν ενεργοποιήσουμε μόνο τον ελεγκτή κίνησης χωρίς βέβαια να λάβουμε υπόψη τα εμπόδια.
- Σχεδίαση του τελικού μονοπατιού πλοήγησης, λαμβάνοντας υπόψη όλα τα εμπόδια του χώρου κίνησης του ρομποτικού μας συστήματος.
- Σχεδίαση εμποδίων στο χώρο του ρομποτικού μας συστήματος. Αυτό επιτυγχάνεται αν με το ποντίκι σύρουμε το δείκτη, ενώ έχουμε πατημένο το αριστερό κουμπί του ποντικιού και το shift.
- Επιλογή με διπλό κλικ των επιθυμητών συντεταγμένων στόχου. Με πατημένο το δεξί κλικ του ποντικιού εμφανίζονται οι υποψήφιες συντεταγμένες τερματισμού στον χώρο κίνησης.
- Με την επιλογή του resize η φόρμα αυτή μεγενθύνεται στο μέγιστό της. Αυτό προσφέρεται για να έχουμε καλύτερη και ευκρινέστερη εικόνα του χώρου και των μονοπατιών που σχεδιάζονται.

5.1.3 Φόρμα ταχυτήτων – Γωνιών στροφής

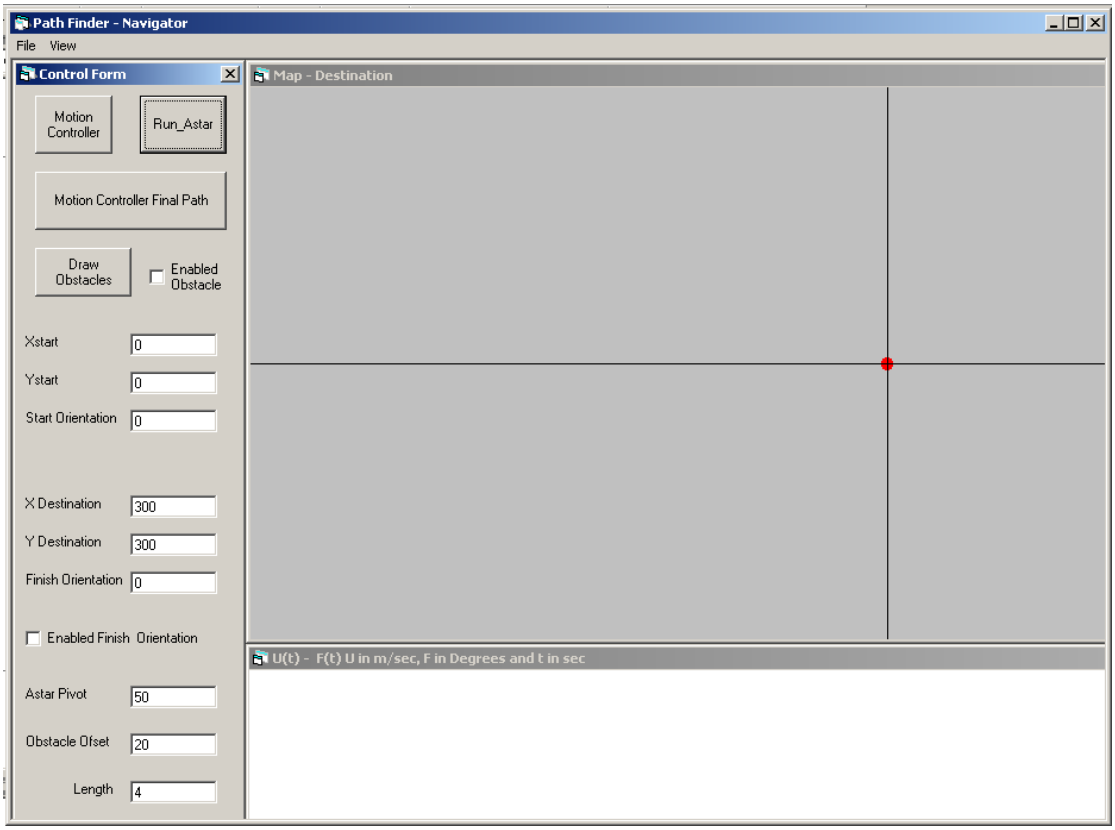


Σχήμα 5.1.4 Φόρμα ταχυτήτων - γωνιών στροφής

Η φόρμα αυτή (σχήμα 5.1.4) χρησιμοποιείται για να αναπαραστήσουμε τις τιμές των μεταβλητών ελέγχου, όπως ορίζονται από τον ελεγκτή κίνησης, ώστε το ρομποτικό μας σύστημα να ακολουθήσει το μονοπάτι που του ανατέθηκε. Ουσιαστικά, είναι ένα διάγραμμα που μας δείχνει σε κάθε χρονική στιγμή τη γωνία στροφής που πρέπει να έχει το τιμόνι του ρομποτικού μας συστήματος και την ταχύτητα που πρέπει να έχει το ρομπότ για να ακολουθήσει την επιθυμητή διαδρομή.

Με το ποντίκι μπορούμε να σύρουμε το δείκτη πάνω στη φόρμα και συγκεκριμένα να ορίσουμε το τμήμα του πεδίου που θέλουμε να δούμε πιο αναλυτικά. Το σύστημα θα εστιάσει στην επιλεγμένη περιοχή. Με διπλό πάτημα μπορούμε να επαναφέρουμε το διάγραμμα στις αρχικές του διαστάσεις.

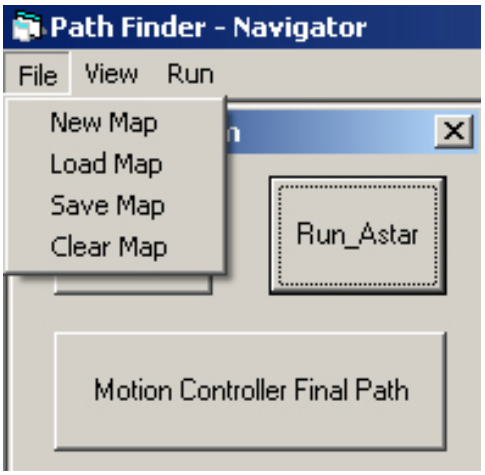
5.1.4 Φόρμα πλοήγησης



Σχήμα 5.1.5 Φόρμα πλοήγησης

Η φόρμα εργασίας αυτή (σχήμα 5.1.5) κρατάει ευθυγραμμισμένες τις υπόλοιπες φόρμες που αναφέραμε. Επιπλέον, περιορίζει την κίνηση τους πέρα από τα όρια της. Τέλος, μέσω του μενού που βρίσκεται στο πάνω αριστερό μέρος της φόρμας μπορούμε να εκτελέσουμε επιπλέον λειτουργίες που απαιτούνται κατά τη χρήση.

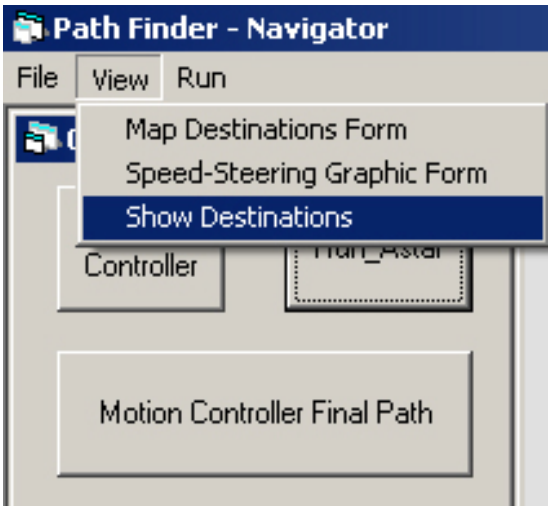
Τα μενού αποτελείται από τρεις βασικές κατηγορίες **File** και **View** και **Run**:



Σχήμα 5.1.6 Φόρμα πλοήγησης επιλογής File

File: Η επιλογή αυτή (σχήμα 5.1.6) μας οδηγεί στις επόμενες επιλογές, που έχουν στόχο να χειριστούν τα δεδομένα των πινάκων που αφορούν τα εμπόδια.

- **New Map:** Με την επιλογή αυτή διαγράφουμε κάποιον ήδη φορτωμένο πίνακα εμποδίων και μπορούμε να δημιουργήσουμε έναν καινούριο στη φόρμα των συντεταγμένων.
- **Load Map:** Μπορούμε να φορτώσουμε έναν οποιοδήποτε αποθηκευμένο πίνακα εμποδίων στη φόρμα των συντεταγμένων.
- **Save Map:** Αποθηκεύουμε έναν πίνακα εμποδίων.
- **Clear Map:** Καθαρίζουμε τη φόρμα συντεταγμένων.

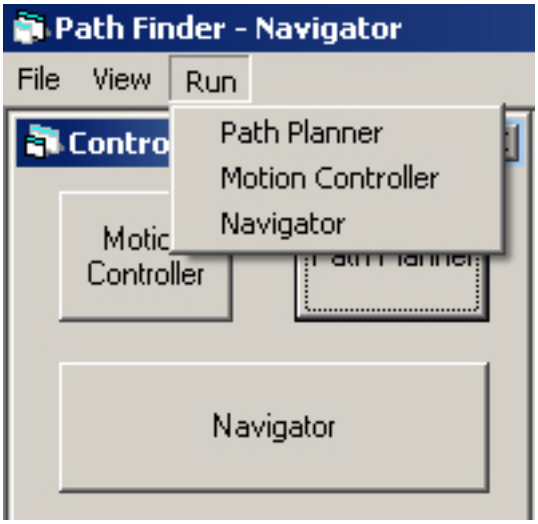


Σχήμα 5.1.7 Φόρμα πλοήγησης επιλογής View

View: Η επιλογή αυτή (σχήμα 5.1.7) μας οδηγεί σε τρεις επιλογές που στόχο έχουν να επαναφέρουν εάν έχουν ήδη χαθεί από το περιβάλλον (σε περίπτωση διαγραφής), τις δύο βασικές φόρμες, τη φόρμα συντεταγμένων κίνησης και τη φόρμα των ταχυτήτων. Επίσης, να μας σχεδιάζει τις αρχικές και τελικές συντεταγμένες.

- **Map Destination:** Επαναφέρει την αντίστοιχη φόρμα
- **Speed – Steering Graph:** Επαναφέρει την αντίστοιχη φόρμα
- **Show Origin and Destinations:** Εμφανίζει τις αρχικές και τις τελικές συντεταγμένες πλοήγησης στη φόρμα συντεταγμένων.

Run: Η επιλογή αυτή (σχήμα 5.1.8) μας οδηγεί σε ένα μενού που μας δίνει έναν εναλλακτικό τρόπο να τρέχουμε τους αντίστοιχους αλγόριθμους.



Σχήμα 5.1.8 Φόρμα πλοήγησης επιλογή Run

- **Path Planner:** Ενεργοποιείται ο αλγόριθμος σχεδιασμού διαδρομής
- **Motion Controller:** Ενεργοποιείται ο ελεγκτής κίνησης
- **Navigator:** Ενεργοποιείται η πλοήγηση συνολικά

Κεφάλαιο 6 Αποτελέσματα προσομοίωσης

Έχοντας μελετήσει και κατανοήσει το πρόβλημα της πλοήγησης, καθώς και τους τρόπους αντιμετώπισης του, σ' αυτό το κεφάλαιο θα προσπαθήσουμε να καλύψουμε όσο το δυνατόν μεγαλύτερο μέρος των περιπτώσεων προσομοίωσης του ρομποτικού συστήματος που εξετάσαμε. Πρέπει να αναφερθεί ότι η συγκεκριμένη υλοποίηση έχει αναπτυχθεί λαμβάνοντας υπόψη ότι το ρομποτικό σύστημα κινείται σε ένα προκαθορισμένο πεδίο κίνησης (εμπόδια, όρια κίνησης, κλπ).

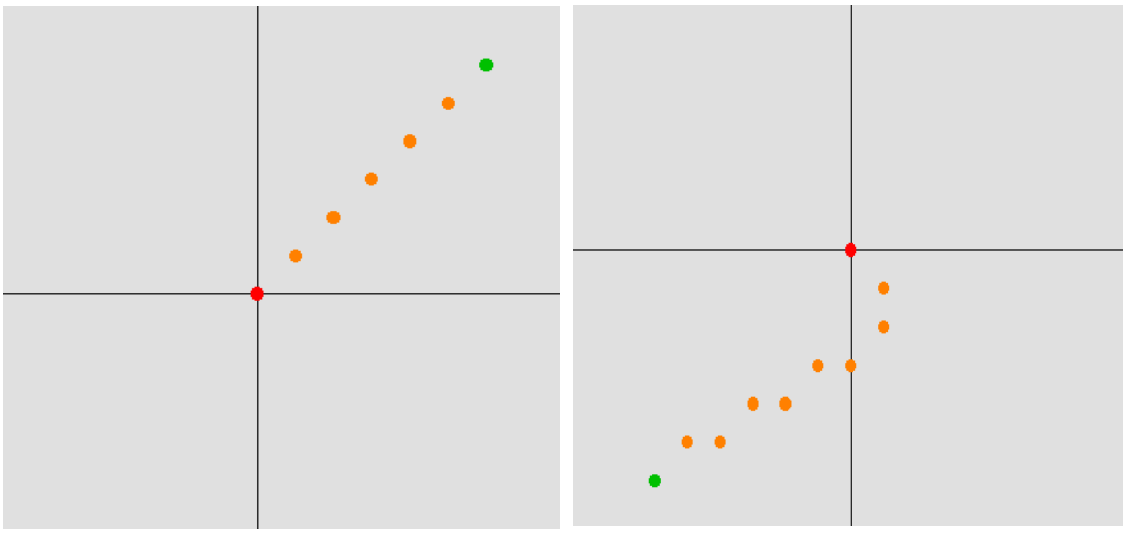
Επιπλέον, με τη συγκεκριμένη υλοποίηση προσομοιώνουμε θεωρητικά τη θέση, τη διεύθυνση και την ταχύτητα του ρομποτικού συστήματος, σύμφωνα με τις εξισώσεις κίνησης και τις τεχνικές των αλγορίθμων που μελετήσαμε. Στην πράξη, η θέση, η διεύθυνση και η ταχύτητα ενός πραγματικού ρομποτικού συστήματος που λειτουργεί σε φυσικά περιβάλλοντα εργασίας επηρεάζονται σε μεγάλο βαθμό από πολλούς παράγοντες, συγκεκριμένα, από την ακρίβεια των μετρητικών οργάνων, των συστημάτων ανάδρασης, το θόρυβο από τους αισθητήρες, την τριβή του εδάφους κλπ.

