ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ

ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ /
TECHNICAL
UNIVERSITY
OF CRETE

HMMY189: Αναφορά Αυτόνομοι Πράκτορες

Εξαμινιαία Εργασία:
Αυτόνομη Ρομποτική Σκούπα: Rooby

Φοιτητής :
Σκλάβος Παναγιώτης 2018030170

Διδάσκων :
Μιχαήλ Λαγουδάκης

Πίνακας Περιεχομένων

1. Εισαγωγή

Στα πλαίσια του μαθήματος Αυτόνομοι Πράκτορες εκπονήθηκε η εργασία σχεδίασης ενός πράκτορα Αυτόνομης Ρομποτικής Σκούπας, η οποία έγγειται στο θεματικό πλαίσιο Path Planning. Συγκεκριμένα επιχειρήθηκε να υλοποιηθεί μια ηλεκτρική σκούπας, με λειτουργικότητα αντίστοιχη σε εκείνη που απαντάται στις σκούπες εμπορικής χρήσης που έχουν αναπτυχθεί και έχουν πλημμυρίσει την αγορά τα τελευταία χρόνια. Ο Πράκτοράς που υλοποιήθηκε για να ικανοποιήσει τις ζητούμενες απαιτήσεις αξιοποιεί ένα συνονθύλευμα διάφορων path-planning αλγορίθμων οι οποίοι ξεχωρίζουν ο καθένας σε διαφορετικές συνθήκες και επιλύουν όλα τα πιθανά προβλήματα που μπορούν να απαντηθούν σε ένα δεδομένο περιβάλλον.

2. World

Ο κόσμος που χρησιμοποιήθηκε αποτελεί ένα χώρο προσομοίωσης αντίστοιχο με του πρωτότυπου της iRobot Create για την μοντελοποίηση της σκούπας Roumba της ομώνυμης εταιρείας που υπάρχει ήδη στο περιβάλλον του Webots. Δοκιμάστηκαν 3 διαφορετικές παραλλαγές του περιβάλλοντος αυτού, ώστε να μπορέσουμε να συγκρίνουμε τα αποτελέσματα σε διαφορετικές συνθήκες. Συγκεκριμένα, επιχειρήθηκε καθαρισμός του χώρου χωρίς εμπόδια (Εικόνα 1), καθαρισμός με περιορισμένα εμπόδια (Εικόνα 2), και τέλος πλήρως επιπλωμένης κατοικίας (Εικόνα 3). Συνεπώς ελέγχθηκαν όλες οι δυνατές εκδοχές που θα μπορούσε να συναντήσει ο πράκτορας, ώστε να ελεγχθεί η αποτελεσματικότητά του ακόμα διεξοδικά.

Εικόνα 2.1: Άδειο Δωμάτιο

Εικόνα 2.2: Ελαφρώς επιπλωμένο δωμάτιο

Εικόνα 2.3: Πλήρως επιπλωμένο δωμάτιο

Και στις Τρείς παραπάνω περιπτώσεις το ζητούμενο είναι ίδιο: Η απολύμανση του χώρου ξεκινώντας από κάποιο σημείο στην κάτω πλευρά, όπως φαίνεται στις εικόνες. Επίσης ο χώρος και στις τρείς περιπτώσεις έχει διαστάσεις 10m x 10m συνεπώς, πειραματιζόμαστε σε μια επιφάνεια 100 τ.μ. Η σκούπα θα εκκινείται από την αφετηρία της και θα πρέπει να καθαρίσει τόσο τους κενούς χώρους όσο και εκείνους που γίνονται δυσκολότερα προσβάσιμοι λόγω των διαφόρων επίπλων.

Ιδιαίτερο χαρακτηριστικό του κόσμου αυτού, και ένας από τους κύριους λόγους που προτιμήθηκε, αποτελεί το γεγονός ότι έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι εμφανές ότι το έδαφος είναι βρώμικο. Όποτε μπορούμε εύκολα να παρατηρήσουμε την σκόνη καθώς και την απώλειά της αντίστοιχα, όταν η σκούπα περνάει από πάνω καθαρίζοντας την επιφάνεια. Αυτό επιτυγχάνεται μέσω ενός supervisor με ονομασία ground.c ο οποίος διαχειρίζεται την εμφάνιση του εδάφους. .

3. Vacuum Cleaner Model

Εικόνα 3.1: Το ρομπότ iRobot Create της IRobot που βασισμένο στο εμπορικό μοντέλο Roumba της εταιρείας.

3.α. Αισθητήρες και Εξαρτήματα

Το μοντέλο σκούπας που χρησιμοποιήθηκε αξιοποιεί πληθώρα αισθητήρων και λοιπών εξαρτημάτων τα οποία καθιστούν δυνατή την κίνησή της στον χώρο καθώς και της παρέχουν την δυνατότητα να αντιλαμβάνεται τυχόν εικονικά (virtual) ή υπαρκτά εμπόδια. Στην περίπτωσή μας ωστόσο χρησιμοποιούνται μόνο οι αισθητήρες για υπαρκτά εμπόδια καθώς δεν υλοποιήθηκε διεπαφή με τον χρήστη ώστε να μπορεί να κατευθύνει την λειτουργία του πράκτορα χειροκίνητα. Παρακάτω παρατίθεται μια απογραφή των αισθητήρων και εξαρτημάτων που βοηθούν τόσο σε αυτόνομη όσο και manual λειτουργία.

ΣΥΣΚΕΥΗ	***ΠΛΗΘΟΣ***	***ΛΕΙΤΟΥΡΓΙΑ***
LEDs	3	Ενδείξεις λειτουργίας
Motors	2	Μετακίνηση του ρομπότ στον χώρο μέσω της Δεξιάς και Αριστερής ρόδας
Position Sensors	2	Θέση Δεξιά και αριστερής ρόδας
Touch Sensors-Bumpers	2	Ανίχνευση σύγκρουσης με εμπόδιο
Distance Sensors	4	Cliff Sensors: Απόσταση από το έδαφος για ανίχνευση υψώματος και αποφυγή πτώσης
Receiver	1	Για χειροκίνητη ένδειξη εμποδίου (virtual walls)

Πίνακας 3.1: Πίνακας αισθητήρων και εξαρτημάτων

Αξίζει να σημειωθεί ότι στις πρώτες προσπάθειες προγραμματισμού του πράκτορα, προστέθηκαν στο ρομπότ 4 αισθητήρες απόστασης με σκοπό την προσέγγιση του τρόπου λειτουργίας του ρομποτικού μοντέλου e-puck. Ωστόσο αργότερα αφαιρέθηκαν καθώς δεν χρησιμοποιούνται αντίστοιχοι αισθητήρες από άλλες ρομποτικές σκούπες του εμπορίου συνεπώς η προσθήκη τους θα ήταν μη αναγκαία, το οποίο σημαίνει αχρείαστα κατασκευαστικά έξοδα. Επίσης σημαντική θα ήταν η προσθήκη κάμερας για την διευκόλυνση υλοποίησης λειτουργικότητας SLAM.

Σε κάθε περίπτωση για να γίνει εφικτή η προσθήκη οποιουδήποτε νέου εξαρτήματος στο ρομποτικό μοντέλο iRobot Create μέσω του προσομοιωτή, ή γενικότερα η επεξεργασία του, θα πρέπει πρώτα να γίνει η μετατροπή του σε Base Node κάνοντας δεξί κλικ πάνω του στο menu ένδειξης όλων το χρησιμοποιημένων μοντέλων στο Webots. Με τον τρόπο αυτό στο Base Node μπορούμε να κάνουμε add οποιοδήποτε άλλου Node στο κατάλληλο σημείο κάτω από την κατηγορία children.

