



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ
ΔΙΟΙΚΗΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Σχεδιασμός Τροχιάς Εντρόχων Ρομποτικών Οχημάτων
με Χρήση Ευρετικών Αλγορίθμων**

ΙΩΑΝΝΗΣ ΝΤΙΖΟΣ

ΑΜ: 2013010104

Χανιά, Ιούλιος 2021

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή κ. Δοϊτσίδη Ελευθέριο για την πολύτιμη βοήθεια του, την ευκαιρία που μου προσέφερε, για όλες τις γνώσεις που αποκόμισα και την άριστη συνεργασία με σκοπό την εκπόνηση της παρούσας διπλωματικής εργασίας καθώς και την καθοδήγηση του καθ' όλη την διάρκεια της. Επίσης, θα ήθελα να ευχαριστήσω θερμά τον διδάκτορα Αθανάσιο Καπουτση για τη φιλική του στάση, για τις γνώσεις του και την βοήθεια που έλαβα σε επιμέρους ζητήματα της εργασίας. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους αγαπημένους μου ανθρώπους για την στήριξη που παρείχαν και την υπομονή που έδειξαν σε όλες τις δύσκολες στιγμές κατά την πορεία μου ως εδώ.

Αφιερωμένη σε ένα ξεχωριστό πλάσμα.

Σε ευχαριστώ για όλες τις όμορφες στιγμές που μου έδωσες.

I.

ΠΕΡΙΛΗΨΗ ΔΙΠΛΩΜΑΤΙΚΗΣ

Σκοπός της παρούσας εργασίας είναι η μελέτη τριών διαφορετικών προσεγγίσεων για την εύρεση του βέλτιστου μονοπατιού ενός ρομποτικού οχήματος που κινείται σε ένα άγνωστο περιβάλλον. Ο έλεγχος των ρομποτικών οχημάτων γίνεται με δυο τρόπους είτε με τη χρήση ενός απλού λογικού ελεγκτή ή με τη χρήση ενός ελεγκτή βασισμένο σε ασαφή λογική.

Για την εύρεση του μονοπατιού γίνεται χρήση τριών διαφορετικών αλγορίθμων και συγκεκριμένα του A^* , CIA^* και $Weighted A^*$. Η απόδοση τους, συγκρίνεται με χρήση διαφορετικών σεναρίων (διαφορετικό μέγεθος χώρου πειραματισμού και πυκνότητα εμποδίων). Ο πειραματισμός, πραγματοποιήθηκε στο προσομοιωτή ρομποτικών συστημάτων CoppeliaSim, που προσφέρει τη δυνατότητα ρεαλιστικής προσομοίωσης.

Τα αποτελέσματα αναδεικνύουν το βέλτιστο αλγόριθμο ανά περίπτωση και τα σχετικά χαρακτηριστικά του, καθώς και τις ιδιαιτερότητες του τρόπου λειτουργίας του.

SUMMARY

In the context of this work, we focused in the path planning problem for a mobile robot, controlled by a: (i) simple reactive controller and (ii) a fuzzy logic controller. Three different algorithms namely the A*, CIA* and Weighted A* were tested with extensive simulations, using the CoppeliaSim simulator. Each algorithm produced a collision free path per case, in different environments with varying dimensions and obstacle density.

Their performance was evaluated and compared, while the optimal solution and the applicability of each approach was highlighted. .

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	1
ΕΙΣΑΓΩΓΗ	1
1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ	1
ΚΕΦΑΛΑΙΟ 2	3
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	3
2.1 ΕΙΣΑΓΩΓΗ	3
2.2 ΑΣΑΦΗΣ ΛΟΓΙΚΗ	3
2.3 ΑΣΑΦΗ ΣΥΝΟΛΑ	4
2.4 ΑΣΑΦΗΣ ΕΛΕΓΚΤΗΣ	5
2.5 ΚΙΝΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΔΙΑΦΟΡΙΚΗΣ ΚΙΝΗΣΗΣ	7
2.5.1 Ευθεία Κινηματική	9
2.5.2 Αντίστροφη Κινηματική	10
2.6 ΕΛΕΓΧΟΣ ΕΝΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΑΣΑΦΗ ΛΟΓΙΚΗ	11
2.7 ΣΧΕΔΙΑΣΜΟΣ ΒΕΛΤΙΣΤΗΣ ΜΟΝΟΠΑΤΙΟΥ ΜΕ ΧΡΗΣΗ ΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ	14
2.8 ΑΛΓΟΡΙΘΜΟΣ A*	14
2.9 ΑΛΓΟΡΙΘΜΟΣ WEIGHTED A*	20
2.10 ΑΛΓΟΡΙΘΜΟΣ CIA* (Continuously Informed Heuristic A*)	21
ΚΕΦΑΛΑΙΟ 3	24
ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ	24
3.1 ΕΙΣΑΓΩΓΗ	24
3.2 ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ PIONEER 3-DX	24

3.3 ΣΥΝΔΕΣΗ ΠΡΟΣΟΜΟΙΩΤΗ CORPELIA ROBOTICS ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ MATLAB	25
3.4 ΑΝΑΠΤΥΞΗ ΠΡΟΣΟΜΟΙΩΣΗΣ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ CORPELIA ROBOTICS	27
3.5 ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ ΓΙΑ ΧΕΙΡΙΣΜΟ ΤΟΥ CORPELIA ROBOTICS	29
ΚΕΦΑΛΑΙΟ 4	35
ΕΦΑΡΜΟΓΗ / ΥΛΟΠΟΙΗΣΗ	35
4.1 ΕΙΣΑΓΩΓΗ	35
4.2 ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΕΛΕΓΚΤΩΝ ΚΙΝΗΣΗΣ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ	36
4.3 ΣΕΝΑΡΙΑ ΠΟΛΛΑΠΛΩΝ ΒΑΘΜΩΝ ΔΥΣΚΟΛΙΑΣ.....	41
4.4 ΕΥΡΕΣΗ ΒΕΛΤΙΣΤΗΣ ΔΙΑΔΡΟΜΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΕΥΡΕΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ	45
4.4.1 Αποτελέσματα προσομοίωσης με χρήση του αλγορίθμου A*	46
4.4.2 Αποτελέσματα προσομοίωσης με χρήση του αλγορίθμου CIA* (Continuously Informed Heuristic A*)	53
4.4.3 Αποτελέσματα προσομοίωσης με χρήση του αλγορίθμου Weighted A*	60
4.4.4 Σύγκριση δεδομένων για τη περίπτωση χρήσης του απλού ελεγκτή.....	68
4.4.5 Αποτελέσματα προσομοίωσης αλγορίθμου A* με χρήση ασαφούς ελεγκτή	70
4.4.6 Αποτελέσματα προσομοίωσης αλγορίθμου CIA* με χρήση ασαφούς ελεγκτή	83
4.4.7 Αποτελέσματα προσομοίωσης αλγορίθμου Weighted A* με χρήση ασαφούς ελεγκτή	95
4.4.8 Σύγκριση των ελεγκτών κίνησης	108
ΚΕΦΑΛΑΙΟ 5	113
ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	113
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	113

5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	114
ΒΙΒΛΙΟΓΡΑΦΙΑ	115

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ

Ο έλεγχος και η πλοήγηση ρομποτικών οχημάτων τα τελευταία χρόνια φαίνεται να απασχολεί αρκετά τον τομέα της τεχνολογίας με την ανάπτυξη διαφόρων τεχνικών να είναι ραγδαία. Καθοριστικό ρόλο σε αυτή παίζουν οι ανάγκες που παρουσιάζονται, καθώς η τεχνολογία εξελίσσεται και δίνεται η ευκαιρία για χρήση ρομποτικών συστημάτων σε εργασίες, που προσεγγίζουν αυτές των ανθρώπων-χειριστών ή και εργασίες οι οποίες είναι αδύνατον ή επικίνδυνο να πραγματοποιηθούν από έναν άνθρωπο.

Βασική ικανότητα και ο κύριος στόχος ανάπτυξης στον τομέα των κινούμενων ρομποτικών οχημάτων, είναι η πλοήγηση τους. Η ικανότητα αυτή απαιτεί το κινούμενο ρομπότ από μία αρχική θέση στην οποία τοποθετείται, σε ένα άγνωστο, σύνθετο, στατικό ή δυναμικό περιβάλλον, να κινείται και να κατευθύνεται σε ένα τελικό σημείο-στόχο μειώνοντας το κόστος που χρειάζεται για να φτάσει σε αυτό (χρόνος, ενέργεια, κλπ.). Έτσι το ρομπότ πρέπει να είναι σε θέση να επιλύει διάφορα προβλήματα για την επίτευξη αυτού του στόχου. Αρχικά, να είναι ικανό να κινείται γρήγορα, με ασφάλεια και να αποφεύγει στατικά ή/και δυναμικά πιθανά εμπόδια. Επίσης, καθ' όλη την διάρκεια του έργου αυτού πρέπει να συλλέγει πληροφορίες για το περιβάλλον του και να γνωρίζει την θέση στην οποία βρίσκεται.

Εκτός από το κομμάτι της πλοήγησης, σημαντικός στόχος είναι ο σχεδιασμός διαδρομής των αυτόνομων ρομποτικών οχημάτων. Στην σημερινή εποχή η ανάγκη για την ανάπτυξη

αυτών είναι μεγάλη και πολλοί τομείς έχουν δείξει ενδιαφέρον όπως αυτός της αυτοκινητοβιομηχανίες, βιομηχανίες κατασκευής υποθαλάσσιων και εναέριων οχημάτων, στον τομέα του διαστήματος, της άμυνας, ασφάλειας. Ο σχεδιασμός της διαδρομής είναι μία μέθοδος εύρεσης της βέλτιστης διαδρομής ενός αντικειμένου από ένα σημείο σε ένα άλλο. Στα αυτόνομα οχήματα υλοποιούνται κατάλληλοι αλγόριθμοι εύρεσης διαδρομής ώστε να καθοριστούν τα σημεία που διασχίζει στον χώρο το ρομπότ.

Σκοπός της παρούσας εργασίας είναι η μελέτη τριών διαφορετικών προσεγγίσεων για την εύρεση του βέλτιστου μονοπατιού ενός ρομποτικού οχήματος που κινείται σε ένα άγνωστο περιβάλλον. Η εύρεση του βέλτιστου μονοπατιού γίνεται με τη χρήση διαφορετικών ευρετικών αλγορίθμων και η απόδοση τους συγκρίνεται σε διαφορετικά σενάρια. Το ρομποτικό όχημα κινείται αυτόνομα με χρήση είτε ενός απλού λογικού ελεγκτή, είτε με τη χρήση ενός ελεγκτή βασισμένο σε ασαφή λογική. Η υπόλοιπη εργασία είναι δομημένη ως όπως παρουσιάζεται στη συνέχεια.

Στο δεύτερο κεφάλαιο γίνεται βιβλιογραφική αναφορά για τις βασικές αρχές κίνησης εντρόχων οχημάτων, της ασαφούς λογικής και ελεγκτών, των ευρετικών αλγορίθμων και τη διαδικασία εύρεσης διαδρομής με χρήση αυτών.

Στο τρίτο κεφάλαιο παρουσιάζονται οι λεπτομέρειες του ρομποτικού οχήματος που προσομοιώθηκε, το περιβάλλον που κινείται, η τρόπος με τον οποίο γίνονται οι σχετικές προσομοιώσεις και αναλύονται οι βιβλιοθήκες που αναπτύχθηκαν γι' αυτό το σκοπό.

Στο τέταρτο κεφάλαιο παρουσιάζονται αναλυτικά όλα τα πειράματα που πραγματοποιήθηκαν για τους τρεις αλγορίθμους ευρετικής αναζήτησης και για τους δύο τοπικούς ελεγκτές κίνησης ξεχωριστά και γίνεται σύγκριση μεταξύ τους.

Τέλος, στο πέμπτο κεφάλαιο γίνεται μια σύνοψη της εργασίας, παρουσιάζονται τα συμπεράσματα που εξάχθηκαν και παρουσιάζονται κάποιες σκέψεις για μελλοντικές επεκτάσεις.

ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

2.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό παρουσιάζονται οι βασικές αρχές της ασαφούς λογικής στον έλεγχο συστημάτων και ειδικότερα στα έντροχα ρομποτικά οχήματα. Στη συνέχεια παρουσιάζονται αλγόριθμοι ευρετικής αναζήτησης με έμφαση στον αλγόριθμο A* και παρουσιάζονται διαφορετικές εφαρμογές του στο χώρο των έντροχων ρομποτικών οχημάτων.

2.2 ΑΣΑΦΗΣ ΛΟΓΙΚΗ

Η ασαφής λογική έχει τις ρίζες της, στη θεωρία των ασαφών συνόλων (Fuzzy Sets), που διατυπώθηκαν αρχικά από τον καθηγητή Lofti Zadeh το 1965. Συγκεκριμένα, διατυπώθηκε η άποψη ότι στον κόσμο που ζούμε τα αντικείμενα γύρω μας ανήκουν σε διάφορα σύνολα με διαφορετικούς βαθμούς συμμετοχής. Η ασαφής λογική, δίνει μία ικανοποιητική λύση βασιζόμενη στη λεγόμενη αρχή του ασυμβίβαστου, που υποστηρίζει ότι καθώς η πολυπλοκότητα ενός συστήματος αυξάνει, η ικανότητα μας να προβαίνουμε σε ακριβείς και σημαντικές δηλώσεις για την συμπεριφορά του, μειώνεται μέχρι που να φθάσουμε σε ένα όριο (κατώφλι-threshold), πέρα του οποίου ακρίβεια και σημαντικότητα καθίστανται σχεδόν αμοιβαίως αποκλειόμενα χαρακτηριστικά. Με τον όρο ασαφή λογική ορίζουμε ένα υπεрсύνολο της κλασσικής λογικής, όπου έχει επεκταθεί ώστε να μπορεί να χειριστεί τιμές αλήθειας μεταξύ του “απολύτως αληθούς” και του “απολύτως ψευδούς”. Με λίγα λόγια είναι η επέκταση της έννοιας του κλασσικού συνόλου που ορίζεται στο δίτιμο σύνολο $\{0,1\}$, στη γενικευμένη έννοια του

ασαφούς συνόλου που ορίζεται στο απειροδιάστημα $[0,1]$. Η επέκταση αυτής της λογικής με την εισαγωγή ασάφειας, αοριστίας, αβεβαιότητας προσεγγίζει την εκφραστική δύναμη και απλότητα της φυσικής γλώσσας και διατηρεί όσο το δυνατόν περισσότερο την γνωστή μαθηματική δομή της κλασσικής λογικής.

Όπως αναφέρθηκε προηγουμένως η ασαφής λογική είναι η επέκταση του συνόλου $\{0,1\}$ που ορίζεται στο απειροδιάστημα $[0,1]$. Ένα δίτιμο σύνολο A ως προς το σύνολο αναφοράς X μπορεί να παρασταθεί ισοδύναμα μέσω της χαρακτηριστικής συνάρτησης του I_A , δηλαδή:

$$I_A: x \in X \rightarrow I_A(x) = \begin{cases} 1, & \text{όταν } x \in A \\ 0, & \text{όταν } x \notin A \end{cases} \quad (2.1)$$

$$\text{δηλ. } I_A: x \in X \rightarrow I_A(x) \in \{0,1\} \quad (2.2)$$

Έτσι για την αντιμετώπιση τέτοιων γλωσσικών ασαφών εκφράσεων, το ασαφές σύνολο επεκτείνει την έννοια ενός δίτιμου συνόλου μέσω της συνάρτησης συμμετοχής, δηλαδή:

$$\mu_A: x \in X \rightarrow \mu_A(x) \in [0,1] \quad (2.3)$$

Ο αριθμός $\mu_A(x) \in [0,1]$ δηλώνει τον βαθμό συμμετοχής με τον οποίο το στοιχείο $x \in X$ ανήκει (συμμετέχει) στο ασαφές υποσύνολο A του X .

Δηλαδή:

$\mu_A(x) = 1$, σημαίνει ότι το x ανήκει ολοκληρωτικά στο A ,

$\mu_A(x) = 0$, σημαίνει ότι το x δεν ανήκει καθόλου στο A ,

$0 < \mu_A(x) < 1$, σημαίνει ότι το x ανήκει μερικά, δηλαδή κατά κάποιο βαθμό στο A .

Στην ασαφή λογική η τιμή αλήθειας μίας πρότασης δεν παίρνει μόνον μία από τις δύο τιμές 0 ή 1, αλλά μπορεί να παίρνει και οποιαδήποτε τιμή μεταξύ 0 και 1, όποτε μία πρόταση που δεν είναι αληθής δεν σημαίνει αναγκαία ότι είναι ψευδής, αλλά μπορεί να είναι μερικά αληθής και μερικά ψευδής, όπως ένα μισογεμάτο ποτήρι [1].

2.3 ΑΣΑΦΗ ΣΥΝΟΛΑ

Το ασαφές σύνολο είναι ένα από τις βασικότερες έννοιες για την ασαφή λογική. Ο Zadeh πρότεινε ένα διευρυμένο τρόπο αναπαράστασης όπου μία τιμή ανήκει ταυτόχρονα σε πολλά υποσύνολα, στο κάθε ένα με ένα *βαθμό συμμετοχής*. Κάθε τέτοιο υποσύνολο

που περιλαμβάνει στοιχεία που το κάθε ένα έχει ένα βαθμό συμμετοχής είναι το *ασαφές σύνολο* (fuzzy sets). Η περιγραφή μιας αυστηρά αριθμητικής μεταβλητής με λεκτικούς όρους ονομάζεται *ασαφοποίηση* (fuzzyfication) και ο βαθμός συμμετοχής μιας τιμής σε ένα ασαφές σύνολο αποτελεί το *βαθμό βεβαιότητας* (degree of certainty). Το *πλάτος* ενός ασαφούς συνόλου είναι το εύρος του πεδίου ορισμού του και αποτελεί ένα μέτρο *ασάφειας* (fuzziness) του όρου που περιγράφει. Έτσι έχουμε την δυνατότητα να δημιουργήσουμε ασαφείς κανόνες (*rules*), δηλαδή έναν συνδυασμό ασαφών συνόλων που αντιστοιχούν σε λεκτικούς όρους, παρέχοντάς μας έναν μηχανισμό αναπαράστασης της γνώσης παρόμοιο με τον τρόπο λειτουργίας της ανθρώπινης σκέψης. Αποτελείται από δύο βασικά μέρη, το τμήμα της υπόθεσης και το τμήμα της απόδοσης και Ένα απλό παράδειγμα είναι:

IF x is A THEN y is B

Όπου το IF x is A είναι το τμήμα υπόθεσης και το THEN y is B είναι το τμήμα της απόφασης, A και B τα ασαφή σύνολα, x η τιμή μεταβλητής εισόδου, η οποία *ασαφοποιείται* (fuzzyfication), δηλαδή αποκτά ένα βαθμό συμμετοχής στο ασαφές σύνολο A και y η έξοδος του συστήματος που εκφράζει την απόφαση του κανόνα και παρέχεται από τον μηχανισμό του συμπεράσματος (*inference*) σε ασαφή μορφή. Τέλος το συμπέρασμα αποασαφοποιείται από τον μηχανισμό αποασαφοποίησης (*defuzzyfication*) και προκύπτει μια αυστηρή (*crisp*) τιμή που είναι το τελικό αριθμητικό συμπέρασμα που μπορεί να χειριστεί η υπολογιστική μηχανή.

2.4 ΑΣΑΦΗΣ ΕΛΕΓΚΤΗΣ

Οι ασαφείς ελεγκτές είναι ένα σημαντικό εργαλείο που χρησιμοποιούνται ώστε να χειριστούμε στην πράξη καταστάσεις παρόμοιες με αυτές που θα λειτουργούσε η ανθρώπινη λογική.. Η ανάπτυξη και η υλοποίηση ενός ελεγκτή πρέπει να σχεδιαστεί όσο περισσότερο αποτελεσματικά γίνεται, ώστε να πλησιάζει τον καλύτερο άνθρωπο χειριστή μίας διεργασίας που προσπαθούμε να αντιγράψουμε. Για να επιτευχθεί αυτό, δηλαδή να αναπτυχθεί ένας καλά σχεδιασμένος ελεγκτής πρέπει να καταγράψουμε τους κανόνες με τους οποίους θα λειτουργούσε ο χειριστής ώστε ακόμα και κάτω από απρόοπτες συνθήκες να παραμένει το ίδιο λειτουργικό.

Τα βασικά δομικά στοιχεία ενός ασαφούς ελεγκτή (*fuzzy controller*) είναι:

- **Η βάση της γνώσης (knowledge base)** στην οποία είναι αποθηκευμένοι οι κανόνες για τον έλεγχο της διαδικασίας.
- **Τα ασαφή σύνολα (fuzzy sets)** που χρησιμοποιούνται για να αναπαραστήσουν τις μεταβλητές εισόδου και εξόδου με λεκτικούς όρους.
- **Ο ασαφοποιητής (fuzzyfier)** ο οποίος μετατρέπει τις πραγματικές τιμές της εισόδου σε ασαφή σύνολα.
- **Ο μηχανισμός συμπερασμού (inference engine)** όπου επεξεργάζεται τις εξόδους του ασαφοποιητή και με χρήση της βάσης γνώσης εξάγει τα ασαφή σύνολα των συμπερασμάτων.
- **Ο αποασαφοποιητής (defuzzifier)** ο οποίος μετατρέπει τα συμπεράσματα που εξάγει ο μηχανισμός συμπερασμού σε πραγματικούς αριθμούς για να μπορεί να γίνει μετάδοση της δράσης ελέγχου στην διαδικασία.

Για τον σχεδιασμό ενός ασαφούς ελεγκτή πρέπει να γίνουν τα ακόλουθα βήματα:

- **Λεκτική κατανομή των εισόδων:** Αναπαράσταση μεταβλητών εισόδου και εξόδου με λεκτικούς όρους.
- **Διατύπωση των κανόνων:** Τα ασαφή σύνολα μετά την κατανομή εισόδων και εξόδων αποθηκεύονται υπό την μορφή συναρτήσεων συμμετοχής στον υπολογιστή και στη συνέχεια ακολουθεί η διατύπωση των κανόνων.
- **Καθορισμό του τύπου της ασαφούς συνεπαγωγής:** Μετά τη διατύπωση των κανόνων είναι απαραίτητος ο καθορισμός του τύπου της ασαφούς συνεπαγωγής.

Οι πιο γνωστοί τύποι ασαφούς ασαφούς συνεπαγωγής είναι:

α) του Mamdani , χρησιμοποιείται ο τελεστής max-min ο οποίος λαμβάνει το μικρότερο από τους βαθμούς συμμετοχής των ασαφοποιημένων τιμών και παράγει το βαθμό εκπλήρωσης του κάθε κανόνα. Ο βαθμός εκπλήρωσης του κανόνα δηλώνει τη βαρύτητα που έχει το αποτέλεσμα του κανόνα.

β) του Larsen, χρησιμοποιείται ο τελεστής max-product, ο οποίος πολλαπλασιάζοντας τους βαθμούς συμμετοχής των ασαφοποιημένων τιμών υπολογίζει το βαθμό εκπλήρωσης του κανόνα.

- **Αποασαφοποίηση:** Η αποασαφοποίηση παράγει μία αυστηρή ή crisp τιμή από ένα ασαφές σύνολο. Με λίγα λόγια, η αντίθετη διαδικασία από την ασαφοποίηση. Μέθοδοι:

- a) **Κεντρικής τιμής (centroid defuzzycation ή center of area ή COA)** όπου υπολογίζεται το κέντρο βάρους της κατανομής του ασαφούς συνόλου της εξόδου:

$$x_{coa} = \frac{\int x \cdot \mu(x) dx}{\int \mu(x) dx} \quad (2.4)$$

- b) **Αποασαφοποίηση μέσου όρου των μεγίστων (Mean of Maxima ή MOM)**, όπου υπολογίζεται ο μέσος όρος των τιμών εξόδου που έχουν τον μεγαλύτερο βαθμό συμμετοχής:

$$x_{mom} = \frac{1}{m} \sum_{i=1}^m \max \mu(x) \quad (2.5)$$

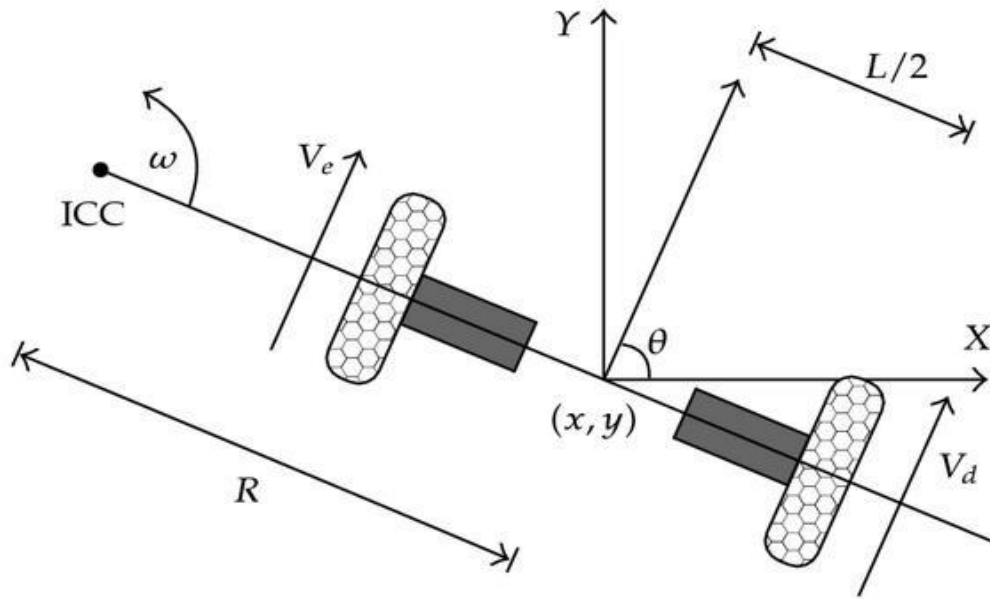
Πλέον η ασαφής λογική εφαρμόζεται ευρέως σε πολλούς τεχνολογικούς και θεωρητικούς κλάδους, όπως τα μαθηματικά, οικονομία, στατιστική, φιλοσοφία, ψυχολογία, βιολογία, στον έλεγχο συστημάτων, στη ρομποτική, στη λήψη αποφάσεων, στα συστήματα παραγωγής. Ακόμα συσκευές που λειτουργούν με ασαφείς ελεγκτές και χρησιμοποιούνται καθημερινά (πλυντήρια, ψυγεία, κλιματιστικά κτλ.) αλλά και ασαφείς ελεγκτές μη επανδρωμένων οχημάτων-αεροσκαφών, ιατρικής, διάγνωσης, χειρουργικής, κτλ.

2.5 ΚΙΝΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΔΙΑΦΟΡΙΚΗΣ ΚΙΝΗΣΗΣ

Στα ρομποτικά οχήματα υπάρχουν διαφορετικοί σχηματισμοί που επιτρέπουν την κίνηση με διαφορετικό τρόπο. Στη συγκεκριμένη παράγραφο θα περιγραφεί η κίνηση ενός οχήματος που χρησιμοποιεί ένα μηχανισμό κίνησης που είναι γνωστός ως διαφορική οδήγηση.

Αποτελείται από δύο κινητήριους τροχούς τοποθετημένους σε έναν κοινό άξονα και κάθε τροχός μπορεί ανεξάρτητα να κινείται είτε προς τα εμπρός είτε προς τα πίσω. Τα οχήματα διαφορικής κίνησης έχουν δύο βαθμούς ελευθερίας και η κίνηση τους γίνεται μόνο στο επίπεδο XY (Σχήμα 2.1). Ενώ μπορούμε να αλλάξουμε την ταχύτητα κάθε

τροχού, το ρομπότ πρέπει να περιστρέφεται γύρω από ένα σημείο που βρίσκεται κατά μήκος του κοινού άξονα αριστερού και δεξιού τροχού. Αυτό το σημείο που περιστρέφεται το ρομπότ ονομάζεται ICC-*Instantaneous Center of Curvatur*.



Σχήμα 2.1. Κινηματικό μοντέλο διαφορικής κίνησης [3]

Όπου:

ICC: Στιγμιαίο Κέντρο Στροφής,

V_e, V_d : ταχύτητα αριστερού και δεξιού τροχού αντίστοιχα,

ω : γωνιακή ταχύτητα,

R : απόσταση του ICC από το μέσο της απόστασης μεταξύ των δύο τροχών,

L : απόσταση μεταξύ των δύο τροχών,

(x, y) : θέση του ρομπότ στο σύστημα αναφοράς,

θ : γωνία που έχει στραφεί το όχημα σε σχέση με το σύστημα αναφοράς.

Γνωρίζουμε ότι η γωνιακή ταχύτητα είναι ίδια και για τους δύο τροχούς και η ταχύτητα περιστροφής είναι ανάλογη του γινομένου της γωνιακής ταχύτητας (**ω**) και της ακτίνας περιστροφής (**r**), σύμφωνα με τη σχέση $u = \omega \cdot r$ μπορούμε να γράψουμε τις εξής εξισώσεις:

$$\omega \cdot \left(R + \frac{L}{2} \right) = V_d \quad (2.6)$$

$$\omega \cdot \left(R - \frac{L}{2} \right) = V_e \quad (2.7)$$

Λύνοντας τις εξισώσεις (2.6) και (2.7) ως προς R και ω έχουμε:

$$R = \frac{L (V_e + V_d)}{2 (V_d - V_e)} \quad (2.8)$$

$$\omega = \frac{V_d + V_e}{L} \quad (2.9)$$

Με βάση τις παραπάνω εξισώσεις είναι δυνατός ο υπολογισμός των κινήσεων που μπορεί να κάνει το όχημα μας. Συγκεκριμένα, εάν:

Εάν $V_d = V_e$, δηλαδή οι ταχύτητες είναι ίσες, τότε το όχημα κινείται προς το εμπρός. Η απόσταση R γίνεται άπειρη, ενώ η γωνιακή ταχύτητα ω είναι μηδενική και συνεπώς δεν υπάρχει περιστροφή του οχήματος.

Εάν $V_e = -V_d$, δηλαδή οι ταχύτητες είναι ίσες με μέτρο αντίθετης φοράς, τότε το όχημα περιστρέφεται γύρω από τον εαυτό του αφού η απόσταση R γίνεται μηδέν.

Εάν $V_d = 0$ ή $V_e = 0$, δηλαδή ένας από τους δύο τροχούς κινείται, τότε το όχημα περιστρέφεται γύρω από τον ακίνητο τροχό (είτε αριστερό είτε δεξιό).

2.5.1 Ευθεία Κινηματική

Ας υποθέσουμε λοιπόν στο σχήμα μας ότι κάποια χρονική στιγμή t το όχημα βρίσκεται στη θέση (x,y) με γωνία που έχει στραφεί θ , σε σχέση με το σύστημα αναφοράς [3].

Από τις σχέσεις (2.8) και (2.9) μπορούμε να βρούμε τη θέση του ICC:

$$ICC = [x - R \cdot \sin \theta, y + R \cdot \cos \theta] \quad (2.10)$$

Τη χρονική στιγμή $t + \delta t$ το όχημα θα βρίσκεται σε ένα σημείο (x', y') και με γωνία θ' , το διάνυσμα $[x', y', \theta']$ θα είναι η καινούρια θέση και κατεύθυνση στην οποία θα βρίσκεται:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \cdot \delta t) & -\sin(\omega \cdot \delta t) & 0 \\ \sin(\omega \cdot \delta t) & \cos(\omega \cdot \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} \cdot \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \cdot \delta t \end{bmatrix} \quad (2.11)$$

2.5.2 Αντίστροφη Κινηματική

Σε αντίθεση με την ευθεία κινηματική που δεδομένης της γωνιακής ταχύτητας υπολογίζαμε τη καινούρια θέση του οχήματος, στην αντίστροφη κινηματική έχουμε ότι δεδομένης της θέσης του οχήματος υπολογίζουμε την ταχύτητα που χρειάστηκε για να φτάσει σε αυτή. Μπορούμε να περιγράψουμε λοιπόν τη θέση που είναι δυνατόν να κινηθεί το όχημα με ταχύτητα $V(t)$ υπό γωνία θ ως εξής:

$$x(t) = \int_0^t V(t) \cos[\theta(t)] dt \quad (2.12)$$

$$y(t) = \int_0^t V(t) \sin[\theta(t)] dt \quad (2.13)$$

$$\theta(t) = \int_0^t \omega(t) dt \quad (2.14)$$

και για την διαφορική κίνηση ενός ρομποτικού οχήματος οι εξισώσεις γίνονται:

$$x(t) = \frac{1}{2} \int_0^t [V_e(t) + V_d(t)] \cos[\theta(t)] dt \quad (2.15)$$

$$y(t) = \frac{1}{2} \int_0^t [V_e(t) + V_d(t)] \sin[\theta(t)] dt \quad (2.16)$$

$$\theta(t) = \frac{1}{L} \int_0^t ([V_d(t) - V_e(t)]) dt \quad (2.17)$$

Ένας από του κυριότερους περιορισμούς της διαφορικής οδήγησης είναι πως το όχημα είναι *μη ολονομικό*, δηλαδή δεν μπορεί να κινηθεί προς όλες τις κατευθύνσεις. Πιο αναλυτικά, έχει τη δυνατότητα να κινηθεί είτε προς τα εμπρός είτε προς τα πίσω ακόμα και υπό γωνία, αλλά δεν μπορεί να κινηθεί κατά τη διεύθυνση του άξονα των τροχών του. Γι' αυτό απαιτείται ένας συνδυασμός κινήσεων ώστε να φτάσει στο στόχο όπως είναι να προχωρήσει ευθεία μέχρι ένα σημείο, μετά να περιστραφεί γύρω από τον εαυτό

του και να προχωρήσει πάλι ευθεία μέχρι το επιθυμητό σημείο [3]. Η θέση του οχήματος για κάθε μία από αυτές τις κινήσεις βρίσκεται από τις παρακάτω εξισώσεις:

$V_d=V_e=V$, κίνηση σε ευθεία γραμμή:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \delta t \\ y + V \sin(\theta) \delta t \\ \theta \end{bmatrix} \quad (2.18)$$

$V_d=-V_e=V$, περιστροφή γύρω από τον εαυτό του:

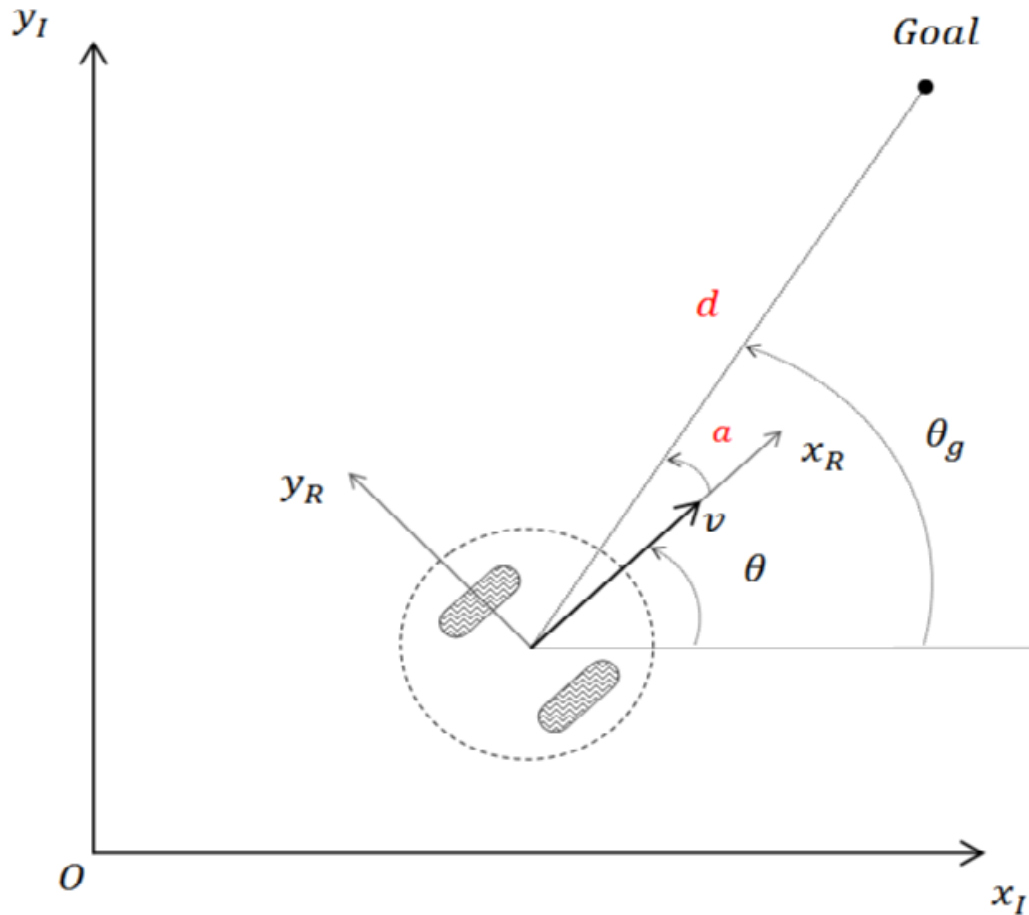
$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + 2V\delta t / L \end{bmatrix} \quad (2.19)$$

2.6 ΕΛΕΓΧΟΣ ΕΝΤΡΟΧΟΥ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ ΜΕ ΑΣΑΦΗ ΛΟΓΙΚΗ

Τα αυτόνομα ρομποτικά οχήματα έχουν ως στόχο να εκτελούν διάφορες διαδικασίες σε ένα γνωστό ή άγνωστο περιβάλλον όπως είναι η εξερεύνηση, η χαρτογράφηση κ.α. Με την βοήθεια αισθητήρων που διαθέτουν μπορούν να λαμβάνουν κατάλληλες πληροφορίες όπως την θέση στην οποία βρίσκονται, την ταχύτητα τους, την κίνηση προς κάποιο στόχο, την εύρεση εμποδίων, ώστε να κινούνται αυτόνομα και να μαθαίνουν πώς να προσαρμόζονται σε ένα άγνωστο περιβάλλον. Όλες αυτές οι διαδικασίες πρέπει να είναι παρόμοιες και οι δυσκολίες να αντιμετωπίζονται παρόμοια με αυτή της ανθρώπινης λογικής χωρίς όμως την παρέμβαση του ανθρώπου. Έτσι η ασαφής λογική είναι ένα χρήσιμο εργαλείο για τον έλεγχο αυτόνομων οχημάτων. Οι ασαφείς ελεγκτές είναι η σύνδεση της ασαφούς λογικής με τα ρομποτικά οχήματα καθώς οι πληροφορίες που λαμβάνονται από τα αισθητήρια αποτελούν τις εισόδους (inputs), ο συνδυασμός δυσκολιών περιγράφονται στους κανόνες (rules) και τέλος η αντιμετώπιση αυτών αποτελούν τις εξόδους (outputs).

Μια μεθοδολογία ανάπτυξης ασαφών ελεγκτών για την πλοήγηση ρομποτικών οχημάτων, έχει προταθεί στις [4-7].

Στην [7] προτείνεται ένας ελεγκτής ασαφούς λογικής τύπου Mamdani σε ένα προσομοιωμένο ρομποτικό όχημα διαφορικής κίνησης, το Pioneer 3-DX, όπου οδηγεί το όχημα από μία δεδομένη αρχική θέση σε έναν τελικό κινούμενο στόχο και παράλληλα αποφεύγει οποιοδήποτε εμπόδιο ανιχνεύει κατά την πορεία του. Η πορεία του προς τον επιθυμητό στόχο επιτυγχάνεται με τον υπολογισμό του σφάλματος κατεύθυνσης, με την λογική αν ο στόχος είναι αριστερά ή δεξιά στρίψε αντίστοιχα προς την κατεύθυνση αυτή και αν ο στόχος είναι ευθεία συνέχισε την πορεία ευθεία.



Σχήμα 2.2 Θέση και προσανατολισμός ρομποτικού οχήματος [7]

Στο Σχήμα 2.2, παρουσιάζεται η θέση και ο προσανατολισμός του ρομπότ (x, y, θ) καθώς και του στόχου (x_g, y_g, θ_g) . Η ευκλείδεια απόσταση μεταξύ της τρέχουσας θέσης και της επιθυμητής θέσης είναι d :

$$d = \sqrt{(x_g - x)^2 + (y_g - y)^2} \quad (2.20)$$

Όπου θ , η γωνία μεταξύ του x_I άξονα και του προσανατολισμού του ρομπότ στη τρέχουσα θέση και θ_g η γωνία μεταξύ του x_I άξονα και του προσανατολισμού του ρομπότ στην τελική θέση:

$$\theta_g = \tan^{-1} \frac{y_g - y}{x_g - x} \quad (2.21)$$

Η γωνία α , ονομάζεται σφάλμα κατεύθυνσης που είναι η γωνία μεταξύ του προσανατολισμού του ρομπότ και μεταξύ της τελικής θέσης και υπολογίζεται ως εξής:

$$\alpha = \begin{cases} \theta_g - \theta, & \text{εάν } \theta_g - 180 < \theta < 180 \\ (\theta_g - \theta) - 360, & \text{εάν } -180 < \theta < \theta_g - 180 \end{cases} \quad (2.21)$$

Εκτός της κατεύθυνσης του οχήματος από μία δεδομένη θέση σε ένα τελικό στόχο, το όχημα πρέπει να έχει τη δυνατότητα να αποφεύγει τα εμπόδια που πιθανόν να βρίσκονται στην πορεία του. Για αυτό τον λόγο πρέπει να συμπεριληφθούν ως είσοδοι και οι τιμές από τα αισθητήρια του αυτόνομου ρομποτικού οχήματος. Έτσι λοιπόν η βασική δομή του προτεινόμενου ασαφούς ελεγκτή αποτελείται από τρία στάδια όπως έχουν αναφερθεί προηγουμένως:

1. Ασαφοποίηση (*fuzzyfication*) που μετατρέπει τις πραγματικές τιμές των εισόδων (γωνία μεταξύ αρχικής θέσης και τελικού στόχου, τιμές αισθητηρίων) σε γλωσσικούς όρους ασαφών συνόλων.
2. Ασαφές συμπέρασμα (*fuzzy inference*) που συνδυάζει τα δεδομένα που λαμβάνονται από την ασαφοποίηση της βάσης της γνώσης και διεξάγει τη λειτουργία της ασαφούς συλλογιστικής.
3. Αποασαφοποίηση (*defuzzyfication*) που μετατρέπει τα υποσύνολα των εξόδων που υπολογίζονται από τον μηχανισμό συμπερασμού.

Η γενική κατασκευή ενός ελεγκτή ασαφούς λογικής με βάση τα παραπάνω ξεκινάει με την κατάλληλη επιλογή συνόλου εισόδων και της συνάρτησης συμμετοχής, στη συνέχεια ορίζουμε το σύνολο των κανόνων και τέλος επιλέγουμε τη μέθοδο αποασαφοποίησης [7]. Η συγκεκριμένη προσέγγιση ακολουθήθηκε και στη συγκεκριμένη εργασία και θα αναλυθεί στη συνέχεια.

2.7 ΣΧΕΔΙΑΣΜΟΣ ΒΕΛΤΙΣΤΗΣ ΜΟΝΟΠΑΤΙΟΥ ΜΕ ΧΡΗΣΗ ΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Ο όρος σχεδιασμός μονοπατιού (*path planning*), στην ρομποτική, αναφέρεται στο πως μπορεί να κινηθεί ένα ρομπότ από ένα σημείο σε ένα άλλο. Με την πρόοδο στη ρομποτική ο σχεδιασμός μονοπατιού περιλαμβάνει πολλές δυσκολίες, όπως την αβεβαιότητα, την ύπαρξη πολλαπλών ρομπότ κ.α. Γενικά στη ρομποτική, ο σχεδιασμός μονοπατιού επικεντρώνεται στον σχεδιασμό αλγορίθμων που δημιουργούν χρήσιμες κινήσεις με την επεξεργασία απλών ή πολύπλοκων γεωμετρικών μοντέλων. Στο συγκεκριμένο πρόβλημα, βρίσκουν λύση οι ευρετικοί αλγόριθμοι αναζήτησης. Ένας από τους ευρύτερα διαδεδομένους αλγορίθμους ευρετικής αναζήτησης, με εφαρμογές στα έντροχα ρομποτικά οχήματα, είναι ο A-star (A^*) [8], που σε συνδυασμό με κάποιες παραλλαγές του όπως ο CIA* (Continuously Informed Heuristic A^*) και ο Weighted A^* , θα αποτελέσουν το αντικείμενο της εργασίας.

2.8 ΑΛΓΟΡΙΘΜΟΣ A^*

Ο αλγόριθμος A^* , έχει χρησιμοποιηθεί για να λυθούν προβλήματα σε διάφορους κλάδους όπως στην ευθυγράμμιση πολλαπλών DNA στη βιολογία, στον σχεδιασμό διαδρομής στη ρομποτική, σε διάφορα προβλήματα τεχνητής νοημοσύνης κ.α.. Στόχος του A^* είναι να βρίσκει τη συντομότερη διαδρομή από έναν αρχικό κόμβο x_{start} σε ένα τελικό x_{goal} .

Χρησιμοποιεί μία συνάρτηση f που οδηγεί στην επιλογή του επόμενου κόμβου που θα επεκταθεί. Η συνάρτηση f είναι μία εκτίμηση της $f^*(x)$ που είναι το κόστος της συντομότερης διαδρομής που περνά μέσω ενός κόμβου x και επιτυγχάνει τον στόχο. Οι δύο αυτές συναρτήσεις υπολογίζονται ως:

$$f(x) = g(x) + h(x) \text{ και } f^*(x) = g^*(x) + h^*(x),$$

όπου $g(x)$ η εκτίμηση της $g^*(x)$ δηλαδή το κόστος της συντομότερης διαδρομής από το x_{start} στο x και $h(x)$ η εκτίμηση του $h^*(x)$ το οποίο είναι το κόστος της συντομότερης διαδρομής από το x στο x_{goal} .

Εάν $h(x) < h^*(x)$, για όλα τα x , τότε η απόσταση είναι αποδεκτή και ο αλγόριθμος μπορεί να αποδειχθεί βέλτιστος. Η εφαρμογή του A^* χρησιμοποιεί δύο λίστες που ονομάζονται OPEN και CLOSED. Η λίστα OPEN αποθηκεύει τους κόμβους που βρίσκονται στα σύνορα της αναζήτησης και η λίστα CLOSED αποθηκεύει τους κόμβους που έχει ήδη προσπελάσει. Σε κάθε επανάληψη ο αλγόριθμος αφαιρεί από την OPEN τον κόμβο με την μικρότερη αξία f , ανοίγει αυτόν τον κόμβο και τον αποθηκεύει στην CLOSED ως διευρυμένο. Οι επαναλήψεις συνεχίζονται έως ότου αφαιρέσει τον x_{goal} από την λίστα OPEN ή δεν υπάρχουν άλλοι διαθέσιμοι κόμβοι στην OPEN, όπου στην πρώτη περίπτωση ο A^* έχει υπολογίσει τη συντομότερη διαδρομή μεταξύ x_{start} και x_{goal} ενώ αν η δεύτερη περίπτωση είναι αληθής τότε δεν υπάρχουν άλλες διαθέσιμες λύσεις [9].

Για την καλύτερη κατανόηση του αλγορίθμου παρουσιάζεται ένα απλό παράδειγμα σε ένα πεπλεγμένο πλαίσιο όπου αναζητείται η βέλτιστη διαδρομή μεταξύ μίας αρχικής θέσης S σε έναν τελικό στόχο G με την χρήση του αλγορίθμου A^* . Στο Σχήμα 2.3, παρουσιάζεται το πλέγμα και με μαύρο είναι το εμπόδιο που βρίσκεται στο χώρο. Αρχικά εισάγουμε στη λίστα CLOSED τον κόμβο S που έχει ήδη εξεταστεί και συνεχίζουμε.

				G
	f=6 g=1 h=5			
f=8 g=1 h=7	S	f=6 g=1 h=5		
	f=8 g=1 h=7			

Σχήμα 2.3 Αρχικό Πλέγμα

Έστω ότι το κόστος να μεταβεί από τον έναν κόμβο στον επόμενο είναι 1 και ξεκινώντας από τον κόμβο S ως αρχική θέση, υπολογίζεται η συνάρτηση $f(x) = g(x) + h(x)$ όπου x ο κάθε γειτονικός κόμβος. Όπως φαίνεται (Σχήμα 2.3) τη μικρότερη τιμή έχουν ο κόμβος (2,3) και ο (3,2) με τιμή ίση με 6.

Στο επόμενο βήμα επιλέγουμε τον κόμβο με τη μικρότερη τιμή f το τοποθετούμε στην CLOSED και βρίσκουμε τους γειτονικούς κόμβους τους οποίους και εισάγουμε στην OPEN λίστα. Σε αυτή την περίπτωση έχουμε δύο κόμβους με f=6 και η απόσταση αυτών των κόμβων μέχρι τον τελικό είναι ίδια h=5. Θα πρέπει λοιπόν να εξετάσουμε και τις δύο περιπτώσεις οι οποίες εισάγονται στη CLOSED. Ξεκινάμε από το δεξί κελί με κόστος ίσο με 6. Λόγω ότι ο αριστερά κόμβος 'S' είναι ήδη στην CLOSED και το δεξί κελί είναι εμπόδιο, οι γείτονες που παίρνουμε είναι ο (3,3) και (1,3). Επιλέγεται ο (3,3) με f=6, αφαιρείται από την OPEN και εισάγεται στην CLOSED (Σχήμα 2.4).

				G
	f=6 g=1 h=5	f=6 g=2 h=4		
f=8 g=1 h=7	S	f=6 g=1 h=5		
	f=8 g=1 h=7	f=8 g=2 h=6		

Σχήμα 2.4 Βήμα 2 του A*

Σε αυτό το σημείο ο κόμβος (3,3) δεν παράγει γείτονες καθώς δεξιά του και πάνω του έχει εμπόδια και τα άλλα δύο βρίσκονται στην CLOSED. Γι' αυτό ψάχνουμε τον επόμενο

κόμβο με το μικρότερο κόστος και αυτό είναι το (3,2) που είχαμε από την αρχή (Σχήμα 2.5). Συνεχίζουμε τις επαναλήψεις ομοίως, και επιλέγουμε τον κόμβο (4,2) με κόστος 6, τον αφαιρούμε από την OPEN και τον εισάγουμε στην CLOSED.

