ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΘΕΜΑ : ΠΛΟΗΓΗΣΗ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ (NAVIGATION FOR ROBOTIC SYSTEMS)

ΛΕΒΕΝΤΗΣ ΔΗΜΗΤΡΙΟΣ

(Α.Μ.40706)

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Δρ. ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΙΓΑΛΕΩ ΙΟΥΛΙΟΣ 2018

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος Λεβέντης Δημήτριος του Εμμανουήλ, φοιτητής του Τμήματος **Μηχανικών Αυτοματισμού Τ.Ε,** του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε πρέπει να ολοκληρώσει εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσής της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18. παρ.5 του ισχύοντος Εσωτερικού Κανονισμού».

Ο Δηλών **Ημερομηνία**

Λεβέντης Δημήτριος

Περίληψη

Η παρούσα πτυχιακή πραγματεύεται την πλοήγηση ρομποτικών συστημάτων και πιο συγκεκριμένα σε περιβάλλον εξομοίωσης.

Σε γενική βάση, να αναφέρουμε ότι η εκτέλεση των διερευνητικών πλοηγήσεων, πραγματοποιείται, η μεν πρώτη στο ανοιχτό λογισμικό Linux Ubuntu, και η δε δεύτερη σε λογισμικό Microsoft Windows 10 περιορισμένης χρήσης. Μέσα από αυτή την εργασία αναδεικνύονται δύο (2) διαφορετικές διερευνητικές πλοηγήσεις ρομποτικών συστημάτων. Στην πρώτη διερευνητική πλοήγηση ρομποτικού συστήματος πραγματοποιείται αφενός, η χρήση της πλατφόρμας ανοιχτού κώδικα Ros Indigo, αφετέρου το αλληλοεπιδρών περιβάλλον προσομοίωσης Gazebo το οποίο είναι και αυτό ανοιχτού κώδικα. Στη δεύτερη διερευνητική πλοήγηση ρομποτικού συστήματος, χρησιμοποιείται η μαθηματική υπολογιστική πλατφόρμα του Matlab, το οποίο αλληλεπιδρά με την εκπαιδευτική έκδοση του λογισμικού προσομοίωσης Vrep περιορισμένης χρήσης.

Πιο συγκεκριμένα, ξεκινώντας την Πτυχιακή Εργασία και προκειμένου αυτή να γίνει ευκολότερα αντιληπτή από τον αναγνώστη, παρατίθενται κάποια θεωρητικά στοιχεία, τα οποία, κατόπιν εκτενούς μελέτης, χρησιμοποιήθηκαν ως βάσεις για την υλοποίηση του σκοπού της. Εν συνεχεία, εκτελείται ένα σύνολο ενεργειών με παραδείγματα, τα οποία εστιάζουν στην απόλυτη κατανόηση των διερευνητικών πλοηγήσεων. Ένα από τα παραδείγματα αυτά, αποτελεί η δημιουργία χάρτη προσομοίωσης και η παραμετροποίησή του, έτσι ώστε αυτός να είναι συμβατός με την πλατφόρμα, με τελικό στόχο τον συντονισμό τους. Ακόμη ένα παράδειγμα, αποτελεί η προσθήκη του ρομποτικού συστήματος και η παραμετροποίηση του στο περιβάλλον προσομοίωσης. Στη συνέχεια, παρατίθεται η τοπικοποίηση του ρομπότ και η αναγνώριση του χάρτη εξομοίωσης, για να καταλήξουμε τελικά στην πλοήγησή του σε αναγνωρισμένο και μη, περιβάλλον με τη βοήθεια, κάθε φορά, διαφορετικών ρομποτικών εφαρμογών.

Καταλήγοντας, παρατίθεται μια σειρά συμπερασμάτων μερικά από τα οποία αφορούν τη σύγκριση μεταξύ των δύο (2) διαφορετικών διερευνητικών πλοηγήσεων ρομποτικών συστημάτων, καθώς και την ανταπόκριση, κάθε φορά, του καθενός ξεχωριστά στις απαιτήσεις της πλοήγησης.

Κλείνοντας, είναι απαραίτητο να σημειώσουμε πως οι υπάρχουσες έρευνες στην πλοήγηση των ρομποτικών συστημάτων βρίσκονται ακόμα σε πρώιμο στάδιο, ενώ μελλοντικά, πέρα από τη χρήση τους για επιστημονικές μελέτες, για στρατιωτικές βάσεις αλλά και διαστημικές αποστολές,

αναμένεται η ολοένα και μεγαλύτερη είσοδός τους στις καθημερινές δραστηριότητες καθώς και στην εξυπηρέτηση των ειδικών αναγκών ανθρώπων με κινητικά προβλήματα.

Abstract

This thesis deals with the navigation of robotic systems and more specifically with environmental simulation.

In general it focuses on two different exploratory navigation researches. The first research is set up of Ubuntu Linux, which uses the Ros Indigo open source platform in combination with interactive Gazebo environmental simulation, which is also an open source simulation platform. The second research is set up of Microsoft Windows '10 and uses the computational platform Matlab, which interacts with the simulation software platform of the educational version of Vrep.

Beginning with the thesis, and more specifically trying to make it comprehendible to the reader, there are some theoretical elements mentioned. Fundamental knowledge which has been reached after extensive study and this has been used as a basis for the adaptation of the thesis' purpose. Furthermore, overall actions have been provided through examples which focus on the ultimate understanding of exploratory navigation which is performed. One of these examples is the creation of a simulation map and its parameterization so that it will be compatible with the platform that is being used every time. The purpose of this compatibility is to gain the ultimate goal and the coordination of each movement of the robot. Another example of this, is the addition of a robotic system and its configuration in the environmental simulation.

In continuation, there is a reference to the robot localization which is being listed for the identification of the simulation map, in order to navigate to a recognized and unmanaged environment with the help of a different robotic application each time. A series of examples are set up, some of which are related to the comparison of the two different exploratory robotic navigation systems.

Summarizing, it is necessary to mention that the existing robotic navigation system researches are still at an early stage. Their purpose will be for use in scientific studies, for military bases, space missions, developing and improving our daily lives, and especially meeting the needs of people with mobility limitation in the near future and beyond.

Περιεχόμενα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Γρηγόριο Νικολάου. Η εμπιστοσύνη του, η καθοδήγηση και ο πολύτιμος χρόνος που αφιέρωσε ήταν τα πιο σημαντικά συστατικά, για την ολοκλήρωση της πτυχιακής μου εργασίας με επιτυχία.

Ιδιαίτερες ευχαριστίες και απέραντη ευγνωμοσύνη θα ήθελα να εκφράσω για τους αγαπημένους μου γονείς, οι οποίοι με υποστήριξαν και μου συμπαρασταθήκαν όλα αυτά τα χρόνια για την αποπεράτωση των σπουδών μου. Τα λόγια είναι περιττά. Τους αγαπώ πολύ.

Τέλος ένα μεγάλο ευχαριστώ σε όλους τους καθηγητές μου, που όλα αυτά τα χρόνια αγωνίστηκαν για να μου μεταδώσουν γνώσεις και μόρφωση. Απώτερος σκοπός των κόπων τους, να με κάνουν καλύτερο, ικανότερο ώστε να με προετοίμασουν κατάλληλα για την αγορά εργασίας και την ζωή γενικότερα.

Εισαγωγή-Πρόλογος

Η Ρομποτική είναι ο κλάδος της επιστήμης που μελετά τις μηχανές εκείνες που μπορούν να αντικαταστήσουν τον άνθρωπο στην εκτέλεση μιας εργασίας, η οποία συνδυάζει τη φυσική δραστηριότητα με τη διαδικασία λήψης αποφάσεων.Η ανάπτυξη ρομποτικών συστημάτων έχει σημειώσει ικανοποιητική πρόοδο στη διάρκεια των τελευταίων δέκα χρόνων. Σήμερα υπάρχουν παραδείγματα ρομποτικών εφαρμογών στη βιομηχανία για την εκτέλεση εργασιών σε συνεργασία με τους ανθρώπους. Υπάρχουν, ωστόσο, αρκετά κομμάτια του συγκεκριμένου κλάδου που πρέπει να αναπτυχθούν, ώστε αυτά τα συστήματα να αλληλεπιδρούν με ασφάλεια με τους ανθρώπους.

Κάποιες από τις εφαρμογές που αναμένεται να εξελιχθούν είναι ο τομέας των ανθρωποειδών ρομπότ αναμένεται να είναι στους τομείς της βοήθειας κατ' οίκον για τη φροντίδα των ηλικιωμένων, καθώς και ατόμων με ειδικές ανάγκες. Άλλος ένας τομέας είναι της υποβοήθησης των ανθρώπων στην αντιμετώπιση έκτακτων καταστάσεων και κρίσεων, που συμβαίνουν μετά από σοβαρές καταστροφές. Σταδιακά αυτά τα ρομπότ εισέρχονται και σε άλλες εφαρμογές, όπως η φύλαξη, η εκτέλεση εργασιών εφοδιασμού σε επιχειρήσεις, διάφορες εργασίες σε συνεργασία με τους ανθρώπους ή εργασίες που είναι δύσκολες και κουραστικές για τον άνθρωπο να τις εκτελέσει μόνος του.

Η ανάπτυξη της ρομποτικής τεχνολογίας και η χρήση των ρομπότ στο περιβάλλον θα δημιουργίσει νέες θέσεις εργασίας, χωρίς να τις στερήσει από τους ανθρώπους. Η κατασκευή ρομπότ έχει δημιουργίσει ένα νέο βιομηχανικό τομέα παρόμοιο με την αυτοκινητοβιομηχανία. Η χρήση των ρομπότ απαιτεί ανθρώπους εξειδικευμένους πάνω στα ρομποτικά μηχανήματα, ώστε να μπορού να τα πραγραμματίζουν ανάλογα με τις απαιτήσεις της εκάστοτε εργασίας, να συντηρούν και να ελέγχουν τη λειτουργία τους. Η ανάγκη της τεχνικής συντήρησης γι' αυτά τα ρομπότ έχει ανοίξει νέες θέσεις εργασίας, που παρέχουν τεχνικές υπηρεσίες παρόμοιες με τις τεχνικές υπηρεσίες του αυτοκινήτου σήμερα, για βλάβες, συντήρηση, ανταλλακτικά κλπ. Επιπλέον, οι περισσότερες από τις εφαρμογές των ρομπότ απαιτούν την εποπτεία από έναν άνθρωπο και τη συνεργασία με έναν άνθρωπο χειριστή. Ρομπότ και άνθρωποι συνεργάζονται στενά, έτσι ώστε να αυξηθεί η παραγωγικότητα και να πέσει η τιμή του προϊόντος, τα οποία έχουν σαν αποτέλεσμα μεγαλύτερη αγοραστική ικανότητα του καταναλωτή και μεγαλύτερη ζήτηση. Η εξέλιξη της ρομποτικής όχι μόνο δημιουργεί νέες θέσεις εργασίας αλλά κάνει και τη ζωή των ανθρώπων πιο εύκολη. Δεν θα πρέπει να εκφράζουμε την ανησυχία μας για την ραγδαία εξέλιξη της ρομποτικής, στην οποία οφείλεται η δημιουργία νέων θέσεων εργασίας για τους ανθρώπους. αλλά περισσότερο ανησυχία θα πρέπει να εκφράσουμε για την πορεία του (υπό ανάπτυξη) τομέα της τεχνητής νοημοσύνης.

Η ανάπτυξη της τεχνητής νοημοσύνης είναι ακόμη σε πρώιμο στάδιο. Βασικός στόχος της έρευνας είναι η κατασκευή ρομπότ με βασικές ικανότητες λήψης αποφάσεων καθώς και η ικανότητα τους να αντιλαμβάνονται τα συναισθήματα των ανθρώπων. Αυτό μπορεί να επιτευχθεί μέσα από μία τυποποιημένη διαδικασία αλγόριθμων οι οποίοι δίνουν δυνατότητα στο ρομποτικό σύστημα να αποθηκεύει τα ερεθίσματα ή διαταράχες από το περιβάλλον που βρίσκεται και να το αποθηκεύει σαν εμπειρία στη μνήμη του. Έτσι το ρομποτικό σύστημα αποκομίζει εμπειρίες. Οι εμπειρίες αυτές κάθε φορά που το ρομποτικό σύστημα έχει ένα στόχο, συνδυάζονται μεταξύ τους έτσι ώστε να καταλήξει στο πιθανότερα σωστό αποτέλεσμα, αποφεύγοντας έτσι τον προγραμματισμό τεράστιων αλγορίθμων οι οποίοι καθυστερούν το ρομποτικό σύστημα από την λήψη γρήγορής απόφασης. Αυτό βέβαια δίνει μία επικίνδυνη ελευθερία στο ρομπότ, το οποίο θα πρέπει να ληφθεί υπόψιν από τον υπεύθυνο προγραμματισμού, έτσι ώστε το ρομποτικό σύστημα να μπορεί να το χειριστεί ανα πάσα στιγμή και να μη βγει εκτός ελέγχου ή πέσει σε λάθος χέρια. Ειδικά η κατασκευή ρομπότ που θα μπορούν να αντιλαμβάνονται τα συναισθήματα των ανθρώπων, θα πάρει πολύ περισσότερο χρόνο για να υλοποιηθεί. Η τεχνολογία μελλοντικά θα φτάσει και εκεί, καθώς η πλήρης αποδοχή αυτών των ρομπότ στο περιβάλλον μας θα απαιτήσει όχι μόνο μηχανές που μπορούν να ελέγχονται από ανθρώπους αλλά και να είναι σε θέση να εκτελούν εργασίες μαζί τους. Τέλος θα είναι σε θέση να αλληλοεπιδράσουν γνωστικά με τους ανθρώπους και να κατανοήσουν τα συναισθήματα των ανθρώπων χειριστών τους.

Σκοπός Πτυχιακής Εργασίας

Ο σκοπός της 1ης και 2ης διερευνητικής πλοήγησης ρομποτικών συστημάτων είναι η μελέτη, κατανόηση και ανάπτυξη κώδικα έτσι ώστε να μπορέσουν να συνδυάστούν όλες τις γνώσεις που μεταλαμπάδευσαν οι δάσκαλοι-καθηγητές κατά τη διάρκεια των εκπαιδευτικών εξαμήνων στο Τμήμα του Αυτοματισμού. Θα μελετήθεί και θα διερευνήθεί ο ανοικτός ρομποτικός κώδικας, μέσω της διεθνής βιβλιογραφίας ώστε να υπάρχει ικανότητα και οι απαραίτητες γνώσεις για να σχεδιάστεί και να παραμετροποιήθεί με τη βοήθεια του οποίου το ρομποτικό σύστημα θα κάνει αυτόνομη πλοήγηση σε γνωστό και άγνωστο περιβάλλον προσομοίωσης. Το γνωστό περιβάλλον προσομοίωσης θα δημιουργηθεί μέσω χαρτογράφησης του χώρου με τη βοήθεια ανοιχτού κώδικα τηλεκίνησης.Αυτό σημαίνει ότι θα τυποποιηθεί μία διαδικασία μέσω της εύρεσης καλά ορισμένων βημάτων, τα οποία πρέπει να ακολουθηθούν για να παραχθεί το επιθυμητό αποτέλεσμα.

Κεφάλαιο 1

Υπόβαθρο Γενικών Εννοιών

1.1.1 Τι είναι ο αυτοματισμός;

Ο αυτοματισμός είναι ένα σύνολο ελέγχου διεργασιών μηχανικών, ηλεκτρολογικών, ηλεκρονικών που μέσα από μία τυποποιημένη διαδικασία ενός συστήματος επιδιώκουμε να φτάσουμε στο επιθυμητό αποτέλεσμα. Για παράδειγμα, βάζοντας σε ένα σύστημα μία τυχαία είσοδο θέλουμε να πάρουμε μία επιθυμητή έξοδο μέσα από ένα σύστημα ανάδρασης είτε με ελεγκτή, είτε με αλγόριθμο, έτσι ώστε να μπορέσουμε να διατηρήσουμε την καθορισμένη επιθυμητή κατάσταση που θέλουμε να πάρουμε στην έξοδο χώρις ανθρώπινη παρέμβαση.

Αυτοματισμός, είναι το πεδίο της επιστήμης και της τεχνολογίας που ασχολείται με την επιβολή επιθυμητής συμπεριφοράς σε διαφορετικές συνθήκες περιβάλλοντος (διαταραχές). Αυτό επιτυγχάνεται με την κατανόηση των μηχανισμών μέσω των οποίων καθορίζεται η λειτουργία μίας διαταραχής. Αυτοματισμός ονομάζεται και η Επιστήμη του Ελέγχου.Το αντικείμενο του Αυτοματισμού, είναι γενικό και πολύπλευρο. Εφαρμογές του συναντάμαι πολυάριθμες στην καθημερινή ζωή και στη βιομηχανία.

Ο Αυτοματισμός είναι ένα από τα πιο “ιστορικά” πεδία της επιστήμης, διότι η ανάπτυξή του συνοδεύει την εξέλιξη όλων των άλλων τεχνολογιών. Η γνώση του Αυτοματισμού, επομένως, αποτελεί γνώση της “τεχνολογικής ιστορίας” μας και της κληρονομιάς μας. Στην περίπτωση του Αυτοματισμού, η ιστορία αυτή είναι ιδιαίτερα πλούσια, μιας και οι Αρχαίοι Έλληνες επέδειξαν ιδιαίτερη εφευρετικότητα και ανέπτυξαν πολλές και σημαντικές λύσεις αυτοματισμού, που χρησιμοποιούμε μέχρι και σήμερα. Χάρη στην καθολική και γενικευμένη διάδοση των εφαρμογών του, ο Αυτοματισμός αποκτά ένα σημαντικό ρόλο στη ζωή μας.

1.1.2 Κατανόηση του Αυτοματισμού μέσα από παραδείγματα

Συχνά αναφερόμαστε σε μια "αυτόματη" συσκευή π.χ. μια φωτογραφική μηχανή, που επιλέγει το χρόνο έκθεσης και το διάφραγμα, χωρίς να χρειάζεται ρύθμιση από το φωτογράφο ή ένα κιβώτιο ταχυτήτων που αλλάζει σχέση μετάδοσης, χωρίς να κινήσουμε το μοχλό ταχυτήτων.

Επίσης, συμβαίνει να έχουμε μια ακούσια κίνηση ή ρίγος και να λέμε ότι έγινε αντανακλαστικά ή "αυτόματα", δηλαδή ότι υπακούει σε κάποια εσωτερική εντολή του οργανισμού, χωρίς να προέρχεται από τη δική μας βούληση ή να μπορεί να ελεγχθεί από το νου.

Γενικότερα, ονομάζουμε "αυτόματες" αυτές τις μηχανές και τις διατάξεις, που εκτελούν τις αναμενόμενες λειτουργίες "από μόνες τους", δηλαδή χωρίς την καταβολή της ανθρώπινης προσπάθειας.

Ο Αυτοματισμός είναι το πεδίο της επιστήμης και της τεχνολογίας που ασχολείται με αυτές ακριβώς τις διεργασίες. Ειδικότερα, ο Αυτοματισμός περιλαμβάνει:

- την εξέταση και κατανόηση των μηχανισμών μέσω των οποίων μία διεργασία, ένα σύστημα οδηγείται στο να έχει τη μια ή την άλλη συμπεριφορά. Αυτή η αντίληψη των αιτίων που καθορίζουν τις λειτουργίες ενός σύστηματος ονομάζεται και ανάλυση των συστημάτων.
- τον έλεγχο, δηλαδή την επιβολή σε ένα σύστημα μίας επιθυμητής ή της συμφέρουσας συμπεριφοράς ή, ακόμη, την αποτροπή μιας επικίνδυνης ή ζημιογόνας εξέλιξης.

Οι δύο αυτές διαστάσεις του Αυτοματισμού είναι, βέβαια, αλληλένδετες και συμπληρωματικές. Η ικανότητα να ελέγξουμε ένα σύστημα στηρίζεται, καταρχήν, στην προηγούμενη κατανόηση των επιμέρους γεγονότων και συνθηκών που το προκαλούν. Έτσι, στο παράδειγμα του κιβωτίου ταχυτήτων, πρέπει πρώτα να κατανοήσουμε πότε και γιατί ο οδηγός αλλάζει τις σχέσεις μετάδοσης και, έπειτα, να υποκαταστήσουμε αυτόν το χειρισμό με μια "αυτόματη" διάταξη (το αυτόματο κιβώτιο ταχυτήτων).

Όπως και τα άλλα πεδία της σύγχρονης τεχνολογίας, ο Αυτοματισμός ενσωματώνει όλες τις σχετικές δραστηριότητες και αποτελέσματα, δηλαδή:

- τις μεθόδους και τεχνικές που χρησιμοποιούνται για την ανάλυση και τον έλεγχο των συστημάτων με μαθηματικές ή άλλες μεθόδους, καθώς και τις γενικότερες θεωρητικές προσεγγίσεις από τις οποίες προέρχονται και στις οποίες στηρίζονται αυτές οι μέθοδοι
- όλα τα βήματα της διαδικασίας ανάπτυξης των αυτόματων συστημάτων: την αρχική θεώρηση, σύλληψη, τη σχεδίαση, την κατασκευή, την εγκατάσταση, τη δοκιμαστική λειτουργία και την οριστική εφαρμογή
- τις διάφορες τεχνολογίες που χρησιμοποιούνται ή υποβοηθούν στην εφαρμογή των μεθόδων του αυτοματισμού και περιλαμβάνουν όργανα και ειδικό εξοπλισμό, τεχνικές γνώσεις, τεκμηρίωση κλπ
- τις ίδιες τις τεχνητές ή φυσικές διατάξεις και μηχανές που χρησιμοποιούμε για να επιβάλλουμε την επιθυμητή συμπεριφορά και οι οποίες, επομένως, επιτελούν την "αυτόματη" λειτουργία.

Όλες αυτές οι περιοχές συναποτελούν την "ύλη" του Αυτοματισμού και το αντικείμενο εργασίας των τεχνικών που απασχολούνται σε αυτόν τον τομέα (οι Τεχνικοί Αυτοματισμού).

1.1.3 Αυτοματισμοί σε οικιακές συσκευές

Ένα τεράστιο πλήθος από εφαρμογές του αυτοματισμού βρίσκονται σε όλες τις σύγχρονες τεχνολογικές εγκαταστάσεις, συσκευές, διατάξεις, μηχανές κλπ. που συναντάμε στη βιομηχανία και τις μεταφορές. Το σχήμα που ακολουθεί απεικονίζει νοηματικά το ρόλο μιας διάταξης αυτοματισμού, που ελέγχει το βαρύ και ογκώδες πηδάλιο ενός πλοίου. Η διάταξη επιβάλλει στο πηδάλιο την επιθυμητή συμπεριφορά, δηλαδή εξασφαλίζει ότι η κίνησή του, ακολουθεί πιστά τη θέση του (κατά πολύ ελαφρύτερου και εύχρηστου) τροχού πηδαλιούχησης.

- Αυτοματισμός πηδαλίου

Μια άλλη διάταξη αυτοματισμού καθοδηγεί τις κινήσεις της μηχανής συσκευασίας που απεικονίζεται στο πιο κάτω σχήμα, έτσι ώστε τα δοχεία να συσκευάζονται στην επιθυμητή μορφή, δηλαδή τακτοποιημένα σε κιβώτια.

- Αυτόματη μηχανή συσκευασίας

Οι εφαρμογές του αυτοματισμού δεν περιορίζονται στον τεχνικό τομέα. Θέμα του αυτοματισμού είναι, για παράδειγμα, και η διερεύνηση του τρόπου με τον οποίο οι αγορές διαμορφώνουν τις τιμές των αγαθών. Δηλαδή, ο αυτοματισμός χρησιμοποιείται για την ανάλυση και μελέτη των φαινομένων της οικονομίας. Ακόμη, ο αυτοματισμός εξετάζει την επίδραση των διαφόρων παραγόντων στην ανάπτυξη των έμβιων πληθυσμών και τους νόμους που διέπουν την εξέλιξη των οικοσυστημάτων.

Οι πιο εξελιγμένες μορφές "αυτόματης" λειτουργίας βρίσκονται, φυσικά, στους ζωντανούς οργανισμούς. Το ανθρώπινο σώμα, για παράδειγμα, περιλαμβάνει πολυάριθμες "αυτόματες" διαδικασίες που εξασφαλίζουν την ομαλή λειτουργία του οργανισμού. Ενδεικτικά:

- όταν κάποιο όργανο του σώματος χρειασθεί οξυγόνο, π.χ. όταν καταβάλλουμε μυϊκή προσπάθεια, η καρδιά πυκνώνει τις σφίξεις και έτσι επιταχύνει την κυκλοφορία του αίματος, που μεταφέρει το οξυγόνο σε όλο το σώμα
- όταν η ένταση του φωτός αυξάνεται, η κόρη συστέλλεται και περιορίζει την έκθεση του εσωτερικού οφθαλμού στο φως, προστατεύοντας έτσι την ευαίσθητη ωχρή κηλίδα από κορεσμό ή βλάβη

- όταν η θερμοκρασία του οργανισμού ανέβει, οι αδένες του δέρματος εκκρίνουν τον ιδρώτα με μεγάλη περιεκτικότητα σε νερό, που εξατμίζεται και αφαιρεί θερμότητα από την επιφάνεια του σώματος.

1.2 Αυτόνομη Πλοήγηση Ρομποτικών Συστημάτων

Ένα ρομπότ μπορεί να δράσει αυτόνομα σε ένα οποιοδήποτε περιβάλλον και είναι σε θέση να αντιλαμβάνεται, να σκέφτεται και να ενεργεί. Τα ρομπότ μπορούν να χρησιμοποιηθούν ώστε να κάνουν εργασίες οι οποίες είτε είναι δύσκολες ή επικίνδυνες για να γίνουν απευθείας από έναν άνθρωπο. Ωστόσο είναι κατάλληλα εργαλεία για την διερεύνηση προβλημάτων τεχνητής νοημοσύνης που περιλαμβάνουν την αποφυγή εμποδίων ή την σχεδίαση μονοπατιού. Ένα από τα χαρακτηριστικά των ρομποτικών συστημάτων είναι η πολυπλοκότητα των περιβαλλόντων μέσα στα οποία κινούνται. Η πλοήγηση είναι μια ουσιώδη ικανότητα ενός αυτόνομου κινούμενου ρομποτικού συστήματος. Του δίνει την δυνατότητα να παραμένει σε λειτουργία αποφεύγοντας τη σύγκρουσή του με εμπόδια, ενώ του επιτρέπει επίσης να φθάσει σε συγκεκριμένες περιοχές ενός άγνωστου περιβάλλοντος που σχετίζονται με κάποιο συγκεκριμένο έργο που πρέπει να φέρει εις πέρας. Πρακτικά το ρομπότ δεν μπορεί να ανακαλύψει άμεσα ένα μονοπάτι από κάποιο αρχικό σημείο προς ένα προορισμό. Για το λόγο αυτό θα πρέπει να χρησιμοποιηθούν τεχνικές εύρεσης μονοπατιού που συνεπάγονται τη μετάβαση από μια αρχική θέση σε κάποιο προορισμό ενώ ταυτόχρονα ελαχιστοποιούν κάποιο κόστος. Η πλοήγηση σχετίζεται με τρία τμήματα:

1. Την εύρεση του ρομπότ στο χώρο (localization)
2. Τη χαρτογράφηση του χώρου (mapping)
3. Το σχεδιασμό του μονοπατιού (path planning)

Ο σχεδιασμός του μονοπατιού αποτελεί το σημαντικότερο βήμα της διαδικασίας της πλοήγησης καθώς με τη διαδικασία αυτή το ρομπότ βρίσκει το βέλτιστο μονοπάτι που πρέπει να ακολουθήσει προκειμένου να φθάσει στο στόχο αποφεύγοντας εμπόδια, επικίνδυνες περιοχές κτλ. στην πορεία του. Αρκετές προσεγγίσεις έχουν προταθεί για το πρόβλημα της σχεδίασης της κίνησης ενός ρομποτικού συστήματος μέσα σε ένα περιβάλλον. Αυτές οι προτάσεις περιλαμβάνουν αλγορίθμους είτε off-line, δηλαδή που παράγουν εκ των προτέρων ένα μονοπάτι για ένα ήδη γνωστό στατικό περιβάλλον, είτε απευθείας, on-line, έχοντας την δυνατότητα εύρεσης του καινούργιου μονοπατιού εξαιτίας κάποιας αλλαγής που συμβαίνει στο περιβάλλον.

1.3 Ιστορία ρομποτικών συστημάτων

Από τα πρώτα ρομπότ που αναφέρονται στη λογοτεχνία είναι ο Τάλως από την ελληνική μυθολογία και οι 20 τρίποδες λέβητες του θεού Ηφαίστου θεωρούμενοι «θαύμα ήδεσθε», αλλά και άλλες περιπτώσεις.

Με την ανάπτυξη και μελέτη των ρομπότ ασχολείται η ρομποτική, επιστήμη που αποτελεί συνδυασμό πολλών κλάδων άλλων επιστημών, κυρίως δε της πληροφορικής, της ηλεκτρονικής και της μηχανολογίας.

1.4 Εφαρμογές Ρομποτικών Συστημάτων

Οι εφαρμογές της ρομποτικής διαφέρουν μεταξύ ιατρικής, και κυρίως της χειρουργικής, της εξερεύνησης του διαστήματος αλλά και τις πολυάριθμες άλλες καθημερινές εργασίες που είναι επαναλαμβανόμενες, ιδιαιτέρως πολύπλοκες, βαρετές, βρώμικες ή επικίνδυνες για να κάνουν οι άνθρωποι.

- Τα ρομπότ στο Διάστημα
- Τα ρομπότ στην Ιατρική
- Τα ρομπότ στη καθημερινή ζωή και εργασία
- Τα ρομπότ στο Διάστημα
- Η ρομποτική στη σύγχρονη εκπαίδευση(LAMS)
- Τα ρομπότ στη ταινίες επιστημονική φαντασίας.

1.5 Επίπεδα αυτονομίας Ρομποτικών Συστημάτων

1. Απευθείας έλεγχος χρησιμοποιείται για ρομπότ που τηλεκατευθύνονται, στα οποία ο χρήστης έχει πλήρη έλεγχο.
2. Υποβοηθούμενη λειτουργία όπου ο χρήστης επιλέγει τους στόχους μεσαίου και υψηλού επιπέδου, τους οποίους το ρομπότ πρέπει να φέρει εις πέρας.
3. Ένα αυτόνομο ρομπότ μπορεί να μη χρειάζεται την επέμβαση του ανθρώπου για αρκετό καιρό. Για να υλοποιηθεί ένα αυτόνομο ρομπότ δεν είναι απαραίτητο να υλοποιηθούν και περίπλοκες εργασίες. Για παράδειγμα, τα ρομπότ σε γραμμές παραγωγής είναι αυτόνομα αλλά έχουν μια πολύ συγκεκριμένη λειτουργία.

1.5.1 Επιπρόσθετη Ταξινόμηση Αυτονομίας Ρομποτικών Συστημάτων.

1. Τηλεκατεύθυνση. Ο άνθρωπος ελέγχει κάθε κίνηση. Κάθε αλλαγή μέσω του χειριστηρίου της μηχανής καθορίζεται πλήρως από τον χειριστή.
2. Με επίβλεψη. Ο άνθρωπος ελέγχει μόνο γενικές κινήσεις και αλλαγές στη θέση. Η μηχανή επιλέγει τις κινήσεις από το χειριστήριο της.
3. Αυτονομία σε επίπεδο στόχων. Ο χειριστής καθορίζει τους στόχους και το ρομπότ προσπαθεί να τους πετύχει.
4. Πλήρης αυτονομία. Η μηχανή θα επιλέξει και θα πετύχει τους στόχους της χωρίς καμία παρέμβαση από άνθρωπο.

1.6 Τι είναι η πλοήγηση ρομποτικού Συστήματος

Ένα ρομποτικό σύστημα είναι μια ευφυής μηχανή ικανή να λειτουργεί αυτόνομα σε ένα οποιοδήποτε περιβάλλον και η οποία είναι σε θέση να αντιλαμβάνεται (αντίληψη του περιβάλλοντος), να σκέφτεται (σχεδίαση) και να ενεργεί (κίνηση). Ωστόσο, τα ρομπότ είναι κατάλληλα εργαλεία για την διερεύνηση προβλημάτων τεχνητής νοημοσύνης όπως η αποφυγή εμποδίων (obstacle avoidance), σχεδίαση μονοπατιού (path planning), κλπ. Ένα από τα χαρακτηριστικά των ρομποτικών συστημάτων είναι η πολυπλοκότητα των περιβαλλόντων μέσα στα οποία κινούνται. Ως εκ τούτου, ένα από τα κρισιμότερα προβλήματα για τα ρομπότ είναι η σχεδίαση του μονοπατιού. Η πλοήγηση (navigation) είναι ένα ζωτικής σημασίας συστατικό ενός αυτόνομου ρομποτικού συστήματος που προσπαθεί να το κατευθύνει μέσα σε κάποιο άγνωστο περιβάλλον ενώ αυτό κινείται. Στόχος των συστημάτων πλοήγησης είναι η οδήγηση του ρομπότ σε κάποιο προορισμό μέσα σε ένα γνωστό, άγνωστο, ή μερικά γνωστό περιβάλλον, χωρίς να χαθεί ή να προσκρούσει σε κάποιο εμπόδιο. Πρακτικά, το ρομπότ δεν μπορεί να βρει άμεσα ένα μονοπάτι από κάποιο αρχικό σημείο προς ένα προορισμό. Για το λόγο αυτό θα πρέπει να χρησιμοποιηθούν τεχνικές εύρεσης μονοπατιού που συνεπάγονται τη μετάβαση από μια αφετηρία σε κάποιο προορισμό ενώ ταυτόχρονα ελαχιστοποιούν κάποιο κόστος, όπως για παράδειγμα το χρόνο που δαπανάται για τη μετάβαση στο προορισμό του. Όταν ένα ρομπότ αρχίζει να κινείται ακολουθώντας ένα ήδη σχεδιασμένο μονοπάτι υπάρχει πιθανότητα να συναντήσει κάποιο εμπόδιο, τότε θα πρέπει να αποφύγει το συγκεκριμένο εμπόδιο και να σχεδιάσει ένα καινούριο μονοπάτι. Με το τρόπο αυτό επιτυγχάνεται η εργασία της πλοήγησης. Η εργασία της πλοήγησης περιλαμβάνει συνήθως τη σχεδίαση μονοπατιού (path planning) και τη σχεδίαση πορείας (trajectory planning). Η σχεδίαση μονοπατιού είναι η εύρεση ενός μονοπατιού απαλλαγμένου από εμπόδια σε ένα περιβάλλον με εμπόδια και η βελτιστοποίηση του σύμφωνα με κάποια κριτήρια.

Η σχεδίαση πορείας είναι ο προγραμματισμός των κινήσεων ενός ρομπότ 0κατά μήκος του σχεδιασμένου μονοπατιού. Αρκετές προσεγγίσεις έχουν προταθεί για το πρόβλημα σχεδίασης της κίνησης ενός ρομποτικού συστήματος μέσα σε ένα περιβάλλον. Ένας αλγόριθμος ονομάζεται εξομοιούμενος (off-line) εάν παράγει εκ των προτέρων ένα μονοπάτι για ένα ήδη γνωστό στατικό περιβάλλον. Αντίθετα ονομάζεται απευθείας (on-line) εάν έχει τη δυνατότητα εύρεσης ενός καινούριου μονοπατιού εξαιτίας κάποιας αλλαγής του συμβαίνει στο περιβάλλον του. Τα συστήματα που ελέγχουν τη πλοήγηση ενός ρομποτικού συστήματος παρακινούνται συνήθως από θεωρίες της ψυχολογίας οι οποίες εξηγούν με ποιο τρόπο τα έμβια όντα μαθαίνουν διάφορες συμπεριφορές.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο Εφαρμογών Πλοήγησης Ρομποτικών Συστημάτων

2.1 Οι εξέλιξη των μεθόδων.

Οι προσεγγιστικές μέθοδοι έχουν πάνω από 100 χρόνια ζωής και στη πορεία της ιστορικής εξέλιξης έχουν αλλάξει ριζικά μορφή και πεδία εφαρμογής. Στην αρχική τους μορφή είχαν μόνο θεωρητικό ενδιαφέρον, έπειτα γίνονταν εκτιμίσεις πάνω σε πραγματικά προβλήματα μετά την ανακαλυψη των ηλεκτρονικών υπολογιστών μπορούσαν να γίνουν ακόμα και προσομοιώσεις φυσικών συστημάτων. Τα τελευταία χρόνια τα πεδία εφαρμογής καλύπτουν πολούς επιστημονικούς κλάδους και αντιμετωπίζονται ως σύνθετα τα θέματα που ήταν αδύνατα να μελετηθούν μέχρι τώρα με τις γνωστές μεθόδους.

2.2 Δυναμικός Προγραμματισμός

Με τον όρο δυναμικός προγραμματισμός (ΔΠ) αναφερόμαστε σε μία συλλογή αλγορίθμων που μπορούν να χρησιμοποιηθούν για να υπολογίσουμε βέλτιστες πολιτικές δοθέντος του μοντέλου του περιβάλλοντος. Οι κλασσικοί αλγόριθμοι ΔΠ είναι ένα περιορισμένο εργαλείο στην ενισχυτική μάθηση τόσο εξαιτίας της παραδοχής ενός τέλειου μοντέλου και εξαιτίας του υψηλού υπολογιστικού κόστους, αλλά αποτελούν τη θεωρητική βάση των μεθόδων ενισχυτικής μάθησης. Η κύρια ιδέα στις μεθόδους ΔΠ είναι η χρήση των συναρτήσεων αξίας για την αναζήτηση καλών πολιτικών. Στη συνέχεια παρουσιάζονται δύο από τους πιο γνωστούς αλγόριθμους ΔΠ ο αλγόριθμος επανάληψης ως προς τη πολιτική (policy iteration) και ο αλγόριθμος επανάληψης ως προς την αξία (value iteration).