3.β. Μηχανισμός-Λειτουργικότητα

Όλα τα εξαρτήματα που καταγράφηκαν παραπάνω, βρίσκονται πάνω στην σκούπα διεκπεραιώνοντας απαραίτητες εργασίες για την αποτελεσματική λειτουργία της. Πιο Συγκεκριμένα:

- Το Ρομπότ που χρησιμοποιείται για τους σκοπούς της εργασίας, με το πρώτο LED σηματοδοτεί το αν λειτουργεί ή όχι (POWER ON/OFF), ενώ με το δεύτερο LED σηματοδοτεί το αν καθαρίζει την δεδομένη στιγμή.
- Τα δύο Motors αξιοποιούνται προκειμένου να περιστρέφονται οι δύο τροχοί που βρίσκονται στο κάτω μέρος του αμαξώματος της σκούπας για να μετακινείται στον κόσμο. Σημαντικό είναι η ταχύτητα που προσδίδεται στην σκούπα να μην ξεπερνάει το οριακό επίπεδο 16 που δίνεται από τις προδιαγραφές καθώς σε τέτοιες περιστάσεις είναι πολύ πιθανόν να προκύψουν προβλήματα με τις αρθρώσεις της σκούπας.
- Οι Position Sensors χρησιμοποιούνται ώστε να προσδιορίζεται η θέση των τροχών σε κάποια δεδομένη στιγμή. Με τον τρόπο αυτό μπορούμε συγκρίνοντας την θέση των τροχών δύο διακριτές στιγμές, να προσδιορίσουμε το πόσο και το προς τα πού μετακινήθηκε η σκούπα.
- Οι δύο Touch Sensors-Bumpers αποτελούν ζωτικής σημασίας για την λειτουργία της Ρομποτικής σκούπας καθώς μέσω αυτών ανιχνεύονται τα εμπόδια που υπάρχουν στο περιβάλλον, παρέχοντας μια ένδειξη για την ύπαρξη σύγκρουσης σε αριστερό ή δεξί μέρος της σκούπας αντίστοιχα.
- Οι Distance Sensors χρησιμοποιούνται για την ανίχνευση cliffs (γκρεμών), κατά την κίνηση του ρομπότ. Συγκεκριμένα ελέγχεται η απόσταση από το έδαφος σε 4 διαφορετικά σημεία στο εμπρόσθιο τμήμα της σκούπας, αν η ένδειξη που επιστρέφεται είναι μικρότερη από κάποιο αποδεκτό threshold, σημαίνει ότι η σκούπα πρόκειται να πέσει σε κάποιο κενό αν συνεχίσει απαράλλακτα την λειτουργία της. Τα cliffs λαμβάνουν ίδια αντιμετώπιση με τα φυσικά εμπόδια που ανιχνεύονται από τους bumpers.
- Ο Receiver, αν και δεν χρησιμοποιείται στην παρούσα υλοποίηση, δέχεται μηνύματα από κάποιον Emitter που χειρίζεται ο χρήστης (π.χ μέσω διεπαφής σε κινητό). Με τον τρόπο αυτό η σκούπα αντιλαμβάνεται τους περιορισμούς που θέτει ο χρήστης αναγνωρίζοντάς τους ως virtual walls.

4. Supervisor: vacuumCleaner

4.α. Δομή Supervisor

Για την υλοποίηση του Πράκτορα της Rooby έγινε δημιουργία ενός νέου controller με ονομασία vacuumCleaner. Χρήσιμες ήταν οι ήδη υπάρχουσες γνώσεις σε controllers, που είχαμε αποκομίσει από την εμπειρία μας στο εργαστήριο με τον controller Rat για το e-puck, καθώς και νέες που αποκτήθηκαν παρατηρώντας τον default controller που παρείχε η iRobot στο μοντέλο της Create. Κάνοντας έτσι ένα συνδυασμό τον δύο παραπάνω, δομήθηκε ο νέος supervisor που ικανοποιεί τις ανάγκες για την λειτουργία της σκούπας. Η υλοποίηση έγινε σε Java.

Περιληπτικά ο κώδικας του supervisor εκτελεί τις ακόλουθες λειτουργίες: Αρχικά ορίζονται όλα τα εξαρτήματα που χρησιμοποιούνται πάνω στο ρομπότ καθώς και οι διάφορες σταθερές που χρησιμοποιούνται στην υλοποίηση. Έπειτα ορίζεται ένας Constructor ο οποίος είναι υπεύθυνος για την αρχικοποίηση των στιγμιότυπων της συσκευής, όταν αυτά

δημιουργηθούν. Συνεχίζοντας ορίζονται οι συναρτήσεις υπεύθυνες για τον εντοπισμό εμποδίων στο διάβα της Rooby καθώς και συναρτήσεις οι οποίες διαχειρίζονται την μετακίνηση της. Ακολούθως υλοποιούνται οι αλγόριθμοι που εκτελούν το path planning για τον πράκτορα, για να καταλήξουμε στην run() όπου ο ορίζεται εν κατακλείδι, η τελική λειτουργία του πράκτορα αξιοποιώντας όλα τα παραπάνω.

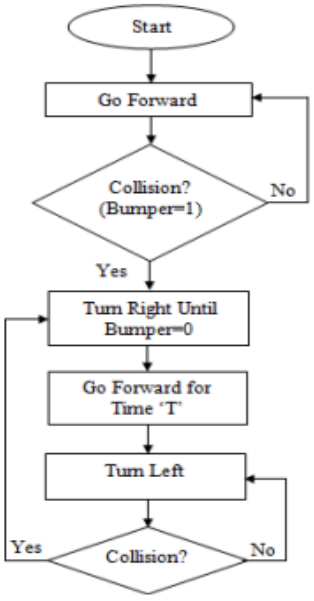
4.β. Path Planning Αλγόριθμοι

A. Wall Following

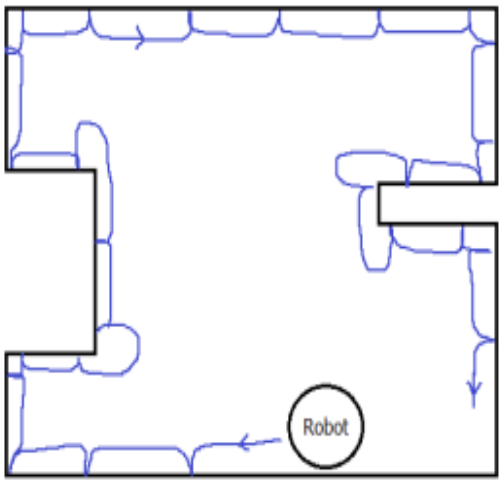
Ο αλγόριθμος Wall follow επιτρέπει στο ρομπότ να μετακινείται διατηρώντας δεξιά ή αριστερά του τον τοίχο ανάλογα με ποιά εκδοχή του αλγορίθμου χρησιμοποιούμε (left wall following or right wall following). Με τον τρόπο αυτό η σκούπα μπορεί να καθαρίζει τα δύσβατα σημεία κοντά στον τοίχο καθώς και στις γωνίες. Εντούτοις ο αλγόριθμος αυτός είναι αρκετά περιοριστικός, καθώς κατά κύριο λόγο εξασφαλίζει την απολύμανση της περιμέτρου του δωματίου και όχι της εσωτερικής του επιφάνειας.