Τέλος επιτυγχάνουμε την εύρεση του βέλτιστου μονοπατιού με τη χρήση του αλγορίθμου A* (Σχήμα 2.5-9) και αφού έχουμε κρατήσει τον δείκτη κάθε κόμβου που επιλέξαμε μέσω αυτής ως καλύτερο έχουμε την ακόλουθη βέλτιστη διαδρομή:

$(2,2) \rightarrow (3,2) \rightarrow (4,2) \rightarrow (5,2) \rightarrow (5,3) \rightarrow (5,4) \rightarrow (5,5)$

				G
	f=6 g=1 h=5	f=6 g=2 h=4		
f=8 g=1 h=7	S	f=6 g=1 h=5		
	f=8 g=1 h=7	f=8 g=2 h=6		

Σχήμα 2.5 Βήμα 3 του A*

				G
	f=6 g=2 h=4			
f=8 g=2 h=6	f=6 g=1 h=5			
f=8 g=1 h=7	S			
	f=8 g=1 h=7			

Σχήμα 2.5 Βήμα 4 του A^*

f=8 g=4 h=4	f=6 g=3 h=3	f=6 g=4 h=2		G
f=8 g=3 h=5	f=6 g=2 h=4			
f=8 g=2 h=6	f=6 g=1 h=5			
f=8 g=1 h=7	S			
	f=8 g=1 h=7			

Σχήμα 2.6 Βήμα 5 του A^*

f=8 g=4 h=4	f=6 g=3 h=3	f=6 g=4 h=2	f=6 g=5 h=1	G
f=8 g=3 h=5	f=6 g=2 h=4			
f=8 g=2 h=6	f=6 g=1 h=5	f=6 g=2 h=4		
f=8 g=1 h=7	S	f=6 g=1 h=5		
	f=8 g=1 h=7			

Σχήμα 2.7 Βήμα 6 του A^*

f=8 g=4 h=4	f=6 g=3 h=3	f=6 g=4 h=2	f=6 g=5 h=1	G
f=8 g=3 h=5	f=6 g=2 h=4			
f=8 g=2 h=6	f=6 g=1 h=5			
f=8 g=1 h=7	S			
	f=8 g=1 h=7			

Σχήμα 2.8 Βήμα 7 του A^*

f=8 g=4 h=4	f=6 g=3 h=3	f=6 g=4 h=2	f=6 g=5 h=1	G
f=8 g=3 h=5	f=6 g=2 h=4			
f=8 g=2 h=6	f=6 g=1 h=5			
f=8 g=1 h=7	S			
	f=8 g=1 h=7			

Σχήμα 2.9 Βήμα 8 του A^*

2.9 ΑΛΓΟΡΙΘΜΟΣ WEIGHTED A^*

Ο αλγόριθμος weighted A^* είναι μία παραλλαγή του κλασσικού A^* με απόδοση βάρους στην ευρετική αξιολόγηση πολλαπλασιάζοντας την με έναν όρο w [10]. Σε πολύπλοκα προβλήματα αναζήτησης ο A^* είναι πιθανό να χρειάζεται αρκετή ώρα ώστε να βρει την βέλτιστη λύση, γι αυτό το λόγο ήταν αναγκαία μία εναλλακτική λύση η οποία θα ήταν ικανή να δώσει αποτελέσματα σε μικρότερο χρόνο. Ο αλγόριθμος weighted A^* είναι μία παραλλαγή του κλασσικού A^* ο οποίος δίνει λύση σε αυτό το πρόβλημα με απόδοση βάρους στην ευρετική αξιολόγηση πολλαπλασιάζοντας της με έναν όρο w .

Η συνάρτηση ορίζεται ως $f(x) = g(x) + w \cdot h(x)$ με την παράμετρο $w \geq 1$ που ορίζεται από τον χρήστη (A^* αλγόριθμος για τιμή ίση με 1). Εάν το $w > 1$ η αναζήτηση δεν είναι αποδεκτή και η πρώτη λύση που μπορεί να βρεθεί δεν είναι η βέλτιστη, παρόλο που βρίσκεται αρκετά πιο γρήγορα. Εάν η $h(x)$ είναι αποδεκτή, η βέλτιστη λύση που βρέθηκε με τον Weighted A^* περιορίζεται, δηλαδή το κόστος της λύσης δεν μπορεί να είναι μεγαλύτερο από το βέλτιστο κόστος περισσότερο από έναν παράγοντα w . Μία

τέτοια λύση λέγεται ότι είναι ϵ -αποδεκτή όπου $\epsilon = w-1$. Ένας σταθμισμένος ευρετικός αλγόριθμος επιταχύνει την αναζήτηση, διότι κάνει τους κόμβους που είναι πιο κοντά στο στόχο να φαίνονται ως καλύτερες λύσεις, δηλαδή δίνει προτεραιότητα στα κελιά αυτά, και προσαρμόζει την προσπάθεια της αναζήτησης με την ποιότητας της λύσης. Ο σταθμισμένος A^* είναι ο αλγόριθμος με την καλύτερη απόδοση για γενική χρήση με περιορισμένη όμως βέλτιστη αναζήτηση [10].

2.10 ΑΛΓΟΡΙΘΜΟΣ CIA^* (CONTINUOUSLY INFORMED HEURISTIC A^*)

Ο αλγόριθμος CIA^* βασίζεται στον A^* και εγγυάται την εύρεση της βέλτιστης απόδοσης της διαδρομής. Ωστόσο ο CIA^* αποκλίνει από τον πρωτότυπο A^* στον υπολογισμό της εκτίμησης της απόστασης $h^*(x)$ προς το στόχο. Αντί να χρησιμοποιεί μία "υποτιμημένη" σταθερή ευρετική συνάρτηση όπως ο A^* , αναθεωρεί ουσιαστικά τις ευρετικές αξιολογήσεις βάση των επιβαλλόμενων που προκύπτουν από τη διαμόρφωση των εμποδίων [11].

Η βασική ιδέα του CIA^* βασίζεται στην ασφαλή αναθεώρηση της ευρετικής απόστασης, χρησιμοποιώντας τη γνώση της θέσης των εμποδίων η οποία μεταφέρεται σε όλους τους "διαδόχους/γείτονες" και αξιοποιείται για να ενημερώσει την απόσταση τους από τον στόχο. Η διαδικασία αναθεώρησης αποτελείται δύο βασικούς πυλώνες.

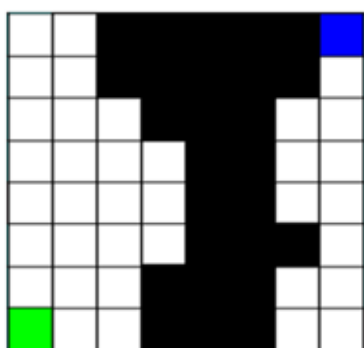
Ο πρώτος πυλώνας βασίζεται στον υπολογισμό της ευρετικής. Για να επιτευχθεί ο υπολογισμός αυτός, σε σχέση με τις αντίστοιχες δομές της A^* , το σύνολο των εμποδίων που έχουν ανακαλυφθεί διατηρείται. Το σύνολο αυτό μαζί με το CLOSED, τη λίστα των διευρυμένων κελιών, χρησιμοποιείται για τον ορισμό του BLOCKED συνόλου. Ο αλγόριθμος εξετάζει αν ο σχηματισμός του BLOCKED περιορίζει περαιτέρω την εύρεση της ελάχιστης διαδρομής που μπορεί να βρεθεί μεταξύ ενός κελιού από το OPEN και του στόχου. Για να γίνει αυτό, χρησιμοποιείται μία επαναληπτική διαδικασία στην οποία υπολογίζεται το ορθογώνιο ελάχιστου μεγέθους, στο οποίο συνδέονται το αντίστοιχο κελί από το OPEN και το κελί του στόχου. Η διαδικασία της ευρετικής ενημέρωσης, σημειώνει όλα τα κελιά που ανήκουν στο BLOCKED ως κατειλημμένα και σταδιακά αυξάνει το ορθογώνιο που ορίζεται από το κελί του στόχου και το κελί της θέσης που βρίσκεται. Όταν τα δύο αυτά κελιά θεωρηθούν συνδεδεμένα η διαδικασία τερματίζει.

Επομένως η ευριστική τιμή αυτού του κελιού μπορεί να αυξηθεί με έναν *συντελεστή διόρθωσης* που ορίζεται ως δύο φορές το μέγεθος αυτής της μετατόπισης. Ο συντελεστής αυτός, αντιπροσωπεύει την αξία που η ευριστική τιμή μπορεί να αυξηθεί περαιτέρω από την σταθερή εκτίμηση της A^* (Manhattan distance) χωρίς να υπονομευθεί η ιδιότητα του.

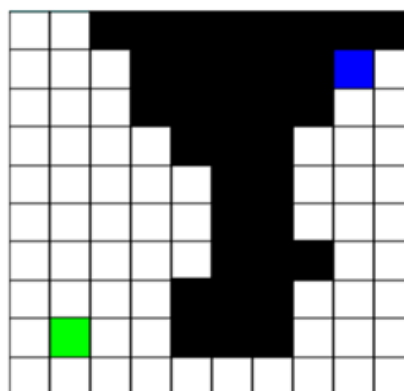
Ο δεύτερος πυλώνας έχει στόχο να αντιμετωπίζει προβλήματα σχετικά με την αξιολόγηση των κελιών, τα οποία είναι πιθανόν να προκληθούν από την χρήση της συνεχώς ενημερωμένης ευριστικής συνάρτησης. Στην ουσία κάθε σύγκριση μεταξύ των κελιών που έχουν αξιολογηθεί με διαφορετικές ευριστικές συναρτήσεις θα ήταν άδικη και θα μπορούσε να διακινδυνεύσει την δυνατότητα παραδοχής. Για αυτό τον λόγο η CIA* επανεξετάζει την ευριστική τιμή του ελάχιστου εξαγόμενου κελιού ώστε να είναι ενημερωμένη με τη τρέχουσα γνώση των θέσεων των εμποδίων. Το ενημερωμένο κελί επανατοποθετείται στο OPEN και το επόμενο με την ελάχιστη τιμή εξάγεται. Η διαδικασία αυτή επαναλαμβάνεται μέχρι η ευριστική αξιολόγηση του τρέχοντος εξαγόμενου κελιού να έχει ήδη αλλάξει ή να μην μπορεί να επηρεαστεί από τις πληροφορίες των νέων κατειλημμένων κελιών.



(α)



(β)



(γ)

Σχήμα 2.10 Ο αλγόριθμος CIA*: Ανανέωση ευριστικής τιμής (α) Έδαφος με τις μέχρι τώρα γνωστές πληροφορίες (β) Αναζήτηση με *offset* 0 (δεν υπάρχει διαδρομή μεταξύ των δύο κελιών) (γ) Το *offset* αυξάνεται κατά 1 κελί, τα δύο κελιά θεωρούνται συνδεδεμένα [11].

ΚΕΦΑΛΑΙΟ 3

ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ

3.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό παρουσιάζεται το περιβάλλον προσομοίωσης που χρησιμοποιήθηκε. Περιγράφεται ο τρόπος λειτουργίας του CorreliaSim και ο τρόπος διασύνδεσης του με το περιβάλλον Matlab. Παρουσιάζεται το ρομποτικό όχημα Pioneer 3-dx, που χρησιμοποιήθηκε για πειραματισμό και δόθηκε έμφαση στις εντολές που χρησιμοποιήθηκαν για την δημιουργία αρένων, εμποδίων και της κίνησης του ρομπότ από μία αρχική θέση σε ένα τελικό σημείο-στόχο.

3.2 ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ PIONEER P3-DX

Για τον πειραματισμό και τη μελέτη των αλγορίθμων, χρησιμοποιήθηκε ένα προσομοιωμένο έντροχο ρομποτικό όχημα το Pioneer 3-dx (Σχήμα 3.1), η κίνηση του οποίου βασίζεται στις αρχές της διαφορικής κίνησης. Αποτελείται από δύο σταθερούς τροχούς που ελέγχονται ανεξάρτητα από τους κινητήρες του. Διαθέτει αρκετούς αισθητήρες όπως υπέρυθρων, υπερήχων, laser scanner και κάμερα για τον έλεγχο του περιβάλλοντός του για εμπόδια. Τα λειτουργικά χαρακτηριστικά του ρομποτικού οχήματος παρουσιάζονται στο πίνακα 3.1.



Σχήμα 3.1 Ρομποτικό όχημα Pioneer p3-dx

Operation	Robot Weight: 8kg, Operating Payload: 17kg
Differential Drive Movement	Turn Radius: 0cm, Swing Radius: 26.7cm Max.Forward/Backward Speed: 1.2m/s, Rotation Speed: 300°/s, Max Traversable Step: 2.5cm, Max Traversable Gap: 5cm, Max Traversable Grade: 25%, Traversable Terrain: Indoor, wheelchair accessible
Power	Run time: 8-10 hours w/3 batteries(with no accessories) Charge Time: 12 hours(standard) or 2.4 hours(optional high capacity charger) Available Power Supplies: 5V 1.5A switched 5V 2.5A switched
Microcontroller I/O	System Serial, 32 digital inputs, 8 digital outputs, 7 analog outputs, 3 serial expansion ports
User Control Panel	MIDI programmable piezo buzzer, Main power indicator, Battery charge indicator, 2 AUX power switches, System reset, Motor enable pushbutton

Πίνακας 3.1 Λειτουργικά Χαρακτηριστικά του ρομποτικού οχήματος

3.3 ΣΥΝΔΕΣΗ ΠΡΟΣΟΜΟΙΩΤΗ CORPELIA ROBOTICS ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ MATLAB

Το λογισμικό CoppeliaSim είναι ένας ισχυρός 3D προσομοιωτής που χρησιμοποιείται στη βιομηχανία, στην εκπαίδευση και στην έρευνα που δίνει τη δυνατότητα για μοντελοποίηση, επεξεργασία, προγραμματισμό και προσομοίωση ρομποτικών

συστημάτων. Έχει αναπτυχθεί, γύρω από μία πολύπλευρη αρχιτεκτονική που χρησιμοποιεί μία κινηματική μηχανή για υπολογισμούς ευθείας και αντίστροφης κινηματικής. Σημαντικό εργαλείο είναι η παροχή plug-ins που περιλαμβάνουν σχεδιασμό διαδρομής, δημιουργία οπτικού περιβάλλοντος και επεξεργασία αυτού, υπολογισμό ελάχιστης απόστασης, ανίχνευση σύγκρουσης κ.α.

Ακόμα διαθέτει μηχανισμό που δίνει πρόσβαση στη βιβλιοθήκη Coppelia API σε πολλές γλώσσες προγραμματισμού όπως Python, Java, Matlab, C/C++. Στην εργασία αυτή επιλέχθηκε η σύνδεση αυτή με το λογισμικό της Matlab, στην οποία προγραμματίζεται και υλοποιούνται όλοι οι μαθηματικοί υπολογισμοί, συναρτήσεις και πολύπλοκα scripts απαραίτητα για την κίνηση του οχήματος. Για να γίνει αυτή η σύνδεση των δύο προγραμμάτων, αρχικά πρέπει να γίνει η φόρτωση της βιβλιοθήκης remote API, η αρχικοποίηση των συναρτήσεων για την σύνδεση Coppelia / Matlab (initialiseRoboticsToolbox) και τέλος να κλείσουν όλες οι ανοιχτές συνδέσεις και μέσω της κατάλληλης θύρας στέλνεται αίτημα για την εκκίνηση της προσομοίωσης. Η διαδικασία παρουσιάζεται στο Σχήμα 3.1

```

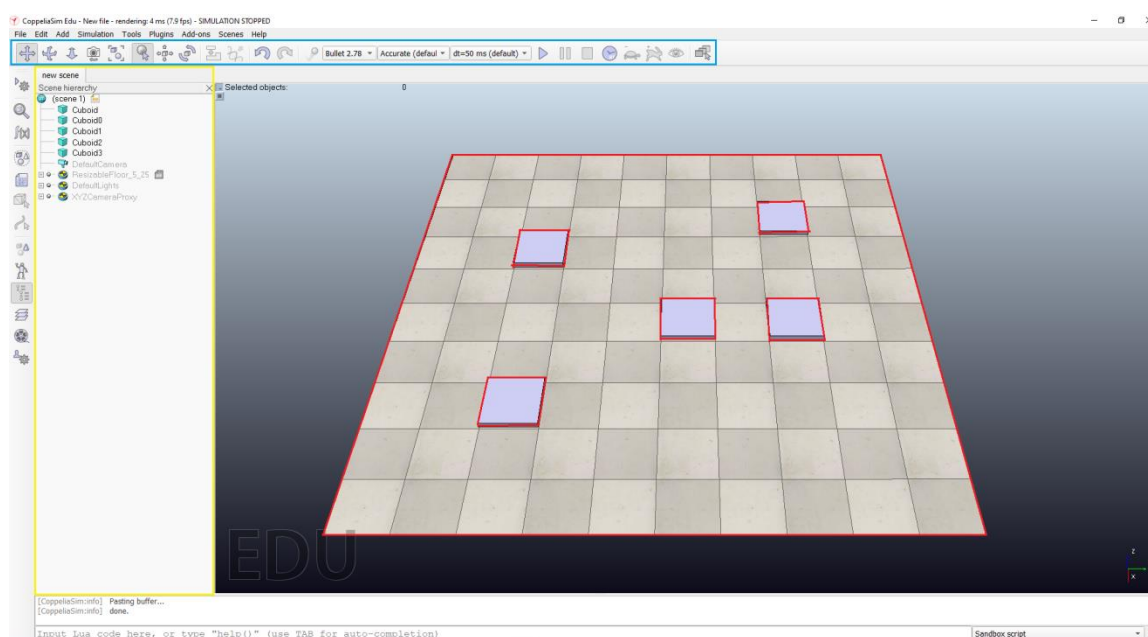
1      %% Connection Matlab/Coppelia
2
3      disp('Program started')
4      sim=remApi('remoteApi');%build the object and load the library
5
6      %initialiseRoboticsToolbox
7      %We use the loopback
8      %IP address / connection Port / wait Until Connected /
9      %do Not Reconnect Once Disconnected / time Out In (ms)
10     sim.simxFinish(-1); % close all opened connections
11     clientID=sim.simxStart('127.0.0.1',19999,true,true,2000,5);
12
13
14     if clientID == -1
15         error('the connection to the server was not possible');
16     else
17         disp('Connected to remote API server');
18
19         %Requests a start of a simulation (clientID / operationMode)
20         [errorCode]=sim.simxStartSimulation(clientID, sim.simx_opmode_oneshot_wait);
21
22         if errorCode ~=0
23             error('Vrep error of requests a start of a simulation');
24         else
25             disp('OK request simulation');
26         end
27     end

```

Σχήμα 3.1 Διαδικασία σύνδεσης Matlab - CoppeliaSim

3.4 ΑΝΑΠΤΥΞΗ ΠΡΟΣΟΜΟΙΩΣΗΣ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ CORPELIA ROBOTICS

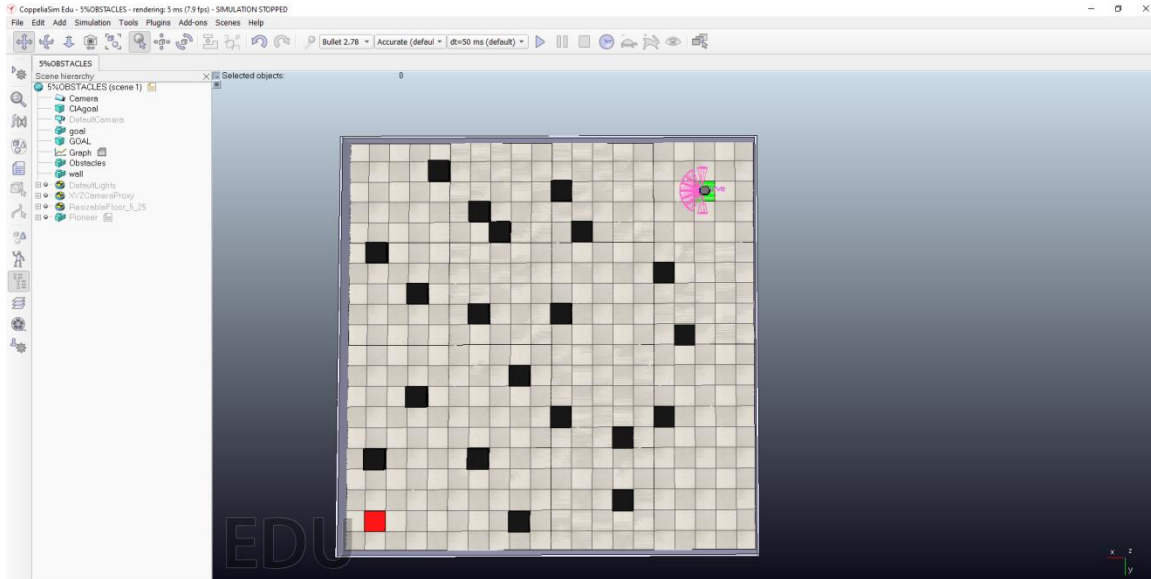
Στο περιβάλλον του προσομοιωτή CoppeliaSim, υπάρχει η δυνατότητα δημιουργίας χώρων πειραματισμού που ονομάζονται αρένες. Σε αυτές υπάρχει η δυνατότητα να εισαχθούν αντικείμενα, στατικά ή και δυναμικά, κινούμενα ρομπότ, δίτροχα, τετράτροχα, ευκίνητα ή δυσκίνητα αλλά και μη κινούμενα ρομπότ με αρθρώσεις, αισθητήρες υπέρυθρων, αισθητήρες υπερήχων διάφορων τύπων και κάμερες για αναγνώριση αντικειμένων. Μια ενδεικτική αρένα παρουσιάζεται στο Σχήμα 3.2.



Σχήμα 3.2 Ενδεικτική αρένα πειραματισμού

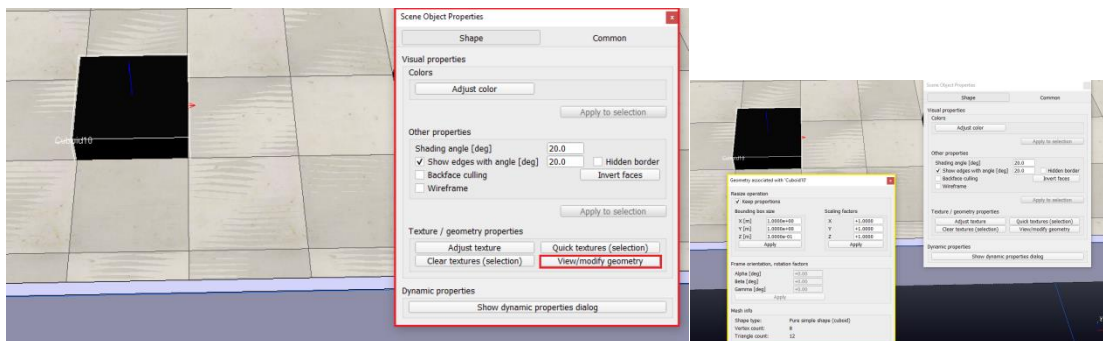
Ακόμα μπορούμε να τοποθετήσουμε αντικείμενα διάφορων σχημάτων (δίσκους, κύβους, σφαίρες, κυλίνδρους) σε όποιο σημείο επιθυμούμε, από την επιλογή Add που δίνεται στο πάνω μέρος του προγράμματος. Στο αριστερό μέρος εμφανίζεται η ιεραρχία της σκηνής που αποτελείται από τα αντικείμενα που έχουν εισαχθεί σε αυτή, όπως είναι το δάπεδο, οι κύβοι, η κάμερα και φωτισμός, τα οποία και επεξεργάζονται από αυτό το τμήμα του προγράμματος. Στο πάνω μέρος του βρίσκονται τα εργαλεία για την τοποθέτηση των αντικειμένων, την περιστροφή τους, έναρξη της προσομοίωσης, παύση κλπ.

Για την προσομοίωση στο CoppeliaSim, δημιουργήθηκαν διαφορετικές αρένες, στις οποίες έχουν εισαχθεί το ρομποτικό μας όχημα, αισθητήρες, κάμερα, γραφήματα, τείχη, εμπόδια, στατικοί και κινούμενοι στόχοι.

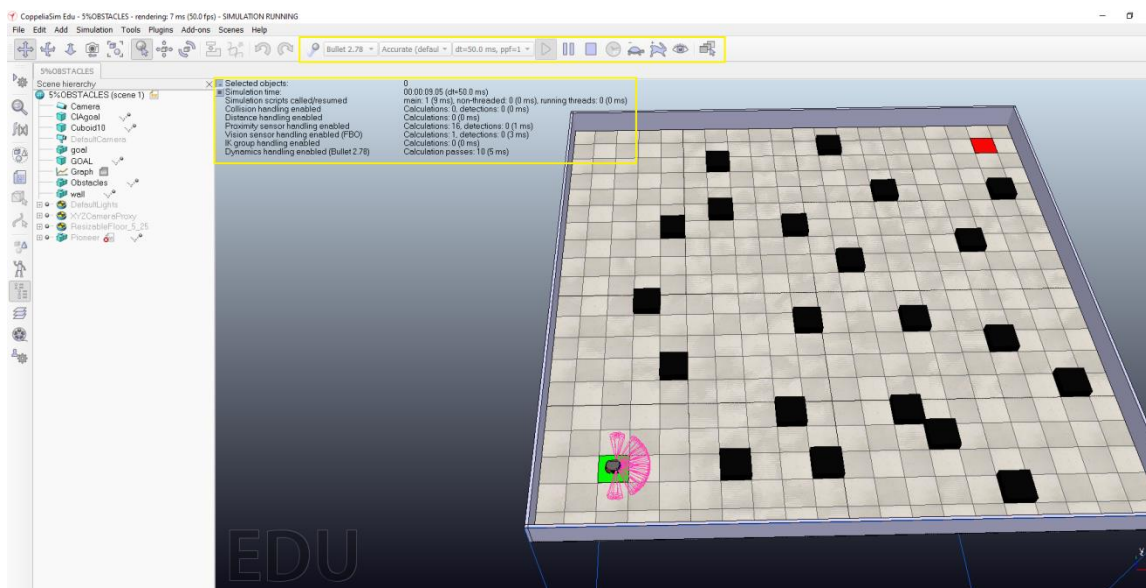


Σχήμα 3.3 Αρένα με εμπόδια και το ρομποτικό όχημα

Κάθε αντικείμενο στην ιεραρχία της σκηνής έχει συγκεκριμένες ιδιότητες τις οποίες μπορούμε να τις μεταβάλλουμε με βάση τις ανάγκες της προσομοίωσης που παρουσιάζονται, για παράδειγμα σε ένα εμπόδιο δίνεται η δυνατότητα επεξεργασίας της γεωμετρίας του, του χρώματος, της κατεύθυνσής του, των δυναμικών του ιδιοτήτων (Σχήμα 3.4).



Σχήμα 3.4 Επεξεργασία αντικειμένων



Σχήμα 3.5 Ενδείξεις δεδομένων κατά την προσομοίωση

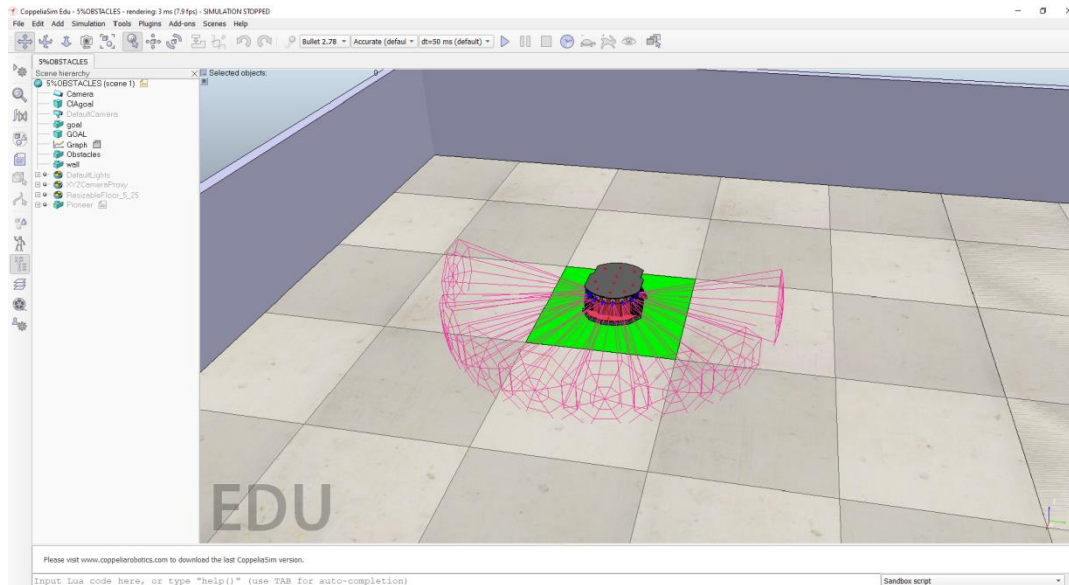
Ακόμα μία σημαντική δυνατότητα του προγράμματος, είναι να παρέχει πληροφορίες από τις ενδείξεις που λαμβάνονται κατά την διάρκεια της προσομοίωσης όπως είναι ο χρόνος της, τα scripts που καλούνται, δεδομένα από τους αισθητήρες, ανίχνευση πιθανής σύγκρουσης κ.α. Επίσης, διαθέτει την επιλογή μείωσης ή αύξησης της ταχύτητας προσομοίωσης καθώς και την επιλογή διαφορετικής προβολής της.

3.5 ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ ΓΙΑ ΧΕΙΡΙΣΜΟ ΤΟΥ CORPELIA ROBOTICS

Όπως αναφέρθηκε προηγουμένως, οι διαδικασίες για την δημιουργία και υλοποίηση της προσομοίωσης είναι εφικτό να πραγματοποιηθούν από το λογισμικό της Matlab ή από κάποια άλλη συμβατή γλώσσα προγραμματισμού. Αφού συνδεθούν αυτά τα δύο προγράμματα, απαραίτητη προϋπόθεση είναι να αρχικοποιηθούν όλα τα αντικείμενα στον κύριο κώδικα της Matlab, μέσω βιβλιοθηκών στην αντίστοιχη γλώσσα προγραμματισμού, όπου δίνει την δυνατότητα του ελέγχου της προσομοίωσης από μία εξωτερική εφαρμογή. Οι βιβλιοθήκες αυτές είναι διαθέσιμες στην ιστοσελίδα του CoppeliaSim, στο εγχειρίδιο χρήσης, για διάφορες γλώσσες προγραμματισμού.

Αρχικά, αρχικοποιούνται όλα τα χαρακτηριστικά που είναι απαραίτητα για το όχημα Pioneer 3-DX, δηλαδή οι τροχοί του και οι 16 αισθητήρες υπερήχων που χρειάζονται για την προσομοίωση. Στη συνέχεια τα σημεία και οι στόχοι που υπάρχουν στην αρένα όπως

η αρχική θέση, τελικός στόχος και κινούμενος στόχος που ακολουθεί καθώς και η κάμερα που έχει ενσωματωθεί πάνω από την αρένα που καταγράφει όλα τα αντικείμενα που βρίσκονται σε αυτή.



Σχήμα 3.6 Το ρομποτικό όχημα με τους αισθητήρες ενεργοποιημένους

Οι απαραίτητες εντολές για τις αρχικοποιήσεις αυτές λαμβάνονται από το εγχειρίδιο χρήσης του Coppelia και επεξεργάζονται κατάλληλα για τα αντικείμενα που χρειάζεται να δηλωθούν και περιγράφονται αναλυτικά οι παράμετροι που αντιστοιχούν σε αυτά. Μια ενδεικτική σύνταξη και οι σχετικές παράμετροι παρουσιάζονται στον Πίνακα 3.2.

simxGetObjectHandle (regular API equivalent: sim.getObjectHandle)	
Description	Retrieves an object handle based on its name.
Matlab synopsis	[number returnCode,number handle]=simxGetObjectHandle(number clientID,string objectName,number operationMode)
Matlab parameters	clientID : the client ID. refer to simxStart . objectName : name of the object. operationMode : a remote API function operation mode . Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	returnCode : a remote API function return code handle : the handle
Other languages	C/C++, Python, Java, Octave, Lua

Πίνακας 3.2 Συνάρτηση ελέγχου αντικειμένου

Στο πρώτο μέλος έχουμε τις τιμές που επιστρέφονται, το returnCode γυρνάει τις τιμές 0 σε περίπτωση που δεν βρέθηκε σύνδεση με το πρόγραμμα ή δεν βρέθηκε αντιστοιχία με το εμπόδιο και 1 αν η σύνδεση έγινε με επιτυχία. Με τον όρο handle, αναφερόμαστε στο αντικείμενο ενδιαφέροντος και το όνομα του στον κώδικα που θα χρησιμοποιηθεί για

μελλοντική χρήση. Στο δεξί μέλος βρίσκονται οι παράμετροι που δέχεται σαν είσοδοι όπως είναι το `clientID` που αναφέρεται στη σύνδεση των δύο προγραμμάτων, το όνομα όπως περιγράφεται στον προσομοιωτή CoppeliaSim και ο τρόπος λειτουργίας της συνάρτησης της βιβλιοθήκης. Ένας ενδεικτικός κώδικας επικοινωνίας με τη βοήθεια του Matlab είναι ο ακόλουθος.

```
%Pioneer
[errorCode, h.Pioneer] = sim.simxGetObjectHandle(clientID, 'Pioneer', ...
    sim.simx_opmode_oneshot_wait);
if errorCode ~= 0
    error('Error simxGetObjectHandle function for Pioneer object');
end
%Left Motor
[errorCode, h.leftMotor] = sim.simxGetObjectHandle(clientID, ...
    'Pioneer_p3dx_leftMotor', sim.simx_opmode_oneshot_wait);
if errorCode ~= 0
    error('Error simxGetObjectHandle function with left motor Pioneer
object');
end
%Sensor
[errorCode, h.US1] = sim.simxGetObjectHandle(clientID, ...
    'Pioneer_p3dx_ultrasonicSensor1', sim.simx_opmode_oneshot_wait);
if errorCode ~= 0
    error('Error simxGetObjectHandle function for US1 object');
end
%Goal
[errorCode, h.goal] = sim.simxGetObjectHandle(clientID, 'goal', ...
    sim.simx_opmode_oneshot_wait);
if errorCode ~= 0
    error('Error simxGetObjectHandle function for goal object');
end
```

Με τον τρόπο αυτό έχουμε τον πλήρη έλεγχο των παραμέτρων της αρένας που θα χρειαστούμε για την προσομοίωση. Για την λήψη της θέσης σε συντεταγμένες (x,y,z) και της κατεύθυνσης σε γωνίες Euler κάθε επιθυμητού αντικειμένου στην αρένα, δίνουν λύση οι εντολές `simxGetObjectIntParameter` και `simxGetObjectOrientation` οι οποίες και χρησιμοποιήθηκαν για το όχημα και τον στόχο.

Έξοδοι της συνάρτησης είναι επιτυχής ή όχι σύνδεση με το αντικείμενο και ένας πίνακας τριών θέσεων που αντιπροσωπεύουν τη θέση του (x,y,z). Οι είσοδοι της συνάρτησης είναι το αντικείμενο όπως το έχουμε αρχικοποιήσει προηγουμένως, η τιμή -1 που αντιστοιχεί στην απόλυτη θέση ή `sim_handle_parent` για επιλογή άλλου αντικειμένου ως

πλαίσιο αναφοράς και τέλος η λειτουργία της συνάρτησης. Το σχετικό τμήμα κώδικα παρατίθεται στη συνέχεια.

```
%% Retrieves goal position
[errorCode, goal_position] =
sim.simxGetObjectPosition(clientID,h.goal,-
1,sim.simx_opmode_streaming);
pause(0.1)
[errorCode, goal_position] = sim.simxGetObjectPosition(clientID,
h.goal, -1,sim.simx_opmode_buffer);
if errorCode ~= 0
    error('Error with simxGetObjectPosition for goal');
end
```

```
%% Retrieve the position of the car
[errorCode, actual_position] =
sim.simxGetObjectPosition(clientID,h.Pioneer,-
1,sim.simx_opmode_streaming);
pause(0.1)
[errorCode, actual_position] = sim.simxGetObjectPosition(clientID,
h.Pioneer,-1,sim.simx_opmode_buffer);
if errorCode ~= 0
    error('Error, cannot find the actual position of the car');
end
```

```
%% Retrieve the orientation (Euler angles) of the car
[errorCode,actual_orientation]=sim.simxGetObjectOrientation(clientID,h.
Pioneer,-1,sim.simx_opmode_streaming);
pause(0.1)
[errorCode,actual_orientation]=sim.simxGetObjectOrientation(clientID,h.
Pioneer,-1,sim.simx_opmode_buffer);
if errorCode ~= 0
    error('Error, cannot find the actual orientation of the car');
end
```

Εκτός του οχήματος, δίνεται η δυνατότητα ελέγχου των αισθητήρων ως προς τις τιμές τις οποίες λαμβάνουν. Η συνάρτηση δεν εκτελεί ανίχνευση αλλά μετράει τα αποτελέσματα από προηγούμενη συνάρτηση η οποία καλείται στο προκαθορισμένο main script.

Με τις τιμές των εξόδων να αναφέρονται στην σύνδεση με το πρόγραμμα και το αντικείμενο, στην κατάσταση της ανίχνευσης (1=ανίχνευση ή 0=δεν υπάρχει ανίχνευση), τις συντεταγμένες του σημείου ανίχνευσης σε σχέση με το πλαίσιο αναφοράς του

αισθητήρα, το αντικείμενο που ανιχνεύεται, ο ομαλοποιημένος άξονας της ανιχνεύσιμης επιφάνειας σε σχέση με το πλαίσιο αναφοράς του αισθητήρα.

Χρησιμοποιείται ο προτεινόμενος τρόπος λειτουργίας σε δύο στάδια. Στη πρώτη κλήση της συνάρτησης η εντολή αποστέλλεται και μία προηγούμενη απάντηση επιστρέφεται στην ίδια εάν υπάρχει, η εντολή εκτελείται συνέχεια στον διακομιστή και απαιτείται μία καθυστέρηση f προτού εκτελεστεί ξανά η ίδια εντολή από τον διακομιστή. Στη δεύτερη κλήση της η απάντηση επιστρέφεται πάλι στην ίδια εντολή, αλλά η εντολή δεν αποστέλλεται και η συνάρτηση δεν περιμένει πραγματική απάντηση.

```
%% Recieves data sensor
[errorCode,detect1,detectedP1,detectedObjectHandle,detectedSurfaceNV]=
...

sim.simxReadProximitySensor(clientID,h.US1,sim.simx_opmode_streaming+f)
;
pause(k);
[errorCode,detect1,detectedP1,detectedObjectHandle,detectedSurfaceNV]=
...
    sim.simxReadProximitySensor(clientID,h.US1,sim.simx_opmode_buffer);
pause(o);
```

Για την κάμερα χρησιμοποιήθηκε διαφορετική συνάρτηση αφού πρόκειται για άλλο είδος αισθητήρα η οποία τοποθετείται πάνω από την αρένα. Ο σκοπός της είναι να λαμβάνει την εικόνα της αρένας μαζί με όσα περιλαμβάνει αυτή, όπως το όχημα, τους τοίχους, τους στόχους και τα εμπόδια.

Ως έξοδοι της συνάρτησης λαμβάνονται η ανάλυση της εικόνας και η εικόνα ως πίνακας που έχει διάσταση ίση με το μέγεθος της ανάλυσης (δηλ. 128x128 ανάλυση επιστρέφει πίνακα διαστάσεων 128x128) και παίρνει τιμές στο διάστημα 0-255 με το άσπρο να αντιστοιχεί στην τιμή 255 και το μαύρο στη τιμή 0. Οι είσοδοι που απαιτούνται είναι το αντικείμενο όπως έχει αρχικοποιηθεί, οι επιλογές της εικόνας με 0 να αναφέρεται σε κλίμακα του γκρι (ασπρόμαυρη) και 1 σε RGB (Red-Green-Blue έγχρωμη) και η λειτουργία της συνάρτησης. Ο σχετικός κώδικας είναι ο ακόλουθος.

```

%% Recieves Camera Points
[errorCode,resolution,image] =
sim.simxGetVisionSensorImage2(clientID,h.Camera,1,sim.simx_opmode_streaming
pause(0.1)
[errorCode,resolution,image] = sim.simxGetVisionSensorImage2(clientID,
h.Camera, 1,sim.simx_opmode_buffer
if errorCode ~= 0
    error('Error with simxGetVisionSensorImage2 for Camera');
end

```

Τελευταία και από τις πιο χρήσιμες συναρτήσεις που εφαρμόστηκαν είναι αυτή της κίνησης του οχήματος. Η κίνηση δίνεται στις αρθρώσεις του αντικειμένου, οι οποίες στο ρομποτικό όχημα Pioneer είναι οι τροχοί του. Η εντολή αυτή έχει νόημα μόνο εάν οι αρθρώσεις κινούνται με ροπή / δύναμη και παράλληλα πρέπει να είναι ενεργοποιημένη η δυναμική λειτουργία και ο κινητήρας των αρθρώσεων. Εδώ λαμβάνεται μία έξοδος, η σύνδεση με το αντικείμενο και οι είσοδοι του αποτελούνται από την άρθρωση που απαιτείται να κινηθεί, την επιθυμητή ταχύτητα της άρθρωσης που εξαρτάται από τον τύπο της, γραμμική ή γωνιακή ταχύτητα, και την λειτουργία της συνάρτησης.

```

[errorCode]=sim.simxSetJointTargetVelocity(clientID,h.leftMotor,...
-1,sim.simx_opmode_streaming);

[errorCode]=sim.simxSetJointTargetVelocity(clientID,h.rightMotor,...
3,sim.simx_opmode_streaming);

[errorCode,linearVelocity1,angularVelocity1]=sim.simxGetObjectVelocity(
clientID, h.leftMotor,sim.simx_opmode_streaming);

[errorCode,linearVelocity2,angularVelocity2]=sim.simxGetObjectVelocity(
clientID, h.rightMotor,sim.simx_opmode_streaming);

```

Στο τέλος της υλοποίησης του κώδικα για να διακόψουμε τη σύνδεση μεταξύ των δύο προγραμμάτων χρειάζεται η εντολή η οποία καλείται μετά την επιτυχημένη κλήση της αρχικής συνάρτησης `simxStart`.

```
sim.simxFinish(-1);
```

ΚΕΦΑΛΑΙΟ 4

ΕΦΑΡΜΟΓΗ / ΥΛΟΠΟΙΗΣΗ

4.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό μελετάται η συμπεριφορά ενός ρομποτικού οχήματος σε ένα περιβάλλον στο οποίο παρουσιάζονται πολλαπλοί βαθμοί δυσκολίας ως προς την κίνηση του (εμπόδια, συνεχής παρακολούθηση στόχων), από μία αρχική θέση εκκίνησης μέχρι την άφιξή του σε ένα τελικό σημείο-στόχο. Στην κίνηση του οχήματος εφαρμόζονται τρεις διαφορετικοί ευρετικοί αλγόριθμοι (A^* , CIA^* , $Weighted A^*$) κατά τους οποίους πραγματοποιείται ανίχνευση του περιβάλλοντος και γίνεται η μελέτη και η σύγκριση τους στην εύρεση της συντομότερης διαδρομής, πιο συγκεκριμένα της διαδρομής με το μικρότερο κόστος (μικρότερη απόσταση που διανύει, ελάχιστος χρόνος περάτωσης κτλ). Όπως αναφέρθηκε στο κεφάλαιο 3 χρησιμοποιήθηκε ο ρεαλιστικός προσομοιωτής CoppeliaSIM για την δημιουργία αρένων και την επεξεργασία τους και ένα πραγματικό ρομποτικό όχημα που επιλέχθηκε, το Pioneer 3-DX. Σε πρώτο στάδιο πραγματοποιήθηκαν προσομοιώσεις εκτός του CoppeliaSIM ώστε να ανακτηθούν και αργότερα να επαληθευτούν οι βέλτιστες διαδρομές για τον κάθε κόσμο και κάθε ευρετικό αλγόριθμο ξεχωριστά. Στη συνέχεια εκτελέστηκαν στον προσομοιωτή όπου το ρομποτικό όχημα πραγματοποιεί ανίχνευση της διαδρομής όπου στο τέλος της εξάγεται η βέλτιστη διαδρομή. Οι προσομοιώσεις έγιναν στον ίδιο ηλεκτρονικό υπολογιστή με χαρακτηριστικά επεξεργαστή Intel® Core™ i5-8250U 1.80GHz, λειτουργικό σύστημα 64bit υπό σταθερές συνθήκες.

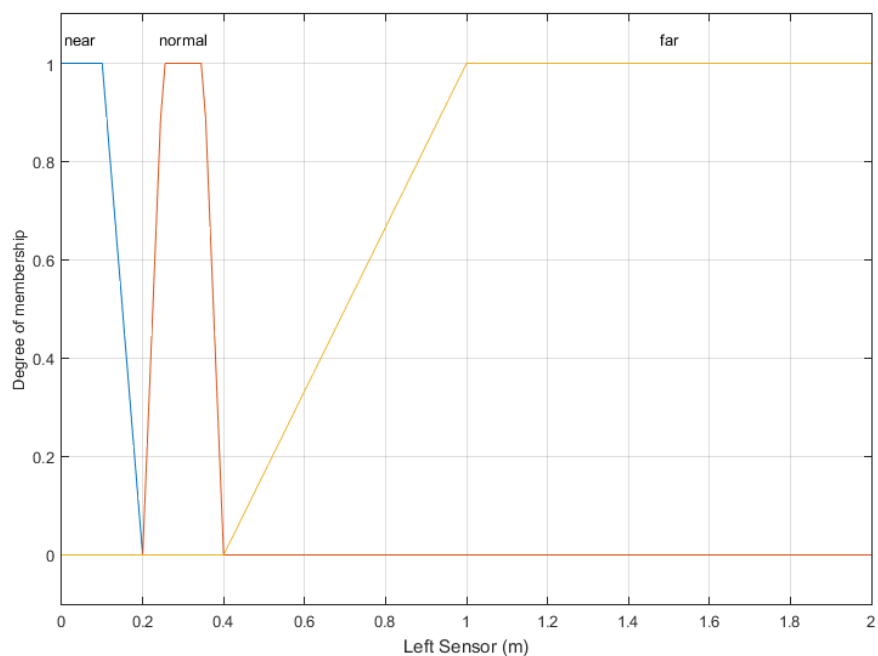
4.2 ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΕΛΕΓΚΤΩΝ ΚΙΝΗΣΗΣ ΡΟΜΠΟΤΙΚΟΥ ΟΧΗΜΑΤΟΣ

Σκοπός σε αυτό το στάδιο είναι υλοποίηση ενός ελεγκτή που οδηγεί το διαφορικό όχημα μέσα στο χώρο, αποφεύγοντας τα εμπόδια με αισθητήρες υπερήχων και την άφιξη του σε ένα στόχο μέσω ενδιάμεσων σημείων-στόχων. Οι αισθητήρες που έχουν ενεργοποιηθεί για την ανίχνευση εμποδίων είναι οι 8 μπροστά και έχουν κατανεμηθεί σε τρεις ομάδες όπου οι τρεις αριστερά αποτελούν αριστερή πλευρά, οι δύο την κεντρική πλευρά και οι τρεις υπόλοιποι την δεξιά πλευρά. Η ομαδοποίηση αυτή έγινε για την απλούστευση των δεδομένων που λαμβάνονται από τους αισθητήρες και για αυτό το λόγο επιλέγεται η ελάχιστη τιμή που καταγράφεται από τον κάθε αισθητήρα της ομάδας. Με βάση τα δεδομένα αυτά δίνεται η κατάλληλη ταχύτητα των τροχών ώστε να αποφεύγει τα εμπόδια. Για παράδειγμα εάν βρεθεί τιμή από την αριστερή πλευρά του οχήματος αυξάνεται η ταχύτητα του αριστερού τροχού και μειώνεται ή μένει σταθερή αυτή του δεξιού τροχού ανάλογα με την εξαγόμενη τιμή των αισθητήρων. Παράλληλα με την διαδικασία αποφυγής εμποδίων το ρομποτικό όχημα πρέπει να είναι σε θέση να ακολουθεί τους στόχους που προκύπτουν κατά τη διάρκεια της προσομοίωσης ώστε να εκτελεί τις διαδικασίες του αλγορίθμου και να μεταβεί στην τελική θέση που του έχει ανατεθεί. Έτσι μετράτε η γωνία μεταξύ μεταξύ του οχήματος και της επιθυμητής θέσης-στόχου και δίνονται οι ταχύτητες των τροχών ανάλογα με την τιμή της γωνίας σφάλματος.

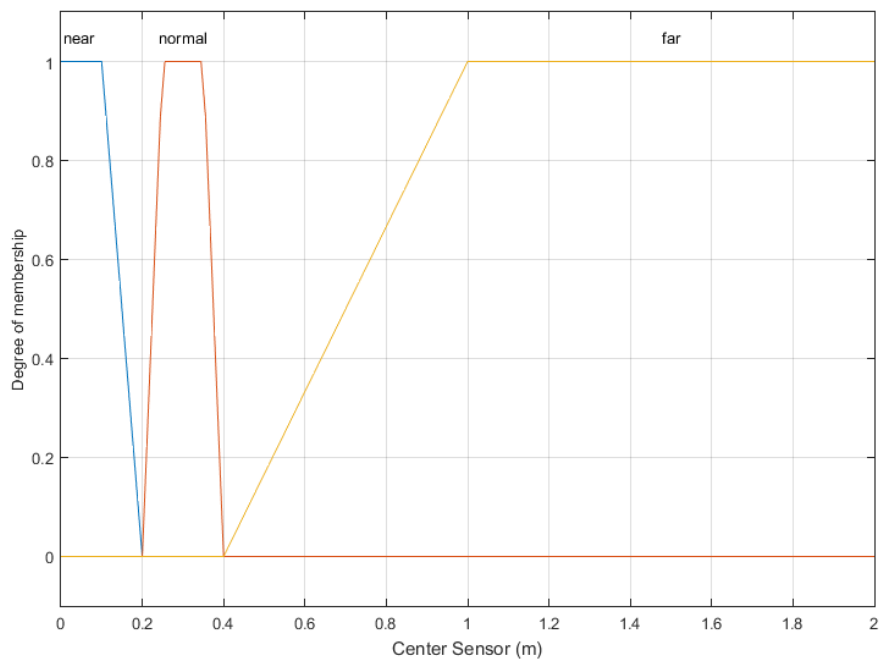
Αφού έγινε η μελέτη του πρώτου ελεγκτή κίνησης αναπτύχθηκε ένας ακόμα με την διαδικασία της ασαφούς λογικής, παρόμοιου με τον προηγούμενο για λόγους σύγκρισης συμπεριφοράς του οχήματος. Έτσι οι τρεις πρώτοι είσοδοι για τον ασαφή ελεγκτή που δημιουργήθηκε είναι οι τιμές των αισθητήρων (αριστερός, κεντρικός, δεξιός) και η τέταρτη είσοδος η γωνία σφάλματος. Οι έξοδοι του ελεγκτή είναι δύο και αντιστοιχούν στους δύο τροχούς του οχήματος. Παρουσιάζονται οι συναρτήσεις συμμετοχής των εισόδων και των εξόδων του ασαφούς ελεγκτή που δημιουργήθηκε για την κίνηση του ρομποτικού οχήματος.

Οι πρώτες τρεις είσοδοι είναι παρόμοιες και αποτελούνται από 3 ασαφή σύνολα που χαρακτηρίζονται από τις συναρτήσεις συμμετοχής κοντά, κανονικά και μακριά. Το εύρος

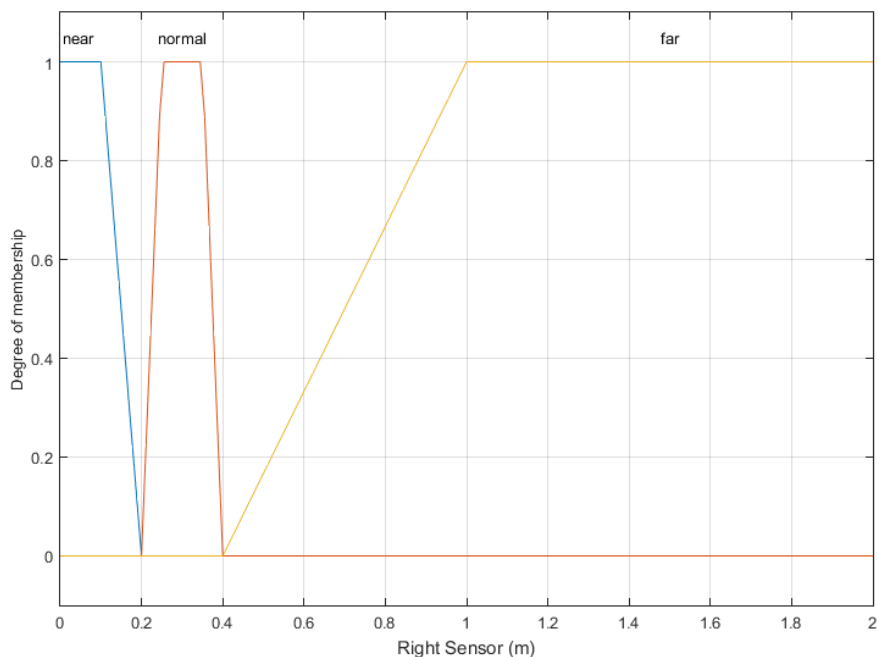
των τιμών είναι από 0-2 μέτρα ενώ η περιοχή ανίχνευσης των εμποδίων των αισθητήρων είναι από 0-1 μέτρο.



