

Έκθεση εργασίας Ευφυών Αυτόνομων Συστημάτων

Ιωάννης Οικονομίδης

17 Ιουνίου 2023

Περιεχόμενα

1	Εισαγωγή	2
2	Παραδοχές και σχεδιαστικές επιλογές	2
3	Ανάλυση αλγορίθμου - κώδικα	3
3.1	Main	3
3.2	Env	3
3.3	MyRobot	4
3.4	Tools	4
3.5	SimpleBehaviors	5
3.5.1	SimpleBehaviors.stop	5
3.5.2	SimpleBehaviors.moveForward	5
3.5.3	SimpleBehaviors.orient	5
3.5.4	SimpleBehaviors.circumNavigate	6
3.6	IBug	6
3.6.1	Κατάσταση Orient	7
3.6.2	Κατάσταση MoveForward	8
3.6.3	Κατάσταση CircumNavigate	9
3.6.4	Κατάσταση Stop	10

1 Εισαγωγή

Ονομάζομαι Ιωάννης Οικονομίδης. Η εργασία αυτή υλοποιεί τον αλγόριθμο i-Bug στο περιβάλλον Simbad. Η συγκεκριμένη έκδοση του Simbad είναι το Simbad 1.7 που έχετε ανεβάσει στο e-Learning. Για τη Java3D χρησιμοποιώ την έκδοση Java3D 1.5.1. Το JDK που χρησιμοποιώ είναι το jdk1.8.0_311 που είναι αυτό που έχετε στις οδηγίες εγκατάστασης για τα Windows.

Στο συμπίεμένο αρχείο που ανέβασα στο e-Learning υπάρχει αυτό το pdf αρχείο που είναι η έκθεσή μου γραμμένη σε Latex, ο φάκελος "iBugProject" που είναι ο πλήρης φάκελος με το πηγαίο κώδικα του ρομπότ γραμμένος με το NetBeans και πέντε αρχεία βίντεο mp4, τα οποία είναι τα βίντεο εκτέλεσης του ρομπότ με τον αλγόριθμό μου. Έχω βάλει πέντε βίντεο επειδή έχω φτιάξει τόσα διαφορετικά περιβάλλοντα με εμπόδια όπως εξηγώ στη συνέχεια και υπάρχει ένα βίντεο εκτέλεσης ανά περιβάλλον. Τα βίντεο έχουν ονόματα "spheres.mp4", "box.mp4", "bottle.mp4", "spiral.mp4", και "cage.mp4".

Σημειώνεται πως στο τέταρτο βίντεο "spiral.mp4" και στο πέμπτο βίντεο "cage.mp4" επιλέγω time factor στο Simbad μεγαλύτερο από 1 λίγο μετά από την αρχή και ξαναεπιλέγω το 1 λίγο πριν το τέλος ώστε να κουνιέται γρήγορα το ρομπότ γιατί με time factor = 1 σε όλη τη κίνηση αργεί πάρα πολύ να την ολοκληρώσει. Επιπλέον, το ρομπότ πηγαίνει από τη λάθος μεριά του σπινάλ στο τέταρτο βίντεο "spiral.mp4", οπότε έχω γυρίσει στο αντίστοιχο σημείο τη κάμερα ώστε να φαίνεται καλύτερα. Παρόλα αυτά φτάνει στον στόχο. Επίσης όταν σας ρώτησα μου είπατε πως δεν χρειάζεται να βρίσκει το στόχο στο συγκεκριμένο περιβάλλον, οπότε δεν είναι τόσο σημαντικό, αλλά εφόσον το έκανα το βάζω και αυτό σαν βίντεο.

2 Παραδοχές και σχεδιαστικές επιλογές

- Έχω χρησιμοποιήσει τη μορφή κλάσεων που έχετε στο παράδειγμα RobotBehaviors που έχετε ανεβάσει στο e-Learning. Επιπλέον έχω χρησιμοποιήσει αρχιτεκτονική με καταστάσεις για την κλάση που υλοποιεί τον αλγόριθμο i-Bug, παρόμοια με την αρχιτεκτονική της κλάσης Bug1 του ίδιου παραδείγματος.
- Αντί για αισθητήρες επαφής που αναφέρει ο ψευδοκώδικας του αλγορίθμου i-Bug έχω χρησιμοποιήσει sonars. Ο τρόπος χρήσης τους αναλύεται παρακάτω.
- Κάνω τη παραδοχή πως η λάμπα έχει κενό χώρο στο πάτωμα γύρω της σε μια ακτίνα από το κέντρο της με μήκος ίσο με την εμβέλεια των sonars, δηλαδή ακτίνα 1.5m. Αλλιώς αν μπορούσαν να υπάρχουν και εμπόδια δίπλα στη λάμπα, το ρομπότ θα σταματούσε και όταν έχει εμπόδιο μπροστά του και η φωτεινότητα είναι παρόμοια με τη φωτεινότητα που μετράει όταν απέχει 0.6m από τη λάμπα.
- Σχεδόν όλα τα περιβάλλοντα είναι όμοια με αυτά που έχετε στο παράδειγμα BugAlgorithms και όλα δεν έχουν τοίχους στα όριά τους.