Μοναδική εξαίρεση αποτελεί όταν ο αλγόριθμος αυτός συνδυαστεί με κάποιον άλλο που τον οδηγεί στα ενδότερα σημεία του χώρου. Όταν συμβαίνει αυτό, ως «τοίχος» μπορούν να θεωρηθούν ενδιάμεσα εμπόδια, όπως τα έπιπλα, καθαρίζοντας έτσι πολύ αποτελεσματικά τον χώρο που τα περιβάλλει. Συνεπώς, παρόλο που το ποσοστό του χώρου που διαβένεται χάρη στην χρήση του αλγορίθμου είναι μικρό σε σύγκριση με άλλους αλγορίθμους, ο αλγόριθμος αυτός καθιστά εφικτό τον αποτελεσματικό καθαρισμό των πιο δυσπρόσιτων σημείων στον χώρο.

Ενδεικτικά Το flowchart της λειτουργίας του left wall following αλγορίθμου καθώς και η αναμενόμενη πορεία της σκούπας με την χρήση του φαίνονται παρακάτω:



Εικόνα 4.1: Flowchart of Left Wall Follow



Εικόνα 4.2: Προβλεπόμενη τροχιά Left Wall Follow

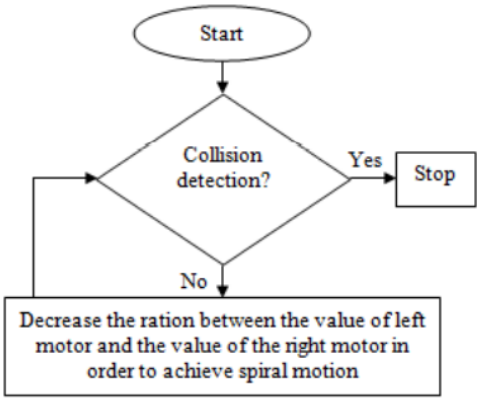
B. Spiral Algorithm

Ο αλγόριθμος παρέχει την δυνατότητα στην σκούπα να κινείται σε πορεία αυξανόμενου κύκλου. Συγκεκριμένα, σε πρώτη φάση ελέγχεται αν υπάρχει κάποιο εμπόδιο το οποίο να εμποδίζει την κίνηση του ρομπότ σε κυκλική τροχιά. Αν υπάρχει, η σκούπα αναγκάζεται να σταματήσει. Στην περίπτωση που δεν υπάρχει όμως, η σκούπα αρχίζει να κινείται κυκλικά με ωρολόγια ή αντιωρολόγια φορά, αυξάνοντας σταδιακά την ακτίνα από το κέντρο του κύκλου.

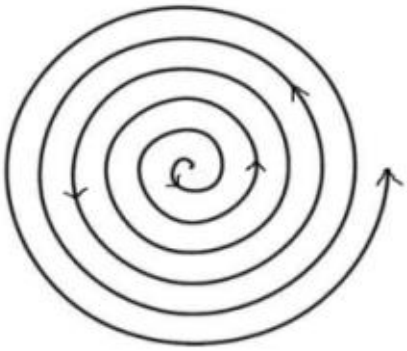
Ενδεικτικά για αντιωρολόγια φορά, ο αλγόριθμος θα λειτουργούσε ως εξής. Ξεκινώντας την πορεία της η σκούπα ο δεξιός τροχός κινείται με μέγιστη ταχύτητα ενώ ο αριστερός με κάποια ελάχιστη αρχική τιμή για δεδομένο χρονικό διάστημα dt. Σταδιακά λοιπόν, καθώς η κίνηση συνεχίζεται, θα πρέπει η ταχύτητα του αριστερού άξονα να αυξάνεται με φθίνοντα ρυθμό μεταβολής, ενώ το χρονικό διάστημα dt για το οποίο η σκούπα θα κινείται με μία δεδομένη ταχύτητα, επίσης θα αυξάνεται με αύξοντα ρυθμό αυτή τη φορά. Με τον τρόπο αυτό μειώνεται το κλάσμα $\frac{ταχύτητα\ αριστερού\ τροχού}{ταχύτητα\ δεξιού\ τροχού}$, και κατά συνέπεια η ακτίνα του κύκλου αυξάνεται, με ρυθμό τέτοιο ώστε να μην έχουμε απότομη αποστασιοποίηση από το κέντρο.

Με τον τρόπο αυτό, ο Spiral movement algorithm κρίνεται ιδιαίτερα χρήσιμος για τον καθαρισμό επιφανειών χωρίς εμπόδια καλύπτοντας μεγάλες περιοχές του χώρου που περικλείονται από τα όρια του αυξανόμενου κύκλου. Ωστόσο η αξία του είναι πολύ περιορισμένη όταν η σκούπα περιστοιχίζεται από πληθώρα εμποδίων, καθώς με τον εντοπισμός τους η κίνηση σταματάει.

Το flowchart της λειτουργίας του Spiral αλγορίθμου καθώς και η αναμενόμενη πορεία της σκούπας με την χρήση του φαίνονται παρακάτω:



Εικόνα 4.3: Flowchart of Spiral Algorithm



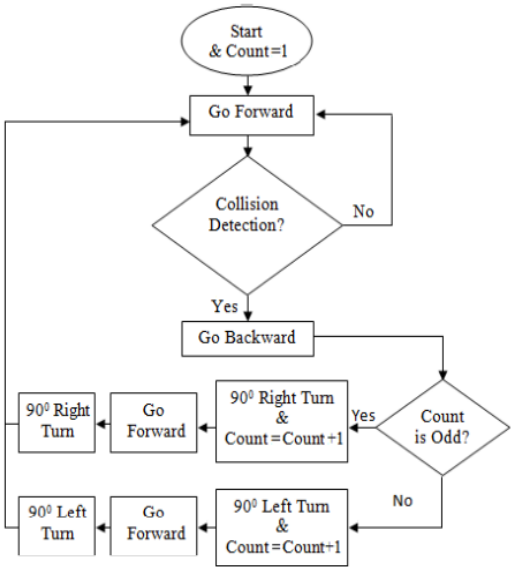
Εικόνα 4.4: Προβλεπόμενη τροχιά Spiral Algorithm

Γ. Boustrophedon -"S" shaped Pathway

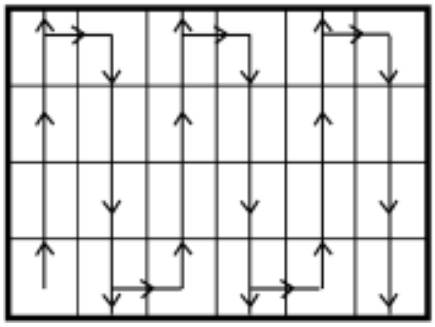
Η ονομασία του αλγορίθμου προέρχεται από τα αρχαία ελληνικά (*βου-στροφη-δόν)* και περιγράφει τον τρόπο με τον οποίο στρίβει(«στροφή») η κουκουβάγια(«βους»). Εφαρμόζοντας αυτό τον path-planning αλγόριθμο σχηματίζεται μια διαδρομή που μοιάζει με τον χαρακτήρα "S". Ειδικότερα, έστω ότι το ρομπότ ξεκινάει κινούμενο προς τα πάνω, προχωράει μπροστά ώσπου να ενεργοποιηθούν οι εμπρόσθιοι bumpers και να αναγνωριστεί κάποιο εμπόδιο. Τότε η σκούπα στρίβει κατά 90 μοίρες δεξιά (δίχως βλάβη της γενικότητας), και προχωράει για μια απόσταση όσο το πλάτος της επιφάνειας που έχει καθαριστεί. Φθάνοντας στο τέρμα στρίβει κάθετα, ξανά δεξιά, κινούμενο προς την κάτω κατεύθυνση. Η κίνηση συνεχίζεται και όταν συναντήσει εμπόδιο ξανά, εκτελεί την ίδια διαδικασία στρίβοντας αριστερά. Η πλευρά από την οποία θα στρίβει κάθε φορά η σκούπα, καθορίζεται από έναν μετρητή τον οποίο αυξάνουμε κάθε φορά που απαντάται ένα εμπόδιο. Έπειτα ο αλγόριθμος επαναλαμβάνεται.