Σχήμα 4.1 Είσοδος «Αριστερός Αισθητήρας» ασαφούς ελεγκτή

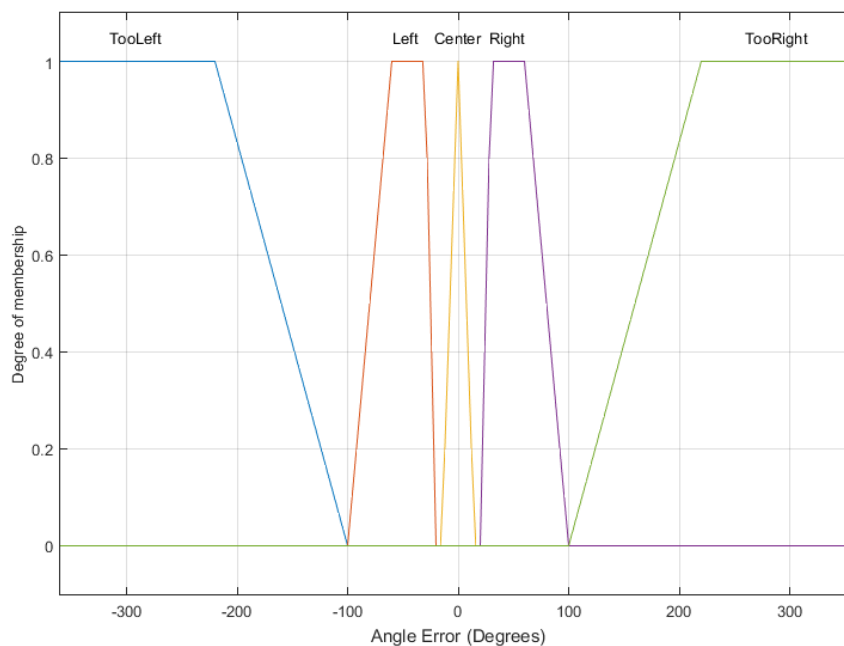


Σχήμα 4.2 Είσοδος «Κεντρικός Αισθητήρας» ασαφούς ελεγκτή



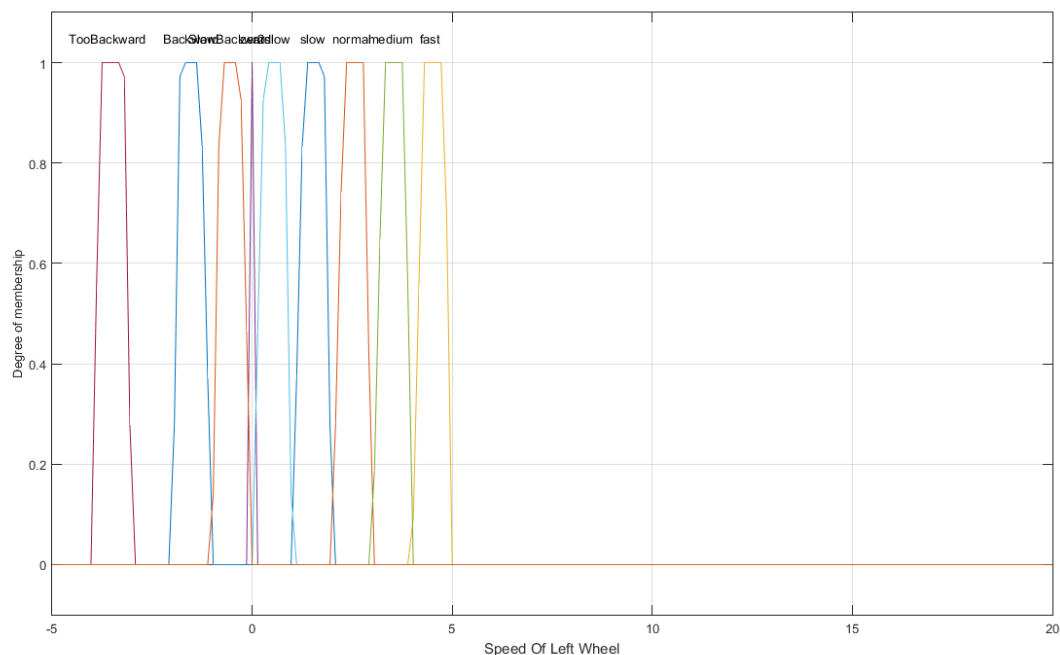
Σχήμα 4.3 Είσοδος «Δεξιός Αισθητήρας» ασαφούς ελεγκτή

Η τέταρτη είσοδος αναφέρεται στην γωνία σφάλματος και αποτελείται από 5 ασαφή σύνολα με συναρτήσεις συμμετοχής πολύ αριστερά, αριστερά, κέντρο, δεξιά, πολύ δεξιά και εύρος από -360° έως 360° .

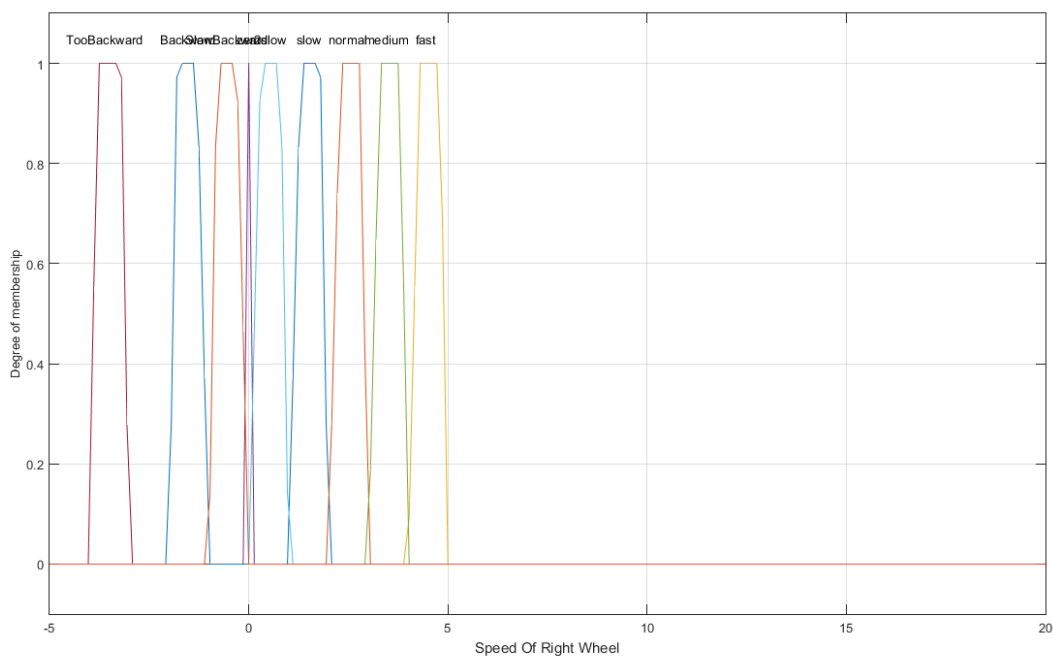


Σχήμα 4.4 Είσοδος «Γωνία Σφάλματος» ασαφούς ελεγκτή

Οι έξοδοι του ελεγκτή είναι οι ταχύτητες των δύο τροχών και αποτελούνται από 9 ασαφή σύνολα με συναρτήσεις συμμετοχής πολύ όπισθεν, όπισθεν, αργά όπισθεν, μηδέν, πολύ αργά, αργά, κανονικά, μέτρια, γρήγορα με εύρος από -4 έως 5 meter/second.



Σχήμα 4.5 Έξοδος «Ταχύτητα Αριστερού Τροχού» ασαφούς ελεγκτή



Σχήμα 4.6 Έξοδος «Ταχύτητα Δεξιού Τροχού» ασαφούς ελεγκτή

Αφού έχει γίνει η κατανομή εισόδων και εξόδων και έχουν αποθηκευτεί υπό την μορφή συναρτήσεων συμμετοχής ακολουθεί η διατύπωση των κανόνων. Δημιουργήθηκαν 31 λεκτικοί κανόνες μερικούς από τους οποίους αναφέρονται στον παρακάτω πίνακα.

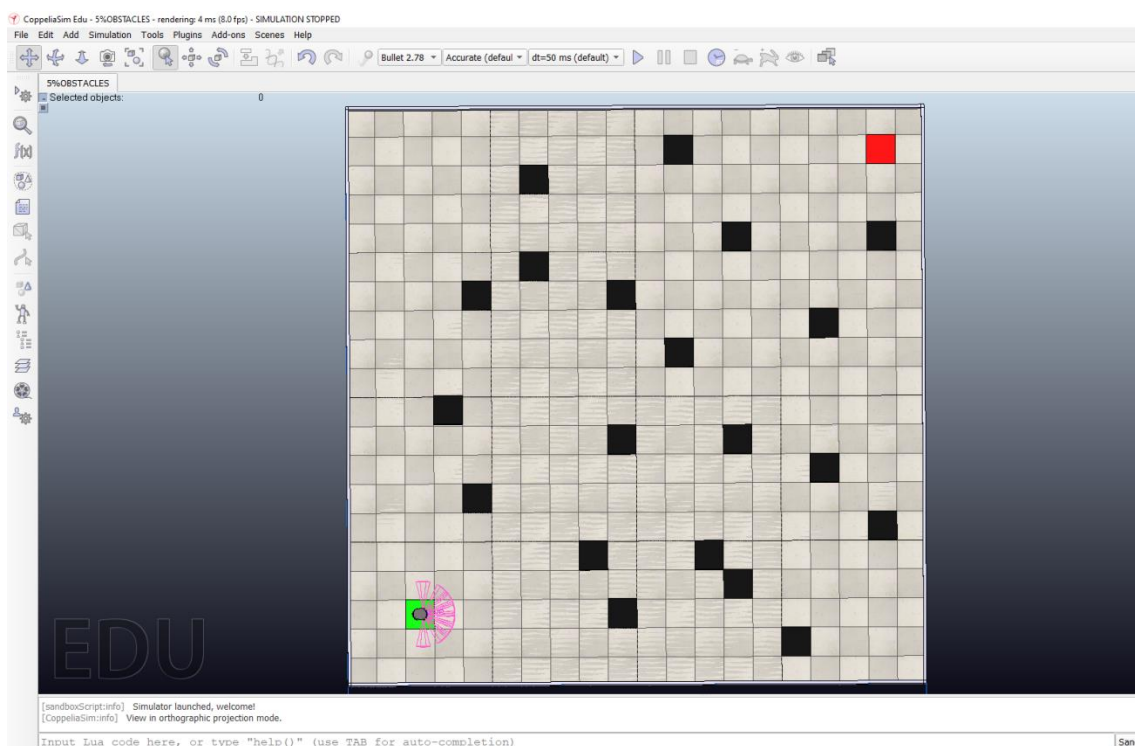
ΛΕΚΤΙΚΟΙ ΚΑΝΟΝΕΣ							
	Είσοδος 1	Είσοδος 2	Είσοδος 3	Είσοδος 4		Έξοδος 1	Έξοδος 2
	Αριστερά Αισθητήρας	Κεντρικός Αισθητήρας	Δεξιός Αισθητήρας	Γωνία Σφάλματος		Αριστερή Ταχύτητα	Δεξιά Ταχύτητα
	Μακριά	Μακριά	Κανονικά	-		Αργά Όπισθεν	Κανονικά
	Μακριά	Κανονικά	Μακριά	-		Αργά	Αργά Όπισθεν
	Μακριά	Κοντά	Μακριά	-		Μέτρια	Μηδέν
	Μακριά	Κοντά	Κοντά	-		Μηδέν	Γρήγορα
	Κανονικά	Μακριά	Κανονικά	-		Κανονικά	Κανονικά
	Κανονικά	Κανονικά	Μακριά	-		Μέτρια	Αργά Όπισθεν
Εάν	Κανονικά	Κοντά	Μακριά	-	Τότε	Μέτρια	Μηδέν
	Κανονικά	Κοντά	Κοντά	-		Μηδέν	Μέτρια
	Κοντά	Μακριά	Κοντά	-		Μηδέν	Αργά
	Κοντά	Κανονικά	Κοντά	-		Μέτρια	Μηδέν
	Κοντά	Κοντά	Κοντά	-		Μέτρια	Πολύ Όπισθεν
	Κοντά	Κοντά	Μακριά	-		Αργά	Όπισθεν
	Μακριά	Μακριά	Μακριά	Πολύ Αριστερά		Πολύ Όπισθεν	Γρήγορα
	Μακριά	Μακριά	Μακριά	Κέντρο		Μέτρια	Μέτρια
	Μακριά	Μακριά	Μακριά	Δεξιά		Μέτρια	Όπισθεν
	Μακριά	Μακριά	Μακριά	Πολύ Δεξιά		Γρήγορα	Πολύ Όπισθεν

Πίνακας 4.1 Βάση κανόνων ασαφούς ελεγκτή για αποφυγή εμποδίων

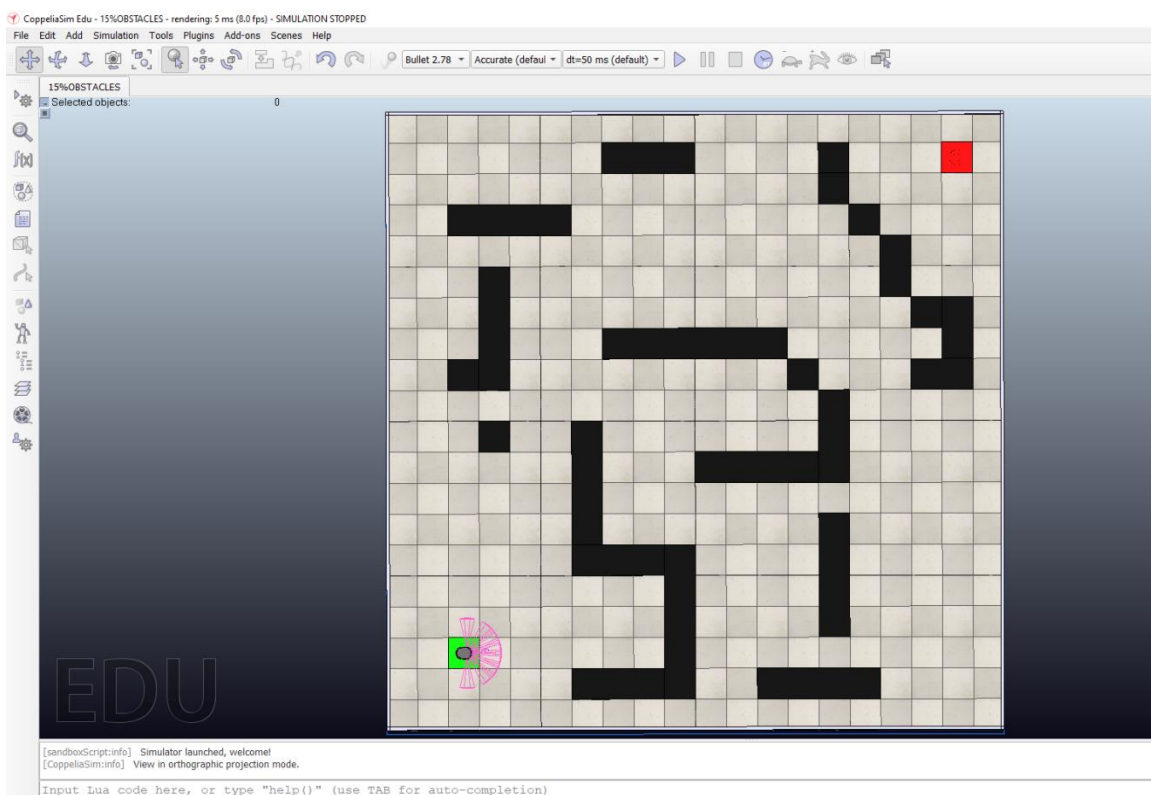
Τέλος, είναι απαραίτητος ο καθορισμός του τύπου συνεπαγωγής όπου και επιλέχθηκε ο τελεστής max-min τύπου Mamdani.

4.3 ΣΕΝΑΡΙΑ ΠΟΛΛΑΠΛΩΝ ΒΑΘΜΩΝ ΔΥΣΚΟΛΙΑΣ

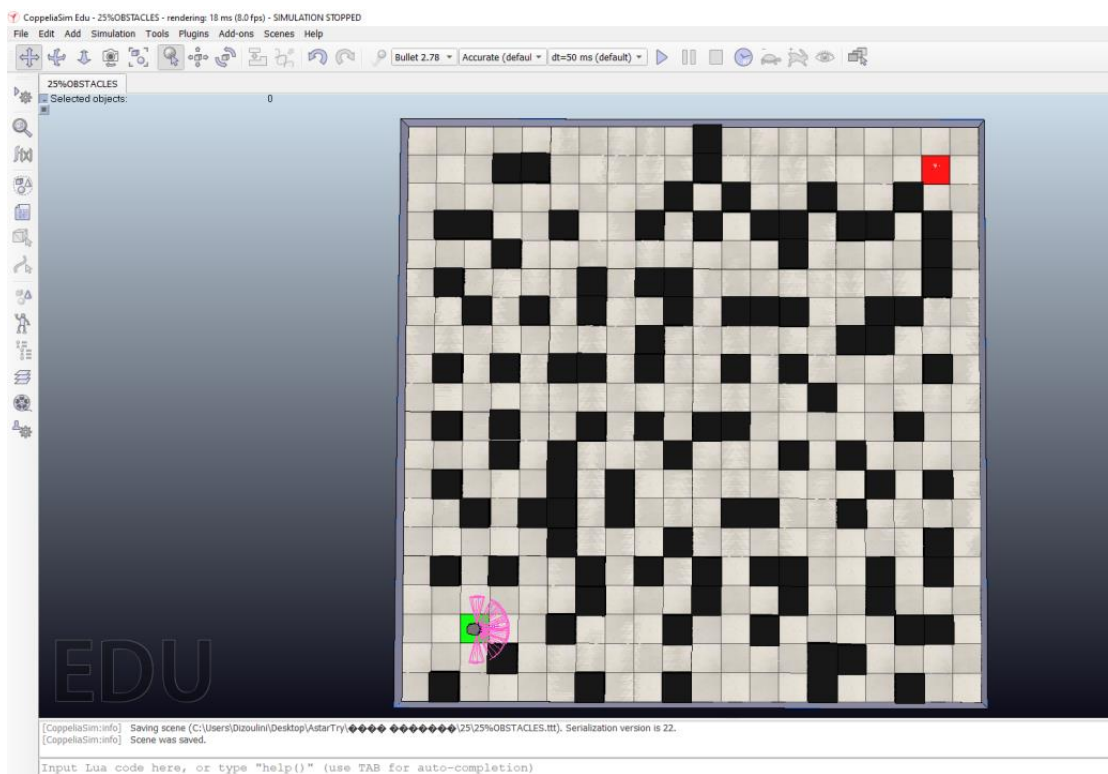
Για τον σκοπό της εργασίας δημιουργήθηκαν 12 αρένες στις οποίες απαιτείται να έχουν αυξημένο αριθμό δυσκολίας κίνησης του οχήματος στο χώρο ώστε να μελετηθεί η συμπεριφορά της κίνησης και γι' αυτό εντάχθηκε μεγαλύτερος αριθμός εμποδίων σε σχέση με τις διαστάσεις της αρένας. Οι διαστάσεις που επιλέχθηκαν είναι μεγέθους 20x20 μέτρα και 40x40 μέτρα με το κάθε κελί να είναι 1x1 μέτρο και ο βαθμός δυσκολίας ανέρχεται σε 6 επίπεδα ως προς τον αριθμό των εμποδίων, μεγέθους 5- 30%. Οι θέσεις των εμποδίων που έχουν εισαχθεί, επιλέχθηκαν τυχαία και το μέγεθος τους είναι ίσο με αυτό ενός κελιού (1x1 μέτρο). Ενδεικτικά παρουσιάζονται ορισμένες αρένες στις παρακάτω εικόνες.



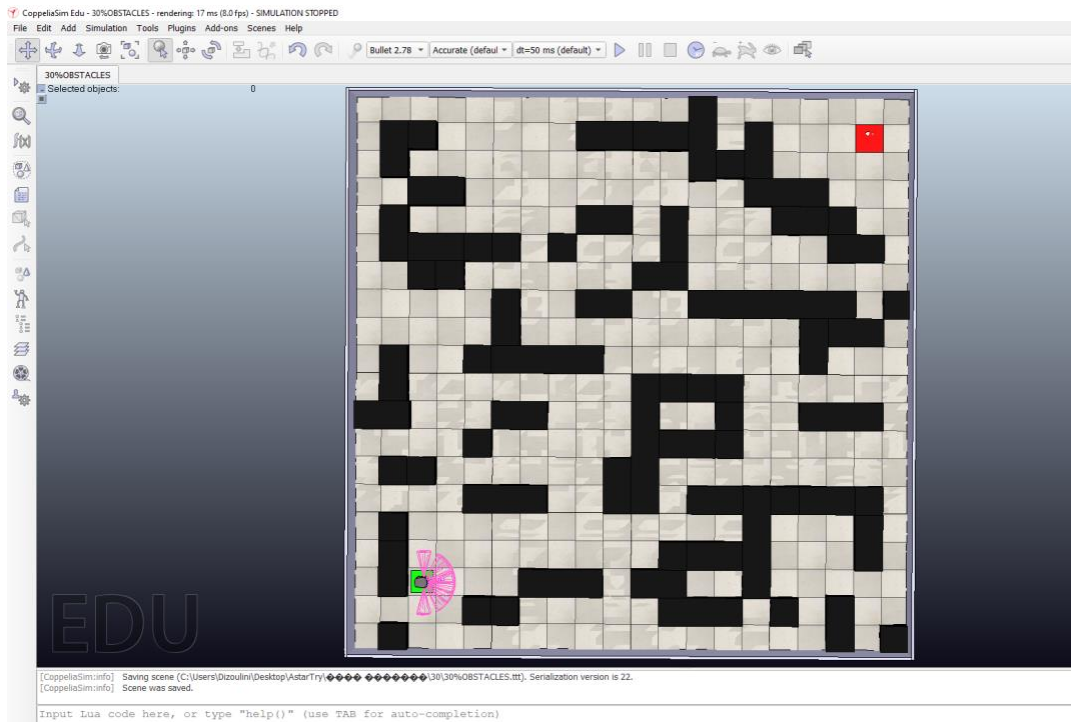
Εικόνα 4.7 Αρένα διαστάσεων 20x20 με βαθμό δυσκολίας 5% (20 εμπόδια)



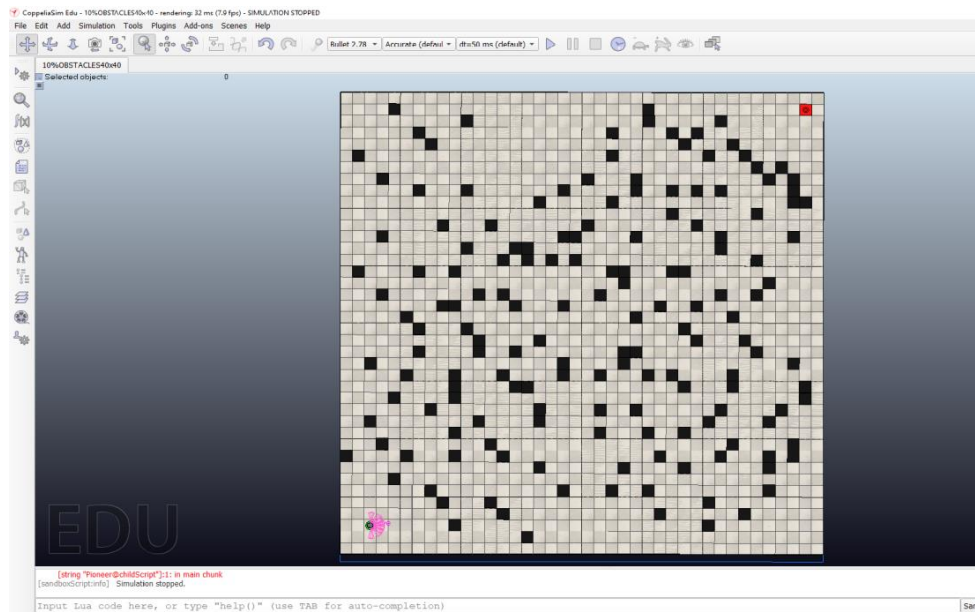
Εικόνα 4.8 Αρένα διαστάσεων 20x20 με βαθμό δυσκολίας 15% (60 εμπόδια)



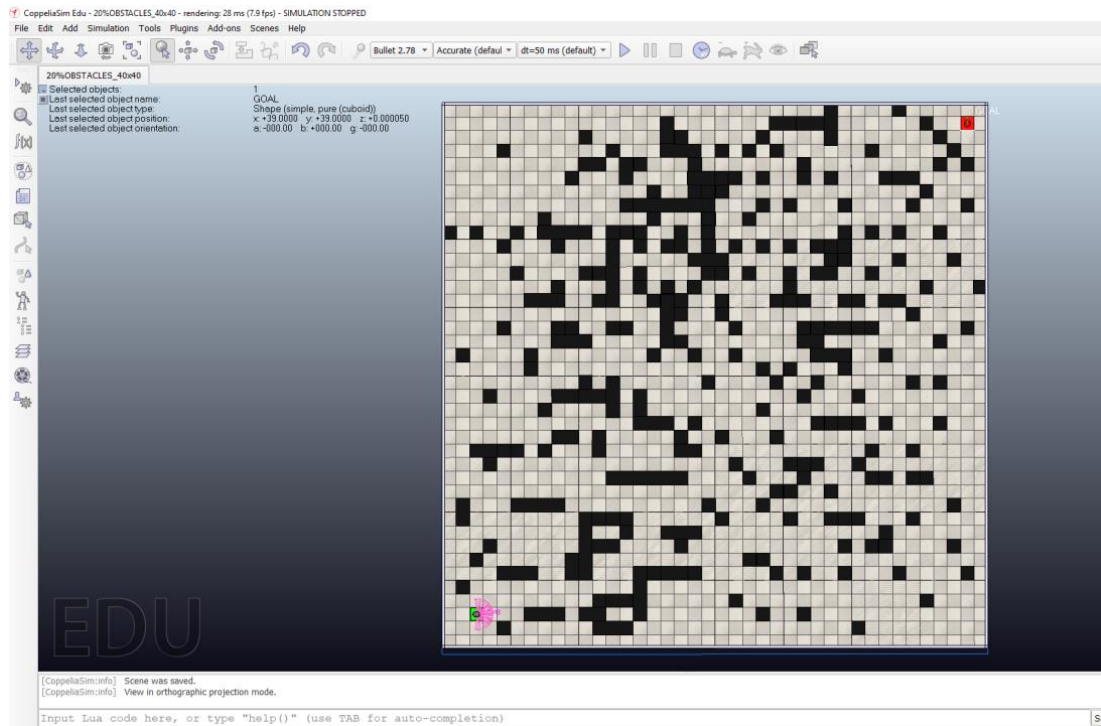
Εικόνα 4.9 Αρένα διαστάσεων 20x20 με βαθμό δυσκολίας 25% (100 εμπόδια)



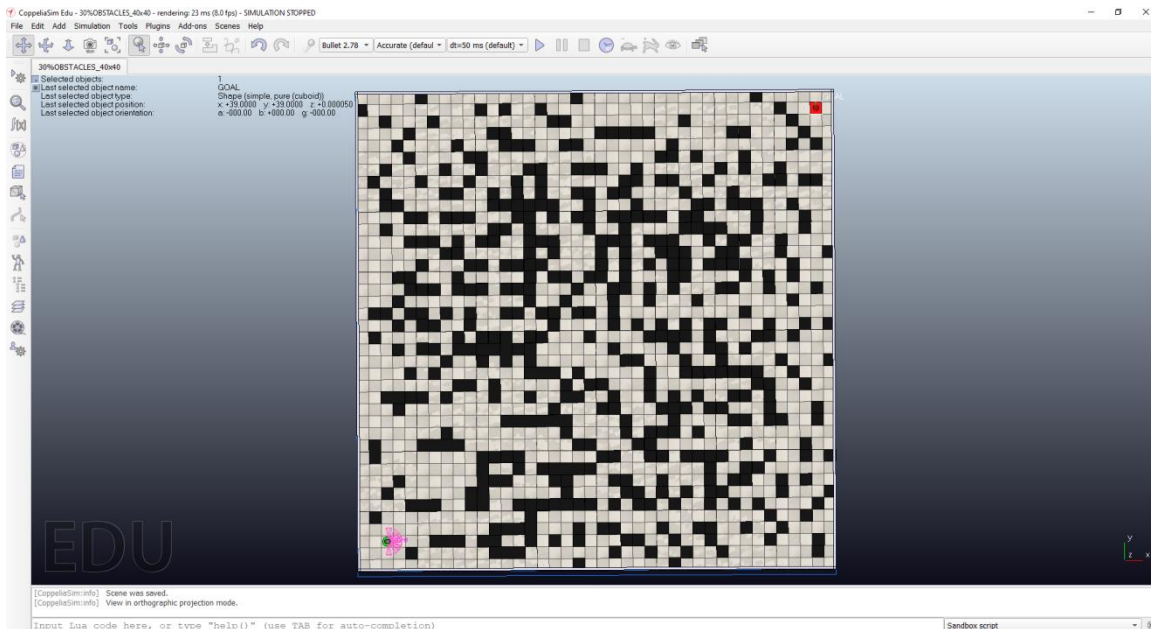
Εικόνα 4.10 Αρένα διαστάσεων 20x20 με βαθμό δυσκολίας 30% (120 εμπόδια)



Εικόνα 4.11 Αρένα διαστάσεων 40x40 με βαθμό δυσκολίας 10% (160 εμπόδια)



Εικόνα 4.12 Αρένα διαστάσεων 40x40 με βαθμό δυσκολίας 20% (320 εμπόδια)



Εικόνα 4.13 Αρένα διαστάσεων 40x40 με βαθμό δυσκολίας 30% (480 εμπόδια)

4.4 ΕΥΡΕΣΗ ΒΕΛΤΙΣΤΗΣ ΔΙΑΔΡΟΜΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΕΥΡΕΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

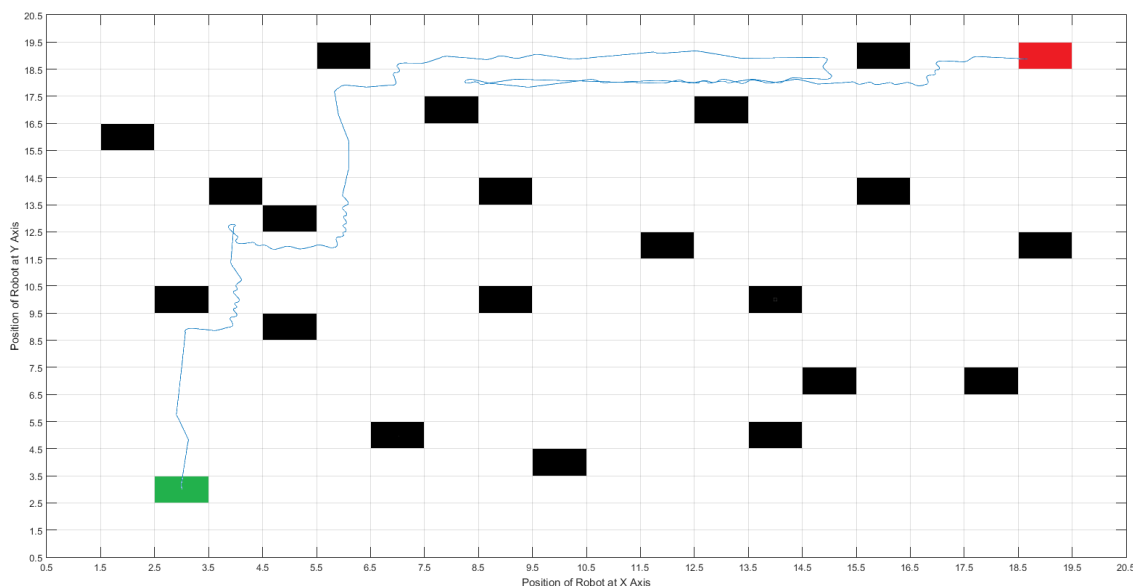
Σκοπός του δεύτερου σταδίου της εργασίας είναι η σύγκριση στην εύρεση της βέλτιστης διαδρομής από την αρχική θέση του ρομποτικού οχήματος (πράσινο σημείο) στον τελικό στόχο του (κόκκινο). Αυτό επιτυγχάνεται με την υλοποίηση τριών διαφορετικών ευριστικών αλγορίθμων οι οποίοι συνδυάζονται μεταξύ τους. Σε κάθε σενάριο εισήχθηκε μία κάμερα μικρής ανάλυσης η οποία λαμβάνει την εικόνα ολόκληρης της αρένας έτσι ώστε να καταγράφει την θέση και τον αριθμό των εμποδίων.

Τα δεδομένα από την κάμερα μετατρέπονται σε πίνακα διαστάσεων ίσο με τις διαστάσεις της αρένας, καταγράφεται η τρέχουσα θέση του, το μέγεθος των κελιών, η δυνατότητα μετάβασης από ένα κελί σε ένα άλλο (μετάβαση αριστερά, δεξιά, ευθεία πίσω) και το κόστος για αυτή την μετακίνηση. Έτσι λοιπόν σε κάθε βήμα της προσομοίωσης υπολογίζεται η επόμενη κίνηση του οχήματος με βάση των εκάστοτε χρησιμοποιούμενο αλγόριθμο (A^* , CIA^* , $Weighted A^*$) και δίνεται στο ρομπότ το επόμενο σημείο στο οποίο πρέπει να μεταβεί. Κατά την υλοποίηση των αλγορίθμων το ρομποτικό όχημα πραγματοποιεί ανίχνευση των εμποδίων και η πληροφορία και γίνεται επαναπροσδιορισμός του μονοπατιού.

Ο τρόπος με τον οποίο δουλεύουν οι προτεινόμενοι αλγόριθμοι έχει αναλυθεί, και συγκεκριμένα τα σημεία που βρίσκονται στην λίστα CLOSED, δηλαδή αυτά που έχουν επιλεγθεί με το μικρότερο κόστος f , είναι πιθανόν να επανεξετάζονται και έτσι οι στόχοι που δίνονται στο ρομποτικό όχημα δεν βρίσκονται σε διαδοχικές θέσεις. Για την ασφαλή μετάβαση σε αυτά τα σημεία υλοποιείται ο αλγόριθμος A^* που καλείται ως συνάρτηση στο πρόγραμμα και υπολογίζει την εύρεση της διαδρομής με τη χρήση διαφορετικών εισόδων. Ως τελικός στόχος ορίζεται το εξαγόμενο σημείο και ως αρχική θέση είναι η τρέχουσα θέση του ρομποτικού οχήματος που βρίσκεται σε εκείνο το βήμα της προσομοίωσης. Στη συνέχεια θα μελετηθεί η απόδοση των τριών προτεινόμενων αλγορίθμων με τη χρήση ενός απλού ελεγκτή όπου στη περίπτωση ύπαρξης εμποδίου, που εμποδίζει τη μετάβαση σε ένα σημείο στόχο το όχημα απλά στρίβει προς μια συγκεκριμένη κατεύθυνση. Ανάλογα πειράματα θα πραγματοποιηθούν και για ένα όχημα ελεγχόμενο από ένα ελεγκτή ασαφούς λογικής.

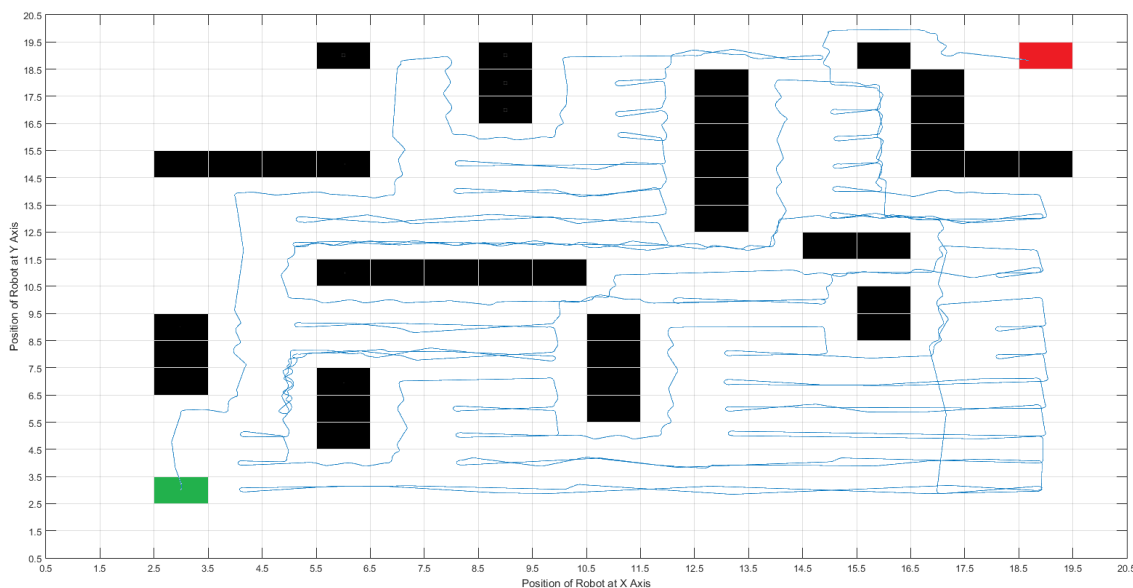
4.4.1 Αποτελέσματα προσομοίωσης με χρήση του αλγορίθμου A*

Στα σχήματα 4.14 με 4.19 παρουσιάζονται ο χώρος που κινήθηκε το όχημα και η διαδρομή που ακολούθησε με βάση τον αλγόριθμο A* για την εύρεση της βέλτιστης διαδρομής σε αρένες διαστάσεων 20x20 με αυξανόμενο βαθμό δυσκολίας.



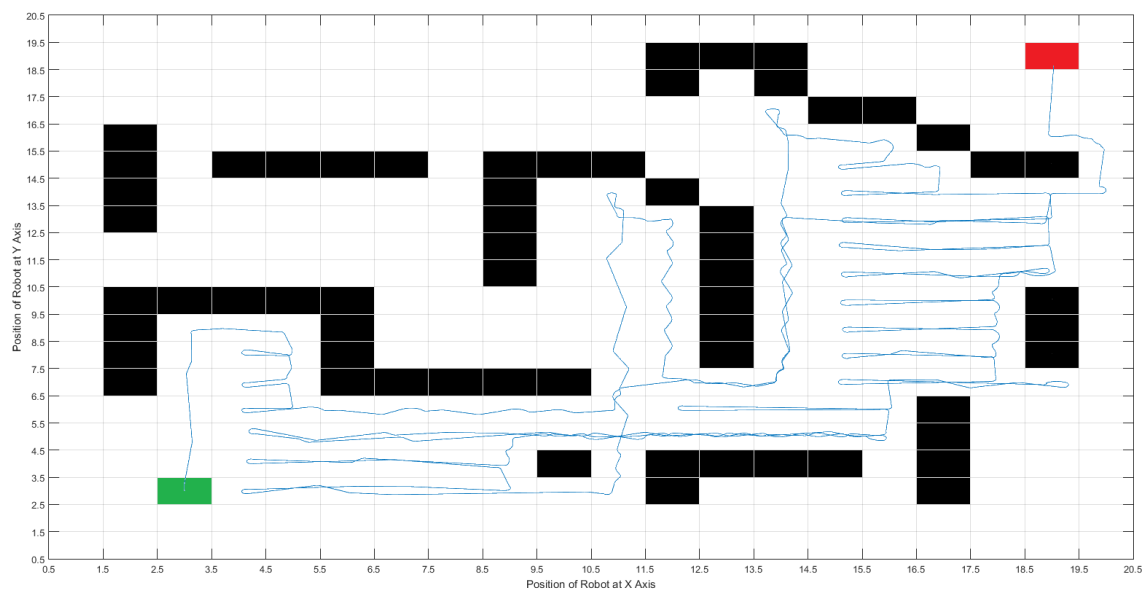
Σχήμα 4.14 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%

Το όχημα χρειάστηκε 99.61 δευτερόλεπτα για την εύρεση την βέλτιστης διαδρομής και διήγνυσε συνολικά 50.13 μέτρα.



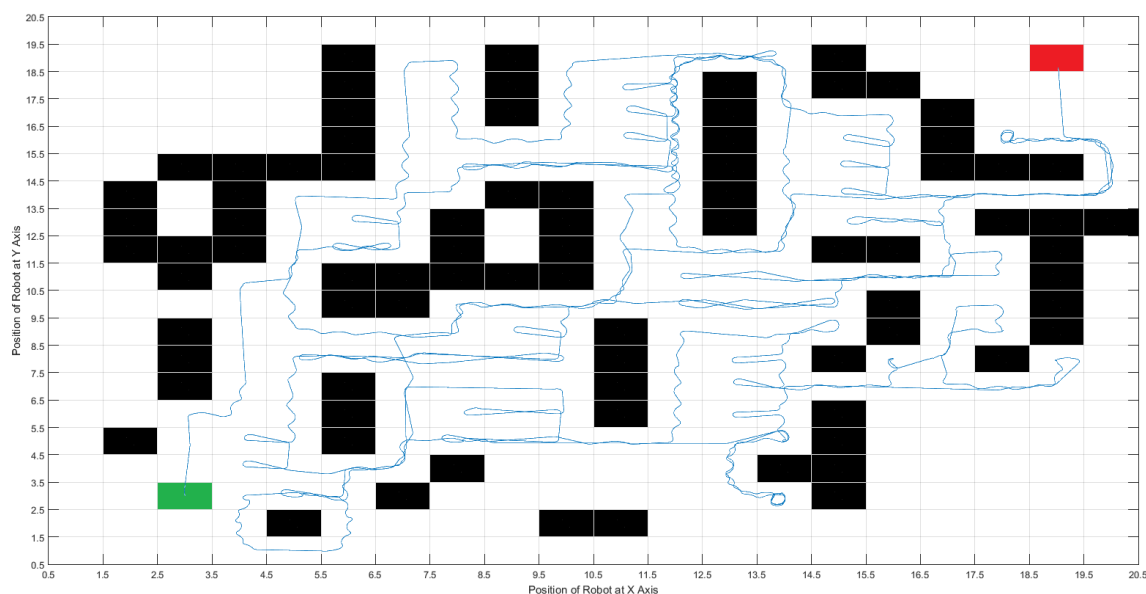
Σχήμα 4.15 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%

Ο χρόνος για την εύρεση της διαδρομής είναι 729.33 δευτερόλεπτα και η απόσταση που διανύθηκε είναι 380,04 μέτρα.



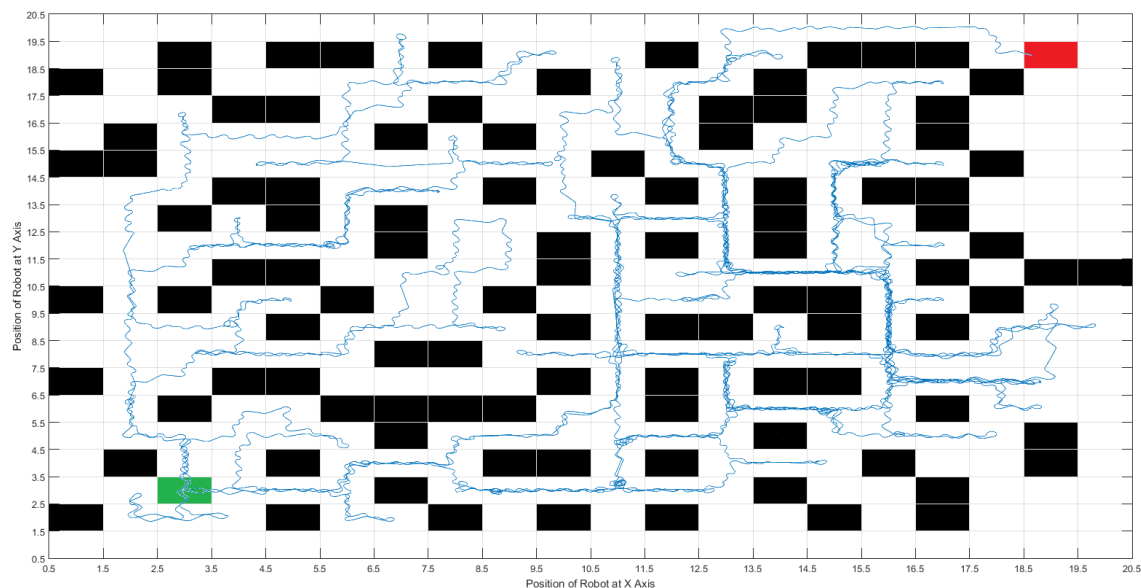
Σχήμα 4.16 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%

Ο χρόνος για την εύρεση της βέλτιστης διαδρομής είναι 415.03 δευτερόλεπτα και διήνυσε 215.79 μέτρα.



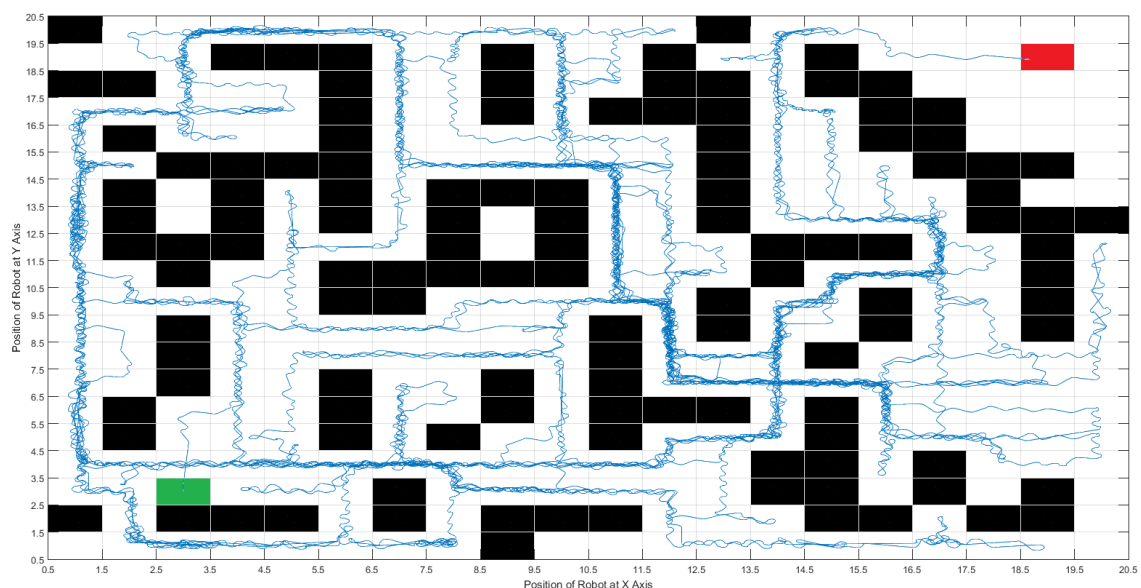
Σχήμα 4.17 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%

Εδώ ο χρόνος ανέρχεται σε 433.78 δευτερόλεπτα και η απόσταση είναι 313.99 μέτρα.



Σχήμα 4.18 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%

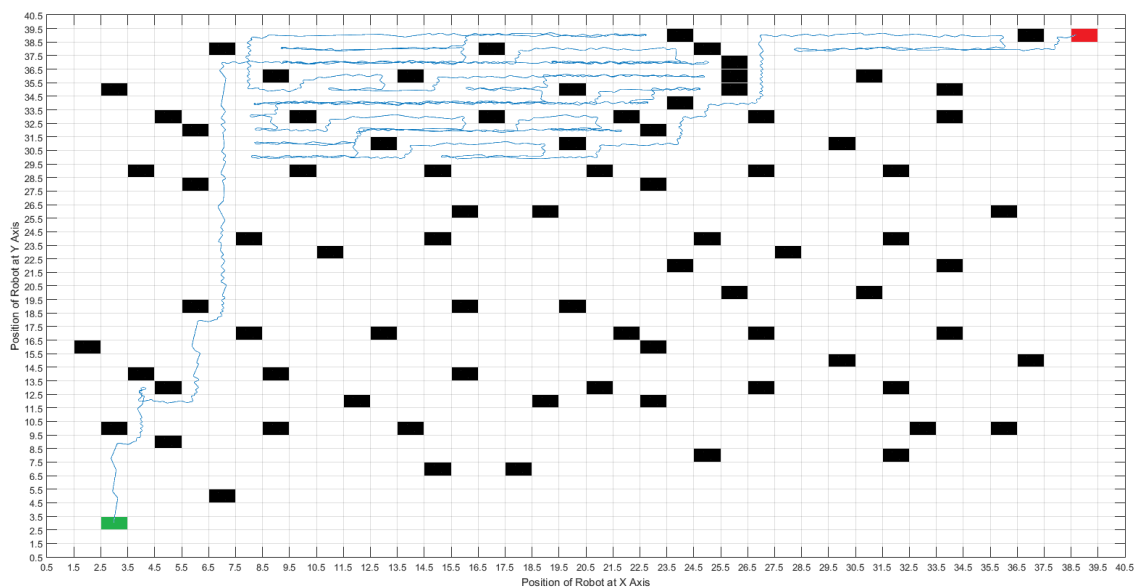
Ο χρόνος εδώ αυξάνεται αρκετά και χρειάζεται 1190.76 δευτερόλεπτα και διήνυσε 558.75 μέτρα.



Σχήμα 4.19 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%

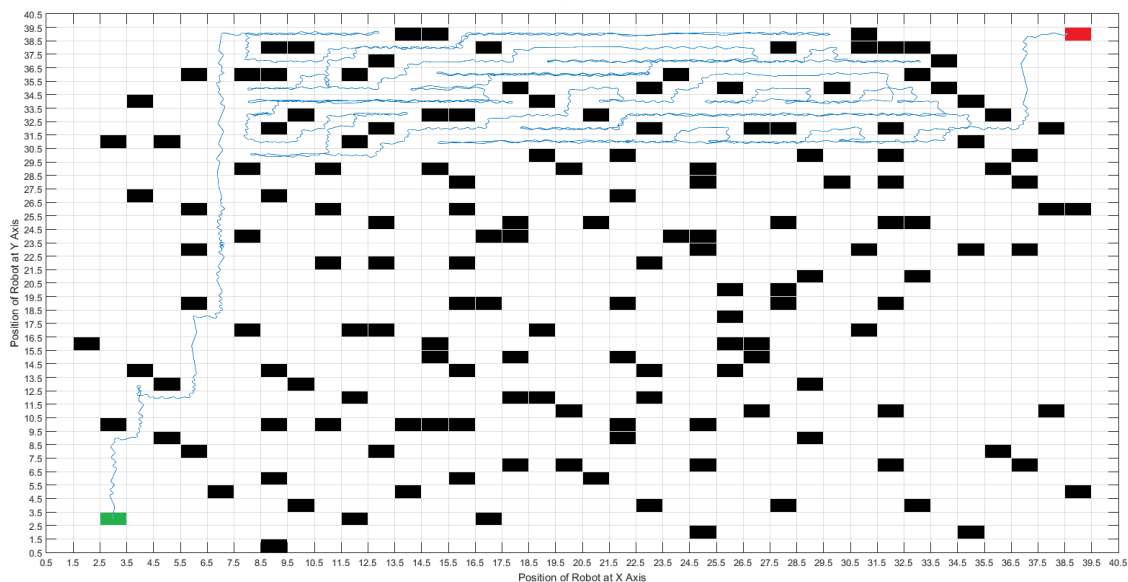
Τέλος, ο χρόνος που χρειάστηκε για την εύρεση της βέλτιστης διαδρομής είναι 2043.89 δευτερόλεπτα και η απόσταση που κάλυψε είναι 1198.90 μέτρα.

Στα σχήματα 4.20 με 4.25 παρουσιάζεται η κίνηση του οχήματος σε αρένες μεγαλύτερων διαστάσεων, μεγέθους 40x40 μέτρων και με βαθμό δυσκολίας παρόμοιο με τις προηγούμενες.