- Τέλος, στο περιβάλλον cage που είναι αδιέξοδο το ρομπότ δεν ανιχνεύει τον κύκλο πάντα με τη πρώτη φορά που περνάει από ολόκληρη τη περιφέρεια του εμποδίου. Μερικές φορές (γύρω στις 1/10 των περιπτώσεων) ανιχνεύει τον κύκλο με την ολοκλήρωση του δεύτερου ή τρίτου κύκλου λόγω της ανακρίβειας των αισθητήρων φωτός. Βέβαια πάλι όταν σας ρώτησα για αυτό μου είπατε πως δεν είναι απαραίτητο να σταματάει όταν το πρόβλημα είναι μη επιλύσιμο και πως το σημαντικό είναι να λύνει τα επιλύσιμα προβλήματα, αλλά το είχα ήδη κάνει, οπότε άφησα μέσα και αυτό το περιβάλλον.
- Οι υπόλοιπες σχεδιαστικές επιλογές και λεπτομέρειες υλοποίησης του αλγορίθμου αναφέρονται στη συνέχεια όπου αναλύεται ο κώδικας κάθε κλάσης. Αν σας ενδιαφέρει μόνο η υλοποίηση του αλγορίθμου i-Bug δείτε την ανάλυση του κώδικα των κλάσεων SimpleBehaviors και IBug.

3 Ανάλυση αλγορίθμου - κώδικα

Στα Source Packages του project υπάρχει ένα package με όνομα "ibugproject" μέσα στο οποίο βρίσκεται ο πηγαίος κώδικας όλου του project. Υπάρχουν έξι κλάσεις Java, η Main, η Env, η MyRobot, η Tools, η SimpleBehaviors, και η IBug. Στη συνέχεια αναλύεται τι κάνει η κάθε κλάση.

3.1 Main

Είναι η Main κλάση του προγράμματος που έχει και την αντίστοιχη main μέθοδο. Σε αυτή τη μέθοδο αρχικοποιείται το επιλεγμένο περιβάλλον και το πλαίσιο του Simbad. Υπάρχει μια γραμμή για την αρχικοποίηση κάθε ενός από τα πέντε διαθέσιμα περιβάλλοντα. Για την επιλογή ενός περιβάλλοντος αφήνεται χωρίς σχόλιο η γραμμή που αντιστοιχεί στο επιθυμητό περιβάλλον και όλες οι άλλες τέσσερις γίνονται σχόλια. Το default περιβάλλον είναι το spheres που έχει δύο σφαίρες ως εμπόδια. Τα υπόλοιπα διαθέσιμα περιβάλλοντα αναλύονται στην επόμενη κλάση Env.

3.2 Env

Είναι η κλάση του προγράμματος που αντιπροσωπεύει το περιβάλλον του Simbad και κληρονομεί τη κλάση EnvironmentDescription του Simbad. Στον κατασκευαστή χτίζεται το περιβάλλον που αντιστοιχεί στο string που δώθηκε ως παράμετρος με την αντίστοιχη private μέθοδο, όπως κάνατε και στη κλάση Main του παραδείγματος BugAlgorithms που έχετε στο e-Learning. Υπάρχουν πέντε διαθέσιμα περιβάλλοντα, από τα οποία τα τέσσερα έχουν σχεδόν ίδια τοπολογία με τα περιβάλλοντα που ανεβάσατε στο e-Learning στη κλάση που αναφέρθηκε προηγουμένως. Οι διαφορές από τα αντίστοιχα περιβάλλοντα του παραδείγματος είναι πως και οι δύο λάμπες του περιβάλλοντος είναι τοποθετημένες δύο μέτρα πάνω από το CherryAgent που είναι ο στόχος και η πρώτη είναι ανοιχτή ενώ η δεύτερη είναι σβηστή. Τα διαθέσιμα περιβάλλοντα (δηλαδή τα strings που τα επιλέγουν) είναι τα παρακάτω:

- **spheres**: ίδιο με το αντίστοιχο των παραδειγμάτων εκτός από τις λάμπες.
- **box**: ίδιο με το αντίστοιχο των παραδειγμάτων εκτός από τις λάμπες.
- **bottle**: ίδιο με το αντίστοιχο των παραδειγμάτων εκτός από τις λάμπες.
- **spiral**: ίδιο με το αντίστοιχο των παραδειγμάτων εκτός από τις λάμπες.
- **cage**: είναι ένα τετράγωνο με κενό στη μέση του, με το ρομπότ να έχει ως αρχική θέση το κέντρο του κενού του τετραγώνου και τη λάμπα να είναι έξω από τη περίμετρο του τετραγώνου, με αποτέλεσμα το ρομπότ να είναι παγιδευμένο.