Ο αλγόριθμος αυτός αποτελεί τον ταχύτερο τρόπο να καλύψουμε μια επιφάνεια στο σύνολό της, έχοντας ωστόσο, κι αυτός κάποια σημαντικά μειονεκτήματα. Συγκεκριμένα, διαπρέπει όταν χρησιμοποιείται σε κάποιο χώρο τον οποίο γνωρίζουμε εκ των προτέρων αξιοποιώντας μια Διαίρει και Κυρίευε τακτική. Γνωρίζοντας, δηλαδή, τα εμπόδια στον χώρο, τον διαμοιράζει σε μικρότερου μεγέθους κελιά τα οποία μπορεί εύκολα να διανύσει καθαρίζοντας με πολύ μεγάλη αποτελεσματικότητα. Όταν καθαριστούν όλα τα κελία ο χώρος θα έχει καλυφθεί εξ' ολοκλήρου. Ωστόσο, όταν δεν μπορούμε να γνωρίζουμε εκ των προτέρων τον περιβάλλοντα χώρο, ο αλγόριθμος αυτός είναι ιδιαίτερα επιρρεπείς σε καταστάσεις όπου η σκούπα κολλάει σε μη ανιχνεύσιμους κύκλους, καθιστώντας αδύνατη την επαναφορά της σε λειτουργικότητα.

Αξίζει επιπλέον να σημειωθεί ότι έχει σημασία αν η σκούπα κινείται ακολουθώντας τον τοίχο μεγαλύτερης διάστασης αντ' αυτού με την μικρότερη. Αυτό διότι κινούμενη προς την μεγαλύτερη, θα χρειαστεί να εκτελέσει λιγότερες στροφές, διατηρώντας έτσι για περισσότερο χρόνο σταθερή πορεία. Με τον τρόπο αυτό εξοικονομείται μπαταρία. Αυτό βέβαια δεν έχει καμία επίπτωση στο αποτέλεσμα, καθώς η κάλυψη του χώρου θα είναι ίδια και στις δυο περιπτώσεις. Ακολουθούν το διάγραμμα ροής και η προβλεπόμενη τροχιά:



Εικόνα 4.5: Flowchart of Boustrophedon



Εικόνα 4.6: Προβλεπόμενη τροχιά Boustrophedon

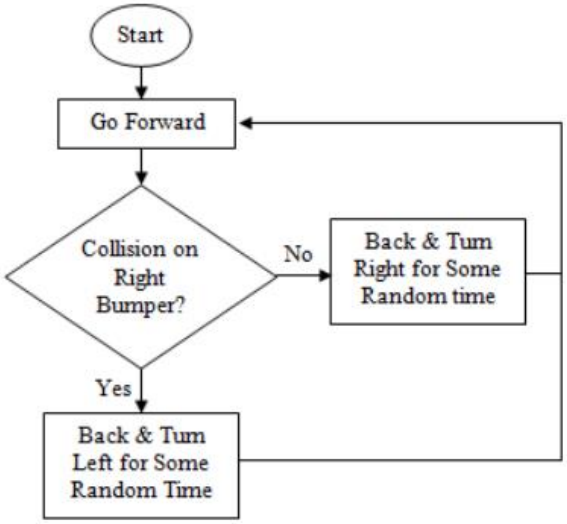
Δ. Random Walk

Ο Random Walk αποτελεί ένα ιδιαιτέρα απλό αλγόριθμο στην σύλληψη και την κατανόησή του, καθώς και δεν χρειάζεται πολλούς πόρους για να αποδώσει, σε υλικό και μηχανικό επίπεδο. Με άλλα λόγια, δεν χρειάζεται τίποτα περισσότερο από τους bumpers τις σκούπας, καθώς δεν έχει να πραγματοποιήσει κάποιον εξεζητημένο υπολογισμό, οπότε και οποιαδήποτε πληροφόρηση θα του ήταν άχρηστη.

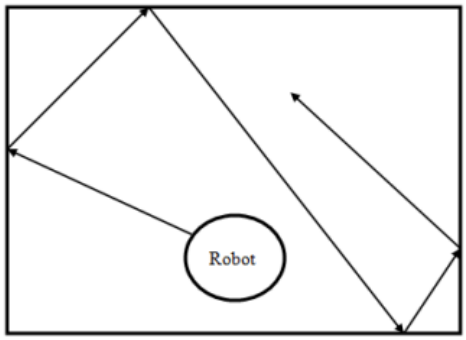
Το ρομπότ, λοιπόν, μπορεί να κινείται προς οποιαδήποτε κατεύθυνση μέχρις ότου να ενεργοποιηθεί ένας εκ των δυο bumpers. Όταν αυτό συμβεί, το ρομπότ ανάλογα με το ποιος bumper ενεργοποιήθηκε, π.χ. ο δεξιός, μετακινείται λίγο πίσω και στρίβει κατά την άλλη πλευρά, δηλαδή αριστερά. Αυτό γίνεται ώστε να αποφευχθούν συγκρούσεις. Η γωνία με την οποία ωστόσο θα στρίψει το ρομπότ δεν είναι γνωστή a priori, καθορίζεται κάθε φορά μέσω μιας ψευδοτυχαίας γεννήτριας η οποία επιστρέφει κάποιον αριθμό στο εύρος [0,1) ο οποίος πολλαπλασιάζεται με το π. Με τον τρόπο η γωνία της στροφής είναι μέσα στο εύρος [ο,π) διαβεβαιώνοντας ότι η στροφή θα είναι με ασφάλεια προς τα αριστερά (για τις συνθήκες του προηγούμενου παραδείγματος).

Γίνεται εύκολα αντιληπτό ότι ο αλγόριθμος αυτός καθ' αυτός δεν θα μπορούσε αξιόπιστα να χρησιμοποιηθεί μόνος του για την επίτευξη του ολοκληρωτικού καθαρισμού ενός χώρο. Ωστόσο η συνδυαστική χρήση του με τους προηγούμενους μπορεί να αποφέρει πολύ ενθαρρυντικά αποτελέσματα καθώς η τυχαιότητα που παρέχει στην υλοποίηση είναι κρίσιμη για την μετακίνηση της σκούπας σε όλα τα σημεία εντός του χώρου καθώς μέσω αυτής είναι πολύ ευκολότερο να μεταφερόμαστε σε απόμακρα σημεία σε σχέση με τη παροντική μας θέση.