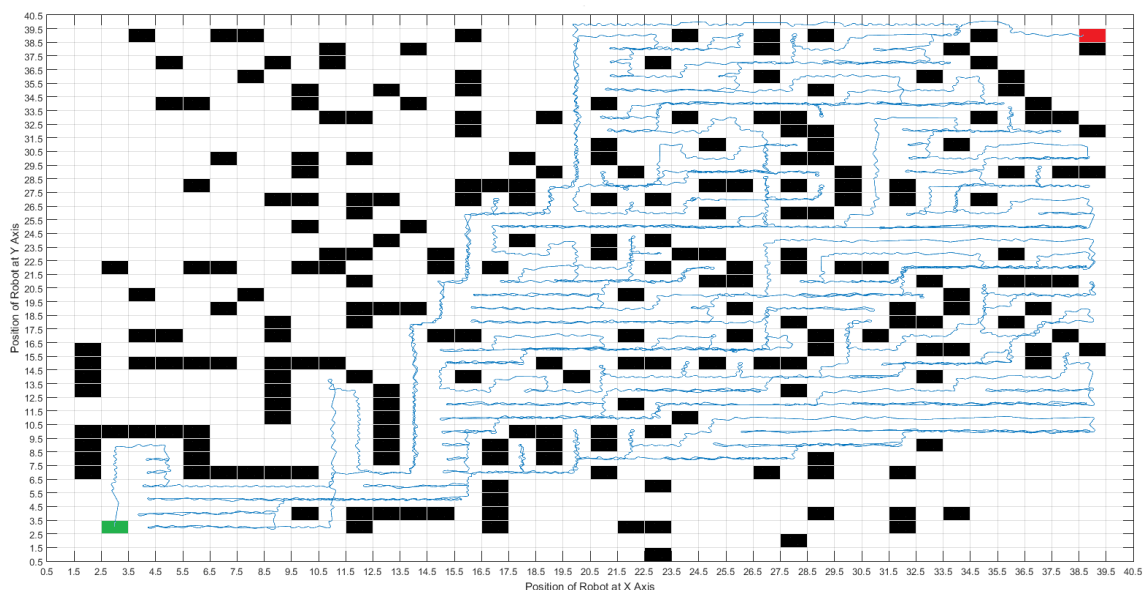
Σχήμα 4.20 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 5%

Ξεκινώντας με την πρώτη αρένα μεγαλύτερων διαστάσεων και με μικρό βαθμό δυσκολίας ο χρόνος για να βρεθεί η βέλτιστη λύση είναι 507.52 δευτερόλεπτα και η απόσταση που κάλυψε είναι 365.46 μέτρα.



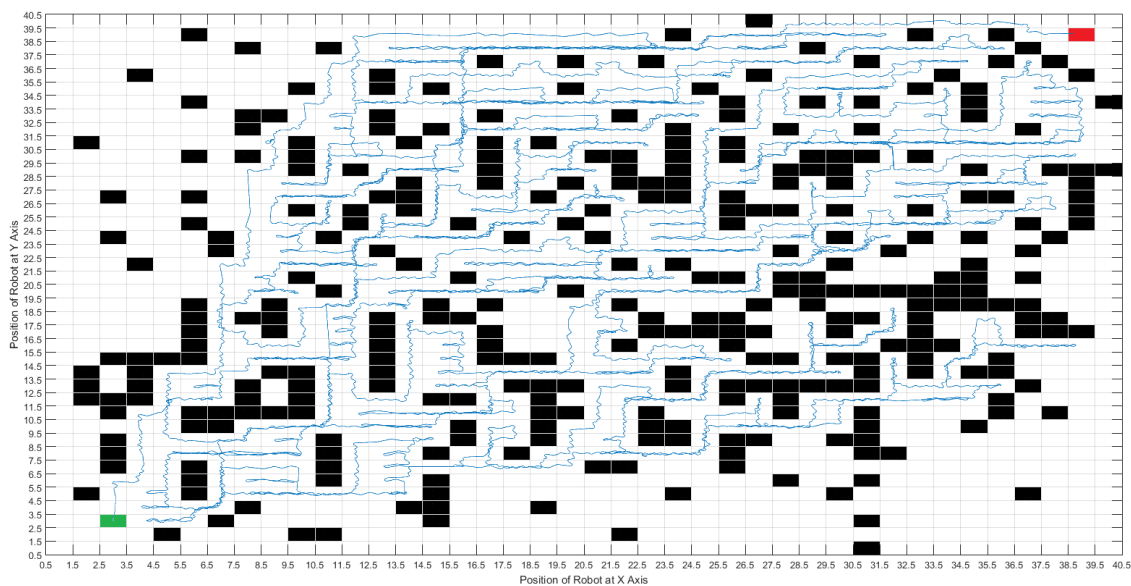
Σχήμα 4.21 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 10%

Ο χρόνος που χρειάστηκε είναι 676.44 δευτερόλεπτα και η απόσταση που διήνυσε είναι 437.12 μέτρα.



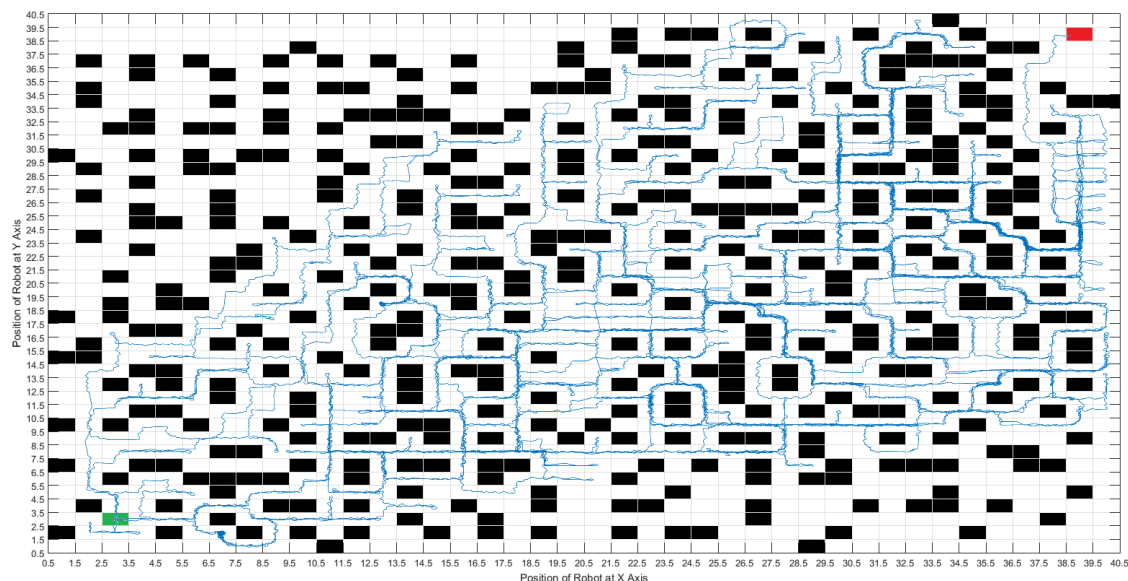
Σχήμα 4.22 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 15%

Ο χρόνος εδώ είναι 2204.68 δευτερόλεπτα και η απόσταση 1300,78 μέτρα.



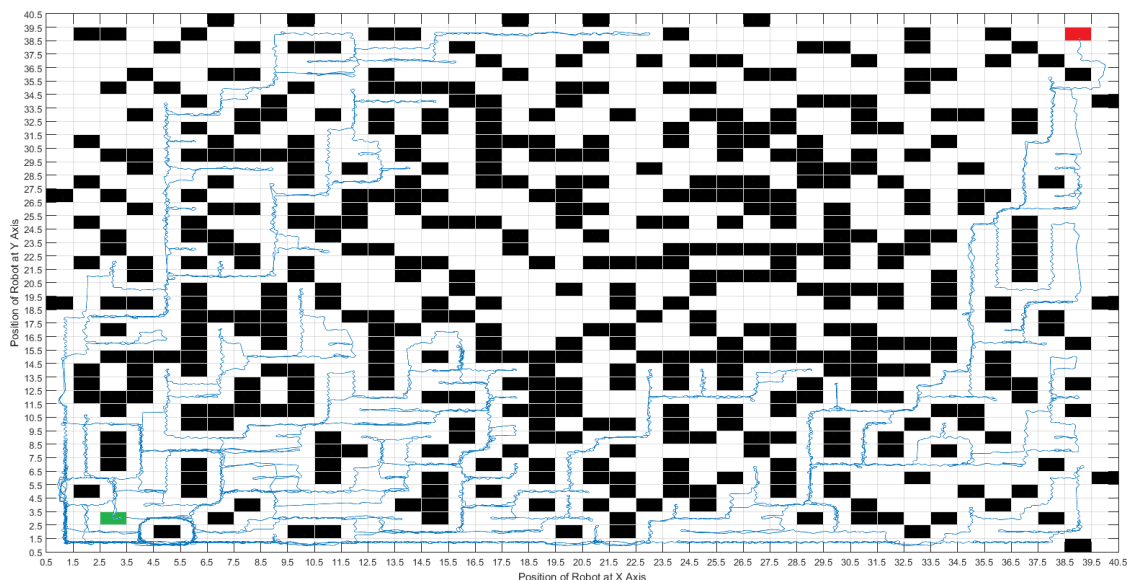
Σχήμα 4.23 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 20%

Ο χρόνος που απαιτήθηκε για την εύρεση της βέλτιστης λύσης είναι 4124.03 δευτερόλεπτα και η απόσταση που διανύθηκε είναι 1359.46 μέτρα.



Σχήμα 4.24 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%

Ο χρόνος και η απόσταση σε αυτή την αρένα αυξάνονται αρκετά για την εύρεση της λύσης και ανέρχονται σε 11818.56 δευτερόλεπτα και 2956.51 μέτρα αντίστοιχα.



Σχήμα 4.25 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%

Στην τελευταία αρένα με τον μεγαλύτερο βαθμό δυσκολίας ο χρόνος που χρειάστηκε είναι 4054.03 δευτερόλεπτα και η απόσταση 1340 μέτρα. Ακολουθώντας αυτές τις διαδρομές, η ταχύτητα του οχήματος μεταβάλλεται ανάλογα με την θέση του στόχου και την πιθανή εύρεση εμποδίου και αποφυγή του. Οι ταχύτητες των δύο τροχών του οχήματος που αναπτύχθηκαν κατά την διάρκεια της προσομοίωσης αναλύονται στα

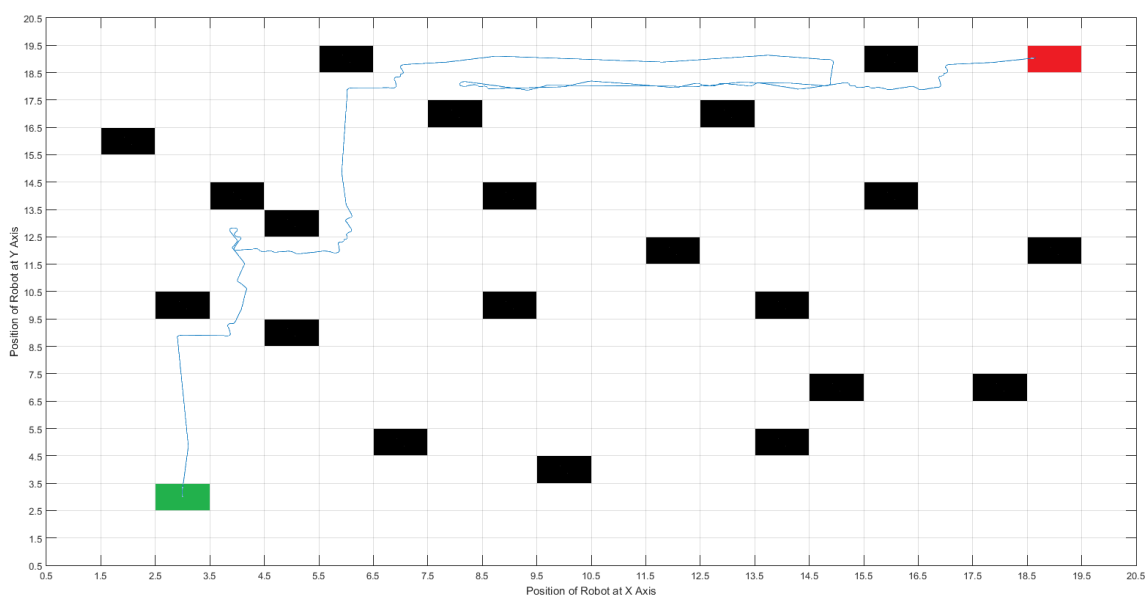
παρακάτω σχήματα. Τέλος, καταγράφονται όλα τα δεδομένα των προσομοιώσεων για τον ευριστικό αλγόριθμο A* στον πίνακα 4.1 και γίνεται ανάλυση αυτών.

<i>Degree of Difficulty</i>	Αλγόριθμος A*				
	<i>Arenas (Dimension)</i>	<i>Time (Seconds)</i>	<i>Distance (Meters)</i>	<i>Expanded Nodes (Cells)</i>	<i>Path Solution (Cells)</i>
5%	20x20	99.6114	50.1359	43	32
	40x40	507.5222	365.4653	228	72
10%	20x20	729.3304	380.0437	226	34
	40x40	1028.2998	437.1210	263	72
15%	20x20	415.0394	215.7916	130	34
	40x40	2204.6825	1300.7822	659	74
20%	20x20	891.2219	320.7817	198	34
	40x40	4124.0329	1349.4612	706	74
25%	20x20	1190.7645	558.7510	200	36
	40x40	11818.5600	2956.5142	739	76
30%	20x20	2043.8902	1198.9065	247	40
	40x40	4054.0350	1340.6445	565	78

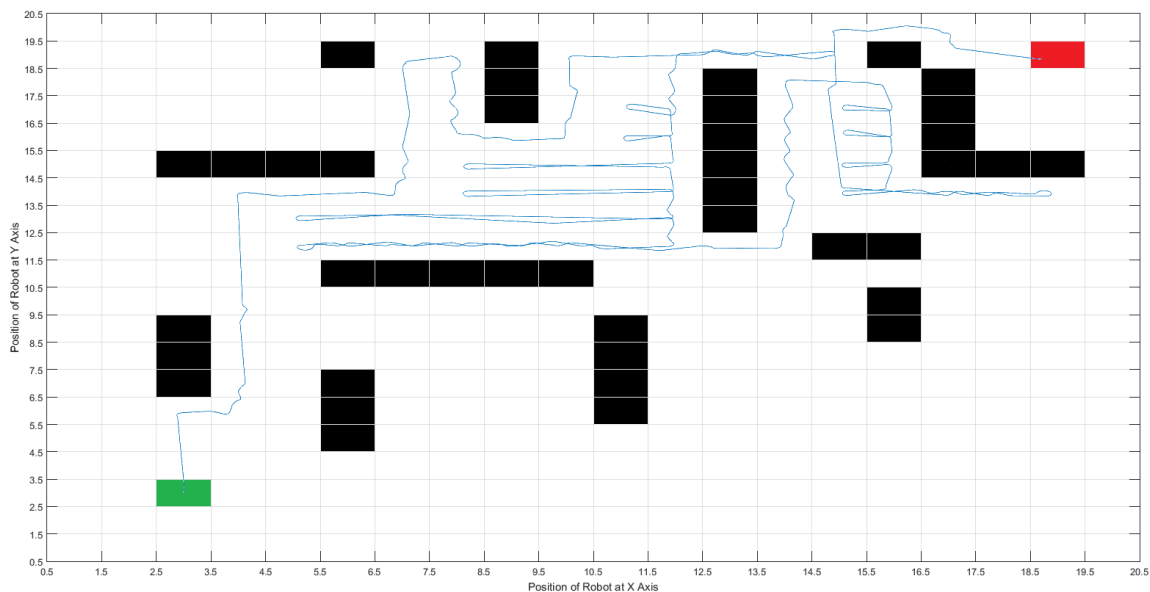
Πίνακας 4.1 Δεδομένα προσομοίωσης αλγορίθμου A*

Στις προσομοιώσεις που πραγματοποιήθηκαν τα δεδομένα όπως φαίνονται στον πίνακα 4.1 μεταβάλλονται καθώς αυξάνεται ο βαθμός δυσκολίας κάθε αρένας, η εύρεση της λύσης απαιτεί περισσότερο χρόνο όπως και η απόσταση την οποία διανύει το ρομποτικό όχημα. Αυτό οφείλεται στο ότι ο αλγόριθμος A* επίσης απαιτεί περισσότερη διερεύνηση των κελιών για να βρει την βέλτιστη λύση. Ακόμα μεγαλύτερη αύξηση παρουσιάζεται στις τιμές σε μεγαλύτερες αρένες, με πλήθος εμποδίων αντίστοιχο των διαστάσεων τους. Οι τιμές των δεδομένων πιθανόν να διαφέρουν καθώς μεταβάλλεται η πολυπλοκότητα των αρένων ως προς τα εμπόδια. Για παράδειγμα στην αρένα 40x40 με βαθμό δυσκολίας 30%, ο χρόνος, η απόσταση και η διερεύνηση των κελιών είναι

Στη συνέχεια στα ίδια σενάρια με προηγουμένως υλοποιήθηκε ο αλγόριθμος CIA* για την σύγκριση με τον κλασσικό αλγόριθμο A* ως προς τις τιμές των δεδομένων που λαμβάνονται. Στα παρακάτω σχήματα παρουσιάζονται οι διαδρομές που ακολούθησε το όχημα με τη χρήση του CIA*.

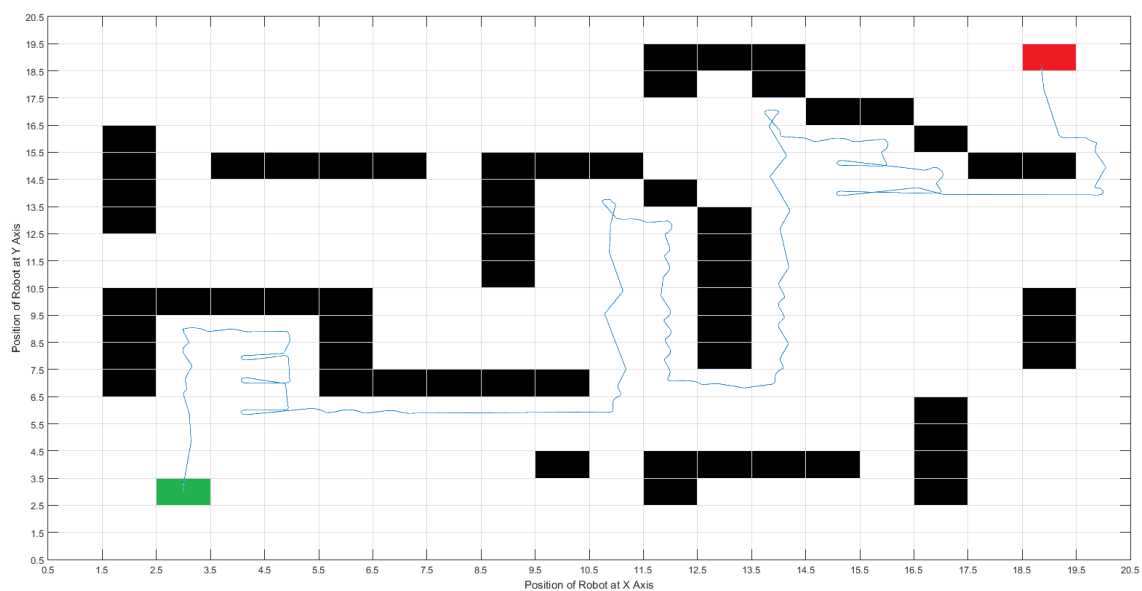


Ο χρόνος που χρειάστηκε για την εύρεση της βέλτιστης διαδρομής είναι 88,75 δευτερόλεπτα και η απόσταση είναι 50,04 μέτρα.



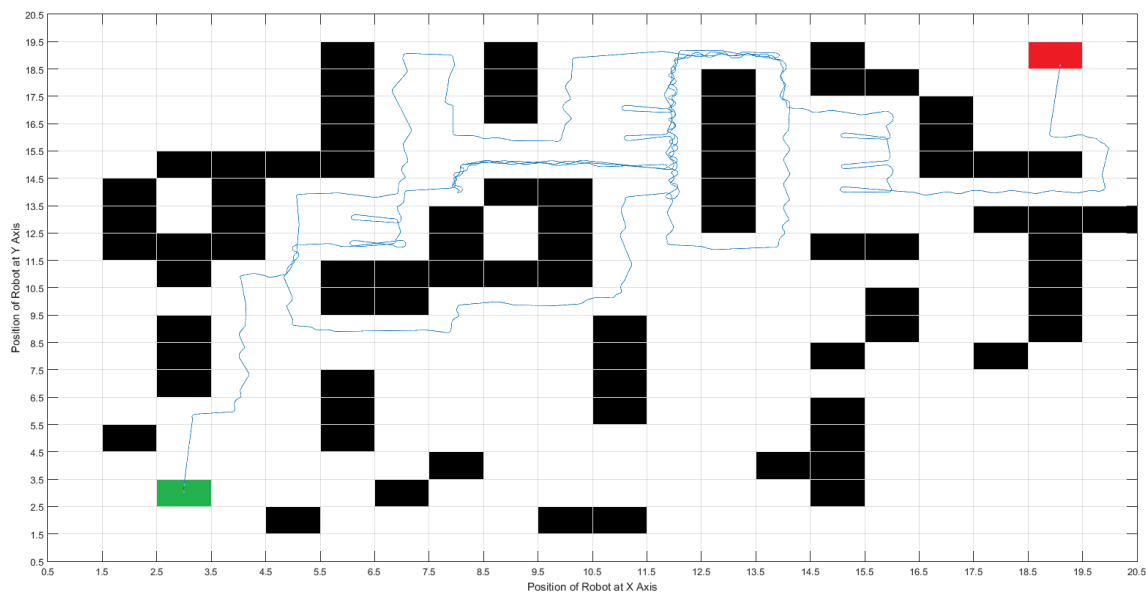
Σχήμα 4.27 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%

Ο χρόνος για το όχημα να φτάσει στον στόχο και να ανιχνεύσει την βέλτιστη διαδρομή με χρήση του αλγορίθμου CIA* είναι 253.58 δευτερόλεπτα και η απόσταση είναι 130.34 μέτρα.



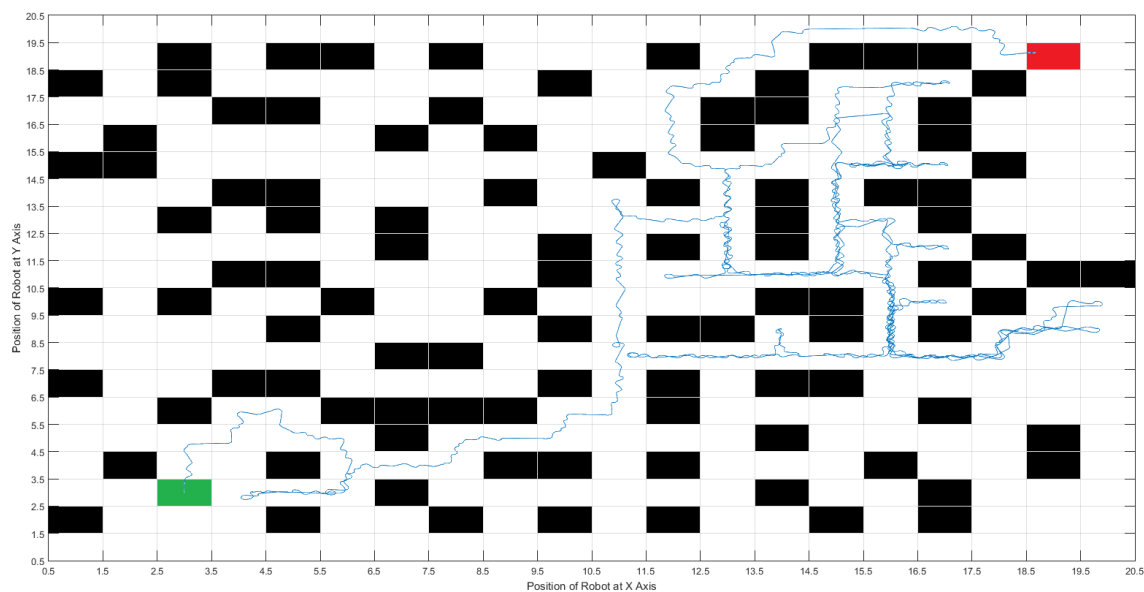
Σχήμα 4.28 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%

Ο χρόνος της διαδρομής είναι 238.79 δευτερόλεπτα και η απόσταση της είναι 71.73 μέτρα.



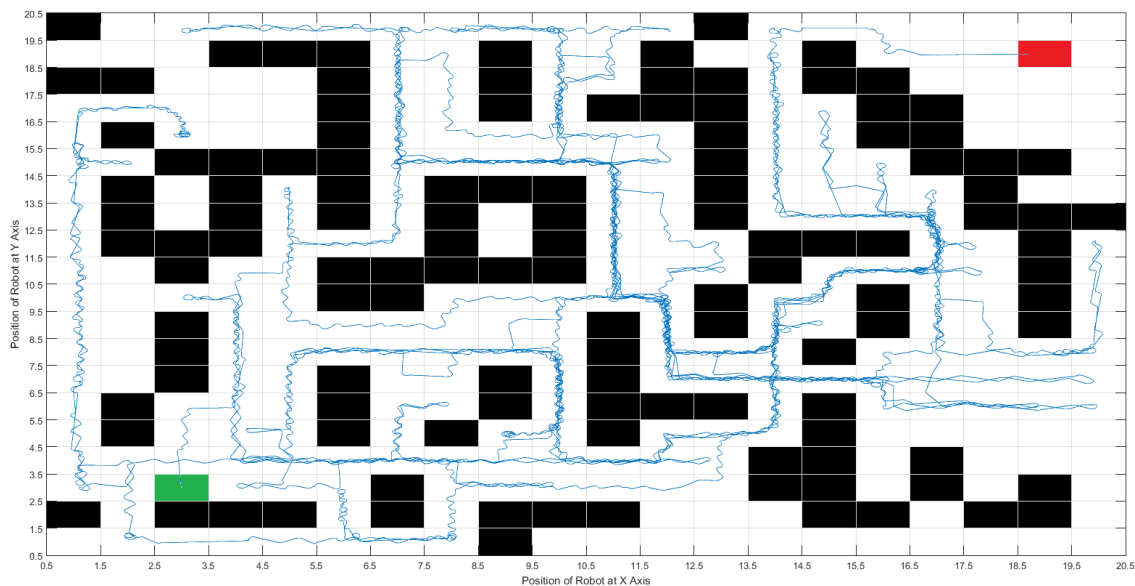
Σχήμα 4.29 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%

Το όχημα χρειάστηκε 386.54 δευτερόλεπτα για να φτάσει στον στόχο και διάνυσε 131.78 μέτρα.



Σχήμα 4.30 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%

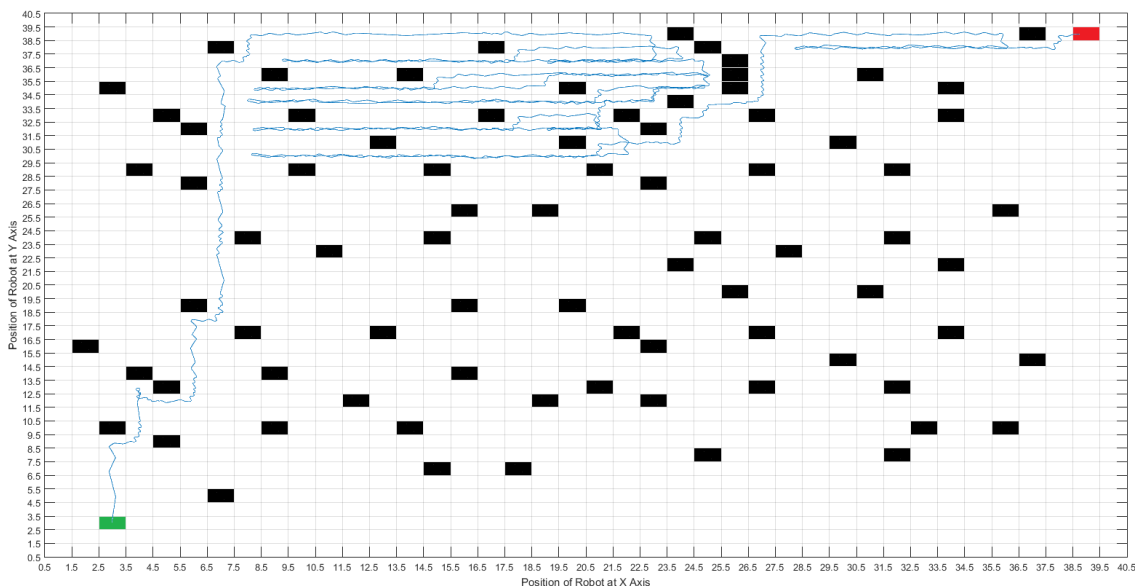
Ο χρόνος στην αρένα αυτή είναι 356.52 δευτερόλεπτα και η απόσταση 177.48 μέτρα.



Σχήμα 4.31 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%

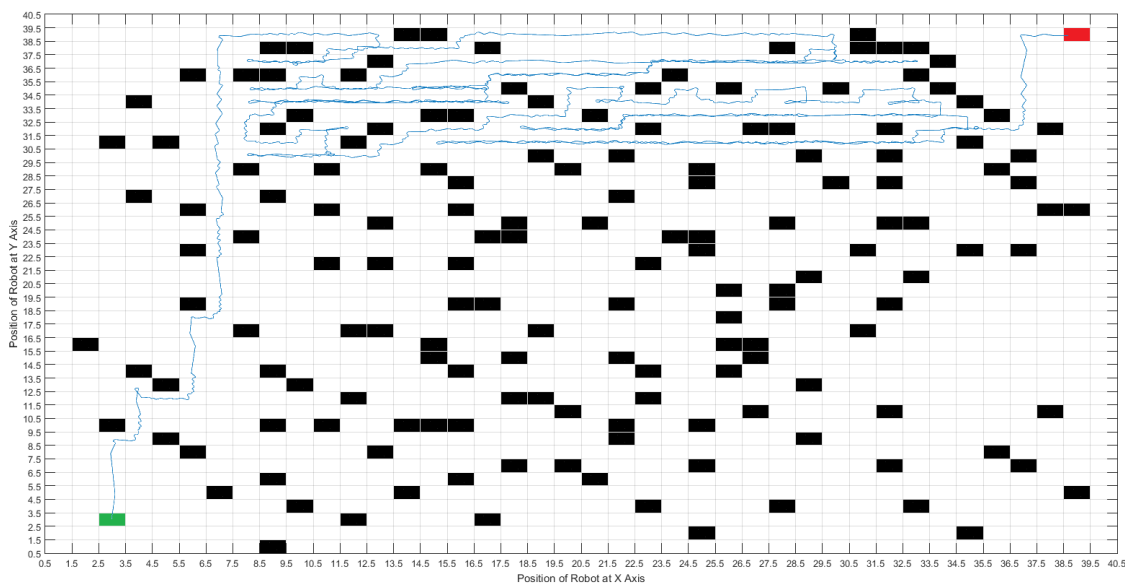
Τέλος, στην αρένα με τα περισσότερα εμπόδια χρειάστηκαν 1161.49 δευτερόλεπτα για να φτάσει στον στόχο το όχημα και διάνυσε 702.74 μέτρα.

Στη συνέχεια αυξάνονται οι διαστάσεις των αρένων ώστε να μελετηθεί ο αλγόριθμος σε μεγαλύτερες διαστάσεις με περισσότερα εμπόδια. Παρουσιάζονται στα παρακάτω σχήματα αρένες διαστάσεων 40x40.



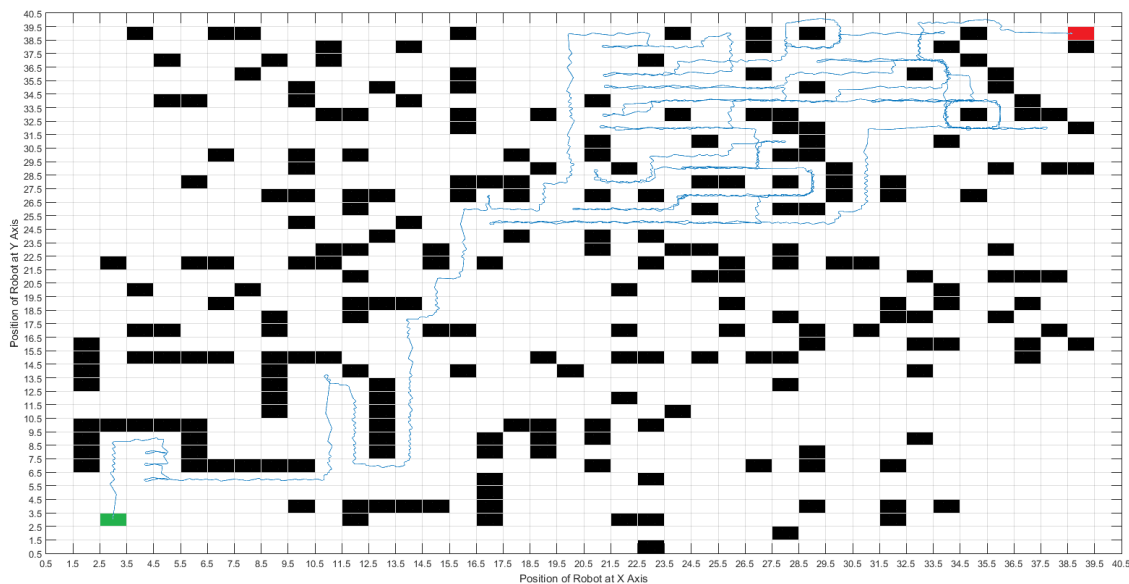
Σχήμα 4.32 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 5%

Στην πρώτη αρένα ο χρόνος για την εύρεση βέλτιστης διαδρομής είναι 653.48 δευτερόλεπτα και η απόσταση που διάνυσε είναι 309.66 μέτρα.



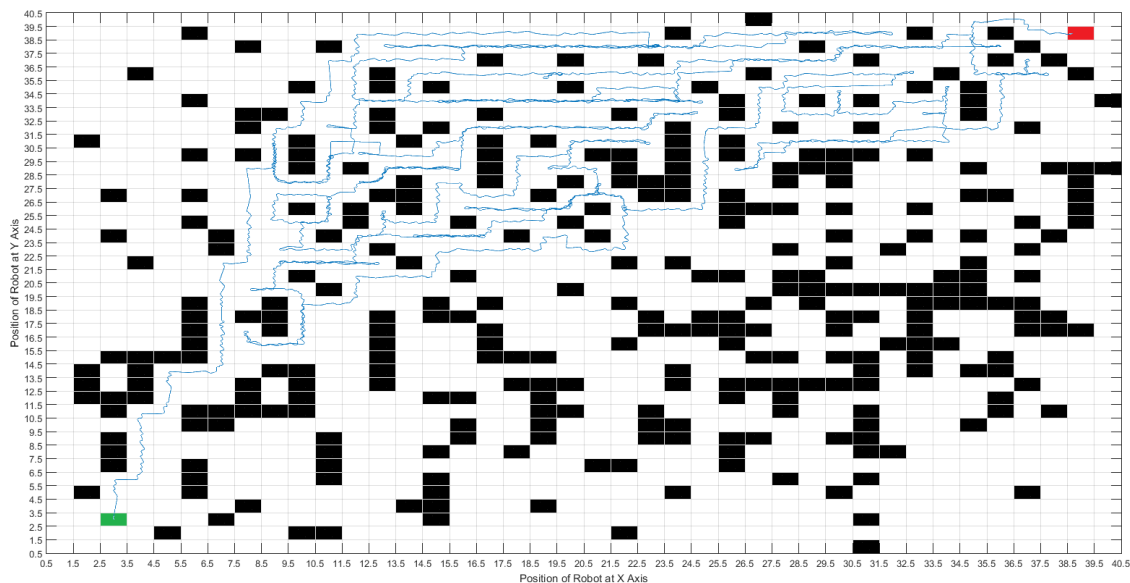
Σχήμα 4.33 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 10%

Ο χρόνος εδώ είναι 840.92 δευτερόλεπτα και η απόσταση 360.93 μέτρα.



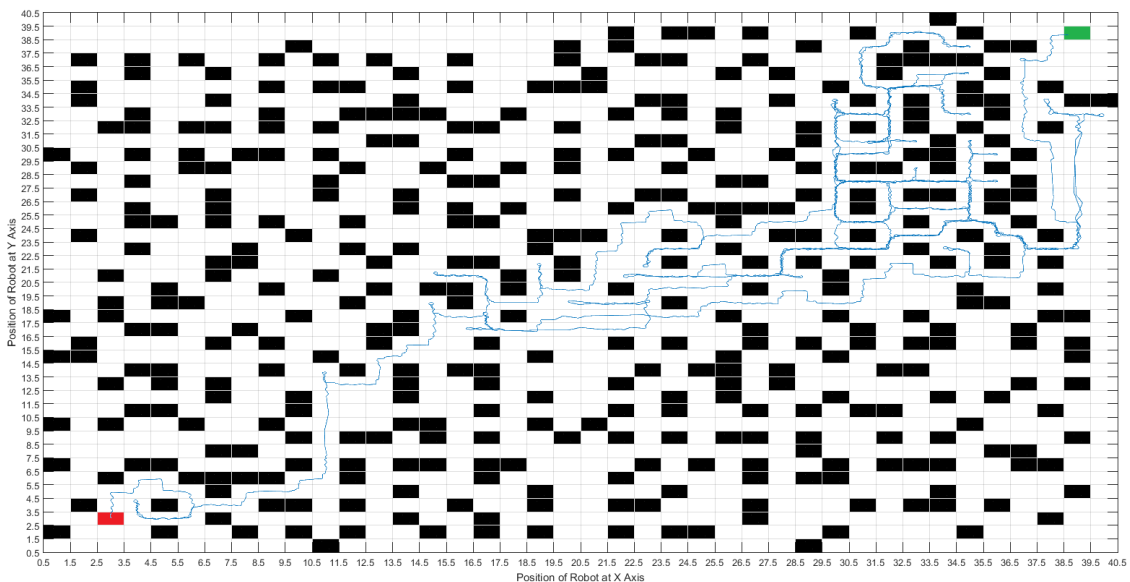
Σχήμα 4.34 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 15%

Ο χρόνος που χρειάστηκε είναι 967.32 δευτερόλεπτα και η απόσταση 377.40 μέτρα.



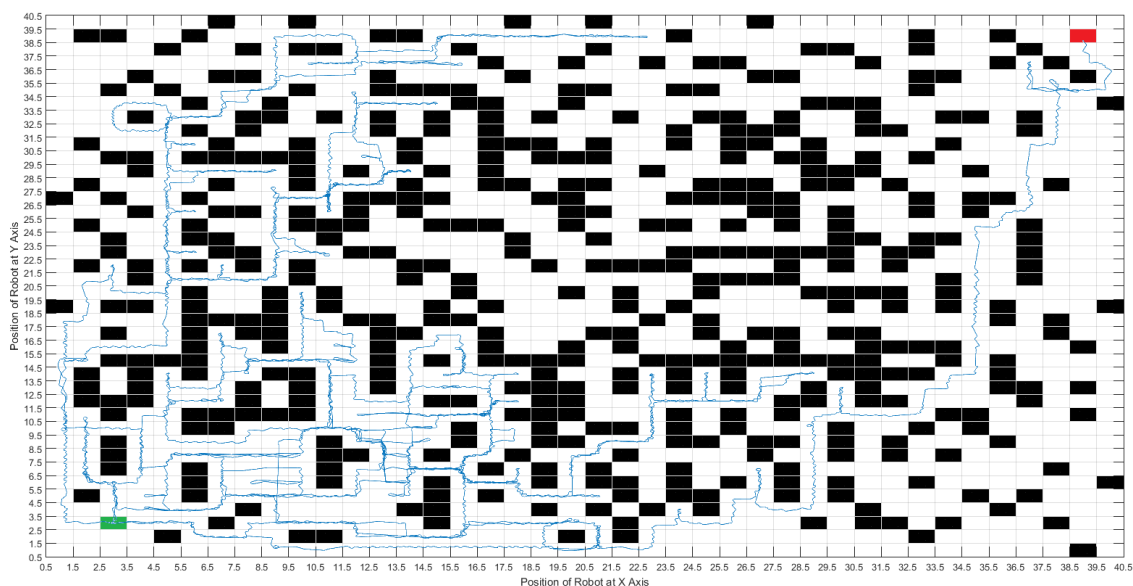
Σχήμα 4.35 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 20%

Ο χρόνος που χρειάστηκε είναι 1055.40 δευτερόλεπτα και η απόσταση 537.64 μέτρα.



Σχήμα 4.36 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%

Αυξάνοντας και άλλο το πλήθος των εμποδίων ο χρόνος που απαιτείται είναι 1573.79 δευτερόλεπτα και η απόσταση 442.16 μέτρα.



Σχήμα 4.37 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%

Τέλος, η αρένα με τα περισσότερα εμπόδια, το όχημα χρειάζεται 3161.91 δευτερόλεπτα για να φτάσει στον στόχο και να βρει την βέλτιστη διαδρομή, καθώς η απόσταση που διάνυσε το όχημα είναι 832.69 μέτρα.

Στον πίνακα 4.2 παρουσιάζονται τα δεδομένα που λαμβάνονται από την κίνηση του οχήματος για την εύρεση της βέλτιστης διαδρομής καθώς και μία σύγκριση μεταξύ αυτών των δεδομένων.

Όπως παρουσιάζεται στον πίνακα 4.2, οι τιμές των δεδομένων μεταβάλλονται με την αύξηση του πλήθους των εμποδίων καθώς και των διαστάσεων των αρένων όπως και στον κλασσικό αλγόριθμο A*. Όπως αναφέρθηκε και προηγουμένως οι τιμές των διευρυμένων κελιών και της λύσης την βέλτιστης διαδρομής εξαρτώνται από την πολυπλοκότητα του χώρου ως προς το πλήθος και τη θέση των εμποδίων. Στις προσομοιώσεις που πραγματοποιήθηκαν μεγαλύτερη διερεύνηση του χώρου παρουσιάζεται στον κόσμο διαστάσεων 40x40 με βαθμός δυσκολίας 30%, την αρένα με τα περισσότερα εμπόδια.

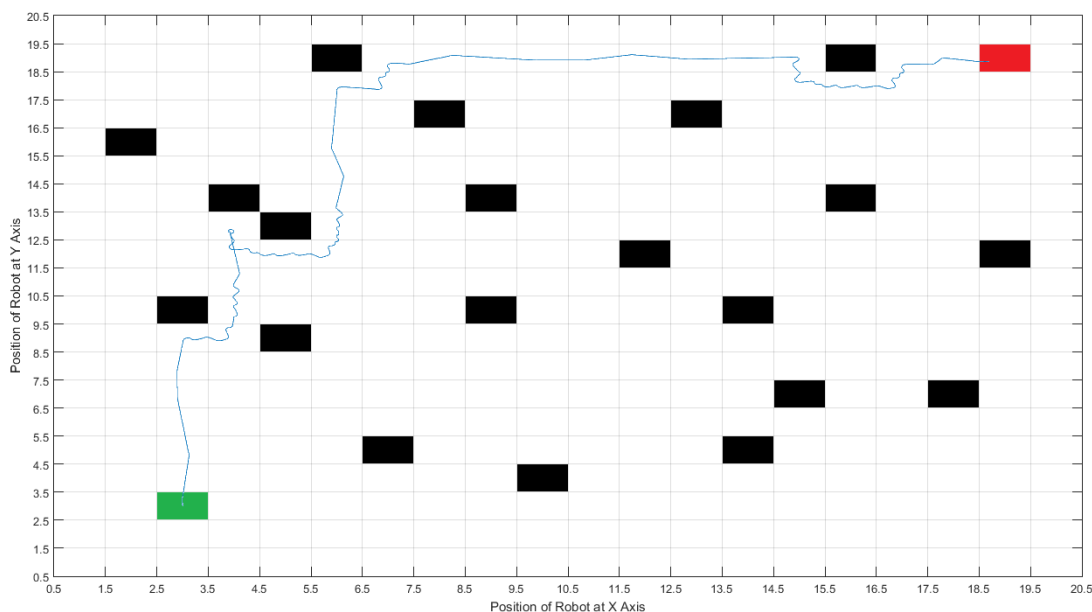
<i>Degree of Difficulty</i>	<i>Arenas (Dimension)</i>	Αλγόριθμος CIA*			
		<i>Time (Seconds)</i>	<i>Distance (Meters)</i>	<i>Expanded Nodes (Cells)</i>	<i>Path Solution (Cells)</i>
5%	20x20	88.7575	50.0420	43	32
	40x40	653.4880	309.6691	190	72
10%	20x20	253.5846	130.3457	92	34
	40x40	840.9225	360.9367	221	72
15%	20x20	238.7933	71.7348	65	34
	40x40	967.3234	377.4094	219	74
20%	20x20	386.5406	131.7803	85	34
	40x40	1055.4075	537.6484	304	74
25%	20x20	356.5200	177.4869	85	36
	40x40	1573.7991	442.1684	205	76
30%	20x20	1161.4909	702.7458	191	40
	40x40	3161.9170	832.6980	385	78

Πίνακα 4.2 Δεδομένα προσομοίωσης αλγορίθμου CIA*

4.4.3 Αποτελέσματα προσομοίωσης με χρήση του αλγορίθμου Weighted A*

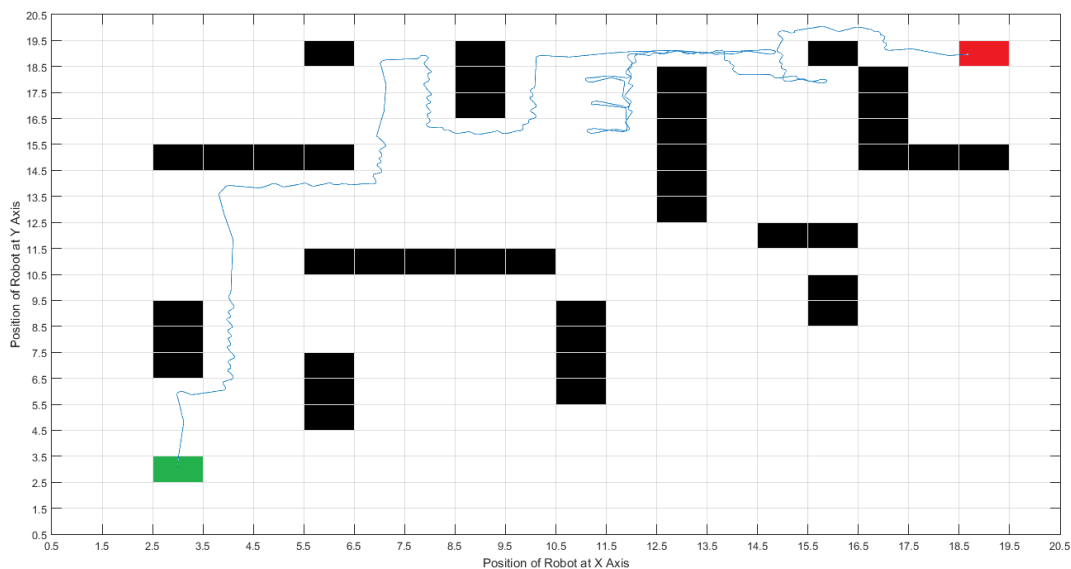
Τέλος, χρησιμοποιήθηκε ο αλγόριθμος ευριστικής αναζήτησης Weighted A* στο ρομποτικό όχημα, με τις ίδιες αρένες και βαθμό δυσκολίας όπως στους δύο προηγούμενους αλγορίθμους για την σύγκριση των αποτελεσμάτων τους. Σκοπός της χρήσης και της υλοποίησης του αλγορίθμου Weighted A* είναι η σύγκριση του με τον CIA*. Για την σύγκριση αυτή αναγκαίο ήταν η κατάλληλη απόδοση βάρους ώστε ο χρόνος ο οποίος απαιτείται για την εύρεση της βέλτιστης λύσης να είναι ίδιος ή μικρότερος από αυτόν που χρειάζεται ο αλγόριθμος CIA*. Έτσι επιλέχθηκε τιμή βάρους 1.3 με την οποία ο Weighted A* βρίσκει λύση σε μικρότερο χρόνο από αυτή του CIA* με τη διαφορά ότι η λύση που εξάγεται πιθανόν να μην είναι βέλτιστη. Αυτό συμβαίνει επειδή ο αλγόριθμος "θυσιάζει" περιοχή από την λύση ώστε να μειωθεί σημαντικά ο

χρόνος εύρεσής της. Στα παρακάτω σχήματα παρουσιάζεται ο χώρος και η διαδρομή που πραγματοποίησε το όχημα για την εύρεση της βέλτιστης διαδρομής.



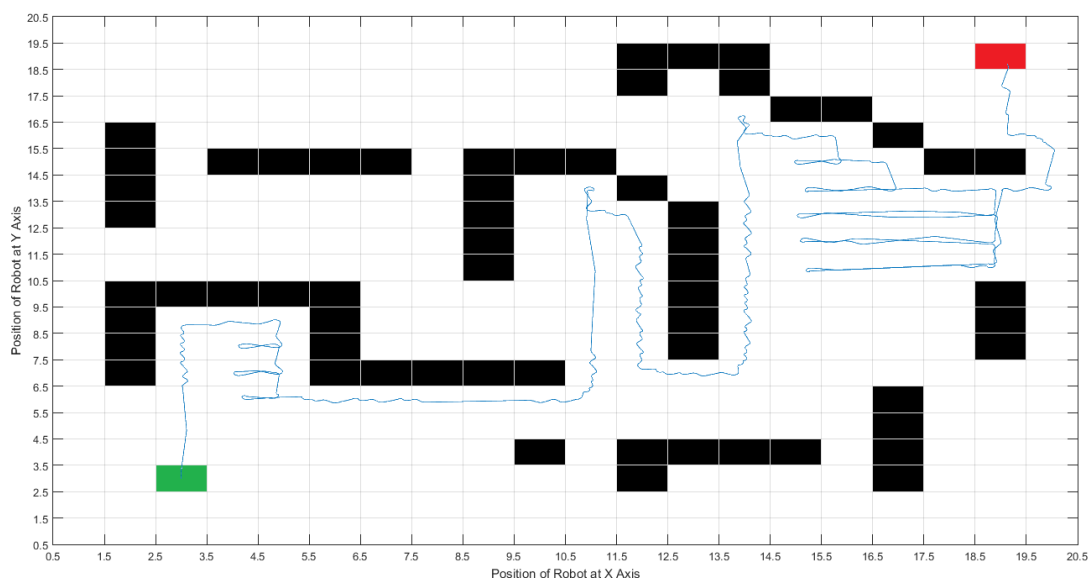
Σχήμα 4.38 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%

Ο χρόνος για να καταφέρει το όχημα να βρει την βέλτιστη λύση και να φτάσει στον στόχο είναι 68.51 δευτερόλεπτα και η απόσταση που διάνυσε είναι 36.33 μέτρα.