3.3 MyRobot

Είναι η κλάση του προγράμματος που αντιπροσωπεύει το ρομπότ και κληρονομεί τη κλάση Agent του Simbad. Μοιάζει πάλι αρκετά με τις αντίστοιχες κλάσεις που έχετε στα παραδείγματα. Στον κατασκευαστή αρχικά τοποθετούνται στο ρομπότ οι παρακάτω αισθητήρες:

- **sonars**: 12 sonars τοποθετημένα περιμετρικά του ρομπότ.
- **centerLight**: αισθητήρας φωτός, τοποθετημένος στο κέντρο του ρομπότ με τον τρόπο που αναφέρατε στις διευκρινήσεις για την εργασία.
- **leftLight** και **rightLight**: αισθητήρες φωτός, τοποθετημένοι στην αριστερή και τη δεξιά μεριά του ρομπότ αντίστοιχα, με τις έτοιμες αντίστοιχες μεθόδους της κλάσης Agent.

Στη συνέχεια ενεργοποιείται η επιλογή να ενημερώνονται οι αισθητήρες φωτός σε κάθε frame ώστε να δίνουν πιο πρόσφατες μετρήσεις και αρχικοποιείται ένα αντικείμενο της κλάσης IBug που υλοποιεί τον αντίστοιχο αλγόριθμο.

Η μέθοδος `initBehavior` είναι κενή. Μέσα στην μέθοδο `performBehavior`, αν το ρομπότ δεν έχει ολοκληρώσει ακόμα τη κίνησή του εκτελείται το επόμενο βήμα του αλγορίθμου `iBug`, αλλιώς δεν εκτελείται τίποτα, δηλαδή το ρομπότ παραμένει ακίνητο. Οι λόγοι που μπορεί να οδηγήσουν στην ολοκλήρωση της κίνησης είναι ότι το ρομπότ έφτασε στον στόχο, δεν ανιχνεύει φως ή εντόπισε κύκλο κατά την περιμετρική του κίνηση.

3.4 Tools

Είναι κλάση που περιέχει στατικές μεθόδους που χρειάζονται οι άλλες κλάσεις του project. Περιέχει δύο μεθόδους που είναι ακριβώς ίδιες με τις αντίστοιχες μεθόδους της κλάσης `Tools` που έχετε στο παράδειγμα `robotBehaviots` που ανεβάσατε στο e-Learning. Αυτές είναι οι `getSensedPoint` και `wrapToPi`. Και οι δύο χρειάζονται κατά για την εκτέλεση της περιμετρικής κίνησης από το ρομπότ.

3.5 SimpleBehaviors

Είναι κλάση που περιέχει τις μεθόδους που υλοποιούν τις βασικές κινήσεις του αλγορίθμου i-Bug. Οι διάφορες σταθερές που ελέγχουν τις μέγιστες ταχύτητες και το threshold και χρησιμοποιούνται από τις μεθόδους της κλάσης παίρνουν τιμές μέσα στη κλάση, πριν τη δήλωση των μεθόδων.

3.5.1 SimpleBehaviors.stop

Είναι μέθοδος που θέτει τη γωνιακή και τη γραμμική ταχύτητα του ρομπότ στο μηδέν ώστε να σταματήσει να κουνιέται και να περιστρέφεται.

3.5.2 SimpleBehaviors.moveForward

Είναι μέθοδος η οποία υλοποιεί τη κίνηση u_{fwd} του αλγορίθμου i-Bug. Αρχικά η γωνιακή ταχύτητα του ρομπότ μηδενίζεται ώστε να μη περιστρέφεται. Στη συνέχεια υπολογίζεται η απόσταση από το κοντινότερο αντικείμενο που ανιχνεύουν τα sonars, δηλαδή η ελάχιστη απόσταση από εμπόδια. Λαμβάνονται υπόψη μόνο οι μετρήσεις των sonars που βρίσκονται στη μπροστινή μεριά του ρομπότ, δηλαδή αυτών που βρίσκονται στις ώρες 9 μέχρι 3 αν σκεφτούμε το sonar belt σαν να έχει ένα sonar τοποθετημένο σε κάθε ώρα ενός 12 ωρου ρολογιού (το 9 είναι η αριστερή μεριά και το 3 η δεξιά).