Το διάγραμμα ροής και η προβλεπόμενη κίνηση του ρομποτικού μοντέλου είναι τα ακόλουθα:



Εικόνα 4.7: Flowchart of Random Walk



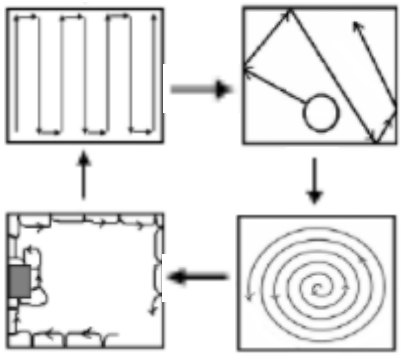
Εικόνα 4.8: Προβλεπόμενη τροχιά Random Walk

4.γ Αυτόνομο Καθάρισμα

Γίνεται λοιπόν κατανοητό, ότι καθένας από τους προηγούμενους αλγορίθμους, έχει κάτι διαφορετικό να προσφέρει στην υλοποίηση ενός πράκτορα ο οποίος θα κατευθύνει την σκούπα. Έχει βέβαια, και κάποια αδυναμία, η οποία λειτουργεί λίγο περιοριστικά ως προς την απόδοση του, όταν εφαρμόζεται μεμονωμένα. Προκειμένου λοιπόν να μεγιστοποιηθεί η αποτελεσματικότητα του καθαρίσματος και να περιοριστούν όσο το δυνατόν οι μη θεμιτές καταστάσεις που προκύπτουν από την εφαρμογή κάθε αλγορίθμου, για την συνολική υλοποίηση του πράκτορα θα εφαρμοστεί ένα κράμα όλων των προηγούμενων. Συγκεκριμένα θα αποδίδεται τυχαία χρόνος λειτουργίας για κάθε αλγόριθμο, και έπειτα με την χρήση ενός timer θα εναλλάσσουμε μεταξύ διαφορετικών υλοποιήσεων.

Η μεθοδολογία προκύπτει κατά τον τρόπο αυτό, καθώς το παρόν ρομποτικό μοντέλο δεν επεξεργάζεται κάποια πληροφορία για τον αν έχει ήδη καθαριστεί κάποια επιφάνεια, ούτε είναι γνωστή κάποια χαρτογράφηση του χώρου. Θα λέγαμε ότι κινείται λοιπόν, στα «τυφλά» κάνοντας χρήση μόνο των 2 bumpers του για να μετακινείται στον χώρο. Συνεπώς κάποια τυχαιότητα είναι αναγκαία προκειμένου να καλυφθεί όλος ο χώρος, καθώς αναγκαία είναι επίσης και η χρήση πολλαπλών αλγορίθμων για να μπορούν να αντιμετωπίζονται, αν όχι να αποφεύγονται, οι δυσχερείς καταστάσεις.

Η ακολουθία με την οποία εναλλάσσονται οι αλγόριθμοι, ή με άλλα λόγια ο κύκλος καθαρίσματος(Cleaning Cycle), παρουσιάζεται παρακάτω:



Εικόνα 4.9: Κύκλος Καθαρίσματος

Συνεπώς ξεκινάμε με Boustrophedon, καθώς η σκούπα συνήθως ξεκινά από κάποιο εύχρηστο χώρο, τον οποίο μπορεί με μεγάλη ταχύτητα να καλύψει ο αλγόριθμος. Άλλωστε είναι, όπως ειπώθηκε και προηγουμένως, ο ταχύτερος αλγόριθμος για την κάλυψη κατάλληλα οριοθετημένου χώρου. Στην υλοποίηση του boustrophedon έχει προτεθεί επιπλέον ένας έλεγχος για το αν το ρομπότ μετά από την στροφή του, αφότου έρθει σε επαφή με εμπόδιο, συνεχίζει να συναντά εμπόδιο. Στην περίπτωση αυτή, αλλάζουμε αλγόριθμο άμεσα καθώς αυτό σημαίνει ότι η σκούπα έχει βρεθεί σε μια κατάσταση όπου δεν μπορεί να εξέλθει με boustrophedon.

Έπειτα εφαρμόζεται Random Walk, προκειμένου η σκούπα να απομακρύνεται από τα σημεία που «κολλάει» συχνά με την χρήση του Boustrophedon, αλλά και με κάποια τυχαιότητα να αλλάζει δωμάτια και χώρους ώστε να μην επαναλαμβάνεται συνέχεια καθαρισμός του ίδιου σημείου.

Μετά χρησιμοποιείται ο Spiral Algorithm, στο σημείο όπου μας έχει μεταφέρει το Random Walk, το οποίο θα μπορούσε να είναι ένα εσωτερικό σημείο του χώρου όπου ο αλγόριθμος θα είναι ιδιαίτερα αποδοτικός. Ο Spiral Algorithm εκτελείται ώσπου να τελειώσει ο χρόνος του, ή έως ότου συναντήσει κάποιο εμπόδιο που αποτρέπει την περιστροφική κίνηση. Θα μπορούσαμε μάλιστα να αφήσουμε τον αλγόριθμο να εκτελείται πάντα για χρόνο, τόσο ώστε να έρθει σε επαφή με εμπόδιο, καθώς αυτό μπορεί να συμβεί πολύ σύντομα, αν βρισκόμαστε κοντά σε εμπόδιο, ή πολύ αργά, αν βρισκόμαστε στην μέση, καθαρίζοντας σημαντικές επιφάνειες. Ωστόσο, με τον τρόπο αυτό πολλές φορές χάνεται περισσότερος χρόνος απ' ότι χρειάζεται καθαρίζοντας επαναληπτικά σημεία από τα οποία έχει ήδη περάσει η σκούπα.

Τέλος, εφόσον ολοκληρωθεί η εκτέλεση του Spiral Algorithm, αρχίζει να εφαρμόζεται Left Wall Following (LWF) προκειμένου, να καθαριστούν όλα τα οριακά σημεία του χώρου, καθώς οι μεγάλες εσωτερικές επιφάνειες έχουν ήδη καλυφθεί από τους προηγούμενους. Ωστόσο, προτού η εκτέλεση επανέλθει ξανά στον Boustrophedon, κλείνοντας τον κύκλο, η σκούπα περιστρέφεται κατά 90 μοίρες δεξιά. Αυτό συμβαίνει διότι πιθανότατα η σκούπα ακολουθεί κάποιον τοίχο και είναι στραμμένη προς κάποια γωνία, που σχηματίζει ο τοίχος που ακολουθεί μαζί με τον επόμενο. Έτσι στρίβοντας δεξιά, το ενδιαφέρον στρέφεται ξανά προς τον εσωτερικό χώρο και όχι τα σύνορά του.

5. Αποτελέσματα

Προκειμένου να εκλάβουμε μια γενικότερη εικόνα για το πώς λειτουργεί ο πράκτοράς μας σε διαφορετικές περιβαλλοντικές συνθήκες δοκιμάστηκαν οι 3 κόσμοι που φαίνονται στις εικόνες 2.1, 2.2, 2.3. Τα πειράματα εκπονήθηκαν κατά τέτοιο τρόπο ώστε να αξιολογήσουμε την αποδοτικότητα της σκούπας στα ακόλουθα περιβάλλοντα: α) δίχως εμπόδια , β) περιορισμένα εμπόδια, γ) πολλαπλά εμπόδια.

5.α. Παράμετροι σε πειραματισμούς

Η αποδοτικότητα της σκούπας θα καθοριστεί, παρατηρώντας και συγκρίνοντας τα αποτελέσματα του καθαρισμού μετά από 40 Cleaning Cycles, δηλαδή 40 πλήρεις εναλλαγές μεταξύ των αλγορίθμων. Ο αριθμός αυτός τον κύκλων είναι αρκετός για να έχει καθαριστεί σε μεγάλο ποσοστό η ζητούμενη επιφάνεια.