6.1 Σχεδιασμός διαδρομής

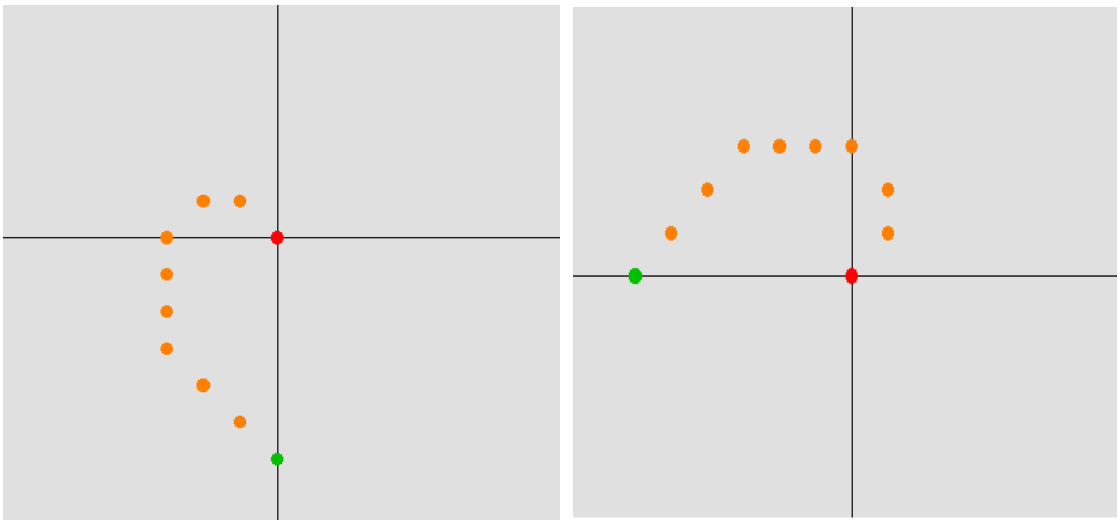
Όπως αναφέραμε στην ενότητα 4.1.1 θεωρήσαμε ότι στο ρομποτικό μας σύστημα επιτρέπονται μόνο κινήσεις προς τα μπρος και επιπλέον περιορίζονται στις 45° δεξιά και αριστερά από τη διεύθυνση κίνησης του. Σύμφωνα με αυτές τις παραδοχές θα προσπαθήσουμε να εξετάσουμε όσο το δυνατόν περισσότερες περιπτώσεις σχεδιασμού διαδρομής για διαφορετικές συντεταγμένες, σενάριο εμποδίων και βαθμό δυσκολίας. Σ' αυτό το στάδιο της προσομοίωσης τα γεωμετρικά χαρακτηριστικά του ρομποτικού συστήματος και το χρονικό βήμα ελέγχου δεν επηρεάζουν το αποτέλεσμα. Κατά συνέπεια οι τιμές για το L, DT δεν λαμβάνονται υπόψη. Αντίθετα οι τιμές για το βήμα Astar Pivot και το Obstacle Offset είναι καθοριστικές για τα αποτελέσματα της προσομοίωσης.

6.1.1 Παράδειγμα σχεδιασμού διαδρομής 1

Θα ξεκινήσουμε από μερικά απλά παραδείγματα όπου το ρομποτικό σύστημα κινείται στο χώρο χωρίς την παρουσία εμποδίων. Επίσης, θα δούμε πως ο αλγόριθμος υπολογίζει τα μονοπάτια, όταν δεν μας ενδιαφέρει η διεύθυνση με την οποία το ρομποτικό όχημα θα φτάσει στο στόχο. Το βήμα διακριτοποίησης που έχουμε επιλέξει είναι Astar Pivot=50.



Σχήμα 6.1.1Left: Start(0, 0, 0)[1], Destination (300, 300, X).
Right: Start (0, 0, 0), Destination (-300, -300, X).



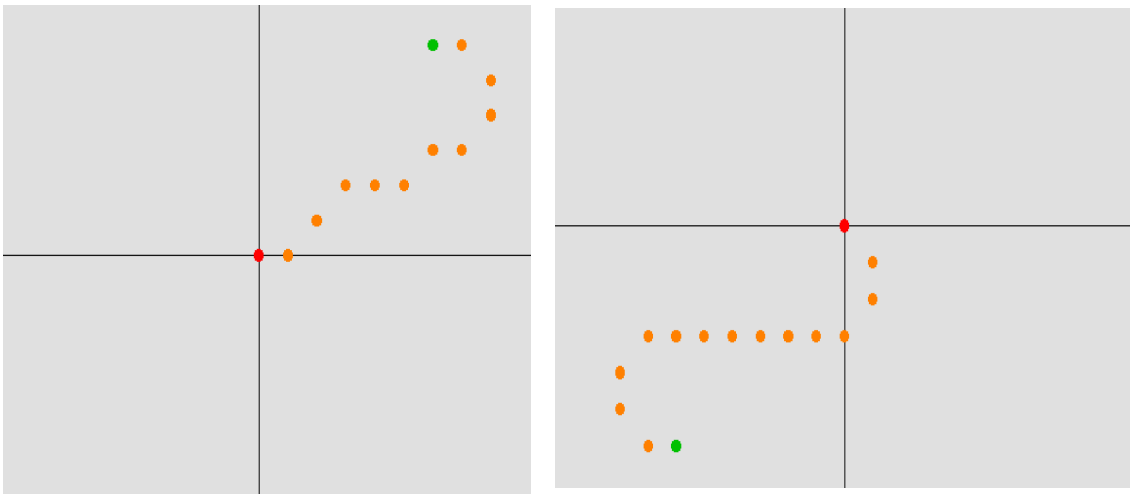
Σχήμα 6.1.2 Left: Start(0, 0, 90), Destination (0, -300, X).
Right: Start (0, 0, 0), Destination (-300, 0, X).

Σε κάθε μια περίπτωση από αυτές που φαίνονται σχήματα 6.1.1 και 6.1.2, ο αλγόριθμος σχεδιασμού διαδρομής αποφασίζει με βάση το κόστος βημάτων για να επιτύχει την πιο σύντομη διαδρομή. Έτσι, το αποτέλεσμα είναι ένα βέλτιστο μονοπάτι που θα οδηγήσει στο στόχο με τον καλύτερο δυνατό τρόπο.

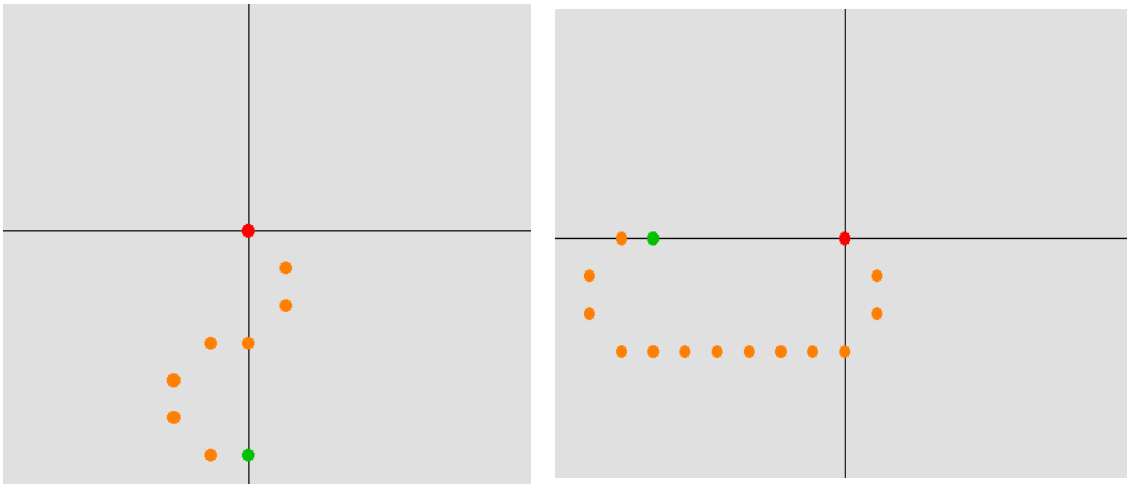
[1] Αρχικές συντεταγμένες Start(Xstart,Ystart, Θstart). Το πεδίο Θstart=X σημαίνει ότι δεν έχει προσδιοριστεί η συγκεκριμένη παράμετρος και μπορεί να πάρει μια οποιαδήποτε τιμή.

6.1.2 Παράδειγμα σχεδιασμού διαδρομής 2

Στα παρακάτω παραδείγματα εξετάζουμε τις ίδιες περιπτώσεις με το παράδειγμα 1 μόνο που εδώ τα μονοπάτια υπολογίζονται και αποφασίζονται με βάση την επιθυμητή τελική διεύθυνση του οχήματος. Το βήμα αναζήτησης έχει επιλεχθεί Astar Pivot =50.



Σχήμα 6.1.3 Left: Start(0, 0, 0), Destination (300, 300, 180).
Right: Start (0, 0, 0), Destination (-300, -300, 0).

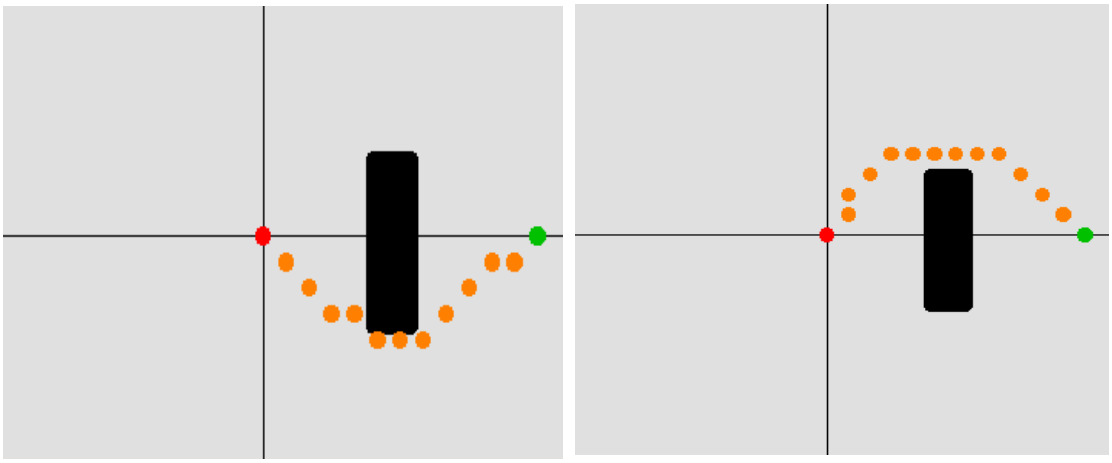


Σχήμα 6.1.4 Left: Start(0, 0, 0), Destination (0, -300,0).
Right: Start (0, 0, 0), Destination (-300, 0, 0).

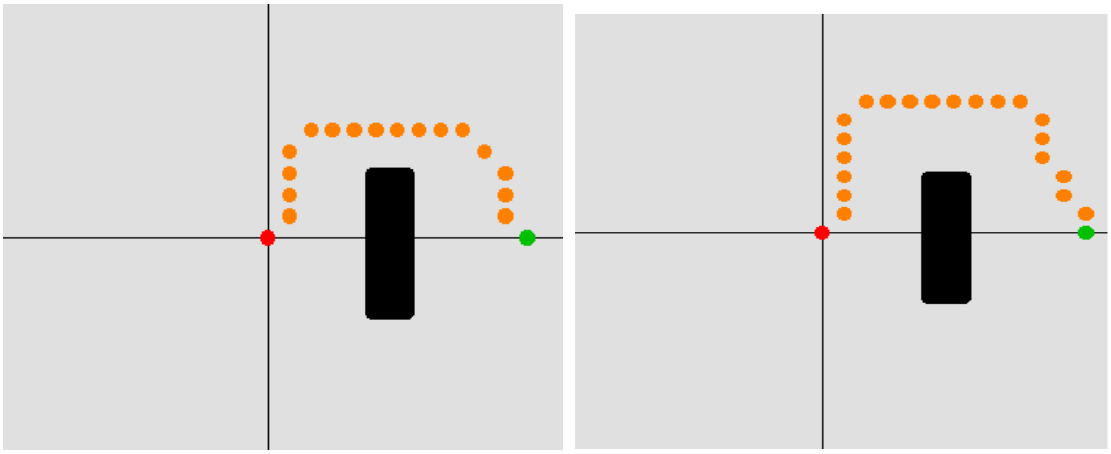
Παρατηρούμε λοιπόν, ότι σε όλες τις παραπάνω περιπτώσεις (σχήματα 6.1.3 και 6.1.4) ο αλγόριθμος υπολογίζει βέλτιστα μονοπάτια ανάλογα με τις παραμέτρους που ορίζουμε και με βάση την επιθυμητή τελική διεύθυνση. Συγκεκριμένα, ο αλγόριθμος σ' αυτές τις περιπτώσεις θα βρει βέλτιστο μονοπάτι όχι όμως μόνο με βάση το κόστος της διαδρομής, αλλά και με βάση την τελική διεύθυνση που θα θέλαμε να έχει το ρομποτικό μας σύστημα στις τελικές συντεταγμένες.

6.1.3 Παράδειγμα σχεδιασμού διαδρομής 3

Στο παράδειγμα αυτό εξετάζουμε τη συμπεριφορά του αλγόριθμου υπό την παρουσία εμποδίων. Επίσης, εξετάζουμε πως η παράμετρος Obstacle Offset μεταβάλλει τον τρόπο με τον οποίο ρομποτικό σύστημα αποφεύγει τα εμπόδια. Το βήμα αναζήτησης έχει επιλεχθεί Astar Pivot=50.