Παρακάτω, αν η ελάχιστη απόσταση είναι μικρότερη από την εμβέλεια των sonars, τότε η γραμμική ταχύτητα γίνεται ίση με το λόγο της ελάχιστης απόστασης προς την μέγιστη εμβέλεια των sonars επί την μέγιστη γραμμική ταχύτητα. Για παράδειγμα, αν η ελάχιστη απόσταση είναι 0.5m και η μέγιστη εμβέλεια 1.5m, τότε η γραμμική ταχύτητα θα είναι το $\frac{1}{3}$ της μέγιστης γραμμικής ταχύτητας. Αλλιώς αν η ελάχιστη απόσταση είναι μεγαλύτερη ή ίση με την εμβέλεια (στη πράξη δεν μπορεί να είναι μεγαλύτερη), τότε η γραμμική ταχύτητα γίνεται ίση με το λόγο της μέσης φωτεινότητας των δύο πλαγίων αισθητήρων φωτός προς τη φωτεινότητα του κεντρικού αισθητήρα φωτός, πολλαπλασιασμένη με τη μέγιστη γραμμική ταχύτητα. Αυτό χρειάζεται επειδή όσο πάει πιο κοντά το ρομπότ στη λάμπα, η φωτεινότητα των πλαγίων αισθητήρων φωτός μειώνεται, αλλά του κεντρικού παραμένει ίδια. Επομένως όσο πιο κοντά βρίσκεται στη λάμπα, τόσο πιο μικρή είναι η γραμμική ταχύτητα του ρομπότ και όσο πιο μακριά βρίσκεται, τόσο πιο μεγάλη είναι.

3.5.3 SimpleBehaviors.orient

Η τρίτη μέθοδος είναι η orient, η οποία υλοποιεί τη κίνηση u_{ori} του αλγορίθμου i-Bug. Αρχικά μηδενίζεται η γραμμική ταχύτητα. Παρακάτω, αν η μέση τιμή των δύο πλαγίων αισθητήρων φωτός είναι μηδέν, τότε η γωνιακή ταχύτητα γίνεται η μισή της μέγιστης επιτρεπτής γωνιακής ταχύτητας. Ο λόγος που χρειάζεται αυτό θα εξηγηθεί αργότερα. Αλλιώς αν η απόλυτη διαφορά της φωτεινότητας των δύο πλαγίων αισθητήρων είναι μικρότερη ή ίση από ένα άνω όριο (το οποίο λειτουργεί σαν μέγιστο επιτρεπτό σφάλμα), θεωρείται πως έχουν την ίδια φωτεινότητα, άρα το ρομπότ έχει ευθυγραμμιστεί με τη λάμπα, οπότε η γωνιακή ταχύτητά του μηδενίζεται.

Αλλιώς, δηλαδή αν η μέση φωτεινότητα των πλαγίων αισθητήρων δεν είναι μηδέν και η φωτεινότητα των δύο αισθητήρων διαφέρει περισσότερο από το επιτρεπτό όριο, τότε η γωνιακή ταχύτητα γίνεται ίση με τη διαφορά της φωτεινότητας των δύο πλαγίων αισθητήρων πολλαπλασιασμένη με τη μέγιστη γωνιακή ταχύτητα. Οπότε αν ο αριστερός αισθητήρας έχει μεγαλύτερη φωτεινότητα, το οποίο σημαίνει πως το ρομπότ πρέπει να περιστραφεί ανίστροφα με τη φορά του ρολογιού, θα περιστραφεί με θετική ταχύτητα, άρα αντίστροφα με τη φορά του ρολογιού. Το αντίθετο γίνεται αν ο δεξιός αισθητήρας έχει μεγαλύτερη φωτεινότητα.

Επιπλέον, όσο μεγαλύτερη διαφορά έχει η φωτεινότητα των δύο αισθητήρων, τόσο πιο γρήγορα θα περιστρέφεται το ρομπότ και όσο πιο μικρή είναι η διαφορά, τόσο πιο αργά θα περιστρέφεται. Άρα όσο απέχει μεγάλη γωνία από την επιθυμητή ευθυγραμμισμένη κατεύθυνση, θα περιστρέφεται γρήγορα και όσο πιο κοντά φτάνει στην ευθυγράμμιση, τόσο πιο αργά θα περιστρέφεται. Αυτό δουλεύει για ± 90 μοίρες από την επιθυμητή γωνία ευθυγράμμισης. Αν η τωρινή γωνία διαφέρει από την επιθυμητή γωνία ευθυγράμμισης περισσότερο (δηλαδή διαφορά από ± 90 μοίρες μέχρι ± 179.99 μοίρες), τότε γυρνάει όσο πιο αργά γίνεται στις ± 179.99 μοίρες και όσο μειώνεται η διαφορά (σε απόλυτη τιμή), αυξάνεται η ταχύτητα περιστροφής μέχρι και τις ± 90 μοίρες και μετά μειώνεται ξανά μέχρι να μηδενιστεί η διαφορά των γωνιών.