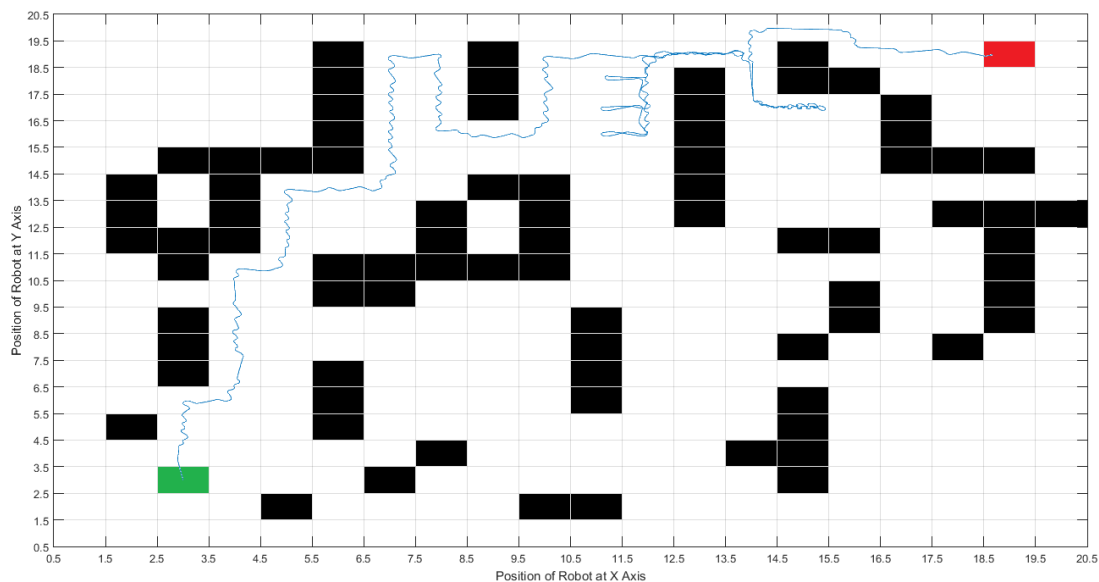
Σχήμα 4.39 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%

Ο χρόνος που χρειάστηκε το όχημα είναι 143.34 δευτερόλεπτα και η απόσταση που διάνυσε 61.89 μέτρα.



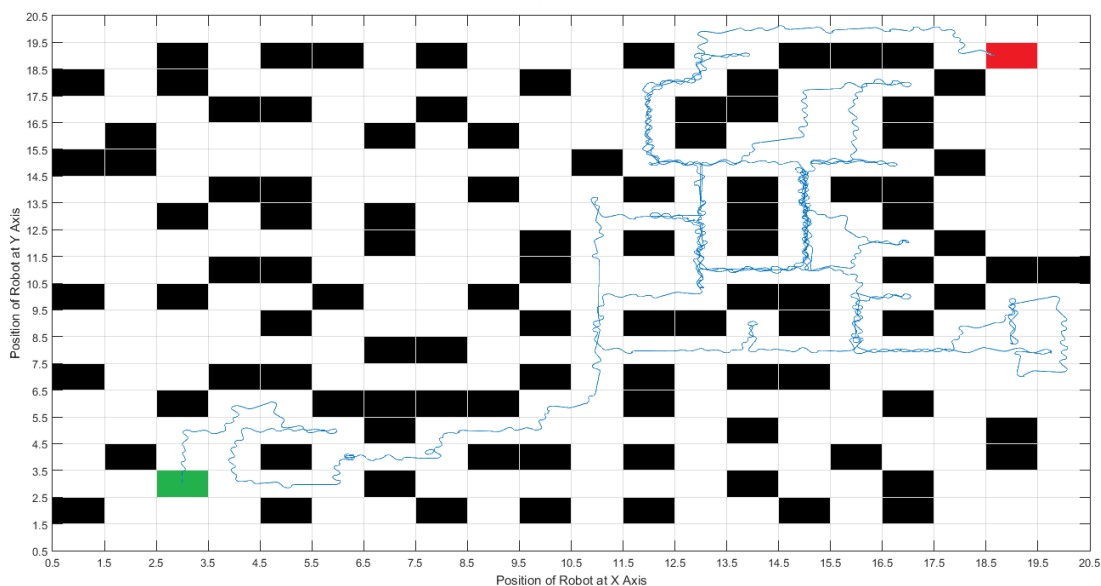
Σχήμα 4.40 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%

Εδώ ο χρόνος είναι 221.41 δευτερόλεπτα και η απόσταση 101.36 μέτρα.



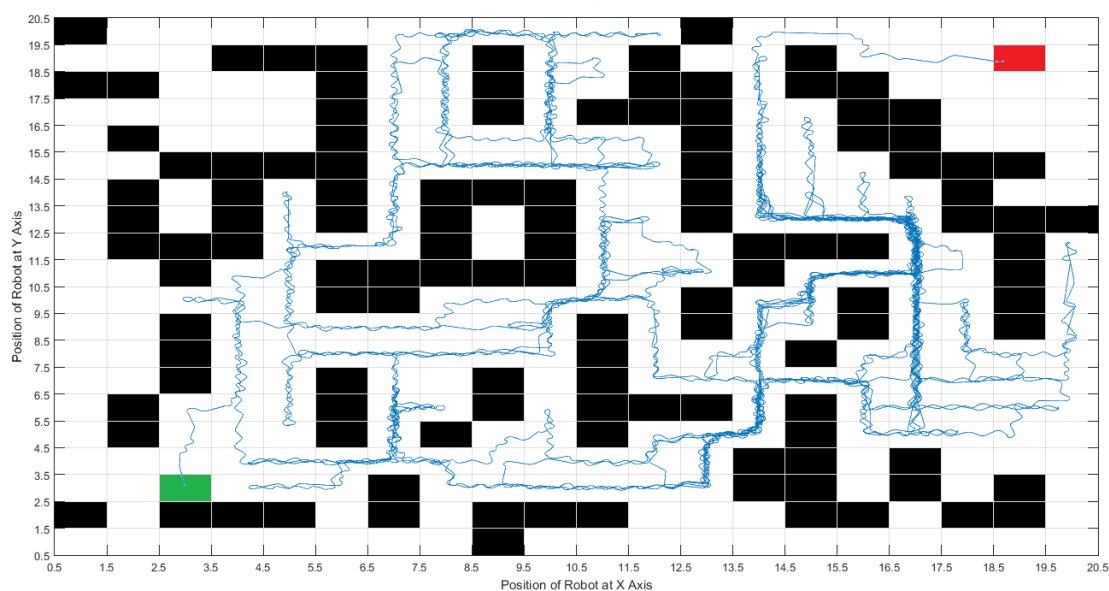
Σχήμα 4.41 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%

Το όχημα χρειάστηκε 170.53 δευτερόλεπτα και διάνυσε 66.83 μέτρα για να φτάσει στον στόχο και να βρει τη βέλτιστη λύση.



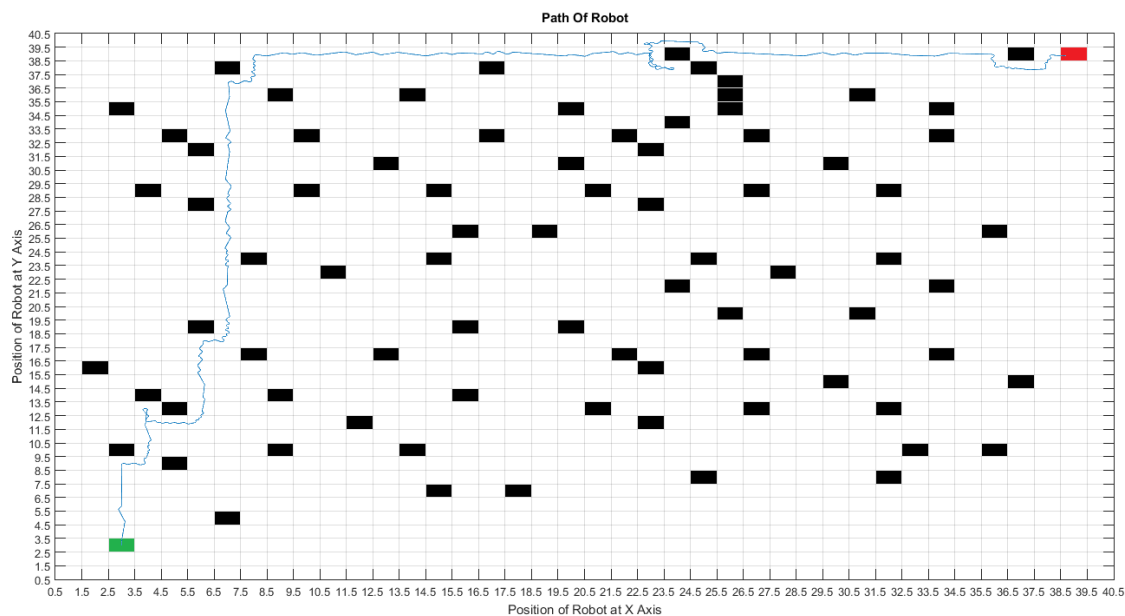
Σχήμα 4.42 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%

Σε αυτό το επίπεδο για την εύρεση της βέλτιστης λύσης απαιτήθηκαν 349.69 δευτερόλεπτα και το όχημα κάλυψε 189.94 μέτρα.



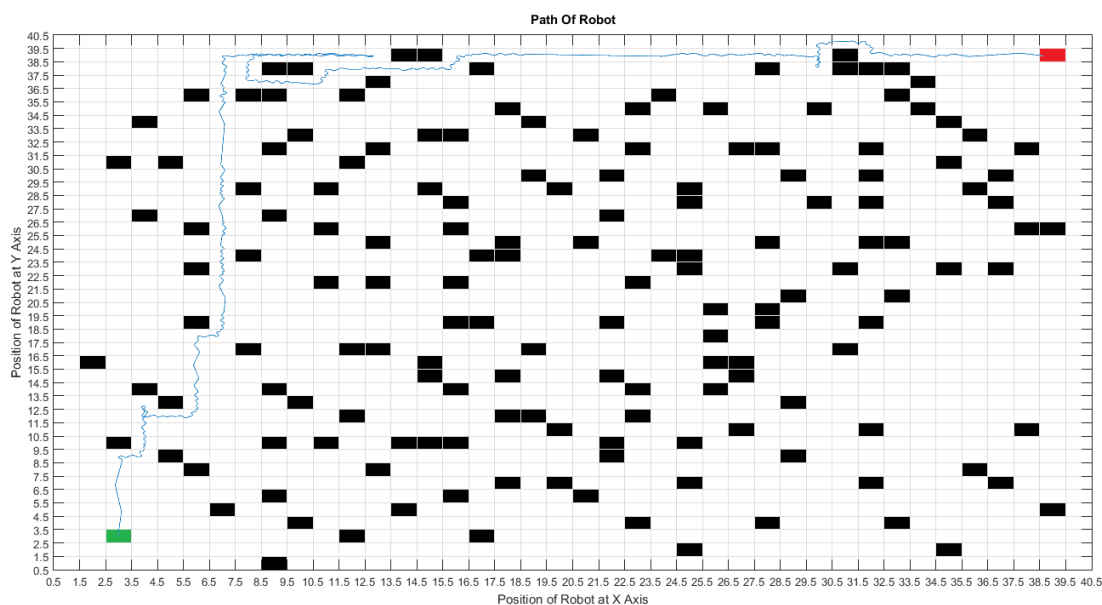
Σχήμα 4.43 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%

Στο τελευταίο βαθμό δυσκολίας για τις αρένες διαστάσεων 20x20 ο χρόνος είναι 1186.13 δευτερόλεπτα και η απόσταση την οποία διάνυσε το όχημα είναι 745.99 μέτρα.



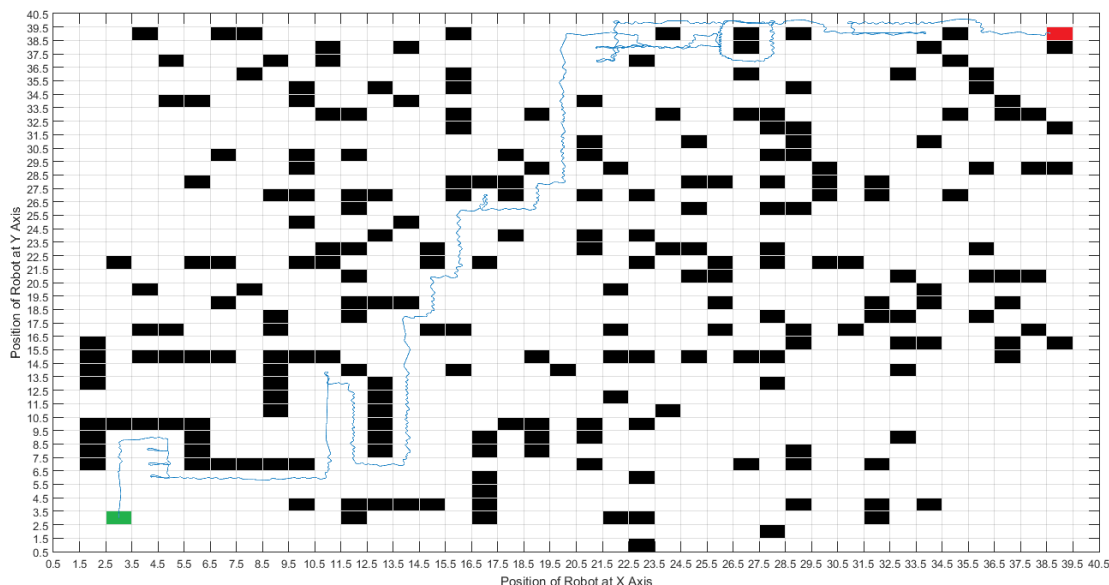
Σχήμα 4.44 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%

Αυξάνοντας τις διαστάσεις της αρένας ο χρόνος για την βέλτιστη λύση είναι 184.81 δευτερόλεπτα και η απόσταση 83.32 μέτρα.



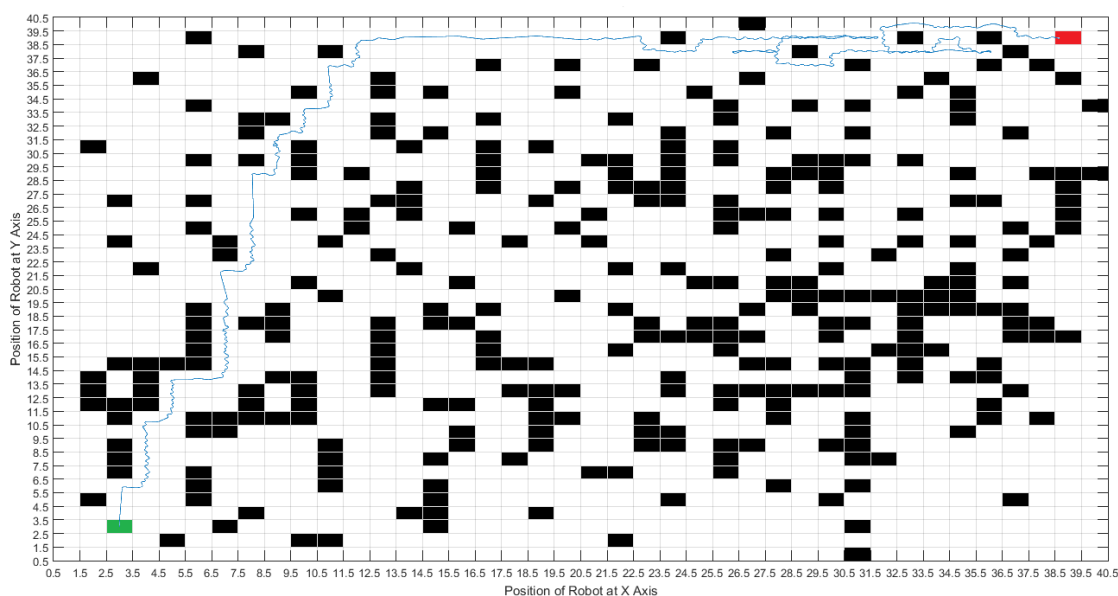
Σχήμα 4.45 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%

Ο χρόνος που χρειάστηκε είναι 153.38 δευτερόλεπτα και η απόσταση 99.12 μέτρα.



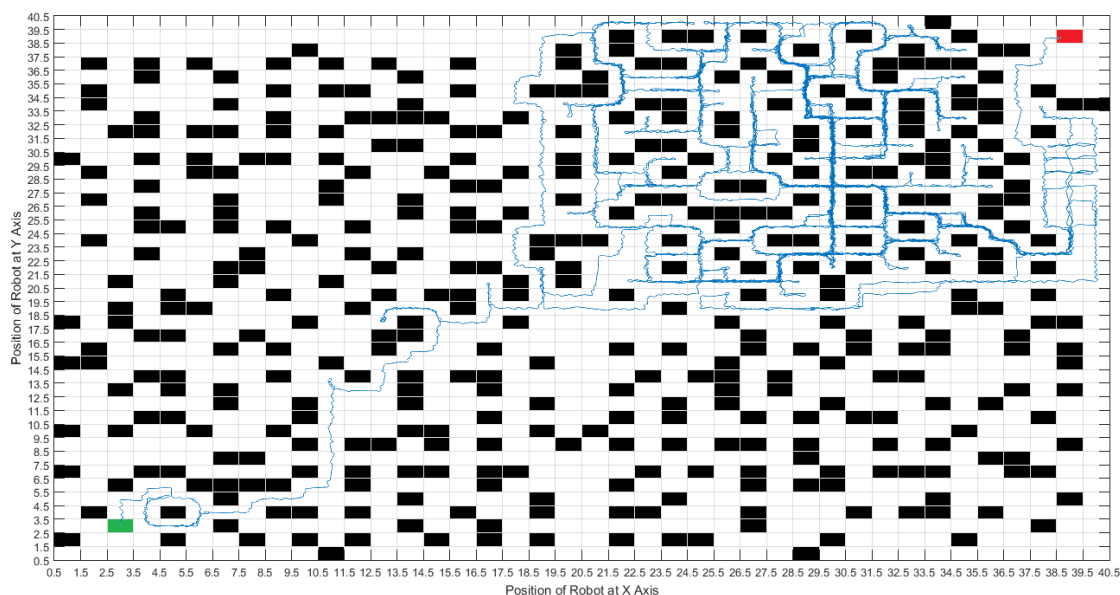
Σχήμα 4.46 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%

Ο χρόνος για την εύρεση της λύσης και την άφιξη του οχήματος στον τελικό στόχο είναι 277 δευτερόλεπτα και η απόσταση 151.32 μέτρα.



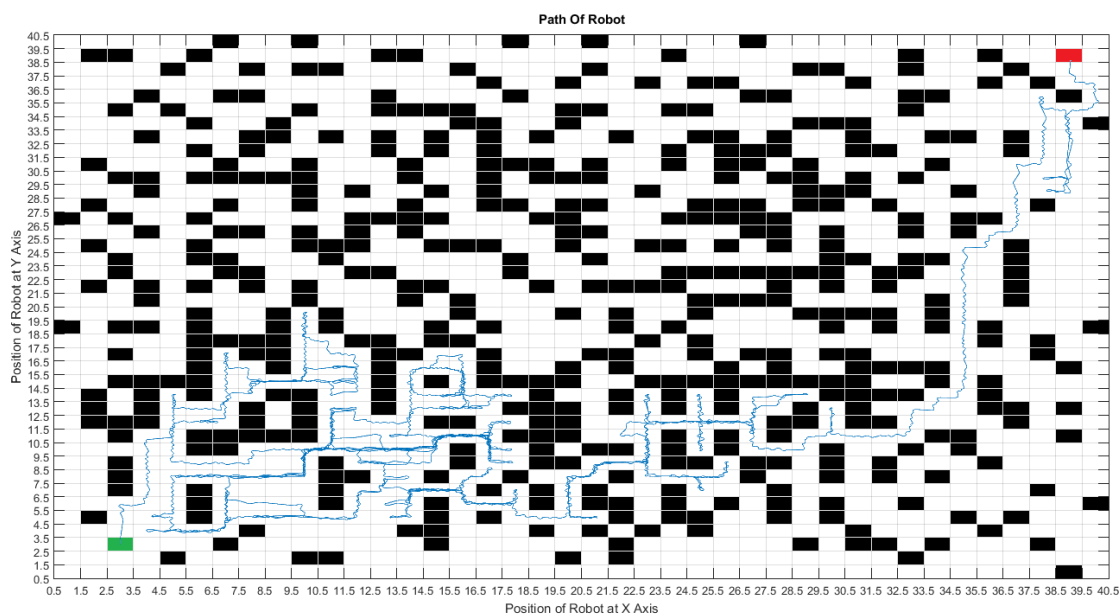
Σχήμα 4.47 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%

Ο χρόνος που απαιτείται στην αρένα αυτή είναι 194.79 δευτερόλεπτα και η απόσταση 109.27 μέτρα.



Σχήμα 4.48 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%

Το όχημα χρειάζεται 3557.10 δευτερόλεπτα για την εύρεση της βέλτιστης λύσης και η απόσταση που διανύει είναι 1391 μέτρα.



Σχήμα 4.49 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%

Στην τελευταία αρένα το όχημα χρειάζεται 1008.68 δευτερόλεπτα για να φτάσει στον τελικό στόχο και κάλυψε απόσταση 515.34 μέτρων.

Παρουσιάζεται παρακάτω ο πίνακας στον οποίο καταγράφονται όλα τα δεδομένα των προσομοιώσεων για κάθε σενάριο ώστε να γίνει σύγκριση αυτών με τους προηγούμενους αλγορίθμους.

		Αλγόριθμος Weighted A*			
<i>Degree of Difficulty</i>	<i>Arenas (Dimension)</i>	<i>Time (Seconds)</i>	<i>Distance (Meters)</i>	<i>Expanded Nodes (Cells)</i>	<i>Path Solution (Cells)</i>
5%	20x20	68.5171	36.3360	36	34
	40x40	184.8137	83.3276	80	76
10%	20x20	143.3424	61.8957	49	34
	40x40	153.3801	99.1296	86	74
15%	20x20	221.4109	101.3639	80	34
	40x40	277.0056	151.3291	111	74
20%	20x20	170.5362	66.8392	48	34
	40x40	194.7979	109.2764	87	76
25%	20x20	349.6965	189.9450	91	36
	40x40	3557.1092	1391.0112	313	82
30%	20x20	1186.1305	745.9919	181	40
	40x40	1008.6864	515.3425	233	82

Πίνακας 4.3 Δεδομένα προσομοίωσης αλγορίθμου Weighted A*

Όμοια με τους αλγορίθμους A* και CIA* φαίνεται πως καθώς οι διαστάσεις και ο βαθμός δυσκολίας αυξάνονται, οι τιμές των δεδομένων αυξάνονται με τη σειρά τους. Τέτοιου είδους μεταβολές οφείλονται ακόμα και στην πολυπλοκότητα όπως έχει αναφερθεί. Για παράδειγμα στις αρένες διαστάσεων 40x40 με βαθμό δυσκολίας 15-20% και 25-30% ενώ το πλήθος εμποδίων αυξάνεται, κάτι το οποίο κάνει πιο δύσκολη την εύρεση της λύσης αλλά και της κίνησης του οχήματος, παρουσιάζεται μείωση στα διευρυμένα κελιά που οδηγεί σε μείωση στον χρόνο της προσομοίωσης και στην απόσταση που διανύθηκε.

4.4.4 Σύγκριση δεδομένων για τη περίπτωση χρήσης του απλού ελεγκτή

Παρουσιάζεται παρακάτω ένας συγκεντρωτικός πίνακας στον οποίο περιέχονται ο χρόνος και η απόσταση που χρειάστηκαν για το ρομποτικό όχημα για όλους τους αλγορίθμους που υλοποιήθηκαν, ώστε να φτάσει στο τελικό σημείο με στόχο την εύρεση της βέλτιστης διαδρομής.

Η υλοποίηση των τριών αυτών αλγορίθμων έχει ως στόχο την σύγκρισή μεταξύ των αποδόσεων τους στην εύρεση της βέλτιστης λύσης (χρόνος, απόσταση, διευρυμένα κελιά) και της συμπεριφοράς του ρομποτικού οχήματος κατά την ανίχνευσή της. Αρχικά έγινε σύγκριση μεταξύ του κλασσικού αλγορίθμου A^* και του αλγορίθμου CIA^* όπου και οι δύο πραγματοποιούν εύρεση της βέλτιστης λύσης. Βάση των δεδομένων που ανακτήθηκαν από τις προσομοιώσεις που πραγματοποιήθηκαν, στο μεγαλύτερο ποσοστό ο αλγόριθμος CIA^* είναι εμφανώς πιο αποδοτικός ως προς τα κελιά τα οποία εξερευνήθηκαν και ως αποτέλεσμα την σημαντική μείωση στον χρόνο αναζήτησης και της απόστασης η οποία διανύθηκε από το όχημα. Έτσι ο αλγόριθμος CIA^* επιλέγεται ως αποδοτικότερος αλγόριθμος για την εύρεση της βέλτιστης διαδρομής σε ένα γνωστό ή άγνωστο περιβάλλον σε σχέση με τον κλασσικό A^* .

Στην συνέχεια, η υλοποίηση του αλγορίθμου $Weighted A^*$ (WA^*) έχει ως σκοπό την σύγκριση του με τον CIA^* . Η συγκεκριμένη σύγκριση γίνεται στην κατάλληλη απόδοση βάρους για τον WA^* , τέτοια ώστε ο χρόνος εύρεσης να είναι ίδιος ή και μικρότερος με αυτόν του CIA^* και να ερευνηθεί αν η λύση η οποία εξάγεται είναι βέλτιστη. Από τον πίνακα 4.4 φαίνεται πως σε αρένες διαστάσεων 20×20 ο WA^* δεν παρουσιάζει απόκλιση, όπως αναμενόταν λόγω του μεγέθους της αρένας, στην εύρεση της βέλτιστης λύσης και είναι αρκετά πιο αποδοτικός από τους δύο προηγούμενους. Καθώς όμως αυξάνεται το μέγεθος της αρένας σε 40×40 και η πολυπλοκότητα των εμποδίων παράλληλα, ο WA^* βρίσκει λύση η οποία δεν είναι βέλτιστη παρόλο που πετυχαίνει μικρότερο εύρος αναζήτησης και μικρότερους χρόνους από αυτόν του αλγορίθμου CIA^* .

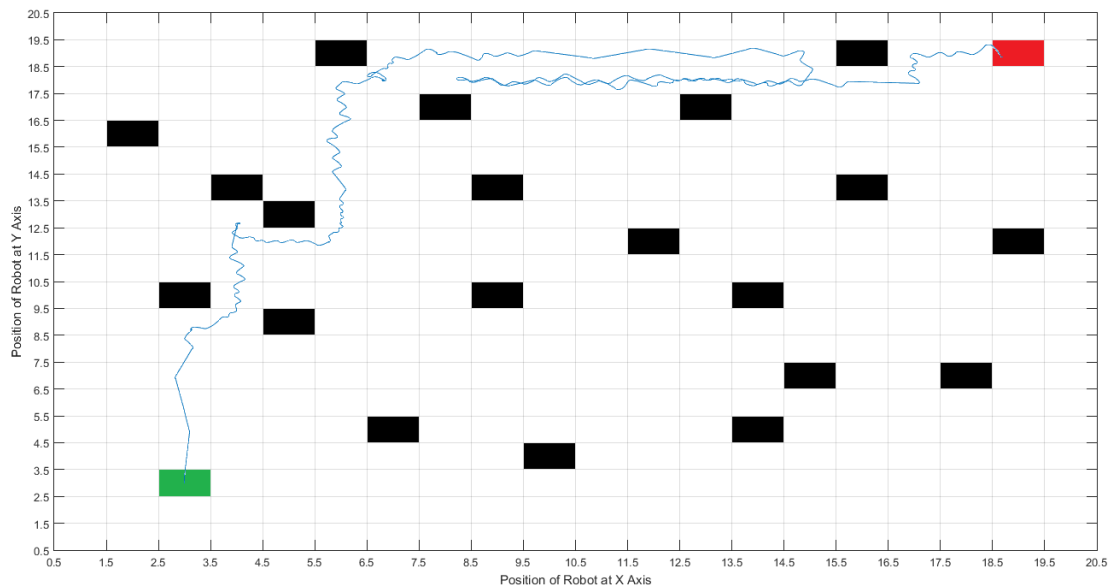
Αρένα (Διαστάσεις)	Εμπόδια (Ποσοστό κάλυψης)	A*		CIA*		Weighted A*		Solution		
		Χρόνος	Απόσταση	Χρόνος	Απόσταση	Χρόνος	Απόσταση	A*	CIA*	WA*
20x20	5% - 20	99.6114	50.1359	88.7575	50.042	68.5171	36.3360	32	32	32
	10% - 40	729.3304	380.0437	253.5846	130.3457	143.3424	61.8957	34	34	34
	15% - 60	415.0394	215.7916	238.7933	71.7348	221.4109	101.3639	34	34	34
	20% - 80	433.7872	313.9959	386.5406	131.7803	170.5362	66.8392	34	34	34
	25% - 100	1190.7645	558.751	356.5200	177.4869	349.6965	189.9450	36	36	36
	30% - 120	2043.8902	1298.9065	1161.4909	702.7458	1186.1305	745.9919	40	40	40
40x40	5% - 80	507.5222	365.4653	653.4880	309.6691	184.8137	83.3276	72	72	76
	10% - 160	1028.2998	437.1210	840.9225	360.9367	153.3801	99.1296	72	72	74
	15% - 240	2204.6825	1300.7822	967.3534	377.4094	277.0056	151.3291	74	74	74
	20% - 320	4124.0392	1349.4612	1055.4075	537.6484	194.7979	109.2764	74	74	76
	25% - 400	11818.560	2956.5142	1573.7991	442.1684	3557.1092	1391.011	76	76	82
	30% - 480	4054.0350	1340.6445	3161.9170	832.6980	1008.6864	515.3425	78	78	82

Πίνακας 4.4 Χρόνος προσομοίωσης αλγορίθμων

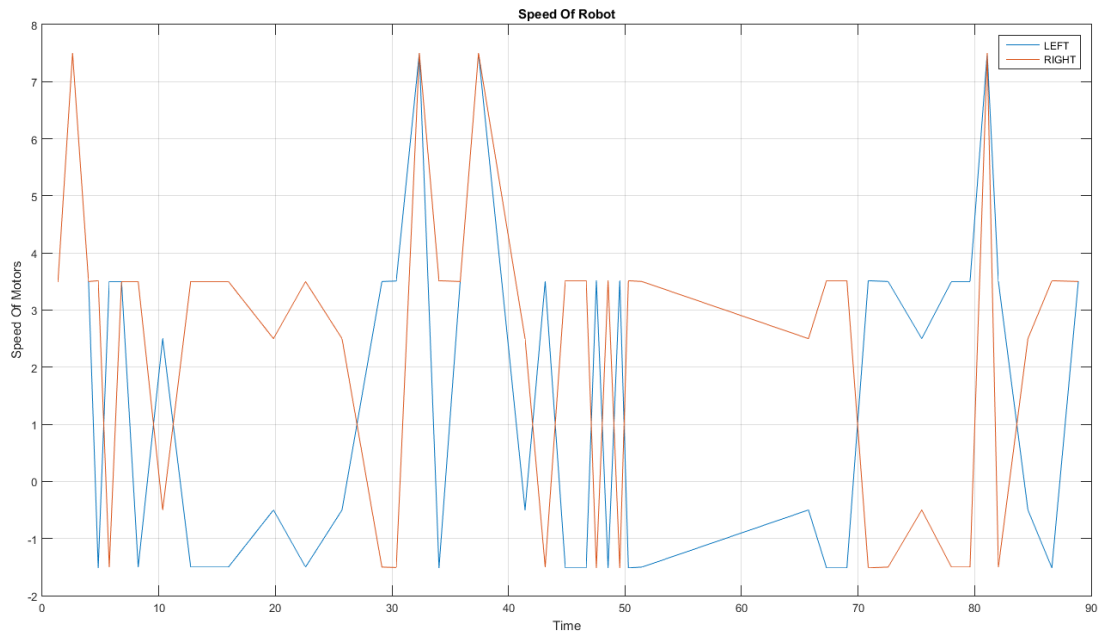
4.4.5 Αποτελέσματα προσομοίωσης αλγορίθμου A* με χρήση ασαφούς ελεγκτή

Στις προηγούμενες ενότητες παρουσιάστηκε και αναλύθηκε η αναζήτηση της βέλτιστης διαδρομής και η συμπεριφορά του ρομποτικού οχήματος με χρήση ενός απλού ελεγκτή με προκαθορισμένο τρόπο λειτουργίας. Στην ενότητα αυτή καταγράφονται τα αποτελέσματα με τη χρήση ενός ασαφούς ελεγκτή, με χαρακτηριστικά όπως αναφέρθηκαν στο κεφάλαιο 4.2, με το οποίο κινείται το ρομποτικό όχημα. Η υλοποίηση πραγματοποιήθηκε και για τους τρεις ευριστικούς αλγορίθμους και έχει ως σκοπό της σύγκριση και την μελέτη μεταξύ των δύο αυτών ελεγκτών ως προς την συμπεριφορά του οχήματος. Παρουσιάζονται παρακάτω τα δεδομένα από τους δύο ελεγκτές, όπως η διαδρομή που ακολούθησε το όχημα, οι ταχύτητες των τροχών, ο απαιτούμενος χρόνος για την εύρεση της βέλτιστης λύσης και η απόσταση την οποία διήνυσε το ρομποτικό όχημα.

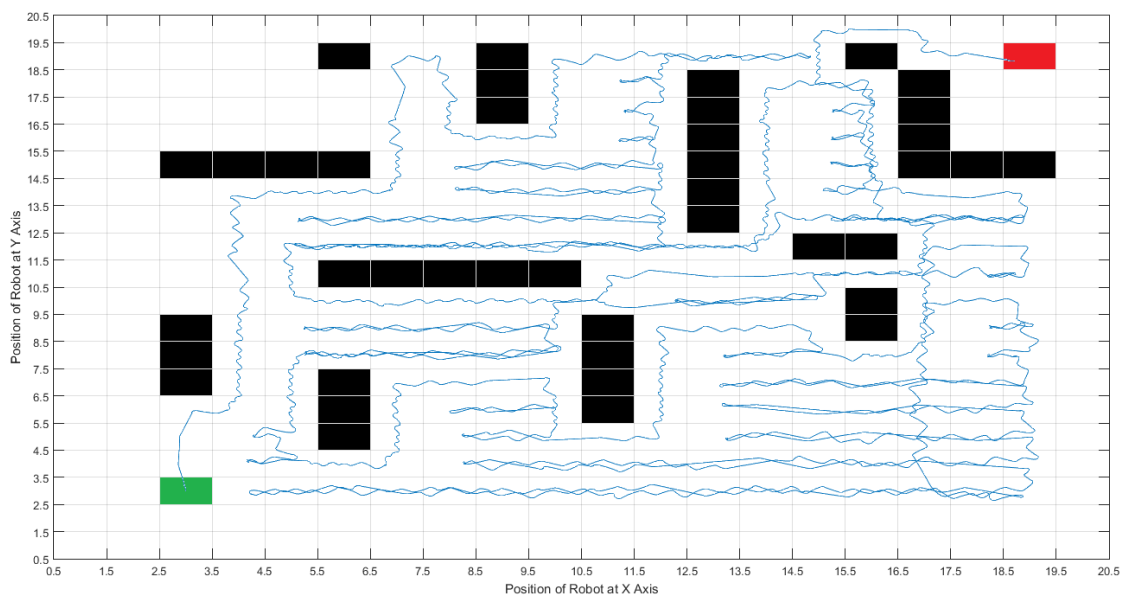
Η αναζήτηση της βέλτιστης διαδρομής πραγματοποιήθηκε με τον αλγόριθμο A*. Αρχικά, στα σχήματα 4.50 με 4.61 παρουσιάζονται οι αρένες διαστάσεων 20x20, την διαδρομή που ακολούθησε το όχημα και την ταχύτητα των δύο τροχών με τη χρήση του ασαφούς ελεγκτή.



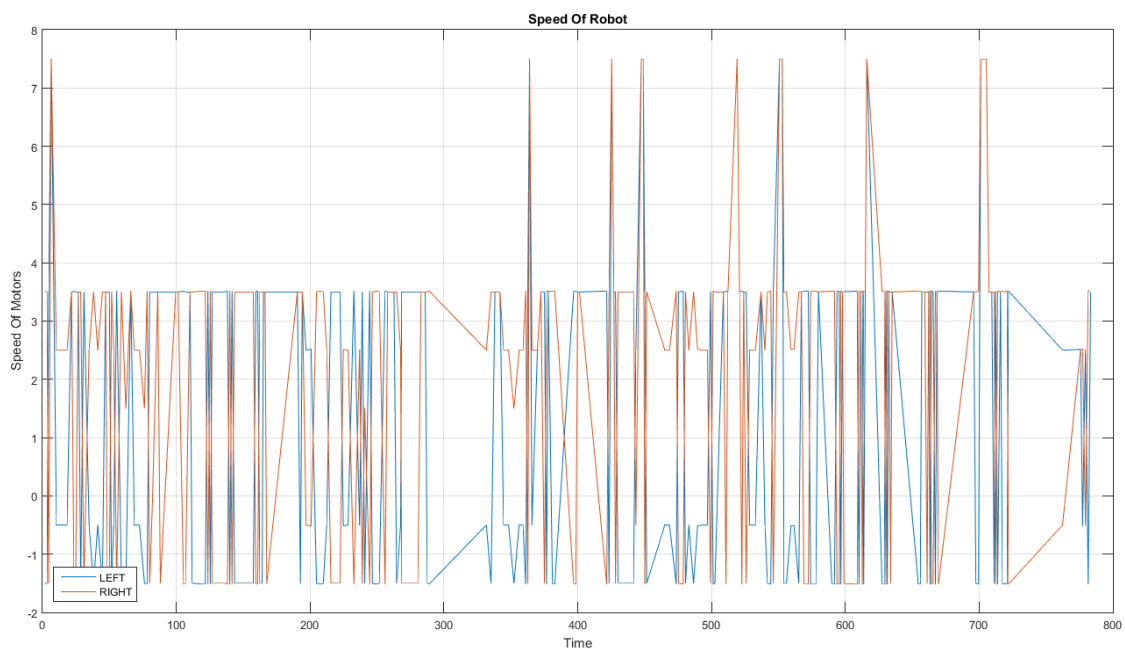
Σχήμα 4.50 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%



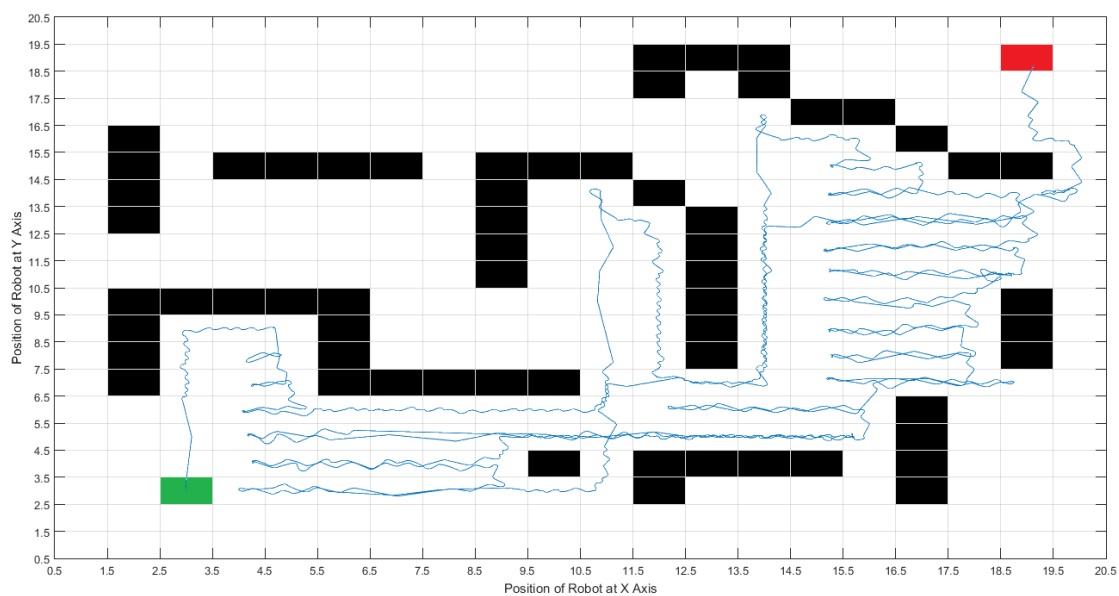
Σχήμα 4.51 Ταχύτητα τροχών με ασαφή ελεγκτή



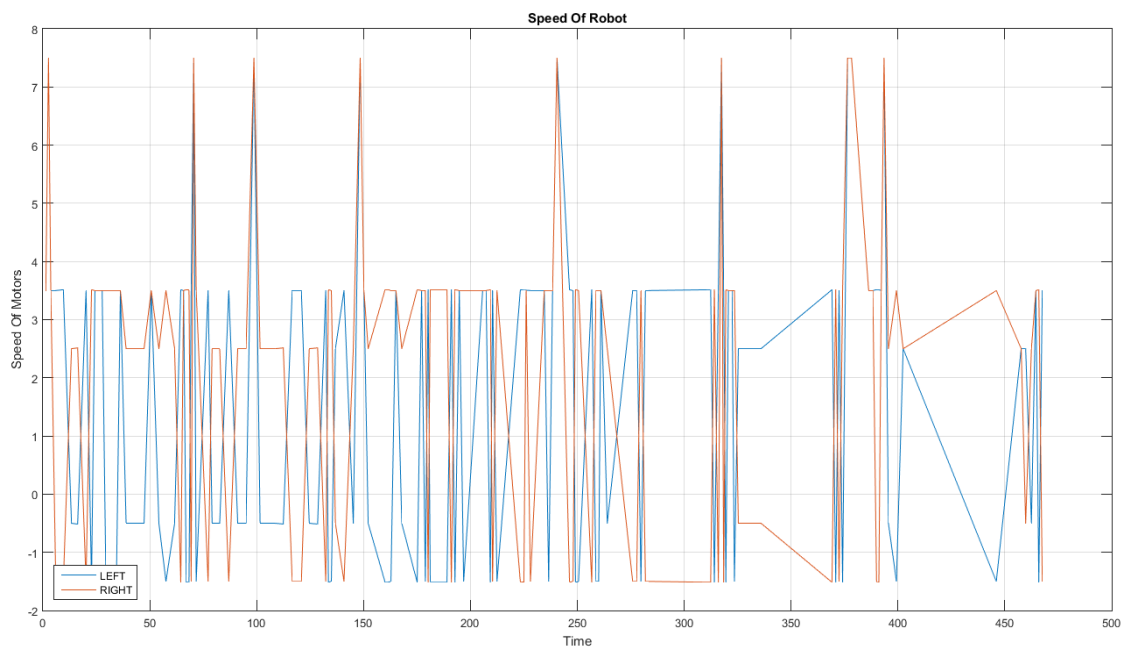
Σχήμα 4.52 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%



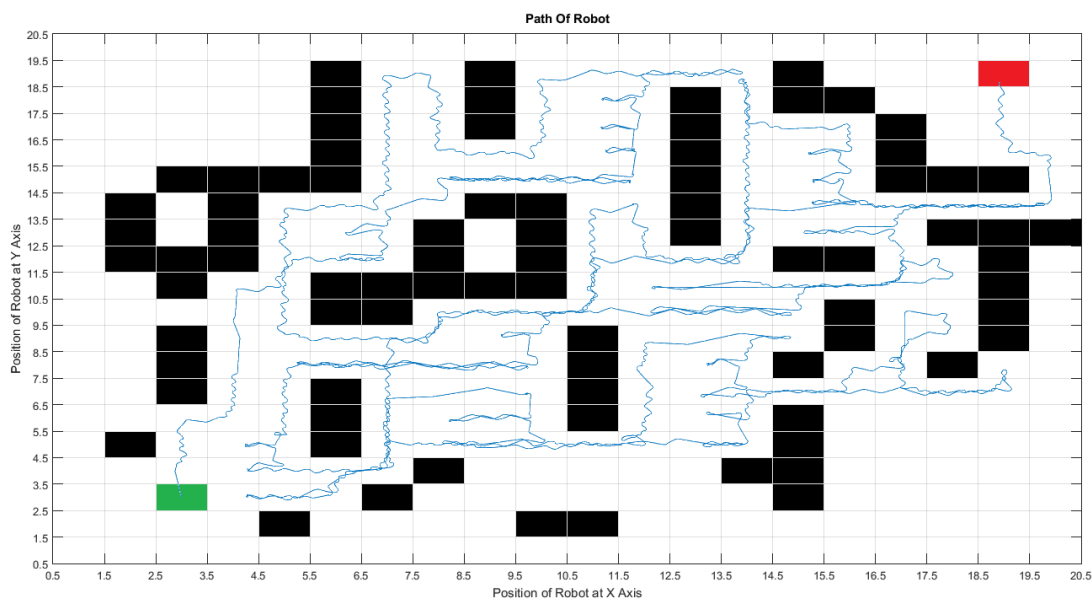
Σχήμα 4.53 Ταχύτητα τροχών με ασαφή ελεγκτή



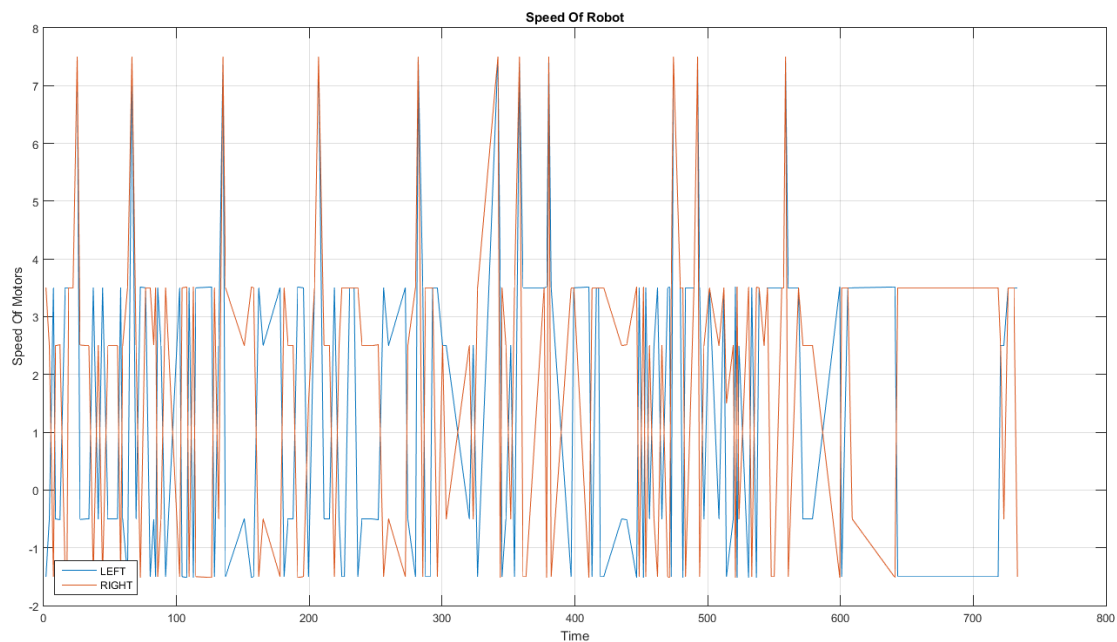
Σχήμα 4.54 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%



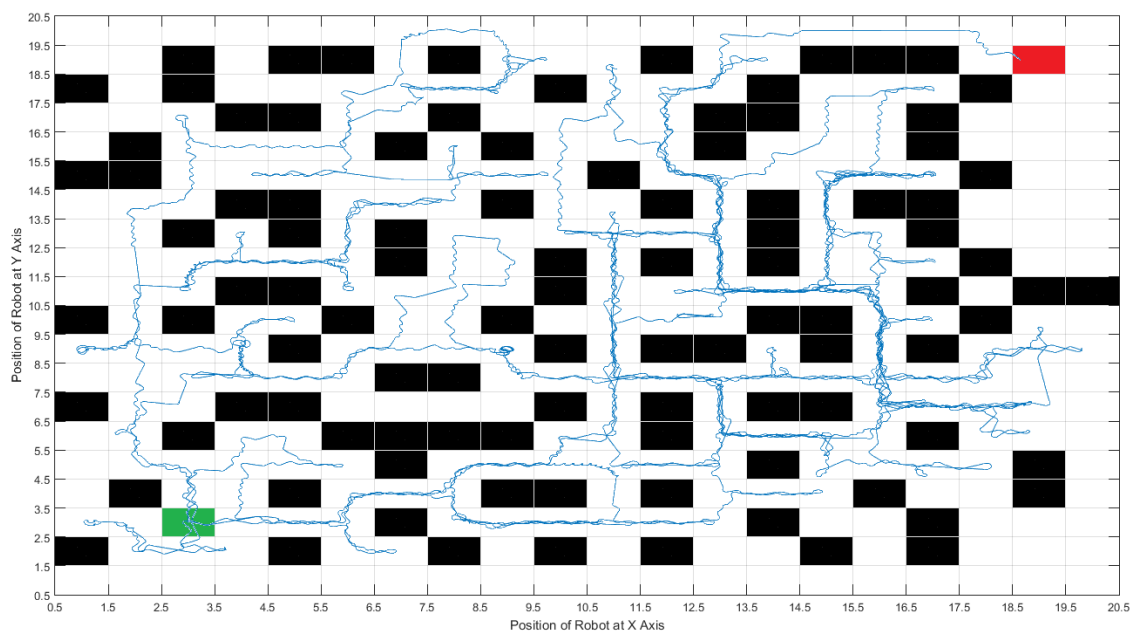
Σχήμα 4.55 Ταχύτητα τροχών με ασαφή ελεγκτή



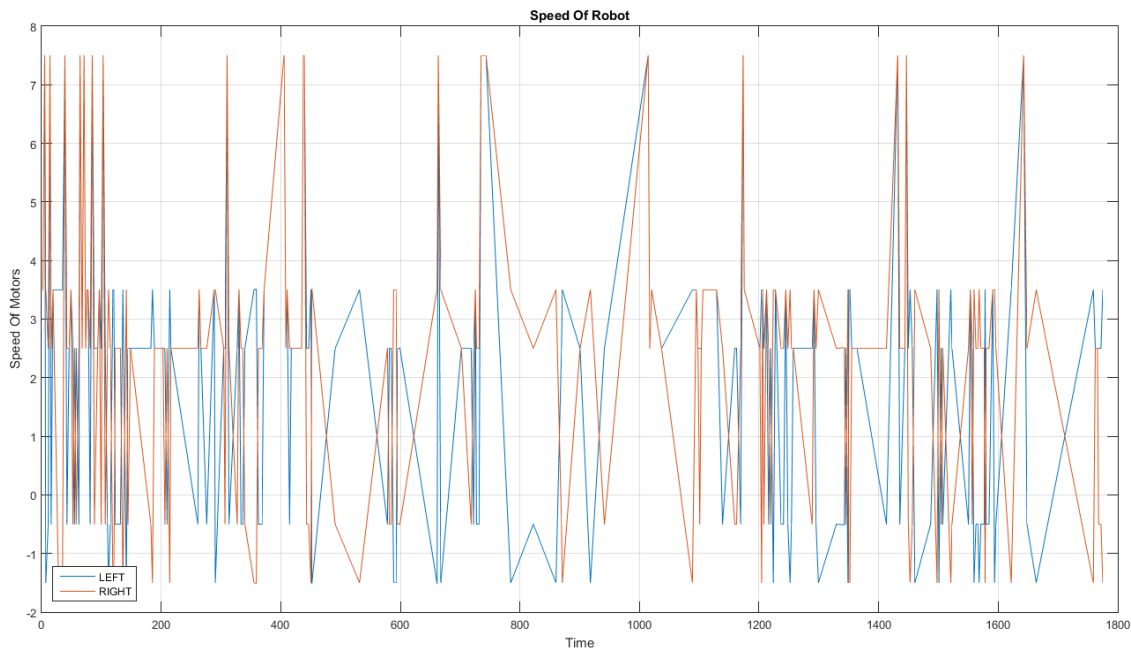
Σχήμα 4.56 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%