Σχήμα 6.1.5 Left: Obstacle_Offset=0, Start(0, 0, 0), Destination (600, 0, X).
Right: Obstacle_Offset=50 Start(0, 0, 0), Destination (600, 0, X).

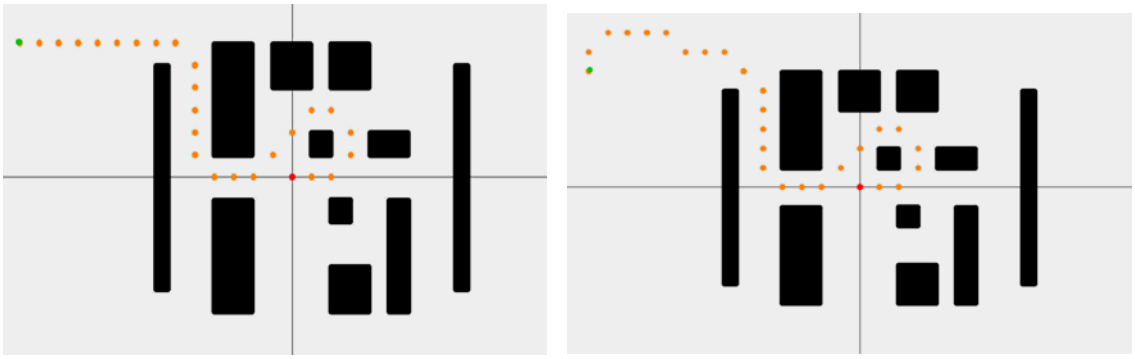


Σχήμα 6.1.6 Left: Obstacle_Offset=100, Start(0, 0, 0), Destination (600, 0, X).
Right: Obstacle_Offset=150 Start (0, 0, 0), Destination (600, 0, X).

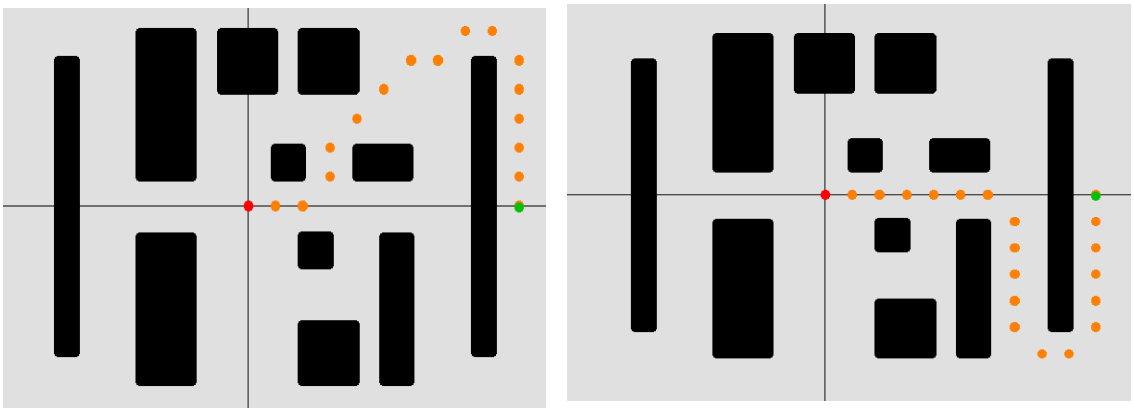
Από τα αποτελέσματα της προσομοίωσης (σχήματα 6.1.5 και 6.1.6) παρατηρούμε ότι το ρομποτικό μας σύστημα αποφεύγει ένα εμπόδιο που βρίσκεται ακριβώς μπροστά του. Επιπλέον, όσο μεγαλύτερη είναι η παράμετρος Obstacle Offset τόσο μεγαλύτερη είναι η απόσταση που θα διατηρήσει από το εμπόδιο για να το αποφύγει.

6.1.4 Παράδειγμα σχεδιασμού διαδρομής 4

Σ' αυτό το παράδειγμα εξετάζουμε δυο περιπτώσεις μονοπατιών μετ' εμποδίων λαμβάνοντας και μη λαμβάνοντας υπόψη τη διεύθυνση τερματισμού. Το βήμα έχει επιλεχθεί Astar Pivot=50.



Σχήμα 6.1.7 Left: Obstacle_Offset=20, Start(0, 0, 0), Destination (600, -300, X).
Right: Obstacle_Offset=20 Start (0, 0, 0), Destination (600, -300, -90).

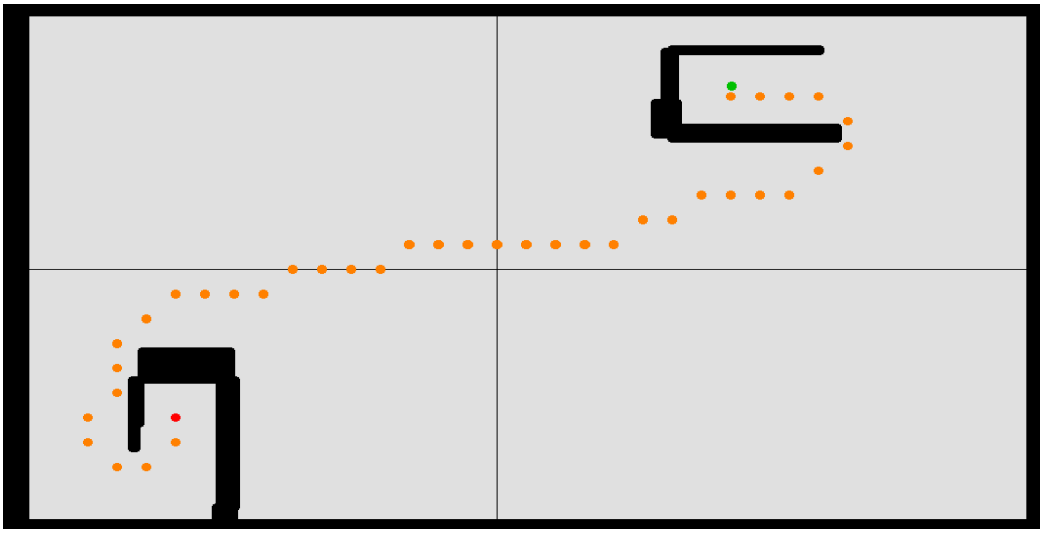


Σχήμα 6.1.8 Left: Obstacle_Offset=20, Start(0, 0, 0), Destination (500, 0,, X).
Right: Obstacle_Offset=20 Start (0, 0, 0), Destination (500, 0, -90).

Παρατηρούμε λοιπόν πως υπολογίζονται τα μονοπάτια αν δεν μας ενδιαφέρει ο τρόπος τερματισμού στο στόχο και πώς αν μας ενδιαφέρει (σχήματα 6.1.7 και 6.1.8). Ο αλγόριθμος πάντα αποφασίζει με βέλτιστο τρόπο τον υπολογισμό του μονοπατιού απορρίπτοντας διαδρομές με εμπόδια και λαμβάνοντας υπόψη όλες τις παραμέτρους που ορίσαμε για τις συγκεκριμένες προσομοιώσεις.

6.1.5 Παράδειγμα σχεδιασμού διαδρομής 5

Σ' αυτό το παράδειγμα έχουμε προσομοιώσει μια περίπτωση αποστάθμευσης από ένα χώρο με στάθμευση σε έναν άλλο.

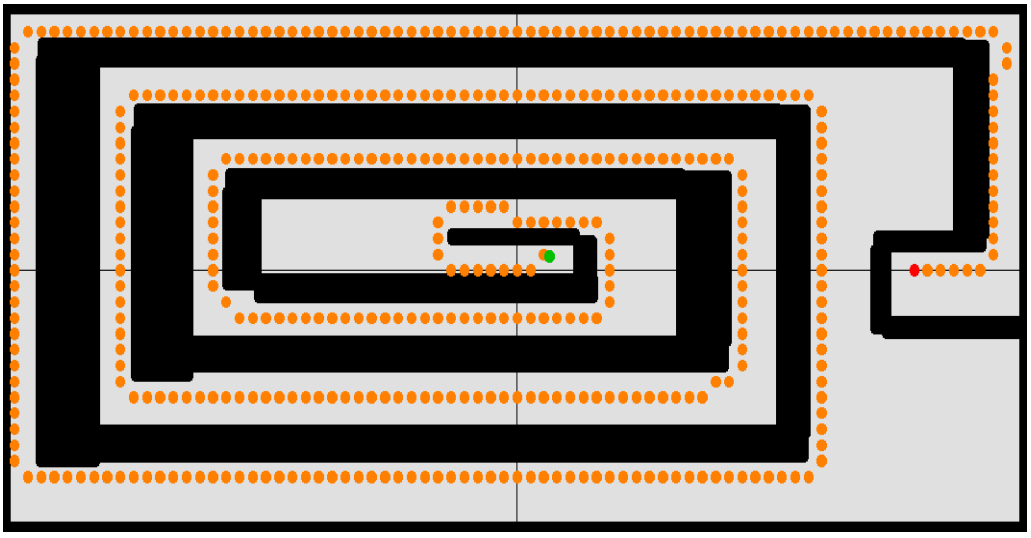


Σχήμα 6.1.9 Astar Pivot=100, Obstacle_Offset=100, Start(-1100, -600,-90), Destination (800,700, X).

Ο τρόπος με τον οποίο ο αλγόριθμος αποφασίζει για το μονοπάτι φαίνεται αναλυτικά στο σχήμα 6.1.9. Η διαδρομή που ακολουθείται, όπως φαίνεται και στο παραπάνω σχήμα, είναι βέλτιστη.

6.1.6 Παράδειγμα σχεδιασμού διαδρομής 6

Στην περίπτωση αυτή εξετάζουμε τον σχεδιασμό διαδρομής σε ένα σύνθετο ελικοειδή λαβύρινθο.

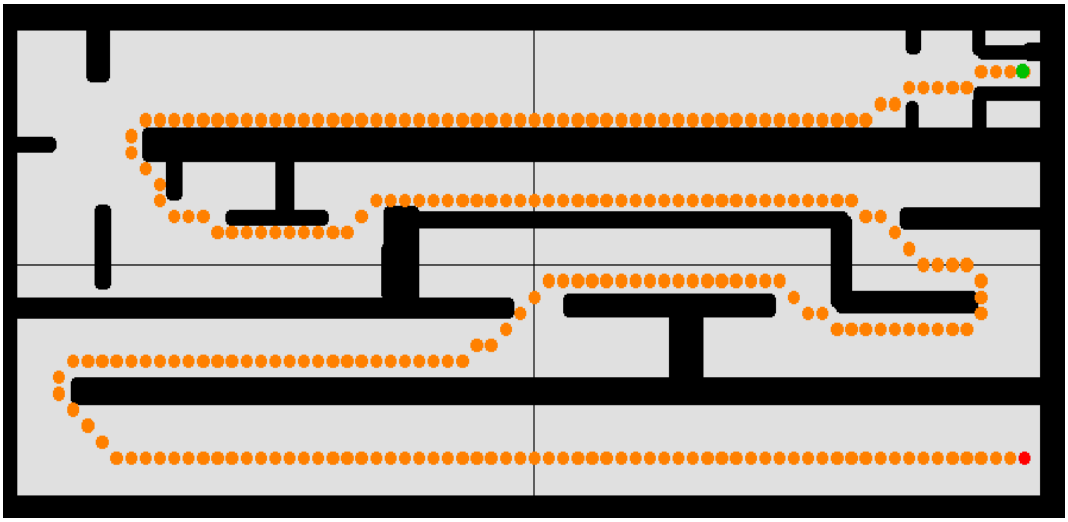


Σχήμα 6.1.10 Astar Pivot =100, Obstacle_Offset=50, Start(-1100, -600, 0), Destination (150, 150, X).

Οι λεπτομέρειες της διαδρομής από τον αλγόριθμο σχεδιασμού διαδρομής φαίνονται στο σχήμα 6.1.10.

6.1.7 Παράδειγμα σχεδιασμού διαδρομής 7

Όμοια με το παραπάνω παράδειγμα μπορούμε να εξετάσουμε μια περίπτωση τυπικού λαβυρίνθου.

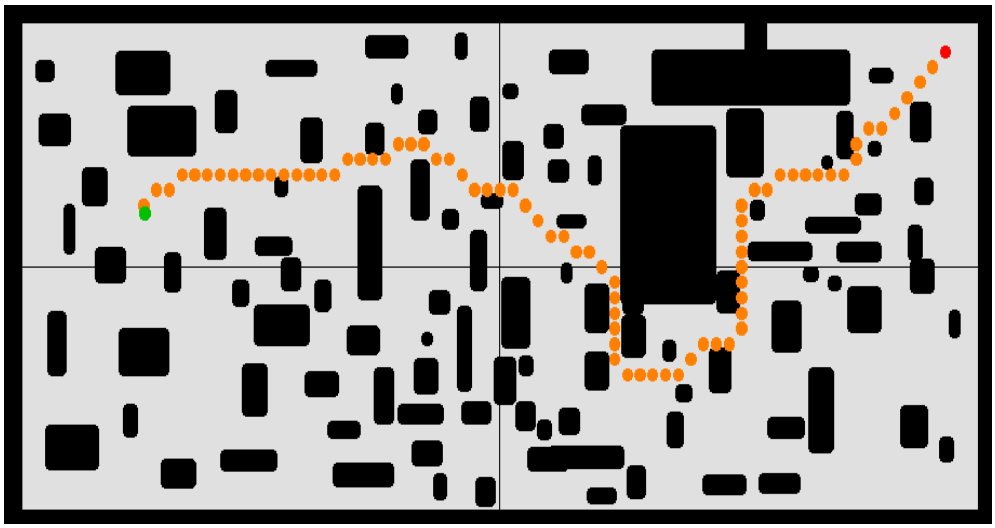


Σχήμα 6.1.11 Astar Pivot =100, Obstacle_Offset=50, Start(1700, 800, 0), Destination (1700, -800, X)

Οι λεπτομέρειες της διαδρομής από τον αλγόριθμο σχεδιασμού διαδρομής φαίνονται στο σχήμα 6.1.11. Ο αλγόριθμος καταφέρνει να βρει το μοναδικό μονοπάτι που οδηγεί στο στόχο.