Η πρώτη συνθήκη του if χρειάζεται γιατί αν το ρομπότ αρχίσει σε θέση που έχει ακριβώς αντίθετη κατεύθυνση από τη λάμπα, τότε οι πλάγιοι αισθητήρες θα έχουν ίδια φωτεινότητα, άρα θα νομίζει το ρομπότ πως ευθυγραμμίστηκε, αλλά στη πραγματικότητα δεν ισχύει αυτό. Ωστόσο, στα πρώτα frame του simulation οι πλάγιοι αισθητήρες φωτός μετράνε μηδενική φωτεινότητα, οπότε βάζοντας γωνιακή ταχύτητα ίση με τη μισή της μέγιστης ταχύτητας περιστροφής στα πρώτα frames, το ρομπότ γυρνάει αντίστροφα από τη φορά του ρολογιού και ξεκολλάει από τη προβληματική αντίστροφη κατεύθυνση από τη λάμπα.

3.5.4 SimpleBehaviors.circumNavigate

Η τελευταία μέθοδος είναι η circumNavigate, η οποία υλοποιεί τη κίνηση u_{fol} του αλγορίθμου i-Bug και είναι ίδια με την αντίστοιχη μέθοδο της κλάσης SimpleBehaviors του παραδείγματος robotBehaviors που ανεβάσατε στο e-Learning.

3.6 IBug

Είναι κλάση που υλοποιεί τον αλγόριθμο i-Bug. Ακολουθεί την αρχιτεκτονική της κλάσης Bug1 του παραδείγματος robotBehaviors που ανεβάσατε στο e-Learning. Συγκεκριμένα, εκτός από τις σταθερές της κλάσης και τις μεταβλητές της, υπάρχει επίσης ένας enumerator με τέσσερις καταστάσεις. Αυτές είναι οι Orient, MoveForward, CircumNavigate και Stop. Αντιστοιχούν στις βασικές κινήσεις του αλγορίθμου i-Bug που υλοποιούνται από τις μεθόδους της κλάσης SimpleBehaviors. Δηλαδή Orient = u_{ori} , MoveForward = u_{fwd} , CircumNavigate = u_{fol} , και η Stop αντιστοιχεί στη κατάσταση κατά την οποία το ρομπότ έχει σταματήσει έχοντας ολοκληρώσει τη κίνησή του επειδή έφτασε στον στόχο, δεν ανιχνεύει φως

ή εντόπισε κύκλο κατά την περιμετρική του κίνηση. Η αρχική κατάσταση είναι η Orient, όπως λέει και ο αλγόριθμος i-Bug.

Σημειώνεται πως υπάρχει μια σταθερά που λέγεται DEBUG και έχει ως τιμή την false by default. Αν είναι ενεργοποιημένη, τότε εμφανίζεται σε κάθε frame κατά το οποίο εκτελείται ο υλοποιημένος αλγόριθμος i-Bug ένα μήνυμα στο command line που περιγράφει ποιά περίπτωση/βήμα του αλγορίθμου εκτελείται. Με την default false τιμή δεν εμφανίζονται επιπλέον μηνύματα στο command line, εκτός από όταν το ρομπότ έχει ολοκληρώσει τη κίνησή του όταν και εμφανίζεται ο λόγος για τον οποίο σταμάτησε τη κίνηση άσχετα από τη τιμή της σταθεράς DEBUG.

Υπάρχουν δύο μέθοδοι. Η μέθοδος step είναι αυτή που εκτελεί το επόμενο βήμα του αλγορίθμου. Αποτελείται από μια μεγάλη switch case εντολή, η οποία ανάλογα με την τωρινή κατάσταση εκτελεί τις ενέργειες που χρειάζονται και αλλάζει τη κατάσταση αν είναι απαραίτητο.

Η δεύτερη μέθοδος είναι η stopped, η οποία είναι getter για τη σημαία stopped που δείχνει αν το ρομπότ έχει ολοκληρώσει τη κίνησή του ή όχι. Χρειάζεται ώστε η μέθοδος performBehavior της κλάσης MyRobot να μην καλεί τη μέθοδο step αν το ρομπότ έχει ολοκληρώσει τη κίνησή του.

Οι τέσσερις καταστάσεις της κλάσης καθώς και ο τρόπος που χρησιμοποιούνται από τη πρώτη μέθοδο step της κλάσης αναλύεται παρακάτω.