2.3 Μοντελοποίηση Προβλημάτων Πλοήγησης Ρομποτικού Συστήματος

Ρομποτικός πράκτορας ονομάζεται οποιαδήποτε ευφυής μηχανική συσκευή που μπορεί να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες.Το πρόβλημα της μοντελοποίησης πλοήγησης ρομποτικού συστήματος είναι πως ο πράκτορας μπορεί να μάθει μια συμπεριφορά μέσω αλληλεπίδρασης στο περιβάλλον για την επίτευξη του στόχου. Ο πράκτορας αλληλοεπιδρά συνεχώς με το περιβάλλον. Στόχος είναι η μεγιστοποίηση της συνολικής απόκρισης του περιβάλλοντος.

Ο πράκτορας και το περιβάλλον αλληλοεπιδρούν σε κάθε χρονική στιγμή t με τον πράκτορα να λαμβάνει αναπαράσταση της κατάστασης του περιβάλλοντος st όπου s είναι το σύνολο των καταστάσεων. Με βάση τη τρέχουσα κατάσταση επιλέγει μία ενέργεια A(s) όπου A(s) είναι το

σύνολο των διαθέσιμων ενεργειών στην κατάσταση s(t) . Την επόμενη χρονική στιγμή (t+1) ο πράκτορας λαμβάνει από το περιβάλλον ένα ερέθισμα r(t+1) ως συνέπεια της ενέργειας του και μεταβαίνει σε μία καινούρια κατάσταση s(t+1).

Σε κάθε χρονική στιγμή ο πράκτορας απεικονίζει τις καταστάσεις σε πιθανότητες επιλογής δυνατών ενεργειών . Αυτή η απεικόνιση ονομάζεται πολιτική του πράκτορα και συμβολίζεται Pt είναι η πιθανότητα του πράκτορα την χρονική στιγμή t να επιλέξει την ενέργεια at=a η οποία να βρίσκεται στην κατάσταση st=s. Οι μέθοδοι ενισχυτικής μάθησης προσδιορίζουν τον τρόπο με τον οποίο ο πράκτορας αλλάζει πολιτική του ως αποτέλεσμα της εμπειρίας του. Ο στόχος του πράκτορα είναι να μεγιστοποιήσει τη συνολική ανταμοιβή που λαμβάνει μακροπρόθεσμα.

Το συγκεκριμένο πλαίσιο είναι αφηρημένο και ευέλικτο, έτσι μπορεί να εφαρμοστεί σε πολλά διαφορετικά προβλήματα και με πολλούς διαφορετικούς τρόπους. Για παράδειγμα, οι χρονικές στιγμές δεν χρειάζεται να αναφέρονται σε πραγματικό χρόνο αλλά μπορεί να αναφέρονται σε διαδοχικά στάδια λήψης απόφασης και δράσης.

Αυτό το οποίο προτείνει το πλαίσιο της ενισχυτικής μάθησης είναι ότι οποιοδήποτε πρόβλημα μάθησης συμπεριφοράς, οδηγούμενο από κάποιο συγκεκριμένο στόχο, μπορεί να αναπαρασταθεί σε τρία σήματα τα οποία ανταλλάσσονται μεταξύ του πράκτορα και του περιβάλλοντος, ένα σήμα για να αναπαρασταθούν οι επιλογές του πράκτορα (ενέργειες), ένα σήμα για να αναπαρασταθεί η πληροφορία στην οποία βασίζονται οι επιλογές των ενεργειών του πράκτορα (καταστάσεις), και ένα σήμα για να προσδιοριστεί ο στόχος του πράκτορα (ανταμοιβή).

Στη γενική περίπτωση του προβλήματος για τη διεύρυνση των γνώσεων μας, οι ενέργειες που εκτελεί ο πράκτορας δεν καθορίζουν μόνο την άμεση ανταμοιβή που λαμβάνει, αλλά επίσης και την επόμενη κατάσταση του περιβάλλοντος. Τέτοια περιβάλλοντα μπορούν να εκληφθούν ως δίκτυα, όπου ο πράκτορας πρέπει να λαμβάνει υπόψη τόσο την επόμενη κατάσταση όσο και την άμεση ανταμοιβή, για να αποφασίσει ποια ενέργεια πρέπει να επιλέξει. Για το λόγο αυτό, το περιβάλλον των προβλημάτων ενισχυτικής μάθησης μοντελοποιείται ως Μαρκοβιανή Διαδικασία Απόφασης (Markov Decision Process).

2.4 Μαρκοβιανή Διαδικασία

Απόφασης περιγράφεται από μία τετράδα (S, A,T, Pr(S)) όπου, S το πεπερασμένο σύνολο όλων των καταστάσεων A το πεπερασμένο σύνολο των ενεργειών T: S x A -> Pr(S) είναι η συνάρτηση μετάβασης (transition function), όπου Pr(S)είναι μία κατανομή πιθανοτήτων στο σύνολο καταστάσεων S, η οποία δεδομένης μίας κατάστασης και μίας ενέργειας μας επιστρέφει τις

πιθανότητες μετάβασης σε κάθε πιθανή επόμενη κατάσταση. Γράφουμε TgS, για την πιθανότητα μετάβασης από την κατάσταση S στην κατάσταση S' εκτελώντας την ενέργεια A.

R: S x A x S -> Είναι η συνάρτηση ανταμοιβής, η οποία καθορίζει την επόμενη αναμενόμενη ανταμοιβή ως συνάρτηση της τρέχουσας κατάστασης και ενέργειας, καθώς και της επόμενης κατάστασης. Γράφουμε Rass', για την αναμενόμενη ανταμοιβή που θα πάρουμε αν στην κατάσταση επιλέξουμε S την ενέργεια A και μεταβούμε στην κατάσταση S'

Για να είναι το μοντέλο Μαρκοβιανό θα πρέπει να ισχύει η ιδιότητα Markov, η οποία ορίζει ότι η απόκριση του περιβάλλοντος τη χρονική στιγμή t+1 εξαρτάται μόνο από την αναπαράσταση της κατάστασης και της ενέργειας τη χρονική στιγμή t και όχι από όλες τις προηγούμενες χρονικές στιγμές, δηλαδή

$$\boldsymbol{P_r\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} = P_r\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0\}}$$

2.6 Συναρτήσεις Αξίας

Όπως προαναφέρθηκε, στόχος του πράκτορα είναι να μεγιστοποιήσει την συνολική ανταμοιβή που λαμβάνει μακροπρόθεσμα. Γενικά, ο πράκτορας θέλει να μεγιστοποιήσει την αναμενόμενη απολαβή, όπου η απολαβή , $\boldsymbol{R_t}$, ορίζεται ως μία ειδική συνάρτηση της ακολουθίας των ανταμοιβών. Αν η σειρά των ανταμοιβών που λαμβάνονται μετά τη χρονική στιγμή t συμβολίζεται ως

$$\boldsymbol{D_t = r_{t+1} + r_{t+2} + \cdots + r_T}$$

Όπου T είναι το τελικό χρονικό βήμα. Αυτή η προσέγγιση έχει νόημα σε εφαρμογές που υπάρχει η έννοια του τελικού χρονικού βήματος, δηλαδή εργασίες (tasks) που εκτελούνται σε επεισόδια. Η ακολουθία των ενεργειών που εκτελούνται από κάποιον πράκτορα για να φθάσει από μια αρχική κατάσταση σε μια τελική είναι ένα επεισόδιο. Κάθε επεισόδιο τερματίζει σε μια ειδική κατάσταση που ονομάζεται τερματική κατάσταση όπου όλες οι ενέργειες οδηγούν στην ίδια κατάσταση λαμβάνοντας μηδενική ανταμοιβή. Στη συνέχεια, ο πράκτορας μεταβαίνει στην αρχική κατάσταση ή σε κάποια από τις αρχικές καταστάσεις με την ίδια πιθανότητα να ξεκινά ένα καινούριο επεισόδιο. Σε προβλήματα όπου ο πράκτορας παίρνει κάποια ανταμοιβή μόνο ως στόχος. Μερικές φορές είναι αναγκαίο να διακρίνουμε το σύνολο όλων των μη τερματικών καταστάσεων, συμβολίζοντας αυτό ως $\boldsymbol{S^+}$.

Αντίθετα, σε πολλές περιπτώσεις η αλληλεπίδραση του πράκτορα με το περιβάλλον δεν διακόπτεται σε αναγνωρίσιμα επεισόδια, αλλά συνεχίζεται χωρίς κάποιο περιορισμό επ'άπειρον. Ονομάζουμε αυτή την εργασία, συνεχόμενη. Ο ορισμός της απολαβής είναι προβληματικός για συνεχόμενες εργασίες διότι το τελικό χρονικό βήμα θα είναι $\boldsymbol{T = \infty}$, και η απολαβή, που είναι αυτό που προσπαθούμε να μεγιστοποιήσουμε μπορεί εύκολα να γίνει ίση με το άπειρο από μόνη της. Για τον λόγο αυτό, συνήθως χρησιμοποιούμε έναν ορισμό για την απολαβή που είναι ελαφρώς πιο περίπλοκος εννοιολογικά αλλά απλούστερος μαθηματικά. Η επιπλέον ιδέα που εισάγουμε ελαφρός πιο περίπλοκος εννοιολογικά αλλά απλούστερος μαθηματικά. Η επιπλέον ιδέα που εισάγουμε είναι αυτή της έκπτωσης . Σύμφωνα με αυτή τη προσέγγιση, ο πράκτοράς προσπαθεί να επιλέξει ενέργειες έτσι ώστε το άθροισμα των υπό έκπτωση ανταμοιβών που λαμβάνει να μεγιστοποιείται. Συγκεκριμένα, επιλέγει την ενέργεια $\boldsymbol{a_t}$ έτσι ώστε να μεγιστοποιήσει την αναμενόμενη υπό έκπτωση απολαβή.

$$\boldsymbol{D_t = r_{t+1} + \gamma * r_{t+2} + \gamma^2 * r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^2 * r_{t+k+1}}$$

Όπου $\boldsymbol{0 \leq \gamma \leq 1}$ είναι ο ρυθμός έκπτωσης

Ο ρυθμός έκπτωσης και ορίζει την παρούσα ενέργεια των μελλοντικών ανταμοιβών μία ανταμοιβή που λαμβάνεται για k χρονικά βήματα στο μέλλον αξιώνει μόνο $\boldsymbol{\gamma^{k-1}}$φορές σε σχέση με την αξία που θα έχει ληφθεί άμεσα. Εάν γ<1, το άπειρο άθροισμα έχει μία πεπερασμένη ενέργεια όσο η ακολουθία των ανταμοιβών rk είναι οριοθετημένη. Εάν γ=0, ο πράκτορας ενδιαφέρεται μόνο αν μεγιστοποιήσει τις άμεσες ανταμοιβές, στόχος είναι να μάθει πως θα επιλέξει την ενέργεια at έτσι ώστε να μεγιστοποιήσει τις άμεσες ανταμοιβές, στόχος του είναι να μάθει πως θα επιλέξει την ενέργεια at έτσι ώστε να μεγιστοποιήσει rt+1 . Καθώς το γ προσεγγίζει το 1 ,οι μελλοντικές ανταμοιβές λαμβάνονται περισσότερο υπόψη όπου σαν αποτέλεσμα ο πράκτορας γίνεται περισσότερο προνοητικός.

Η αξία μίας κατάστασης s υπό μία πολιτική π, συμβολίζεται $\boldsymbol{V^{\pi}(s)}$ και είναι η αναμενόμενη μέση απολαβή αν ο πράκτορας ξεκινήσει από τη κατάσταση s και ακολουθήσει την πολιτική π. Για Μαρκοβιανές Διαδικασίες Απόφασης (ΜΔΑ) μπορούμε να ορίσουμε τη συνάρτηση Vπ(s) ως:

$$\boldsymbol{V^{\pi}(s) = E_{\pi}\{D_t | s_t = s,\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k * r_{t+k+1} | s_t = s,\right\}}$$

Όπου Επ υποδηλώνει την μέση τιμή δοθέντος ότι ο πράκτορας ακολουθεί τη πολιτική π. Η αξία τερματικής κατάστασης είναι πάντα μηδέν. Η συνάρτηση $\boldsymbol{V^{\pi}(s)}$ ονομάζεται συνάρτηση αξίας κατάστασης.

Παρόμοια ορίζουμε την Qπ(s,a) για τη λήψη μία ενέργειας α στη κατάσταση s υπό μια πολιτική π ως την αναμενόμενη απολαβή ξεκινώντας από τη κατάσταση s, επιλέγοντας την ενέργεια α και ακολουθώντας στη συνέχεια τη πολιτική π

$$\boldsymbol{Q^{\pi}(s,a) = E_{\pi}\{D_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k * r_{t+k+1} | s_t = s, a_t = a\right\}}$$

Ονομάζουμε τη συνάρτηση Qπ (s,a) ως συνάρτηση αξίας κατάστασης-ενέργειας

Οι συναρτήσεις $\boldsymbol{V^{\pi}(s)}$ **και** $\boldsymbol{Q^{\pi}(s,a)}$ μπορούν να υπολογιστούν εμπειρικά. Για παράδειγμα, αν ένας πράκτορας ακολουθεί μία πολιτική π και διατηρεί έναν μέσο όρο, για κάθε κατάσταση που συναντά, πραγματικών απολαβών που αντιστοιχούν σε αυτή τη κατάσταση, τότε ο μέσος όρος θα συγκλίνεις στην αξία της α κατάστασης $\boldsymbol{V^{\pi}(s)}$,καθώς ο αριθμός επισκέψεων στη κατάσταση τείνει στο άπειρο.. εάν επιπλέον διατηρούνται οι μέσοι όροι για κάθε ενέργεια που εκτελείται σε κάθε κατάσταση, τότε αυτοί ο μέσοι όροι θα συγκλίνουν στις αξίες της κατάστασης ενέργειας $\boldsymbol{Q^{\pi}(s,a)}$.

Ονομάζουμε τις παραπάνω μεθόδους Monte Carlo μεθόδους. Εάν το πλήθος των καταστάσεων είναι μεγάλο και πρακτικό να κρατάμε τους μέσου όρους για κάθε κατάσταση χωριστά. Αντίθετα ο πράκτορας πρέπει να διατηρεί $\boldsymbol{V^{\pi}(s)}$ και $\boldsymbol{Q^{\pi}(s,a)}$ ως παραμετροποιημένες συναρτήσεις και να προσαρμόζει τις παραμέτρους έτσι ώστε να ταιριάζουν καλύτερα στις παρατηρούμενες απολαβές.

Μια θεμελιώδης ιδιότητα των συναρτήσεων αξίας είναι ότι ικανοποιούν συγκεκριμένες επαναληπτικές σχέσεις. Για μία οποιαδήποτε πολιτική π και κατάσταση s διατηρείται η ακόλουθη σχέση ανάμεσα στην αξία s και στην αξία της πιθανής διαδοχικής κατάστασης

$$\boldsymbol{V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k * r_{t+k+1} | s_t = s,\right\}}$$

2.7 Μέθοδοι Monte Carlo

Σε αντίθεση με τις μεθόδους Δυναμικού Προγραμματισμού, οι μέθοδοι Monte Carlo (MC) δεν απαιτούν τη γνώση του μοντέλου του περιβάλλοντος. Βασίζονται μόνο στη συλλεγόμενη εμπειρία τους (ακολουθίες καταστάσεων, ενεργειών και ανταμοιβών), που αποκτήθηκε από την απευθείας (on-line) ή εξομοιωμένη (off-line) αλληλεπίδρασή τους με το περιβάλλον. Η μάθηση με απευθείας αλληλεπίδραση δεν απαιτεί καμία προηγούμενη γνώση των δυναμικών του περιβάλλοντος και

μπορεί να επιτύχει βέλτιστη συμπεριφορά. Παρόλο που η μάθηση με εξομοιωμένη εμπειρία απαιτεί τη γνώση του μοντέλου, δεν χρειάζεται τη γνώση του πλήρους μοντέλου αλλά αρκεί μια προσέγγιση αυτού.

Οι μέθοδοι MC είναι τρόποι επίλυσης προβλημάτων ενισχυτικής μάθησης βασιζόμενοι στους μέσους όρους των απολαβών του πράκτορα. Για να εξασφαλίσουμε ότι οι απολαβές είναι καλά ορισμένες, ορίζουμε τις μεθόδους MC μόνο για εργασίες με επεισόδια. Μόνο μετά την ολοκλήρωση ενός επεισοδίου μεταβάλλονται οι εκτιμήσεις των αξιών και οι πολιτικές. Για την εκτίμηση μίας πολιτικής, οι μέθοδοι MC χρησιμοποιούν την συνάρτηση αξίας κατάστασης, η οποία εκφράζει την αναμενόμενη απολαβή πού λαμβάνει ο πράκτορας ξεκινώντας από αυτή τη κατάσταση. Ο πιο προφανής τρόπος για τον υπολογισμό της από την εμπειρία είναι να υπολογίσουμε το μέσο όρο των απολαβών πού παρατηρήθηκαν μετά από κάθε επίσκεψη στη συγκεκριμένη κατάσταση. Όσες περισσότερες απολαβές παρατηρούνται τόσο περισσότερο ο μέσος όρος συγκλίνει στην αναμενόμενη αξία. Η συγκεκριμένη ιδέα χαρακτηρίζει όλες τις μεθόδους MC. Κάθε εμφάνιση της κατάστασης s σε ένα επεισόδιο ονομάζεται επίσκεψη (visit) της s.

Η μέθοδος κάθε επίσκεψης MC (every-visit MC method), εκτιμά τη V*(s) ως το μέσο όρο των απολαβών πού λαμβάνονται μετά από κάθε επίσκεψη της $ σε κάθε επεισόδιο. Η πρώτη μας επίσκεψη στη κατάσταση 5 σε ένα επεισόδιο, ονομάζεται πρώτη επίσκεψη (first visit) της s. Η μέθοδος πρώτης επίσκεψης MC (first-visit MC method) για τον υπολογισμό του μέσου όρου των απολαβών λαμβάνει υπόψη μόνο τις απολαβές που λαμβάνονται μετά τη πρώτη επίσκεψη στη κατάσταση

Τόσο η μέθοδος κάθε επίσκεψης MC όσο και η μέθοδος πρώτης επίσκεψης MC συγκλίνουν στη !**($) καθώς ο αριθμός των επισκέψεων (ή των πρώτων επισκέψεων) στην s πλησιάζει στο άπειρο. Ο επόμενος Αλγόριθμος περιγράφει τη μέθοδο πρώτης επίσκεψης MC.

Αρχικοποίηση: Πολιτική προς εκτίμηση, τυχαία συνάρτηση αξίας κατάστασης

(a) Παραγωγή ενός επεισοδίου χρησιμοποιώντας τη πολιτική π

(b) Για κάθε κατάσταση s πού εμφανίζεται στο επεισόδιο:

D «— απολαβή πού λαμβάνεται μετά τη πρώτη εμφάνιση της s

Προσθήκη της D στη λίστα

V'(s) «— average(R etu rn $ (s)) να εκτιμήσουμε τις αξίες κατάστασης-ενέργειας παρά τις αξίες κατάστασης. Γνωρίζοντας το μοντέλο, οι αξίες κατάστασης είναι αρκετές για να προσδιορίσουμε μια πολιτική. Αντίθετα, θα πρέπει να εκτιμήσουμε την αξία κάθε ενέργειας έτσι ώστε να μπορέσουμε να ορίσουμε μια πολιτική. Για το λόγο αυτό, ένας από τους σημαντικότερους στόχους των μεθόδων MC είναι ο υπολογισμός της Q*.

Το πρόβλημα εκτίμησης πολιτικής για αξίες ενέργειας είναι ο υπολογισμός της (^ (s, α), δηλαδή της αναμενόμενης απολαβής που λαμβάνουμε ξεκινώντας από τη κατάσταση s. επιλέγοντας την ενέργεια α και έπειτα ακολουθώντας τη πολιτική π. Οι μέθοδοι MC παραμένουν στην ουσία ίδιες με τις μεθόδους που παρουσιάστηκαν για την αξία κατάστασης. Το μόνο πρόβλημα που προκύπτει είναι πώς αρκετά ζεύγη κατάστασης-ενέργειας μπορεί να μην επισκεφτούν ποτέ. Στη περίπτωση που η π είναι μια ντετερμινιστική πολιτική, ακολουθώντας κάποιος τη π θα παρατηρήσει απολαβές μόνο για μια ενέργεια για κάθε κατάσταση. Για να υπολογίσει το μέσο όρο χωρίς απολαβές, οι MC εκτιμήσεις για τις άλλες ενέργειες δε θα βελτιωθούν με την εμπειρία. Αυτό είναι ένα πολύ σημαντικό πρόβλημα διότι ο σκοπός της μάθησης των αξίων κατάστασης-ενέργειας είναι να βοηθήσει στην επιλογή ανάμεσα στις ενέργειες που είναι διαθέσιμες σε κάθε κατάσταση. Δηλαδή, χρειάζεται να υπολογίσουμε τις αξίες όλων των ενεργειών για κάθε κατάσταση. Για να λειτουργήσει η εκτίμηση πολιτικής για αξίες κατάστασης-ενέργειας, θα πρέπει να εξασφαλίσουμε πλήρη εξερεύνηση. Ένας τρόπος για να συμβεί αυτό είναι υποθέτοντας πως το πρώτο βήμα κάθε επεισοδίου ξεκινά από ένα ζεύγος κατάστασης-ενέργειας και κάθε ζεύγος αυτής της μορφής έχει μη μηδενική πιθανότητα για να επιλεχθεί κατά το ξεκίνημα. Αυτό μας εγγυάται πως όλα τα ζεύγη κατάστασης-ενέργειας θα επισκεφθούν άπειρο αριθμό φορών καθώς ο αριθμός των επεισοδίων γίνεται άπειρος. Αυτή η παραδοχή ονομάζεται εξερεύνηση αφετηρίας (exploring starts).Η βελτίωση της πολιτικής (policy improvement) επιτυγχάνεται κάνοντας τη πολιτική άπληστη σε σχέση με τη τρέχουσα συνάρτηση αξίας. Στη συγκεκριμένη περίπτωση είναι διαθέσιμη η συνάρτηση αξίας κατάστασης-ενέργειας, με αποτέλεσμα να μη χρειάζεται το μοντέλο για τη παραγωγή της άπληστης πολιτικής. Για κάθε συνάρτηση αξίας κατάστασης-ενέργειας Q, άπληστη πολιτική είναι αυτή που για κάθε κατάσταση, $ G <S, ντετερμινιστικά επιλέγει την ενέργεια με τη μεγαλύτερη αξία Q:n(s) = argm axQ(*,a).(3.7)a

Είναι φυσιολογική για την εκτίμηση πολιτικής MC η εναλλαγή ανάμεσα στην εκτίμηση και τη βελτίωση, επεισόδιο ανά επεισόδιο. Μετά το τέλος κάθε επεισοδίου, οι παρατηρούμενες απολαβές χρησιμοποιούνται για την εκτίμηση πολιτικής, ενώ αμέσως μετά η πολιτική βελτιώνεται

σε όλες τις καταστάσεις που έχουν επισκεφτεί κατά τη διάρκεια του επεισοδίου. Ο συγκεκριμένο αλγόριθμος (Αλγόριθμος 4) ονομάζεται Monte Carlo ES (MC με εξερεύνηση αφετηρίας).

Στον Monte Carlo ES, όλες οι απολαβές για κάθε ζεύγος κατάστασης-ενέργειας συσσωρεύονται και υπολογίζεται ο μέσος όρος τους, ανεξάρτητα από ποια πολίτικη ήταν σε ισχύ όταν παρατηρήθηκαν. Είναι εύκολο να δειχθεί ότι ο Monte Carlo ES δεν μπορεί να συγκλίνει σε μια μη βέλτιστη πολιτική.

(a) Παραγωγή ενός επεισοδίου χρησιμοποιώντας την πολιτική π και την εξερεύνηση αφετηρίας

(b) Για κάθε ζεύγος s, α πού εμφανίζεται στο επεισόδιο:

D <— απολαβή μετά τη πρώτη εμφάνιση των s, α

Προσθήκη της D στη λίστα

Returns(s, α)

2.7.1 Προσομοίωση Monte Carlo στο MATLAB

Η γλώσσα MATLAB® παρέχει μια ποικιλία μαθηματικών λειτουργιών υψηλού επιπέδου που μπορούμε να χρησιμοποιήσουμε και να δημιουργήσετε ένα μοντέλο προσομοίωσης Monte Carlo και να εκτελέσουμε αυτές τις προσομοιώσεις. Το MATLAB χρησιμοποιείται για τη χρηματοοικονομική μοντελοποίηση, την πρόβλεψη καιρού, την ανάλυση λειτουργιών και πολλές άλλες εφαρμογές.

Στη χρηματοοικονομική μοντελοποίηση, η Monte Carlo Simulation ενημερώνει την τιμή, το ρυθμό και τις οικονομικές προβλέψεις. διαχείριση κινδύνου; και προσομοιώσεις ακραίων καταστάσεων. Το Financial Toolbox ™ παρέχει εργαλεία στοχαστικής διαφορικής εξίσωσης για την κατασκευή και την αξιολόγηση στοχαστικών μοντέλων. Το Risk Management Toolbox ™ διευκολύνει την προσομοίωση πιστώσεων, συμπεριλαμβανομένης της εφαρμογής μοντέλων copula.

Για τον μεγαλύτερο έλεγχο της παραγωγής εισροών, το Statistics and Machine Learning Toolbox ™ παρέχει μια μεγάλη ποικιλία κατανομών πιθανοτήτων που μπορείτε να χρησιμοποιήσετε για να δημιουργήσετε τόσο συνεχείς όσο και διακριτές εισόδους.

2.7.2 Προσομοίωση Monte Carlo στο Simulink

Μπορούμε να μοντελοποιήσουμε και να προσομοιώσουμε συστήματα πολλαπλών τομέων στο Simulink® για να προσομοιώσουμε ελεγκτές, μοτέρ, κέρδη και άλλα εξαρτήματα. Ο σχεδιασμός

και η δοκιμή αυτών των σύνθετων συστημάτων περιλαμβάνει πολλαπλά στάδια, συμπεριλαμβανομένου του προσδιορισμού των παραμέτρων που έχουν τις μεγαλύτερες επιπτώσεις στις απαιτήσεις και τη συμπεριφορά, την καταγραφή και ανάλυση των δεδομένων προσομοίωσης και την επαλήθευση του σχεδιασμού του συστήματος.

Οι προσομοιώσεις του Monte Carlo μας βοηθούν να αποκτήσουμε εμπιστοσύνη στο σχεδιασμό μας, επιτρέποντάς μας να εκτελέσουμε σαρώσεις παραμέτρων, να εξερευνήσουμε τον χώρο σχεδιασμού σας, να δοκιμάσουμε πολλά σενάρια και να χρησιμοποιήσουμε τα αποτελέσματα αυτών των προσομοιώσεων για να καθοδηγήσουμε τη διαδικασία σχεδιασμού μέσω στατιστικής ανάλυσης. Το Simulink Design Optimization ™ παρέχει διαδραστικά εργαλεία για να πραγματοποιήσουμε αυτήν την ανάλυση ευαισθησίας και να επηρεάσουμε το σχεδιασμό μοντέλου Simulink.

2.7.3 Τρέχοντας παράλληλα προσομοιώσεις Monte Carlo

Για να βελτιώσουμε την απόδοση των προσομοιώσεων σας Monte Carlo, μπορούμε να τις διανείμουμε στους υπολογιστές ώστε να εκτελούνται παράλληλα σε πολλούς πυρήνες χρησιμοποιώντας το Parallel Computing Toolbox ™ και το MATLAB Distributed Computing Server ™.

Θεωρητικό Μοντέλο

Νοήθηκε ένα μοντέλο το οποίο θα αποτελείται από τρεις κύριες καταστάσεις. Ανάλογα με την εκάστοτε περίσταση, θα εναλλάσσεται ανάμεσα σε αυτές τις καταστάσεις για να ακολουθεί τον στόχο του. Η κατάληξη σε αυτές τις καταστάσεις προήλθε από την απλούστατη σκέψη, του τι πιθανά σενάρια μπορούμε να συναντήσουμε. Έτσι λοιπόν τα πιο πιθανά σενάρια μπορεί να είναι:

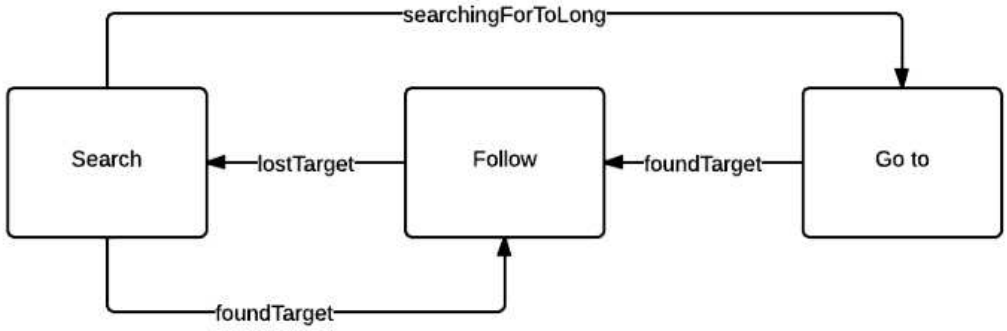
- Ο στόχος να είναι εντός οπτικής εμβέλειας.
- Ο στόχος να είναι εκτός οπτικής εμβέλειας.

2.7.4 Περιγραφή Συστήματος

Ο στόχος να είναι εκτός οπτικής εμβέλειας, για περισσότερο χρόνο από τον αναμενόμενο. Έτσι μπορούν να εξαχθούν οι τρεις καταστάσεις οι οποίες είναι

- Ακολουθώ (Following)
- Ψάχνω (Searching)
- Πηγαίνω (Going to)

Πιο επίσημα, στο σχήμα φαίνονται οι τρεις καταστάσεις και με ποιο τρόπο συνδέονται. Δηλαδή στην κατάσταση όπου το ρομπότ ακολουθεί τον στόχο, εάν γίνει κάτι με αποτέλεσμα την απώλεια του στόχου, τότε γίνεται μετάβαση στην κατάσταση της αναζήτησης. Στην κατάσταση της αναζήτησης, εάν ο στόχος είναι ξανά εντός εμβέλειας, τότε μεταβαίνουμε πάλι στην κατάσταση που το ρομπότ ακολουθεί το στόχο. Επίσης για να γίνει λίγο πιο έξυπνο, αν η αναζήτηση διαρκέσει περισσότερη από την επιθυμητή ώρα, τότε το ρομπότ αρχίζει να πηγαίνει σε συγκεκριμένα σημεία στον χάρτη με μια συγκεκριμένη σειρά. Και τέλος, στην κατάσταση που το ρομπότ πηγαίνει σε αυτά τα σημεία και κάπου στην πορεία του συναντήσει το στόχο, ξαναγυρνάει στην κατάσταση που ακολουθεί το στόχο τροφοδοσία.



Σχήμα 1

2.7.5 Αλγόριθμος Monte Carlo

Ο αλγόριθμος εντοπισμού θέσης του Monte Carlo(MCL) είναι μια επέκταση του αλγορίθμου εντοπισμού θέσης του Markov. Σύμφωνα με τους Burgard et al είναι μια εκδοχή ενός αλγορίθμου που βασίζεται στη δειγματοληψία. Και μετά από μια προεργασία στη εκ των υστέρων δειγματοληψία (sampling/importance re-sampling) ή αλλιώς. Ο αλγόριθμος είναι γνωστός και ως "bootstrap filter", "Monte-Carlo filter", "Condensation algorithm", ή "survival of the fittest algorithm". Γενικά όλοι αυτοί οι αλγόριθμοι είναι τα λεγόμενα φίλτρα σωματιδίων (particle filters). Περισσότερες πληροφορίες μπορούν να βρεθούν στην παρακάτω αναφορά Αλγόριθμοι εντοπισμού θέσης Σε αυτό το κεφάλαιο θα παρουσιαστεί η περιγραφή ενός αλγορίθμου που χρησιμοποιείται για τον εντοπισμό της θέσης του ρομπότ στο χώρο. Γενικά έχουν διεξαχθεί εκτενείς έρευνες σχετικά με αυτόν τον τομέα, γιατί το να έχει το ρομπότ γνώση της θέσης του στο χώρο και γενικότερα πληροφοριών για το περιβάλλον του, είναι μεγάλο πλεονέκτημα για οποιαδήποτε εργασία θα πραγματοποιήσει μετά. Ο εντοπισμός της θέσης διακρίνεται σε δύο κύρια προβλήματα. Το ένα είναι η παρακολούθηση της θέσης του κατά τη διάρκεια μιας κίνησης και το

άλλο είναι ο εντοπισμός της θέσης κατά την εκκίνηση του ρομπότ. Οι πρώτες έρευνες που έγιναν βασίστηκαν στο πρώτο πρόβλημα. Μετά το 1990 άρχισαν να εμφανίζονται έρευνες με αλγορίθμους που λύνουν και τα δύο προβλήματα. Η ιδέα για την εκτίμηση της θέσης μέσω πιθανότητας, έχει τις ρίζες της στα φίλτρα Kalman.

2.8 Τι είναι η μέθοδος PRM

Το PRM δημιουργεί ένα αντικείμενο σχεδιασμού διαδρομής χάρτη πορείας για το χάρτη περιβάλλοντος που καθορίζεται στην ιδιότητα Map. Το αντικείμενο χρησιμοποιεί το χάρτη για να δημιουργήσει έναν χάρτη πορείας, ο οποίος είναι ένα γράφημα δικτύου για πιθανές διαδρομές στο χάρτη βάσει ελεύθερων και κατεχομένων χώρων. Μπορείτε να προσαρμόσετε τον αριθμό των κόμβων, NumNodes και την απόσταση σύνδεσης, ConnectionDistance, για να ταιριάξετε με την πολυπλοκότητα του χάρτη και να βρείτε μια διαδρομή χωρίς εμπόδια από την αρχή μέχρι την τελική θέση.

Αφού οριστεί ο χάρτης, ο προγραμματιστής διαδρομής PRM παράγει τον καθορισμένο αριθμό κόμβων σε όλους τους ελεύθερους χώρους του χάρτη. Μια σύνδεση μεταξύ των κόμβων γίνεται όταν μια γραμμή μεταξύ δύο κόμβων δεν περιέχει εμπόδια και βρίσκεται εντός της καθορισμένης απόστασης σύνδεσης.

Αφού καθορίσετε μια θέση έναρξης και λήξης, για να βρείτε μια διαδρομή χωρίς εμπόδια χρησιμοποιώντας αυτό το δίκτυο συνδέσεων, χρησιμοποιήστε τη μέθοδο findpath. Αν το findpath δεν βρει μια συνδεδεμένη διαδρομή, επιστρέφει έναν κενό πίνακα. Με την αύξηση του αριθμού των κόμβων ή της απόστασης σύνδεσης, μπορείτε να βελτιώσετε την πιθανότητα εύρεσης μιας συνδεδεμένης διαδρομής, αλλά ο συντονισμός αυτών των ιδιοτήτων είναι απαραίτητος. Για να δείτε τον χάρτη πορείας και την παραγόμενη διαδρομή, χρησιμοποιήστε τις επιλογές οπτικοποίησης στην εμφάνιση. Εάν αλλάξετε οποιεσδήποτε από τις ιδιότητες PRM, ενημερώστε κλήση, εμφάνιση ή findpath για να αναδημιουργήσετε τον χάρτη πορείας.

2.8.1 Κατασκευή Μεθόδου PRM.

planner = robotics.PRM δημιουργεί έναν κενό χάρτη πορείας με προεπιλεγμένες ιδιότητες. Για να μπορέσουμε να χρησιμοποιήσουμε τον χάρτη πορείας, πρέπει να ορίσουμε ένα αντικείμενο robotics.BinaryOccupancyGrid στην ιδιότητα Map.

planner = robotics.PRM (map) δημιουργεί χάρτη πορείας με χάρτη που έχει οριστεί ως η ιδιότητα Map, όπου ο χάρτης είναι ένα αντικείμενο της κλάσης robotics.BinaryOccupancyGrid.

planner = robotics.PRM (map, numnodes) ορίζει τον μέγιστο αριθμό κόμβων, numnodes, στην ιδιότητα NumNodes.

2.8.2 Σχεδιασμός διαδρομής σε περιβάλλοντα διαφορετικής πολυπλοκότητας

Αυτό το παράδειγμα δείχνει τον τρόπο υπολογισμού μιας διαδρομής χωρίς εμπόδια μεταξύ δύο τοποθεσιών σε ένα συγκεκριμένο χάρτη χρησιμοποιώντας τον προγραμματιστή διαδρομής πιθανών οδικών χαρτών (PRM). Ο προγραμματιστής πορείας PRM κατασκευάζει έναν χάρτη πορείας στον ελεύθερο χώρο ενός συγκεκριμένου χάρτη χρησιμοποιώντας τυχαία δειγματοληπτικούς κόμβους στον ελεύθερο χώρο και συνδέοντας τους μεταξύ τους. Μόλις κατασκευαστεί ο χάρτης πορείας, μπορείτε να αναζητήσετε μια διαδρομή από μια δεδομένη τοποθεσία έναρξης σε μια δεδομένη τελική θέση στο χάρτη.

Σε αυτό το παράδειγμα, ο χάρτης αναπαρίσταται ως ένας πίνακας πλέγματος πληρότητας χρησιμοποιώντας εισαγόμενα δεδομένα. Όταν οι κόμβοι δειγματοληψίας στον ελεύθερο χώρο ενός χάρτη, το PRM χρησιμοποιεί αυτή τη δυαδική αναπαράσταση του δικτύου πληρότητας για να συμπεράνει τον ελεύθερο χώρο. Επιπλέον, το PRM δεν λαμβάνει υπόψη τη διάσταση των ρομπότ ενώ υπολογίζει μια διαδρομή χωρίς εμπόδια σε ένα χάρτη. Ως εκ τούτου, θα πρέπει να διογκώσουμε το χάρτη με την διάσταση του ρομπότ, ώστε να επιτρέψουμε τον υπολογισμό μιας διαδρομής χωρίς εμπόδια που να αντιπροσωπεύει το μέγεθος του ρομπότ και να διασφαλίζει την αποφυγή σύγκρουσης για το πραγματικό ρομπότ. Ορίστε τις θέσεις έναρξης και λήξης στο χάρτη για τον προγραμματιστή διαδρομής PRM για να βρείτε μια διαδρομή χωρίς εμπόδια.