Επίσης κάθε αλγόριθμος θα εφαρμόζεται από ελάχιστο χρόνο 0 έως μέγιστο περίπου 2 λεπτά. Η ανακρίβεια προκύπτει από το γεγονός ότι για την μέτρηση του χρόνου εφαρμογής

κάθε αλγορίθμου χρησιμοποιείται ένα loop με την συνάρτηση System.*currentTimeMillis().* Οι προσομοιώσεις ωστόσο δεν γίνονται σε πραγματικό χρόνο, για ευνόητους λόγους, αλλά σε αυξομειούμενους χρόνους προσομοίωσης που προκύπτουν από την ικανότητα του υπολογιστή. Το γεγονός αυτό έχει ως αντίκτυπο η παράμετρος maxTime να υπολογίζεται με προσεγγίσεις πειραματικά εκτελώντας την πράξη: $maxTime = (\frac{120}{(\mu\acute{\epsilon}\sigma o\varsigma\ \pi\rho o\sigma\omega\mu o\iota\omega\mu\acute{\epsilon}\nu o\varsigma\ \chi\rho\acute{o}\nu o\varsigma)})\ s.$

Εξαίρεση αποτελεί ο αλγόριθμος LWF στον οποίο αναθέτουμε χρόνο από 0 έως maxTime/2 καθώς η χρήση του είναι κατά κύριο λόγο για οριακές περιοχές οπότε αν εκτελείται για πολύ χρόνο θα έχουμε συχνά επαναπεράσματα από ήδη καθαρά σημεία, οπότε θα αποτελεί μεγάλη σπατάλη χρόνου.

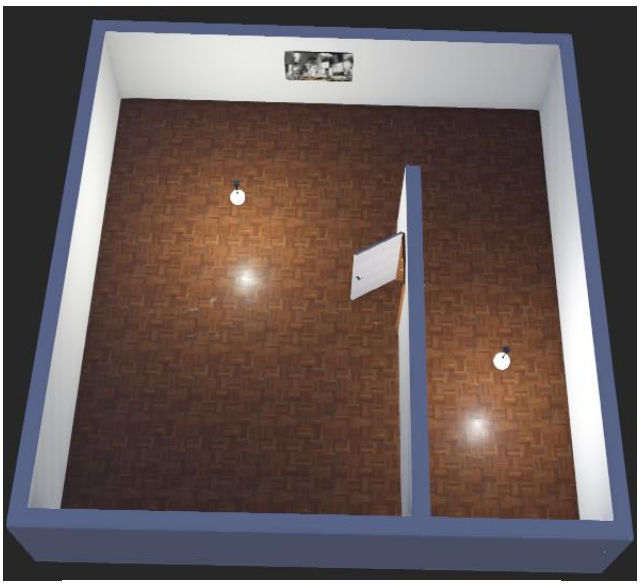
Συνεπώς ο μέσος αναμενόμενος χρόνος για τον κύκλο καθαρισμού, λαμβάνοντας υπόψιν ότι η ψευδοτυχαία επιλογή χρόνου εκτέλεσης για κάθε αλγόριθμο ακολουθεί ομοιόμορφη κατανομή, προκύπτει ως εξής:

$$E[x] = 3\ \frac{a+b}{2} + \frac{c+d}{2} = 3\frac{0+2}{2} + \frac{0+1}{2} = 3 + 0.5 = 3.5\ mins$$

Συνεπώς εκτιμώμενη συνολική εκτέλεση $3.5mins * 40 = 140\ mins = 2h\ 20mins.$ Το παραπάνω ωστόσο δεν αποτελεί ακριβή περιγραφή καθώς ο χρόνος αυτός στην πραγματικότητα θα είναι λίγο μικρότερος(περίπου 2h) καθώς ο Spiral Algorithm διακόπτεται επιπλέον από την συνθήκη εύρεσης εμποδίου, το οποίο σε πολλές περιπτώσεις θα σημαίνει ακαριαία.

5.β. Αποτελέσματα σε χώρο δίχως εμπόδια (Εικόνα 2.1)

Αρχικά θα παρουσιαστούν τα αποτελέσματα του καθαρισμού του χώρου μετά από 40 επαναλήψεις, όπως περιγράφηκε προηγουμένως:



Εικόνα 5.1: Χώρος μετά το καθάρισμα

Εικόνα 5.2: Ground Overlay

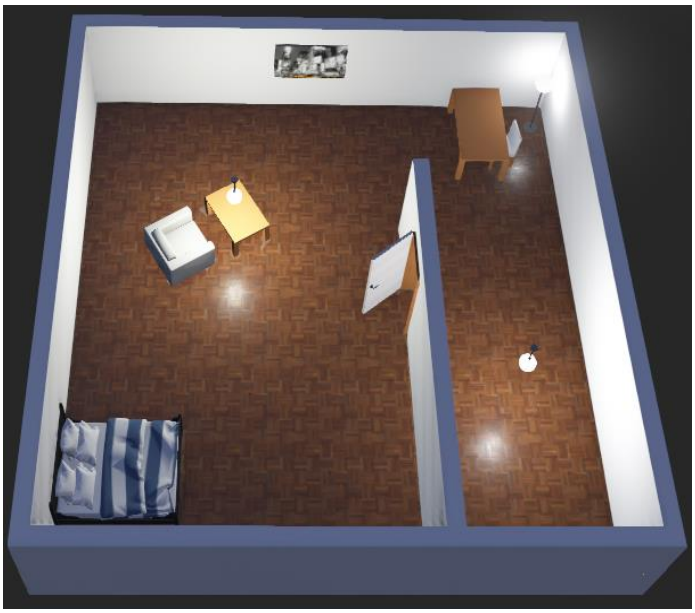
Για την παραπάνω προσομοίωση η παράμετρος maxTime καθορίστηκε ως εξής:

$maxTime = \left(\frac{120}{(\mu έσος\ προσωμοιωμένος\ χρόνος)}\right) s = \frac{120}{26.7} s = 4.5\ s.$ Με τον τρόπο αυτό κάθε αλγόριθμος εφαρμόζεται για μέγιστο χρόνο 2 λεπτά.

Τα αποτελέσματα θα μπορούσαμε να πούμε ότι είναι εντυπωσιακά καθώς εκτιμάται, ότι έχει απολυμανθεί, ένα ποσοστό περίπου 99% της συνολικής επιφάνειας σε διάστημα λίγο μεγαλύτερο της 1h & 30 mins. Συνεπώς η σκούπα θα μπορούσε με πολύ μεγάλη επιτυχία να χρησιμοποιείται για καθαρισμό ενός χώρου 100 τετραγωνικών όταν σε αυτόν δεν υπάρχουν εμπόδια.

5.γ. Αποτελέσματα σε χώρο με περιορισμένα εμπόδια (Εικόνα 2.2)

Τα αποτελέσματα που προκύπτουν από την προσομοίωση στις παραπάνω συνθήκες είναι τα ακόλουθα:



Εικόνα 5.3 Χώρος μετά το καθάρισμα

Εικόνα 5.4: Ground Overlay

Η παράμετρος με όμοιο τρόπο, έλαβε την τιμή 6 προκειμένου να έχουμε ικανοποιητικό χρόνο εκτέλεσης για τους αλγορίθμους.