6.1.8 Παράδειγμα σχεδιασμού διαδρομής 8

Σ' αυτό το παράδειγμα έχουμε σχεδιάσει ένα εντελώς τυχαίο πεδίο εμποδίων όπου το γυμνό μάτι δύσκολα μπορεί να βρει τη βέλτιστη διαδρομή που οδηγεί στο στόχο.



Σχήμα 6.1.12 Astar Pivot =100, Obstacle_Offset=50, Start(1700, 800, 180), Destination (-300, 200, X)

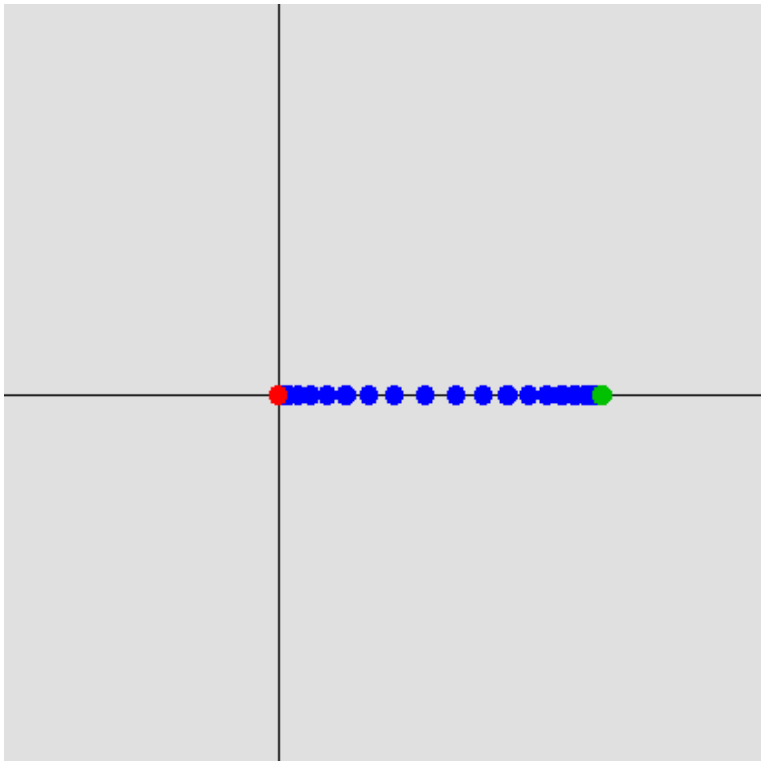
Οι λεπτομέρειες της διαδρομής από τον αλγόριθμο σχεδιασμού διαδρομής φαίνονται στο σχήμα 6.1.12.

6.2 Έλεγχος Κίνησης

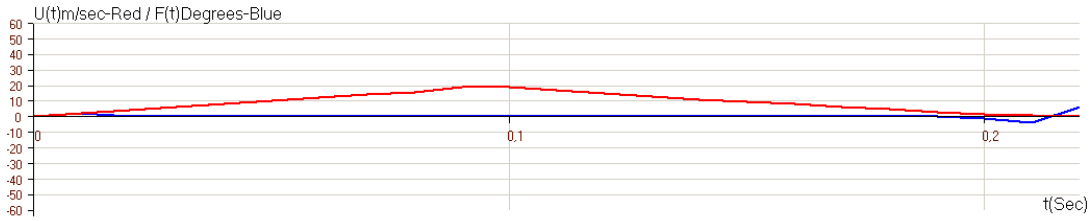
Στην ενότητα 4.2 μελετήσαμε και περιγράψαμε τον τρόπο λειτουργίας και τις δυνατότητες του αλγορίθμου ελεγκτή κίνησης. Στο κεφάλαιο αυτό θα δούμε τα αποτελέσματα της προσομοίωσης του ελεγκτή κίνησης στο εν λόγω ρομποτικό σύστημα. Πρέπει να προσθέσουμε ότι για τις ανάγκες της προσομοίωσης, υποθέσαμε ότι η μέγιστη ταχύτητα με την οποία μπορεί να κινηθεί το ρομποτικό σύστημα μας δεν μπορεί να υπερβαίνει τα 20m/sec και η γωνία στροφής είναι μεταξύ των -45° και 45°. Ο χρόνος κάθε βήματος ελέγχου είναι 10msec. Οι τιμές αυτές επιλέχθηκαν έτσι ώστε να πληρούν προϋποθέσεις λειτουργίας σε πραγματικές συνθήκες.

6.2.1 Παράδειγμα ελέγχου κίνησης 1

Αρχικά εξετάζουμε την πιο απλή περίπτωση όπου δεν χρειάζεται να αλλάξει η γωνία στροφής του οχήματος.



Σχήμα 6.2.1 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 0, X)



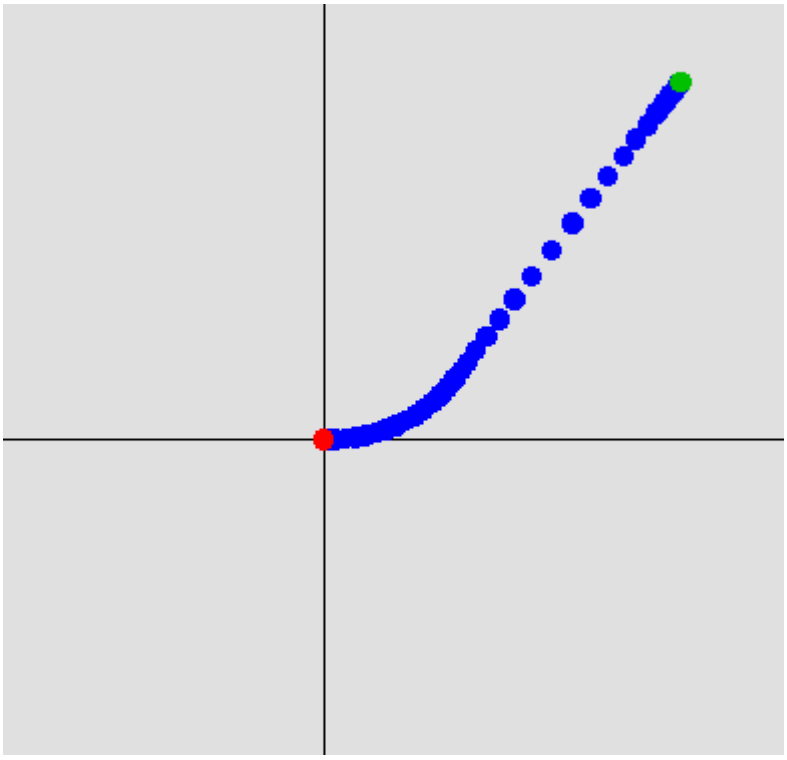
Σχήμα 6.2.2 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 0, X)

Από τα αποτελέσματα της προσομοίωσης (σχήματα 6.2.1 και 6.2.2) φαίνεται ότι ο ελεγκτής κίνησης θα ξεκινήσει να επιταχύνει προοδευτικά το ρομποτικό

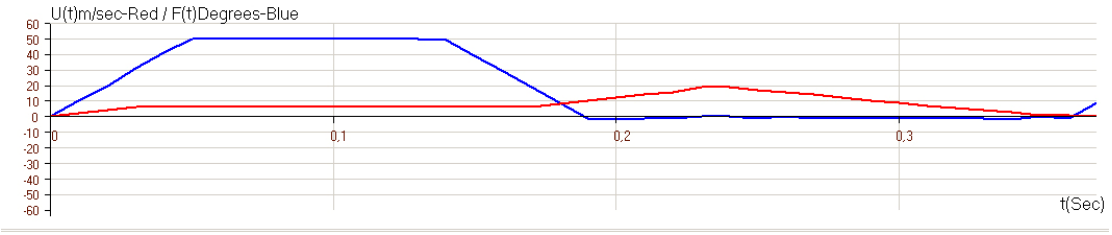
σύστημα μέχρι μια μέγιστη τιμή ταχύτητας. Στη συνέχεια το ρομποτικό σύστημα θα ξεκινήσει να επιβραδύνει προοδευτικά μέχρι η ταχύτητα να γίνει μηδενική, οπότε έχει φτάσει στο στόχο. Μπορούμε να διακρίνουμε επίσης την ομαλότητα των τιμών των παραμέτρων του συστήματος καθώς και την ακρίβεια με την οποία προσέγγισε το σημείο τερματισμού με μηδενική ταχύτητα.

6.2.2 Παράδειγμα ελέγχου κίνησης 2

Στο παράδειγμα αυτό εξετάζουμε μια ενδιάμεση περίπτωση όπου έχουμε και αλλαγή κατεύθυνσης του ρομποτικού συστήματος.



Σχήμα 6.2.3 L=1 dt=0,01, Start(0, 0, 0), Destination (2, 2, X).

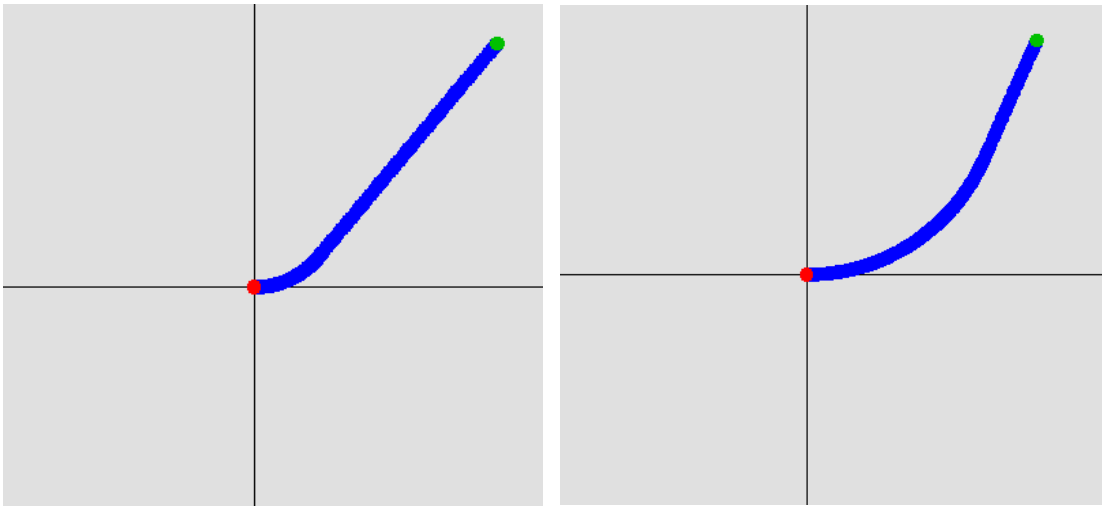


Σχήμα 6.2.4 L=1, dt=0,01, Start(0, 0, 0), Destination (2, 2, X)

Στα σχήματα 6.2.3 και 6.2.4 είναι φανερή η ομαλότητα κίνησης του ρομποτικού συστήματος σε κάθε χρονική στιγμή. Επίσης, παρατηρούμε ότι κατά το διάστημα που αλλάζει πορεία, ο αλγόριθμος δεν επιτρέπει η ταχύτητα να ξεπεράσει ένα άνω όριο που αποφασίζεται από τον αλγόριθμο.

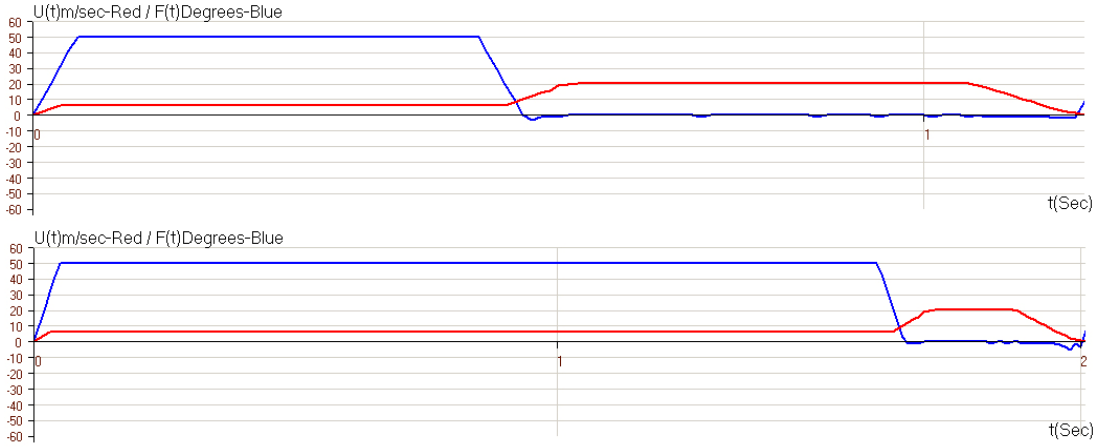
6.2.3 Παράδειγμα ελέγχου κίνησης 3

Στο παράδειγμα αυτό θα δούμε πως αλλάζει η μέγιστη γωνία στροφής σε σχέση με το μήκος L του ρομποτικού οχήματος. Εξετάζουμε δύο περιπτώσεις για δεδομένες συντεταγμένες τερματισμού.



Σχήμα 6.2.5 L=4 (Left), L=10 (Right), dt=0,01, Start(0, 0, 0), Destination (10, 10, X).

Σύμφωνα με τα αποτελέσματα που λάβαμε (σχήμα 6.2.5) και σε συμφωνία με τις κινηματικές εξισώσεις που εξάγαμε στην ενότητα 4.2, παρατηρούμε ότι όσο μεγαλώνει το μήκος του οχήματος τόσο μεγαλώνει και η καμπύλη μέγιστης δυνατής στροφής του οχήματος.