Εισαγωγή χαρτών παραδείγματος για τον προγραμματισμό ενός μονοπατιού

filePath = fullfile (fileparts (τα οποία ('PathPlanningExample')), 'data', 'exampleMaps.mat').

φορτίο (filePath)

Οι χάρτες που εισάγονται είναι: simpleMap, complexMap και ternaryMap. Αυτό αναζητά μεταβλητές που περιέχουν τη συμβολοσειρά «Χάρτης» στο όνομα της μεταβλητής.

Name	Size	Bytes	Class	Attributes
complexMap	41x52	2132	logical	
simpleMap	26x27	702	logical	
ternaryMap	501x501	2008008	double	

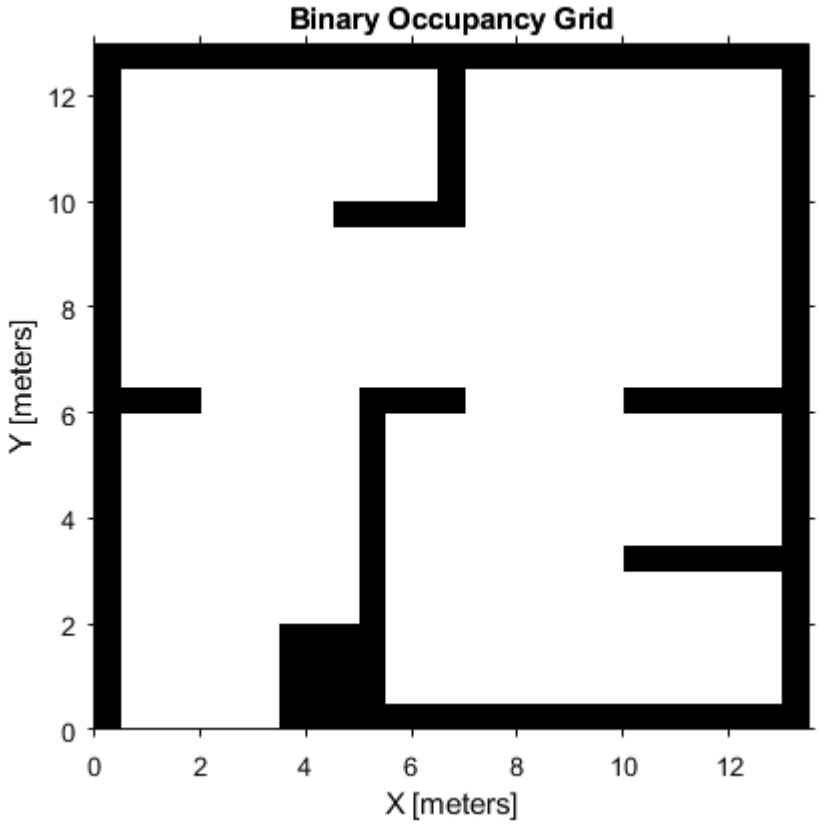
Use the imported simpleMap data and construct an occupancy grid representation using the robotics. BinaryOccupancyGrid class. Set the resolution to 2 cells per meter for this map.

```
map = robotics. BinaryOccupancyGrid (simpleMap, 2)
map =
  BinaryOccupancyGrid with properties:

          GridSize: [26 27]
          Resolution: 2
        XWorldLimits: [0 13.5000]
        YWorldLimits: [0 13]
    GridLocationInWorld: [0 0]
```

Display the map using the show function on the robotics. BinaryOccupancyGrid object

```
show(map)
```



Καθορίζουμε τις διαστάσεις του ρομπότ και μεγαλώνουμε τον πάχος των γραμμών έτσι ώστε να διασφαλίσουμε ότι το ρομπότ δεν συγκρούεται με εμπόδια, θα πρέπει να μεγαλώσουμε τον χάρτη από τη διάσταση του ρομπότ προτού εφαρμόσουμε τη προγραμματιζόμενη διαδρομή PRM.

Στη συνέχεια, μπορούμε να διογκώσουμε το χάρτη με αυτήν τη ιδιότητα χρησιμοποιώντας τη λειτουργία robotics.BinaryOccupancyGrid.inflate.

Δημιουργία πάχους Γραμμών συνυπολογίζοντας την ακτίνα που έχει το ρομπότ έτσι ώστε το πάχος των γραμμών να ικανοποίει τη συνθήκη και να μην συγκρούεται το ρομπότ με εμπόδια

robotRadius = 0,2;

mapInflated = αντίγραφο (χάρτη);

Δημιουργία χάρτη με πάχος γραμμών όσο είναι η ακτίνα του ρομποτ

(mapInflated, robotRadius);

Εμφάνιση χάρτη

εμφάνιση (χάρτη με πάχος γραμμών όσο ακτίνα του ρομπότ)

Δημιουργία PRM και Ορισμός

Τώρα πρέπει να ορίσουμε έναν προγραμματισμό διαδρομών. Δημιουργούμε ένα αντικείμενο robotics.PRM και ορίζουμε τα σχετικά χαρακτηριστικά.

prm = robotics.PRM

Καθορίζουμε τον αριθμό των κόμβων PRM που θα χρησιμοποιηθούν κατά την κατασκευή PRM. Το PRM κατασκευάζει έναν οδικό χάρτη χρησιμοποιώντας έναν δεδομένο αριθμό κόμβων στον συγκεκριμένο χάρτη. Με βάση τη διάσταση και την πολυπλοκότητα του χάρτη εισόδου, αυτό είναι ένα από τα πρωταρχικά χαρακτηριστικά που πρέπει να συντονιστούν για να βρεθεί λύση μεταξύ δύο σημείων στο χάρτη. Ένας μεγάλος αριθμός κόμβων δημιουργούν έναν πυκνό οδικό χάρτη και αυξάνουν την πιθανότητα εξεύρεσης διαδρομής. Ωστόσο, έχοντας περισσότερους κόμβους αυξάνει τον χρόνο υπολογισμού τόσο για τη δημιουργία του χάρτη πορείας όσο και για την εξεύρεση λύσης.

prm.NumNodes = 50;

Καθορίζουμε τη μέγιστη επιτρεπόμενη απόσταση μεταξύ δύο συνδεδεμένων κόμβων στο χάρτη. Το PRM συνδέει όλους τους κόμβους που χωρίζονται από αυτήν την απόσταση (ή λιγότερο) στον χάρτη. Αυτό είναι ένα άλλο χαρακτηριστικό που πρέπει να συντονίσουμε στην περίπτωση μεγαλύτερων και / ή πολύπλοκων χαρτών εισόδου. Μια μεγάλη απόσταση σύνδεσης αυξάνει τη συνδεσιμότητα μεταξύ των κόμβων για να βρει μια διαδρομή ευκολότερη, αλλά μπορεί να αυξήσει το χρόνο υπολογισμού της δημιουργίας πορείας στο χάρτη.

prm.ConnectionDistance = 5;

Βρίσκουμε ένα εφικτό μονοπάτι στο κατασκευασμένο PRM

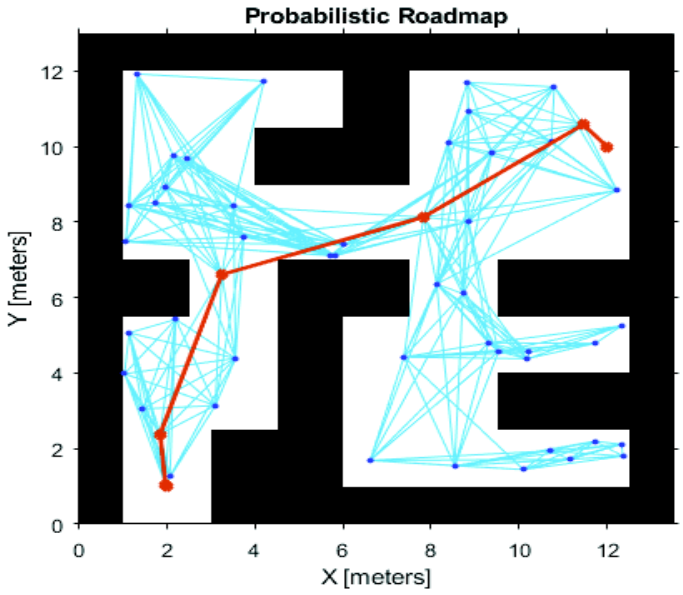
Ορίζουμε τις θέσεις έναρξης και λήξης στο χάρτη για να χρησιμοποιήσετε τον προγραμματιστή διαδρομών. startLocation = [2 1];

endLocation = [12 10].

Αναζητούμε μια διαδρομή μεταξύ των θέσεων έναρξης και λήξης χρησιμοποιώντας τη λειτουργία robotics.PRM.findpath. Η λύση είναι ένα σύνολο σημείων από τη θέση εκκίνησης μέχρι την τελική θέση. Σημειώνουμε ότι η διαδρομή θα είναι διαφορετική λόγω των πιθανοτήτων της φύσης του αλγόριθμου PRM.path = findpath (prm, startLocation, endLocation)

path =

- 2.0000 1.0000
- 1.9569 1.0546
- 1.8369 2.3856
- 3.2389 6.6106

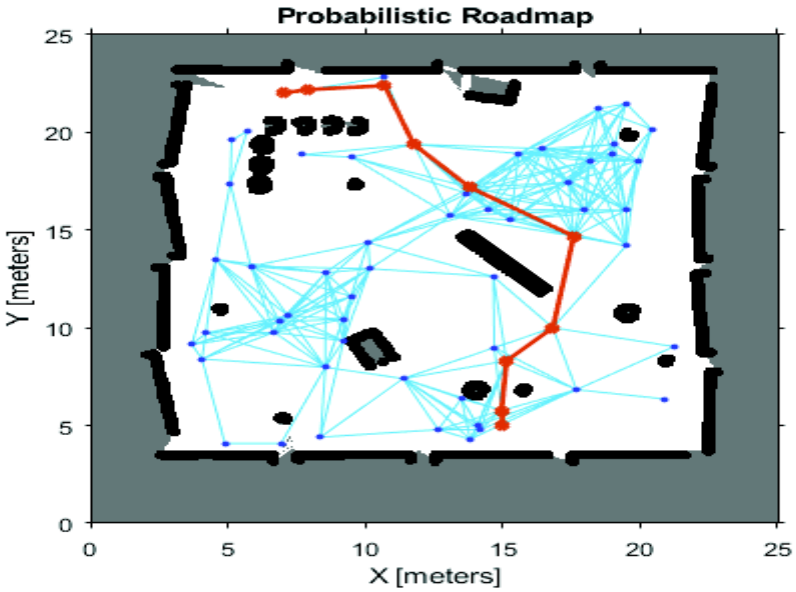


Εικόνα 1 Εμφάνιση Λύσης PRM

2.8.3 Χρησιμοποιούμε το PRM για έναν μεγάλο και περίπλοκο χάρτη.

Χρησιμοποιούμε τα εισαγόμενα δεδομένα complexMap, τα οποία αντιπροσωπεύουν ένα μεγάλο και πολύπλοκο σχέδιο κατόψεων, και κατασκευάστε μια δυαδική αναπαράσταση δικτύου με μια δεδομένη ανάλυση (1 κελί ανά μέτρο)

map = robotics.BinaryOccupancyGrid (complexMap, 1)



Εικόνα 2 Χάρτης με Επιλογη Διαδρομής

2.9 Τι είναι η μέθοδος Kalman Filter

Σύμφωνα με το Wikipedia ο φιλτράρισμα Kalman, γνωστό και ως γραμμική τετραγωνική εκτίμηση (LQE), είναι ένας αλγόριθμος που χρησιμοποιεί μια σειρά μετρήσεων που παρατηρούνται με την πάροδο του χρόνου, που περιέχουν στατιστικό θόρυβο και άλλες ανακρίβειες και παράγει εκτιμήσεις άγνωστων μεταβλητών που τείνουν να είναι ακριβέστερες από εκείνες που βασίζονται σε μόνο μεμονωμένες μετρήσεις, με εκτίμηση μιας κοινής κατανομής πιθανοτήτων στις μεταβλητές για κάθε χρονικό πλαίσιο. Το φίλτρο ονομάζεται από τον Rudolf E. Kálmán, έναν από τους πρωταρχικούς προγραμματιστές της θεωρίας του.

Το φίλτρο Kalman έχει πολλές εφαρμογές στην τεχνολογία. Μια κοινή εφαρμογή είναι η καθοδήγηση, η πλοήγηση και ο έλεγχος των οχημάτων, ιδίως των αεροσκαφών και των διαστημικών σκαφών. Επιπλέον, το φίλτρο Kalman είναι μια ευρέως εφαρμοσμένη έννοια στην ανάλυση χρονοσειρών που χρησιμοποιείται σε πεδία όπως η επεξεργασία σήματος και η οικονομετρία. Τα φίλτρα Kalman αποτελούν επίσης ένα από τα κύρια θέματα στον τομέα του σχεδιασμού και του ελέγχου της ρομποτικής κίνησης και μερικές φορές συμπεριλαμβάνονται στη

βελτιστοποίηση των τροχιών. Το φίλτρο Kalman λειτουργεί επίσης για τον έλεγχο της κίνησης του κεντρικού νευρικού συστήματος. Λόγω της χρονικής καθυστέρησης μεταξύ της έκδοσης και λήψης εντολών από το κινητήρα αισθητηριακής ανάδρασης, η χρήση του φίλτρου Kalman υποστηρίζει ένα ρεαλιστικό μοντέλο για την εκτίμηση της τρέχουσας κατάστασης του κινητήρα και την έκδοση επικαιροποιημένων εντολών

Ο αλγόριθμος λειτουργεί σε μια διαδικασία δύο σταδίων. Στο βήμα πρόβλεψης, το φίλτρο Kalman παράγει εκτιμήσεις των μεταβλητών της τρέχουσας κατάστασης, μαζί με τις αβεβαιότητες τους. Μόλις παρατηρηθεί το αποτέλεσμα της επόμενης μέτρησης (αναγκαστικά καταστραφεί με κάποιο ποσό σφάλματος, συμπεριλαμβανομένου του τυχαίου θορύβου), οι εκτιμήσεις αυτές επικαιροποιούνται χρησιμοποιώντας σταθμισμένο μέσο όρο, δίνοντας μεγαλύτερη βαρύτητα στις εκτιμήσεις με μεγαλύτερη βεβαιότητα. Ο αλγόριθμος είναι αναδρομικός. Μπορεί να λειτουργεί σε πραγματικό χρόνο, χρησιμοποιώντας μόνο τις τρέχουσες μετρήσεις εισόδου και την κατάσταση που είχε υπολογιστεί προηγουμένως και τη μήτρα αβεβαιότητας. δεν απαιτούνται πρόσθετες παρελθούσες πληροφορίες.Χρησιμοποιώντας ένα φίλτρο Kalman δεν υποθέτουμε ότι τα σφάλματα είναι Gaussian. [3] Ωστόσο, το φίλτρο αποδίδει την ακριβή εκτίμηση πιθανότητας υπό όρους στην ειδική περίπτωση που όλα τα σφάλματα είναι Gaussian. Επεκτάσεις και γενικεύσεις στη μέθοδο έχουν επίσης αναπτυχθεί, όπως το εκτεταμένο φίλτρο Kalman και το ασταθές φίλτρο Kalman που λειτουργεί σε μη γραμμικά συστήματα. Το υποκείμενο μοντέλο είναι παρόμοιο με ένα κρυφό μοντέλο Markov, εκτός από το ότι ο χώρος κατάστασης των λανθανόντων μεταβλητών είναι συνεχής και όλες οι λανθάνοντες και παρατηρούμενες μεταβλητές έχουν Gaussian κατανομές.

2.9.1 Παράδειγμα Εφαρμογής Kalman Filter.

Ως παράδειγμα εφαρμογής, εξετάζουμε το πρόβλημα του προσδιορισμού της ακριβούς θέσης ενός ρομπότ. Το ρομπότ μπορεί να είναι εξοπλισμένο με μονάδα GPS που παρέχει μια εκτίμηση της θέσης μέσα σε λίγα μέτρα. Η εκτίμηση GPS είναι πιθανό να είναι θορυβώδης. οι αναγνώσεις «μεταπηδούν γρήγορα», αν και παραμένουν σε λίγα μέτρα από την πραγματική θέση. Επιπλέον, καθώς το ρομπότ αναμένεται να ακολουθήσει τους νόμους της φυσικής, η θέση του μπορεί επίσης να εκτιμηθεί με την ενσωμάτωση της ταχύτητάς του με την πάροδο του χρόνου, που καθορίζεται από την παρακολούθηση των στροφών των τροχών με άλλα λόγια την οδομετρία τους . Αυτή είναι μια τεχνική γνωστή ως νεκρή αναμέτρηση. Συνήθως, ο νεκρός υπολογισμός θα δώσει μια πολύ ομαλή εκτίμηση της θέσης του ρομπότ, αλλά θα μετατοπιστεί με την πάροδο του χρόνου καθώς συσσωρεύονται μικρά σφάλματα.Σε αυτό το παράδειγμα, το φίλτρο Kalman μπορεί να θεωρηθεί

ότι λειτουργεί σε δύο ξεχωριστές φάσεις: να προβλέψει και να ενημερώσει. Στη φάση πρόβλεψης, η παλιά θέση του ρομπότ θα τροποποιηθεί σύμφωνα με τους φυσικούς νόμους της κίνησης (το μοντέλο δυναμικής ή "μεταβατικής κατάστασης"). Όχι μόνο θα υπολογιστεί μια νέα εκτίμηση θέσης, αλλά θα υπολογιστεί και μια νέα συνδιακύμανση. Ίσως η συνδιακύμανση να είναι ανάλογη με την ταχύτητα του ρομπότ και την οδομετρία του κάθε τροχού επειδή είμαστε πιο αβέβαιοι σχετικά με την ακρίβεια της εκτίμησης θέσης νεκρού σε υψηλές ταχύτητες αλλά πολύ σίγουροι για την εκτίμηση θέσης όταν κινούμαστε αργά. Στη συνέχεια, στη φάση ενημέρωσης, μια μέτρηση της θέσης του ρομπότ λαμβάνεται από τη μονάδα GPS. Μαζί με αυτή τη μέτρηση υπάρχει κάποια αβεβαιότητα και η συνδιακύμανση της σε σχέση με την πρόβλεψη από την προηγούμενη φάση καθορίζει πόσο η νέα μέτρηση θα επηρεάσει την ενημερωμένη πρόβλεψη. Στην ιδανική περίπτωση, καθώς οι εκτιμήσεις των νεκρών υπολογισμών τείνουν να απομακρυνθούν από την πραγματική θέση, η μέτρηση GPS πρέπει να τραβήξει την εκτιμώμενη θέση προς την πραγματική θέση, αλλά να μην την διαταράξει πριν γίνουν γρήγορα άλματα και με θόρυβο.

Τεχνική περιγραφή και πλαίσιο. Το φίλτρο Kalman είναι ένα αποδοτικό αναδρομικό φίλτρο που εκτιμά την εσωτερική κατάσταση ενός γραμμικού δυναμικού συστήματος από μια σειρά θορυβώδεις μετρήσεις. Χρησιμοποιείται σε ένα ευρύ φάσμα εφαρμογών μηχανικής και οικονομετρίας από το ραντάρ και την όραση στον υπολογιστή μέχρι την εκτίμηση των διαρθρωτικών μακροοικονομικών μοντέλων και είναι ένα σημαντικό θέμα στη θεωρία ελέγχου και στη μηχανική συστημάτων ελέγχου. Μαζί με τον γραμμικό-τετραγωνικό ρυθμιστή (LQR), το φίλτρο Kalman λύνει το γραμμικό-τετραγωνικό-Gaussian πρόβλημα ελέγχου (LQG). Το φίλτρο Kalman, ο γραμμικός-τετραγωνικός ρυθμιστής και ο γραμμικός-τετραγωνικός-γκαουσιανός ελεγκτής είναι λύσεις σε ό, τι αναμφισβήτητα είναι τα πιο θεμελιώδη προβλήματα στη θεωρία ελέγχου.

Στις περισσότερες εφαρμογές, η εσωτερική κατάσταση είναι πολύ μεγαλύτερη (περισσότεροι βαθμοί ελευθερίας) από τις λίγες "παρατηρήσιμες" παραμέτρους που μετρώνται. Ωστόσο, συνδυάζοντας μια σειρά μετρήσεων, το φίλτρο Kalman μπορεί να εκτιμήσει ολόκληρη την εσωτερική κατάσταση.Στη θεωρία Dempster-Shafer, κάθε εξίσωση ή παρατήρηση ενός κράτους θεωρείται μια ειδική περίπτωση μιας γραμμικής συνάρτησης πεποιθήσεων και το φίλτρο Kalman είναι μια ειδική περίπτωση που συνδυάζει λειτουργίες γραμμικής πεποιθήσεως σε ένα δένδρο ή Markov. Πρόσθετες προσεγγίσεις περιλαμβάνουν τα φίλτρα πεποίθησης που χρησιμοποιούν Bayes ή αποδεικτικά στοιχεία για τις εξισώσεις του κράτους.

Μια μεγάλη ποικιλία από φίλτρα Kalman έχει αναπτυχθεί τώρα, από την αρχική διατύπωση του Kalman, που τώρα ονομάζεται "απλό" φίλτρο Kalman, το φίλτρο Kalman-Bucy, το "εκτεταμένο" φίλτρο του Schmidt, το φίλτρο πληροφοριών και μια ποικιλία " φίλτρα που αναπτύχθηκαν από τον Bierman, τον Thornton και πολλούς άλλους. Ίσως ο συνηθέστερα χρησιμοποιούμενος τύπος πολύ απλού φίλτρου Kalman είναι ο βρόχος με κλειδωμένη φάση, που είναι πλέον πανταχού παρόν σε ραδιόφωνα, ειδικά ραδιόφωνα διαμόρφωσης συχνότητας (FM), τηλεοράσεις, δέκτες δορυφορικών επικοινωνιών, συστήματα επικοινωνιών στο διάστημα και σχεδόν οποιοδήποτε άλλο ηλεκτρονικό εξοπλισμό επικοινωνιών.

Τα φίλτρα Kalman βασίζονται σε γραμμικά δυναμικά συστήματα διακριτοποιημένα στο χρονικό πεδίο. Μοντελοποιούνται σε μια αλυσίδα Markov που είναι χτισμένη σε γραμμικούς χειριστές που διαταράσσονται από σφάλματα που μπορεί να περιλαμβάνουν Gaussian θόρυβο. Η κατάσταση του συστήματος αντιπροσωπεύεται ως φορέας πραγματικών αριθμών. Σε κάθε διακριτή αύξηση χρόνου, ένας γραμμικός χειριστής εφαρμόζεται στην κατάσταση για να δημιουργήσει τη νέα κατάσταση, με κάποιο θόρυβο αναμεμειγμένο, και προαιρετικά μερικές πληροφορίες από τους ελέγχους στο σύστημα, εάν είναι γνωστοί. Στη συνέχεια, ένας άλλος γραμμικός χειριστής που αναμιγνύεται με περισσότερο θόρυβο δημιουργεί τις παρατηρούμενες εξόδους από την πραγματική ("κρυφή") κατάσταση. Το φίλτρο Kalman μπορεί να θεωρηθεί ως ανάλογο με το κρυφό μοντέλο Markov, με τη διαφορά κλειδί ότι οι μεταβλητές κρυφούς κατάστασης παίρνουν τιμές σε ένα συνεχές διάστημα (σε αντίθεση με ένα διακριτό κρατικό χώρο όπως στο κρυφό μοντέλο Markov). Υπάρχει μια ισχυρή αναλογία μεταξύ των εξισώσεων του φίλτρου Kalman και εκείνων του κρυμμένου μοντέλου Markov. Μια ανασκόπηση αυτού και άλλων μοντέλων δίνεται από τους Roweis και Ghahramani (1999), [14] και Hamilton (1994), Κεφάλαιο 13. Για να χρησιμοποιήσουμε το φίλτρο Kalman για να εκτιμήσουμε την εσωτερική κατάσταση μιας διαδικασίας που δίνεται μόνο σε μια ακολουθία θορυβώδεις παρατηρήσεις, πρέπει να μοντελοποιήσουμε τη διαδικασία σύμφωνα με το πλαίσιο του φίλτρου Kalman. Αυτό σημαίνει να προσδιοριστούν οι παρακάτω πίνακες:

- Fk, το μοντέλο μεταβατικής κατάστασης.
- Hk, το μοντέλο παρατήρησης.
- Qk, η συνδιακύμανση του θορύβου της διαδικασίας.
- Rk, η συνδιακύμανση του θορύβου παρατήρησης.
- και μερικές φορές Bk, το μοντέλο εισόδου ελέγχου, για κάθε βήμα χρόνου, k, όπως περιγράφεται παρακάτω.

Το μοντέλο φίλτρου Kalman αναλαμβάνει την πραγματική κατάσταση κατά το χρόνο που το k εξελίσσεται από την κατάσταση στο (k - 1) σύμφωνα με το xk = F kxk - 1 + B kuk + wk

Όπου

- Το Fk είναι το πρότυπο μεταβατικής κατάστασης που εφαρμόζεται στην προηγούμενη κατάσταση xk-1.
- Bk είναι το μοντέλο εισόδου ελέγχου που εφαρμόζεται στο φορέα ελέγχου uk.

- wk είναι ο θόρυβος της διαδικασίας που υποτίθεται ότι προέρχεται από μια μηδενική μέση πολυμεταβλητή κανονική κατανομή, με συνδιακύμανση Qk: wk ~ N (0, Q k}
- Στο χρόνο k μια παρατήρηση (ή μέτρηση) έκ της πραγματικής κατάστασης xk γίνεται σύμφωνα με το $z_k = H_k x_k + vk$

Όπου

- Hk είναι το μοντέλο παρατήρησης που χαρτογραφεί τον πραγματικό χώρο κατάστασης στον παρατηρούμενο χώρο και
- vk είναι ο θόρυβος παρατηρήσεως που θεωρείται ότι είναι μηδενικός μέσος Gaussian λευκός θόρυβος με συνδιακύμανση Rk: vk ~ N (0, Rk
- Η αρχική κατάσταση και οι διανύσματα θορύβου σε κάθε βήμα {x0, w1, ..., wk, v1 ... vk} θεωρούνται όλα ανεξάρτητα μεταξύ τους.Πολλά πραγματικά δυναμικά συστήματα δεν ταιριάζουν ακριβώς με αυτό το μοντέλο. Στην πραγματικότητα, η αδιόρθωτη δυναμική μπορεί να υποβαθμίσει σοβαρά την απόδοση του φίλτρου, ακόμη και όταν υποτίθεται ότι λειτουργεί ως άγνωστο στοχαστικό σήμα ως είσοδο. Ο λόγος γι 'αυτό είναι ότι η επίδραση της απροσδιόριστης δυναμικής εξαρτάται από την είσοδο και επομένως μπορεί να φέρει τον αλγόριθμο εκτίμησης σε αστάθεια (αποκλίνει). Από την άλλη πλευρά, ανεξάρτητα σήματα λευκού θορύβου δεν θα κάνουν τον αλγόριθμο να αποκλίνει. Το πρόβλημα της διάκρισης μεταξύ του θορύβου της μέτρησης και της αδιόρθωτης δυναμικής είναι δύσκολο και αντιμετωπίζεται στη θεωρία ελέγχου στο πλαίσιο του ισχυρού ελέγχου.

2.10 Ενισχυτική Μάθηση για το Λογισμικό και Πλατφόρμες Προσομοίωσης

Πιο συγκεκριμένα στη παρούσα εργασία χρησιμοποιήθηκαν συγκεκριμένα προγράμματα και πλατφόρμες λογισμικού όπως το ROS Indigo Igloo,Matlab,Vrep και το Gazebo τα οποία συνεργάζονται μεταξύ τους για την υλοποίηση του στόχου της Πτυχιακής Εργασίας.

Το ROS Indigo Igloo είναι συμβατό πρωτίστως με την έκδοση του Ubuntu 14.04 LTS (Trusty), αν και άλλα συστήματα Linux, καθώς και Mac OS X, Android και Windows υποστηρίζονται σε διαφορετικό βαθμό το καθένα.

Ένα από τα πλεονεκτήματα της πλατφόρμας Ros και Gazebo είναι το γεγονός της ελεύθερης χρήσης από το κοινό. Με αυτό τον τρόπο δίνεται η δυνατότητα στο καθένα που ενδιαφέρεται να αναπτύξει πλήθος δεξιοτήτων και κατανόησης λογισμικού κώδικα που χρησιμοποιείται στα συστήματα ρομποτικής πλοήγησης.

Το Matlab Το MATLAB είναι ένα περιβάλλον αριθμητικής υπολογιστικής και μια προγραμματιστική γλώσσα τέταρτης γενιάς. Αποθηκεύει και κάνει τις πράξεις με βάση την άλγεβρα μητρών.Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων, ωστόσο είναι πολύ "ισχυρό" και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές από την C++ όπως την while, την switch και την if. Στον τομέα των γραφικών όσον αφορά τον μαθηματικό κλάδο μπορεί να υλοποιήσει συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών και άλλες. Όσον αφορά τον στατιστικό κλάδο μπορεί να υλοποιήσει ιστογράμματα, τομεογράμματα, ραβδοδιαγράμματα, εμβαδογράμματα και άλλα.

Ένας ρομποτικός προσομοιωτής χρησιμοποιείται για να δημιουργήσει μια εφαρμογή για ένα πραγματικό ρομπότ χωρίς να εξαρτάται από πραγματική μηχανή, εξοικονομώντας έτσι κόστος και χρόνο. Σε ορισμένες περιπτώσεις, αυτές οι εφαρμογές μπορούν να μεταφερθούν σε πραγματικό ρομπότ (ή να ξαναχτιστούν) χωρίς τροποποιήσεις.

Ο όρος προσομοιωτής ρομποτικής μπορεί να αναφέρεται σε πολλές διαφορετικές εφαρμογές προσομοίωσης ρομποτικής. Για παράδειγμα, στις εφαρμογές κινητής ρομποτικής, οι προσομοιωτές ρομποτικής με βάση τη συμπεριφορά επιτρέπουν στους χρήστες να δημιουργούν απλούς κόσμους άκαμπτων αντικειμένων και πηγών φωτός και να προγραμματίζουν ρομπότ για να αλληλοεπιδράσουν με αυτούς τους κόσμους. Η προσομοίωση βασισμένη στη συμπεριφορά επιτρέπει δράσεις που είναι πιο βιολογικής φύσης σε σύγκριση με προσομοιωτές που είναι περισσότερο δυαδικοί ή υπολογιστικοί. Επιπλέον, οι προσομοιωτές που βασίζονται στη

συμπεριφορά μπορούν να "μαθαίνουν" από λάθη και είναι σε θέση να επιδείξουν την ανθρωπομορφική ποιότητα της αντοχής.

2.10.1 Παραδείγματα Ρομποτικών Προσομοιωτών

Ρομποτικό προσομοιωτή Robologix.

Μία από τις πιο δημοφιλείς εφαρμογές για εξομοιωτές ρομποτικής είναι η 3D μοντελοποίηση και απόδοση ενός ρομπότ και του περιβάλλοντος του. Αυτός ο τύπος λογισμικού ρομποτικής έχει έναν προσομοιωτή που είναι ένα εικονικό ρομπότ, το οποίο είναι ικανό να εξομοιώνει την κίνηση ενός πραγματικού ρομπότ σε ένα πραγματικό φάκελο εργασίας. Μερικοί προσομοιωτές ρομποτικής χρησιμοποιούν κινητήρα φυσικής για πιο ρεαλιστική κίνηση κίνησης του ρομπότ. Η χρήση ενός προσομοιωτή ρομποτικής για την ανάπτυξη ενός προγράμματος ελέγχου ρομποτικής συνιστάται ανεπιφύλακτα ανεξάρτητα από το αν είναι διαθέσιμο ή όχι ένα πραγματικό ρομπότ. Ο προσομοιωτής επιτρέπει τα ρομποτικά προγράμματα να είναι κατανοητά και να εντοπίζονται με λάθος off-line με την τελική έκδοση του προγράμματος να δοκιμάζεται σε ένα πραγματικό ρομπότ. Αυτό ισχύει κυρίως για βιομηχανικές εφαρμογές ρομπότ, αφού η επιτυχία του off-line προγραμματισμού εξαρτάται από το πόσο παρόμοιο είναι το πραγματικό περιβάλλον του ρομπότ με το προσομοιωμένο περιβάλλον.

Οι λειτουργίες ρομπότ με αισθητήρα είναι πολύ πιο δύσκολο να προσομοιωθούν και / ή να προγραμματιστούν off-line, καθώς η κίνηση του ρομπότ εξαρτάται από τις στιγμιαίες μετρήσεις του αισθητήρα στον πραγματικό κόσμο.

2.10.2 Ρομποτικός Προσομοιωτής Gazebo

Το Gazebo είναι μία ειδική πλατφόρμα η οποία εμπεριέχει σύνολο από πακέτα Ros που παρέχουν τις απαραίτητες δυνατότητες για την προσομοίωση ενός ρομπότ στο περιβάλλον προσομοίωσης Gazebo 3D. Ενσωματώνει τα πακέτα Ros Indigo χρησιμοποιώντας μηνύματα Ros, υπηρεσίες και δυναμική διαμόρφωση του περιβάλλοντος προσομοίωσης.

Προσομοίωση των ρομπότ είναι ένα απαραίτητο εργαλείο στην εργαλειοθήκη κάθε ρομποτικής εφαρμογής. Μια καλά σχεδιασμένη προσομοίωση επιτρέπει να ελέγξουμε γρήγορα αλγόριθμους, το σχεδιασμό ρομπότ, την εκτέλεση δοκιμών παλινδρόμησης, και να εκπαιδεύσει το ρομποτικό μας σύστημα, χρησιμοποιώντας ρεαλιστικά σενάρια. Το Gazebo προσφέρει τη δυνατότητα ακρίβειας και αποτελεσματικότητας της προσομοίωσης του ρομπότ σε πολύπλοκους εσωτερικούς και εξωτερικούς χώρους. Στη διάθεσή μας είναι μια ισχυρή μηχανή φυσικής, υψηλής ποιότητας

γραφικών και άνετου προγραμματιστικού και γραφικού περιβάλλοντός. Επιπλέον είναι ανοιχτό λογισμικό άρα δωρεάν.

2.10.3 Ρομποτικός Προσομοιωτής Vrep

Ο προσομοιωτής ρομπότ V-REP με ενσωματωμένο περιβάλλον ανάπτυξης βασίζεται σε μια αρχιτεκτονική κατανεμημένου ελέγχου: κάθε αντικείμενο / μοντέλο μπορεί να ελέγχεται μεμονωμένα μέσω ενσωματωμένου σεναρίου, ενός plugin, κόμβων ROS, κόμβων BlueZero, απομακρυσμένων υπολογιστών API ή προσαρμοσμένης λύσης . Αυτό καθιστά το V-REP πολύ ευπροσάρμοστο και ιδανικό για εφαρμογές πολλαπλών ρομπότ. Οι ελεγκτές μπορούν να γραφτούν σε C / C ++, Python, Java, Lua, Matlab, Octave ή Urbi.

Μερικές μόνο από τις εφαρμογές του V-REP:

- Προσομοίωση συστημάτων αυτοματισμού εργοστασίων
- Απομακρυσμένη παρακολούθηση
- Έλεγχος υλικού
- Γρήγορο πρωτότυπο και επαλήθευση
- Παρακολούθηση της ασφάλειας
- Ανάπτυξη γρήγορου αλγορίθμου
- Ρομποτική σχετική εκπαίδευση
- Παρουσίαση προϊόντος

Το V-REP μπορεί να χρησιμοποιηθεί ως αυτόνομη εφαρμογή ή μπορεί εύκολα να ενσωματωθεί σε μια κύρια εφαρμογή πελάτη: το μικρό του αποτύπωμα και το περίτεχνο API καθιστά το V-REP ιδανικό υποψήφιο για ενσωμάτωση σε εφαρμογές υψηλότερου επιπέδου. Ένας ενσωματωμένος ερμηνευτής σεναρίου Lua καθιστά το V-REP μια εξαιρετικά ευέλικτη εφαρμογή, αφήνοντας την ελευθερία στον χρήστη να συνδυάζει τις λειτουργίες χαμηλού / υψηλού επιπέδου για να αποκτήσει νέες λειτουργίες υψηλού επιπέδου.

Κεφάλαιο 3

1η ΔΙΕΡΕΥΝΗΣΗ ΠΛΟΗΓΗΣΗΣ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ

3.1 Εγκατάσταση Λογισμικού

Για την υλοποίηση της πρώτης διερεύνησης χρειάστηκε η εγκατάσταση ανοιχτού λογισμικού Ubuntu Linux έκδοση 14.04 LTS. Έπειτα από διεξοδική έρευνα πολλαπλών δοκιμών, κατέληξα ότι το πλέον κατάλληλο και συμβατό για την διεξαγωγή της έρευνας, μέσω της πλατφόρμας Ros Indigo, με σκοπό τη πλοήγηση ρομποτικού συστήματος (Turtlebot) στο χώρο.

Η πλατφόρμα Ros Indigo υποστηρίζει την απελευθέρωση, την τεκμηρίωση και τις δοκιμές ολοκλήρωσης μόνο σε κατασκευασμένα πακέτα από τη βιβλιοθήκη βάση Ros. Αυτό έχει ως ιδιαίτερο στόχο να παράγει μακροπρόθεσμα υποστήριξη αυτής της διανομής και κατασκευής πακέτων και πιο συγκεκριμένα «Rosbuild» πακέτα από την πηγή «Source», τα οποία υποστηρίζονται μέχρι και σήμερα.