3.6.1 Κατάσταση Orient

Για την κατάσταση Orient υπάρχουν οι εξής περιπτώσεις:

1. Αν ο κεντρικός αισθητήρας φωτός δίνει ως μέτρηση το NaN, δηλαδή δεν ανιχνεύει φως, τότε σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στη Stop, ώστε το ρομπότ να σταματήσει και να ολοκληρώσει τη κίνησή του επειδή δεν ανιχνεύει φως.
2. Αλλιώς αν ένας από τους πλάγιους αισθητήρες φωτός δίνει μη-μηδενική μέτρηση και η διαφορά των μετρήσεων των δύο αισθητήρων είναι αρκετά μικρή, οπότε το ρομπότ είναι ευθυγραμμισμένο με τη λάμπα, σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στη MoveForward, ώστε να κινηθεί μπροστά το ρομπότ στο επόμενο frame. Η πρώτη συνθήκη χρειάζεται γιατί στα πρώτα frames οι πλάγιοι αισθητήρες δίνουν πάντα μηδέν οπότε με βάση τη δεύτερη συνθήκη θεωρείται πως το ρομπότ έχει ευθυγραμμιστεί, το οποίο μπορεί να μην ισχύει αφού οι πλάγιοι αισθητήρες δεν δίνουν σωστές μετρήσεις ακόμα.
3. Αλλιώς, δηλαδή αν ο κεντρικός αισθητήρας ανιχνεύει φως και είτε και οι δύο πλάγιοι αισθητήρες δίνουν μηδενική μέτρηση, είτε η διαφορά τους είναι μεγαλύτερη από το όριο, ευθυγράμμισε το ρομπότ με τη λάμπα καλώντας την μέθοδο orient, χωρίς αλλαγή κατάστασης ώστε να συνεχιστεί η ευθυγράμμιση αν χρειάζεται στο επόμενο frame.

Ουσιαστικά η περίπτωση 3 υλοποιεί το πρώτο μέρος του βήματος 2 του i-Bug και η περίπτωση 2 υλοποιεί την αλλαγή του βήματος 2 από την u_{ori} στην u_{fol} .

3.6.2 Κατάσταση MoveForward

Όταν η κατάσταση είναι η MoveForward, πρώτα υπολογίζεται η απόσταση από το κοντινότερο εμπόδιο με βάση τις μετρήσεις των sonars (όπως περιγράφηκε η μέτρηση απόστασης και στην μέθοδο moveForward χρησιμοποιώντας μόνο τους αισθητήρες στις ώρες 9 έως 3) και η μέση μέτρηση των πλαγίων αισθητήρων φωτός. Στη συνέχεια, για την κατάσταση MoveForward υπάρχουν οι εξής περιπτώσεις:

1. Αν η απόλυτη τιμή της διαφοράς των μετρήσεων των πλαγίων αισθητήρων φωτός είναι μεγαλύτερη από το αντίστοιχο όριο που χρησιμοποιεί και η μέθοδος orient, δηλαδή το ρομπότ έχει πάψει να είναι ευθυγραμμισμένο με τη λάμπα, τότε σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στην Orient, ώστε να ξαναευθυγραμμιστεί το ρομπότ με τη λάμπα.
2. Αλλιώς αν η μέση μέτρηση φωτός των πλαγίων αισθητήρων έχει τη κατάλληλη τιμή, δηλαδή είναι ανάμεσα στις τιμές που δίνονται όταν το ρομπότ απέχει 0.6m από το στόχο τις οποίες τις βρήκα με πειράματα, και η ελάχιστη απόσταση από εμπόδια είναι ίση με τη μέγιστη εμβέλεια των sonars, οπότε τα sonars δεν ανιχνεύουν εμπόδιο, τότε θέσε τη σημαία hitGoal στην αλήθεια, σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στην Stop, ώστε το ρομπότ να σταματήσει και να ολοκληρώσει τη κίνησή του επειδή βρίσκεται αρκετά κοντά (απόσταση μικρότερη από 0.6m) στον στόχο-λάμπα.
3. Αλλιώς αν το κοντινότερο εμπόδιο βρίσκεται σε μικρότερη από την ασφαλή απόσταση, τότε αρχικοποίησε τη φωτεινότητα του αρχικού σημείου της περιφερειακής κίνησης στο -1, απενεργοποίησε τη σημαία που ενεργοποιεί τον έλεγχο για κύκλο και αποθήκευσε τη τωρινή μέτρηση του κεντρικού αισθητήρα φωτός ως το τοπικό μέγιστο της φωτεινότητας. Επιπλέον, σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στην Circumnavigate, ώστε το ρομπότ να αρχίσει να κινείται στη περιφέρεια του εμποδίου από το επόμενο frame.
4. Αλλιώς αν καμία από τις προηγούμενες συνθήκες δεν ικανοποιείται, δηλαδή το ρομπότ δεν έχει λόγο να σταματήσει τη μπροστινή κίνησή του, κάνε το ρομπότ να κινηθεί μπροστά καλώντας την μέθοδο moveForward, χωρίς αλλαγή κατάστασης ώστε να ελέγξει αν μπορεί να κινηθεί μπροστά ξανά στο επόμενο frame.