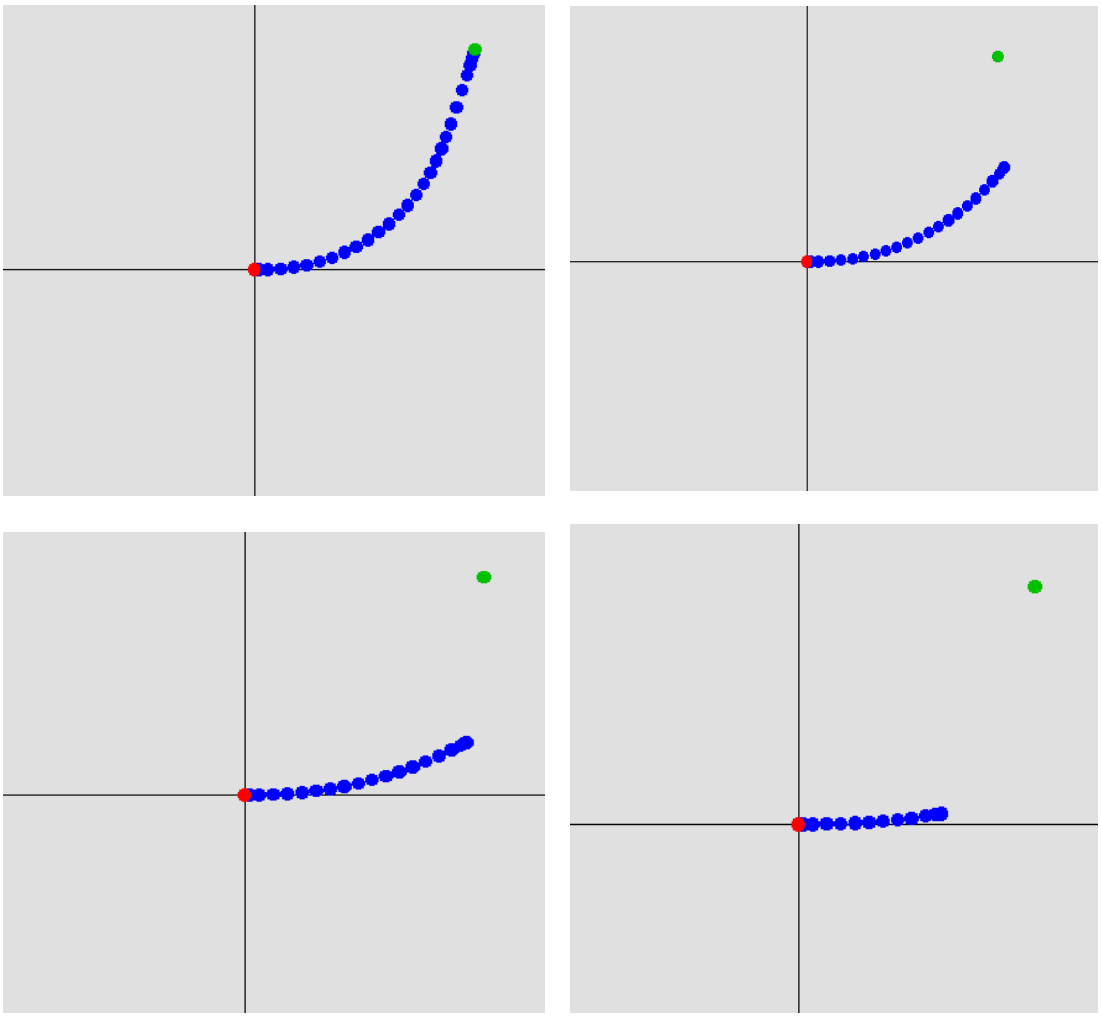


Σχήμα 6.2.6 L=4(Up), L=10(Down), dt=0,01, Start(0, 0, 0), Destination (10, 10, X).

Επιπλέον, παρατηρούμε ότι στην πρώτη περίπτωση που το μήκος του ρομποτικού οχήματος είναι μικρότερο, ο χρόνος που χρειάζεται το όχημα να φτάσει στον στόχο είναι μικρότερος απ' ότι στη δεύτερη περίπτωση. Αυτό συμβαίνει γιατί όπως μπορούμε να παρατηρήσουμε από τα σχήματα 6.2.5 και 6.2.6 για μεγαλύτερο μήκος οχήματος χρειαζόμαστε περισσότερο χρόνο μέχρι το όχημα να περιστραφεί. Επαληθεύουμε λοιπόν τις κινηματικές εξισώσεις κίνησης που χρησιμοποιήθηκαν και υλοποιήθηκαν.

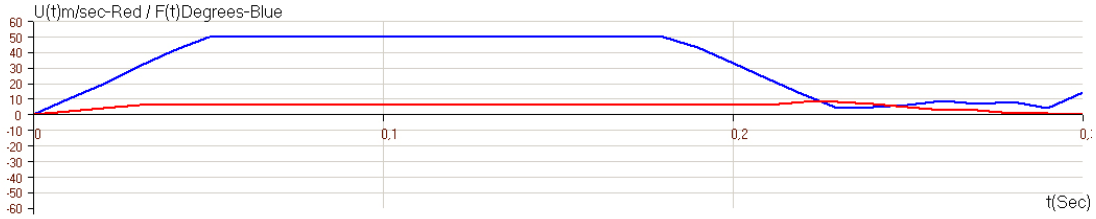
6.2.4 Παράδειγμα ελέγχου κίνησης 4

Στο παράδειγμα αυτό θα δούμε πώς για διαφορετικά μήκη οχήματος L, ο αλγόριθμος δεν καταφέρνει να φτάσει στον επιθυμητό στόχο.

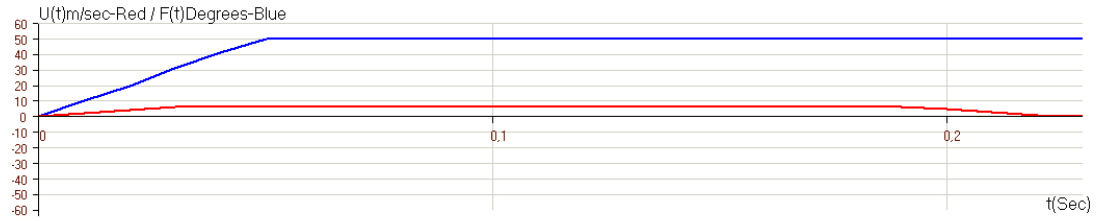


Σχήμα 6.2.7 L=1 (Up Left), L=1.5 (Up Right).
L=2 (Down Left), L=4 (Down Right), dt=0,01, Start(0, 0, 0), Destination (10, 10, X).

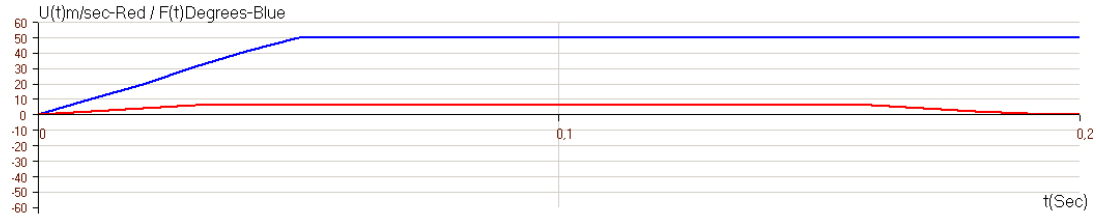
Όπως είναι αναμενόμενο όσο μεγαλώνει το μήκος του οχήματος τόσο πιο νωρίς σταματάει το όχημα ο ελεγκτής κίνησης (σχήμα 6.2.6). Αυτό το κάνει όταν αντιληφθεί ότι δεν θα καταφέρει να φτάσει στον στόχο διαλέγοντας ταχύτητες που ολοένα και φθίνουν μέχρι που μηδενίζονται. Το αποτέλεσμα αυτό είναι απόφαση της αντικειμενικής συνάρτησης κόστους που για να είναι σε κάθε βήμα ελάχιστη πρέπει οι ταχύτητες να μειώνονται και η γωνία στροφής να αυξάνεται.



a. L=1



b. L=1.5



c. L=2

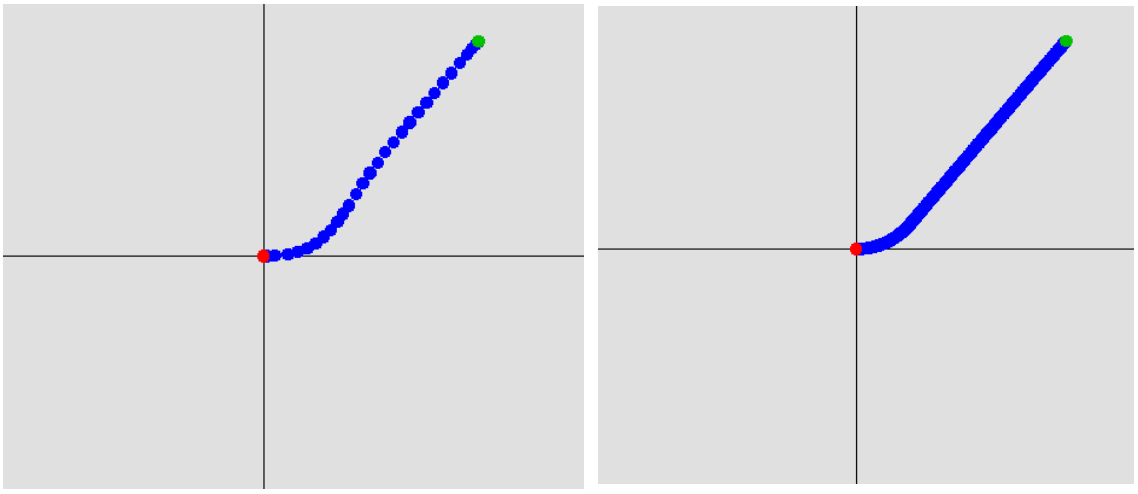
d. L=4

Σχήμα 6.2.8 dt=0,01, Start(0, 0, 0), Destination (10, 10, X).

Στο σχήμα 6.2.8 βλέπουμε αναλυτικά με πιο τρόπο ο ελεγκτής κίνησης θα επιλέξει τις συγκεκριμένες τιμές ελέγχου μέχρι να σταματήσει. Επιπλέον, μπορούμε να δούμε αναλυτικά σε ποιες χρονικές στιγμές και για ποιο διάστημα το όχημα στριβεί μέχρι να ακινητοποιηθεί τελείως. Γενικότερα, ο αλγόριθμος αυτός αποτυγχάνει σε πολλές περιπτώσεις όπου ο στόχος βρίσκεται σε διεύθυνση τέτοια ώστε η τρέχουσα διεύθυνση του ρομπότ δεν του επιτρέπει να κινηθεί προς το στόχο. Η αντικειμενική συνάρτηση είναι αυτή που θα ορίσει πότε και αν μπορεί να εκτελέσει μια οποιαδήποτε τέτοια κίνηση. Γενικότερα, τέτοιου είδους σενάρια δεν επιτρέπουμε να επιλύονται μεμονωμένα από τον ελεγκτή κίνησης, γιατί ο αλγόριθμος αναζήτησης μονοπατιού είναι αυτός που θα αποφασίσει τον τρόπο με τον οποίο θα προσεγγίσει το ρομπότ τις επιθυμητές συντεταγμένες.

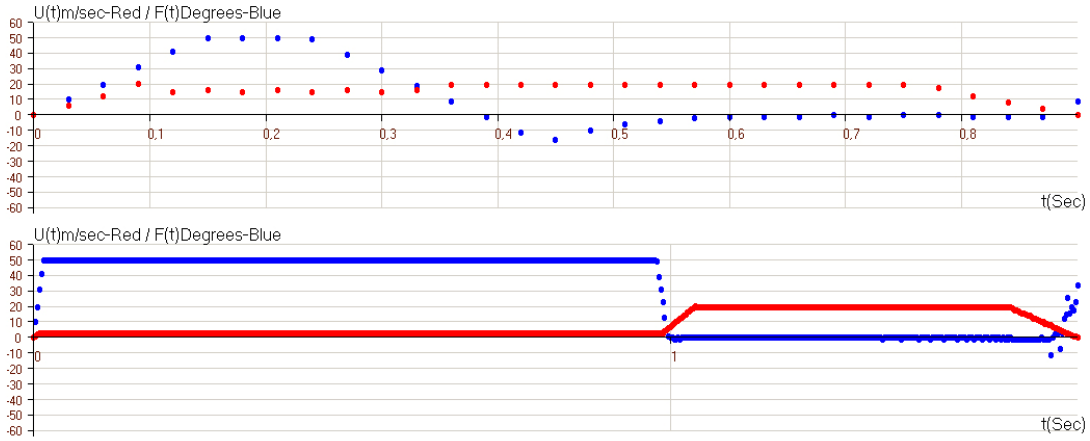
6.2.5 Παράδειγμα ελέγχου κίνησης 4

Στο παράδειγμα αυτό ενεργοποιούμε τον ελεγκτή κίνησης με. διαφορετικές χρονικές μονάδες ελέγχου dt σε κάθε βήμα του αλγορίθμου.



Σχήμα 6.2.8 dt=0,03 (Left), dt=0,003(Right), Start(0, 0, 0), Destination (10, 10, X).

Για δύο διαφορετικές χρονικές μονάδες του ελεγκτή κίνησης παρατηρούμε ότι τα σημεία εξόδου διαφέρουν σε πλήθος.



Σχήμα 6.2.9 dt=0,03(Up) , dt=0,003(Down), Start(0, 0, 0), Destination (10, 10, X).