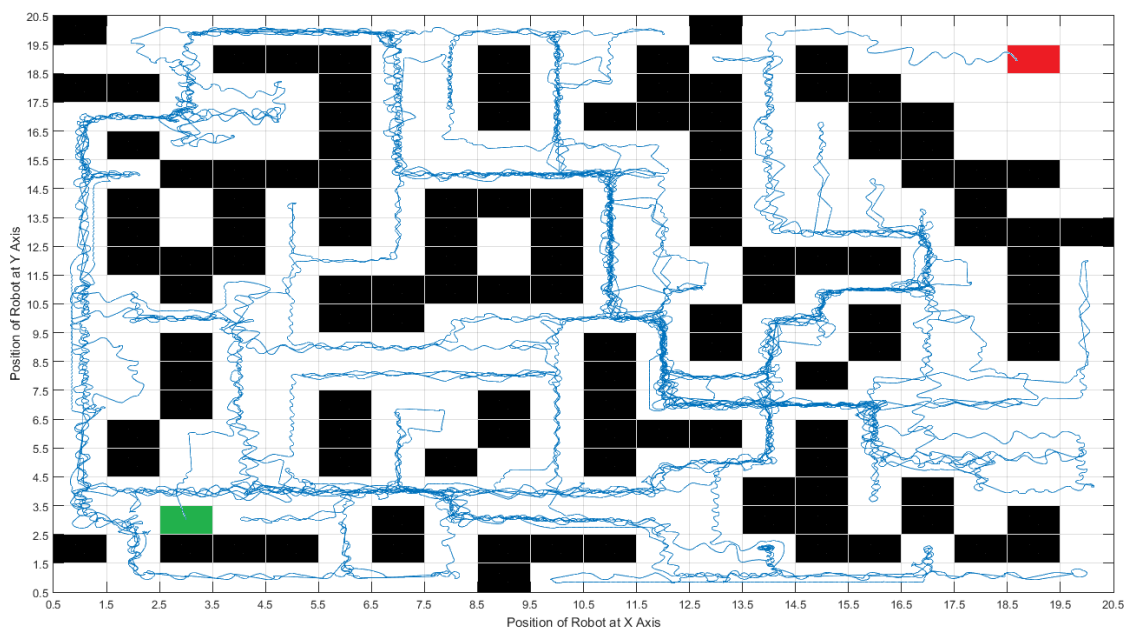
Σχήμα 4.57 Ταχύτητα τροχών με ασαφή ελεγκτή



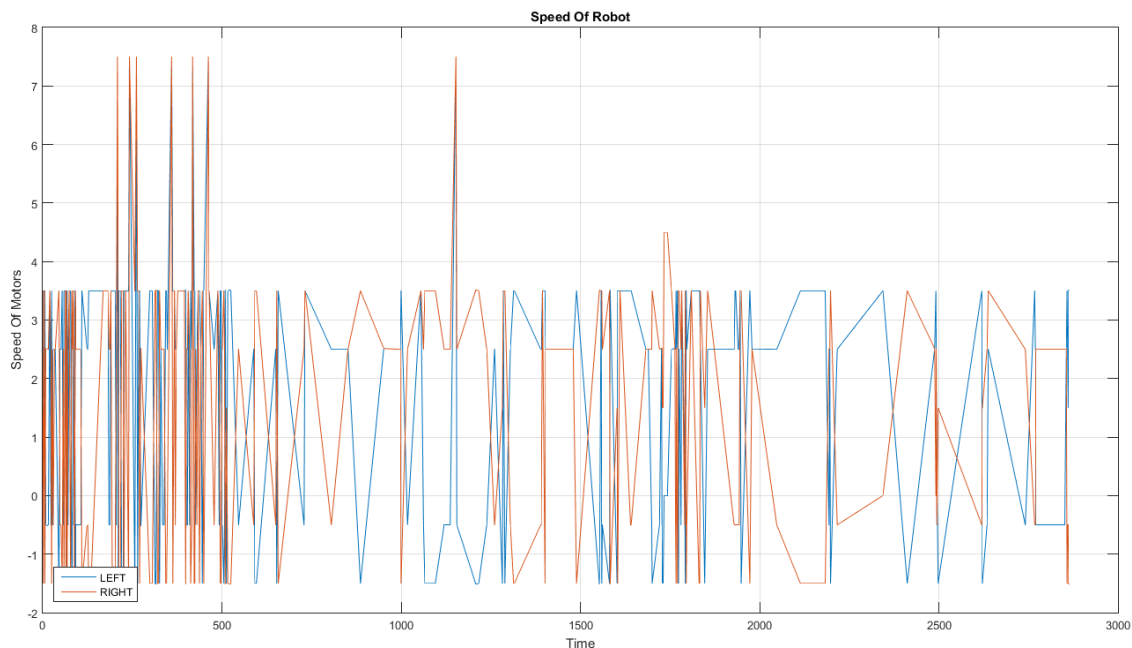
Σχήμα 4.58 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%



Σχήμα 4.59 Ταχύτητα τροχών με ασαφή ελεγκτή

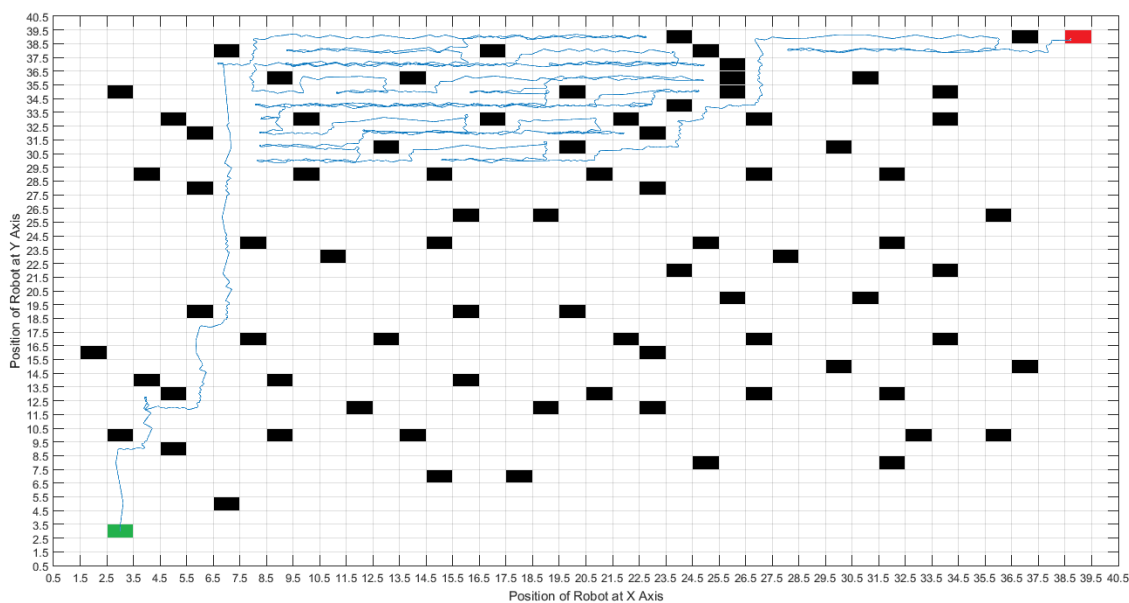


Σχήμα 4.60 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%

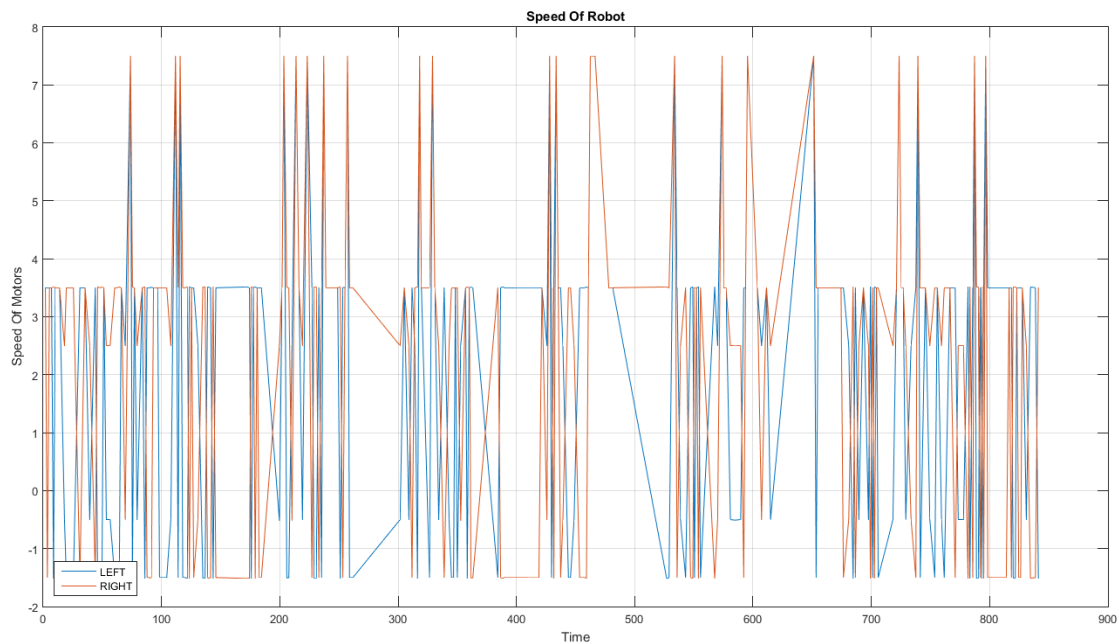


Σχήμα 4.61 Ταχύτητα τροχών με ασαφή ελεγκτή

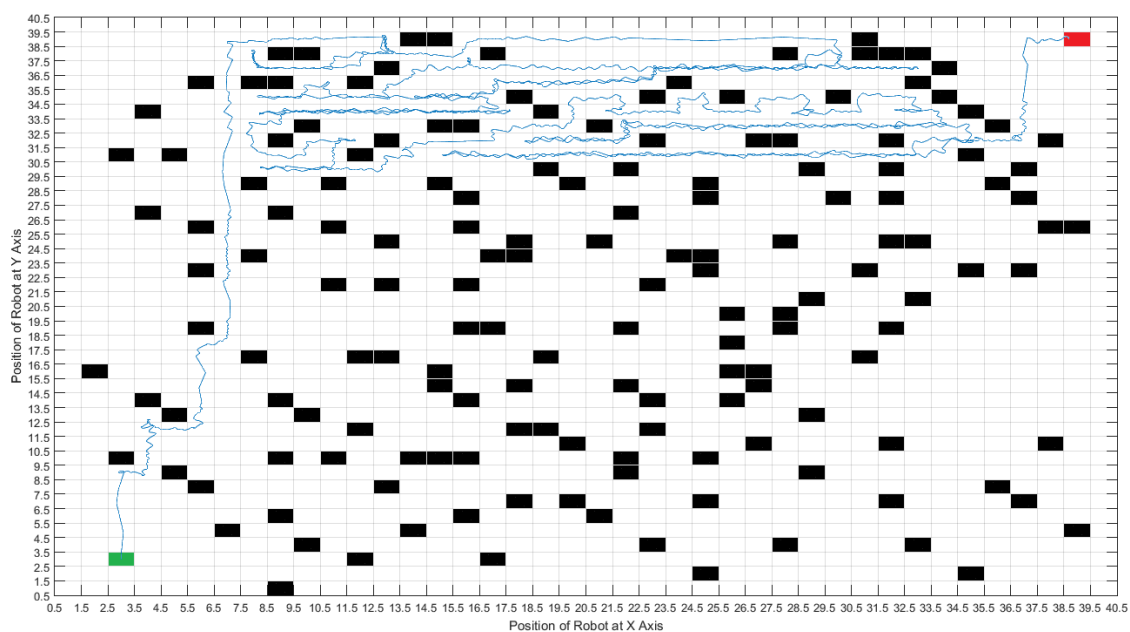
Στη συνέχεια στα σχήματα 4.62 με 4.73 περιέχουν τις αρένες διαστάσεων 40x40 με την διαδρομή του οχήματος και την ταχύτητα των δύο τροχών του.



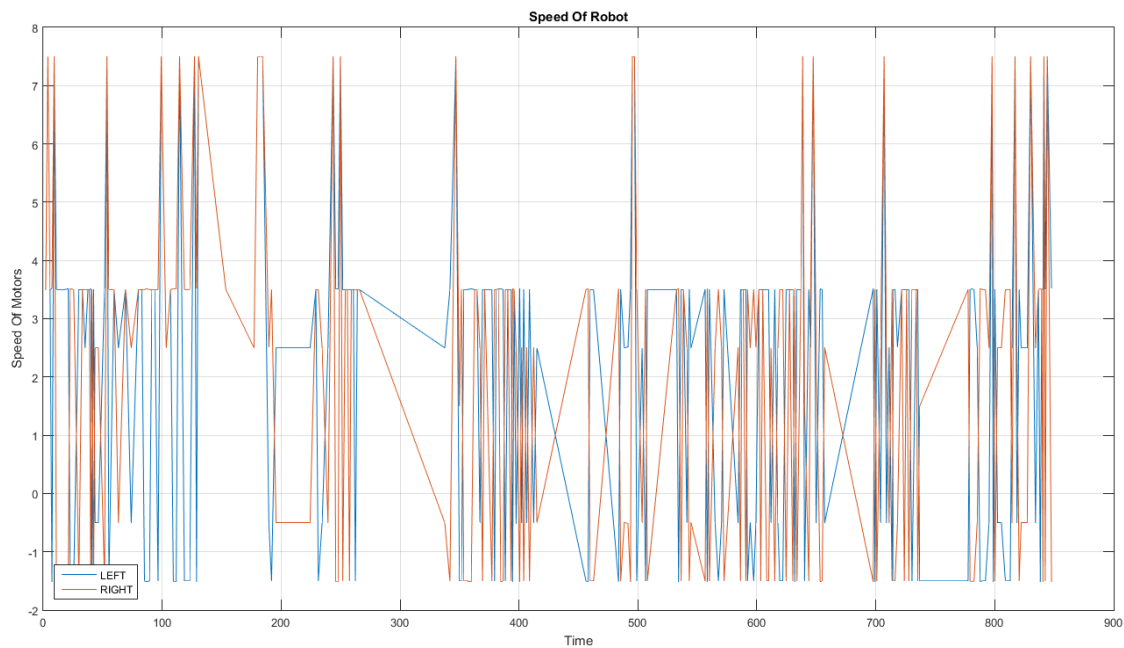
Σχήμα 4.62 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 5%



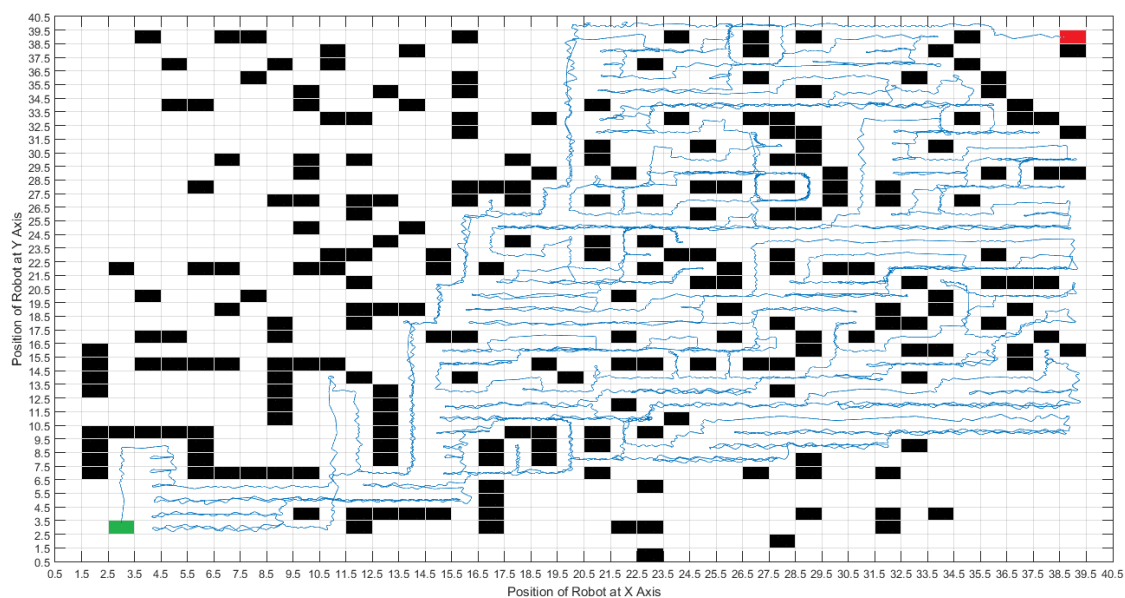
Σχήμα 4.63 Ταχύτητα τροχών με ασαφή ελεγκτή



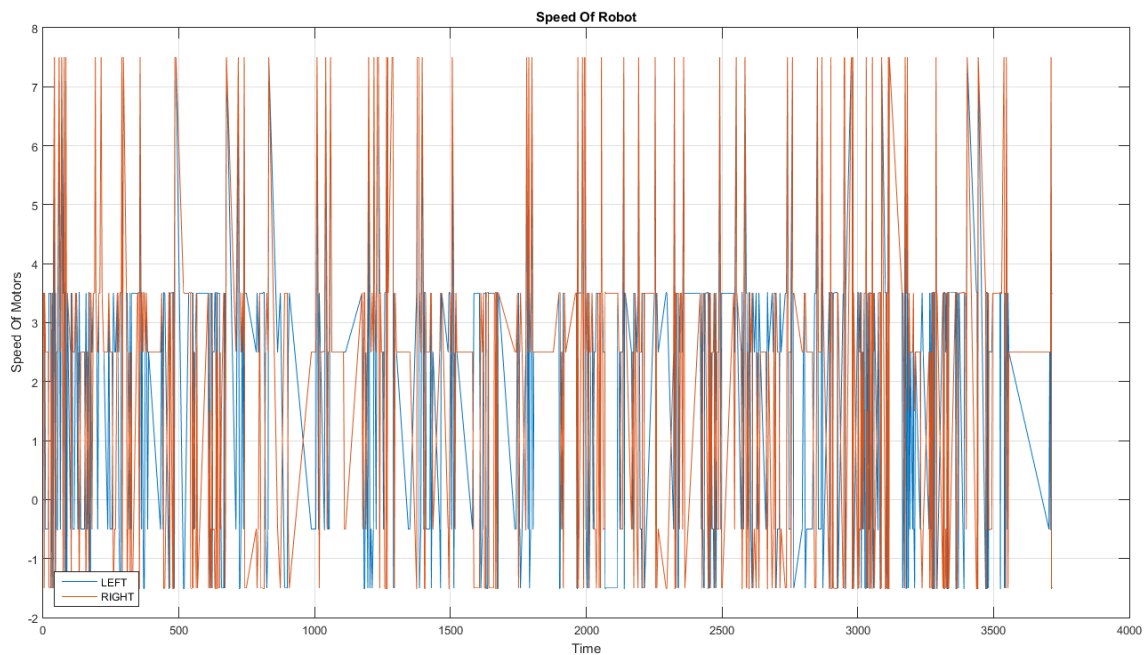
Σχήμα 4.64 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 10%



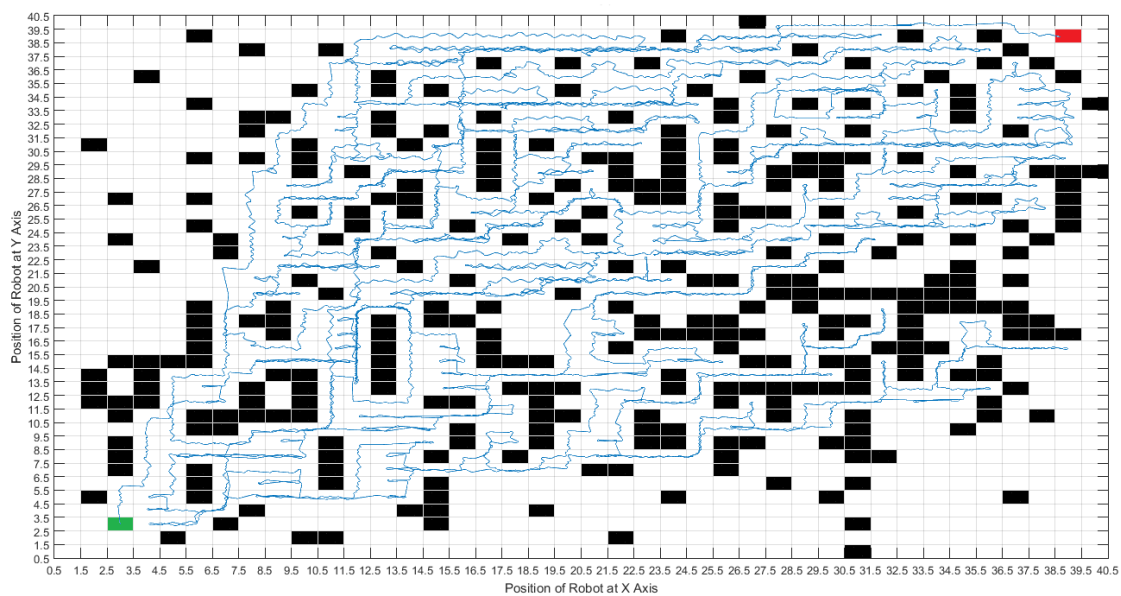
Σχήμα 4.65 Ταχύτητα τροχών με ασαφή ελεγκτή



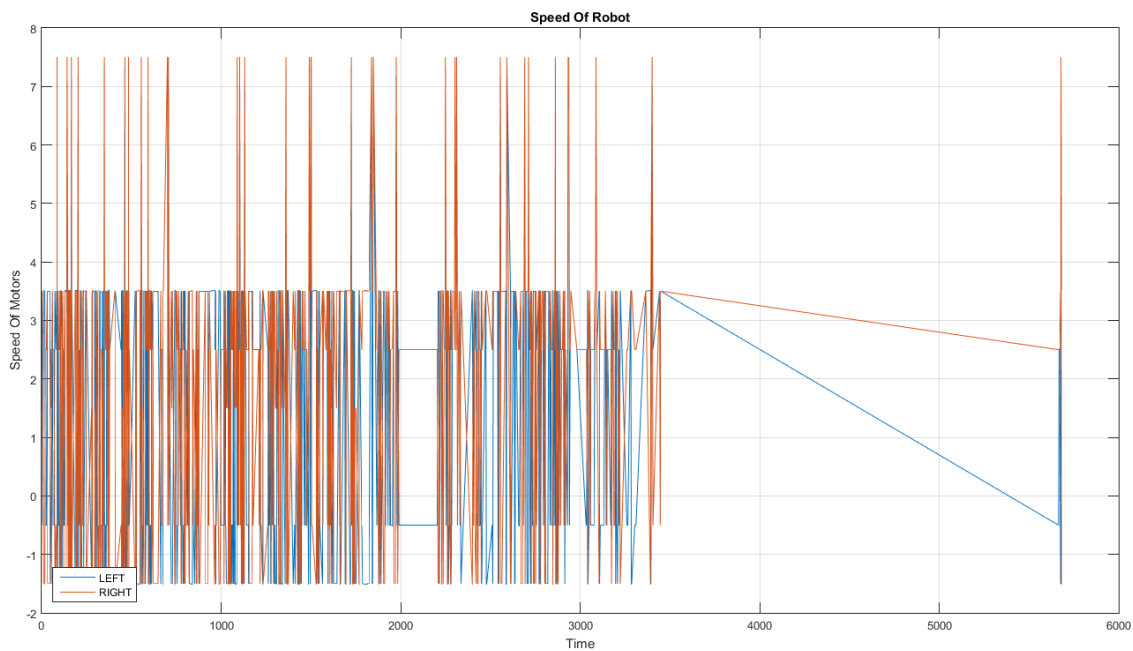
Σχήμα 4.66 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 15%



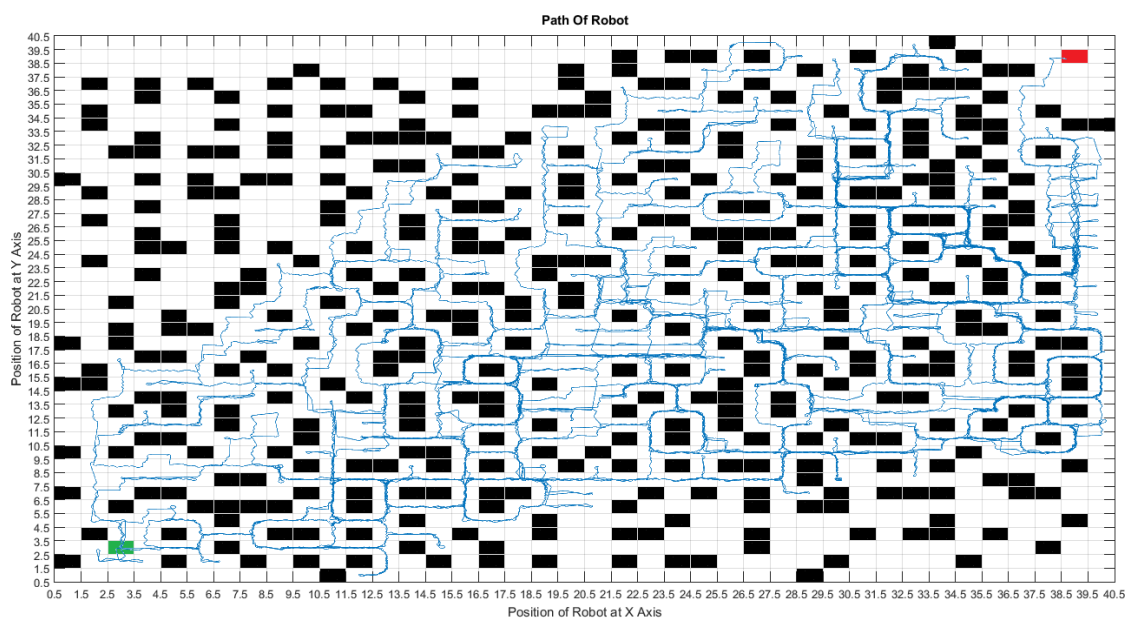
Σχήμα 4.67 Ταχύτητα τροχών με ασαφή ελεγκτή



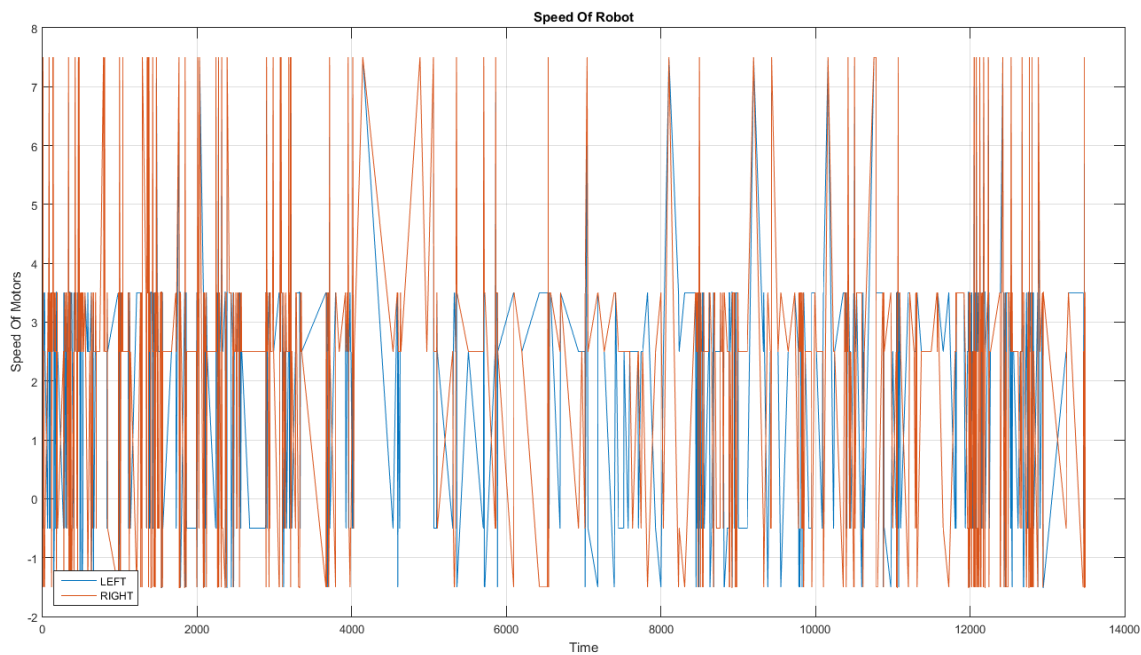
Σχήμα 4.68 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 20%



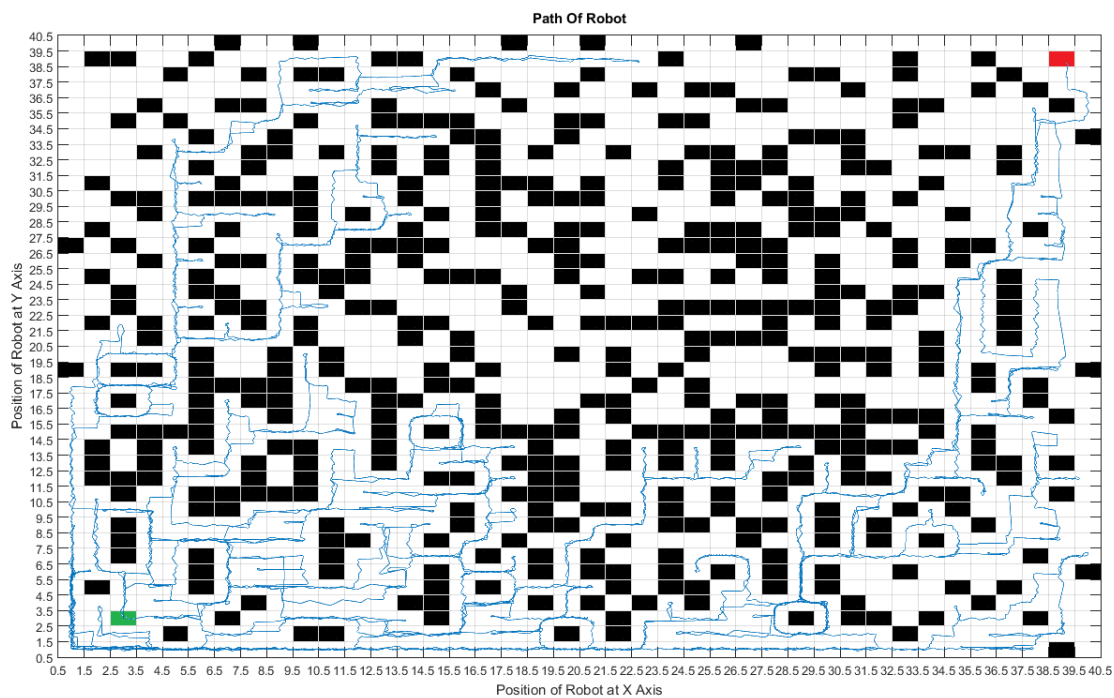
Σχήμα 4.69 Ταχύτητα τροχών με ασαφή ελεγκτή



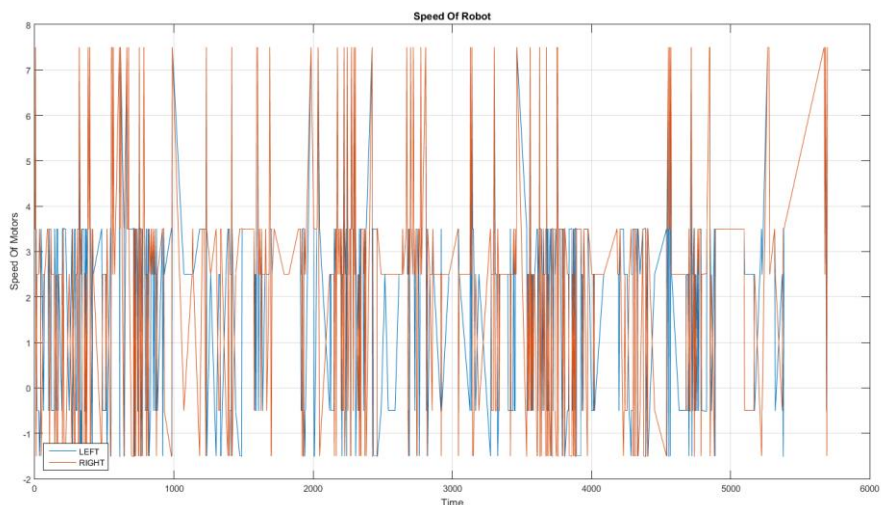
Σχήμα 4.70 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%



Σχήμα 4.71 Ταχύτητα τροχών με ασαφή ελεγκτή



Σχήμα 4.72 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%



Σχήμα 4.73 Ταχύτητα τροχών με ασαφή ελεγκτή

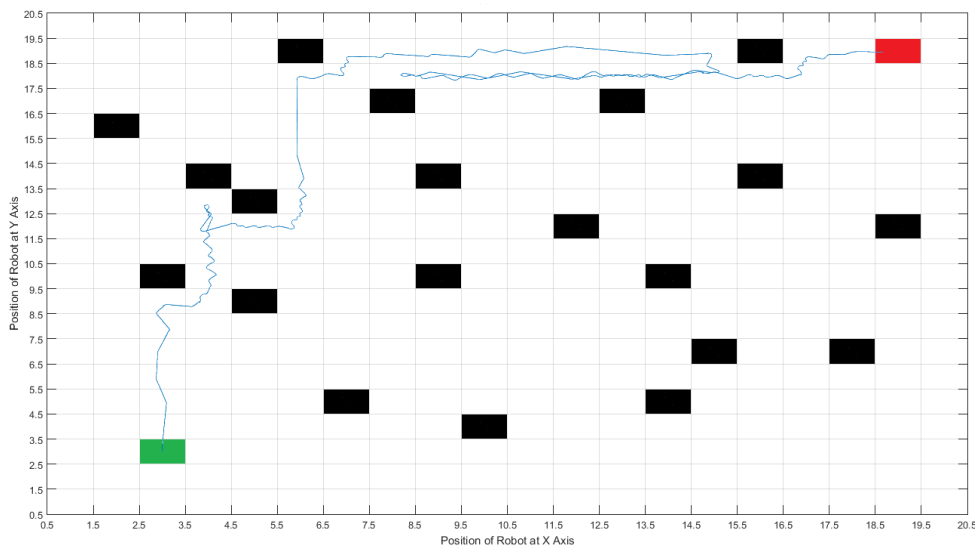
Στον πίνακα 4.5 καταγράφονται συγκεντρωμένα τα δεδομένα των προσομοιώσεων κατά την κίνηση του οχήματος με τον ασαφή ελεγκτή για την αναζήτηση της βέλτιστης διαδρομής με την χρήση του αλγορίθμου A*.

Αλγόριθμος A*					
Degree of Difficulty	Arenas (Dimension)	Time (Seconds)	Distance (Meters)	Expanded Nodes (Cells)	Path Solution (Cells)
5%	20x20	88.8975	57.8201	43	32
	40x40	841.1952	371.8031	228	72
10%	20x20	783.3135	412.6172	226	34
	40x40	847.7304	402.8748	263	72
15%	20x20	467.5159	237.7586	130	34
	40x40	3717.8682	1382.1295	659	74
20%	20x20	733.2827	295.0438	168	34
	40x40	5681.2705	1467.4141	706	74
25%	20x20	1774.8888	573.3751	200	36
	40x40	13490.5033	2642.7515	739	76
30%	20x20	2863.7983	1296.9985	247	40
	40x40	5696.4352	1318.1882	565	78

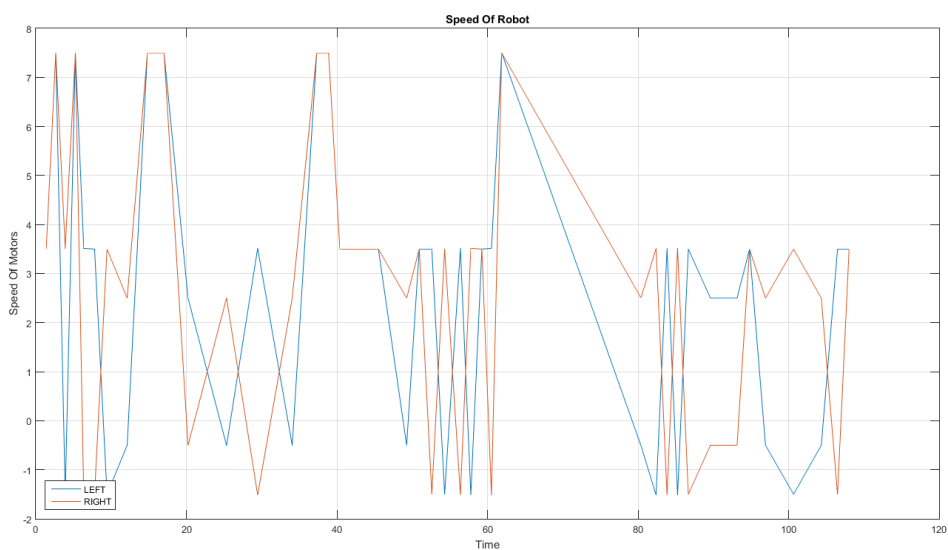
Πίνακα 4.5 Δεδομένα προσομοίωσης αλγορίθμου A* με ασαφή ελεγκτή

4.4.6 Αποτελέσματα προσομοίωσης αλγορίθμου CIA* με χρήση ασαφούς ελεγκτή

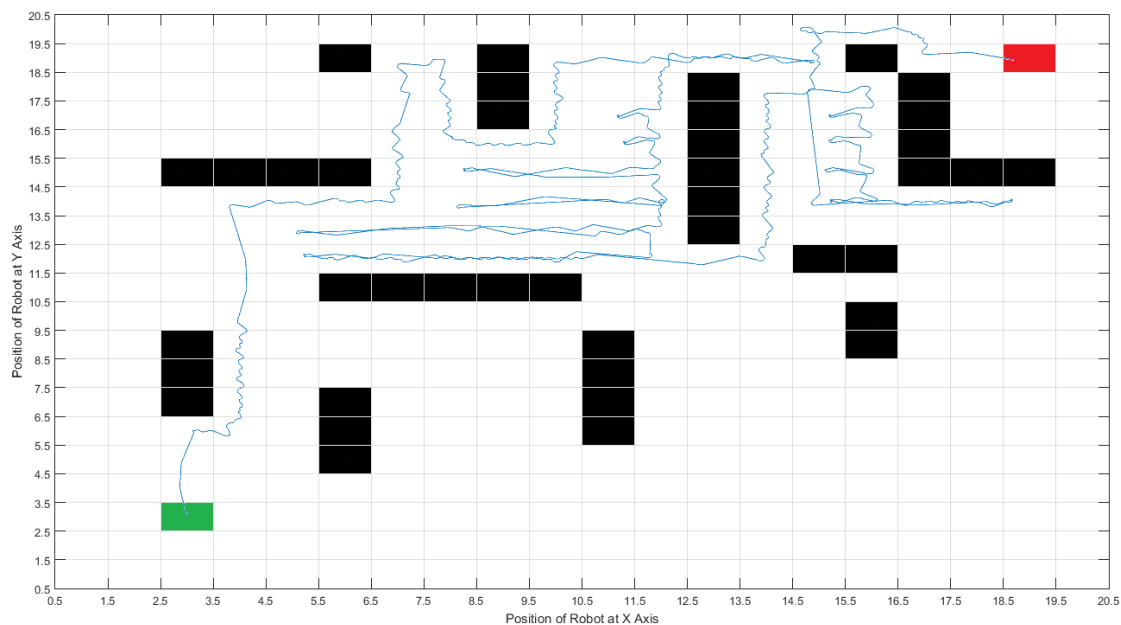
Στη συνέχεια όμοια με τον προηγούμενο ελεγκτή πραγματοποιήθηκαν προσομοιώσεις και με την χρήση του αλγορίθμου CIA*. Οι αρένες και ο βαθμός δυσκολίας είναι παρόμοια με τις προηγούμενες ώστε να είναι δυνατή η σύγκριση μεταξύ των δύο ελεγκτών. Στην αρχή παρουσιάζονται οι αρένες διαστάσεων 20x20 με αυξανόμενο βαθμό δυσκολίας.



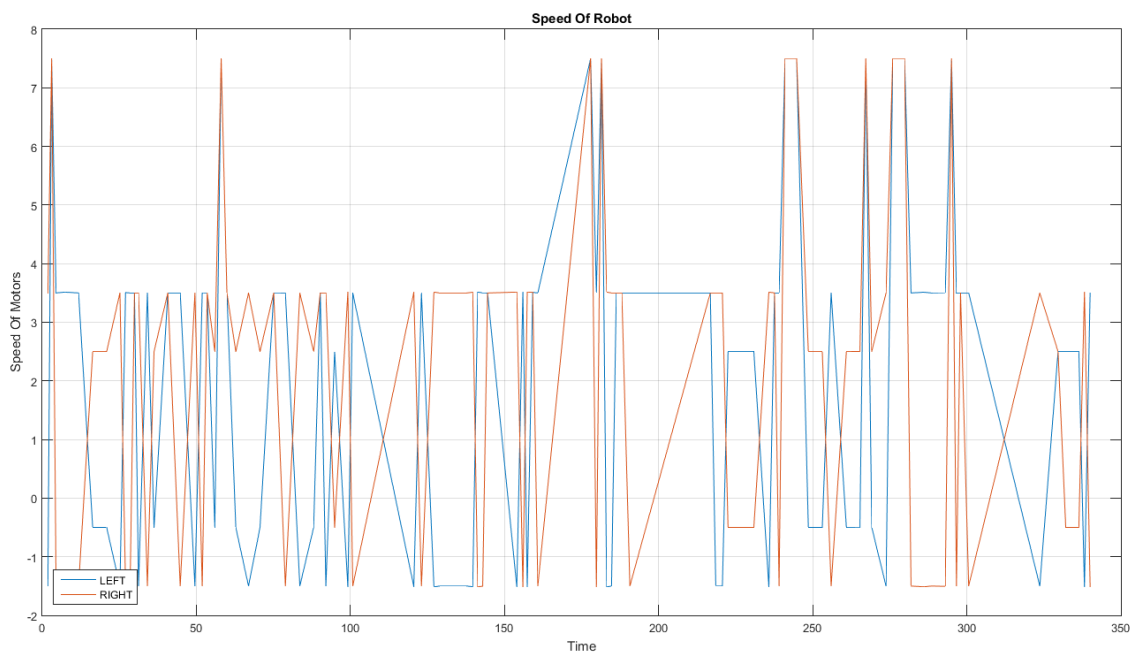
Σχήμα 4.74 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%



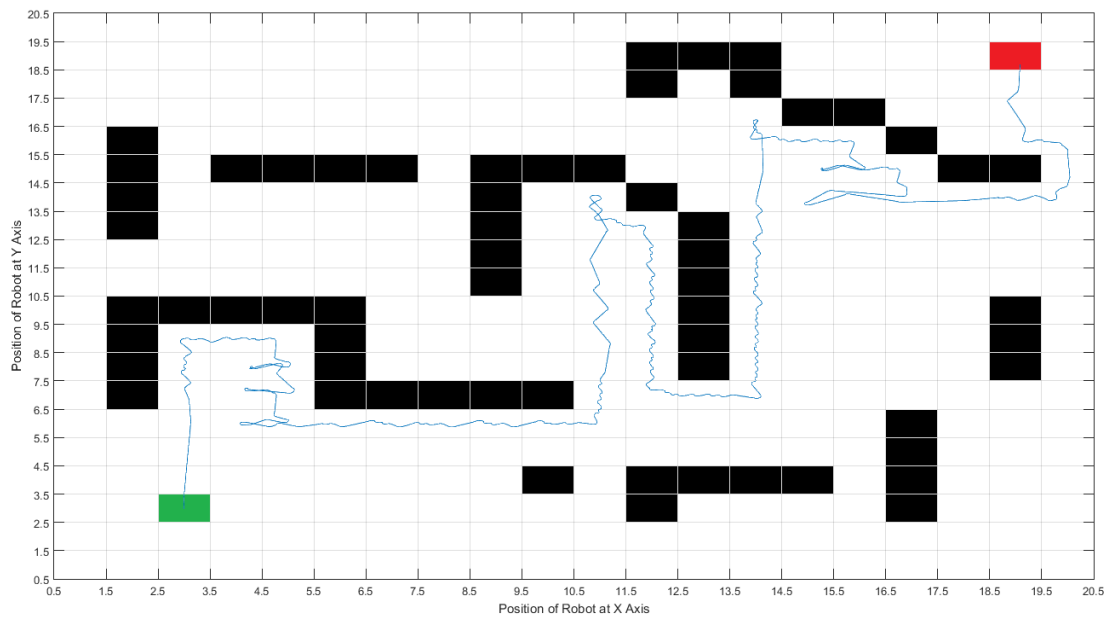
Σχήμα 4.75 Ταχύτητα τροχών με ασαφή ελεγκτή



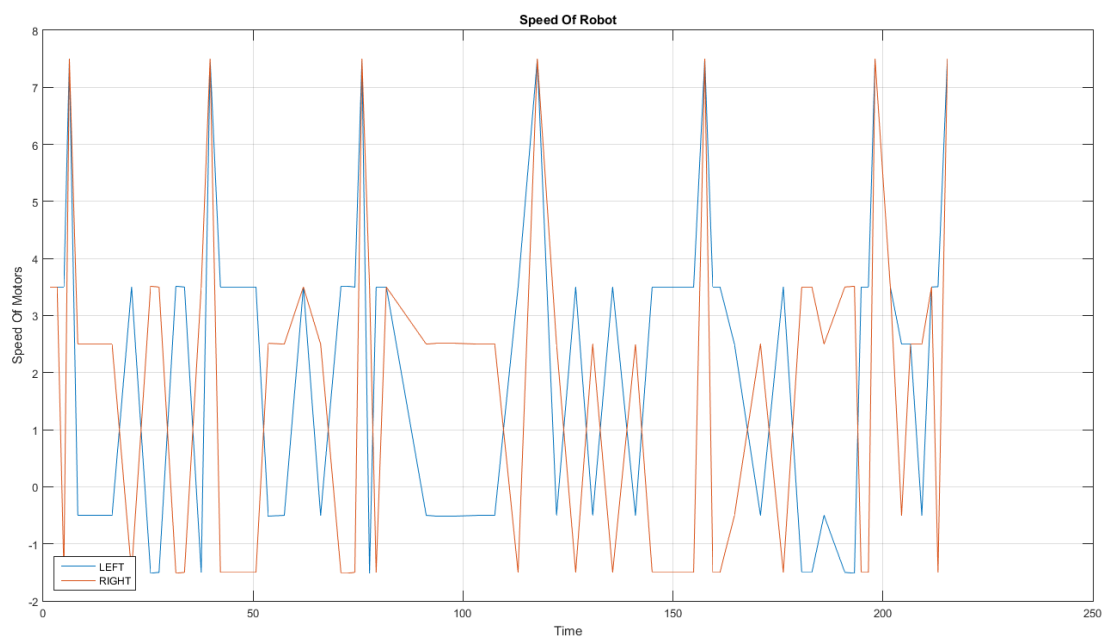
Σχήμα 4.76 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%



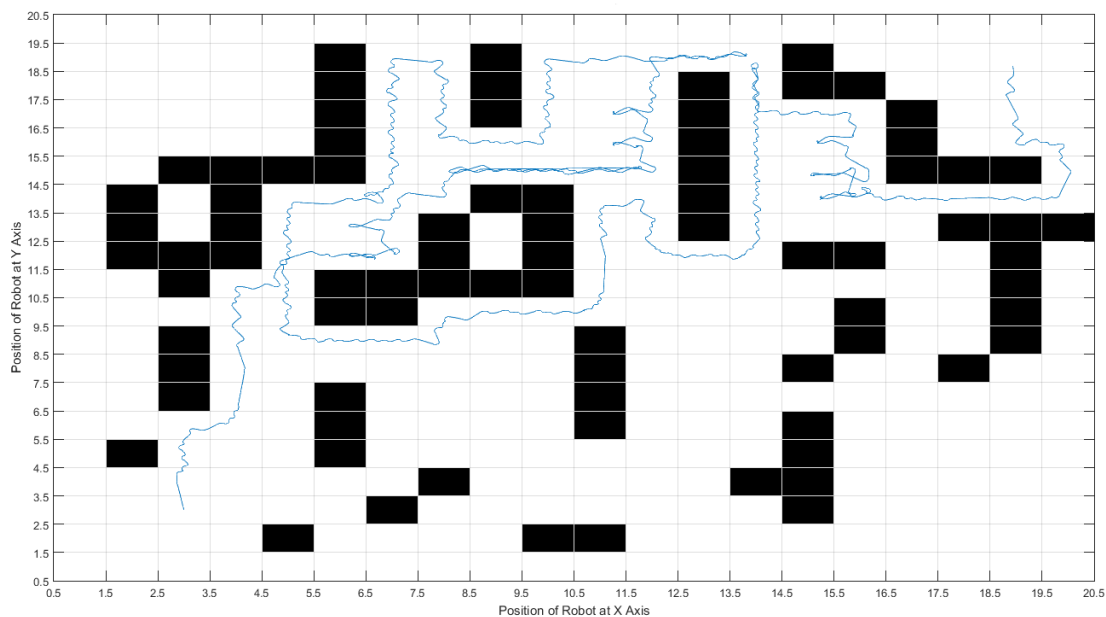
Σχήμα 4.77 Ταχύτητα τροχών με ασαφή ελεγκτή



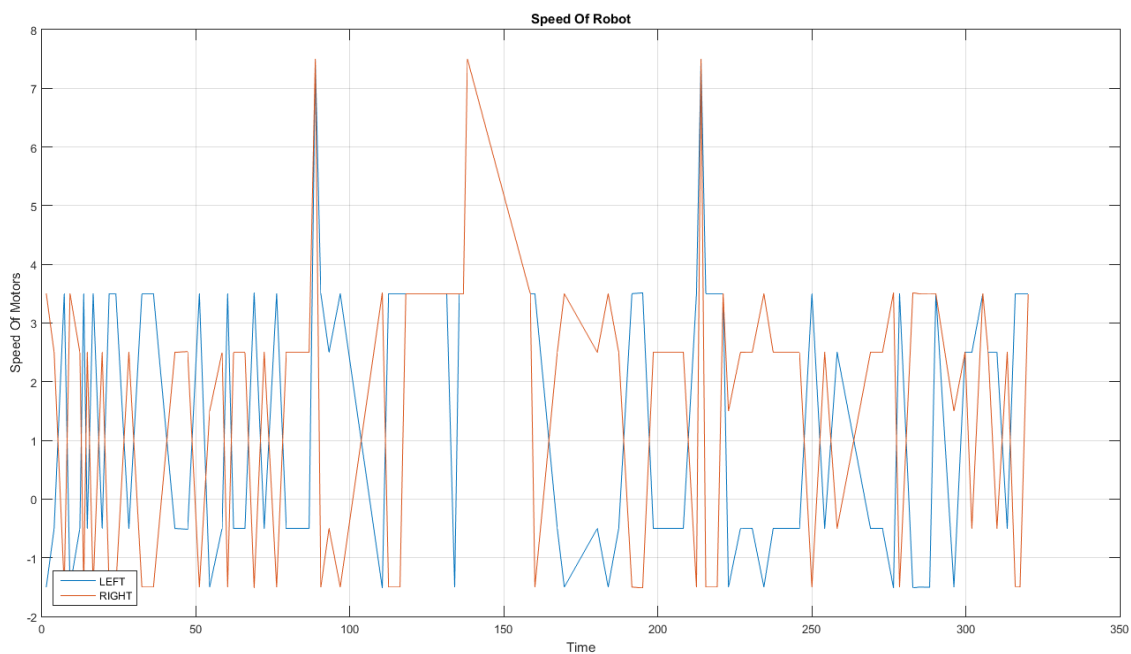
Σχήμα 4.78 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%