Σύμφωνα με τον i-Bug, η u_{fwd} μπορεί να σταματήσει αν το ρομπότ αντιληφθεί τοπικό μέγιστο στην ένταση του σήματος και τότε πάει πίσω στο βήμα 1 του αλγορίθμου αν δεν υπάρχει εμπόδιο, δηλαδή στην u_{ori} . Ωστόσο, επειδή ο κεντρικός αισθητήρας δίνει σχεδόν ίδια τιμή εφόσον το ρομπότ είναι ευθυγραμμισμένο με τη λάμπα άσχετα από το πόσο κοντά είναι κάτω από μια απόσταση, δεν βρήκα χρήση για αυτή τη συνθήκη. Οπότε, την αντικατέστησα με τη περίπτωση 1, σύμφωνα με την οποία αν σταματήσει να είναι ευθυγραμμισμένο το ρομπότ, τότε σταματάει τη μπροστινή κίνηση και ευθυγραμμίζεται, αντικαθιστώντας αυτό το κομμάτι του i-Bug.

Η περίπτωση 2 αντιστοιχεί στο βήμα 3 του i-Bug, απλά αντί για ένταση = 1 ελέγχεται η μέση μέτρηση των πλαγίων αισθητήρων φωτός και αν υπάρχει χώρος κάτω από τη λάμπα. Η περίπτωση 3 υλοποιεί τα βήματα 4 και 6 του αλγορίθμου. Μόλις το ρομπότ ανιχνεύσει εμπόδιο, θέτει τη μέγιστη τιμή φωτεινότητας ως τη τωρινή όπως και στο βήμα 4 του i-Bug και αρχίζει τη περιμετρική κίνηση στο επόμενο frame όπως λέει το βήμα 6. Το βήμα 5 είναι περιττό αφού για να μπήκε σε αυτό το κομμάτι κώδικα, σίγουρα υπάρχει εμπόδιο οπότε δεν μπορεί να κινηθεί μπροστά. Η περίπτωση 4 αντιστοιχεί στο δεύτερο κομμάτι του βήματος 2, σύμφωνα με το οποίο η μπροστινή κίνηση συνεχίζεται μετά την ευθυγράμμιση.

3.6.3 Κατάσταση CircumNavigate

Για την κατάσταση CircumNavigate υπάρχουν οι εξής περιπτώσεις:

1. Αν η διαφορά της μέτρησης του κεντρικού αισθητήρα φωτός από το τωρινό τοπικό μέγιστο φωτεινότητας είναι μεγαλύτερη από ένα σταθερό όριο, δηλαδή υπάρχει νέο τοπικό μέγιστο στην ένταση της φωτεινότητας, τότε σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στην Orient, ώστε να ξαναευθυγραμμιστεί το ρομπότ με τη λάμπα και μετά να κάνει μπροστινή κίνηση αν μπορεί, αλλιώς να αρχίσει νέα περιφερειακή κίνηση.
2. Αλλιώς αν η απόλυτη τιμή της διαφοράς της μέτρησης του κεντρικού αισθητήρα φωτός από τη φωτεινότητα του αρχικού σημείου της περιφερειακής κίνησης είναι μικρότερη από ένα όριο και έχει ενεργοποιηθεί ο έλεγχος για κύκλο, δηλαδή το ρομπότ έχει επιστρέψει κοντά στην θέση από την οποία άρχισε την περιφερειακή κίνηση, τότε σταμάτησε το ρομπότ με τη μέθοδο stop και άλλαξε τη κατάσταση στην Stop, ώστε το ρομπότ να σταματήσει και να ολοκληρώσει τη κίνησή του λόγω ύπαρξης κυκλού, δηλαδή επειδή δεν βρήκε τρόπο να ξεφύγει από το κοντινότερο εμπόδιο αφού έλεγξε ολόκληρη τη περιφέρειά του.
3. Αλλιώς, δηλαδή αν η περιφερειακή κίνηση δεν χρειάζεται να σταματήσει ούτε λόγω νέου τοπικού μεγίστου φωτεινότητας ούτε λόγω εντοπισμού κύκλου, τότε συνέχισε την περιφερειακή κίνηση. Συγκεκριμένα, αν η φωτεινότητα του αρχικού σημείου από το οποίο άρχισε η περιφερειακή κίνηση είναι -1, δηλαδή δεν έχει αποθηκευτεί η φωτεινότητα του αρχικού σημείου ακόμα άρα το τωρινό σημείο είναι το αρχικό, τότε αποθήκευσε τη μέτρηση του κεντρικού αισθητήρα φωτός ως τη φωτεινότητα του αρχικού σημείου. Αλλιώς αν η απόλυτη τιμή της διαφοράς της μέτρησης του κεντρικού αισθητήρα φωτός από τη φωτεινότητα του αρχικού σημείου της περιφερειακής κίνησης είναι μεγαλύτερη από ένα όριο και δεν έχει ενεργοποιηθεί ο έλεγχος για κύκλο, δηλαδή το ρομπότ βρίσκεται αρκετά μακριά από το αρχικό σημείο της περιφερειακής κίνησης ώστε να μην είναι τόσο κοντά η φωτεινότητα του τωρινού σημείου από τη φωτεινότητα του αρχικού σημείου για να νομίζει αμέσως λανθασμένα πως υπάρχει κύκλος, τότε ενεργοποίησε τον έλεγχο για κύκλο. Αλλιώς μην κάνεις κάτι επιπλέον. Και στις τρεις αυτές περιπτώσεις καλείται η μέθοδος CircumNavigate, ώστε το ρομπότ να συνεχίσει τη περιφερειακή κίνηση και