3.1.2 Μεθοδολογία Πλοήγησης Ρομποτικών Συστημάτων

1. Περίγραμμα
2. Έννοιες πλοήγησης
3. Χάρτης, Ρομπότ και Path
4. Ταξινόμηση της πλοήγησης
5. Εντοπισμός
6. Σχεδίαση
7. Δημιουργία χάρτη
8. Αίσθηση - Σχέδιο - Πράξη

Για να ολοκληρώσουμε μια εργασία με ένα το ρομπότ στη προκειμένη περίπτωση τη πλοήγηση του στο χώρο, πρέπει να μελετηθεί και να κατανοηθεί πρώτα το περιβάλλον στο οποίο βρίσκεται το ρομπότ με τη βοήθεια των μετρήσεων από τους αισθητήρες. Επεξεργασία των πληροφοριών, και την αφαίρεση άχρηστων στοιχείων. Στην αυτόνομη πλοήγηση θα πρέπει το ρομποτ να γνωρίζει με τι μοιάζει το περιβάλλον και την τοπικοποίηση του μέσα σε αυτό. Ένας χάρτης είναι μια παράσταση του περιβάλλοντος χώρου όπου το ρομπότ λειτουργεί., για αυτό θα πρέπει να περιέχει αρκετές πληροφορίες για την επίτευξη ενός στόχου ενδιαφέροντος. Οι αισθητήρες στηρίζονται κυρίως στις μετρήσεις των αποστάσεων μέσω του πλέγματος που δημιουργούν και

στις συντεταγμένες του χώρου. Ένας αναγνωρισμένος χάρτης ορίζει ένα σημείο αναφοράς από το οποίο αρχίζει η λειτουργία του. Για να λειτουργείσει ο χάρτης, το ρομπότ πρέπει γνωρίζει τη θέση του σε αυτό, σε σχέση με αυτό το σημείο αναφοράς. Για να επιτευχθεί μία διαδρομή σε μία θέση στο χάρτη χρειάζεται μία ακολουθία σημείων ή ενεργειών. Ο εντοπισμός γίνεται με το να προσδιορίζουμε την τρέχουσα θέση ρομπότ, τις μετρήσεις μέχρι την τρέχουσα στιγμή και στο Χάρτη προσδιορίζουμε (εάν υπάρχει) μια διαδρομή για την επίτευξη ενός συγκεκριμένου στόχου, δεδομένου ότι έχουμε ένα τοπικό ρομπότ και ένα διαστρεβλωμένο χάρτη περιοχών με πολλά σημεία από τα οποία μπορεί να περάσει για την επίτευξη του στόχου αποφεύγωντας πιθανά εμπόδια. Χαρτογράφηση γίνεται έχοντας ως δεδομένο ότι έχουμε στη διάθεση μας ένα ρομπότ που έχει τέλεια εκτίμηση θέσης και ιδανική ακολουθία μετρήσεων με αυτό το τρόπο προσδιορίζουμε το περιβάλλον του χάρτη. Μια τέλεια εκτίμηση για ρομπότ συνήθως δεν είναι εφικτή. Αντ΄αυτού επιλύουμε ένα πιο περίπλοκο πρόβλημα το οποίο είναι ο ταυτόχρονος εντοπισμός και η χαρτογράφηση το επονομαζόμενο SLAM. Αυτό υλοποιείται με την εκτίμηση του περιβάλλοντος του χάρτη , την τροχιά της κινούμενης συσκευής, χρησιμοποιώντας μια σειρά από μετρήσεις των αισθητήρων. Τοποθετώντας τα όλα τα αυτά τα δεδομένα μαζί για να μπορέσει να περιηγηθεί ένα ρομπότ χρειάζεται τα παρακάτω:

1. Ένας χάρτης
2. Μια ενότητα εντοπισμού
3. Μονάδα προγραμματισμού διαδρομής

Αυτά τα στοιχεία επαρκούν αν ο χάρτης αντανακλά πλήρως το περιβάλλον. Αν το περιβάλλον είναι στατικό δεν υπάρχουν σφάλματα στην εκτίμηση, ωστόσο αν το περιβάλλον αλλάζει (π.χ. άνοιγμα / κλείσιμο θυρών), ή είναι δυναμικό (τα πράγματα μπορεί να εμφανίζονται / εξαφανίζονται από την αντιληπτή περιοχή του ρομπότ)Συνεπώς εκτίμηση είναι "θορυβώδης" επομένως, πρέπει να συμπληρώσουμε τον ιδανικό σχεδιασμό μας με τα στοιχεία που αφορούν αυτά τα θέματα, δηλαδή ανίχνευση / αποφυγή εμποδίων. Επιπρόσθετα η ένα άλλο σημαντικό κομμάτι είναι τοπική βελτίωση χαρτών. Όπου εκεί κάποιος που ασχολείται με το προγραμματισμό πρέπει να έχει μια πολυεπίπεδη αρχιτεκτονική (τοπική και παγκόσμια σχεδίαση)η ραφιναρισμένη ανίχνευση εμπόδιου on-line με την κατάλληλη mod(σε τοπικό και παγκόσμιο χάρτη. Για την δημιουργία χάρτη υπάρχει πλήθος αλγορίθμων για τη ταυτόχρονη λειτουργία του εντοπισμού και της χαρτογράφησης (SLAM) .Το ROS χρησιμοποιεί το GMapping, το οποίο υλοποιεί ένα σωματίδιο το οποίο λειτουργεί σαν φίλτρο για την παρακολούθηση των τροχιών ρομπότ.

Για να χτιστεί ένας χάρτη πρέπει να Καταγραφεί με / odom, / scan / και / tf κατά την οδήγηση του ρομπότ γύρω από το περιβάλλον στο οποίο πρόκειται να λειτουργήσει Ο χάρτης είναι ένας χάρτης κατοχής και αντιπροσωπεύεται όπως και μια εικόνα που δείχνει το σχέδιο του περιβάλλοντος. Είναι ένα αρχείο διαμόρφωσης (yaml) που δίνει πληροφορίες για τον χάρτη μετά την δημιουργία και αποθήκευση του χάρτη όπως (προέλευση, μέγεθος ενός pixel σε πραγματικό κόσμο. Ο εντοπισμός ενός ρομπότ στο ROS υλοποιείται μέσω του αλγόριθμου Adaptive Monte Carlo Localization Η AMCL χρησιμοποιεί ένα φίλτρο σωματιδίων για την παρακολούθηση της θέσης του ρομπότ. Κάθε στάση αντιπροσωπεύεται από ένα σωματίδιο. Τα σωματίδια είναι μετακινούμενα σύμφωνα με τη (σχετική) κίνηση που μετρήθηκε με την οδομετρία και καταστέλλεται / αναπαράγεται με βάση το πόσο καλά το λέιζερ μπορεί να χωρέσει στον χάρτη, της συγκεκριμένης θέσης του σωματιδίου. Ο εντοπισμός ενσωματώνεται στο ROS εκπέμποντας μεταμορφώνοντας από ένα πλαίσιο χάρτη μέχρι το όνειρο πλαίσιο που "διορθώνει" το οδομετρία. Για να διερευνηθεί τη θέση του ρομπότ σύμφωνα με τον τοπικό χάρτη θα πρέπει να ζητηθεί μετασχηματισμός του base_footprint στο map πλαίσιο του εντοπισμού και του MARRtino Η AMCL βασίζεται σε λέιζερ αν δεν θέλει κάποιος να περάσει 5 χιλιάδες ευρώ, δεν θα πρέπει να γίνει με ένα λέιζερ, έτσι το ρομπότ σας δεν θα εντοπιστεί με αυτή την απόδειξη Ωστόσο μπορούμε να πάρουμε ένα kinect / xtion αισθητήρα, αυτό παρέχει δεδομένα χρήσιμα για την προσομοίωση ενός λέιζερ σαρωτή. Αυτά τα δεδομένα μπορούν στη συνέχεια να συνδεθούν στην AMCL evoila ' τα οποία μπορούν να τρέξουν στο σύστημά μας

3.2 Τι είναι το Turtlebot

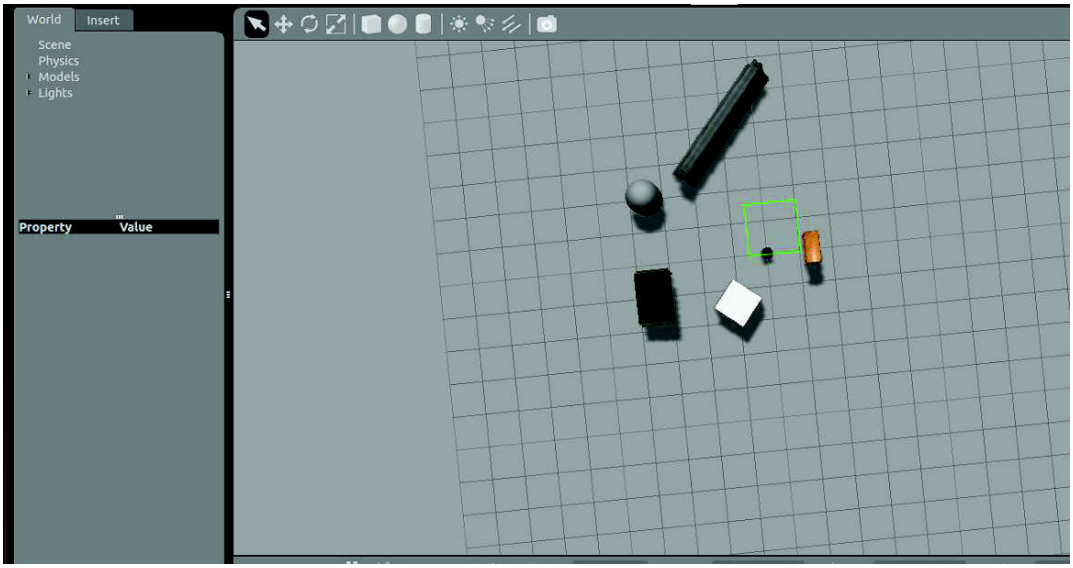
Το «Turtlebot» είναι μια υλική (hardware) πλατφόρμα που είναι ανοιχτή και προσβάσιμη στο κοινό, με φορητή βάση. Το «Tutlebot» μπορεί να χειριστεί όραση, επικοινωνία, κίνηση και την τοπικοποίηση του. Έχει τη δυνατότητα μεταφοράς αντικειμένων, τα οποία είναι τοποθετημένα πάνω του, οπουδήποτε χρειάζεται να πάνε αποφεύγοντας αντικείμενα που μπορεί να υπάρχουν στο πέρασμα του. Αυτό μπορεί να μην είναι από μόνο του κάτι εκπληκτικό , αλλά η τεχνολογία αυτή χρησιμοποιείται από μια μεγάλη γκάμα εταιριών για την υλοποίηση ρομποτικών άκρων.

3.3 Πως θα χρησιμοποιήσουμε το Turtlebot ;

Αρχικά τοποθετούμε το Turtlebot σε ένα περιβάλλον εξομοίωσης το οποίο ανοίγουμε, μέσα από την εκτέλεση διάφορων εντολών στο τερματικό μας, αφού πρώτα έχουμε κάνει εγκατάσταση του λογισμικού Ubuntu Linux 14.04 LTS και της πλατφόρμας Ros Indigo Igloo.

Εκτελούμε τις παρακάτω εντολές στο τερματικό μας:

Roslaunch turltebot_gazebo turtlebot_world.launch



Με αυτή την εντολή ανοίγουμε το περιβάλλον προσομοίωσης του gazebo, το οποίο υπάρχει προ εγκατεστημένο στη βάση δεδομένων που έχουμε δημιουργήσει με την εγκατάσταση των δεδομένων της πλατφόρμας Ros Indigo. Επίσης δίνεται η δυνατότητα επιλογής συγκεκριμένης προσομοίωσης που αφήνει στον χρήστη τη δυνατότητα δημιουργίας εξατομικευμένου περιβάλλοντος και αποθήκευσης.

3.3.1 Εκτέλεση Πλοήγησης Demo

Roslaunch turtlebot_gazebo amcl.demo.launch

Εκτελεί και παίρνει πληροφορίες από το παραπάνω χάρτη για τις συντεταγμένες των αντικειμένων και για το που βρίσκεται το Turtlebot μέσα στο χάρτη.

Η εκτέλεση της παρακάτω εντολής σε συνεργασίας με το gazebo amcl.demo μας αναπαριστά μέσα από τους αισθητήρες του Turtlebot που βρίσκονται τα αντικείμενα στο περιβάλλον προσομοίωσης, που βρίσκεται το Turtlebot μέσα στο χάρτη αλλά και μπορούμε να κάνουμε εκτέλεση εντολών εντός περιβάλλοντος ώστε δίνοντας του ένα στόχο μπορεί να υλοποίηση τη αυτόνομα την πλοήγηση μέσα στο χάρτη και να μην χτυπήσει πάνω στα εμπόδια.

Roslaunch turtlebot_rviz_launchers view_navigation.launch

3.4 Εισαγωγή στο Χάρτη Προσομοίωσης

Δημιουργία φακέλου για καινούρια προσομοίωση:

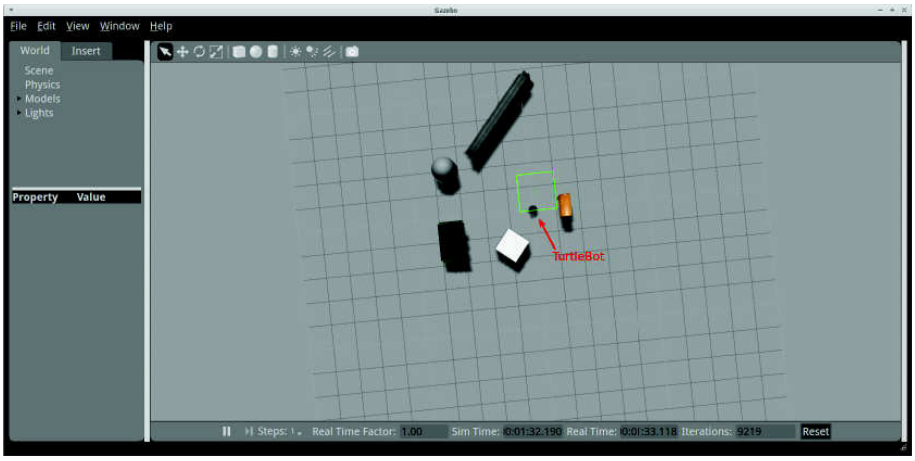
mkdir ~/turtlebot_custom_gazebo_worlds

Τρέχει στο τερματικό το συγκεκριμένο το setup της πλατφόρμα indigo.

Source /opt/ros/indigo/setup.bash

Η εντολή αυτή ανοίγει ένα καινούριο παράθυρο προσομοίωσης με το turtlebot και διάφορα αντικείμενα γύρω του, το οποίο είναι το tutorial της ηλεκτρονικής πλατφόρμας. Η εικόνα της προσομοίωσης αυτής μαζί με τα αναπαριστόμενα σχέδια είναι σε θέση να μας παρέχουν ένα διαδραστικό περιβάλλον.

roslaunch turtlebot_gazebo turtlebot_world.



roslaunch είναι ένα εργαλείο του οποίου η χρήση βοηθάει στην εκτέλεση πολύπλοκων nodes, τοπικά και μέσω απομακρυσμένης διαχείρισης μέσω IP καθώς και στην επεξεργασία παραμέτρων στο Parameter Server. Εμπεριέχει ως επιλογές αυτόματες επαναλαμβανόμενες διεργασίες αναπαραγωγής οι οποίες έχουν ήδη τερματιστεί. Το roslaunch μπορεί να πάρει μία ή παραπάνω διαμορφώσεις XML αρχείων(με κατάληξη launch), αυτό εξιδανικεύει τις παραμέτρους για να αρχικοποιηθούν στα nodes αρχεία ώστε να εκτελεστούν, καθώς και στις μηχανές που πρέπει να τρέξουν.

Maintainer status: maintained

Maintainer: Dirk Thomas <dthomas AT osrfoundation DOT org>

Author: Ken Conley

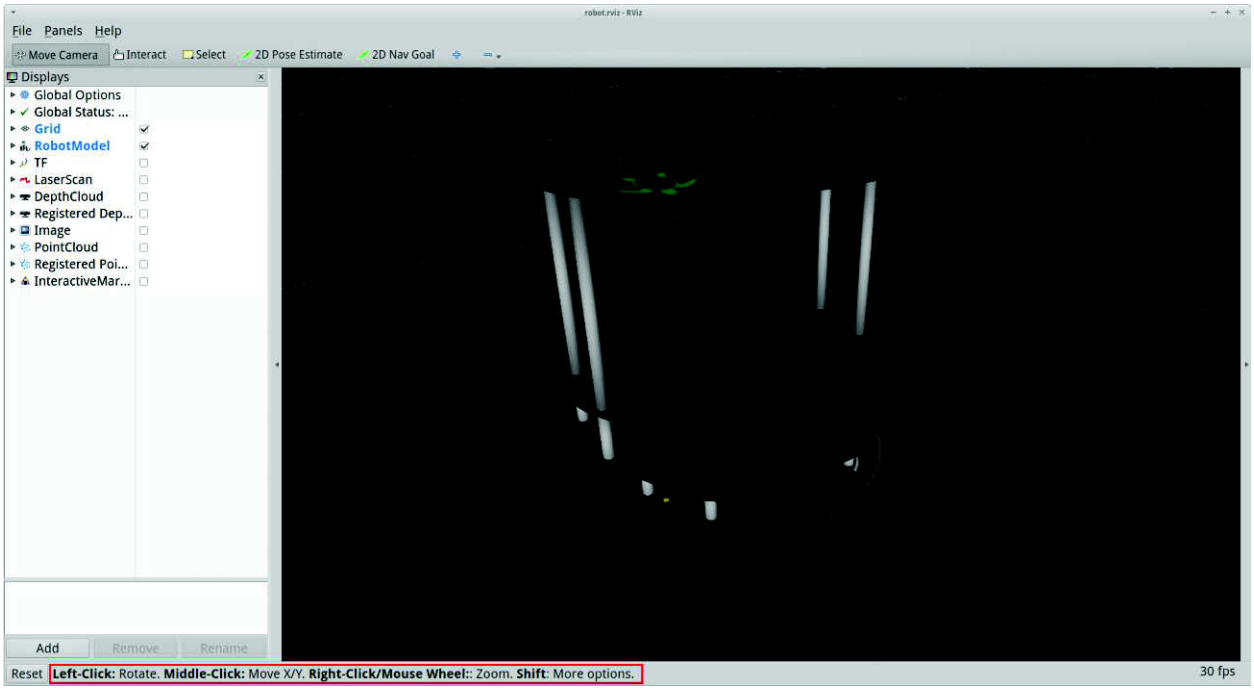
License: BSD

3.5 Εισαγωγή RVIZ VISUALISATION

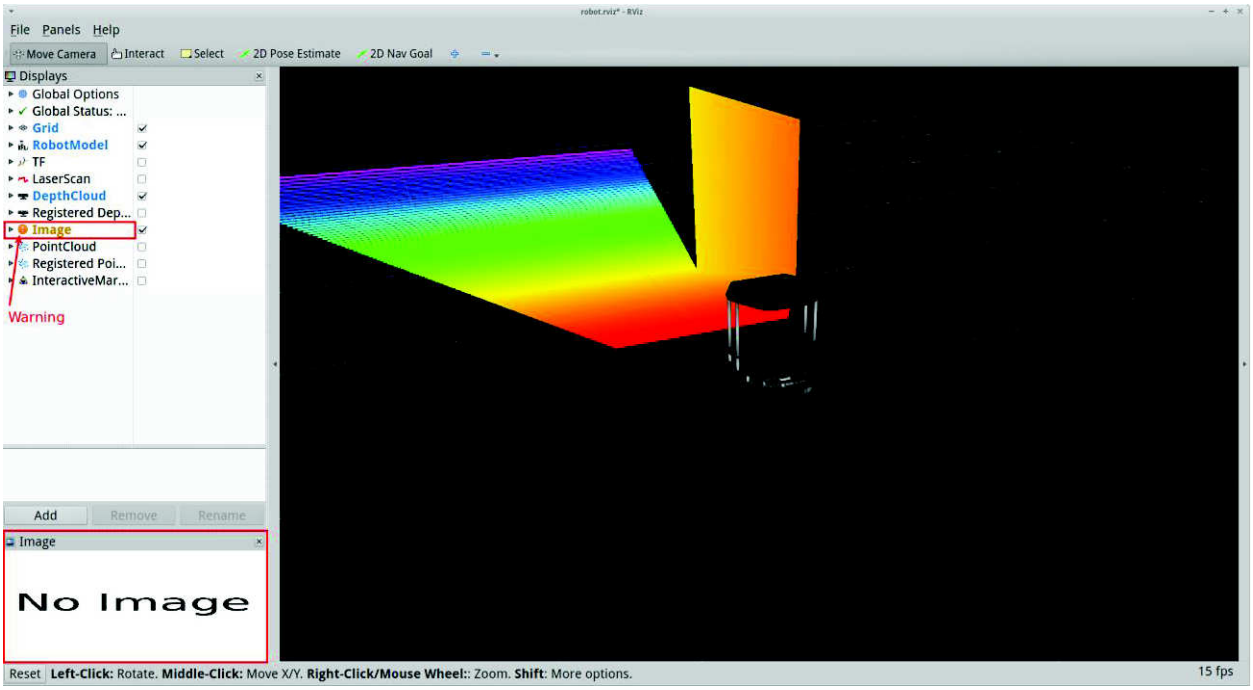
Το RVIZ είναι μία 3D αναπαράσταση του περιβάλλοντος προσομοίωσης της πλατφόρμας ROS. Rviz μας αναπαριστά το περιβάλλον προσομοίωσης μέσα από τα μάτια του Turtlebot. Αυτό μας

δίνει τη δυνατότητα να πάρουμε χρήσιμες πληροφορίες για την εξερεύνηση και την δημιουργία του χάρτη προσομοιώνοντας τους αισθητήρες του. Επίσης, δίνει τη δυνατότητα δοκιμή όλων των δυνατοτήτων που έχουν οι αισθητήρες του turtlebot.

Η εντολή αυτή ανοίγει ένα καινούριο παράθυρο περιβάλλοντος προσομοίωσης μέσα από τα μάτια του turtlebot.



Εικόνα 3 Το Turtlebot στο περιβάλλον RVIZ σε άδειο περιβάλλον



Εικόνα 4 Το Turtlebot στο περιβάλλον RVIZ σε άδειο περιβάλλον Ενεργοποιεί τους Αισθητήρες Laser

3.5.1 Δοκιμή του Simulation

Το Kinect αποτελείται από τρία μέρη που λειτουργούν μαζί:

- Μία κάμερα RGB
- Ένα αισθητήρα βάθους
- Ένα μικρόφωνο πολλαπλών συστοιχιών

Το Turtlebot χρησιμοποιεί το Kinect για να δει τον κόσμο σε 3D και για την ανίχνευση και τον εντοπισμό αντικειμένων.

Για τη χρήση των αισθητήρων αυτών τρέχουμε τις παρακάτω εντολές.

Echo apt-get install ros-indigo-openni-*

3D ΔΟΚΙΜΗ ΤΟΥ ΑΙΣΘΗΤΗΡΑ

Echo $TURTLEBOT _3D_SENSOR

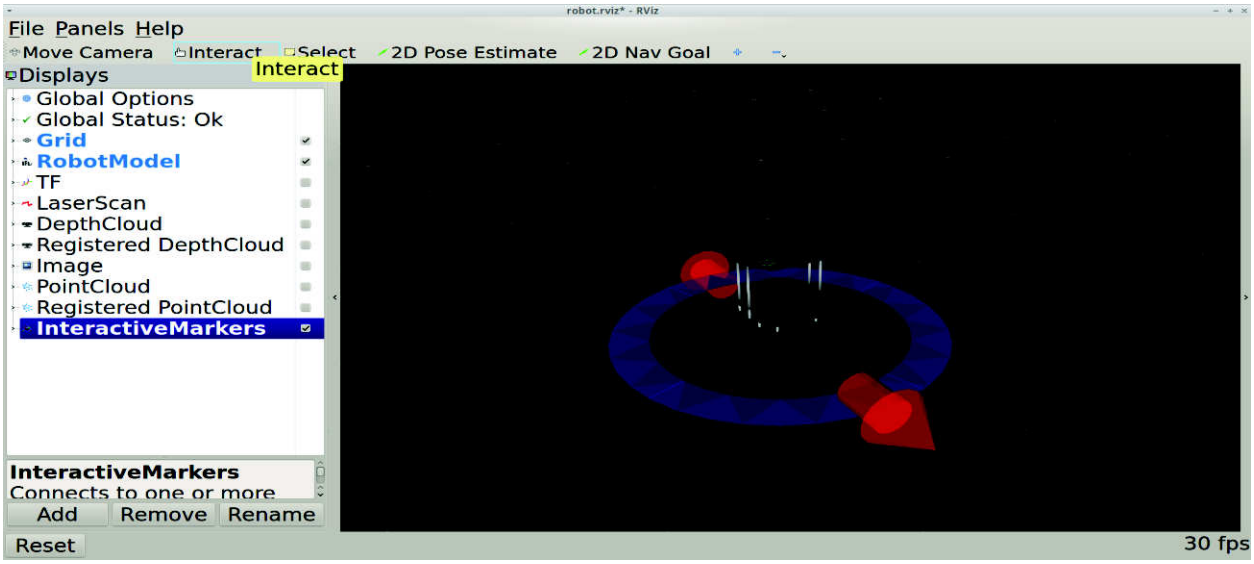
Διαδραστικοί Δείκτες - Interactive Markers

sudo apt-get install ros-indigo-turtlebot-interactive-markers

roslaunch turtlebot_interactive_markers interactive_markers.launch

Ακολουθούμε τα παρακάτω βήματα για να δείτε τις κινήσεις του ρομπότ:

1. Ενεργοποιούμε την επιλογή "Διαλογικές ενδείξεις" από την αριστερή γραμμή.
2. Επιλέγουμε το εργαλείο αλληλεπίδρασης στον πίνακα εργαλείων στο πάνω μέρος της οθόνης. Θα δείτε μπλε δαχτυλίδι και κόκκινα βέλη.
3. Σύρουμε τα κόκκινα βέλη για να οδηγήσετε το Turtlebot προς τα εμπρός και προς τα πίσω.
4. Σύρουμε τον μπλε δακτύλιο για να περιστρέψετε το Turtlebot.
5. Μπορούμε να σύρουμε το μπλε δαχτυλίδι για να περιστρέψουμε και να οδηγήσουμε ταυτόχρονα.



Εικόνα 5 Το Turtlebot στο περιβάλλον RVIZ σε άδειο περιβάλλον Ενεργοποιεί το χειρισμό τηλεκίνησης με Διαδραστικού Δείκτες

3.6 Φτιάχνοντας το Άδειο Χάρτη Προσομοίωσης

Καθορισμός του κόσμου για προσομοίωση

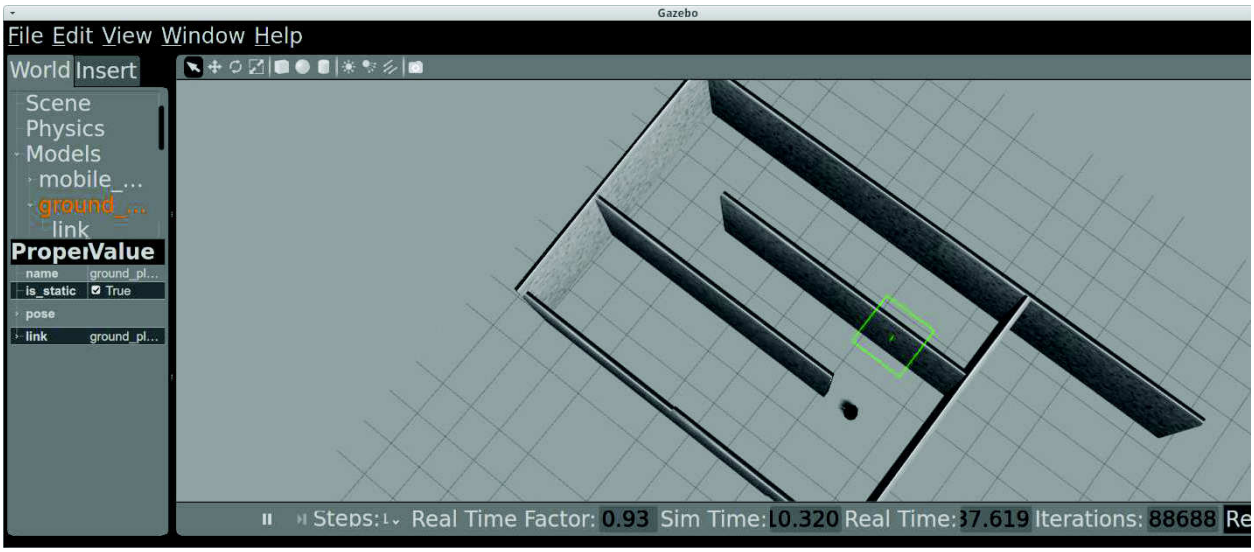
mkdir ~/turtlebot_custom_gazebo_worlds

Χρησιμοποιούμε την ακόλουθη εντολή για να εκτελέσουμε τον κόσμο του Gazebo:

Το roslaunch είναι ένα σημαντικό εργαλείο που διαχειρίζεται την έναρξη και τη διακοπή των διαδικασιών ROS. Πολλά πακέτα ROS έρχονται με αρχεία εκκίνησης, τα οποία μπορούμε να εκτελέσουμε:

Roslaunch turtlebot_gazeboturtlebot_world.launch

world_file:=/opt/ros/indigo/share/turtlebot_gazebo/worlds/corridor.world

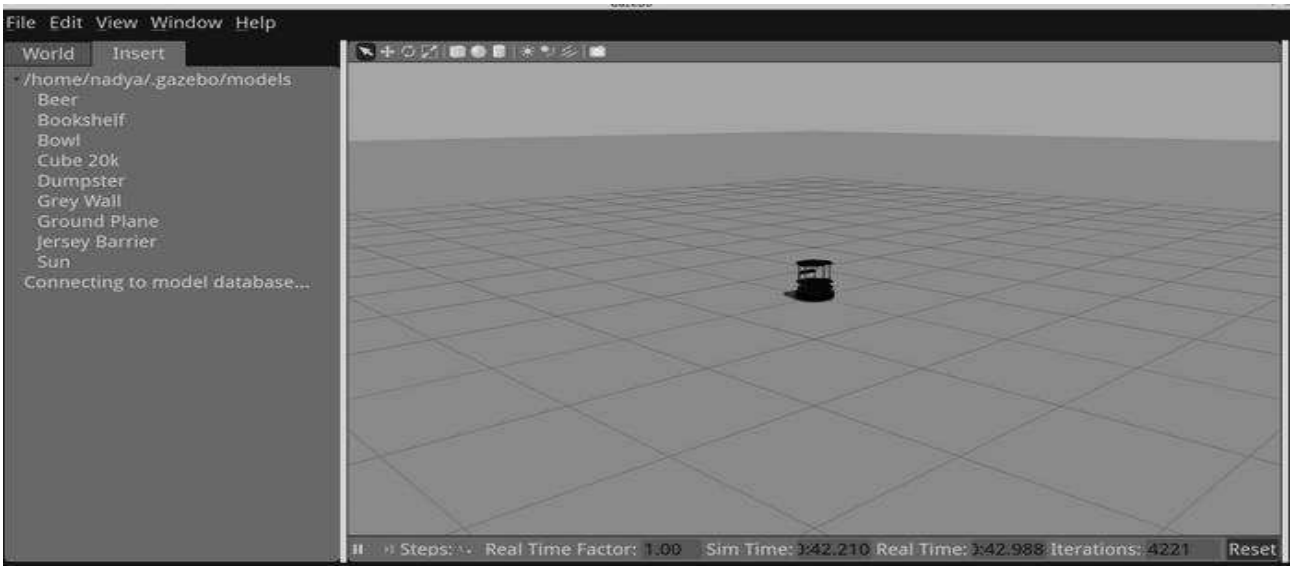


Εικόνα 6 Εμφάνιση Έτοιμου Χάρτη Προσομοίωσης Διάδρομος

Roslaunch turtlebot_gazebo Turltebot_world.launch

mkdir ~/turtlebot_custom_gazebo_worlds

Εκτέλεση εντολής για δημιουργία άδειας προσομοίωσης
roslaunchturtlebot_gazeboturtlebot_world.launch
world_file:=/opt/ros/indigo/share/turtlebot_gazebo/worlds/empty.world



Εικόνα 7 Turtlebot σε περιβάλλον κενό Gazebo

Δημιουργία κόσμου και τοποθέτηση αντικειμένων μέσα σε αυτόν πατώντας το κουμπί εισαγωγή.

Επίσης μπορεί να υπάρξει σύνδεση με τη βάση δεδομένων του μοντέλου του Gazebo.

3.7 Διαδικασία Δημιουργίας χάρτη Προσομοίωσης

Το περιβάλλον που θα ερευνήσουμε στη συγκεκριμένη πτυχιακή είναι ένα διαμέρισμα προσομοίωσης το οποίο το υλοποιήθηκε στο Building Editor του Gazebo, το οποίο σαν δομή του διαμερίσματος και ιδέα αποκτήθηκε από το open Gernan Robocup 2017 με θέμα Cocktail Party.

3.8 Χρήση Building Editor του Gazebo

Η δημιουργία χάρτη με το εργαλειό Building Editor στο Gazebo, μας δίνει τη δυνατότητα δημιουργίας χάρτη διαμερισμάτων με ακρίβεια και σαν επιπρόσθετη λειτουργία μας δίνει τη δυνατότητα να χωριστεί ακόμα και σε ορόφους ο χάρτης μας αυτό γίνεται με την πρόσθεση τοίχων για τη δημιουργία και διαμόρφωση του διαμερίσματος, παραθύρων, σκάλας και πορτών.

Αφού δημιουργηθεί ο χάρτης έτσι όπως απαιτείται αποθηκεύεται σε script .xml ή sdf.

Το επόμενο βήμα είναι η προσαρμογή του χάρτη μέσα σε gazebo world και μπαίνουμε στη διαδικασία δημιουργίας ενός script ενσωματώνοντας μέσα σε έναν απλό χάρτη άδειο gazebo.world με αποτέλεσμα να τοποθετήσουμε το ρομποτάκι μας μέσα στο διαμέρισμα για την επικείμενη εξερεύνηση.

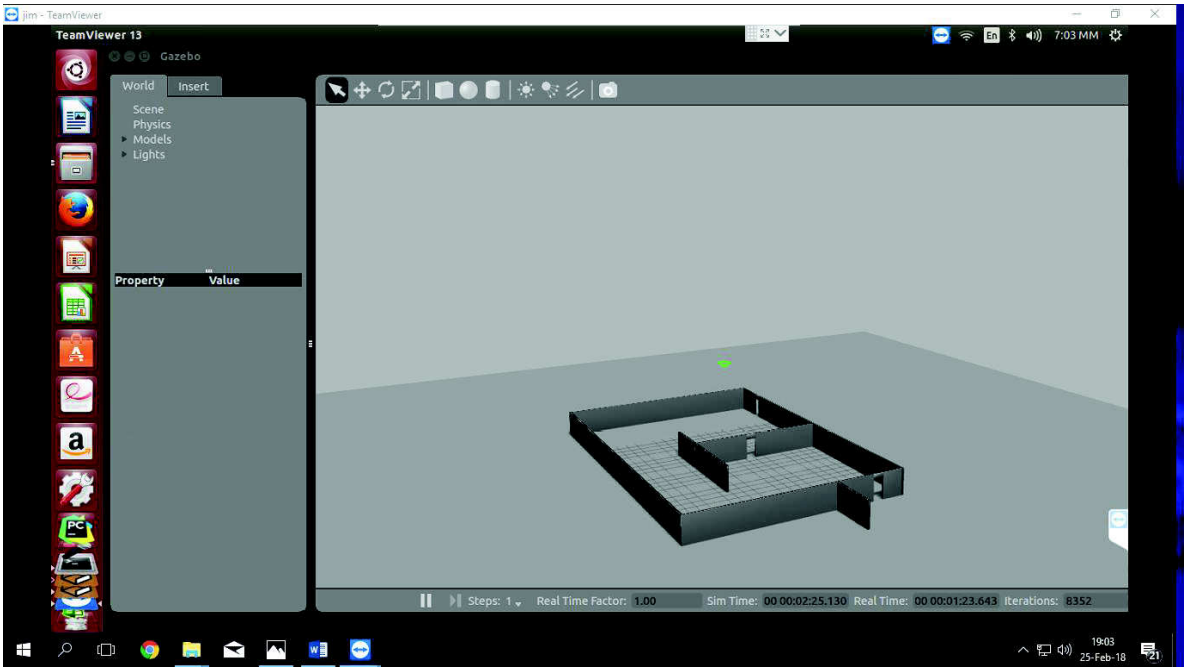
3.9 Προσθήκη Turtlebot σε χάρτη Προσομοίωσης

Αφού πρώτα ενσωματώσουμε το turtlebot μέσα στο χάρτη πρέπει να μπούμε στη διαδικασία αποθήκευσης ως gazebo.world ώστε να μπορέσουμε να το τρέξουμε ως Simulation και να έχουμε τη δυνατότητα πλοήγησης, χαρτογράφησης με ποικίλους και διαφορετικού τρόπους λειτουργίας που μας δίνει το λογισμικό Ros Indigo στο χάρτη που έχουμε δημιουργήσει.