Τα αποτελέσματα κατά την παραπάνω προσομοίωση είναι επίσης πολύ ικανοποιητικά θα λέγαμε. Στο απεικόνιση της 5.4 μπορούμε να παρατηρήσουμε ότι ένα ποσοστό γύρω στο 97% της συνολικής προσβάσιμης επιφάνειας, λόγω των επίπλων, έχει απολυμανθεί. Είναι μάλιστα ευδιάκριτη η περιοχή που δεν μπορεί να καθαριστεί λόγω των εμποδίων, γεγονός που οφείλεται στην πολύ καλή δουλεία που έχει πραγματοποιηθεί στην απολύμναση της περιμετρικής της περιοχής. Συγκεκριμένα βλέπουμε την βάση του κρεβατιού καθώς και τα πόδια των επίπλων του καθιστικού και του γραφείου. Τα σημεία που υστερεί κατά λίγο η

αποδοτικότητα βρίσκονται συγκεντρωμένα κυρίως σε σημεία κοντά στον τοίχο και κυρίως στην περιοχή του γραφείου οπού η πρόσβαση στην περιοχή είναι ιδιαίτερα δύσκολη.

5.δ. Αποτελέσματα σε χώρο με πολλαπλά εμπόδια (Εικόνα 2.3)

Ακολουθούν τα αποτελέσματα προσομοίωσης σε πλήρως εξοπλισμένο οίκημα 100 τ.μ.

Εικόνα 5.5 Χώρος μετά το καθάρισμα

Εικόνα 5.6: Ground Overlay

Η παράμετρος maxTime για το πείραμα έλαβε την τιμή 20 καθώς ο μέσος προσημειωμένος χρόνο ήταν περίπου 6 s.

Ακόμη λοιπόν και με πλήρως επιπλωμένο χώρο βλέπουμε ότι η σκούπα δουλεύει εξαιρετικά καλά καλύπτοντας ένα ποσοστό της τάξεως του 95% του ζητούμενου χώρου. Με τις δυσκολίες τις να *περιορίζονται* σε πολύ δυσπρόσιτες περιοχές πίσω και κάτω από έπιπλα

6. Παρατηρήσεις

Κατά τους διάφορους πειραματισμούς στην προσπάθεια εξαγωγής των ζητούμενων αποτελεσμάτων προέκυψε μια σειρά από παρατηρήσεις σχετικά με την υλοποίηση του πράκτορα.

Αρχικά, παρατηρήθηκε ότι οι παράμετρος minTime και maxTime, έχουν ιδιαίτερα σημαντικό ρόλο για την συνολική απόδοση της σκούπας. Αυτό συμβαίνει διότι, από την μια, όταν οι αλγόριθμοι εκτελούνται για πολύ λίγο χρόνο τα αποτελέσματα που αποκομίζονται είναι μηδαμινά, λόγω του ότι δεν προλαβαίνουν να δράσουν καθώς και εξαντλούνται σημαντικά τα αποθέματα σε ενέργεια εναλλάσσοντας διαρκώς μεταξύ αλγορίθμων. Από την άλλη όταν οι αλγόριθμοι εκτελούνται για μεγάλη διάρκεια χρόνου τότε παρατηρούνται 2 φαινόμενα. Αφενός. Μεγάλα χρονικά διαστήματα επαναλήψεων καθαρίσματος μια

επιφάνειας, και αφετέρου «κολλήματα» για μεγάλα διαστήματα της σκούπας σε περιοχές από τις οποίες δεν μπορεί να απεγκλωβιστεί εύκολα αν δεν αλλάξει αλγόριθμος. Και στις δυο περιπτώσεις ωστόσο, κοινός παράγοντας είναι τα υπερμεγέθη ποσά χρόνου που σπαταλιέται..

Μέσω πολλαπλών πειραμάτων για την σκούπα μας προσδιορίστηκε ότι η απόδοση μεγιστοποιείται αν κάθε αλγόριθμος χρησιμοποιείται για μέγιστο χρόνο 2 λεπτά καθώς επιτυγχάνεται τέλεια ισορροπία.

Δεύτερη παρατήρηση, αποτελεί το γεγονός ότι διαφορετικοί χώροι καθαρίζονται με μεγαλύτερη απόδοση αξιοποιώντας διαφορετικούς αλγορίθμους. Συγκεκριμένα σε ένα περιβάλλον δίχως εμπόδια, προτιμάται βέλτιστα, να έχουμε σύντομες εκτελέσεις wall following αλγορίθμων και εκτεταμένες εκτελέσεις αλγορίθμων εσωτερικού χώρου όπως boustrophedon και spiral. Δυικές παρατηρήσεις ισχύουν για περιοχές εκτενώς επιπλωμένες.

Έπειτα, αξιοσημείωτο είναι το γεγονός ότι κάθε αλγόριθμος βοηθάει με τον δικό του διαφορετικό και συνάμα μοναδικό τρόπο στην συνολική υλοποίηση, ενώ ο κύκλος καθαρίσματος έχει τεθεί με τέτοιο τρόπο ώστε κάθε αλγόριθμος να καλύπτει ι, όσο αυτό είναι εφικτό, τις αδυναμίες του προηγούμενου. Αναλυτικότερα ο boustrophedon καθαρίζει πολύ γρήγορα βολικά οριοθετημένες επιφάνειες αλλά εγκλωβίζεται εύκολα σε μη αποφεύξιμες καταστάσεις. Ο Random Walk ωστόσο που εφαρμόζεται μετά, είναι ιδιαίτερα ικανός στο να αποφεύγει εγκλωβισμούς αξιοποιώντας την τυχαιότητα του για να μεταφέρεται σε οποιοδήποτε μέρος στον χώρο. Ωστόσο με random walk δεν μπορεί να επιτευχθεί συστηματικό, αποδοτικός καθαρισμός ενός δεδομένου χώρου. Εκεί έρχεται να βοηθήσει ο Spiral Algorithm ο οποίος είναι πολύ αποδοτικός στον να καθαρίζει επιφάνειες που δεν είναι περιορισμένες από εμπόδια. Πρόβλημα του spiral με την σειρά του είναι ότι δεν μπορεί να αντιμετωπίσει εμπόδια. Για τον λόγο αυτό χρησιμοποιείται κάποιος Wall Following αλγόριθμος που είναι πολύ ικανός στο να καθαρίζει την περιμετρική επιφάνεια των εμποδίων. Κλείνοντας τον κύκλο, ο Wall Follow αλγόριθμος περιορίζεται μόνο στις περιμετρικές περιοχές και κινείται παράλληλα με το εμπόδιο δίπλα του, Έτσι στρίβοντας 90 μοίρες κατά την ολοκλήρωση του η σκούπα στρέφεται ξανά προς την εσωτερική επιφάνεια ώστε να αποδώσει βέλτιστα ο boustrophedon.

7. Προβλήματα που αντιμετωπίστηκαν

Η υλοποίηση του πράκτορα παρόλο που με μια πρώτη ματιά φαίνεται να είναι απλή αλγοριθμική εργασία, κρύβει από πίσω, πολλές μικρές λεπτομέρειες που κάνουν την διαφορά στην απόδοση. Οι ανακάλυψη και η αξιοποίηση των λεπτομερειών αυτών είναι εκείνη που δημιούργησε ποικίλα προβλήματα που χρειάστηκες να επιλυθούν ώστε να φθάσουμε στον παρόντα πράκτορα.