δεν αλλάζει η κατάσταση ώστε να προσπαθήσει να κάνει περιφερειακή κίνηση ξανά στο επόμενο frame.

Ουσιαστικά η περίπτωση 1 υλοποιεί το βήμα 7 του i-Bug. Η περίπτωση 2 δεν έχει αντίστοιχη στον αλγόριθμο i-Bug γιατί δεν περιγράφει τι γίνεται σε περίπτωση που το ρομπότ περάσει από όλη τη περιφέρεια του κοντινότερου αντικειμένου χωρίς να βρει τοπικό μέγιστο εκτός από το αρχικό σημείο και χωρίς να μπορεί να κάνει μπροστινή κίνηση από το αρχικό σημείο. Υποθέτω πως σταματάει, το οποίο υλοποιεί η δεύτερη περίπτωση.

Η πρώτη υποπερίπτωση της περίπτωσης 3 υλοποιεί το βήμα 6 του i-Bug σε συνεργασία με την τρίτη περίπτωση της κατάστασης MoveForward σύμφωνα με το οποίο αρχίζει η περιφερειακή κίνηση. Η δεύτερη υποπερίπτωση είναι βοηθητική για τον εντοπισμό κύκλου και ο αλγόριθμος i-Bug δεν αναφέρει κάτι αντίστοιχο. Η τρίτη υποπερίπτωση, δηλαδή το να μην γίνει κάποια έξτρα εργασία, αντιστοιχεί στο βήμα 8 του i-Bug όπου η περιφερειακή κίνηση συνεχίζεται.

3.6.4 Κατάσταση Stop

Όταν η κατάσταση είναι η Stop, πρώτα η σημαία stopped παίρνει τη τιμή true ώστε να δείχνει πως το ρομπότ έχει σταματήσει και ολοκληρώσει τη κίνησή του και σταματάει το ρομπότ καλώντας τη μέθοδο stop. Στη συνέχεια, για την κατάσταση Stop υπάρχουν οι εξής περιπτώσεις:

1. Αν η σημαία hitGoal που δείχνει αν το ρομπότ έφτασε στο στόχο είναι ενεργή, τότε εμφανίζεται το αντίστοιχο μήνυμα επιτυχίας.
2. Αλλιώς αν η μέτρηση του κεντρικού αισθητήρα φωτός είναι NaN, δηλαδή ο αισθητήρας δεν ανιχνεύει φως, τότε εμφανίζεται αντίστοιχο μήνυμα αποτυχίας.
3. Αλλιώς, δηλαδή αν το ρομπότ δεν έφτασε στο στόχο αλλά ο κεντρικός αισθητήρας ανιχνεύει φως, αυτό σημαίνει πως υπάρχει κύκλος αλλιώς δεν θα έφτανε στη κατάσταση Stop το ρομπότ υπό αυτές τις συνθήκες, οπότε εμφανίζεται το αντίστοιχο μήνυμα αποτυχίας.

Η περίπτωση 1 της κατάστασης Stop αντιστοιχεί στο βήμα 3 του i-Bug σε συνεργασία με την δεύτερη περίπτωση της κατάστασης MoveForward. Οι άλλες δύο περιπτώσεις δεν έχουν άμεση αντιστοιχία με τον αλγόριθμο i-Bug αφού δεν αναφέρει τη γίνεται αν η ένταση του σήματος είναι μηδενική ή υπάρχει κύκλος κατά την περιφερειακή κίνηση.