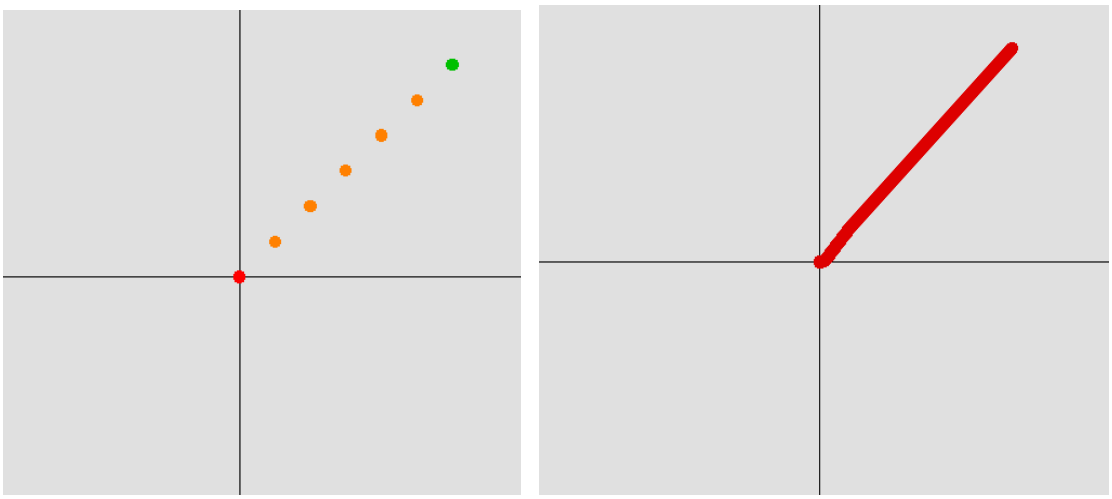
Όπως φαίνεται από τα σχήματα 6.2.8 και 6.2.9 για μικρότερες μονάδες χρόνου έχουμε πολύ περισσότερα σημεία ελέγχου των παραμέτρων και κατά συνέπεια μεγαλύτερη ακρίβεια ως προς τις τιμές των παραμέτρων που πρέπει να δοθούν στο ρομποτικό μας σύστημα.

6.3 Πλοήγηση

Στις προηγούμενες ενότητες μελετήσαμε ξεχωριστά τις δυνατότητες και τα αποτελέσματα της προσομοίωσης των αλγορίθμων σχεδιασμού διαδρομής και του ελεγκτή κίνησης. Σ' αυτή την ενότητα θα παρουσιάσουμε το συνολικό αποτέλεσμα της προσομοίωσης των παραπάνω αλγορίθμων και κατ' επέκταση τη λύση του προβλήματος της πλοήγησης. Ουσιαστικά, αυτό που κάνουμε είναι να επιλύουμε αρχικά με τον αλγόριθμο σχεδιασμού διαδρομής το πρόβλημα της αναζήτησης του βέλτιστου μονοπατιού. Στη συνέχεια ενεργοποιούμε τον αλγόριθμο του ελεγκτή κίνησης για κάθε κόμβο συντεταγμένων που υπολογίστηκε από τον αλγόριθμο σχεδιασμού διαδρομής. Μ' αυτό τον τρόπο καταφέρνουμε να λάβουμε σε πολύ τακτά χρονικά διαστήματα τις τιμές των παραμέτρων που πρέπει να δοθούν στο ρομποτικό μας σύστημα ώστε να κινηθεί με τρόπο τέτοιο ώστε να ακολουθήσει το επιθυμητό μονοπάτι που οδηγεί εν τέλει στο στόχο.

6.3.1 Παράδειγμα Πλοήγησης 1

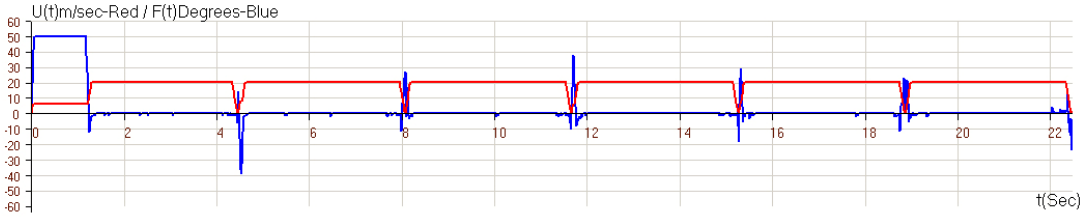
Σ' αυτό το παράδειγμα εξετάζουμε τα βήματα του τελικού μονοπατιού και τον τρόπο υπολογισμού των τιμών των παραμέτρου ελέγχου του ρομποτικού μας συστήματος. Επίσης ελέγχουμε τη συχνότητα με την οποία μεταβάλλονται οι τιμές των παραμέτρων ελέγχου του ρομποτικού συστήματος θέτοντας διαφορετικές τιμές στην παράμετρο delay. Η παράμετρος αυτή ορίζει πόσο κοντά θέλουμε να πλησιάσουμε ένα κόμβο του μονοπατιού πριν ο ελεγκτής κίνησης προχωρήσει στον επόμενο. Αυτό βέβαια ισχύει μόνο για κόμβους που δεν είναι τελευταίοι στη διαδρομή(στόχοι). Στον τελευταίο κόμβο θέλουμε το ρομποτικό μας σύστημα να φτάσει με μηδενική τιμή ταχύτητας. Με την παράμετρο delay επιτυγχάνουμε τον ομαλό έλεγχο της ταχύτητας από κόμβο σε κόμβο.



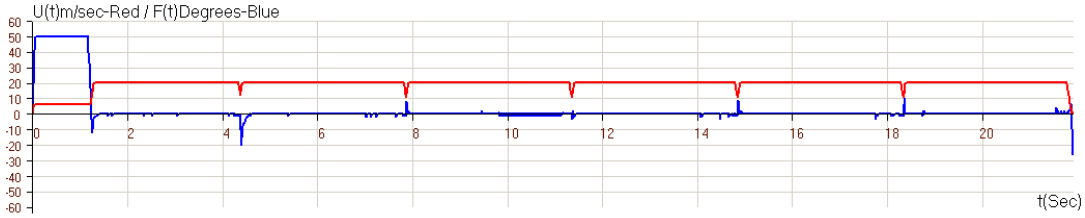
Σχήμα 6.3.1 Left: Path Planning, Astar Pivot =50, Obstacle_Offset=50), Start(0, 0, 0), Destination (300, 300, X) Right: Motion Controller, dt=0,01, L=5.

Στο σχήμα 6.3.1 βλέπουμε την εκτέλεση των αλγόριθμων πλοήγησης. Αρχικά υπολογίζεται με τον A* το μονοπάτι και ο ελεγκτής κίνησης το ακολουθεί

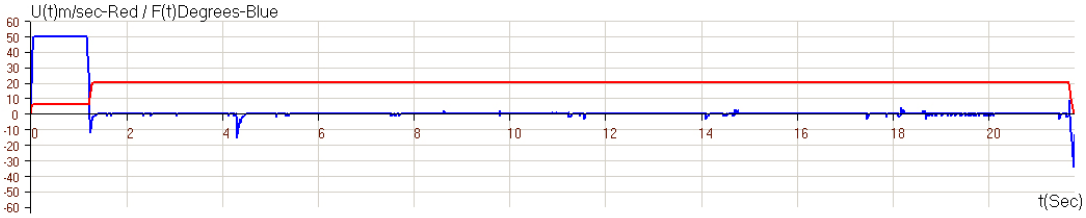
υπολογίζοντας τις τιμές της ταχύτητας και της γωνίας στροφής του ρομποτικού συστήματος. Στα παρακάτω διαγράμματα φαίνονται αναλυτικά οι τιμές αυτές για διαφορετικές τιμές delay.



a. Delay=0



b. Delay=0.5



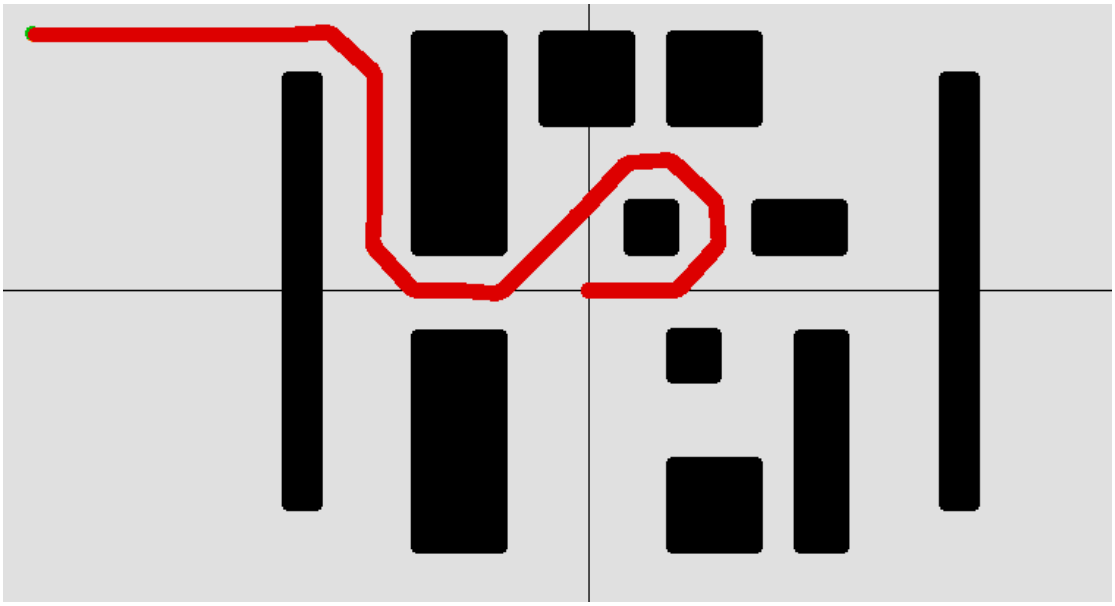
c. Delay=2

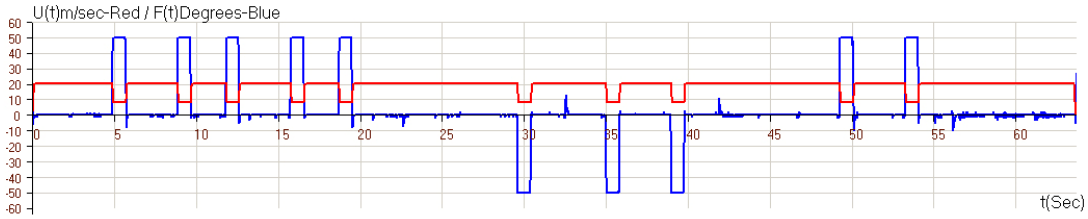
Σχήμα 6.3.2 Motion Controller, για διαφορετικές τιμές delay, Start(0, 0, 0), Destination (300, 300, X).

Από το σχήμα 6.3.2 παρατηρούμε ότι η ταχύτητα του ρομποτικού συστήματος για μηδενική τιμή Delay φθίνει μέχρι να μηδενιστεί εντελώς καθώς πλησιάζει κάθε κόμβο. Για μεγαλύτερες τιμές Delay το όχημα προχωράει στον επόμενο κόμβο νωρίτερα, πριν η ταχύτητα μηδενιστεί. Όσο μεγαλώνει η τιμή του Delay παρατηρούμε ότι η μείωση της ταχύτητας σταματάει να υφίσταται και το ρομποτικό όχημα προσεγγίζει τους κόμβους χωρίς να ελαττώνει ταχύτητα, αν φυσικά δεν υπάρχει κάποιος άλλος λόγος που μπορεί να επιβάλλει να συμβεί αυτό.

6.3.2 Παράδειγμα Πλοήγησης 2

Στο συγκεκριμένο παράδειγμα θα δούμε πως συμπεριφέρονται οι αλγόριθμοι που υλοποιήσαμε για την επίλυση του προβλήματος της πλοήγησης σε ένα τυχαίο περιβάλλον εμποδίων. Τα αποτελέσματα φαίνονται στα παρακάτω σχήματα.



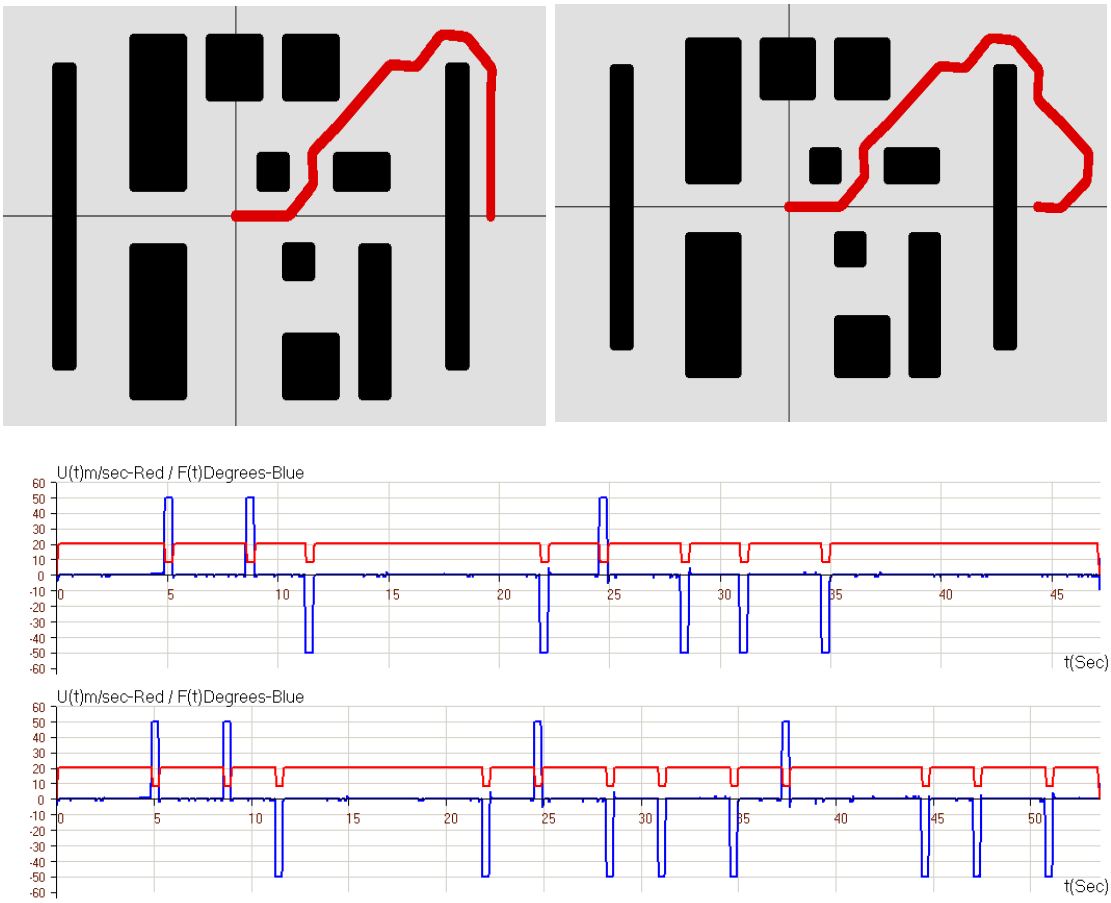


Σχήμα 6.3.3 Astar Pivot =50, Obstacle_Offset=20), Start(0, 0, 0), Destination (-500,300,, X) dt=0,01 L=5.