Πρώτη τέτοια δυσκολία/ιδιαιτερότητα που απαντήθηκε είναι ότι το ρομποτικό μοντέλο που χρησιμοποιήθηκε, σε ορισμένες περιπτώσεις, λόγω των φυσικών παραγόντων του περιβάλλοντος του Webots, ανίχνευε στο διάβα του cliffs στο εμπρόσθιο μέρος της κίνησής του

χωρίς ωστόσο να υπάρχουν. Απόρροια αυτού του γεγονότος ήταν η σκούπα συχνά να ακολουθεί διαδρομές διαφορετικές από την αναμενόμενη. Το πρόβλημα αυτό επιλύθηκε κάνοντας ένα δεύτερο μικρό βήμα μπροστά, τόσο ώστε να μπορέσει η σκούπα να ανιχνεύσει το cliff αλλά να μην πέφτει μέσα του, όταν παρουσιάζεται ένδειξη για ύπαρξή του.

Ένα άλλο ανάλογο πρόβλημα αποτέλεσε η στροφή κατά συγκεκριμένη γωνία, καθώς δεν περιορίζεται στην διαφορετική ταχύτητα των δύο τροχών, αλλά εμφανίζει κάποια πολυπλοκότητα. Η υλοποίηση περιληπτικά είναι η ακόλουθη. Αρχικά σταματάει η κίνηση της σκούπας, και κρατιούνται σε δυο μεταβλητές η θέση αριστερού και δεξιού τροχού στον χώρο. Έπειτα προσδιορίζεται η κατεύθυνση και το μέτρο της ταχύτητας ώστε να αρχίσει επαναλαμβανόμενα να στρίβει η σκούπα. Στο loop αυτό ελέγχεται κάθε φορά η απόσταση που έχει καλύψει κάθε τροχός και υπολογίζεται το delta orientation της κίνησης. Αν ο προσανατολισμός έχει μεταβληθεί κατά μέγεθος μεγαλύτερο ή ίσο τη ζητούμενης γωνίας, η στροφή ολοκληρώνεται.

Περνώντας στους αλγορίθμους, υπήρξαν επίσης αρκετά ζητήματα προς επίλυση. Ξεκινώντας από τον Spiral, μεγαλύτερη δυσκολία αποτέλεσε να μπορέσει να γίνει ομαλή αύξηση της ακτίνας του κύκλου χωρίς να έχουμε κάποια απότομη απομάκρυνση χαλώντας τον κύκλο. Η λύση που δόθηκε στο πρόβλημα αυτό ήταν να δοθούν εξαρχής ο χρόνος για τον οποίο θα διατηρούνταν κάποια ταχύτητα (tourTime), το ποσό κατά το οποίο θα αυξανόταν η ταχύτητα (increaseSpeed) καθώς και το ποσό κατά το οποίο θα αυξανόταν το tourTime (increaseTime). Έπειτα εκτελείται ένα loop στο οποίο πραγματοποιείται η κίνηση με την κατάλληλη ταχύτητα και χρόνο. Σε κάθε επανάληψη του loop επαναπροσδιορίζεται το tourTime και η ταχύτητα για την επόμενη επανάληψη ως εξής: new = old + increase. Το σημαντικό κομμάτι για την ομαλοποίηση του κύκλου είναι ότι κάθε φορά ο ρυθμός αύξησης του χρόνου γίνεται μεγαλύτερος (increaseTime) ενώ ο ρυθμός αύξησης της ταχύτητας γίνεται μικρότερος (increaseSpeed).

Μια επιπλέον λεπτομέρεια στην υλοποίηση, αυτή τη φορά για τον Wall Following Αλγόριθμο, αποτελεί η τελευταία μετάβαση στο διάγραμμα ροής της εικόνας 4.1. Συγκεκριμένα θα έπρεπε η σκούπα να στρίβει επαναληπτικά ώσπου τελικά να συναντήσει κάποιο εμπόδιο. Το πρόβλημα που τίθεται είναι ότι αν για κάποιο λόγο η σκούπα δεν μπορούσε να εντοπίσει εμπόδιο περιστρεφόταν για μεγάλη διάρκεια γύρω από τον εαυτό της. Το πρόβλημα αυτό επιλύθηκε τοποθετώντας τον περιορισμό όταν η μέγιστη γωνία στροφής πρέπει να είναι 2π rad.

Τελειώνοντας το τελευταίο ζήτημα που αντιμετωπίστηκε, αφορά τον αλγόριθμο Boustrophedon και είναι ιδιαίτερα απλό στην σύλληψή του. Κάθε φορά που ζητείται στροφή γωνίας 90 μοιρών το πρόβλημα είναι ότι εκ των απαιτήσεων για να εκτελεστεί ομαλά ο αλγόριθμος θα πρέπει να υπάρχει μεγάλη ακρίβεια στην γωνία αυτή. Διαφορετικά δεν καλύπτεται όλος ο χώρος σχηματίζοντας “S”. Όπως είδαμε και προηγουμένως για στην υλοποίηση της συνάρτησης στροφής κατά μοίρες υπάρχει κάποιο σφάλμα. Συνεπώς προκειμένου να καλυφθεί το σφάλμα αυτό δοκιμάστηκαν πολυάριθμες σταθερές οι οποίες

προστίθενται στην γωνία PI/2 για κανονικοποίηση ώσπου τελικά καταλήξαμε στην τιμή norm που δίνεται στον κώδικα. Με τον τρόπο αυτό επιτυγχάνεται με μεγάλη ακρίβεια ορθή γωνία.

8. Πιθανές Μελλοντικές Βελτιώσεις

Έχοντας εκπληρώσει τα ζητούμενα της εργασίας μας, ο πράκτορας που υλοποιήθηκε είναι πλήρης και έτοιμος να καθαρίσει οποιοδήποτε χώρο χρησιμοποιώντας τους λιγοστούς πόρους και αισθητήρες που του δόθηκαν. Ωστόσο το γεγονός αυτό δεν σημαίνει ότι δεν θα μπορούσε να βελτιωθεί περαιτέρω η αποδοτικότητά του προσθέτοντας μεγαλύτερο βάθος στην λειτουργικότητά του. Ενδεικτικά μερικές απλές και σύντομες μελλοντικές βελτιώσεις που θα μπορούσαν να γίνουν είναι οι παρακάτω:

- Προσθήκη λειτουργικότητας SLAM. Δηλαδή ο πράκτοράς μας να εντοπίζει την θέση του μέσα στον κόσμο και να τον χαρτογραφεί
- Να προστεθεί manual χειρισμός των λειτουργιών του πράκτορα, ο οποίος θα καθιστά δυνατή και την διαχείριση των αλγορίθμων για επίτευξη βέλτιστης απόδοσης σε κάθε περιβάλλον
- Διεπαφή σε κινητό ή υπολογιστή για απομακρυσμένο χειρισμό της σκούπας.
- Συλλογή στατιστικών σε κάθε καθάρισμα, ανάλυση απόδοσης και βελτιστοποίηση.
- Προσθήκη εικονικών ορίων που η σκούπα δεν θα ξεπερνά παρέχοντας στον χρήστη μεγαλύτερο έλεγχο.
- Λογισμικό για optimal διαχείριση μπαταρίας. Σε μερικές περιπτώσεις διατήρηση σε μνήμη ήδη καθαρισμένης επιφάνειας, έως αποφόρτιση ώστε να γίνει συνέχεια του καθαρισμού μετά την επαναφόρτιση της μπαταρίας.

9. Βιβλιογραφία-Πηγές

- https://www.researchgate.net/publication/269297110_Path_planning_algorithm_development_for_autonomous_vacuum_cleaner_robots
- https://kth.diva-portal.org/smash/get/diva2:1214422/FULLTEXT01.pdf
- https://cyberbotics.com/doc/guide/create
- https://www.diva-portal.org/smash/get/diva2:1213970/FULLTEXT02