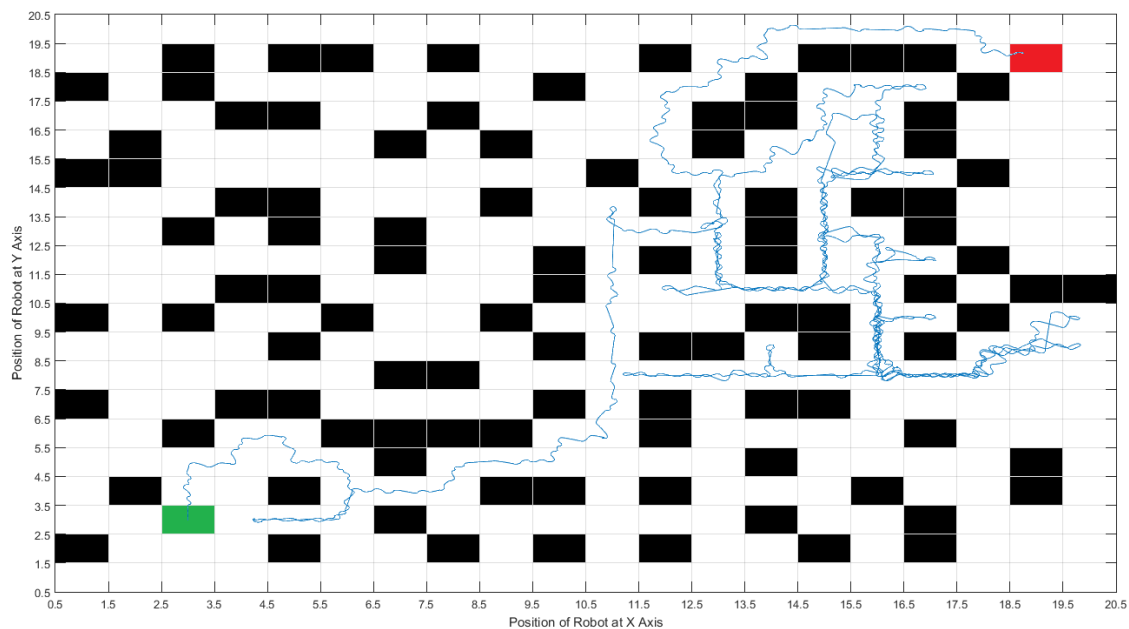
Σχήμα 4.79 Ταχύτητα τροχών με ασαφή ελεγκτή



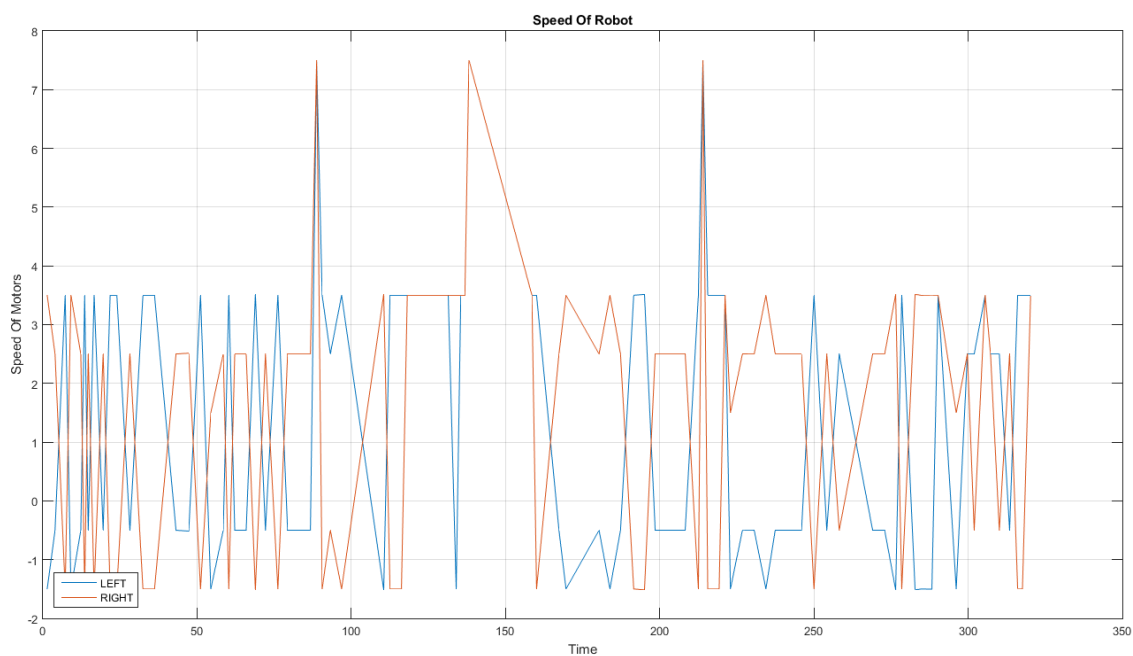
Σχήμα 4.80 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%



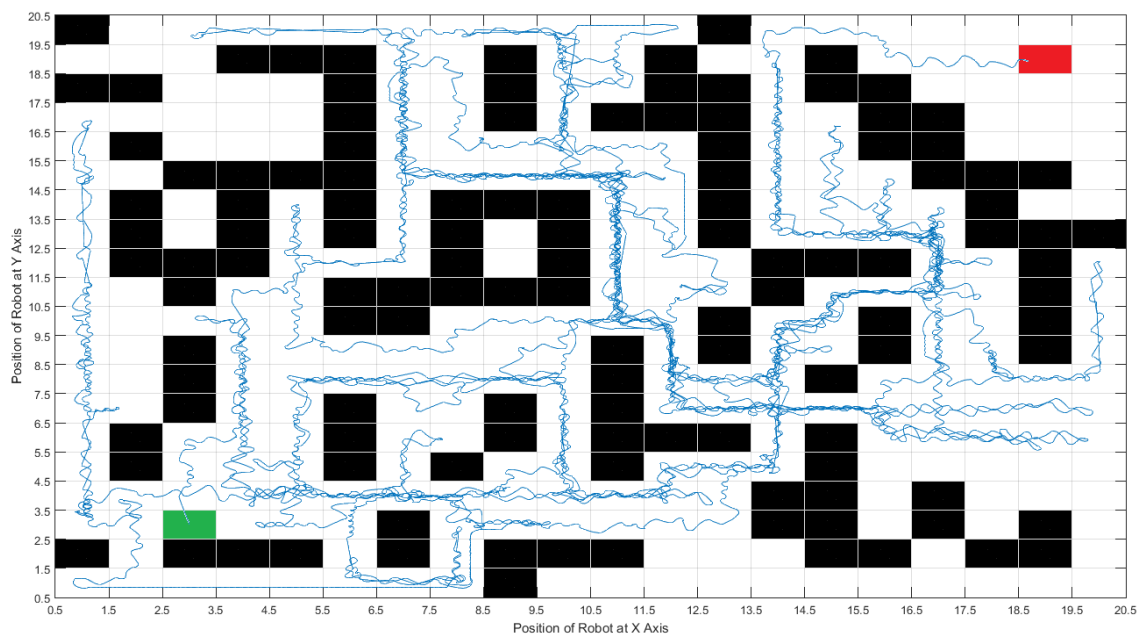
Σχήμα 4.81 Ταχύτητα τροχών με ασαφή ελεγκτή



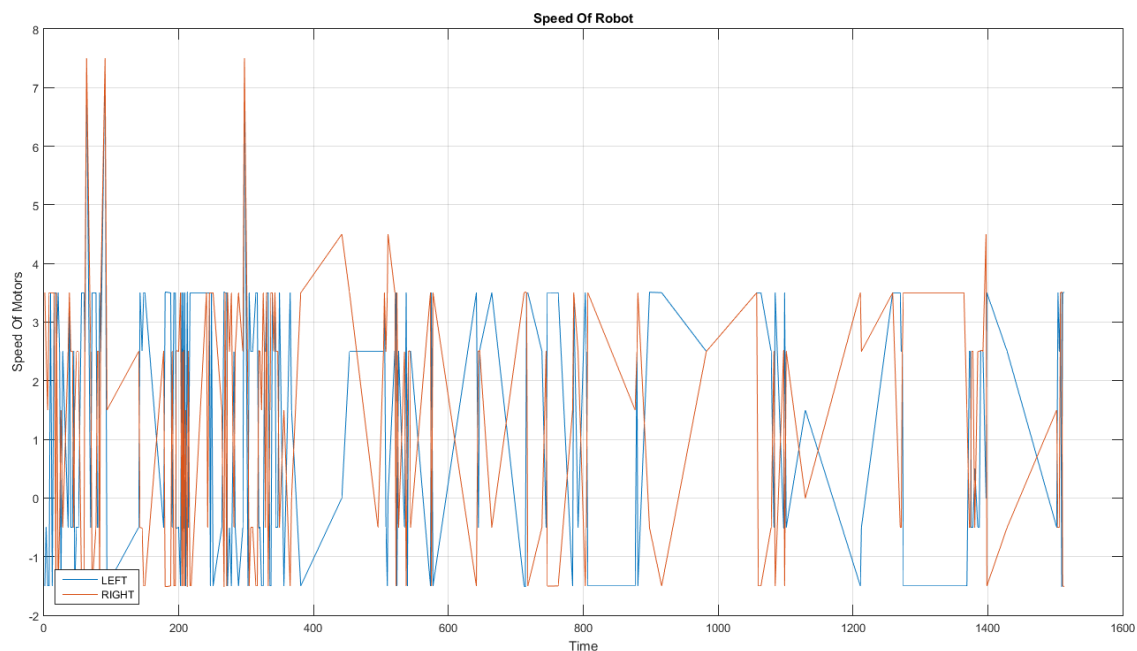
Σχήμα 4.82 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%



Σχήμα 4.83 Ταχύτητα τροχών με ασαφή ελεγκτή

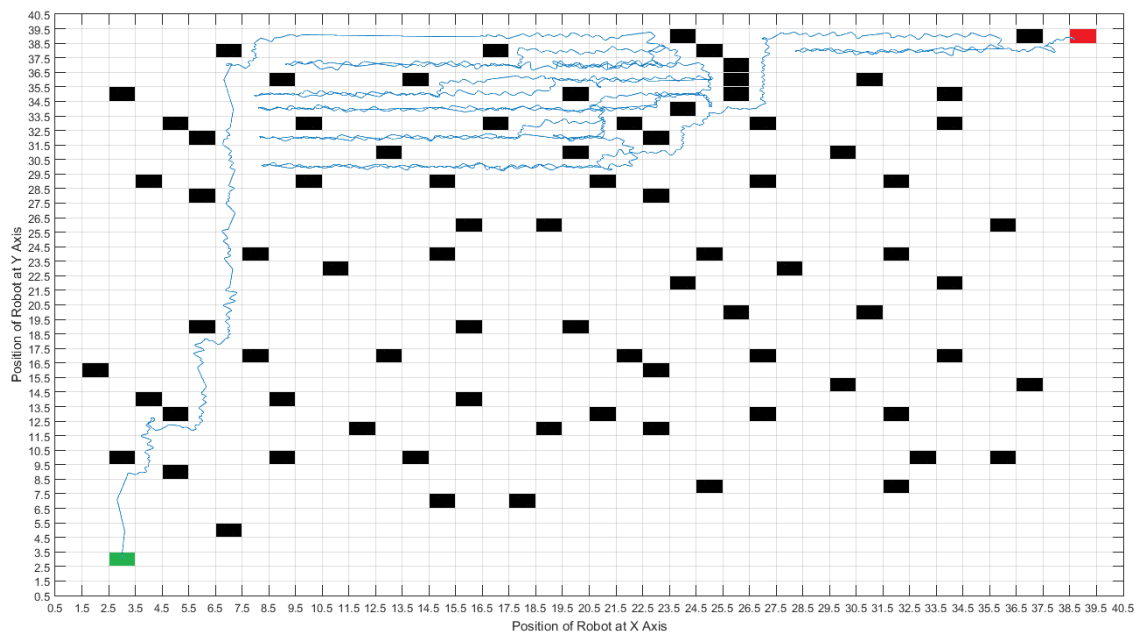


Σχήμα 4.84 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%

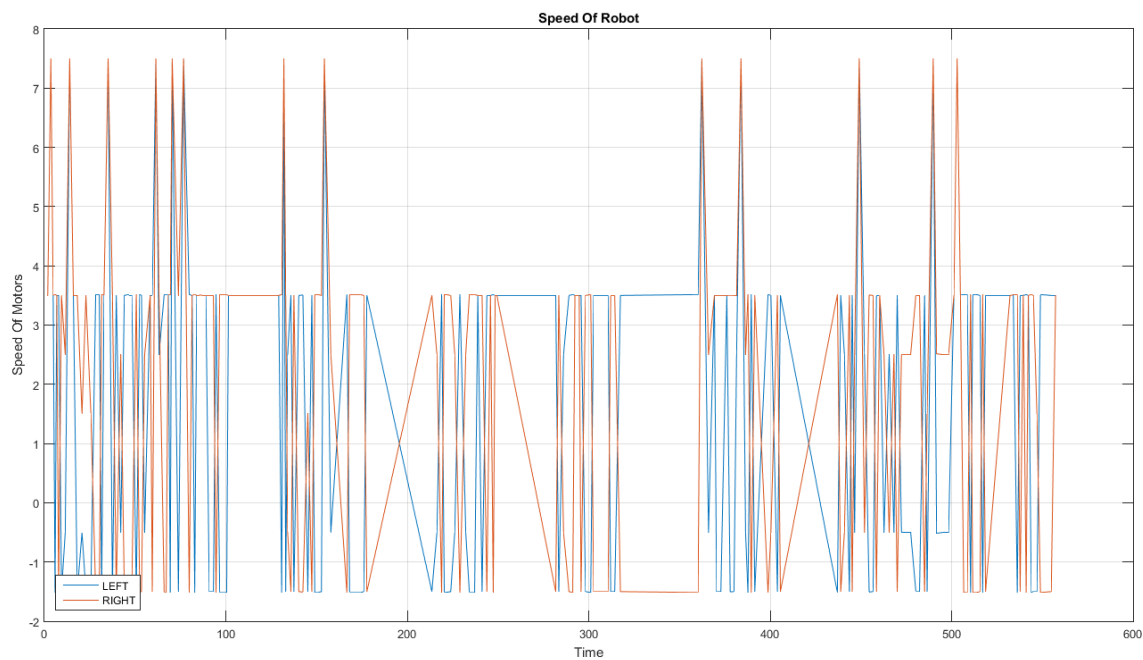


Σχήμα 4.85 Ταχύτητα τροχών με ασαφή ελεγκτή

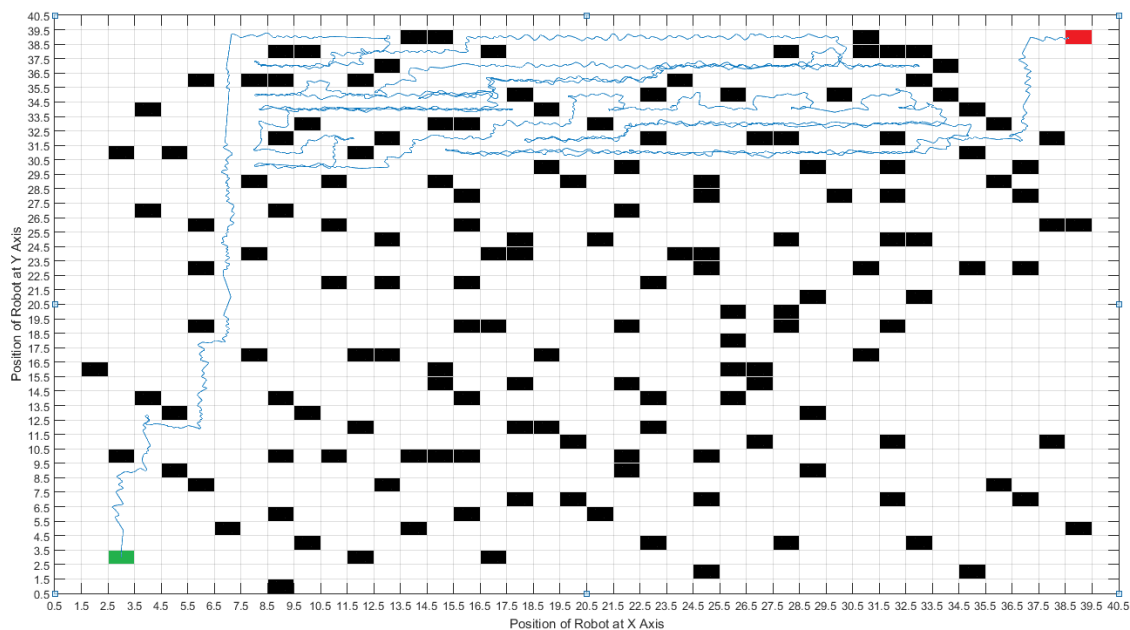
Παρακάτω παρουσιάζονται οι διαδρομές του οχήματος, οι θέσεις των εμποδίων και οι ταχύτητες των τροχών σε μεγαλύτερες διαστάσεις μεγέθους 40x40.



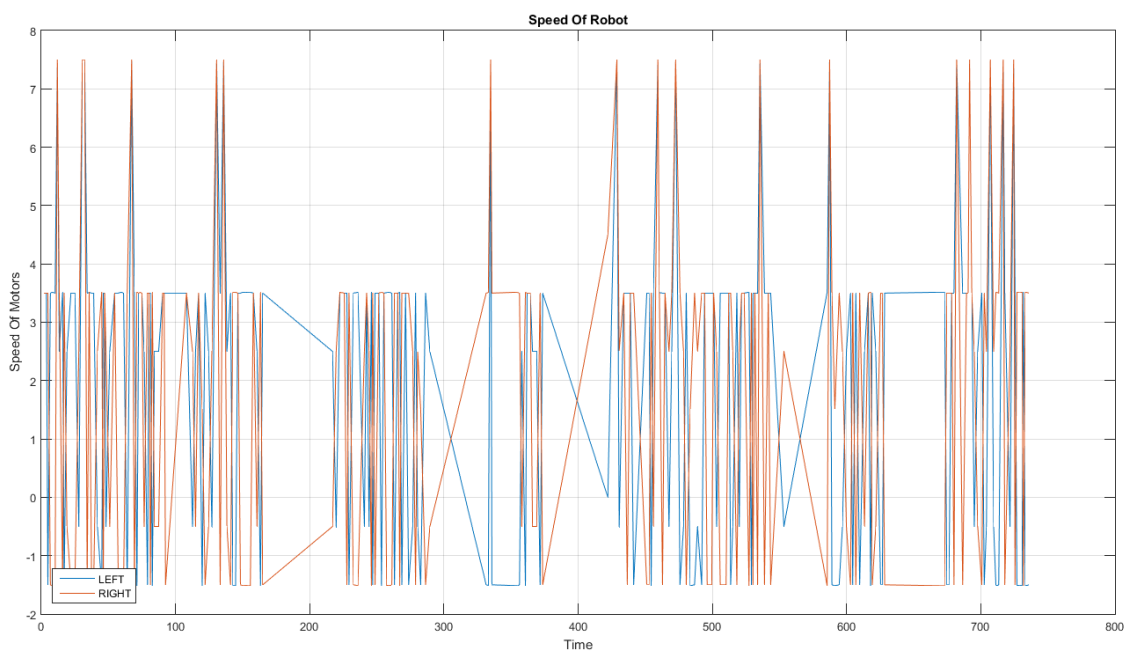
Σχήμα 4.86 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 5%



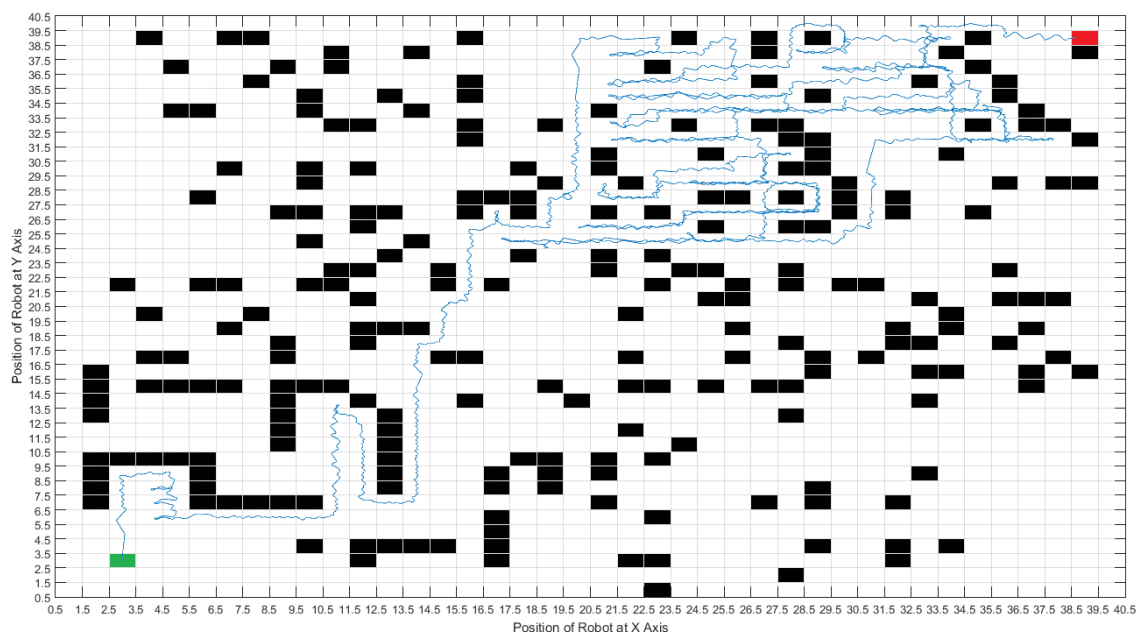
Σχήμα 4.87 Ταχύτητα τροχών με ασαφή ελεγκτή



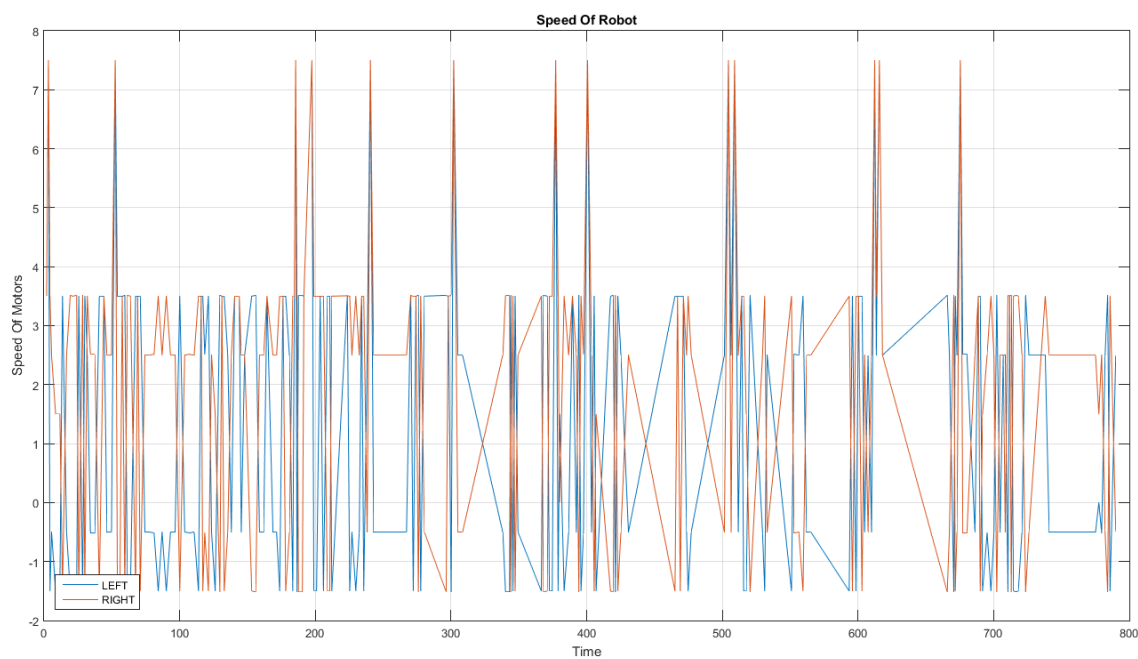
Σχήμα 4.88 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 10%



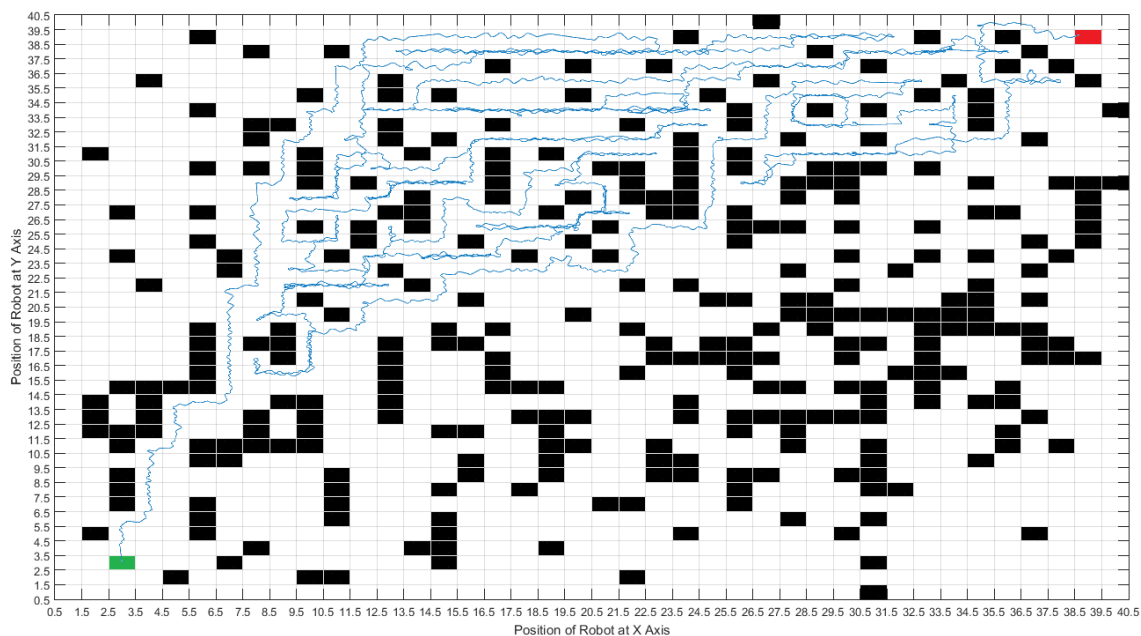
Σχήμα 4.89 Ταχύτητα τροχών με ασαφή ελεγκτή



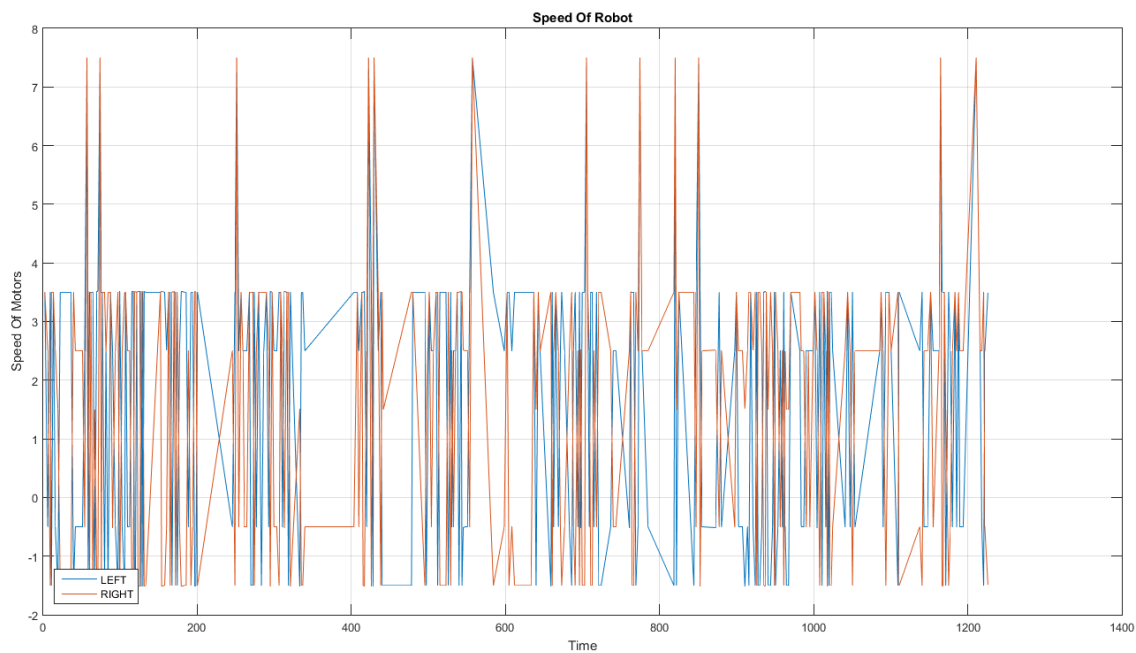
Σχήμα 4.90 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 15%



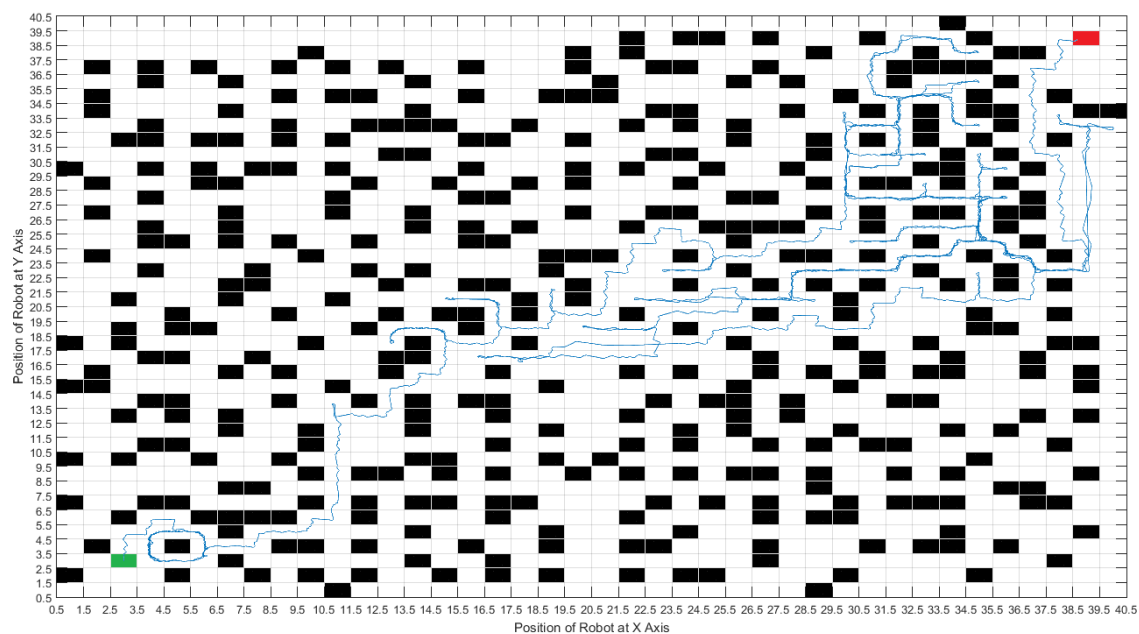
Σχήμα 4.91 Ταχύτητα τροχών με ασαφή ελεγκτή



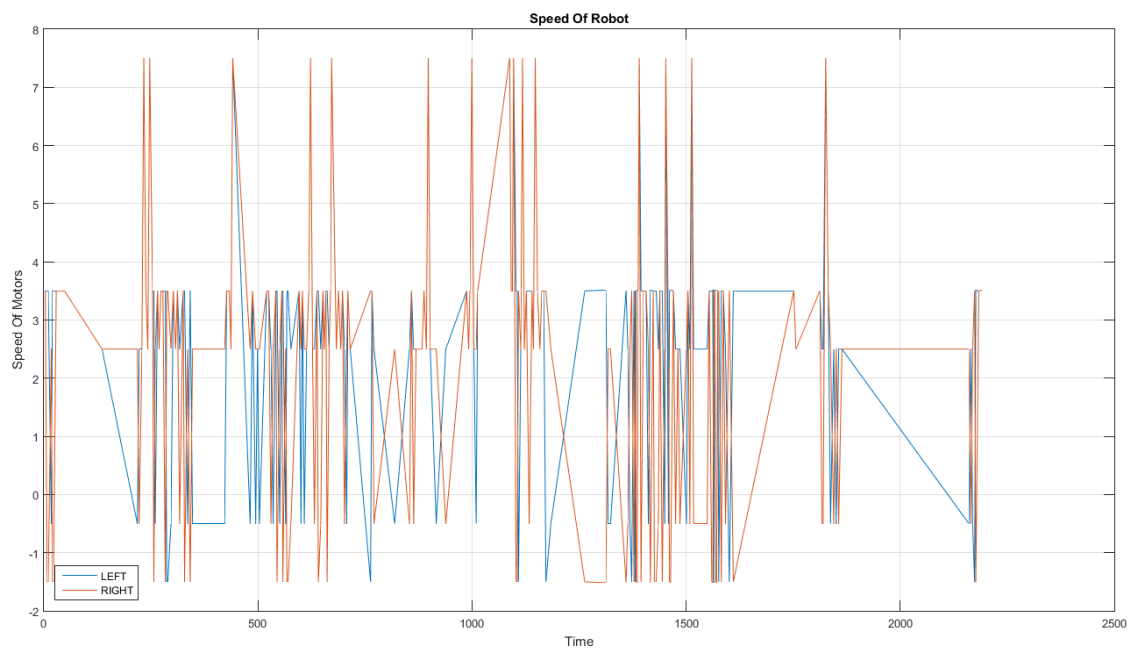
Σχήμα 4.92 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 20%



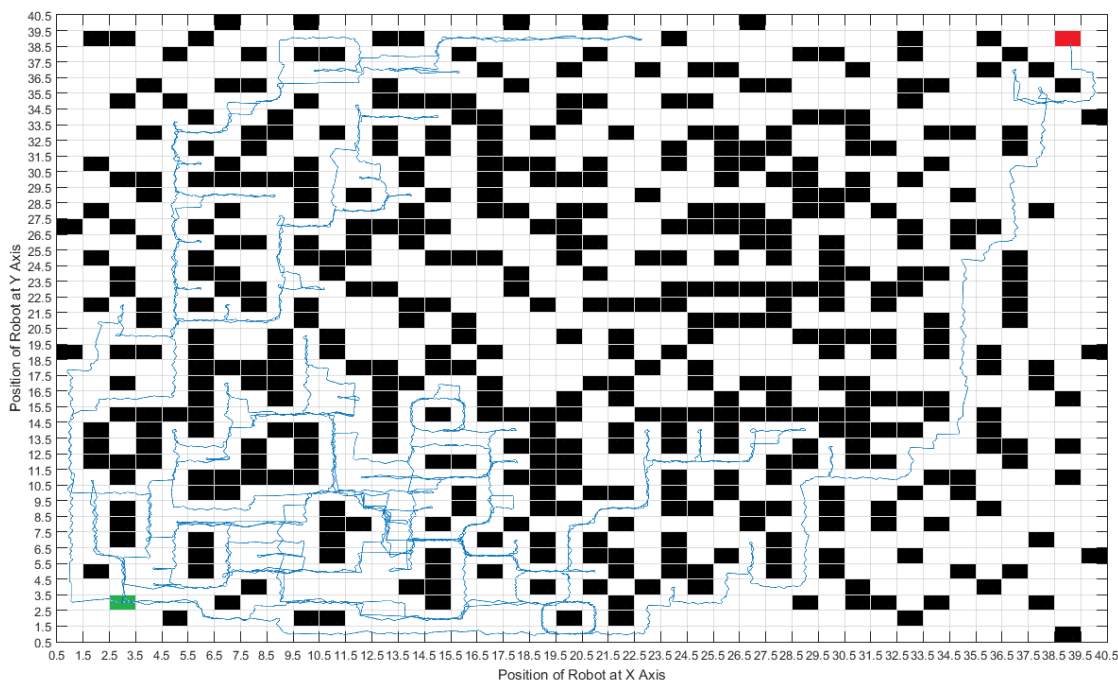
Σχήμα 4.93 Ταχύτητα τροχών με ασαφή ελεγκτή



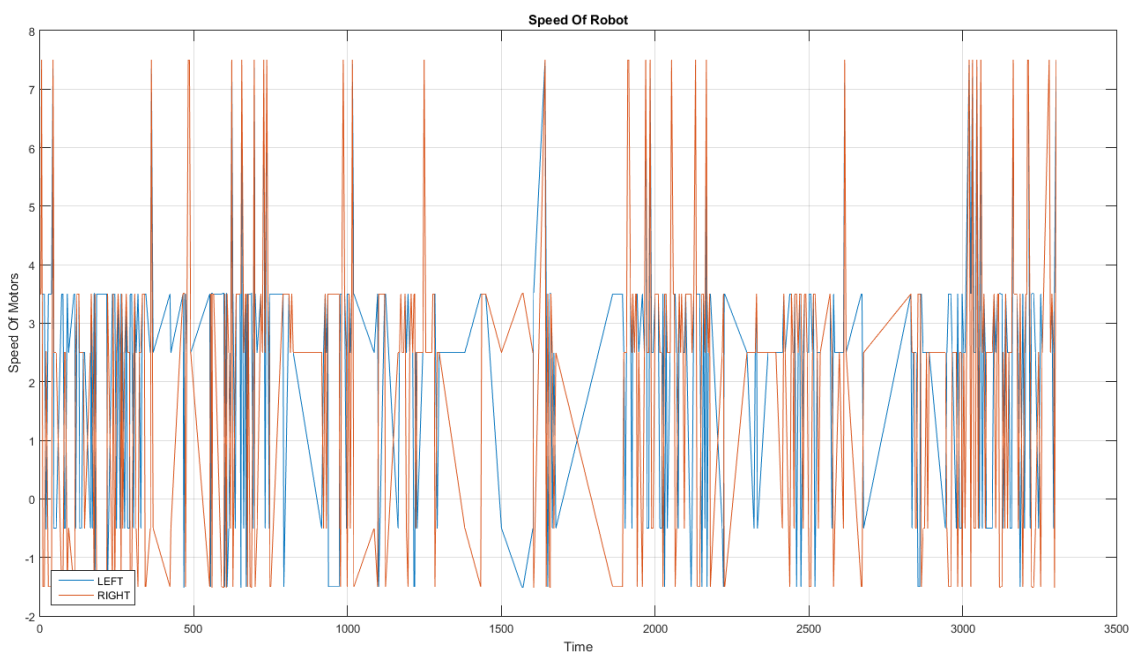
Σχήμα 4.94 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%



Σχήμα 4.95 Ταχύτητα τροχών με ασαφή ελεγκτή



Σχήμα 4.96 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%



Σχήμα 4.97 Ταχύτητα τροχών με ασαφή ελεγκτή

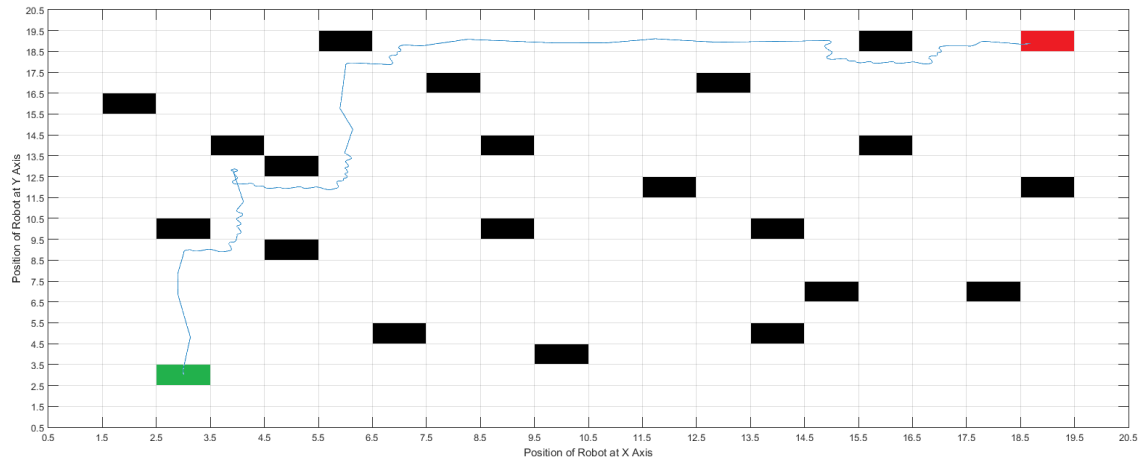
Αφού εκτελέστηκαν όλες οι προσομοιώσεις με τον ασαφή ελεγκτή και την χρήση του αλγορίθμου CIA* αναζήτηση της βέλτιστης λύσης καταγράφονται τα δεδομένα όπως ο χρόνος αναζήτησης, η απόσταση που διανύθηκε, τα κελιά που εξερευνήθηκαν από τον αλγόριθμο και η βέλτιστη λύση.

Αλγόριθμος CIA*					
<i>Degree of Difficulty</i>	<i>Arenas (Dimension)</i>	<i>Time (Seconds)</i>	<i>Distance (Meters)</i>	<i>Expanded Nodes (Cells)</i>	<i>Path Solution (Cells)</i>
5%	20x20	114.9243	52.8437	43	32
	40x40	557.3090	359.0384	190	72
10%	20x20	340.0153	136.6646	92	34
	40x40	736.1058	404.1796	221	72
15%	20x20	215.3547	75.9417	65	34
	40x40	790.0168	406.3233	219	74
20%	20x20	320.2612	121.8752	89	34
	40x40	1226.2265	576.0125	304	74
25%	20x20	374.1830	184.4831	85	36
	40x40	2192.2869	439.8742	205	76
30%	20x20	1514.3988	785.6076	191	40
	40x40	3304.5826	793.9789	385	78

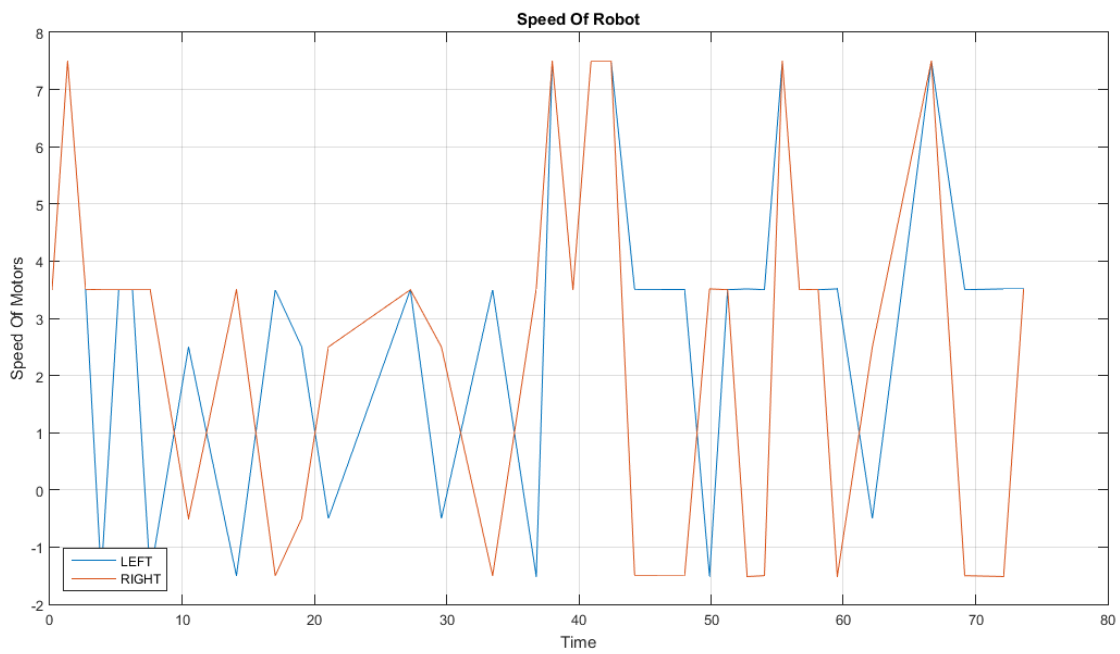
Πίνακας 4.6 Δεδομένα προσομοίωσης αλγορίθμου CIA* με ασαφή ελεγκτή

4.4.7 Αποτελέσματα προσομοίωσης αλγορίθμου Weighted A* με χρήση ασαφούς ελεγκτή

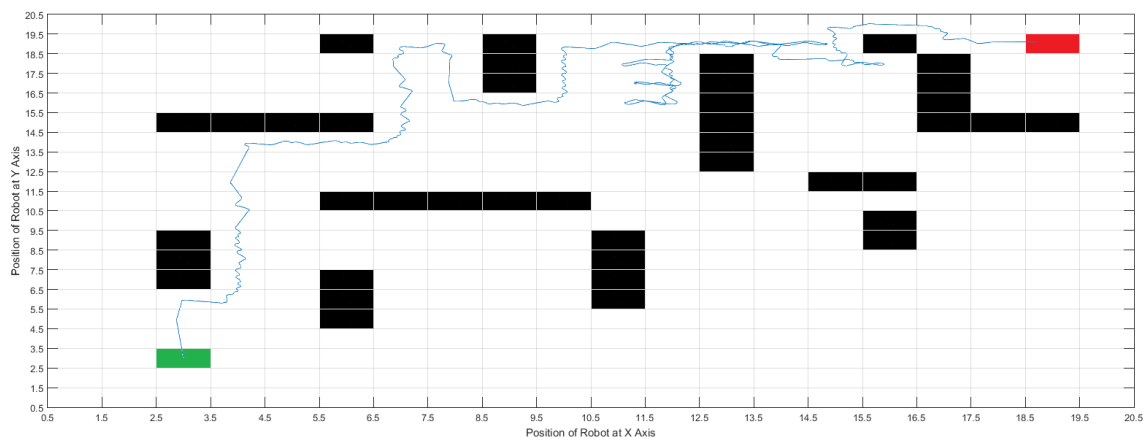
Τέλος, για την σύγκριση των δύο ελεγκτών που αναπτύχθηκαν, απαιτείται και η υλοποίηση του ασαφούς ελεγκτή με τον αλγόριθμο Weighted A*, για την αναζήτηση της βέλτιστης λύσης όμοια με τον προηγούμενο. Παρουσιάζονται και εδώ τα σχήματα με τις προσομοιώσεις που εκτελέστηκαν, με την διαδρομή και τις ταχύτητες του οχήματος.