Είναι φανερό ότι ο ελεγκτής κίνησης ακολουθεί πολύ πιστά το μονοπάτι που ο αλγόριθμος εύρεσης μονοπατιού υπολόγισε (σχήμα 6.3.3). Επίσης οι τιμές της ταχύτητας και της γωνίας στροφής του ρομποτικού συστήματος ακολουθούν ομαλές διακυμάνσεις και οι μέγιστες τιμές δεν παραβιάζονται. Επιπλέον, η τιμή της ταχύτητας όταν η γωνία στροφής υπερβαίνει ένα όριο, πέφτει ομαλά σε μία τιμή πού έχουμε ορίσει για να εξασφαλίσουμε ότι το ρομποτικό όχημα μπορεί να εκτελέσει με ασφάλεια τη στροφή.

6.3.3 Παράδειγμα Πλοήγησης 3

Σ αυτό το παράδειγμα επιλύουμε δυο ίδιες περιπτώσεις πλοήγησης. Στην πρώτη δεν λαμβάνουμε υπόψη τη διεύθυνση τερματισμού του ρομποτικού μας συστήματος, ενώ στην δεύτερη την λαμβάνουμε.

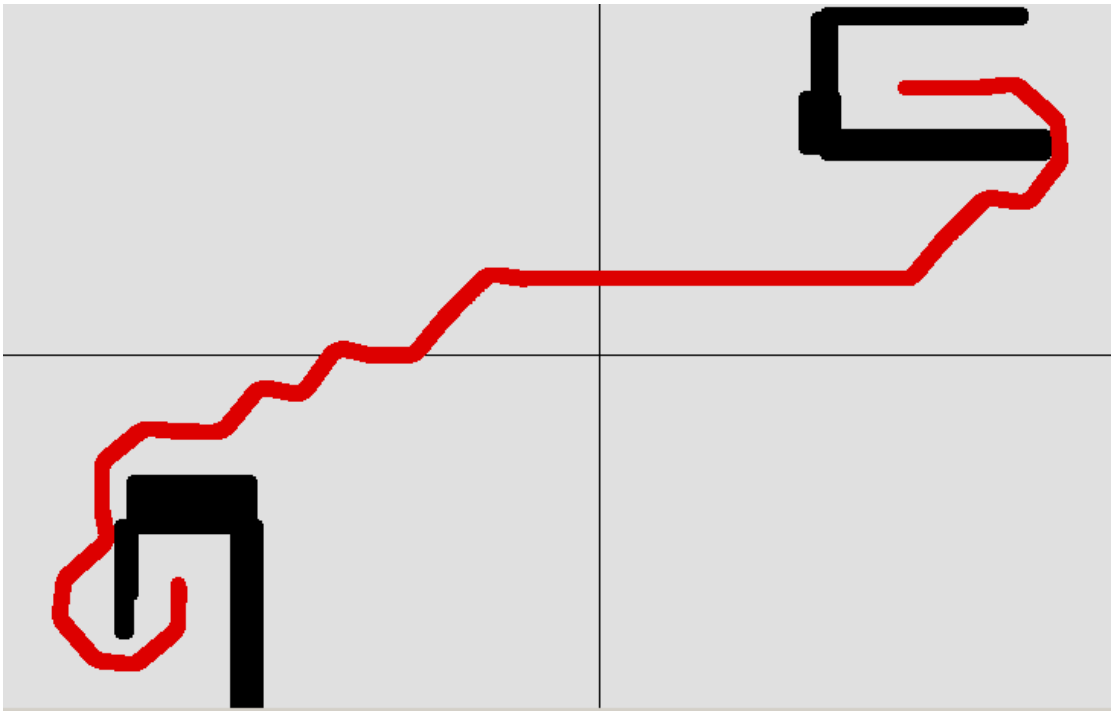


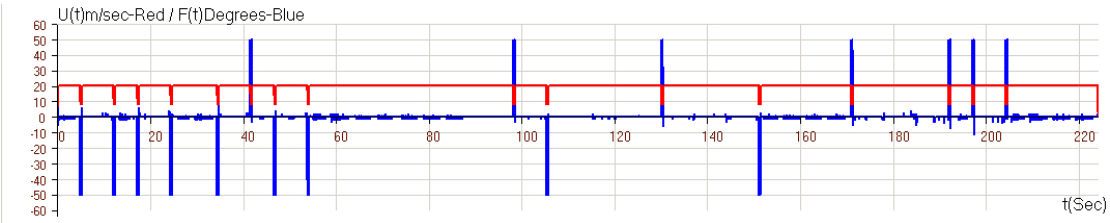
Σχήμα 6.3.4 Astar Pivot =50, Obstacle_Offset=20, dt=0,01, Start(0, 0, 0),
Left & Up-Destination (500, 0, X), Right & Down- Destination (500, 0, 180).

Όπως φαίνεται από το σχήμα 6.3.4 η τεχνική που εφαρμόσαμε υπολογίζει αξιόπιστα το βέλτιστο μονοπάτι το οποίο ακολουθείται πιστά από τον ελεγκτή κίνησης ανεξάρτητα από το ποια θα είναι η διεύθυνση τερματισμού.

6.3.4 Παράδειγμα Πλοήγησης 4

Εδώ έχουμε μια άλλη περίπτωση πλοήγησης που μπορούμε να δούμε τον τρόπο υπολογισμού του τελικού μονοπατιού.



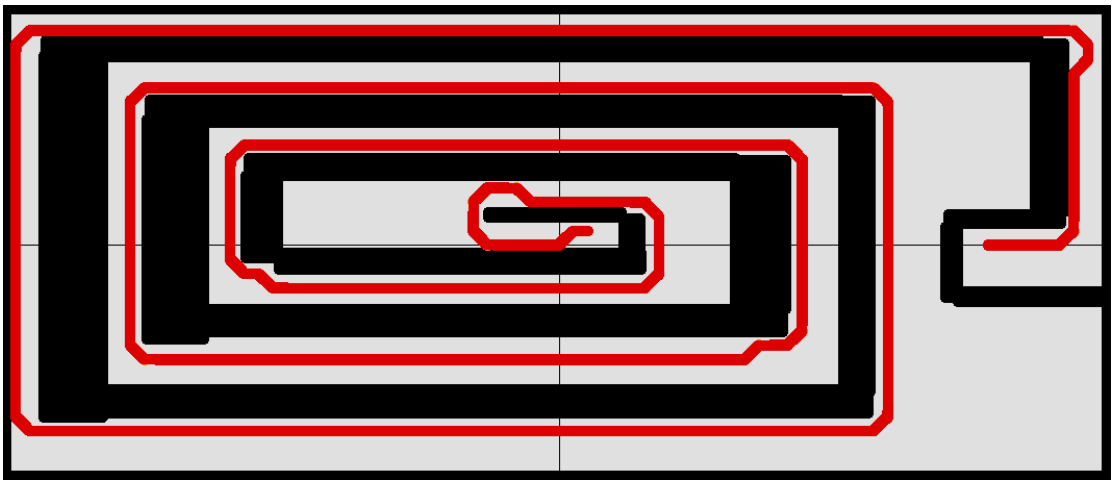


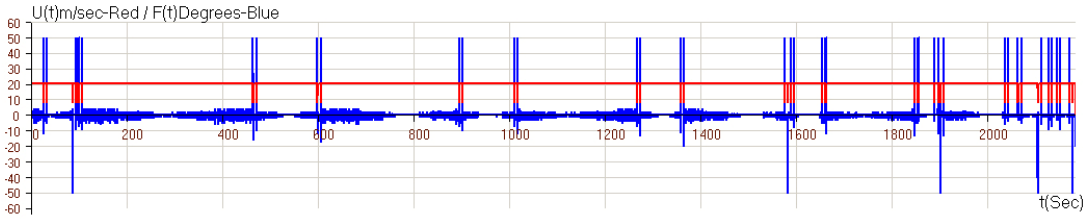
Σχήμα 6.3.5 Astar Pivot =100, Obstacle_Offset=50, dt=0,01, Start(-1100, -600, -90), Destination (800, 700, X).

Όπως φαίνεται από το σχήμα 6.3.5 η τεχνική που εφαρμόσαμε υπολογίζει αξιόπιστα το βέλτιστο μονοπάτι το οποίο ακολουθείται πιστά από τον ελεγκτή κίνησης και οι τιμές των παραμέτρων δεν παραβιάζουν τις μέγιστες τιμές και επιπλέον μεταβάλλονται με ομαλότητα.

6.3.5 Παράδειγμα Πλοήγησης 5

Σ' αυτό το παράδειγμα έχουμε ένα ελικοειδές πεδίο εμποδίων και το ρομποτικό μας σύστημα πρέπει να ακολουθήσει μια διαδρομή τέτοια ώστε να βρεθεί στο κέντρο του χώρου αυτού.



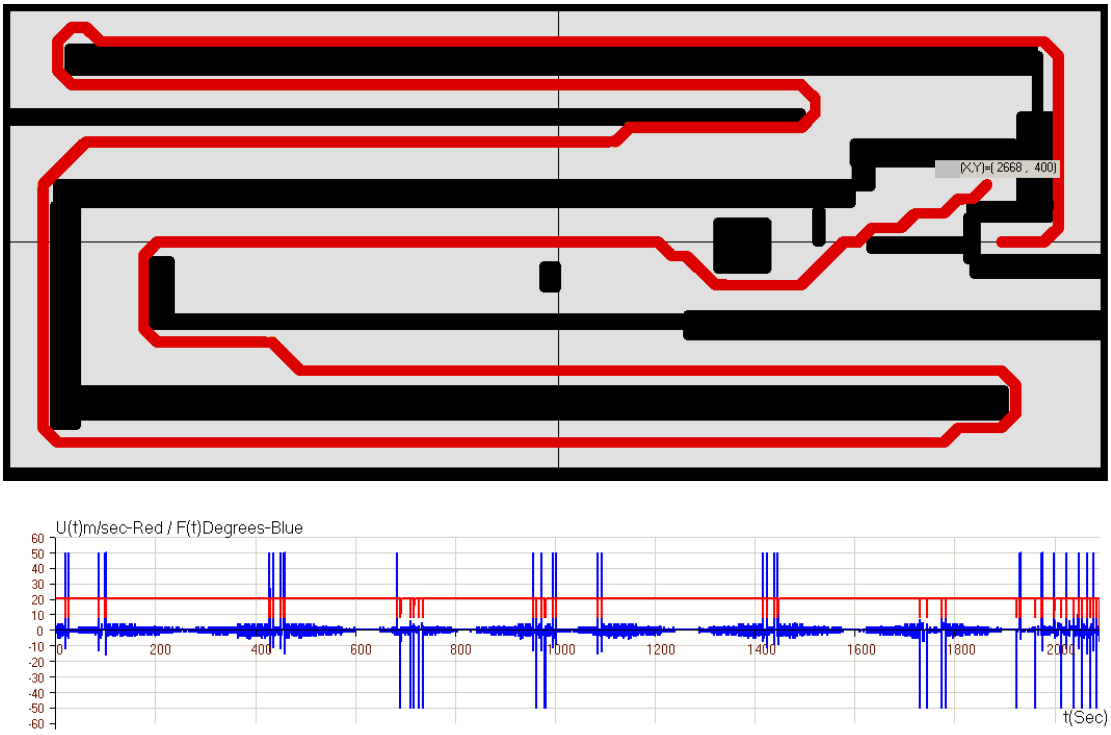


Σχήμα 6.3.6 Astar Pivot =100, Obstacle_Offset=50, dt=0,01 Start(3000, 0, 0), Destination (100, 100, X).

Όπως φαίνεται από το σχήμα 6.3.6, ακόμα και για μια πιο σύνθετη περίπτωση όπως αυτή, η τεχνική που εφαρμόσαμε υπολογίζει αξιόπιστα το βέλτιστο μονοπάτι το οποίο ακολουθείται πιστά από τον ελεγκτή κίνησης και οι τιμές των παραμέτρων δεν παραβιάζουν τις μέγιστες τιμές και επιπλέον μεταβάλλονται με ομαλότητα.

6.3.6 Παράδειγμα Πλοήγησης 6

Στην περίπτωση αυτή το ρομποτικό σύστημα πρόκειται να ακολουθήσει μια διαδρομή μέσα από μια τυχαία διάταξη εμποδίων.



Σχήμα 6.3.7 Astar Pivot=100, Obstacle_Offset=50, dt=0,01 Start(3000, 0, 0), Destination ((3000, 400, X).

Στο σχήμα 6.3.7 βλέπουμε άλλο ένα τυχαίο σενάριο πλοήγησης που σχεδιάζεται σωστά η διαδρομή από τον αλγόριθμο A* και επιπλέον επιβεβαιώνεται η ομαλότητα και η αξιοπιστία που μας παρέχει ο αλγόριθμος του ελεγκτή κίνησης διατηρώντας τις τιμές των παραμέτρων στα επιθυμητά επίπεδα.

Κεφάλαιο 7 Επίλογος

Κλείνοντας, καταγράφουμε κάποια συμπεράσματα από την εκπόνηση της εργασίας, αναφέρουμε κάποιες πιθανές μελλοντικές πτυχές ανάπτυξης και υλοποίησης και περιγράφουμε το μέγεθος και το εύρος των γνώσεων που αποκτήσαμε με την ενασχόληση μας με θέματα όπως αυτό της πλοήγησης ρομποτικών συστημάτων.