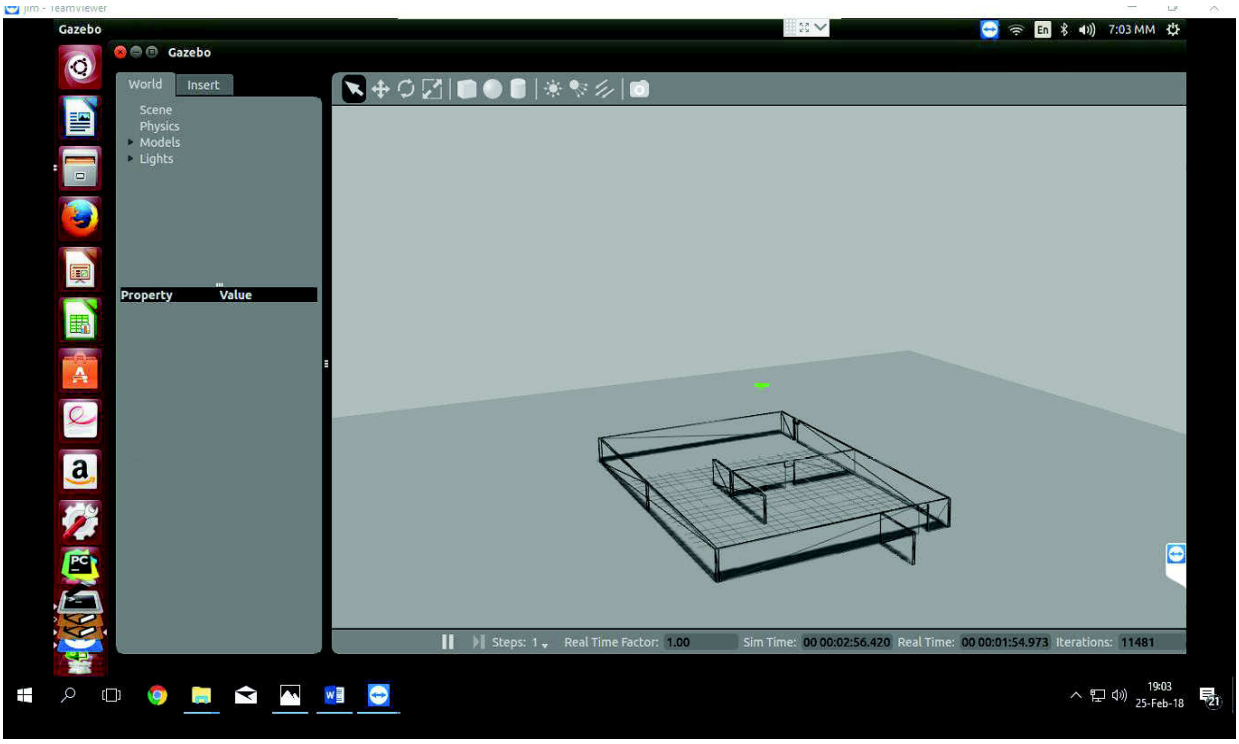
Η χρήση του χάρτη mitsosdokimh2.world στην διερεύνηση μας μέσα από το περιβάλλον ros και το συνδυασμό το χάρτη που έχει δημιουργηθεί για να γίνει πιο κατανοητή η λειτουργία του turtlebot μέσα στο Gazebo και όχι στο Demo. Ο χάρτης προσομοίωσης είναι ένα διαμέρισμα σπιτιού, το οποίο αποτελείται από διαφορετικά δωμάτια με δυνατότητα προσθήκης νέων μοντέλων μέσα από την ηλεκτρονική πλατφόρμα gazebo τα οποία μπορούμε να τα κατεβάσουμε ανοίγοντας απλά το σύνδεσμο του Gazebo που είναι στην αρχική στήλη του παραθύρου προσομοίωσης.

Τρέχοντας την παρακάτω εντολή σε νέο Τερματικό διαπιστώνουμε ότι αποθηκευμένος χάρτης λειτουργεί και είναι αυτός στον οποίο θα εργαστούμε κατά τη διάρκεια αυτής της 1ης Διερευνητικής Πλοήγησης.

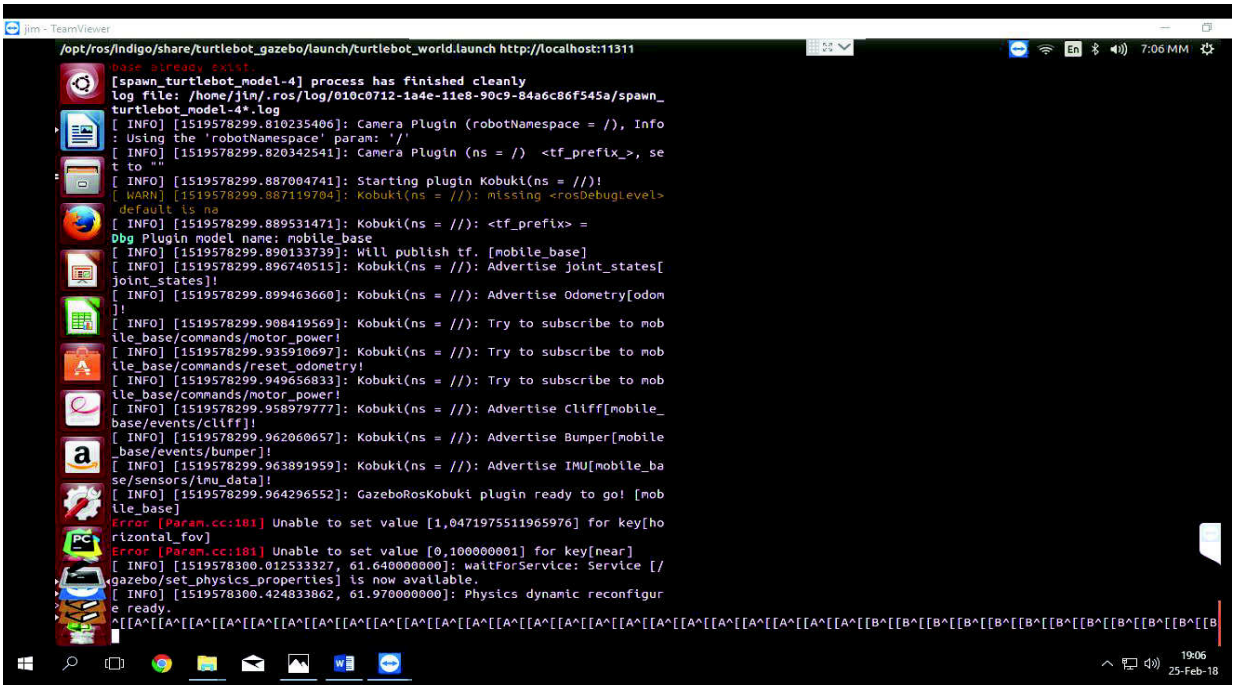
roslaunch turtlebot_gazebo turtlebot_world.launch world_file:=worlds/mitsosdokimh2.world



Εικόνα 8 Δημιουργία Χάρτη Σε Περιβάλλον Gazebo



Εικόνα 9 Δημιουργία Χάρτη Σε Περιβάλλον Gazebo Χωρίς Τοίχο ώστε να δούμε που βρίσκεται το turtlebot.

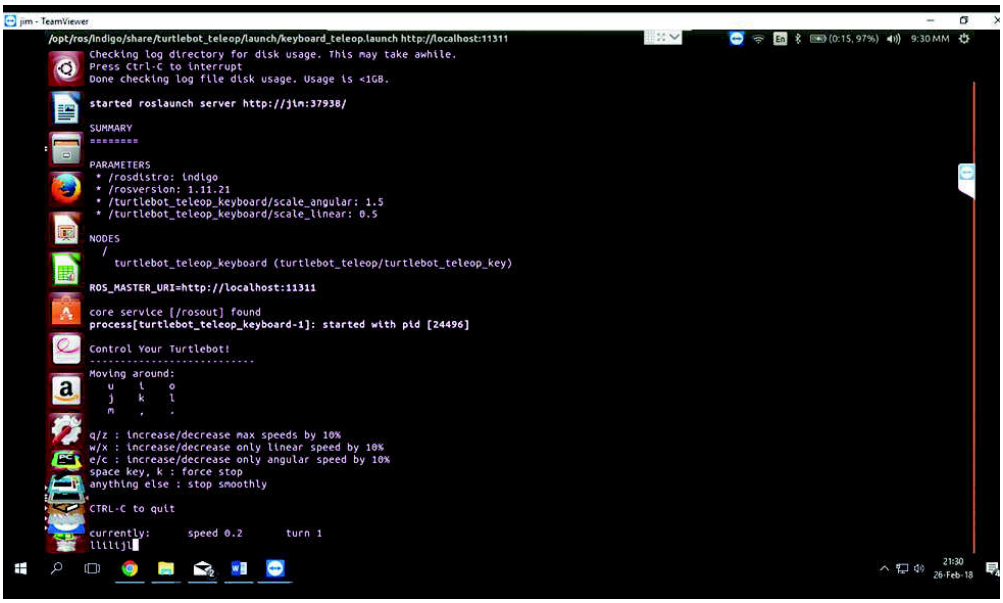


Εικόνα 10 Ο κώδικας που τρέχει στο τερματικό την ώρα που είναι ενεργό το παράθυρο προσομοίωσης

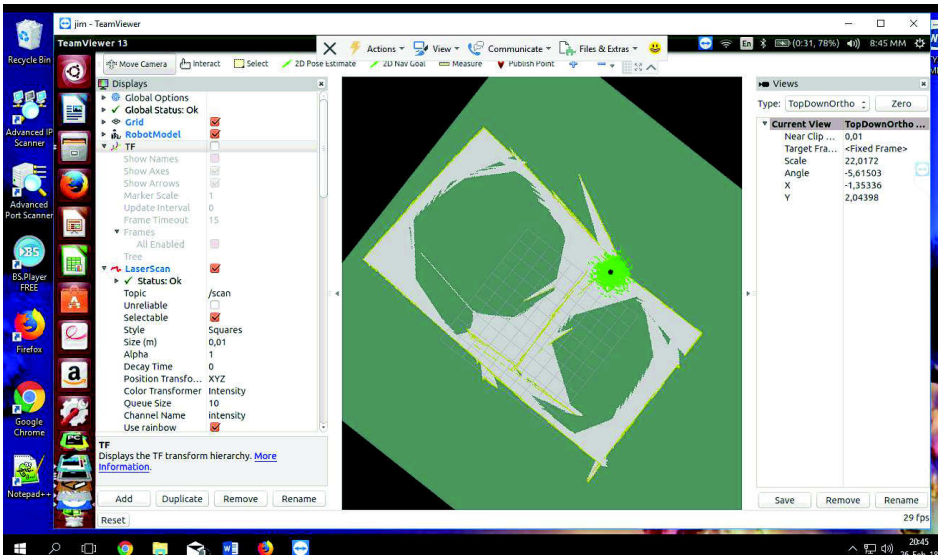
Το επόμενο βήμα πριν πλοηγηθούμε στο χάρτη είναι αναγνώριση του χάρτη με τη βοήθεια της πλατφόρμα του ros rviz view το οποίο κάνει χρήση των αισθητήρων του turtlebot ώστε να καταγράψει το χάρτη προσομοίωσης από άκρη σε άκρη όπως φαίνεται στις παρακάτω εικόνες.

Με τη βοήθεια της τηλεκίνησης ή αλλιώς joystick πλοηγούμαστε μέσα στο χάρτη ώστε να γίνει αναγνώριση του περιβάλλοντος.

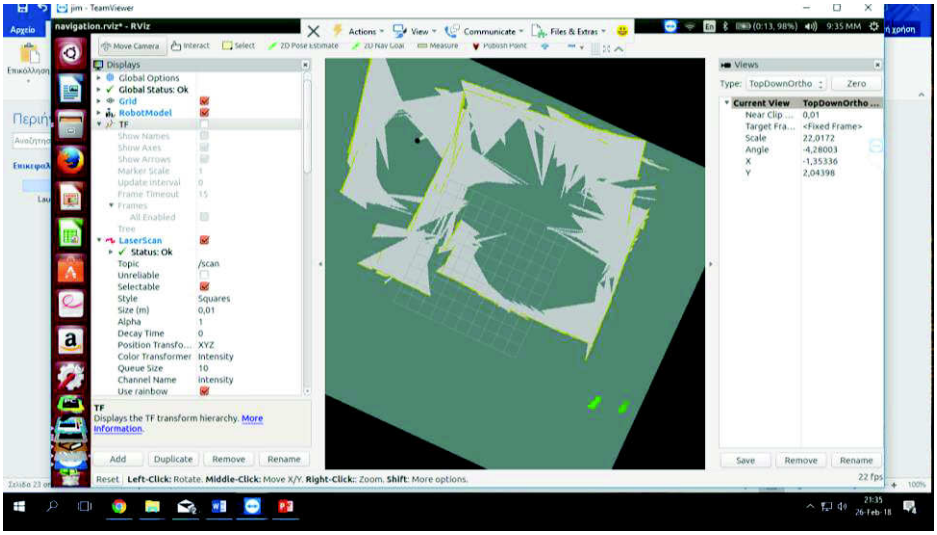
Στη παρακάτω εικόνα φαίνονται τα κουμπιά με τα οποία πλοηγούμαστε μέσα στο χάρτη



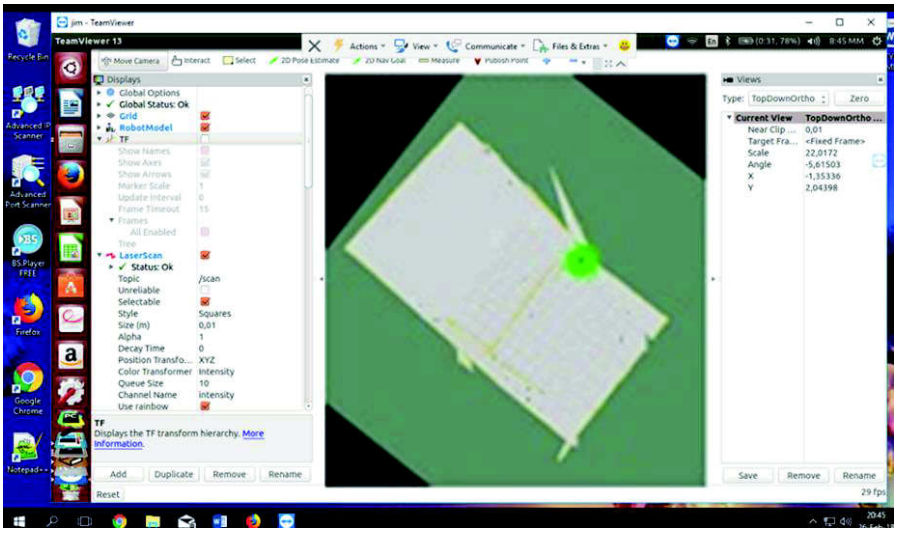
Εικόνα 11 Ενεργοποίηση του Χειρισμου Τηλεκίνησης με τη Βοήθεια του πληκτρολογίου.



Εικόνα 12 Στη εικόνα αυτή έχει γίνει μερική αναγνώριση του χάρτη

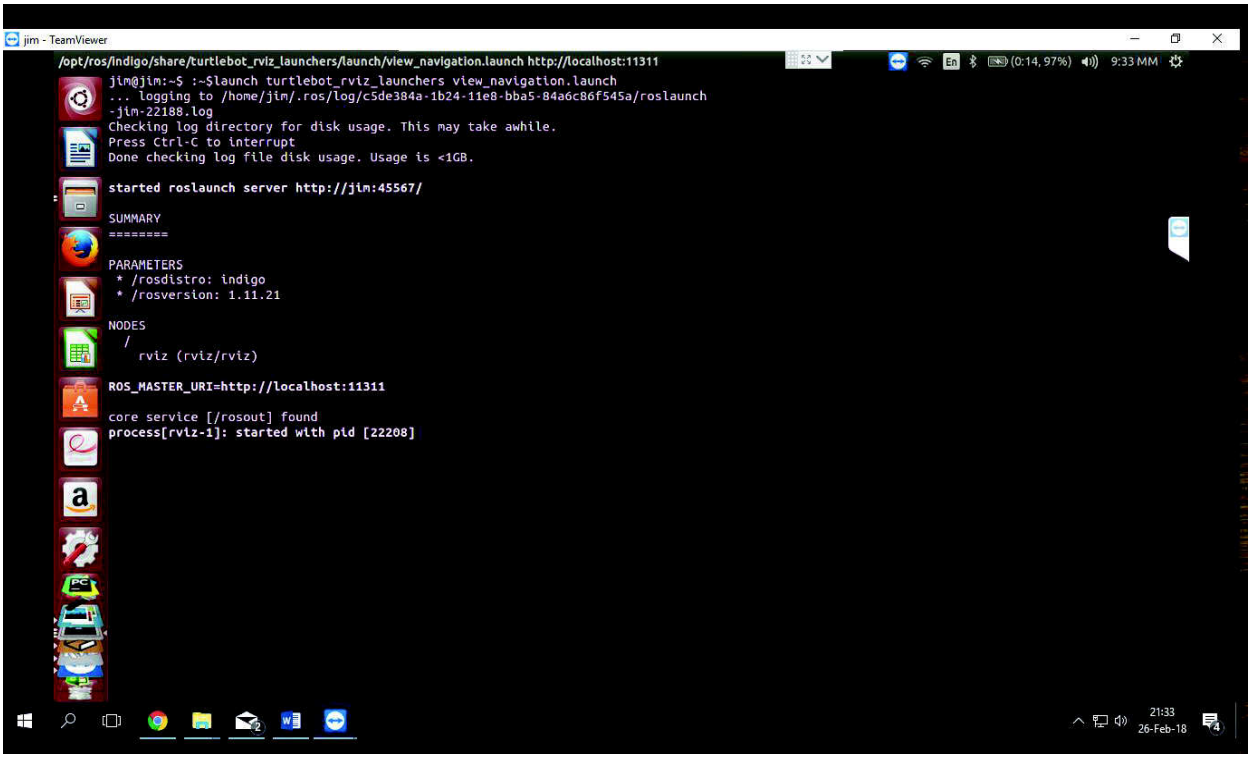


Εικόνα 13 Στη εικόνα αυτή έχει γίνει μερική αναγνώριση του χάρτη

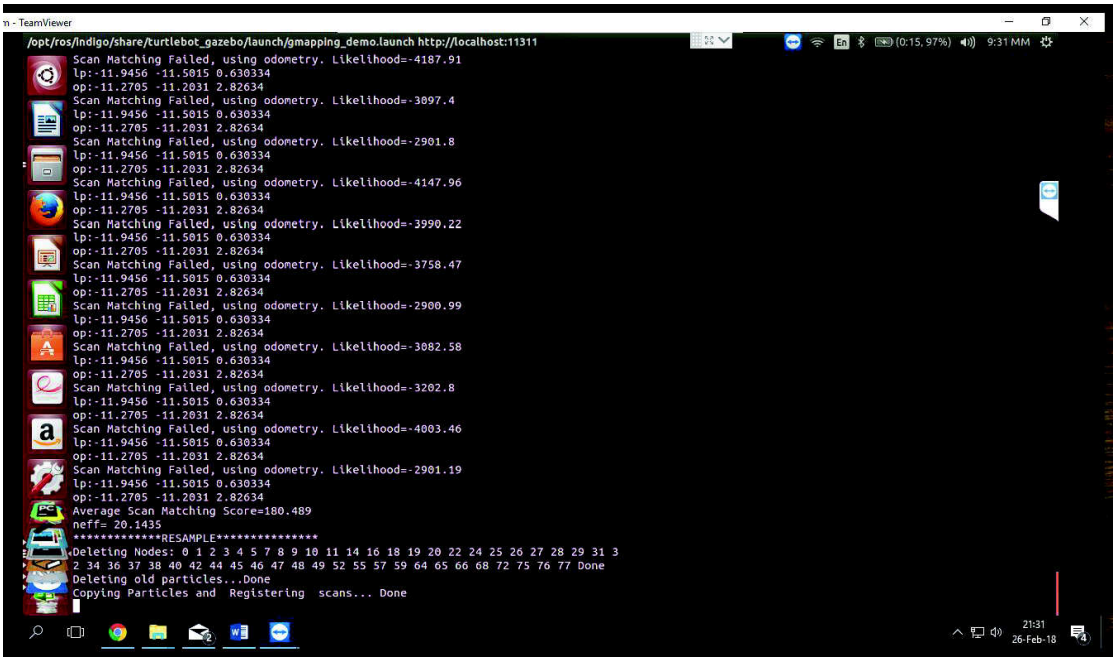


Εικόνα 14 Στη εικόνα αυτή έχει γίνει ολοκλήρωση της αναγνώρισης του χάρτη

Αφού ολοκληρωθεί ο χάρτης αποθηκεύεται και τρέχοντας σε νέο τερματικό εντολές μπορούμε να θέσουμε στόχους στο ρομποτάκι μας έτσι ώστε να μπορέσει να φτιάξει δρόμο και να αποφύγει τυχόν εμπόδια και να φτάσει στο προορισμό του.



Εικόνα 15 Κώδικας που τρέχει στο τοπικό δίκτυο στην βοηθητικη πλατφορμα RVIZ



Εικόνα 16 Αποθηκευση συντεταγμένων του χάρτη προσομοίωσης που αναγνώρισε το turtlebot

Η παρακάτω εντολή ανοίγει ένα καινούριο παράθυρο προσομοίωσης με το turtlebot και ένα διαδραστικό ξύλινο λαβύρινθο με την ονομασία willow garaze μέσα στο οποίο κινείται στο χώρο. Η εικόνα της προσομοίωσης αυτής μαζί με τα αναπαριστόμενα σχέδια είναι σε θέση να μας παρέχουν ένα διαδραστικό περιβάλλον.

roslaunch turtlebot_gazebo turtlebot_world. launch world_file:=worlds/willowgarage.world

export TURTLEBOT_BASE=create

export TURTLEBOT_STACKS=circles

export TURTLEBOT_3D_SENSOR=asus_xtion_pro

roslaunch turtlebot_gazebo gmapping_demo.launch

roslaunch turtlebot_rviz_launchers view_navigation.launch

rosrun map_server map_saver -f <your map name>

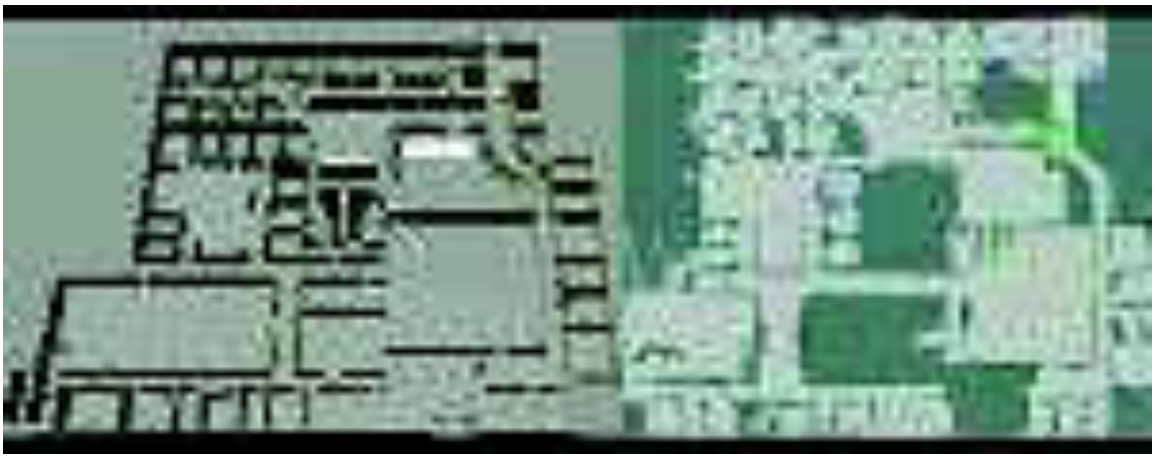
roslaunch turtlebot_gazebo amcl_demo.launch map_file:=<full path to your map YAML file>

roslaunchturtlebot_gazeboturtlebot_world.launch
world_file:=/home/<user_name>/turtlebot_custom_gazebo_worlds/tutorial.world

Ανοίγει το σε νέο παράθυρο προσομοίωσης τον αποθηκευμένο χάρτη.

Τρεξιμο της προσομοίωσης

roslaunchturtlebot_gazeboturtlebot_world.launch
world_file:=/home/<user_name>/turtlebot_custom_gazebo_worlds/tutorial.world
roslaunch turtlebot_gazebo turtlebot_world.launch world_file:=<full path to the world file>



Εικόνα 17 Χάρτης willow garaze έπειτα από αναγνωριση από εικονική Turtlebot

3.10 Χρήση Έτοιμου Κώδικα και εκτέλεση κώδικα σε Χάρτη Προσομοίωσης

Η εκτέλεση του κώδικα θα γίνει σε γλώσσα προγραμματισμού Python, όπως αναφέραμε στην αρχή και επειδή το λογισμικό είναι ανοιχτού κώδικα θα χρησιμοποιήσουμε κώδικα, άλλου δημιουργού με τη βοήθεια της βιβλιοθήκης github.con. Έτσι μπορούμε να κλονοποιήσουμε τον κώδικα επονομαζόμενο markswillman τον οποίο θα χρησιμοποιήσουμε για την εξερεύνηση του περιβάλλοντος προσομοίωσης, την χαρτογράφηση του περιβάλλοντα χώρου προσομοίωσης και την αποφυγή αντικειμένων στον χώρο.

Αρχικά αφού εγκαταστήσουμε όλα τα απαραίτητα αρχεία αρχίζουμε εκτελούμε εντολές σε καινούρια τερματικά ώστε να υλοποιήσουμε το σκοπό μας, που είναι για αρχή η χαρτογράφηση του χάρτη μας που στη περίπτωση μας είναι το περιβάλλον προσομοίωσης willow garaze.

Εκτελώντας την παρακάτω εντολή μπαίνουμε στο περιβάλλον προσομοίωσης του willow garaze η οποία έχει αποθηκευτεί με τη μορφή mitsos .sdf.

Roslaunchturtlebot_gazeboturtlebot_world.launch
world_file=/home/jim/turtlebot_custom_gazebo_worlds/mistos.sdf

Εκτελώ σε νέο τερματικό

Cd ~/helloworld

Cd turtlebot

Εκτελώ την εντολή για να πάει μπροστά το turltebot

Python gorfward.py

Φτιάχνω ένα αντίγραφο του της παραπάνω εντολής και το ονομάζω goincircles.py

Cp goforward.py goingcircles.py (mitsoscode.py)

Αλλάζω τα δεδομένα μέσα, από το ήδη υπάρχον πρόγραμμα, έτσι ώστε να κάνει κύκλο με ακτίνα 0,5 ακτίνια και κέντρο το 0

Μπαίνω μετά στον κώδικα μέσω της παρακάτω εντολής και κάνω τροποποιήσεις.

Emacs –nw goingcircles.py

Linear.x 0.2 =>0 angular.z 0 =>0.5

Press ctrl+X, ctrl+S , save ctrl+X, ctrl+C, exit

και έπειτα τρέχω την εντολή και βλέπω το ρομποτάκι να περιστρέφεται χωρίς σταματημό.

Για να κάνει τετράγωνο τώρα

Python draw_a_square.py

Για να δημιουργήσουμε τώρα χάρτη με χαρτογράφηση

Roslaunch turtlebot_gazebo turtlebot_world.launch

Roslaunch turtlebot_gazebo gmapping_demo.launch

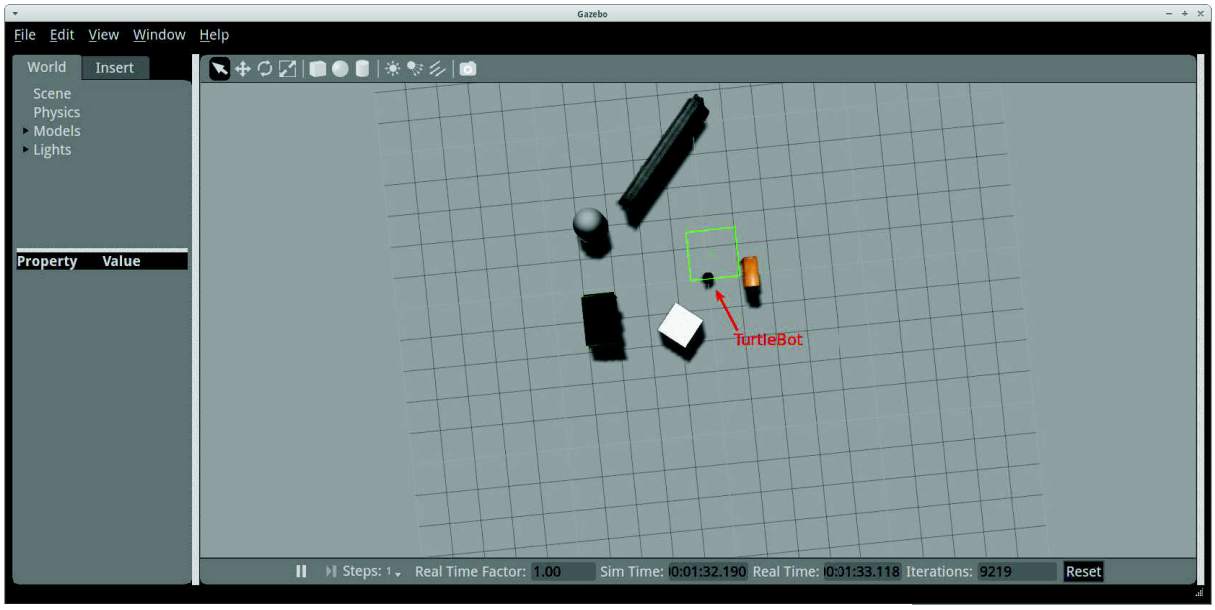
Roslaunch turtlebot_rviz_launchers view navigation.launch

3.11 Αναγνώριση Περιβάλλοντος Προσομοίωσης με χρήση Τηλεκίνησης και Διαδραστικου Δείκτες.

Για να δημιουργήσουμε τώρα χάρτη με χαρτογράφηση για να πιο κατανοητό κάνουμε χρήση ήδη υπάρχοντος αρχείου demo που ανήκει στο ros.

Αρχικά ανοίγουμε το περιβάλλον προσομοίωσης.

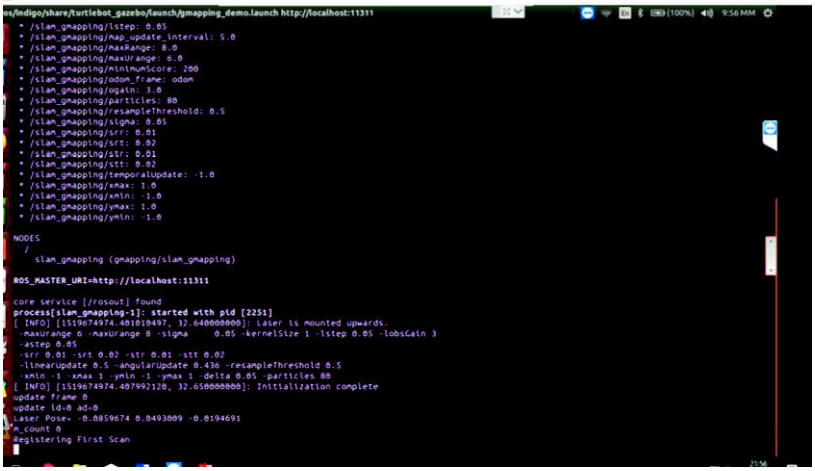
Roslaunch turtlebot_gazebo turtlebot_world.launch



Εικόνα 18 Tutlebot στη πλατφορνα Gazebo στο Tutorial world

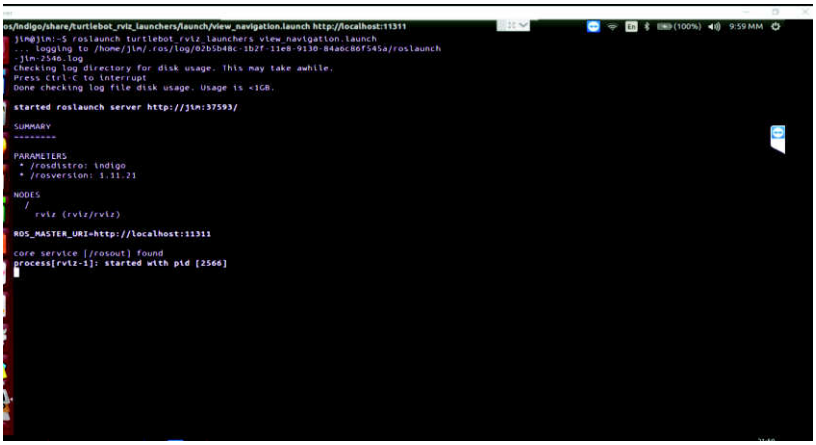
Έπειτα κάνουμε αναγνώριση του περιβάλλοντος με τη βοήθεια των αισθητήρων.

Roslaunch turtlebot_gazebo gmapping_demo.launch

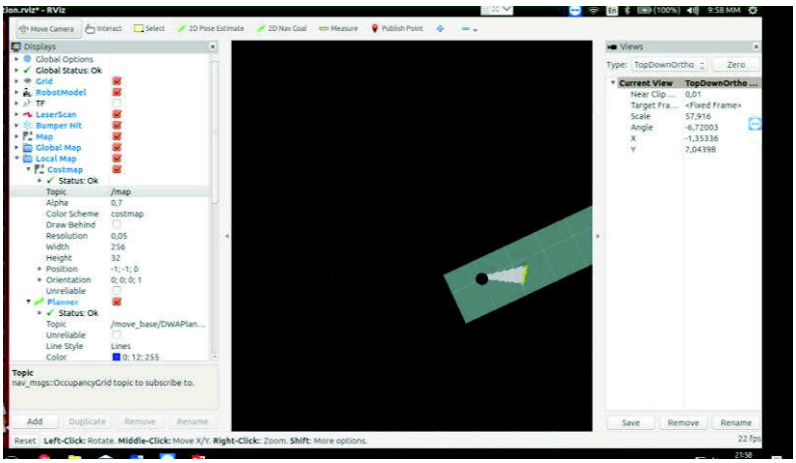


Εικόνα 19 Δημιουργία τύπου data base για την αποθήκευση συντεταγμένων στο χάρτη

Roslaunch turtlebot_rviz_launchers view navigation.launch



Εικόνα 20 Δημιουργία τύπου data base για την αποθήκευση συντεταγμένων στο χάρτη

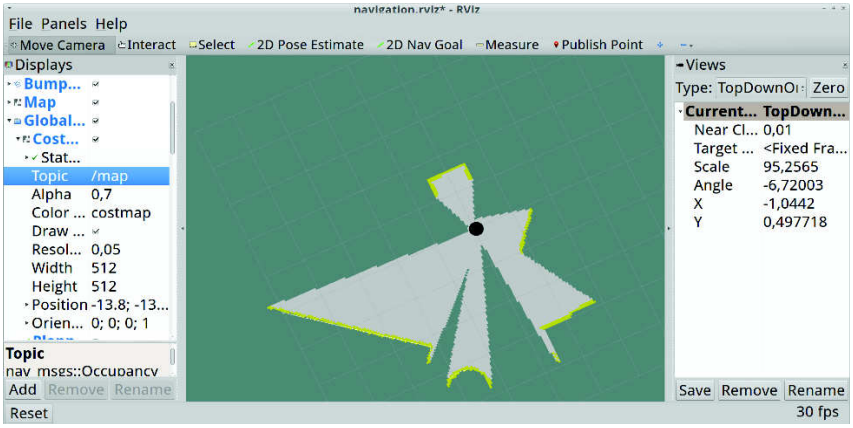


Εικόνα 21 Εκίνηση Αναγνώριση Χάρτη στην πλατφόρμα Rviz

Η λεπτομέρεια η οποία πρέπει να γίνει γνωστή είναι ότι πρέπει να είναι ταυτόχρονα ανοιχτό το τερματικό της τηλεκίνησης του RVIZ και το simulation οποία συνεργάζονται και για να γίνει σωστή εξερεύνηση του χάρτη και το τερματικό του gmappin το οποίο ουσιαστικά αποθηκεύει συντεταγμένες και την αναγνώριση του χάρτη μέσα από το οποίο στο επόμενο κεφάλαιο θα κάνουμε τη χρήση του μέσα από ένα άλλο μοντέλο script το οποίο θα μπορεί να κινηθεί στο χάρτη και θα μπορέσει να αναγνωρίσει τα εμπόδια που έχει δει κατά τη διάρκεια της αρχικής αναγνώρισης.

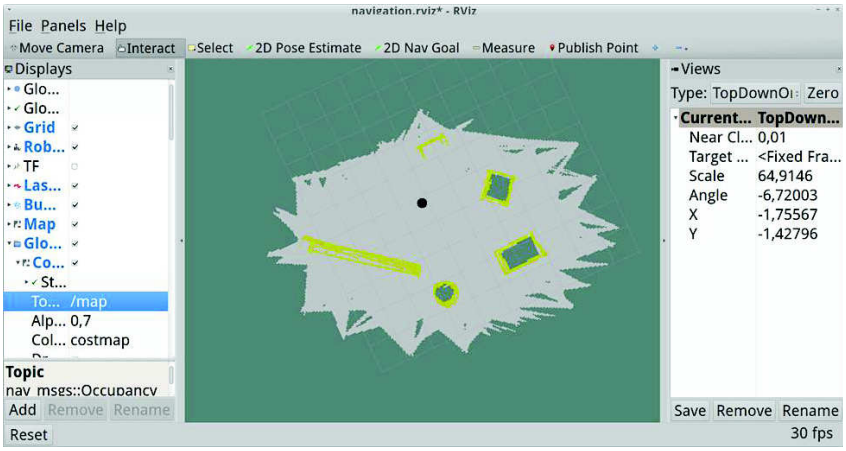
Αφού εκτελέσουμε τις παραπάνω εντολές έχουμε το παρακάτω αποτέλεσμα όπως φαίνεται στην εικόνα.

Ξεκινάμε τη τηλεκίνηση μέσα στο χάρτη προσομοίωσης ώστε να κάνουμε αναγνώριση όλου επιθυμητού χάρτη



Εικόνα 22 Μερική Αναγνώριση Χάρτη στην πλατφόρμα Rviz

Αφού κάνουμε αναγνώριση όλων των αντικειμένων αποθηκεύουμε το χάρτη που έχει δημιουργηθεί.



Εικόνα 23 Ολική Αναγνώριση Χάρτη στην πλατφόρμα Rviz

3.12 Φτιάχνοντας το Χάρτη Προσομοίωσης με Τηλεκίνηση

Χρησιμοποιούμε την παρακάτω εντολή ώστε να ελέγξουμε το χελωνάκι μας με κουμπιά από το πληκτρολόγιο όπως μας δείχνει η παρακάτω εικόνα και να κάνουμε χαρτογράφηση του περιβάλλοντος προσομοίωσης και να οραματιστούμε το περιβάλλον προσομοίωσης μέσα από το χελωνάκι με τη βοήθεια της εφαρμογής που ανοίξαμε στο κεφάλαιο 4.5.1 RVIZ.