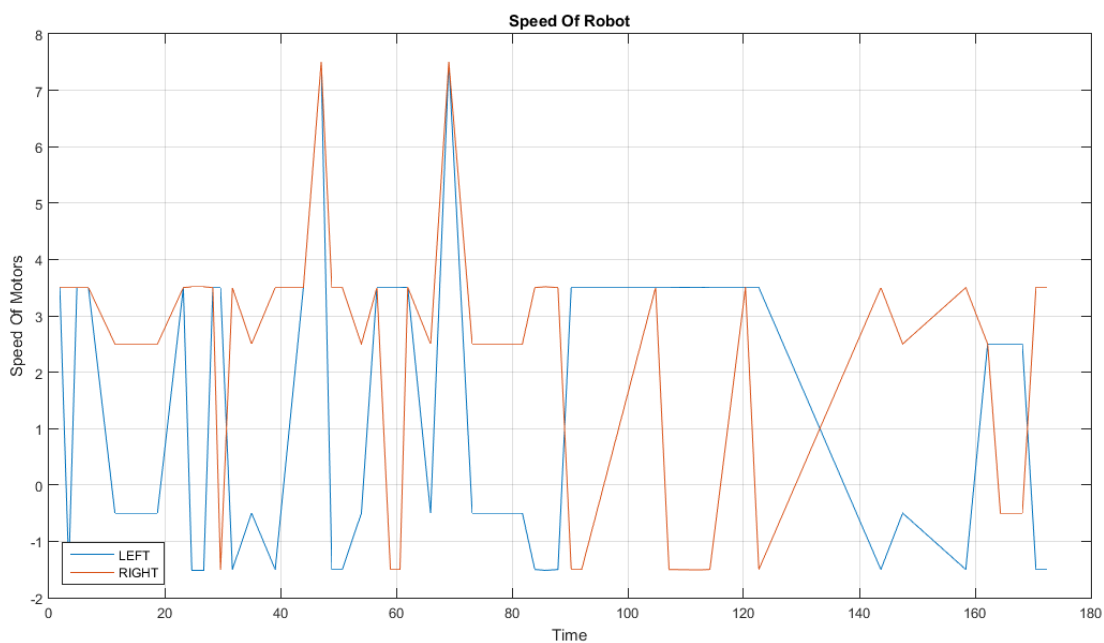
Σχήμα 4.98 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 5%



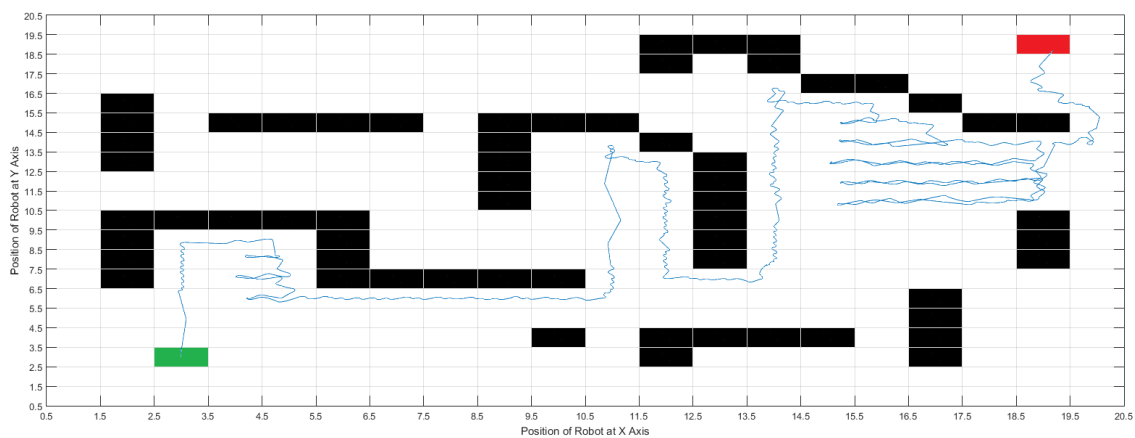
Σχήμα 4.99 Ταχύτητα τροχών με ασαφή ελεγκτή



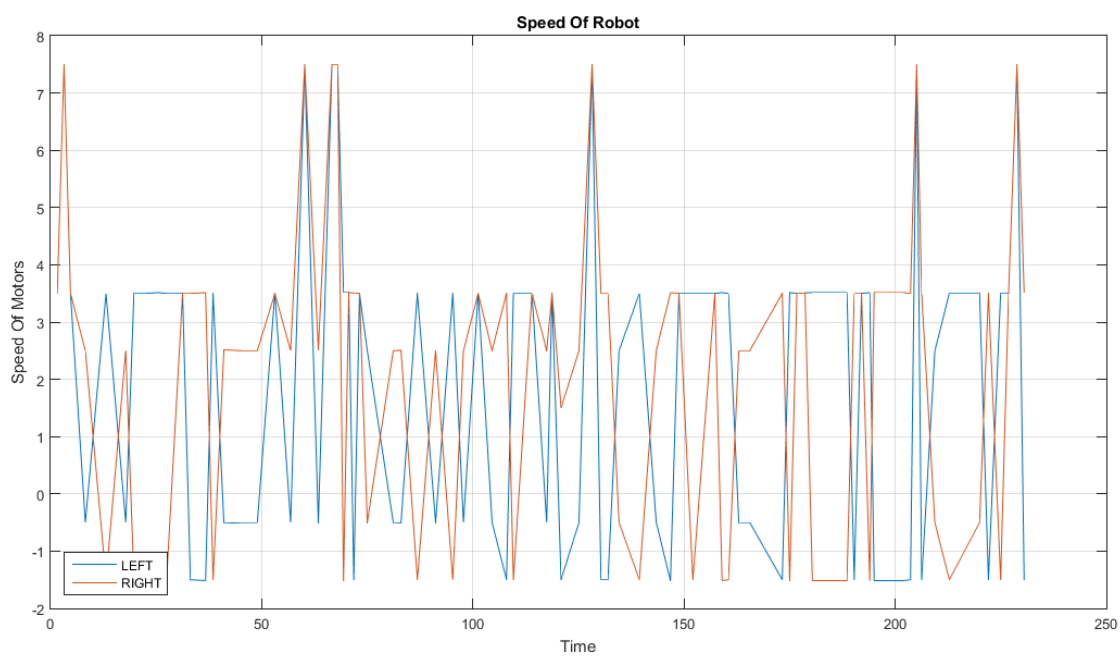
Σχήμα 4.100 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 10%



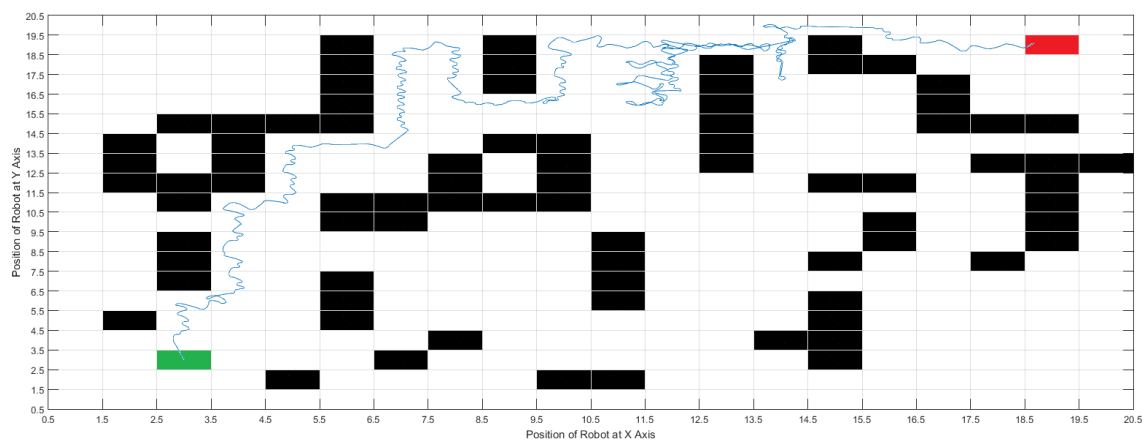
Σχήμα 4.101 Ταχύτητα τροχών με ασαφή ελεγκτή



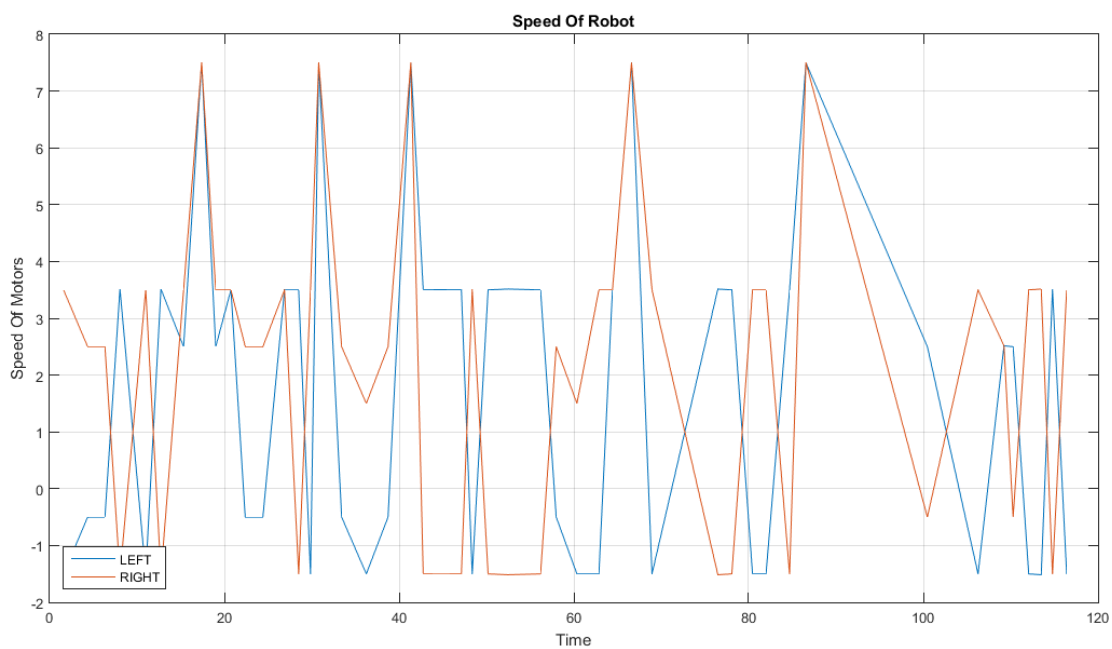
Σχήμα 4.102 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 15%



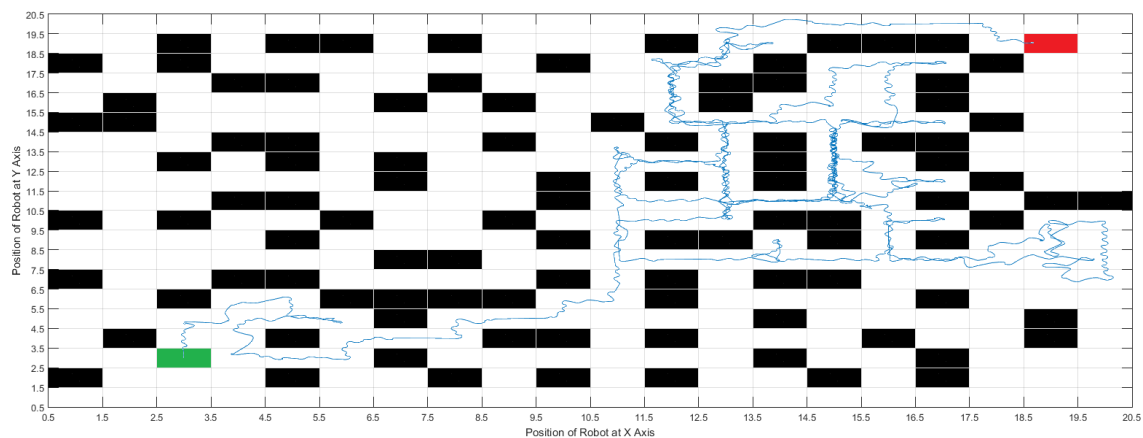
Σχήμα 4.103 Ταχύτητα τροχών με ασαφή ελεγκτή



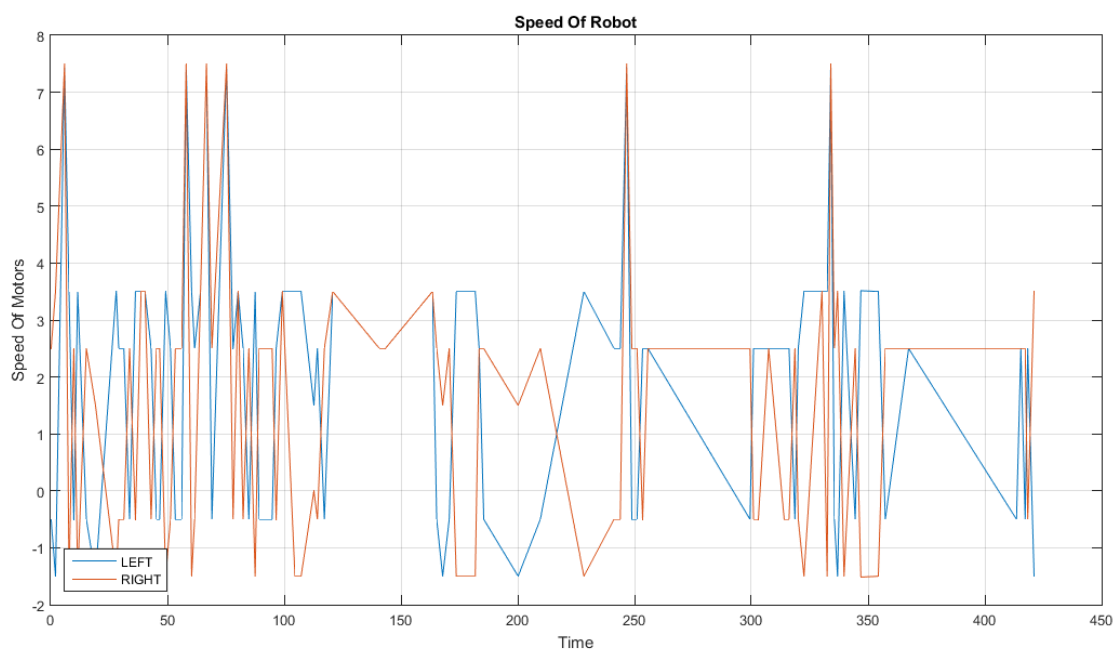
Σχήμα 4.104 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 20%



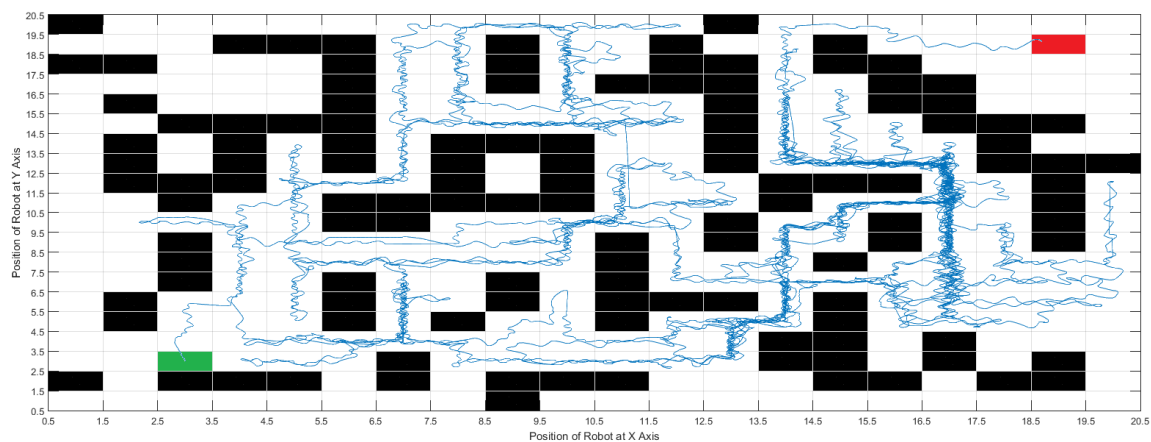
Σχήμα 4.105 Ταχύτητα τροχών με ασαφή ελεγκτή



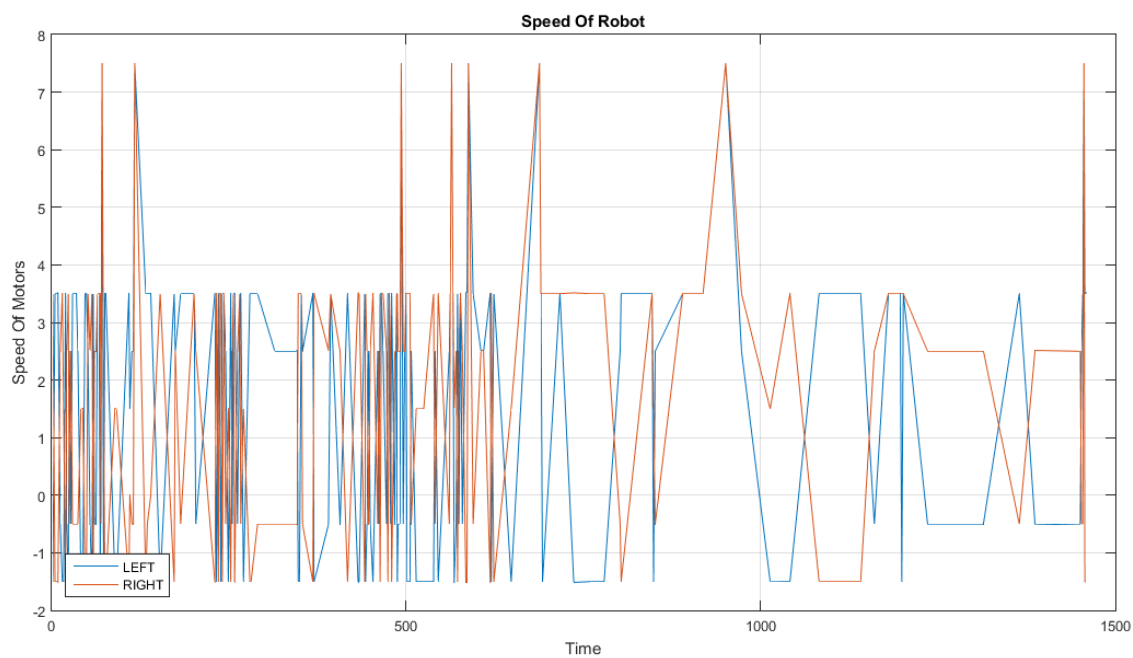
Σχήμα 4.106 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 25%



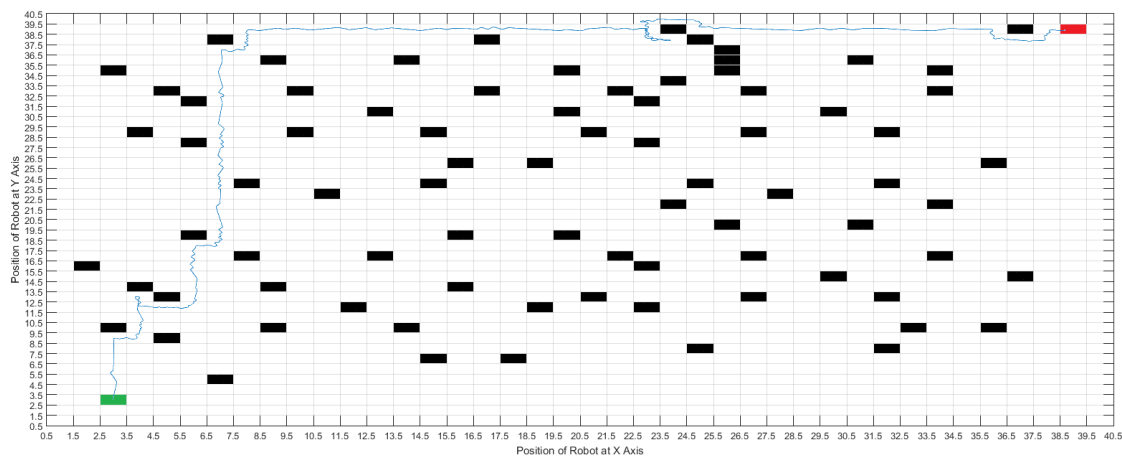
Σχήμα 4.107 Ταχύτητα τροχών με ασαφή ελεγκτή



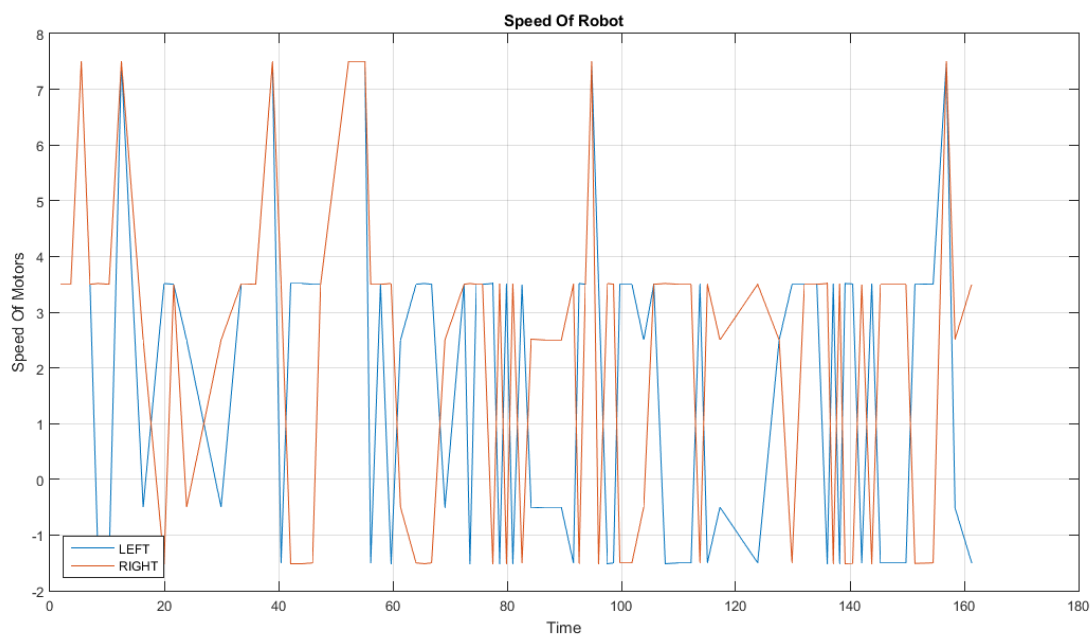
Σχήμα 4.108 Διαδρομή οχήματος σε αρένα 20x20 με βαθμό δυσκολίας 30%



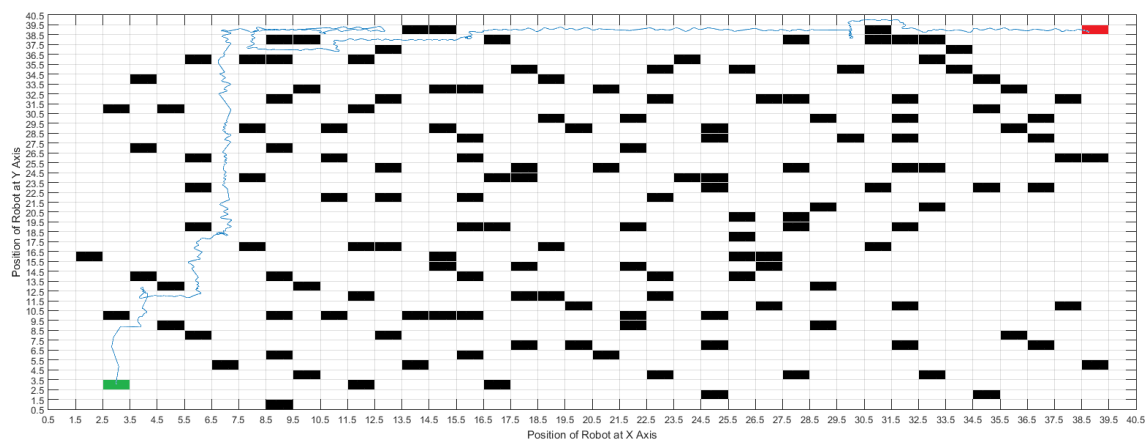
Σχήμα 4.109 Ταχύτητα τροχών με ασαφή ελεγκτή



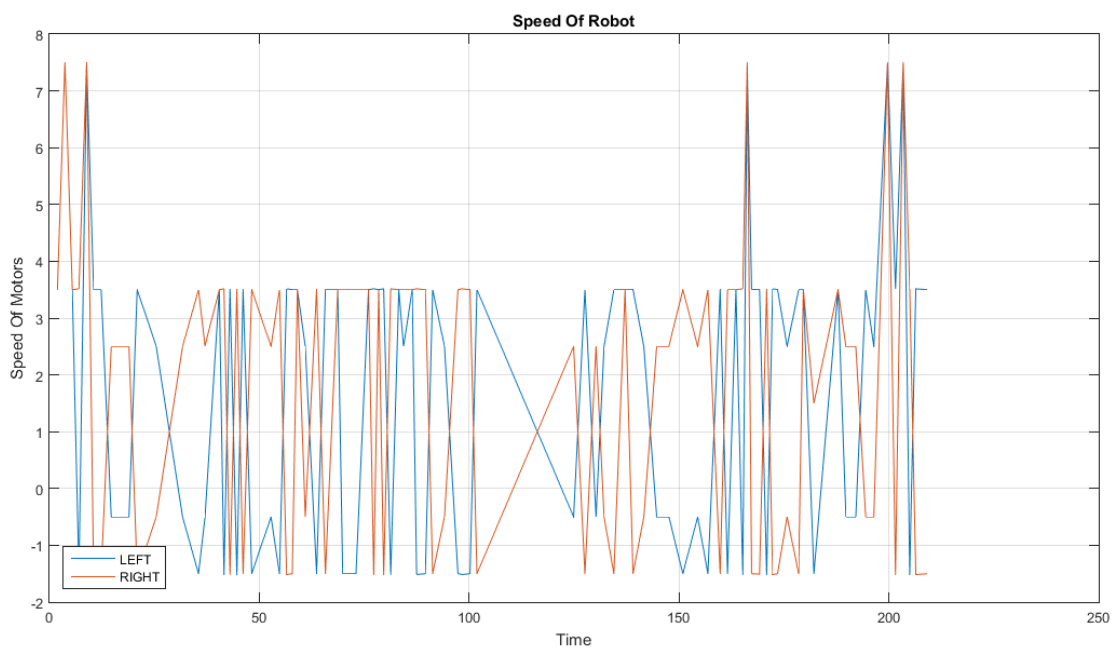
Σχήμα 4.110 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 5%



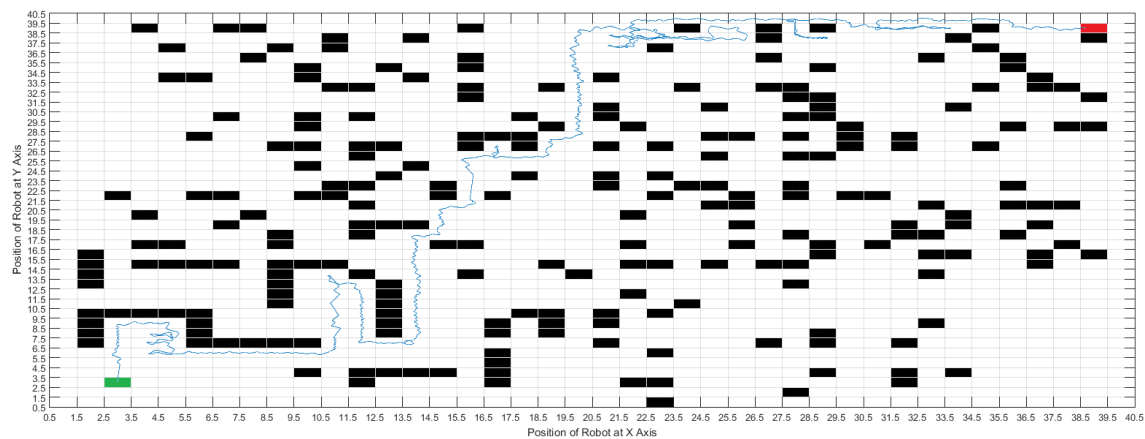
Σχήμα 4.111 Ταχύτητα τροχών με ασαφή ελεγκτή



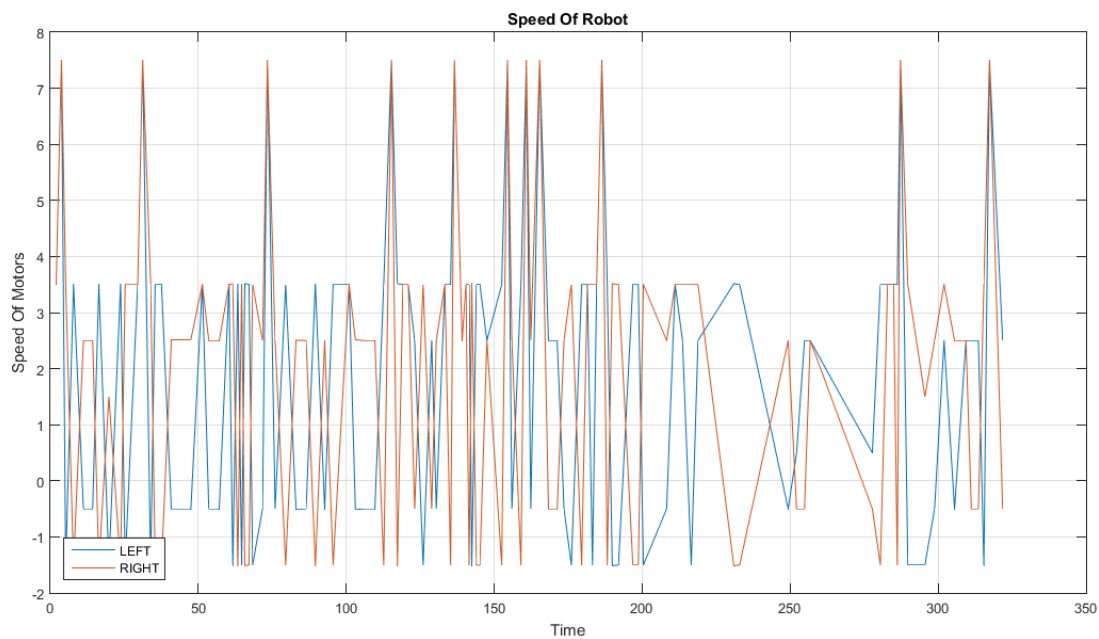
Σχήμα 4.112 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 10%



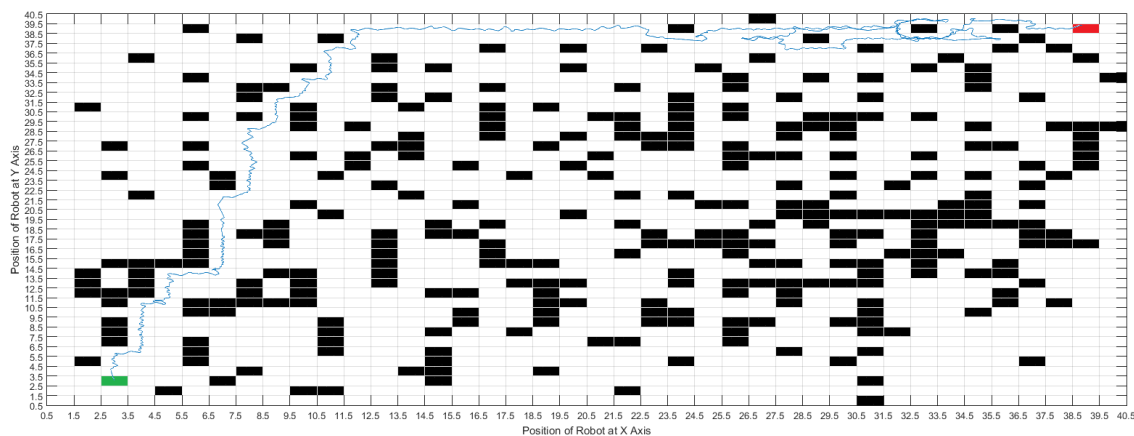
Σχήμα 4.113 Ταχύτητα τροχών με ασαφή ελεγκτή



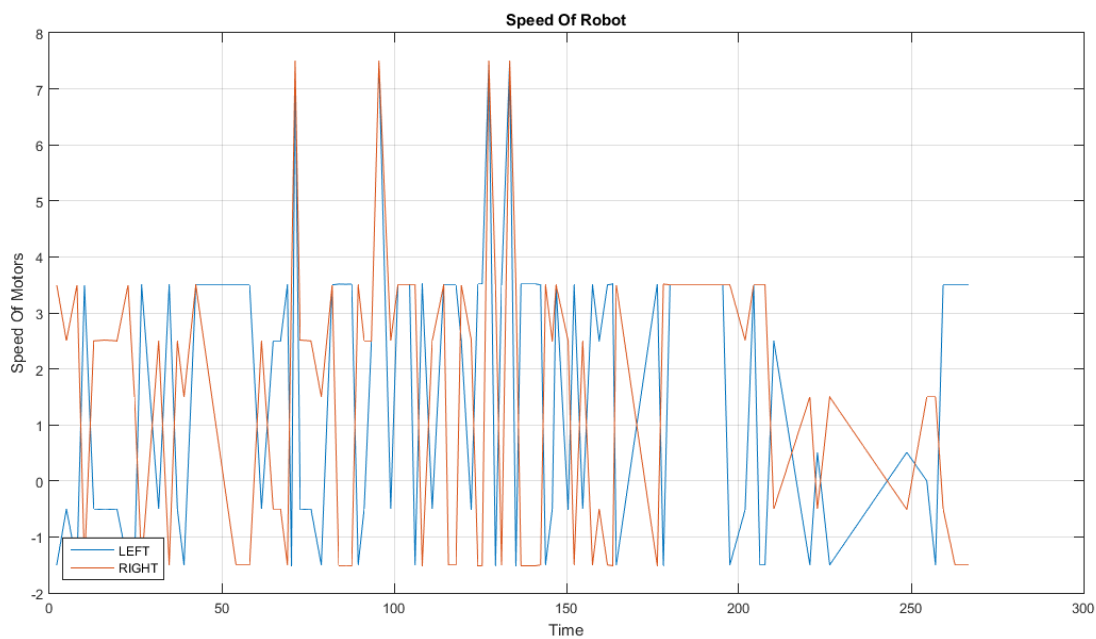
Σχήμα 4.114 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 15%



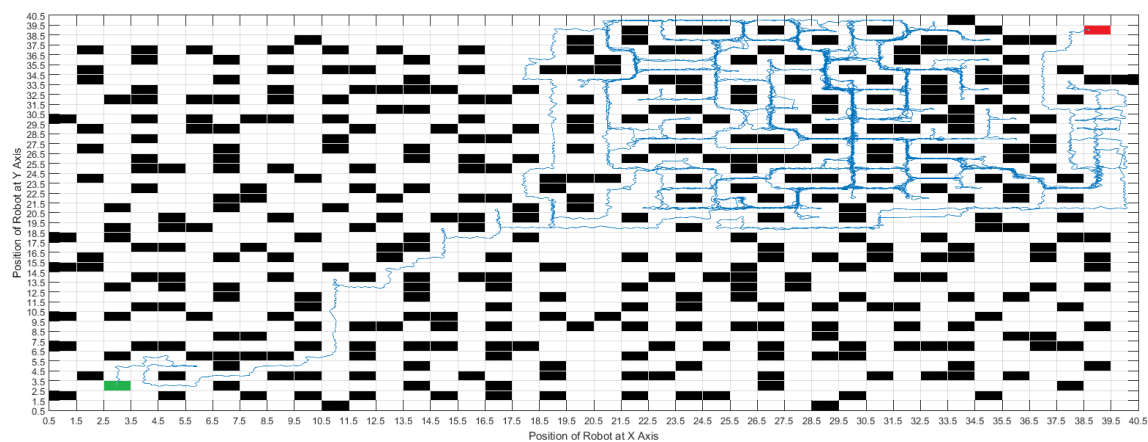
Σχήμα 4.115 Ταχύτητα τροχών με ασαφή ελεγκτή



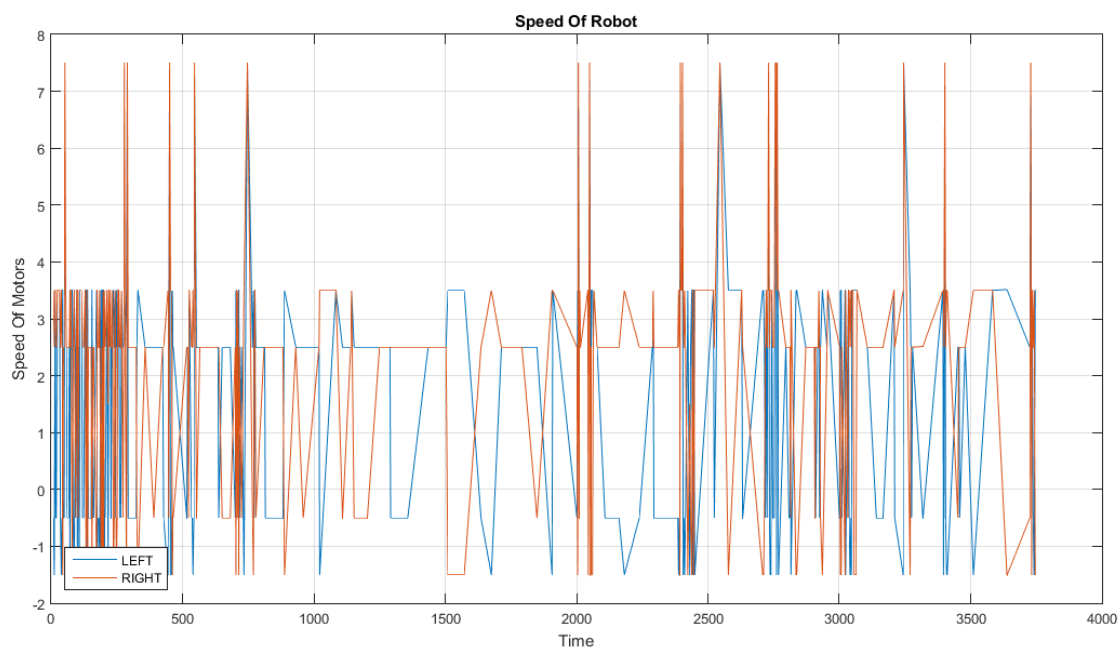
Σχήμα 4.116 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 20%



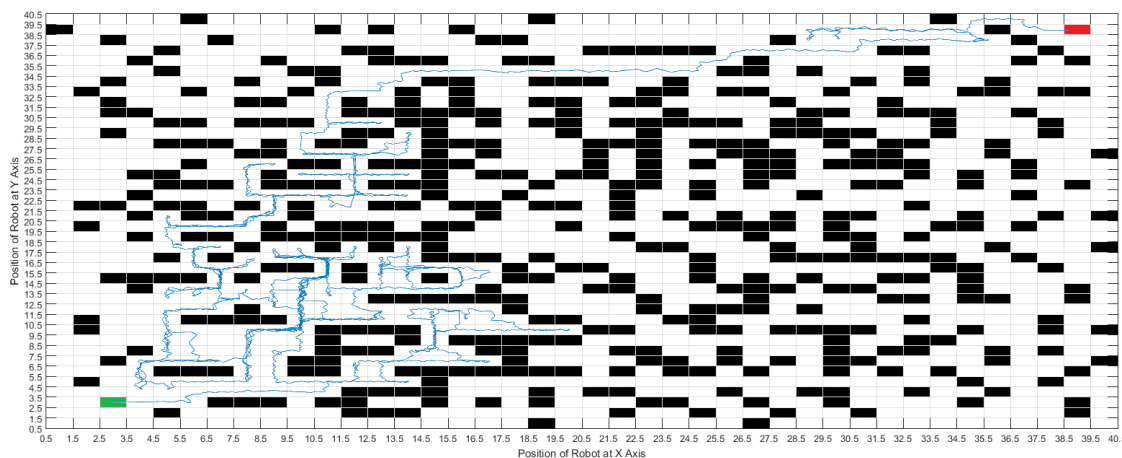
Σχήμα 4.117 Ταχύτητα τροχών με ασαφή ελεγκτή



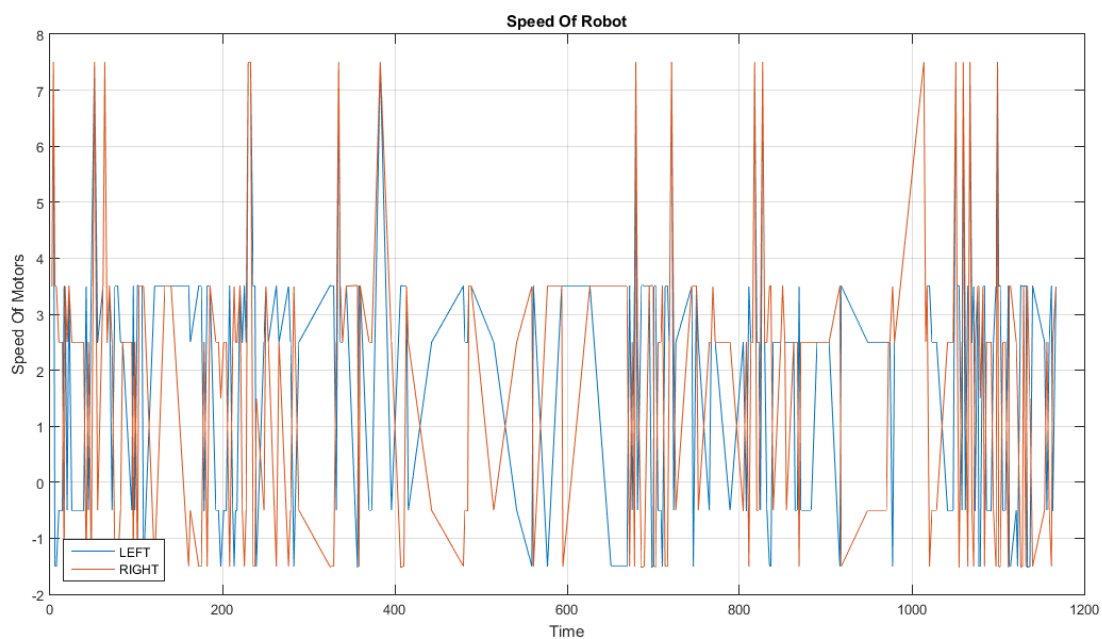
Σχήμα 4.118 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 25%



Σχήμα 4.119 Ταχύτητα τροχών με ασαφή ελεγκτή



Σχήμα 4.120 Διαδρομή οχήματος σε αρένα 40x40 με βαθμό δυσκολίας 30%



Σχήμα 4.121 Ταχύτητα τροχών με ασαφή ελεγκτή

Καθώς εκτελέστηκαν όλες οι προσομοιώσεις και για τον αλγόριθμο Weighted A* καταγράφονται οι χρόνοι αναζήτησης για κάθε αρένα, οι αποστάσεις που διανύθηκαν από το όχημα, το πλήθος των κελιών που επισκέφτηκαν και η λύση της βέλτιστης διαδρομής.

		Αλγόριθμος Weighted A*			
<i>Degree of Difficulty</i>	<i>Arenas (Dimension)</i>	<i>Time (Seconds)</i>	<i>Distance (Meters)</i>	<i>Expanded Nodes (Cells)</i>	<i>Path Solution (Cells)</i>
5%	20x20	74.9193	36.8004	36	34
	40x40	161.3073	96.4231	80	76
10%	20x20	172.3863	64.7215	49	34
	40x40	209.9094	109.8781	86	74
15%	20x20	230.5267	109.0693	80	34
	40x40	321.8013	159.4301	111	74
20%	20x20	116.4047	78.8945	48	34
	40x40	266.6828	129.4188	87	76
25%	20x20	422.6194	201.5261	91	36
	40x40	3745.9449	1436.7863	313	82
30%	20x20	1459.3472	843.8726	181	40
	40x40	1167.2624	518.0217	233	82

Πίνακας 4.7 Δεδομένα προσομοίωσης αλγορίθμου Weighted A* με ασαφή ελεγκτή

4.4.8 Σύγκριση των ελεγκτών κίνησης

Σκοπός της δημιουργίας των δύο ελεγκτών είναι η σύγκριση τους ως προς τη συμπεριφορά του ρομποτικού οχήματος κατά την κίνηση του στις αρένες που δημιουργήθηκαν. Η σύγκριση μεταξύ τους πραγματοποιείται με βάση τα δεδομένα που προκύπτουν από τις προσομοιώσεις, δηλαδή τον χρόνο που χρειάστηκε το όχημα να φτάσει στο τελικό σημείο και με τους δύο ελεγκτές και την απόσταση που διάνυσε. Στον παρακάτω πίνακα συγκεντρώνονται όλα τα δεδομένα που ανακτήθηκαν από την υλοποίηση των αλγορίθμων και για τους δύο ελεγκτές. Οι σχετικές συγκρίσεις παρουσιάζονται στους πίνακες 4.7-9.

A*		Χρόνος		Απόσταση		Ποσοστό Μεταβολής	
		Απλός Ελεγκτής	Ασαφής Ελεγκτής	Απλός Ελεγκτής	Ασαφής Ελεγκτής		
5%	20x20	99.6114	88.8975	50.1359	57.8201	-10.76%	-15.34%
	40x40	507.5222	841.1952	365.4653	371.8031	65.75%	1.73%
10%	20x20	729.3304	783.3135	380.0437	412.6172	7.4%	8.57%
	40x40	1028.2998	847.7304	437.1210	402.8748	-17.56	-7.84
15%	20x20	415.0394	467.5159	215.7916	237.7586	12.64%	10.18%
	40x40	2204.6825	3717.8682	1300.7822	1382.1295	68.64%	6.25%
20%	20x20	891.2219	733.2827	320.7817	295.0438	-17.72%	-8.02%
	40x40	4124.0329	5681.2705	1349.4612	1467.4141	37.76%	8.74%
25%	20x20	1190.7645	1774.8888	558.7510	573.3751	49.05%	2.62%
	40x40	11818.5600	13483.5033	2956.5142	2642.7515	14.09%	-10.61%
30%	20x20	2043.8902	2863.7983	1198.9065	1296.9985	40.12%	8.18%
	40x40	4054.0350	5696.4352	1340.6445	1318.1882	40.51%	-1,68%

Πίνακας 4.8 Σύγκριση ελεγκτών με τον αλγόριθμο A*

CIA*		Χρόνος		Απόσταση		Ποσοστό Μεταβολής	
		Απλός Ελεγκτής	Ασαφής Ελεγκτής	Απλός Ελεγκτής	Ασαφής Ελεγκτής		
5%	20x20	88.7575	114.9243	50.0420	52.8437	29.48%	5.6%
	40x40	653.4880	557.3090	309.6691	359.0384	-14.72%	15.94%
10%	20x20	253.5846	340.0153	130.3457	136.6646	34.08%	4.85%
	40x40	840.9225	736.1058	360.9367	404.1796	-12.46%	11.98%
15%	20x20	238.7933	215.3547	71.7348	75.9417	-9.82%	5.86%
	40x40	967.3234	790.0168	377.4094	406.3233	-18.33%	7.66%
20%	20x20	386.5406	320.2612	131.7803	121.8752	-17.15%	-7.52%
	40x40	1055.4075	1226.2265	537.6484	576.0125	16.19%	7.14%
25%	20x20	356.5200	374.1830	177.4869	184.4831	4.95%	3.94%
	40x40	1573.7991	2192.2869	442.1684	439.8742	39.30%	-0.52%
30%	20x20	1161.4909	1514.3988	702.7458	785.6076	30.38%	11.79%
	40x40	3161.9170	3304.5826	832.6980	793.9789	4.51%	-4.65%

Πίνακας 4.9 Σύγκριση ελεγκτών κατά τον CIA*

Weighted A*		Χρόνος		Απόσταση		Ποσοστό Μεταβολής	
		Απλός Ελεγκτής	Ασαφής Ελεγκτής	Απλός Ελεγκτής	Ασαφής Ελεγκτής		
5%	20x20	68.5171	74.9193	36.3360	36.8004	9.34%	1.28%
	40x40	184.8137	161.3073	83.3276	96.4231	-12.72%	15.72%
10%	20x20	143.3424	172.3863	61.8957	64.7215	20.26%	4.57%
	40x40	153.3801	209.9094	99.1296	109.8781	36.86%	10.84%
15%	20x20	221.4109	230.5267	101.3639	109.0693	4.12%	7.60%
	40x40	277.0056	321.8013	151.3291	159.4301	16.17%	5.35%
20%	20x20	170.5362	116.4047	66.8392	78.8945	-31.74%	18.04%
	40x40	194.7979	266.6828	109.2764	129.4188	36.90%	18.43%
25%	20x20	349.6965	422.6194	189.9450	201.5261	20.85%	6.10%
	40x40	3557.1092	3745.9449	1391.0112	1436.7863	5.31%	3.29%
30%	20x20	1186.1305	1459.3472	745.9919	843.8726	23.03%	13.12%
	40x40	1008.6864	1167.2624	515.3425	518.0217	15.72%	0.52%

Πίνακας 4.10 Σύγκριση ελεγκτών κατά τον Weighted A*

Όπως παρουσιάζεται από τα δεδομένα που καταγράφονται στους παραπάνω πίνακες ο συμβατικός ελεγκτής κατά μεγάλο ποσοστό εκτελεί τη διαδικασία αναζήτησης βέλτιστης λύσης σε μικρότερους χρόνους με ποσοστό που φτάνει περίπου 70% πιο γρήγορα στο τελικό στόχο. Η μεταβολή αυτή, στον χρόνο, τείνει να εξισωθεί με την απόσταση που διανύει το όχημα με τον ασαφή ελεγκτή. Σε περιπτώσεις στις οποίες η αναζήτηση με τον συμβατικό ελεγκτή επιτυγχάνεται σε μικρότερο χρόνο, η απόσταση η οποία διανύεται είναι περίπου ίδια ή και μικρότερη σε αυτή με τον ασαφή ελεγκτή. Αυτή η διαφορά είναι εμφανής και στην κίνηση του οχήματος καθώς με τον ασαφή ελεγκτή η κίνηση παρουσιάζεται πιο ομαλή ως προς τις στροφές που πραγματοποιεί για την αποφυγή εμποδίων, σε σχέση με τον συμβατικό που οι κινήσεις του οχήματος είναι πιο απότομες. Λόγω του ότι ο ασαφής ελεγκτής δημιουργήθηκε με κανόνες τέτοιους ώστε να είναι όμοιος με τον συμβατικό ελεγκτή, η διαφορά στις τιμές αυτές είναι αναμενόμενη. Ένας καλύτερος συντονισμός του ασαφούς ελεγκτή ως προς την συνάρτηση συμμετοχής, την καλύτερη απόδοση κανόνων αλλά και την κατανομή των εισόδων θα παρουσίαζε σημαντική βελτίωση στην απόδοση του σε σχέση με αυτή του συμβατικού ελεγκτή που υλοποιήθηκε.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Σκοπός της παρούσας εργασίας, ήταν να μελετηθεί η απόδοση και τρόπος λειτουργίας τριών ευρετικών αλγορίθμων για τον υπολογισμό της βέλτιστης διαδρομής ενός ρομποτικού οχήματος που κινείται σε ένα άγνωστο χώρο με την ύπαρξη εμποδίων. Ο τοπικός έλεγχος του ρομποτικού οχήματος και αλληλεπίδραση με το περιβάλλον, πραγματοποιούνται με την χρήση ενός απλού λογικού ελεγκτή ή με την χρήση ενός ασαφούς ελεγκτή.

Χρησιμοποιήθηκε ο προσομοιωτής CoppeliaSim και προσομοιώθηκε το ρομποτικό όχημα Pioneer 3-DX. Σχεδιάστηκαν διαφορετικά περιβάλλοντα πειραματισμού με διαφορετικές διαστάσεις και πυκνότητα εμποδίων.

Συγκρίνοντας τους δύο ελεγκτές από τα αποτελέσματα που εξάγονται για την συμπεριφορά του οχήματος ως προς τον χρόνο, την απόσταση φαίνεται πως ο συμβατικός ελεγκτής εκτελεί την διαδικασία της αναζήτησης σε αρκετά μικρότερους χρόνους. Σε ορισμένες περιπτώσεις όμως, όπου ο ασαφής ελεγκτής είναι πιο αργός από τον συμβατικό, φαίνεται ότι η απόσταση που διανύει είναι περίπου ίδια ή και μικρότερη. Αυτό οφείλεται στην μεγαλύτερη ταχύτητα και στην ομαλότερη κίνηση του οχήματος κατά τον ασαφή ελεγκτή όπως φαίνεται κατά την διάρκεια της προσομοίωσης. Ο

συγκεκριμένος ασαφής ελεγκτής δημιουργήθηκε με την λογική να προσομοιώνει όσο καλύτερα τον συμβατικό ελεγκτή που αναπτύχθηκε.

Στο δεύτερο κομμάτι της εργασίας πραγματοποιήθηκε η σύγκριση μεταξύ των τριών αλγορίθμων. Ο κλασσικός αλγόριθμος A^* καθώς και η καλύτερη παραλλαγή του, CIA^* , εγγυούνται την εύρεση της βέλτιστης διαδρομής σε ένα άγνωστο περιβάλλον. Από τα αποτελέσματα που εξάγονται κατά την υλοποίηση τους, φαίνεται πως ο CIA^* είναι αρκετά πιο αποδοτικός ως προς τον χρόνο και την απόσταση που απαιτείται το όχημα να εκτελέσει την αναζήτηση. Σε αντίθεση με τους δύο πρώτους κατά την υλοποίηση του WA^* , η αναζήτηση της βέλτιστης λύσης επιτυγχάνεται σε μικρότερο χρόνο από τους δύο προηγούμενους αλγορίθμους. Η επιλογή του βάρους για τον WA^* έγινε έτσι ώστε η αναζήτηση για την εύρεση της βέλτιστης λύσης να απαιτεί τον ίδιο ή μικρότερο χρόνο. Η μεταβολή όμως αυτή στον χρόνο αναζήτησης δεν εγγυάται τη εύρεση της βέλτιστης λύσης. Έτσι ο αλγόριθμος CIA^* παρ' όλο που απαιτεί περισσότερο χρόνο για την εύρεση της βέλτιστης διαδρομής φαίνεται να είναι ο αποδοτικότερος αλγόριθμος ευρετικής αναζήτησης.

5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η παρούσα εργασία αποτελεί μια μελέτη περίπτωσης, κυρίως όσον αφορά την απόδοση των αλγορίθμων σε ένα άγνωστο περιβάλλον όπου υπάρχουν στατικά εμπόδια. Στο μέλλον θα διερευνηθεί η επίδραση δυναμικών εμποδίων, καθώς και αναμένεται να υλοποιηθεί η συγκεκριμένη προσέγγιση σε πραγματικά ρομποτικά οχήματα. Ακόμα αξίζει να διερευνηθεί και μια παραλλαγή της προτεινόμενης προσέγγισης για τη περίπτωση πολλαπλών οχημάτων, όπου θα υπάρχει ανταλλαγή πληροφορίας σχετικά με το περιβάλλον και δεν θα βασίζονται τα ρομποτικά οχήματα μόνο στους τοπικούς αισθητήρες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ι. Α. Θεοδώρου, Εισαγωγή στην Ασαφή λογική, σελ.23-46, Εκδόσεις Τζίολα, 2010.
- [2] Σ. Παπαδάκης, Π. Αδαμίδης. “Ασαφή Συστήματα Θεωρία και Εργαστηριακές Ασκήσεις”, ΤΕΙ Θεσσαλονίκης, σελ. 5-10, 2004.
- [3] Differential Drive Robots, from Dudek and Jenkin, Computational Principles of Mobile Robotics.
(Ηλεκτρονική πηγή: <https://www.cs.columbia.edu/~allen/F19/NOTES/icckinematics.pdf>)
- [4] N.Tsourveloudis, L. Doitsidis, K. Valavanis, “Autonomous Navigation of Unmanned Vehicles: A Fuzzy Logic Perspective,” Cutting Edge Robotics, pp. 291-310, Pro Literatur Verlag, ISBN:3-86611-038-3, 2005.
- [5] L. Doitsidis, N. C. Tsourveloudis, S. Piperidis, “Evolution of fuzzy controllers for robotic vehicles: The role of fitness function selection,” Journal of Intelligent and Robotics Systems, vol. 56, no.4, pp. 469-484, 2009.
- [6] G. Eleftheriou, L. Doitsidis, Z. Zinonos, S. Chatzichristofis, “A Fuzzy Rule-Based Control System for Fast Line-Following Robots,” in proceedings of the 2nd International Workshop on Intelligent Systems for the Internet of Things, pp. 388-395, 2020.
- [7] M.Khaoula, N.Malek and J.Mohamed, “Free navigation and obstacle avoidance based on fuzzy controller”, in Proceedings of the 2nd World Congress on Computer Application and Information Systems, N&N Global Technology, Hammamet, Tunisia, January 2015.
- [8] František Duchon, Andrej Babinec, Martin Kajan, Peter Beno, Martin Florek, Tomáš Fico, Ladislav Jurišica, ”Path planning with modified A star algorithm for a mobile robot,” Modelling of Mechanical and Mechatronics Systems MMaMMS 2014, Procedia Engineering vol. 96, pp. 59-69, 2014.
- [9] Rios L.H.O., Chaimowicz L. (2010) A Survey and Classification of A* Based Best-First Heuristic Search Algorithms. In: da Rocha Costa A.C., Vicari R.M., Tonidandel F. (eds) Advances in Artificial Intelligence – SBIA 2010. SBIA 2010.
- [10] Eric A. Hansen, Rong Zhou, ”Anytime Heuristic Search”, Journal of Artificial Intelligence Research 28, 2007, pp. 267-269.

- [11] A. C. Kapoutsis, C. M. Malliou, S. A. Chatzichristofis and E. B. Kosmatopoulos, "Continuously informed heuristic A*-optimal path retrieval inside an unknown environment", 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), 2017, pp. 216-222.