7.1 Συμπεράσματα

Στην εργασία αυτή περιγράψαμε μια προσέγγιση επίλυσης του προβλήματος της ρομποτικής πλοήγησης. Θεωρήσαμε ως ρομποτικό μοντέλο ένα μη ολονομικό όχημα τεσσάρων τροχών με δυνατότητα εμπρόσθιας κίνησης στους πίσω τροχούς και γωνία στροφής στους δυο μπροστά. Αναπτύξαμε έναν αλγόριθμο σχεδίασης διαδρομής (Path Planning A*) για την αναζήτηση του βέλτιστου μονοπατιού με δεδομένο το πεδίο κίνησης του ρομποτικού συστήματος. Στη συνέχεια αναπτύξαμε μια τεχνική ελέγχου κίνησης (motion Controller) με την μέθοδο του δυναμικού παραθύρου και με τη χρήση αντικειμενικής συνάρτησης κόστους, λαμβάνοντας βέβαια υπόψη όλους τους περιορισμούς, τους βαθμούς ελευθερίας και τις κινηματικές εξισώσεις του ρομποτικού μας συστήματος. Προσομοιώσαμε τους αλγορίθμους αυτούς μέσω του προγράμματος διασύνδεσης που αναπτύχθηκε. Εξετάσαμε τα αποτελέσματα της προσομοίωσης για τον κάθε αλγόριθμο ξεχωριστά αλλά και για τους δύο μαζί, τα οποία απέδειξαν ότι η συγκεκριμένη μεθοδολογία αντιμετωπίζει με πολύ αξιόπιστο και γρήγορο τρόπο το δύσκολο πρόβλημα της πλοήγησης σε περιβάλλον που είναι εξαρχής γνωστό και μάλιστα για ένα μη ολονομικό ρομποτικό σύστημα τεσσάρων τροχών.

7.2 Περαιτέρω ανάπτυξη

Τα αποτελέσματα της προσομοίωσης μας δίνουν το έναυσμα να δούμε περαιτέρω τη συγκεκριμένη μεθοδολογία και να την εφαρμόσουμε στο πραγματικό ρομποτικό σύστημα.

Παρόλο που το πραγματικό ρομποτικό σύστημα που διαθέτουμε έχει τα χαρακτηριστικά που έχουμε μελετήσει, δεν μας επιτρέπει ακόμα την εφαρμογή των αλγορίθμων σ' αυτό γιατί δεν έχει ολοκληρωθεί η ανάπτυξη του ηλεκτρομηχανικού του εξοπλισμού. Μετά το πέρας της ολοκλήρωσης της ανάπτυξης των υποσυστημάτων αυτών θα μπορούμε να εφαρμόσουμε την τεχνική που υλοποιήσαμε στην παρούσα εργασία.

Ενδιαφέρον επίσης θα είχε η εφαρμογή της συγκεκριμένης μεθοδολογίας και η τροποποίηση των αλγορίθμων για εφαρμογή σε δυναμικά περιβάλλοντα. Αυτό απαιτεί επιπλέον ανάπτυξη του λογισμικού ώστε ανά πάσα χρονική στιγμή ο αλγόριθμος αναζήτησης μονοπατιού να αναθεωρεί τον περιβάλλοντα χώρο για να μπορεί να προβλέπει με ακρίβεια τον τρόπο που θα προσεγγίσει τον στόχο.

Όπως επίσης και η ανάπτυξη αντιστοίχων αλγορίθμων χαρτογράφησης (mapping) και εντοπισμού (localization), οπού το ρομπότ δεν γνωρίζει τίποτα για το περιβάλλον εργασίας του, αλλά διαθέτει όλα τα αισθητήρια όργανα (sonars, laser

scanners) για να το εξερευνήσει και να το αποτυπώσει. Η εφαρμογή όλων των παραπάνω απαιτεί επιπλέον μεθοδολογίες ανάπτυξης και ελέγχου του ρομπότ ανά πάσα χρονική στιγμή, ώστε να γνωρίζουμε την ακριβή του θέση μέσα σε ένα περιβάλλοντα χώρο, αλλά και την αντίληψη-μάθηση για εξερεύνηση αγνώστου.

Η ανάπτυξη των παραπάνω αλγορίθμων με την βοήθεια των αντιστοίχων αισθητηρίων και μεθοδολογιών θα αποτελέσει ένα ολοκληρωμένο πρωτότυπο πλήρως αυτόνομο ρομποτικό όχημα. Η ανάπτυξη ενός τέτοιου ρομποτικού συστήματος απαιτεί σημαντική ερευνητική μελέτη και ανάπτυξη περαιτέρω μεθόδων και τεχνικών πλοήγησης.

7.3 Κατακλείδα

Ο χρόνος εκπόνησης της εργασίας αυτής και η ενασχόλησή μου με ρομποτικά συστήματα αποτέλεσε αστείρευτη πηγή γνώσης και μάθησης, τόσο στο θεωρητικό επίπεδο των τεχνικών της πλοήγησης όσο και στο τεχνικό υπόβαθρο της ανάπτυξης της συγκεκριμένης μελέτης και υλοποίησης. Η ασταμάτητη επαφή με την υλοποίηση βέλτιστων αλγορίθμων, αλλά και ο τρόπος μοντελοποίησης της τεχνικής αυτής με τροφοδότησαν με σημαντική εμπειρία και γνώση τόσο στο υπολογιστικό υπόβαθρο του σχεδιασμού δομών δεδομένων, όσο και στον τρόπο αντιμετώπισης των επιμέρους τεχνικών που απαιτούνται για την επίλυση του συνολικού προβλήματος της πλοήγησης.

Τέλος, η υλοποίηση της συγκεκριμένης εργασίας μου έδωσε την δυνατότητα να μελετήσω ένα γνωστικό αντικείμενο που μέχρι στιγμής μόνο επιφανειακά γνώριζα. Η μελέτη των τεχνικών πλοήγησης ρομποτικών συστημάτων είναι ένα κεφάλαιο που πολλά χρόνια με ενδιέφερε, αλλά ποτέ δεν μου είχε δοθεί η ευκαιρία να ασχοληθώ συστηματικά.

Βιβλιογραφία

[1]. Σ. Κυριαζάκος, *Υλοποίηση συστήματος σκόπευσης εικονολήπτη και αισθητήρα θερμοκρασίας, μέσω ασύρματου δικτύου Η/Υ, για χρήση σε τηλεχειριζόμενο όχημα προστασίας δασών,* ΤΕΙ Κρήτης τμήμα Ηλεκτρονικής, Χανιά 2003.

[2]. J. Canny, *The Complexity of Robot Motion Planning,* Cambridge, massachusetts: MIT Press, 1988.

[3]. J. Schwartz, M. Sharir, and J. Hopcroft, *Planning Geometry and complexity of Robot Motion,* Norwood, New Jersey: Ablex, 1987.

[4]. B. Nagy and A. Kelly, *Trajectory Generation For Car Like Robots Using Cubic Curvature Polunomials*, in Field and Service Robots 2001, Helsinki, Finland, June 11, 2001.

[5]. Y. Guo, Y. Long and W. Sheng, *Global Trajectory Generation for Nonholonomic Robots in Dynamic Environments,* 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007.

[6]. E. Nebot and D. Pagac, *Quadtree Representation and Ultrasonic Information for Mapping an Autonomous Guided Vehicles Environment*, International Journal of Computers and their Applications,Vol. 2, No. 3, pp. 160-170, 1995.

[7]. H. Moravec and M. Martin, *Robot Evidence Grids, C*arnegie Mellon University**,** 1996.

[8]. K. Usher, *Obstacle avoidance for a nonholonomic vehicle using occupancy grids,* Autonomous Systems Laboratory, CSIRO ICT Centre, P.O. Box 883, Kenmore, Qld 4069, Australia

[9]. J. Mirtich and y. Canny*, Using Skeletons for Nonholonomic Path Planning among Obstacles,* Computer Science Division University of California Berkeley, CA 94720.

[10]. I. Trovato, *A* Planning in Discrete Configuration Spaces of Autonomous Systems,* Ph.D. Thesis, Univ. of Amsterdam, Sept. 1996.

[11]. Brooks, A. Rodney, "*A Robust Layered Control System for a Mobile Robot,* IEEE Journal of Robotics and Automation, 2**,** pp. 14-23, 1986.

[12]. R. Arkin, *Behavior-Based Robotics,* Cambridge, Massachusetts: MIT Press, 1998.

[13]. R. Fierro and F. Lewis, *Control of a Nonholonomic Mobile Robot Using Neural Networks*, IEEE Transactions on Neural Networks, Vol 9, No. 4, pp. 589-600, 1998.

[14]. Peng Jinzhu, Yaonan Wang, and Yu Hongshan, *Neural Network-Based Robust Tracking Control for Nonholonomic Mobile Robot,* College of Electrical and Information Engineering, Hunan University, Changsha Hunan, P.R. China, 410082

[15]. Y. Guo and B. Mohanakrishnan, *Complete Coverage Control for Nonholonomic Mobile Robots in Dynamic Environment,* Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006**,** pp. 1704-1709, 2006.

[16]. Min-Soeng Kim, Jin-Ho Shin, Sun-Gi Hong and Ju-Jang Lee, *Designing a robust adaptive dynamic controller for nonholonomic mobile robots under modeling uncertainty and disturbances*, International Journal of Advanced Robotic Systems, Vol. 13, No 5, pp. 507-519, 2008.

[17]. N. Ouadah, L. Ourak and F. Boudjema, *Car-Like Mobile Robot Oriented Positioning by Fuzzy Controllers*, International Journal of Advanced Robotic Systems, Vol. 5, No. 3, pp. 249-256, 2008.
[18]. J. Borenstein, and Y. Koren, *Real-time Obstacle-Avoidance for Fast Mobile Robots*, IEEE Transactions on Systems, Man and Cybernetics, SMC-19, 1989.
[19]. O. Khatib, *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*, International Journal of Robotics Research, Vol 5, No 1, pp. 90-98, 1986.
[20]. D. Carthy, *Animal Navigation,* New York: Charles Scribner's Sons, 1957.
[21]. Gerstner, Wulfram and L. Abbott, *Learning Navigational Maps through Potentiation and Modulation of Hippocampal Place Cells*, Journal of Computational Neuroscience, 1996.
[22]. H. Tanner, S. Loizou and K. Kyriakopoulos, *Nonholonomic Stabilization with Collision Avoidance for Mobile Robots,* Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems Maui, Hawaii, USA, Vol. 3, pp. 1220-1225, 2001.
[23]. G. Laffierriere and H. J. Sussmann, *Motion planning for controllable systems without drift,* In Proceedings IEEE International Conference on Robotics & Automation, 1991.
[24]. J. Laumond, *Singularities and topological aspects in nonholonomic motion planning,* Kluwer Ac. Press, 1992.
[25]. J. Laumond, P. Jacobs, M. Tax and R. Murray, *A motion planner for nonholonomic mobile robot*, IEEE Trans. on Robotics and Automation, Vol. 10, 1994.
[26]. B. Sharma J. and Vanualailai, *Lyapunov Stability of a Nonholonomic Car-like Robotic System Bibhya*, School of Computing, Information & Mathematical Sciences University of the South Pacific, Suva, Fij*i.*
[27]. J. Laumond, *Robot Motion Planning and Control,* Springer, ISBN 3-540-76219-1, 1998.
[28]. R. Murray and S. Sastry*, Steering nonholonomic systems using sinusoids*, IEEE Int. Conf. on Decision and Control, 1990.
[29]. R. Murray, *Robotic Control and Nonholonomic Motion Planning*, PhD Thesis,Memorandum, No. UCB/ERL M90/117, University of California, Berkeley, 1990.
[30]. S. LaValle, *Planning Algorithms,* University of Illinois, Cambridge University Press, Copyright 2006.
[31]. M. Fliess, J. Levine, P. Martin, and P. Rouchon*, Flatness and defect of non-linear systems: Introductory theory and examples*, International Journal of Control, Vol. 61, No. 6, pp. 1327-1361, 1995.
[32]. P. Rouchon, M. Fliess, J. Levine and P. Martin*, Flatness and motion planning: the car with n trailers*, European Control Conference, 1993.
[33]. J. Latombe, *A Fast Path Planner for a Car-Like Indoor Mobile Robot*, Ninth National Conference on Artificial Intelligence, AAAI, 1991.
[34]. J. Laumond, P. Jacobs, M. Tax and R. Murray, *A motion planner for nonholonomic mobile robot*, IEEE Trans. on Robotics and Automation, Vol. 10, 1994.
[35]. M. Lagoudakis, *Incremental Multi-Objective Motion Control of Nonholonomic Mobile Robots,* Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China, pp. 2804-2809, 2006.

[36]. M. Lagoudakis, *Mobile Robot Local Navigation with a Polar Neural map,* Master of Science, University of Louisiana, 1998.
[37]. S. Russell, P. Norvig, *Atificial Intelligence A modern Approach* Second Edition, ISBN 960-209-873-2, 2003.
[38]. D. Fox, W. Burgard, and S. Thrun, *The dynamic window approach to collision avoidance*, IEEE Robotics and Automation Magazine, vol. 4, pp. 23-33, March 1997.
[39]. O. Brock, O. Khatib, *High Speed Navigation Using the Dynamic Window Approach,* Robotic Laboratory, Department of Computer Science Stanford California.
[40]. P. Choudhury and K. Lynch, *Trajectory planning for second-order underactuated mechanical systems in presence of obstacles*, In Proceedings Workshop on Algorithmic Foundations of Robotics, 2002.
[41]. A. Bloch, *Nonholonomic Mechanics and Control*. Springer-Verlag, Berlin, 2003.
[42]. A. Divelbiss and J. Wen, *Nonholonomic path planning with inequality constraints*. In Proceedings IEEE International Conference on Robotics & Automation, pp 52-57, 1994.
[43]. S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*, Springer-Verlag, Berlin, 1999.
[44]. J. Claude, *Models of a wheeled robot named Robudem and design of a state feedback controller for its posture tracking,* Department Of Mechanical Engineering (MSTA), Royal Military Academy, 08 Hobbema Str., 1000, Brussels, Belgium.
[45]. R. Simmons, *The curvature-velocity method for local obstacle avoidance*, in Proceedings of the 1996 IEEE International Conference on Robotics and Automation, G. Lee, Ed. Minneapolis, Minnesota: IEEE Press, 1996.
[46]. Xing-Jian Jing, *Mobile Robots Motion Planning New Challenges*, ISBN 978-953-7619-01-5, 2008.
[47]. A. Pasha, *Path Planning for nonholonomic Vehicles and its Application in Radiation Environment,* Thesis for The Degree of Master of Science, University of Florida, 2003.