Roslaunch turtlebot_teleop keyboard_teleop.launch

http://learn.turtlebot.com/2015/02/03/5/

3.13 Φτιάχνοντας το Χάρτη Προσομοίωσης με Διαδραστικούς Δείκτες

Μπορούμε να ελέγξουμε το Turtlebot σε Rviz. Βεβαιωνόμαστε ότι έχουμε τους διαδραστικούς δείκτες ως εγκατεστημένα πακέτα:

Εκτελώντας την παρακάτω εντολή μας ενημερώνει αν έχουμε τα πακέτα των διαδραστικών δεικτών αλλιώς μας κατεβάζει τα πακέτα που χρειαζόμαστε για τη λειτουργίας τους..

sudo apt-get install ros-indigo-turtlebot-interactive-markers

Έπειτα αφού βεβαιωθούμε ότι έχουμε εγκαταστημένα τα πακέτα το διαδραστικών εντολών εκτελούμε την παρακάτω εντολή παράλληλα με τα προγράμματα που τρέχουνε ανοιχτά του περιβάλλοντος προσομοίωσης και του RVIZ.

roslaunch turtlebot_interactive_markers interactive_markers.launch

Ακολουθούμε τα παρακάτω βήματα για να δούμε τις κινήσεις του ρομπότ:

1. Ενεργοποιούμε την επιλογή Interactive Μαρκαδόροι από την αριστερή μπάρα του RVIZ.

2. Επιλέξτε το εργαλείο Interact στον πίνακα εργαλείων στο πάνω μέρος της οθόνης. Θα δείτε το μπλε δαχτυλίδι και κόκκινα βέλη.

3. Σύρουμε τα κόκκινα βέλη για να οδηγήσει το Turtlebot προς τα εμπρός και προς τα πίσω.

4. Σύρουμε το μπλε δαχτυλίδι για να περιστρέψει τον Turtlebot.

5. Μπορούμε να σύρουμε το μπλε δαχτυλίδι για να περιστρέψουμε και να οδηγηθεί ταυτόχρονα.

3.14 Αποθήκευση Αναγνωρισμένου χάρτη σε αρχείου RVIZ.

Rosrun map_server map_saver –f /home/jim/turtlebot_custom_maps/tutorial

Testing new map

Roslaunchturtlebot_gazebo
amcl_demo.launchmap_file:=/home/jim/turtlebot_custom_maps/tutorial.yaml

3.15 Αναγνώριση γνωστού χάρτη προσομοίωσης με χρήση έτοιμου κώδικα Bukernov.

Άνοιγμα χάρτη προσομοίωσης.

Τρέξιμο της προσομοίωσης

roslaunchturtlebot_gazeboturtlebot_world.launch
world_file:=/home/<user_name>/turtlebot_custom_gazebo_worlds/tutorial.world

Άνοιγμα αναγνωρισμένου χάρτη προσομοίωσης

Roslaunchturtlebot_gazeboamcl_demo.launch
map_file:=/home/jim/turtlebot_custom_maps/mitsosdokimh2.yaml

Άνοιγμα rviz το οποίο βλέπουμε μέσα από το turtlebot

mkdir ~/helloworld

cd ~/helloworld/

Download sources.

git clone https://github.com/markwsilliman/turtlebot/

Change directory.

cd turtlebot

5. Launch script.

python goforward.py

Creating and Modifying a Script

Change directory.

cd ~/helloworld/turtlebot/

2. Create a copy of goforward.py.

cp goforward.py goincircles.py

3. Edit goincircles.py.

emacs -nw goincircles.py

Launching another Script

1. Change directory.

cd ~/helloworld/turtlebot/

2. Launch script.

python draw_a_square.py

Going Forward and Avoiding Obstacles Using Code

Going to a Specific Location on Your Map Using Code

3.16 Αυτόνομη Πλοήγηση

3.16.1 Φτιάχνοντας το Χάρτη Προσομοίωσης

Αρχικά κατασκευάζουμε ένα φάκελο για να αποθηκεύσουμε τους χάρτες μας, όλες οι εντολές του κώδικα εκτελούνται σε τερματικά που δέχονται γραμμές εντολών.

Mkdir ~/turlebot_custom_maps

Έπειτα εκτελούμε Launch Gazebo World για να τρέξουμε σε περιβάλλον προσομοίωσης Gazebo το οποίο είναι αποθηκευμένο στη βάση δεδομένων που έχουμε κατεβάσει με την εγκατάσταση της πλατφόρμας Ros Indigo από το οποίο αντλούμε δεδομένα.

Roslaunch turtlebot_gazebo turtlebot_world.launch

Με την παρακάτω εντολή το turlebot κάνει χαρτογράφηση του χάρτη.

Roslaucnh turltebot_gazebo gmapping_demo.launch

Χρησιμοποιούμε την παρακάτω εντολή ώστε να οραματιστούμε στο νέο παράθυρο εργασίας που μας δημιουργείται ένα περιβάλλον που μας δείχνει πως βλέπει το ρομποτάκι μας το περιβάλλον προσομοίωσης και να έχουμε εικόνα του περιβάλλοντος της χαρτογράφησης που κάνει Tutlebot.

Roslaucnh turltebot_rviz_launchersview_navigation.launch

Επίσης μας δίνεται η δυνατότητα μέσα από αυτή την εφαρμογή να ρυθμίσουμε διάφορες παραμέτρους στο χελωνάκι όπως είναι η θέση αναφοράς του και η ενεργοποίηση και απενεργοποίηση διάφορων αισθητήρων.

roslaunch turtlebot_teleop keyboard_teleop.launch
ΣΗΜΕΙΩΣΗ: Εάν θέλετε να χρησιμοποιήσετε άλλα εργαλεία, για παράδειγμα διαδραστικούς δείκτες, βρείτε τις πληροφορίες εδώ.Οδηγήστε το TurtleBot γύρω.ΣΗΜΕΙΩΣΗ: Το τερματικό με εκίνηση της προσομοίωσης πρέπει να είναι ενεργό όλη την ώρα διαφορετικά δεν θα μπορείτε να χειρίζεστε το TurtleBot Αυτή είναι μια εικόνα στροφής 360 μοιρών:

3.16.2 Δοκιμή Χάρτη Προσομοίωσης που έχει αποθηκευτεί μετά την αναγνώριση του.

Roslaunchturtlebot_gazeboamcl_demo.launch
map_file:=/home/jim/turtlebot_custom_maps/tutorial.yaml

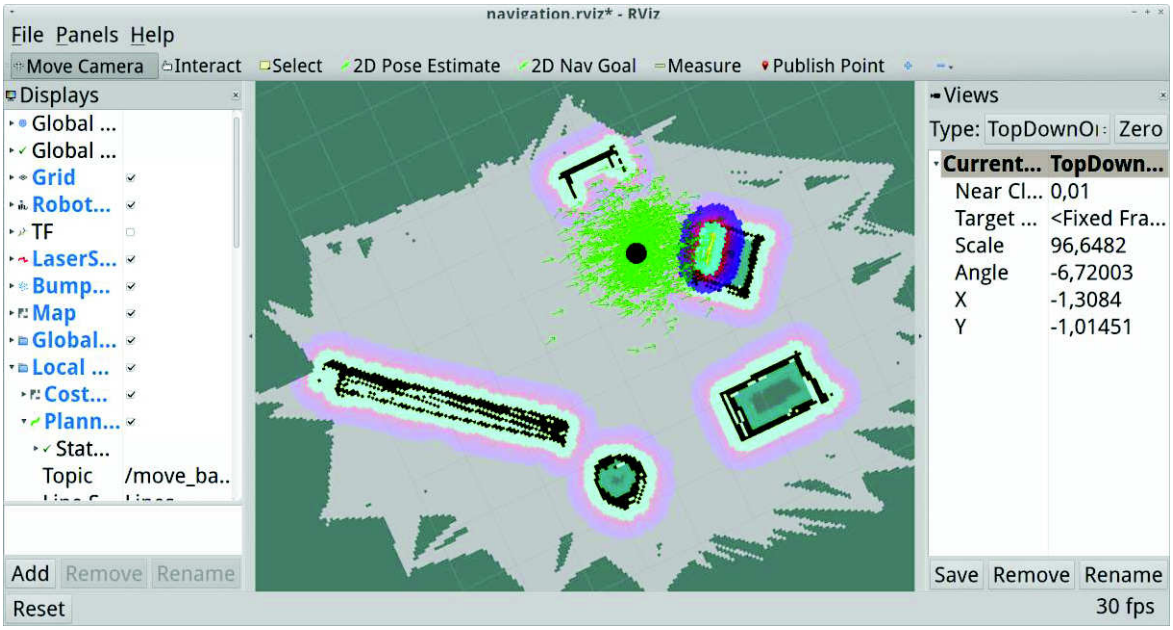
Στη περίπτωση μας κάνουμε χρήση της αποθηκευμένη προσομοίωση που υπάρχει σαν demo στο λογισμικό Ros και σε συνεργασία με τη πλατφόρμα gazebo

roslaunch turtlebot_gazebo amcl_demo.launch

3. Τρέχουμε την παρακάτω εντολή για να μας εμφανίσει το περιβάλλον Rviz.

roslaunch turtlebot_rviz_launchers view_navigation.launch

4. Αν φωτογραφία μας είναι όπως η παρακάτω μπορούμε να κάνουμε αυτόνομη πλοήγηση δίνοντας τις συντεταγμένες στο turlebot και αυτό να φτιάχνει το δρόμο του και να αποφεύγει τα εμπόδια του περιβάλλοντος εξομοίωσης.



Εικόνα 24 Πλοήγηση σε περιβάλλον αναγνωρισμένης προσομοίωσης

Τέλος κλείνουμε το τερματικό

Κεφάλαιο 4

Μεθοδολογία Έρευνας 2ης Διερεύνησης Πλοήγησης Ρομποτικού Συστήματος

4.1 Εγκατάσταση Λογισμικού

Για την υλοποίηση της συγκεκριμένης διερεύνησης χρειάστηκε η εγκατάσταση λογισμικού Window 10 στη περίπτωση αυτή το λογισμικό διατίθεται δωρεάν από τη διαδικτυακή πλατφόρμα της Microsoft που παρέχει ελεύθερο λογισμικό περιορισμένης χρήσης στους φοιτητές του Ιδρύματος. Επιπρόσθετα εγκαταστάστήσαμε το Matlab και την εκπαιδευτική έκδοση του λογισμικού Vrep τα οποία ήταν βάση για τη λειτουργίας της 2ης διερεύνησης πλοήγησης ρομποτικού συστήματος, με σκοπό τη πλοήγηση ρομποτικού συστήματος (kuka) στο χώρο.

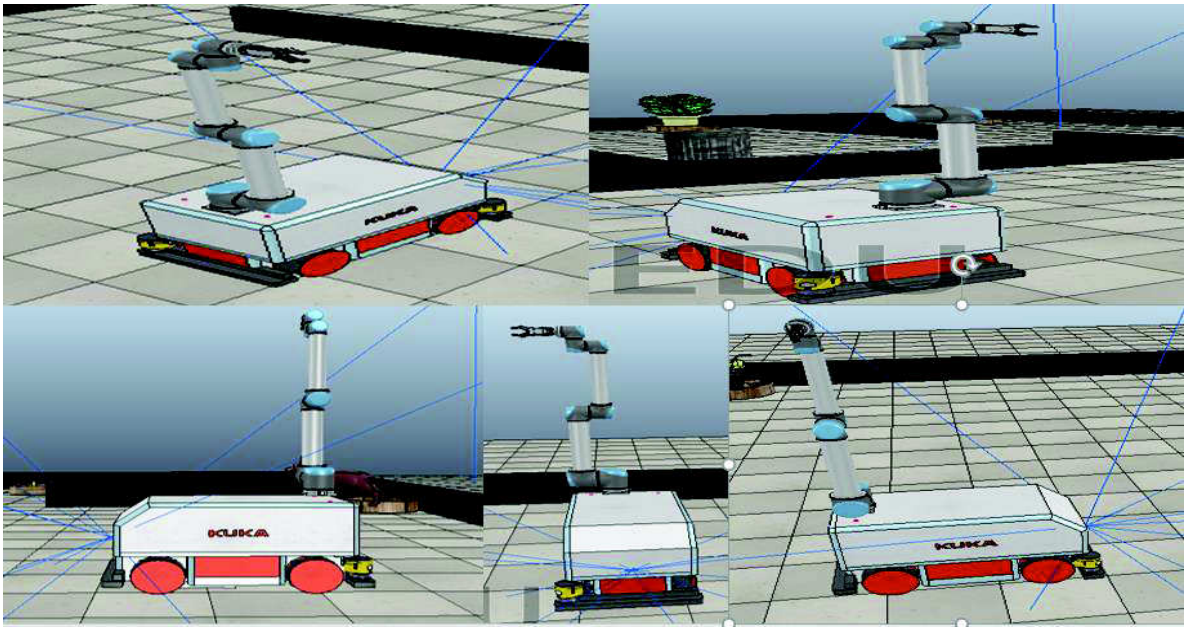
4.2 Τι είναι το Kuka

Η KUKA(KUKA Robotics) είναι Γερμανική εταιρεία που κατασκευάζει βιομηχανικά ρομπότ που πωλούνται σε βιομηχανίες όλων των ειδών, από αυτοκινητοβιομηχανίες και μεταλλοβιομηχανίες μέχρι τροφοβιομηχανίες και πλαστικοβιομηχανίες. Το όνομα KUKA είναι ακρωνύμιο και σημαίνει Κέλερ και Κνάπιχ Ώσμπουργκ (Keller and Knappich Augsburg) και είναι προστατευόμενο εμπορικό σήμα βιομηχανικών ρομπότ και άλλων προϊόντων. Η εταιρία ιδρύθηκε ο 1898 και το 1995 χωρίστηκε σε KUKA Ρομπότ (KUKA Robotics Corporation) και σε KUKA Συστήματα (KUKA Schweißanlagen GmbH, που αργότερα μετονομάστηκε σε KUKA Systems GmbH).

Εικόνα 25 Kuka σε Φωρογραφία

Ο ρομποτικός γίγαντας κάνει μια σοβαρή προσπάθεια χειραγώγησης με το ρομπότ της εφαρμογής OmniRob (φωτογραφίες και βίντεο παρακάτω). Αυτό το ρομπότ κινείται με τη βοήθεια μιας πλατφόρμας κινητής τηλεκατεύθυνσης που βασίζεται σε τροχούς mecanum, αποτελείται από ένα ελαφρύ βραχίονα Kuka και αυτό που φαίνεται να είναι διπλός ανιχνευτής εύρους λέιζερ SICK LMS100 για κάλυψη 360 ° lidar.

Στη συγκεκριμένη διερευνητική πλοήγηση ρομποτικού συστήματος γίνεται παραλλαγή του συγκεκριμένου ρομπότ όπως φαίνεται στη παρακάτω φωτογραφία.



Το «KUKA» μπορεί να χειριστεί όραση, επικοινωνία, κίνηση και την τοπικοποίηση του. Έχει τη δυνατότητα μεταφοράς αντικειμένων, μέσω της αρπάγης η οποία είναι τοποθετημένα πάνω στο βραχίονα, οπουδήποτε χρειάζεται να πάνε αποφεύγοντας αντικείμενα που μπορεί να υπάρχουν στο πέρασμα του. Αυτό μπορεί να μην είναι από μόνο του κάτι εκπληκτικό , αλλά η τεχνολογία αυτή χρησιμοποιείται από μια μεγάλη γκάμα εταιριών για την υλοποίηση ρομποτικών άκρων.

Η KUKA (KUKA Robotics) είναι Γερμανική εταιρεία που κατασκευάζει βιομηχανικά ρομπότ που πωλούνται σε βιομηχανίες όλων των ειδών, από αυτοκινητοβιομηχανίες και μεταλλοβιομηχανίες μέχρι τροφοβιομηχανίες και πλαστικοβιομηχανίες. Η εταιρεία KUKA έχει πάνω από 20 θυγατρικές επιχειρήσεις σε όλο τον κόσμο, συμπεριλαμβανομένων των ΗΠΑ, του Καναδά, του Μεξικό, της Βραζιλίας, της Ιαπωνίας, της Νότιας Κορέας, της Ταιβάν, της Ινδίας και σχεδόν όλων των χωρών της Ευρώπης.

4.3Δημιουργία Χάρτη Προσομοίωσης μέσω της πλατφόρμας Vrep

Ανοίγουμε έναν άδειο χάρτη προσομοίωσης στον οποίο διαλέγουμε σαν Mobile το Kuka.

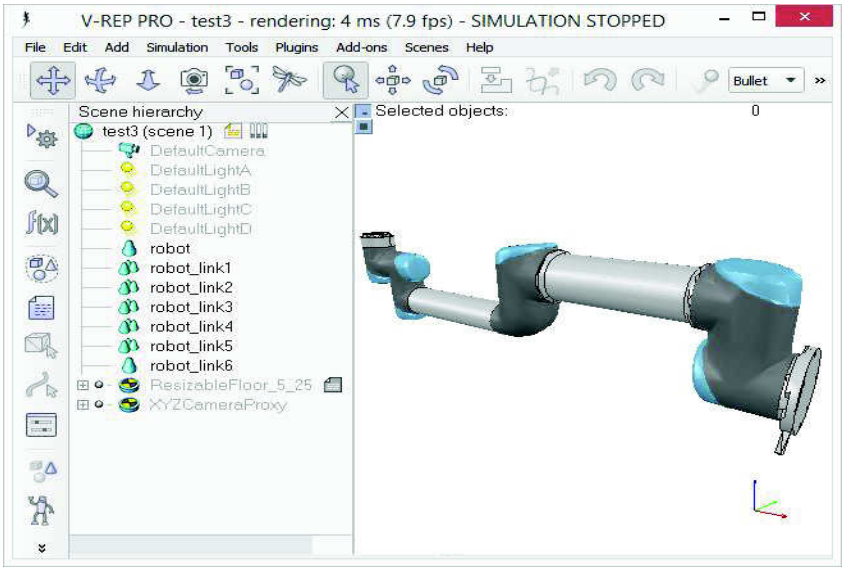
File ->New Scene

Στο επόμενο βήμα, μπορούμε να συγχωνεύσουμε στοιχεία που λογικά ανήκουν μαζί (αν είναι μέρος του ίδιου άκαμπτου στοιχείου και αν έχουν τα ίδια οπτικά χαρακτηριστικά). Στη συνέχεια αλλάζουμε τα οπτικά χαρακτηριστικά των διαφόρων στοιχείων. Το πιο εύκολο είναι να προσαρμόσουμε μερικά σχήματα που έχουν διαφορετικά χρώματα και οπτικά χαρακτηριστικά και εάν ονομάσουμε το χρώμα με μια συγκεκριμένη συμβολοσειρά, μπορούμε αργότερα να αλλάξουμε με προγραμματισμό αυτό το χρώμα, επίσης αν το σχήμα είναι μέρος σύνθετου σχήματος. Στη συνέχεια, επιλέγουμε όλα τα σχήματα που έχουν τα ίδια οπτικά χαρακτηριστικά, έπειτα ελέγχουμε, επιλέγουμε το σχήμα που είχε ήδη ρυθμιστεί και, στη συνέχεια, κάνουμε κλικ στην επιλογή Εφαρμογή, στην επιλογή μία φορά για τα χρώματα, μία φορά για τις άλλες ιδιότητες, στις ιδιότητες σχήματος: οπτικά χαρακτηριστικά στα επιλεγμένα σχήματα (συμπεριλαμβανομένου του ονόματος χρώματος, αν το έχετε παράσχει). Καταλήγουμε με 17 μεμονωμένα σχήματα:

Εικόνα 26 Βραχίονας σε περιβάλλον VREP

Τώρα μπορούμε να ομαδοποιήσουμε τα σχήματα που ανήκουν στον ίδιο σύνδεσμο με το [Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / συγχώνευση -> Ομαδοποίηση επιλεγμένων σχημάτων]. Καταλήγουμε με 7 σχήματα: τη βάση του ρομπότ (ή τη βάση του δέντρου ιεραρχίας του ρομπότ) και 6 κινητές συνδέσεις. Είναι επίσης σημαντικό να ονομάσουνε σωστά τα αντικείμενά σας: το κάνουμε κάνοντας διπλό κλικ στο όνομα του αντικειμένου στην ιεραρχία σκηνών. Η βάση πρέπει

να είναι πάντα το όνομα του ρομπότ ή του μοντέλου και τα άλλα αντικείμενα πρέπει πάντα να περιέχουν το όνομα του αντικειμένου βάσης, όπως: Robot (base), robot_link1, robot_proximitySensor κ.λπ. Με defaut, τα σχήματα θα αντιστοιχούν στο επίπεδο ορατότητας 1, να αλλάξουμε στις κοινές ιδιότητες του αντικειμένου. Από προεπιλογή, μόνο τα επίπεδα ορατότητας 1-8 ενεργοποιούνται για τη σκηνή. Τώρα έχουμε (το μοντέλο ResizableFloor_5_25 έγινε προσωρινά αόρατο στο παράθυρο διαλόγου ιδιοτήτων μοντέλου): μόνο τα επίπεδα ορατότητας 1-8 ενεργοποιούνται για το σκηνικό. Τώρα έχουμε (το μοντέλο ResizableFloor_5_25 έγινε προσωρινά αόρατο στο μοντέλο διαλόγου διαλόγου παραθύρου): ayer 1, αλλά μπορεί να αλλάξει στις κοινές ιδιότητες του αντικειμένου. Από προεπιλογή, μόνο τα επίπεδα ορατότητας 1-8 ενεργοποιούνται για τη σκηνή. Τώρα έχουμε (το μοντέλο ResizableFloor_5_25 έγινε προσωρινά αόρατο στο παράθυρο διαλόγου ιδιοτήτων μοντέλου):

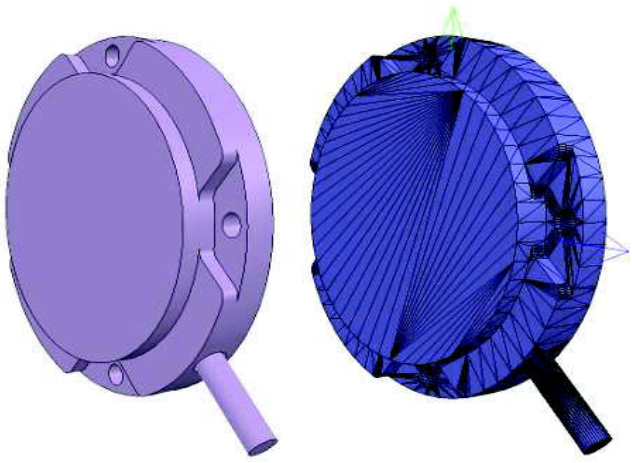


Εικόνα 27 Βραχιονας προς τροποπολιηση

Όταν ένα σχήμα που δημιουργούμε ή τροποποιούμε, το V-REP θα ρυθμίσει αυτόματα τη θέση του πλαισίου αναφοράς και τον προσανατολισμό του. Το πλαίσιο αναφοράς ενός σχήματος θα τοποθετούμε πάντα στο γεωμετρικό κέντρο του σχήματος. Ο προσανατολισμός του πλαισίου θα επιλέξουμε έτσι ώστε το πλαίσιο οριοθέτησης του σχήματος να παραμένει όσο το δυνατόν μικρότερο. Αυτό δεν φαίνεται πάντα ωραίο, αλλά πάντα μπορούμε να αναπροσανατολίζουμε το πλαίσιο αναφοράς ενός σχήματος ανά πάσα στιγμή. Τώρα αναπροσανατολίζουμε τα πλαίσια αναφοράς όλων των σχημάτων που έχουμε δημιουργήσει με [Γραμμή μενού -> Επεξεργασία -> Αναστροφή πλαισίου οριοθέτησης -> με πλαίσιο αναφοράς του κόσμου]. Έχουμε περισσότερες επιλογές για αναπροσανατολισμό ενός πλαισίου αναφοράς στο διάλογο γεωμετρίας σχήματος.

4.4 Εγκατάσταση αρθρώσεων και κινητήρων

Τώρα θα φροντίσουμε για τις αρθρώσεις / κινητήρες. Τις περισσότερες φορές, γνωρίζουμε την ακριβή θέση και τον προσανατολισμό καθεμιάς από τις αρθρώσεις. Σε αυτή την περίπτωση, προσθέτουμε απλώς τις αρθρώσεις με [Γραμμή μενού -> Προσθήκη -> Συνδέσεις -> ...], τότε μπορούμε να αλλάξουμε τη θέση και τον προσανατολισμό τους με το παράθυρο διαλόγου θέσης και προσανατολισμού. Σε άλλες περιπτώσεις, έχουμε μόνο τις παραμέτρους Denavit-Hartenberg (δηλ. D-H). Σε αυτή την περίπτωση, μπορούμε να δημιουργήσουμε τις αρθρώσεις μας μέσω του μοντέλου εργαλείων που βρίσκεται στο μοντέλο / tools / Denavit-Hartenberg κοινό creator.ttm, στο πρόγραμμα περιήγησης μοντέλου. Άλλες φορές, δεν έχουμε πληροφορίες σχετικά με τις κοινές τοποθεσίες και προσανατολισμούς. Στη συνέχεια, πρέπει να τα εξαγάγουμε από το εισαγόμενο πλέγμα. Ας υποθέσουμε ότι αυτή είναι η περίπτωσή μας. Αντί να επεξεργαστούμε το τροποποιημένο, περισσότερο προσεγγιστικό πλέγμα, ανοίγουμε μια νέα σκηνή και εισάγουμε ξανά τα αρχικά δεδομένα CAD. Τις περισσότερες φορές, μπορούμε να εξαγάγουμε τα μάτια ή τα αρχικά σχήματα από το αρχικό πλέγμα. Το πρώτο βήμα είναι να υποδιαιρέσουμε το αρχικό πλέγμα. Εάν αυτό δεν λειτουργεί, το κάνουμε μέσω της λειτουργίας επεξεργασίας τριγώνου. Ας υποθέσουμε ότι μπορούμε να διαιρέσουμε το αρχικό πλέγμα. Τώρα έχουμε μικρότερα αντικείμενα που μπορούμε να επιθεωρήσουμε. Ψάχνουμε για περιστροφικά σχήματα, τα οποία θα μπορούσαν να χρησιμοποιηθούν ως αναφορά για τη δημιουργία αρθρώσεων στις θέσεις τους, με τον ίδιο προσανατολισμό. Καταρχάς, αφαιρούμε όλα τα αντικείμενα που δεν χρειάζονται. Μερικές φορές είναι επίσης χρήσιμο να εργαζόμαστε σε διαφορετικές ανοιχτές σκηνές, για ευκολότερη απεικόνιση / χειραγώγηση. Στην περίπτωσή μας, εστιάζουμε πρώτα στη βάση του ρομπότ: περιέχει έναν κύλινδρο που έχει τη σωστή θέση για την πρώτη άρθρωση. Στη λειτουργία επεξεργασίας τριγώνου, έχουμε:



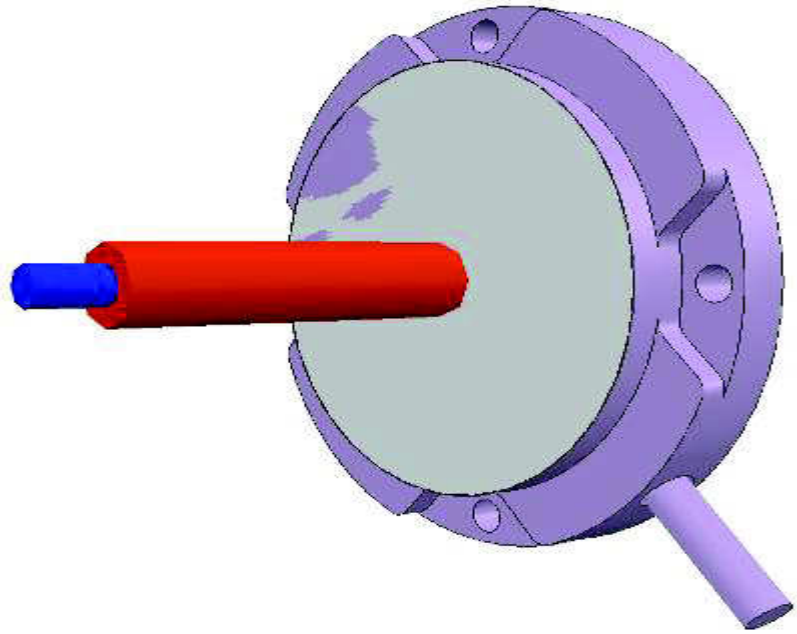
Εικόνα 28 Σημαντικές Λεπτομέρειες για Διαμπρφωση Σχηματος

Αλλάζουμε την προβολή της κάμερας μέσω του κουμπιού της γραμμής εργαλείων επιλογής σελίδας, για να δούμε το αντικείμενο από το πλάι. Το κουμπί της γραμμής εργαλείων εμφάνισης μπορεί να είναι χρήσιμο για να πλαισιώσει σωστά το αντικείμενο σε έκδοση. Στη συνέχεια, μεταβαίνουμε στη λειτουργία επεξεργασίας κορυφών και επιλέγουμε όλες τις κορυφές που ανήκουν στο ανώτερο δίσκο. Θυμημόμαστε ότι με την ενεργοποίηση / απενεργοποίηση ορισμένων στρωμάτων μπορούμε να αποκρύψουμε άλλα αντικείμενα στο σκηνικό. Κατόπιν, γυρίστε πίσω στην λειτουργία επεξεργασίας τριγώνου:

Εικόνα 29 Παραμετροποιήση Κυλίνδρου

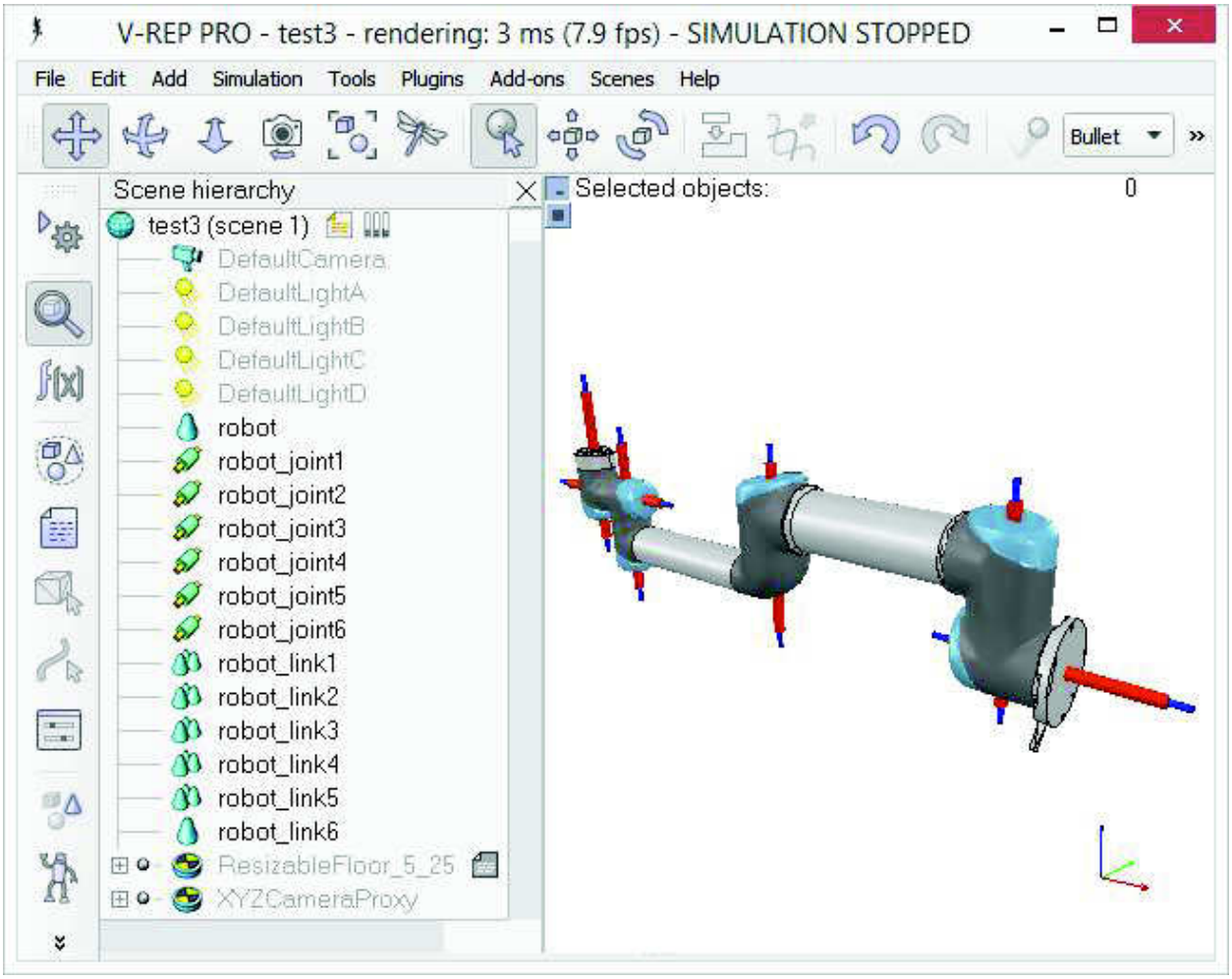
Τώρα κάνουμε κλικ στον κύλινδρο εξαγωγής (το σχήμα εξαγωγής θα λειτουργούσε επίσης σε αυτή την περίπτωση), αυτό ακριβώς δημιούργησαμε ένα σχήμα κυλίνδρου στη σκηνή, με βάση τα επιλεγμένα τρίγωνα. Αφήνουμε τη λειτουργία επεξεργασίας και απορρίπτουμε τις αλλαγές. Τώρα προσθέτουμε μια περιστρεφόμενη άρθρωση με τη [Γραμμή μενού -> Προσθέστε -> Κοινή -> Επανάσταση], κρατήστε την επιλεγμένη και έπειτα ελέγουμε το σχήμα του κυλίνδρου. Στο παράθυρο διαλόγου "Θέση", στην καρτέλα "Θέση", πατάμε Εφαρμογή στην επιλογή: αυτό με το οποίο βασικά αντιγράφουμε τη θέση x / y / z του κυλίνδρου στον σύνδεσμο. Και οι δύο θέσεις είναι ταυτόσημες. Στο διάλογο προσανατολισμού, στην καρτέλα προσανατολισμού, κάνουμε κλικ στην επιλογή Εφαρμογή στην επιλογή: ο προσανατολισμός των επιλεγμένων αντικειμένων είναι τώρα ο ίδιος. Μερικές φορές, θα χρειαστούμε να περιστρέψουμε επιπλέον την άρθρωση περίπου 90/180 μοίρες γύρω από το δικό της πλαίσιο αναφοράς για να λάβουμε τη σωστή κατεύθυνση

προσανατολισμού ή περιστροφής. Θα μπορούσαμε να το κάνουμε αυτό στην καρτέλα περιστροφής αυτού του διαλόγου, εάν χρειαζόταν (σε αυτή την περίπτωση, μην ξεχάσουμε να κάνουμε κλικ στο κουμπί "Κορνίζα"). Με παρόμοιο τρόπο θα μπορούσαμε επίσης να μετατοπίσουμε την άρθρωση κατά μήκος του άξονα της, ή ακόμα και να κάνουμε πιο πολύπλοκες λειτουργίες. Αυτό είναι αυτό που έχουμε:



Εικόνα 30 Διαμόρφωση Αρθρωσης

Τώρα αντιγράφουμε την άρθρωση στην αρχική σκηνή και την αποθηκεύουμε. (δεν ξεχνάμε να αποθηκεύσουμε την εργασία μας σε τακτική βάση! Η λειτουργία αναίρεσης είναι χρήσιμη, αλλά δεν μας προστατεύει από άλλες ατυχίες). Επαναλαμβάνουμε την παραπάνω διαδικασία για όλες τις αρθρώσεις στο ρομπότ μας και στη συνέχεια μετονομάζουμε. Κάνουμε επίσης όλες τις αρθρώσεις λίγο περισσότερο στις ιδιότητες άρθρωσης, για να τις δούμε όλες. Με την αποπληρωμή, οι αρθρώσεις θα εκχωρηθούν στο επίπεδο ορατότητας 2, αλλά μπορούν να αλλαχθούν στις κοινές ιδιότητες του αντικειμένου. Αναθέτουμε τώρα όλες τις αρθρώσεις στο επίπεδο ορατότητας 10 και, στη συνέχεια, ενεργοποιούμε προσωρινά τη στρώση ορατότητας 10 για τη σκηνή, ώστε να απεικονίσει επίσης εκείνες τις αρθρώσεις (από προεπιλογή, ενεργοποιούνται μόνο τα στρώματα ορατότητας 1-8 για τη σκηνή). Αυτό το έχουμε (το μοντέλο ResizableFloor_5_25 έγινε προσωρινά αόρατο στο παράθυρο διαλόγου ιδιοτήτων μοντέλου):



Εικόνα 31 Σύνδεσμοι πάνω στο βραχίονα

αυτό το σημείο, θα μπορούσαμε να αρχίσουμε να κατασκευάζουμε την ιεραρχία μοντέλου και να τελειώνουμε τον ορισμό του μοντέλου. Αλλά αν θέλουμε το δυναμικό ρομπότ να ενεργοποιηθεί δυναμικά, τότε υπάρχει ένα επιπλέον ενδιάμεσο βήμα:

Δημιουργία Δυναμικών Αντικειμένων

Αν θέλουμε το ρομπότ μας να είναι δυναμικά ενεργοποιημένο, δηλ. Να αντιδρά σε συγκρούσεις, πτώση κ.λπ., τότε πρέπει να δημιουργήσουμε / διαμορφώσουμε τα σχήματα κατάλληλα: ένα σχήμα μπορεί να είναι:δυναμική ή στατική: ένα δυναμικό (ή μη στατικό) σχήμα θα πέσει και θα επηρεαστεί από εξωτερικές δυνάμεις / ροπές. Μια στατική (ή μη δυναμική) μορφή από την άλλη πλευρά, θα παραμείνει στη θέση της ή θα ακολουθήσει την κίνηση του γονέα της στην ιεραρχία σκηνών. Ανταπόκριση ή μη ανταπόκριση: ένα ανταποκρινόμενο σχήμα θα προκαλέσει αντίδραση σύγκρουσης με άλλα απαντήσιμα σχήματα. Αυτοί (και / ή) ο επιταχυντής τους, θα επηρεαστούν στην κίνηση τους εάν είναι δυναμικοί. Από την άλλη πλευρά, τα μη απαντήσιμα σχήματα δεν υπολογίζουν την απόκριση σε σύγκρουση αν συγκρούονται με άλλα σχήματα.

Πάνω από δύο σημεία παρουσιάζονται εδώ. Τα απαντήσιμα σχήματα πρέπει να είναι όσο το δυνατόν απλούστερα, ώστε να επιτρέπουν μια γρήγορη και σταθερή προσομοίωση. Ένας κινητήρας φυσικής θα είναι σε θέση να προσομοιώσει πέντε τύπους σχημάτων με διάφορους βαθμούς ταχύτητας και σταθερότητας:

Καθαρά σχήματα: ένα καθαρό σχήμα θα είναι σταθερό και θα χειρίζεται πολύ αποτελεσματικά από τη μηχανή φυσικής. Η επιστροφή είναι ότι τα καθαρά σχήματα είναι περιορισμένα στη γεωμετρία: κυρίως κυβοειδή, κύλινδροι και σφαίρες. Εάν είναι δυνατόν, χρησιμοποιήσουμε τα αντικείμενα που είναι σε επαφή με άλλα αντικείμενα για μεγαλύτερο χρονικό διάστημα (π.χ. τα πόδια ενός ανθρωποειδούς ρομπότ, της βάσης ενός σειριακού χειριστή, των δακτύλων μιας λαβής κ.λπ.). Μπορούν να δημιουργηθούν καθαρά σχήματα με τη [Γραμμή μενού -> Προσθήκη -> Πρωτότυπο σχήμα].

Καθαρά σύνθετα σχήματα: ένα καθαρό σύνθετο σχήμα είναι μια ομάδα πολλών καθαρών σχημάτων. Εκτελεί σχεδόν τόσο καθαρά σχήματα και μοιράζεται παρόμοιες ιδιότητες. Τα καθαρά σύνθετα σχήματα μπορούν να δημιουργηθούν συγκεντρώνοντας αρκετά καθαρά σχήματα [Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Ομαδοποίηση επιλεγμένων σχημάτων].

Κυρτά σχήματα: ένα κυρτό σχήμα θα είναι λίγο λιγότερο σταθερό και θα πάρει λίγο περισσότερο χρόνο υπολογισμού όταν χειρίζομαστε το κινητήρας φυσικής. Επιτρέπει μια γενικότερη γεωμετρία (μόνο απαίτηση: πρέπει να είναι κυρτή) από τα καθαρά σχήματα. Εάν είναι δυνατόν, χρησιμοποιούμε κυρτά σχήματα για αντικείμενα που σποραδικά έρχονται σε επαφή με άλλα αντικείμενα (π.χ. τους διάφορους συνδέσμους ενός ρομπότ). Μπορούν να δημιουργηθούν κυρτά σχήματα με [Γραμμή μενού -> Προσθήκη -> Κυρτό κύτος επιλογής] ή με [Γραμμή μενού -> Επεξεργασία -> Επιλογή μορφοποίησης σε κυρτά σχήματα].

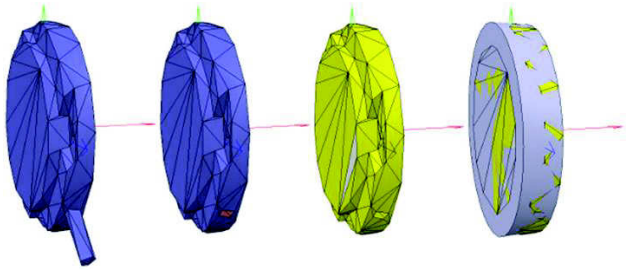
Σύνθετα κυρτά σχήματα ή κυρτά αποσυντιθέμενα σχήματα: ένα κυρτό αποσυντιθέμενο σχήμα είναι μια ομάδα διαφόρων κυρτών σχημάτων. Εκτελεί σχεδόν εξίσου κυρτά σχήματα και μοιράζεται παρόμοιες ιδιότητες. Μπορούμε να δημιουργήσουμε κυρτά αποσυνδεδεμένα σχήματα με την ομαδοποίηση αρκετών κυρτών σχημάτων [Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Ομαδοποίηση επιλεγμένων σχημάτων], με [Γραμμή μενού -> Προσθήκη -> Κυρτή αποσύνθεση επιλογής ...] ή με [Γραμμή μενού -> Επεξεργασία -> Επιλογή Morph στην κυρτή αποσύνθεση ...].

Τυχαία σχήματα: ένα τυχαίο σχήμα είναι ένα σχήμα που δεν είναι κυρτό ούτε καθαρό. Έχει γενικά χαμηλές επιδόσεις (ταχύτητα υπολογισμού και σταθερότητα). Αποφεύγουμε τη χρήση τυχαίων σχημάτων όσο το δυνατόν περισσότερο.

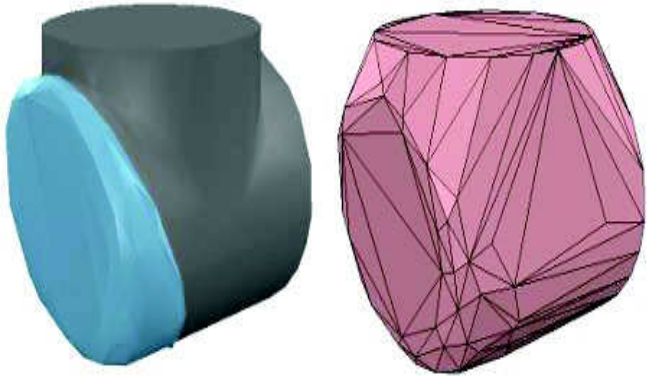
Έτσι, η τάξη προτιμήσεων θα είναι: καθαρά σχήματα, καθαρά σύνθετα σχήματα, κυρτά σχήματα, σύνθετα κυρτά σχήματα και τελικά τυχαία σχήματα. Στην περίπτωση του ρομπότ που θέλουμε να οικοδομήσουμε, θα κάνουμε τη βάση του ρομπότ ως καθαρό κύλινδρο, και οι άλλες συνδέσεις σαν κυρτά ή κυρτά αποσυνδεδεμένα σχήματα.

Θα μπορούσαμε να χρησιμοποιήσουμε τα δυναμικά ενεργοποιημένα σχήματα και ως ορατά μέρη του ρομπότ, αλλά αυτό μάλλον δεν θα φαινόταν αρκετά καλό. Αντ 'αυτού, θα δημιουργήσουμε για κάθε ορατό σχήμα που έχουμε δημιουργήσει στο πρώτο μέρος του tutorial ένα δυναμικά ενεργοποιημένο αντίστοιχο, το οποίο θα κρατήσουμε κρυμμένο: το κρυφό τμήμα θα αντιπροσωπεύει το δυναμικό μοντέλο και θα χρησιμοποιηθεί αποκλειστικά από τον κινητήρα φυσικής, ενώ το ορατό μέρος θα χρησιμοποιηθεί για οπτικοποίηση, αλλά και για υπολογισμούς ελάχιστης απόστασης, ανιχνεύσεις αισθητήρων εγγύτητας κλπ.

Επιλέγουμε το ρομπότ αντικειμένου, κάνουμε αντιγραφή και επικόλληση το σε μια νέα σκηνή (για να διατηρήσουμε το αρχικό μοντέλο ανέπαφο) και ξεκινάμε τη λειτουργία επεξεργασίας τριγώνου. Εάν το ρομπότ αντικειμένων ήταν σύνθετο σχήμα, θα έπρεπε πρώτα να ανακεφαλαιώσουμε ([Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Ομαδοποιημένα σχήματα]) και στη συνέχεια να συγχωνεύσουμε τα μεμονωμένα σχήματα ([Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Συγχώνευση επιλεγμένων σχημάτων]) προτού μπορέσουμε να ξεκινήσουμε τη λειτουργία επεξεργασίας τριγώνου. Τώρα επιλέγουμε τα λίγα τρίγωνα που αντιπροσωπεύουν το καλώδιο τροφοδοσίας και τα διαγράψουμε. Στη συνέχεια, επιλέγουμε όλα τα τρίγωνα σε αυτό το σχήμα και κάντε κλικ στην επιλογή Εξαγωγή κυλίνδρου. Τώρα μπορούμε να αφήσουμε τη λειτουργία επεξεργασίας και έχουμε το βασικό μας αντικείμενο να αντιπροσωπεύεται ως καθαρός κύλινδρος:



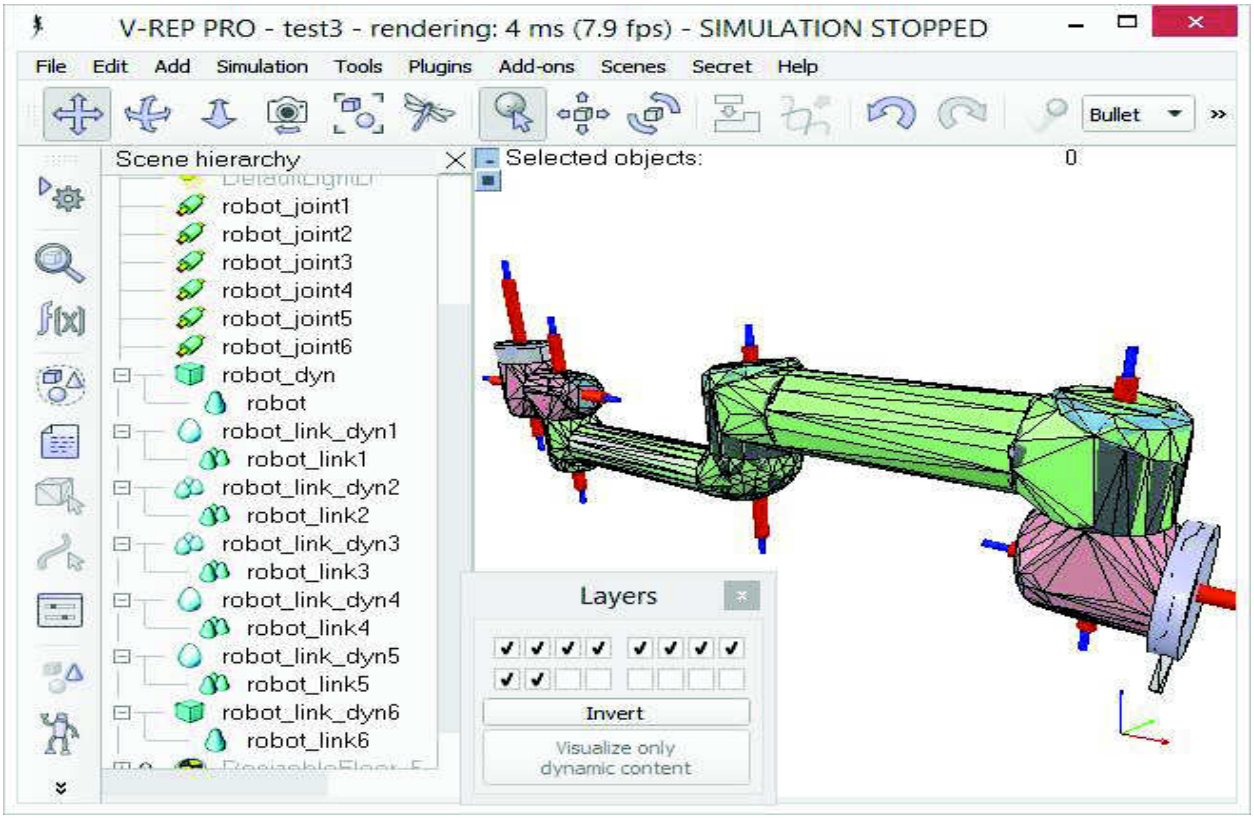
Εικόνα 32 Η διαδικασία διαμόρφωσης και λειτουργίας του τον βραχίωνα omnirob του περιβάλλον προσομοίωσης



Εικόνα 33 Σύνδεσμος Βραχίωνα Omnirob

Αλλάζουμε το νέο σχήμα (με διπλό κλικ στο όνομα του στην ιεραρχία σκηνών) ως robot_dyn, το αντιστοιχίζουμε στο επίπεδο ορατότητας 9 και στη συνέχεια το αντιγράψουμε στην αρχική σκηνή. Οι υπόλοιποι σύνδεσμοι θα διαμορφωθούν ως κυρτά σχήματα ή σύνθετα κυρτά σχήματα. Τώρα επιλέγουμε τον πρώτο κινητό σύνδεσμο (δηλ. Το αντικείμενο robot_link1) και δημιουργούμε ένα κυρτό σχήμα από αυτό με το [Γραμμή μενού -> Προσθήκη -> Κυρτό κύτος επιλογής]. Το μετονομάζουμε σε robot_link_dyn1 και το αναθέτουμε στο επίπεδο ορατότητας 9. Όταν εξάγουμε το κυρτό κύτος δεν διατηρεί αρκετές λεπτομέρειες για το αρχικό σχήμα, τότε μπορούμε να εξαγάγουμε με το χέρι αρκετές κυρτές γάστρες από τα συνθετικά του στοιχεία και στη συνέχεια να ομαδοποιήσουμε όλα τα κυρτά κύτη με [Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Ομαδοποίηση επιλεγμένων σχημάτων]. Εάν αυτό φαίνεται να είναι προβληματικό ή χρονοβόρο, τότε μπορούμε να εξάγουμε αυτόματα ένα κυρτό αποσυντιθέμενο σχήμα με το [Γραμμή μενού -> Προσθήκη -> Κυρτή αποσύνθεση της επιλογής ...]:

Επαναλαμβάνουμε τώρα την ίδια διαδικασία για όλους τους υπόλοιπους συνδέσμους ρομπότ. Μόλις γίνει αυτό, προσαρμόζουμε κάθε ορατό σχήμα στο αντίστοιχο αόρατο δυναμικό κρεμαστό του. Κάνουμε αυτό επιλέγοντας πρώτα το ορατό σχήμα και, στη συνέχεια, επιλέγουμε με το πάτημα του κουμπιού την επιλογή της δυναμικής του κρεμαστής, στη συνέχεια [Γραμμή μενού -> Επεξεργασία -> Εκτέλεση τελευταίου επιλεγμένου αντικειμένου]. Το ίδιο αποτέλεσμα μπορεί να επιτευχθεί με τη μεταφορά του ορατού σχήματος στο δυναμικό του μενταγιόν στην ιεραρχία σκηνών:



Εικόνα 34 Λειτουργία Διαμόρφωσης για Δυναμικά σχήματα πάνω στο βραχίονα

Πρέπει ακόμα να φροντίσουμε για μερικά πράγματα: πρώτον, αφού θέλουμε τα δυναμικά σχήματα να είναι ορατά μόνο στον κινητήρα φυσικής, αλλά όχι στις άλλες υπομονάδες υπολογισμού, καταργούμε όλες τις ειδικές ιδιότητες αντικειμένων για τα δυναμικά σχήματα, στις κοινές ιδιότητες του αντικειμένου .

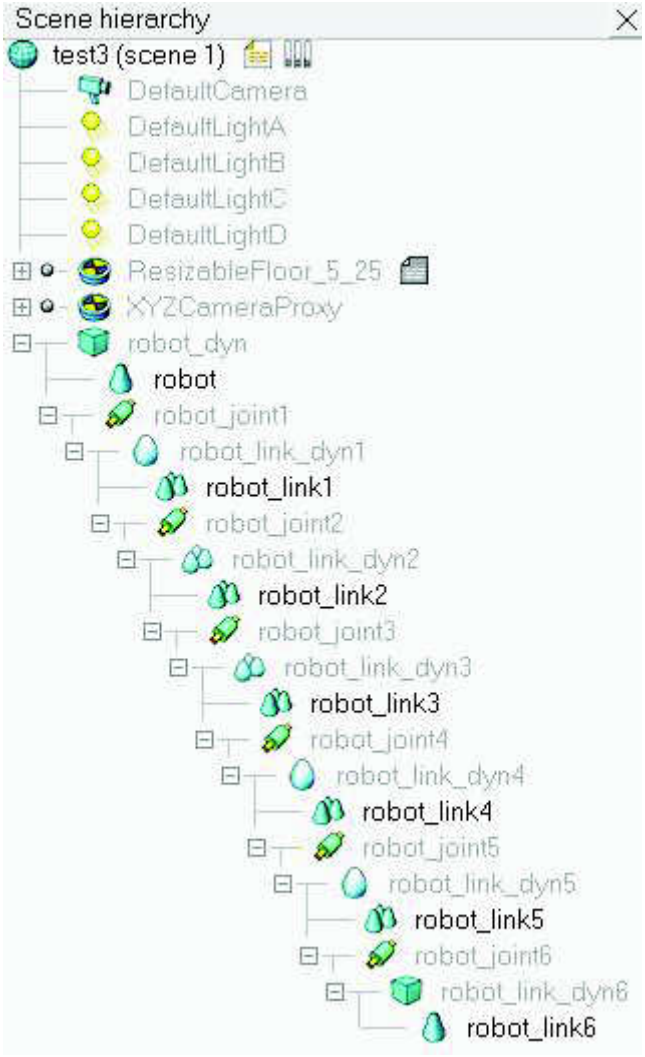
Στη συνέχεια, πρέπει να ρυθμίσουμε τα δυναμικά σχήματα δυναμικά και να ανταποκριθούμε. Αυτό το κάνουμε στις ιδιότητες δυναμικής σχήματος. Επιλέξτε πρώτα το δυναμικό σχήμα βάσης (δηλ. Το robot_dyn) και, στη συνέχεια, ελέγξτε το στοιχείο Body responsable. Ενεργοποιήσουμε τις πρώτες 4 Τοπικές αποκρινόμενες σημαίες μάσκας και απενεργοποιήσουμε τις τελευταίες 4 σημαίες τοπικής αποκρινόμενης μάσκας: είναι σημαντικό οι διαδοχικοί απαντήσιμοι σύνδεσμοι να μην συγκρούονται μεταξύ τους. Για τον πρώτο κινητό δυναμικό σύνδεσμο στο ρομπότ μας (δηλ. Robot_link_dyn1), ενεργοποιούμε επίσης το στοιχείο Body responsable, αλλά αυτή τη φορά απενεργοποιούμε τις πρώτες 4 Τοπικές αποκρινόμενες σημαίες μάσκας και ενεργοποιούμε τις τελευταίες 4 σημαίες τοπικής αποκρινόμενης μάσκας. Επαναλαμβάνουμε την παραπάνω διαδικασία με όλους τους άλλους δυναμικούς συνδέσμους, ενώ πάντα εναλλάσσουμε τις σημαίες τοπικής αποκρινόμενης μάσκας: μόλις καθοριστεί το μοντέλο, τα διαδοχικά δυναμικά σχήματα του ρομπότ δεν θα παράγουν καμία απόκριση σύγκρουσης όταν αλληλεπιδρούν μεταξύ τους. Προσπαθήστε πάντα να καταλήξετε σε μια κατασκευή όπου η δυναμική βάση του ρομπότ και ο

δυναμικός τελευταίος σύνδεσμος του ρομπότ έχουν ενεργοποιήσει μόνο τις πρώτες 4 τοπικές αποκρινόμενες σημαίες μάσκας, ώστε να μπορέσουμε να προσαρτήσουμε το ρομπότ σε μια κινητή πλατφόρμα ή να προσκολλήσουμε λαβή στον τελευταίο δυναμικό σύνδεσμο του ρομπότ χωρίς δυναμικές παρεμβολές σύγκρουσης.

Τέλος, πρέπει να επισημάνουμε τα δυναμικά μας σχήματα καθώς το σώμα είναι δυναμικό. Το κάνουμε αυτό και στις ιδιότητες δυναμικής σχήματος. Στη συνέχεια, μπορούμε να εισαγάγουμε τις ιδιότητες τάσης μάζας και αδράνειας χειροκίνητα ή να υπολογίσουμε αυτομάτως αυτές τις τιμές (συνιστάται) κάνοντας κλικ στο στοιχείο Υπολογισμός μαζών και ιδιοτήτων αδρανείας για επιλεγμένα κυρτά σχήματα. Θυμηθείτε επίσης αυτό και αυτό το δυναμικό σχεδιασμό εκτιμήσεις. Αυτή η δυναμική βάση του ρομπότ είναι μια ειδική περίπτωση: οι περισσότεροι χρόνοι θέλουμε η βάση του ρομπότ να είναι μη δυναμική (δηλ. Στατική), διαφορετικά, αν χρησιμοποιηθεί μόνο, το ρομπότ μπορεί να πέσει κατά τη διάρκεια της κίνησης. Αλλά μόλις συνδέσουμε τη βάση του ρομπότ σε μια κινητή πλατφόρμα, θέλουμε η βάση να γίνει δυναμική (δηλ. Μη στατική). Το κάνουμε αυτό επιτρέποντας στο Set να είναι δυναμικό αν παίρνει το γονικό στοιχείο και, στη συνέχεια, απενεργοποιώντας το Body είναι δυναμικό στοιχείο. Τώρα εκτελέστε την προσομοίωση: όλα τα δυναμικά σχήματα, εκτός από τη βάση του ρομπότ, θα πρέπει να πέσουν. Ότι τα συνημμένα οπτικά σχήματα θα ακολουθούν τα δυναμικά τους μενταγιόν.

Ορισμός μοντέλου

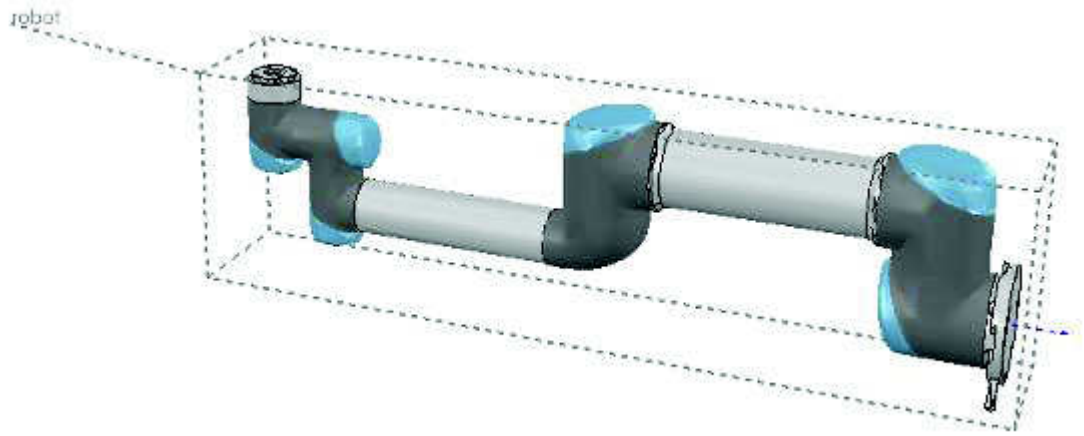
Τώρα είμαστε έτοιμοι να καθορίσουμε το μοντέλο μας. Ξεκινάμε με την οικοδόμηση του μοντέλου herarchy: προσαρμόζουμε την τελευταία δυναμική ζεύξη ρομπότ (robot_link_dyn6) στην αντίστοιχη άρθρωση (robot_joint6) επιλέγοντας robot_link_dyn6, στη συνέχεια επιλέγοντας robot_joint6, στη συνέχεια [Γραμμή μενού -> Επεξεργασία -> μητρική εταιρεία]. Θα μπορούσαμε επίσης να κάνουμε αυτό το βήμα μεταφέροντας απλά το αντικείμενο robot_link_dyn6 στο robot_link6 στην ιεραρχία σκηνών. Συνεχίζουμε τώρα να συνδέουμε το robot_joint6 με το robot_link_dyn5 και ούτω καθεξής, μέχρι να φθάσουμε στη βάση του ρομπότ. Τώρα έχουμε την ακόλουθη ιεραρχία σκηνών:



Εικόνα 35 Στήλη Ιεραχία για διαχείριση αντικείμενων

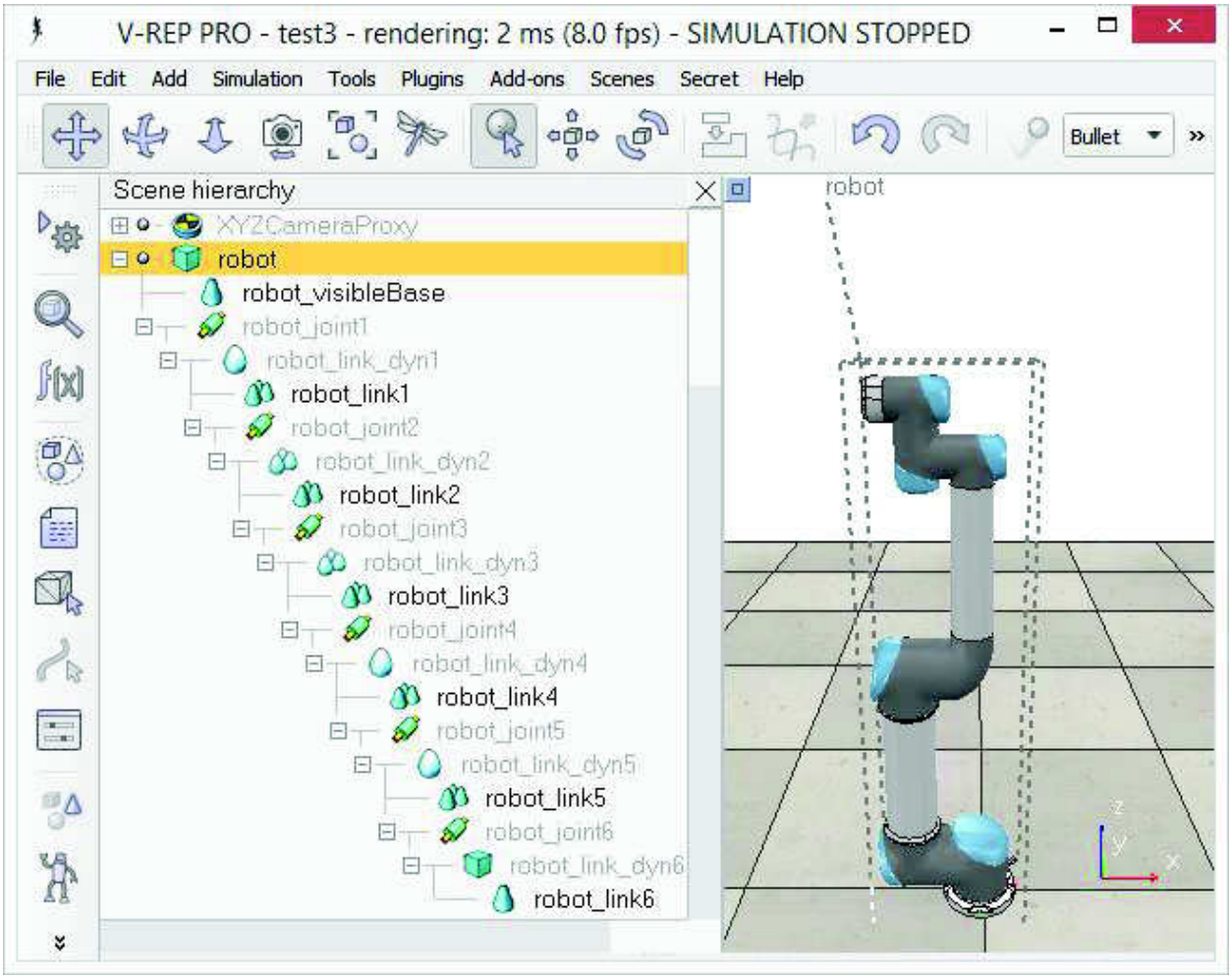
Είναι ωραίο και πιο λογικό να έχουμε ένα απλό όνομα για τη βάση του μοντέλου, καθώς η βάση του μοντέλου θα αντιπροσωπεύει επίσης το ίδιο το μοντέλο. Έτσι, μετονομάζουμε ρομπότ σε robot_visibleBase και robot_dyn σε ρομπότ. Τώρα επιλέγουμε τη βάση του δέντρου ιεραρχίας (δηλ. Το αντικείμενο ρομπότ) και στις κοινές ιδιότητες του αντικειμένου που επιτρέπουμε το αντικείμενο είναι βάση μοντέλου. Ενεργοποιούμε επίσης το Αντικείμενο / μοντέλο να μεταφέρει ή να αποδεχθεί το DNA. Εμφανίστηκε ένα πλαίσιο οριοθέτησης μοντέλου, το οποίο περιλάμβανε ολόκληρο το ρομπότ. Ωστόσο, το πλαίσιο οριοθέτησης φαίνεται να είναι πολύ μεγάλο: αυτό συμβαίνει επειδή το πλαίσιο οριοθέτησης περιλαμβάνει επίσης τα αόρατα στοιχεία, όπως οι αρθρώσεις. Τώρα αποκλείουμε τις αρθρώσεις από το πλαίσιο οριοθέτησης μοντέλου ενεργοποιώντας το στοιχείο Να μην εμφανίζεται ως εσωτερικό στοιχείο επιλογής μοντέλου για όλες τις αρθρώσεις. Θα μπορούσαμε να κάνουμε την ίδια διαδικασία για όλα τα αόρατα αντικείμενα στο μοντέλο μας. Αυτή είναι επίσης μια χρήσιμη επιλογή για να αποκλείσετε επίσης

μεγάλους αισθητήρες ή άλλα αντικείμενα από το πλαίσιο οριοθέτησης μοντέλου. Τώρα έχουμε την ακόλουθη κατάσταση:



Εικόνα 36 Διαδικάσια πριν περάσουμε στο περιβάλλον εξομοίωσης

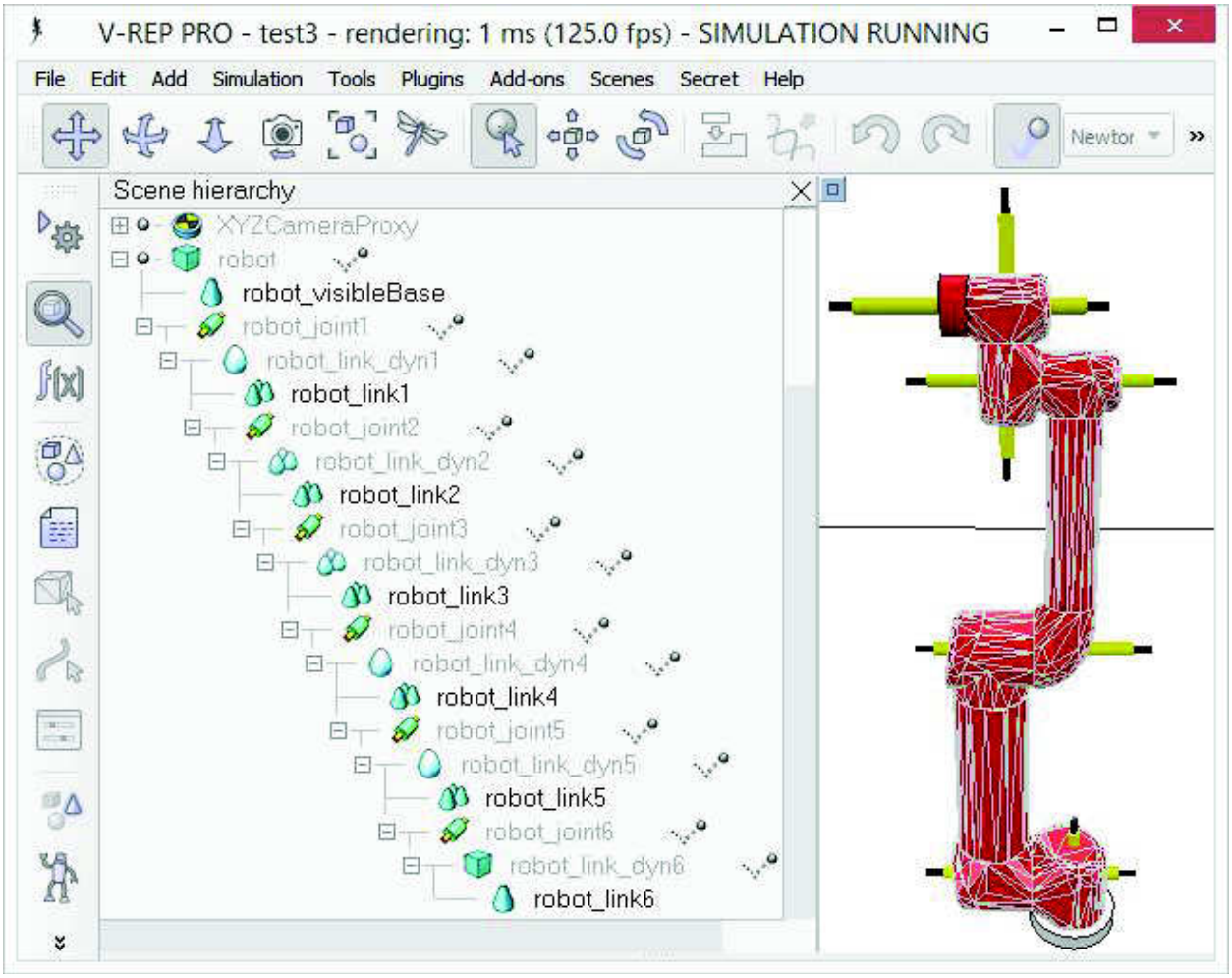
Τώρα προστατεύουμε το μοντέλο μας από τυχαία τροποποίηση. Επιλέγουμε όλα τα ορατά αντικείμενα στο ρομπότ και στη συνέχεια ενεργοποιούμε την επιλογή βάση του μοντέλου αντί: αν τώρα κάνουμε κλικ σε έναν ορατό σύνδεσμο στη σκηνή, θα γίνει επιλογή της βάσης του ρομπότ. Αυτό μας επιτρέπει να χειριζόμαστε το μοντέλο σαν να ήταν ένα μόνο αντικείμενο. Μπορούμε ακόμα να επιλέξουμε ορατά αντικείμενα στο ρομπότ με κλικ ελέγχου στροφής στη σκηνή ή επιλέγοντας το αντικείμενο στην ιεραρχία σκηνών. Βάζουμε τώρα το ρομπότ σε μια σωστή προεπιλεγμένη θέση / προσανατολισμό. Πρώτον, αποθηκεύουμε την τρέχουσα σκηνή ως αναφορά (π.χ. αν σε μεταγενέστερο στάδιο πρέπει να εισαγάγουμε δεδομένα CAD που έχουν τον ίδιο προσανατολισμό στο τρέχον ρομπότ). Στη συνέχεια επιλέγουμε το μοντέλο και τροποποιούμε κατάλληλα τη θέση / τον προσανατολισμό του. Θεωρείται καλή πρακτική η τοποθέτηση του μοντέλου (δηλ. Του αντικειμένου βάσης) στο X = 0 και Y = 0.



Εικόνα 37 Ο βραχίονας στο περιβάλλον εξομοίωσης

Τώρα τρέχουμε την προσομοίωση: το ρομπότ θα καταρρεύσει, αφού οι αρθρώσεις δεν ελέγχονται από προεπιλογή. Όταν προσθέσαμε τις αρθρώσεις στο προηγούμενο στάδιο, δημιουργήσαμε τις αρθρώσεις σε κατάσταση ισχύος / ροπής, αλλά ο κινητήρας ή ο ελεγκτής τους απενεργοποιήθηκε (από προεπιλογή). Τώρα μπορούμε να προσαρμόσουμε τις αρθρώσεις μας στις απαιτήσεις μας. Στην περίπτωσή μας, θέλουμε έναν απλό ελεγκτή PID για κάθε έναν από αυτούς. Στις κοινές δυναμικές ιδιότητες, κάνουμε κλικ στο Motor ενεργοποιημένο και ρυθμίζουμε τη μέγιστη ροπή. Στη συνέχεια, κάντε κλικ στην επιλογή Έλεγχος βρόχου και επιλέξτε Έλεγχος θέσης (PID). Τώρα τρέχουμε ξανά την προσομοίωση: το ρομπότ πρέπει να κρατήσει τη θέση του. Προσπαθήστε να αλλάξετε την τρέχουσα μηχανή φυσικής για να δείτε αν η συμπεριφορά είναι συνεπής σε όλους τους υποστηριζόμενους κινητήρες φυσικής. Μπορείτε να το κάνετε αυτό μέσω του κατάλληλου κουμπιού της γραμμής εργαλείων ή στις γενικές ιδιότητες δυναμικής.

Κατά τη διάρκεια της προσομοίωσης, επαληθεύουμε τώρα το δυναμικό περιεχόμενο σκηνής μέσω του κουμπιού Δυναμικό περιεχόμενο απεικόνισης και επαλήθευσης περιεχομένου. Τώρα, μόνο τα στοιχεία που λαμβάνονται υπόψη από τον κινητήρα φυσικής θα εμφανιστούν και η οθόνη θα είναι χρωματισμένη. Είναι πολύ σημαντικό να κάνετε πάντα αυτό, και ειδικά όταν το δυναμικό σας μοντέλο δεν συμπεριφέρεται όπως αναμενόταν, για να διορθώσετε γρήγορα το μοντέλο. Ομοίως, κοιτάξτε πάντα την ιεραρχία σκηνών κατά τη διάρκεια της προσομοίωσης: τα δυναμικά ενεργοποιημένα αντικείμενα θα πρέπει να εμφανίζουν ένα εικονίδιο οριοθέτησης σφαίρας στη δεξιά πλευρά του ονόματός τους.



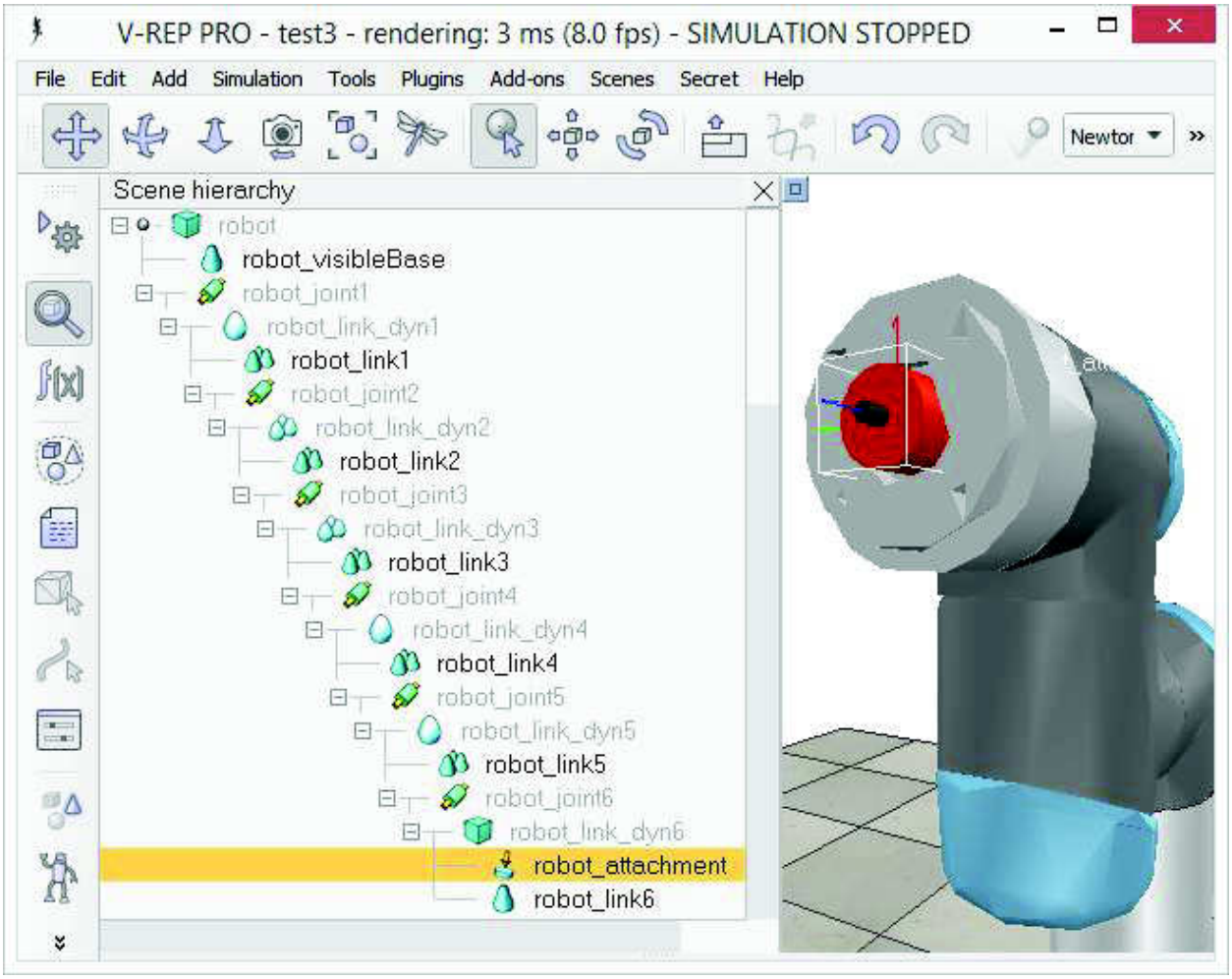
Εικόνα 38 Εξετάζουμε ποια σημεία θέλουμε να είναι δυναμικά

Τέλος, πρέπει να προετοιμάσουμε το ρομπότ για να μπορέσουμε να προσαρμόσουμε εύκολα μια λαβή σε αυτό ή να προσαρτήσουμε εύκολα το ρομπότ σε μια κινητή πλατφόρμα (για παράδειγμα). Δύο δυναμικά ενεργοποιημένα σχήματα μπορούν να συνδεθούν άκαμπτα μεταξύ τους με δύο διαφορετικούς τρόπους:

με την ομαδοποίησή τους: επιλέξτε τα σχήματα, στη συνέχεια [Γραμμή μενού -> Επεξεργασία -> Ομαδοποίηση / Συγχώνευση -> Ομαδοποίηση επιλεγμένων σχημάτων].

με σύνδεση μέσω αισθητήρα δύναμης / ροπής: ένας αισθητήρας ροπής δυνάμεως μπορεί επίσης να λειτουργήσει ως άκαμπτη σύνδεση μεταξύ δύο ξεχωριστών δυναμικά ενεργοποιημένων σχημάτων.

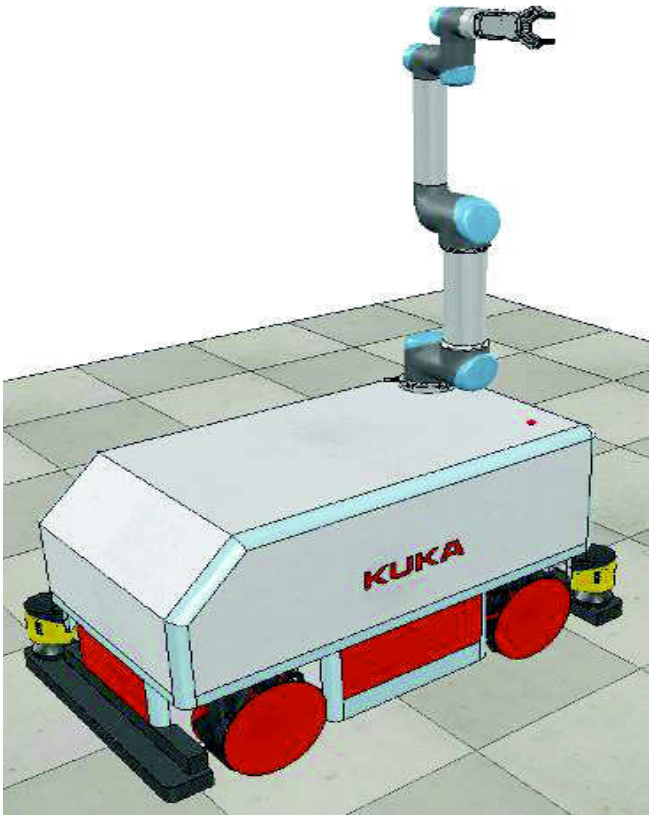
Στην περίπτωσή μας, ενδιαφέρει μόνο η επιλογή 2. Δημιουργούμε έναν αισθητήρα δύναμης / ροπής με τη [Γραμμή μενού -> Προσθήκη -> Αισθητήρας Δύναμης], στη συνέχεια μετακινήστε την στην άκρη του ρομπότ και, στη συνέχεια, συνδέστε το με το αντικείμενο robot_link_dyn6. Αλλάζουμε κατάλληλα το μέγεθος και την οπτική εμφάνιση (ένας κόκκινος αισθητήρας δύναμης / ροπής συχνά θεωρείται ως προαιρετικό σημείο προσάρτησης, ελέγξτε τα διάφορα μοντέλα ρομπότ που υπάρχουν). Αλλάζουμε επίσης το όνομά του στο robot_attachment:



Εικόνα 39 Τοποθέτηση Αρπάγης σε αρθωση βραχίονα

Τώρα σύρετε ένα μοντέλο λαβής στη σκηνή, κρατήστε το επιλεγμένο, έπειτα κάντε κλικ στον έλεγχο του αισθητήρα δύναμης προσάρτησης και, στη συνέχεια, κάντε κλικ στο κουμπί συναρμολόγησης / αποσυναρμολόγησης της γραμμής εργαλείων. Η λαβή μπαίνει στη θέση της:

Η λαβή γνώριζε πώς να συνδέεται, επειδή ήταν κατάλληλα διαμορφωμένη κατά τον ορισμό του μοντέλου. Τώρα πρέπει επίσης να ρυθμίσουμε σωστά το μοντέλο ρομπότ, έτσι ώστε να γνωρίζει πώς να προσαρτάται για παράδειγμα σε μια κινητή βάση. Επιλέγουμε το μοντέλο ρομπότ και, στη συνέχεια, κάντε κλικ στην επιλογή Συναρμολόγηση στις κοινές ιδιότητες του αντικειμένου. Ορίστε μια κενή συμβολοσειρά για τις τιμές αντιστοίχισης "γονέα" και, στη συνέχεια, κάντε κλικ στην επιλογή Ορισμός μήτρας. Αυτό θα απομνημονεύει την τοπική μήτρα μετασχηματισμού του αντικειμένου βάσης και θα το χρησιμοποιήσει για να το τοποθετήσει / προσανατολιστεί σε σχέση με το σημείο προσάρτησης του κινητού ρομπότ. Για να επιβεβαιώσουμε ότι κάναμε τα πράγματα σωστά, μεταφέρουμε το μοντέλο Μοντέλα / ρομπότ / κινητό / KUKA Omnirob.ttm στη σκηνή. Στη συνέχεια, επιλέγουμε το μοντέλο ρομπότ μας και, στη συνέχεια, κάντε κλικ στον έλεγχο ενός από τα σημεία προσάρτησης στην πλατφόρμα κινητής τηλεφωνίας και, στη συνέχεια, κάντε κλικ στο κουμπί Συναρμολόγηση / αποσυναρμολόγηση της γραμμής εργαλείων. Το ρομπότ μας πρέπει να τοποθετηθεί σωστά στην κορυφή του κινητού ρομπότ:



Εικόνα 40 Τοποθέτηση βραχίονα σε άρθοση πάνω στο Omnirob

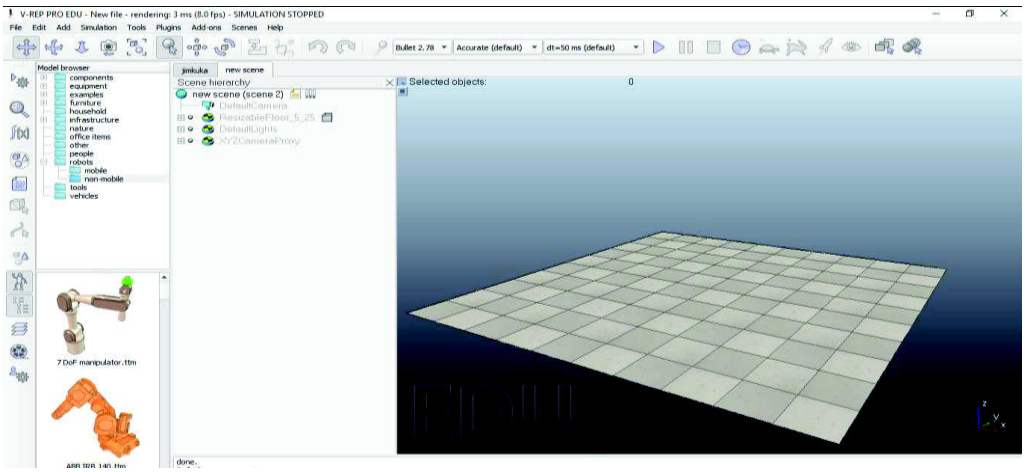
Τώρα μπορούμε να προσθέσουμε επιπλέον στοιχεία στο ρομπότ μας, όπως για παράδειγμα αισθητήρες. Κάποια στιγμή θα θέλαμε επίσης να επισυνάψουμε ενσωματωμένα σενάρια στο μοντέλο μας, προκειμένου να ελέγξουμε τη συμπεριφορά του ή να το διαμορφώσουμε για διάφορους σκοπούς. Σε αυτή την περίπτωση, βεβαιωθείτε ότι έχετε κατανοήσει πώς έχουν πρόσβαση οι χειρολαβές αντικειμένων από ενσωματωμένα σενάρια. Μπορούμε επίσης να ελέγξουμε / να προσπελάσουμε / να διασυνδέσουμε το πρότυπό μας από ένα plugin, από έναν απομακρυσμένο πελάτη API, από έναν κόμβο ROS, από έναν κόμβο BlueZero ή από ένα πρόσθετο.

Τώρα βεβαιώνουμε ότι έχουμε αλλάξει τις αλλαγές που έγιναν κατά τη διάρκεια της προσκόλλησης ρομπότ και αρπαγής, καταρρέουμε το δέντρο ιεραρχίας του μοντέλου ρομπότ μας, επιλέγουμε τη βάση του μοντέλου μας και στη συνέχεια το αποθηκεύουμε με [Γραμμή μενού -> Αρχείο -> Αποθήκευση μοντέλου ...]. Εάν το αποθηκεύσαμε στο φάκελο μοντέλου, τότε το μοντέλο θα είναι διαθέσιμο στο φυλλομετρητή μοντέλου.

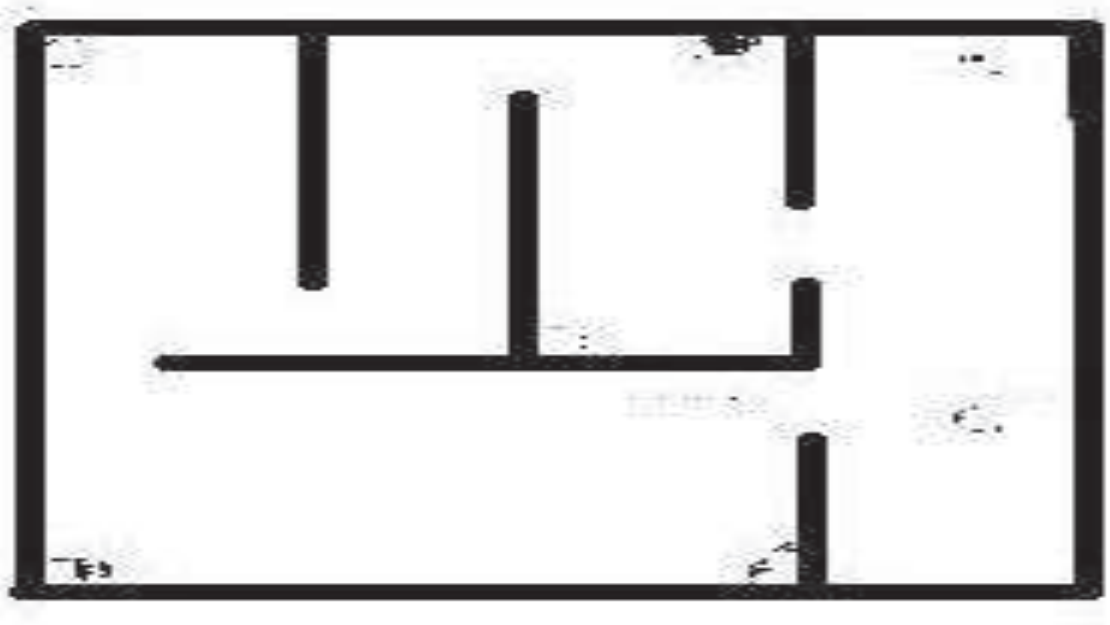
4.5 Πως θα χρησιμοποιήσουμε το Kuka;

Αρχικά αφού τοποθετήσουμε το Kuka στο άδειο περιβάλλον εξομοίωσης αρχίζουμε να δημιουργούμε τοίχους με διάφορα εξαρτήματα τα οποία σε συνεργασία μεταξύ τους θα φτιάξουν ένα χώρο προσομοίωσης σπιτιού στον οποίο θα κάνει πλοηγηθεί το Kuka.

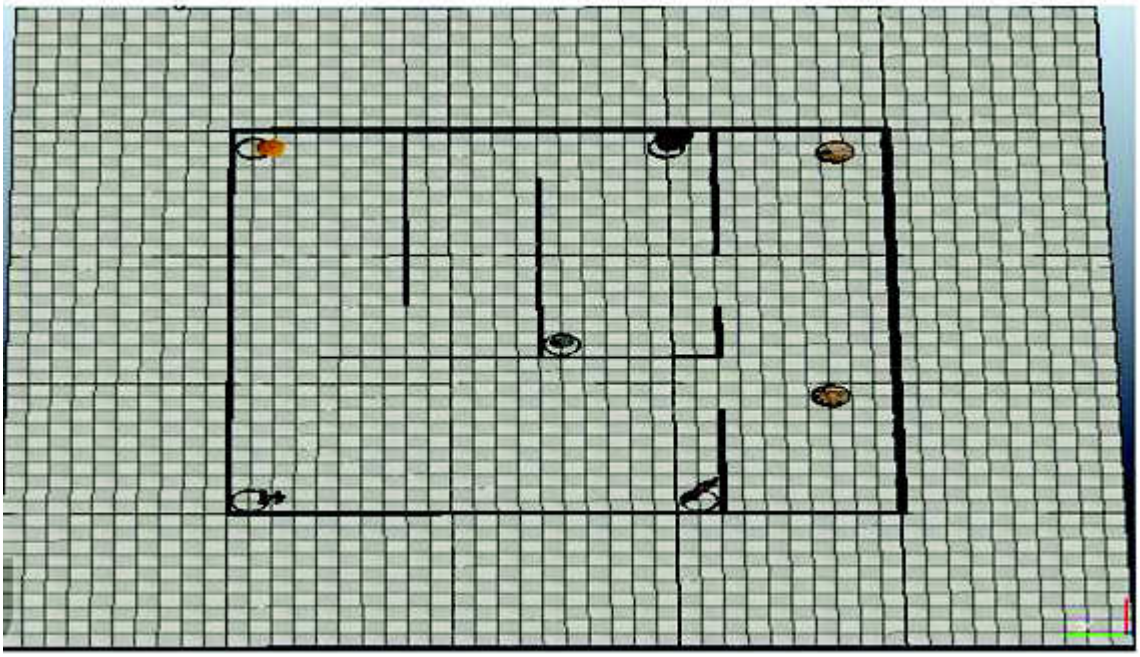
Σκοπός του Kuka είναι η αναγνώριση του χάρτη, η δημιουργία μονοπατιού το οποίο δεν θα χτυπάει στου τοίχους και αποφυγή διάφορων αντικείμενων που διάσπαρτα σε διάφορα σημεία του περιβάλλοντος εξομοίωσης που θα δημιουργήσουμε όπως φαίνεται στις παρακάτω φωτογραφίες.



Εικόνα 41 Ανοιγμα Κενού Χάρτη Vrep



Εικόνα 42 Σχέδοπ Επιθυμητου Χάρτη που πρέπει να υλοποιησουμε την Αναγνώριση στο προσομοιωτή Vrep



Εικόνα 43Εικόνα πως ο χάρτης δημιουργημένος από ψηλά

4.6 Ο συγχρονισμός μεταξύ Matlab και Vrep

Το πιο σημαντικό χαρακτηριστικό για να μπορέσει να λειτουργήσει το Kuka και να κάνει πλοηγήσει σε άγνωστο χάρτη είναι ο συγχρονισμός του με το Matlab ο οποίος θα πρέπει να είναι τέτοιος ώστε οι εντολές του Matlab να συγχρονίζονται με το χρόνο απόκρισης του Kuka στο Vrep.

Το πρώτο σκέλος του συγχρονισμού γίνεται με τη μεταφορά φακέλων του Remote Api στο φάκελο εργασίας μας που θα τρέχει το λογισμικό μας, αυτό επιτυγχάνεται

4.6.1 Ενεργοποίηση της πλευράς απομακρυσμένου API - διακομιστή

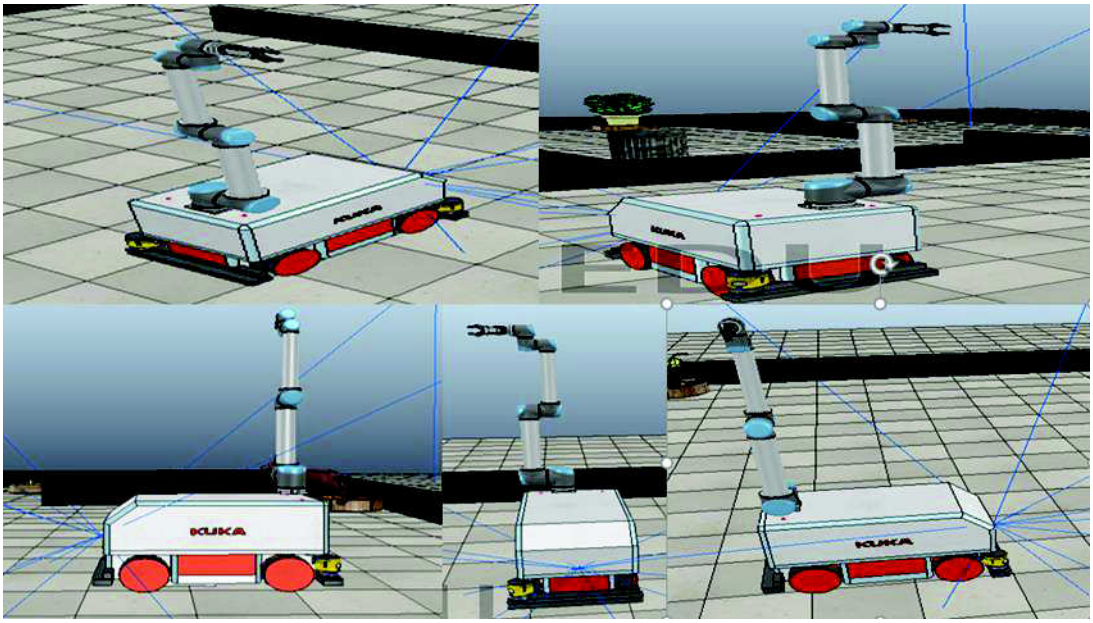
Η πλευρά του απομακρυσμένου διακομιστή API υλοποιείται μέσω ενός plugin V-REP που βασίζεται στο κανονικό API. Το έργο plugin API απομακρυσμένης πρόσβασης βρίσκεται εδώ. Εάν χαθεί κάποια συγκεκριμένη λειτουργία, τότε μπορούμε εύκολα να το εφαρμόσουμε μόνοι μας στο πλαίσιο απομακρυσμένου API

Για να ενεργοποιήσουμε το απομακρυσμένο API από την πλευρά του διακομιστή, από την πλευρά του V-REP, βεβαιωνόμαστε ότι το απομακρυσμένο API plugin που έχει να κάνει με το Matlab έχει εκτελεστεί εντός του κώδικα μας και έπειτα κοιτάμε αν φορτώθηκε με επιτυχία σε εκκίνηση V-REP επιθεωρούμε το παράθυρο της κονσόλας για πληροφορίες σχετικά με τη φόρτωση του plugin μέσα στο command window ή αν υπάρχει κάποιο σφάλμα. Αλλιώς είμαστε έτοιμοι να θέσουμε σε λειτουργία και να δημιουργήσουμε κώδικα έτσι ώστε να ξεκινήσουμε τη πλοήγηση του ρομποτικού μας συστήματος Kuka. Κάθε υπηρεσία επικοινωνεί σε διαφορετική θύρα για τον απομακρυσμένο έλεγχο API plugin. Μια υπηρεσία διακομιστή μπορεί να ξεκινήσει με δύο διαφορετικούς τρόπους

Στην εκκίνηση V-REP Το απομακρυσμένο API plugin θα προσπαθήσει να διαβάσει ένα αρχείο διαμόρφωσης που ονομάζεται remoteApiConnections.txt και σύμφωνα με το περιεχόμενό του, να ξεκινήσει τις κατάλληλες υπηρεσίες διακομιστή. Χρησιμοποιούμε αυτήν τη μέθοδο για τον τηλεχειρισμό του ίδιου του προσομοιωτή. Με αυτήν τη μέθοδο, οι λειτουργίες API θα εκτελούνται πάντοτε από την πλευρά του διακομιστή, ακόμη και αν η προσομοίωση δεν εκτελείται (κάτι που δεν συμβαίνει πάντοτε με την επόμενη μέθοδο παρακάτω). Υπάρχει μια άλλη μέθοδος για την εκκίνηση μιας συνεχούς απομακρυσμένης υπηρεσίας διακομιστή API, μέσω της γραμμής εντολών.

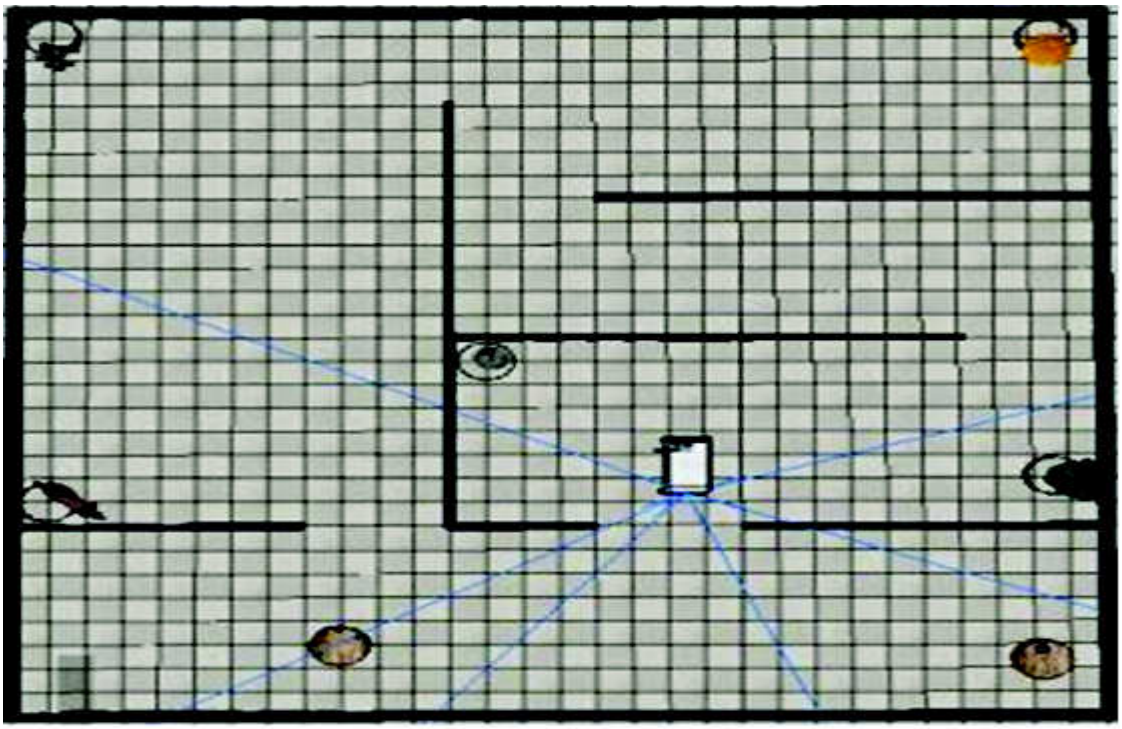
Από μέσα σε μια δέσμη ενεργειών (προσωρινή απομακρυσμένη υπηρεσία διακομιστή API). Αυτό είναι οι περισσότερες φορές η προτιμώμενη μέθοδος εκκίνησης μιας απομακρυσμένης υπηρεσίας διακομιστή API. Ο χρήστης έχει τον έλεγχο όταν η υπηρεσία ξεκινά ή σταματά. Όταν μια προσωρινή απομακρυσμένη υπηρεσία διακομιστή API ξεκινά από ένα σενάριο προσομοίωσης, ωστόσο, η υπηρεσία θα σταματήσει αυτόματα στο τέλος προσομοίωσης. Μια προσωρινή απομακρυσμένη υπηρεσία διακομιστή API μπορεί να ξεκινήσει ή να σταματήσει με τις ακόλουθες 2 προσαρμοσμένες λειτουργίες Lua (οι δύο λειτουργίες εξάγονται από το plugin):

4.7 Φωτογραφίες Περιβάλλοντος Προσομοίωσης



Εικόνα 44 Το kuka ρομποτ ξεκινάει την αναγνώριση

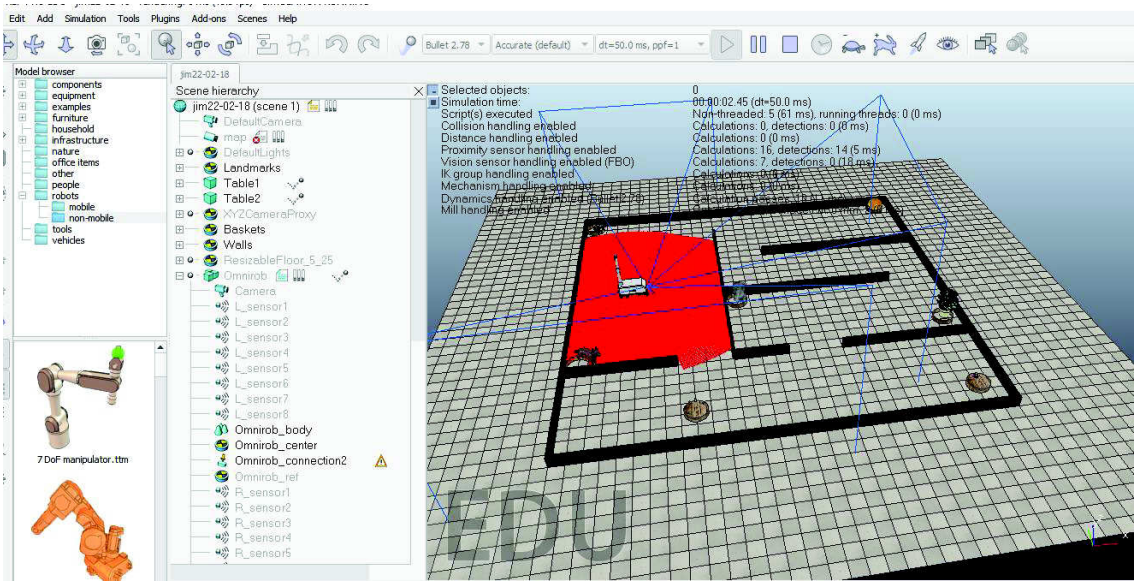
Στη συγκεκριμένη διερευνητική εργασία έγινε χρήση 2 διαφορετικών των μεθόδων για τη πλοήγησης του ρομποτικού συστήματος το 1ο είναι PRM και το 2ο σύστημα είναι το Monte Carlo



Εικόνα 45 Ενεργοποίηση Αισθητήρων

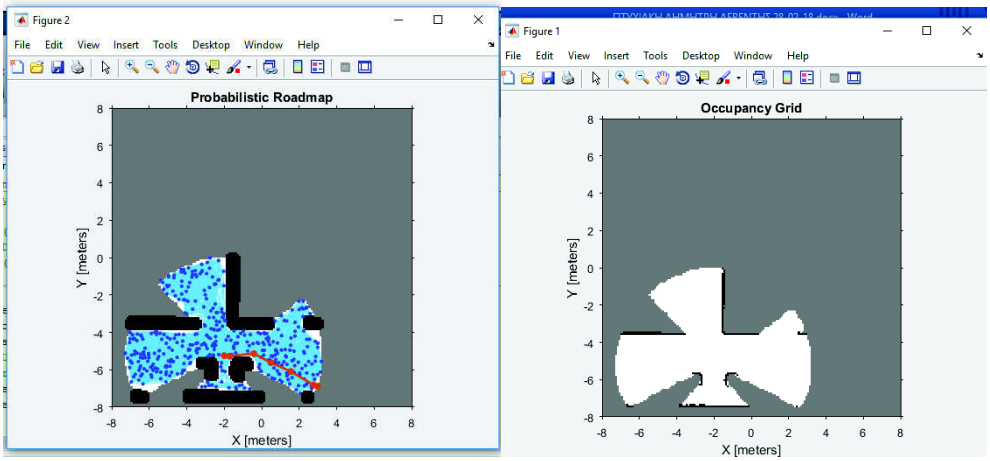
Το σύστημα Kuka με τη μέθοδο PRM έχει ως μέθοδο να ανιχνεύει το χώρο με τους αισθητήρες του πριν από κάθε κίνηση και κάνοντας επεξεργασία του χώρου που βρίσκεται την εκάστοτε

στιγμή τοποθετεί σε διαφορετικά πιθανολογικά σημεία τελείες, όπως στη παρακάτω εικόνα με σκοπό να κάνει σχεδιασμό της διαδρομής για τη αποφυγή εμποδίων εντός του χάρτη.



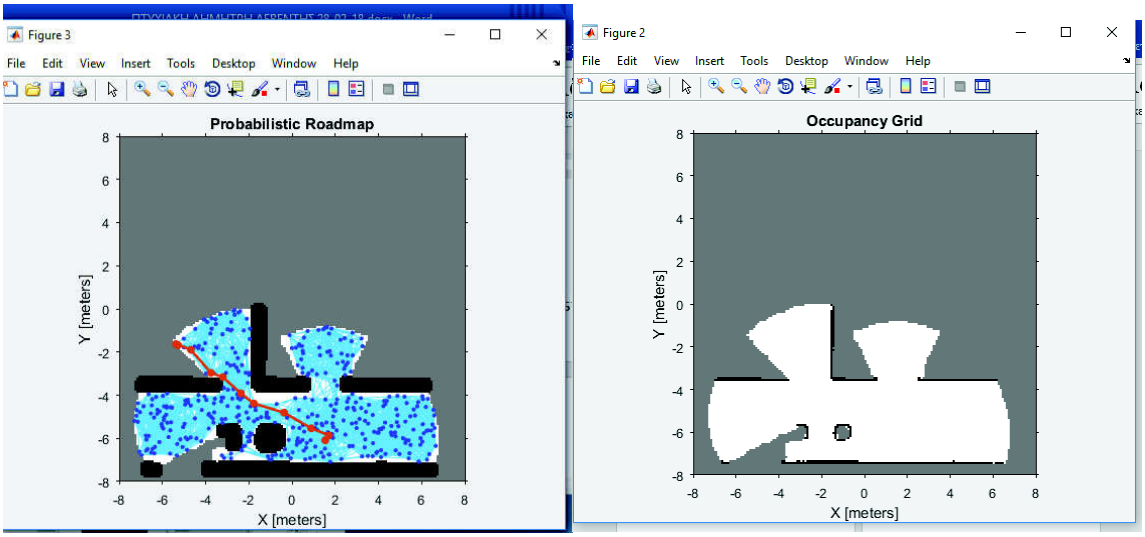
Εικόνα 46 Ενεργοποίηση Αισθητήρων Laser Περιστροφικης Εμβέλειας 360⁰

Έτσι κάθε φορά το Kuka που σχεδιάζει μία διαδρομή αναθεωρεί αυτές τι τελείες σε διαφορετικά σημεία έτσι ώστε να σχεδιάσει μία διαδρομή καινούρια για την αποφυγή εμποδίων. Αυτό μας δίνει τη δυνατότητα με τη βοήθεια κώδικα να μπορέσουμε να ομαδοποιήσουμε κάθε φορά το σχεδιασμό της διαδρομής που κάνει το Kuka στο χάρτη και να δημιουργήσουμε έναν ολοκληρωμένο χάρτη με τελείες στο κώδικα μας ομαδοποιώντας όλα τα κομμάτια του χάρτη που έχει χρησιμοποιήσει κάθε φορά όπως παρουσιάζεται στις παρακάτω εικόνες.

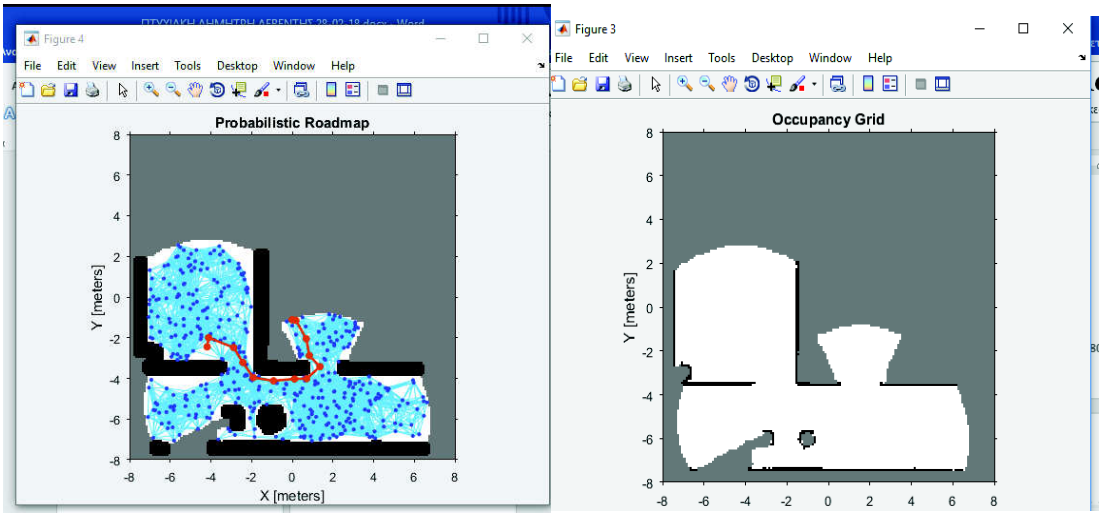


Εικόνα 47 Έναρξη Συνεργασίας Vrep και Matlab

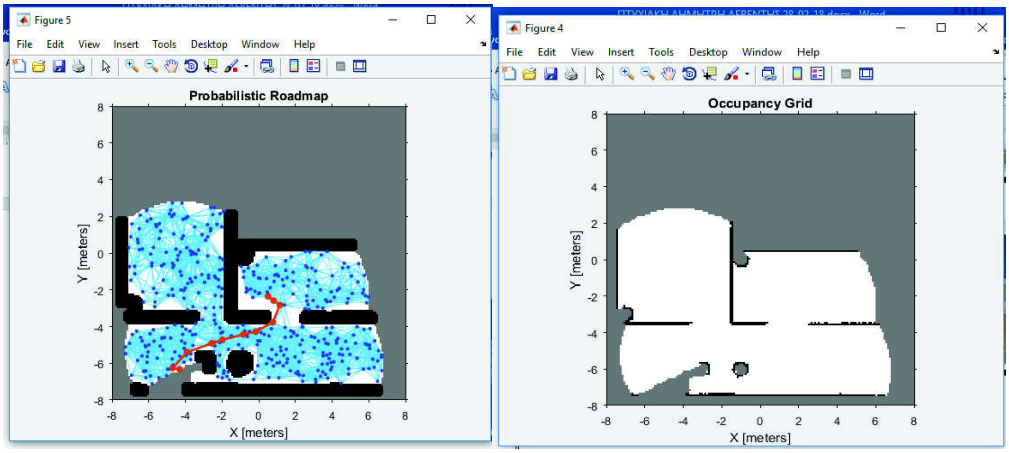
Η εικόνα από περιβάλλον Matlab που έχει ξεκινήσει την επιλογή διαδρομής του το Kuka, αυτό γίνεται έτσι ώστε να κάνει αναγνώριση του περιβάλλοντος πλοήγησης και να μπορέσει να πλοηγηθεί στο χώρο ανάλογα με τη τοπικοποίηση του.



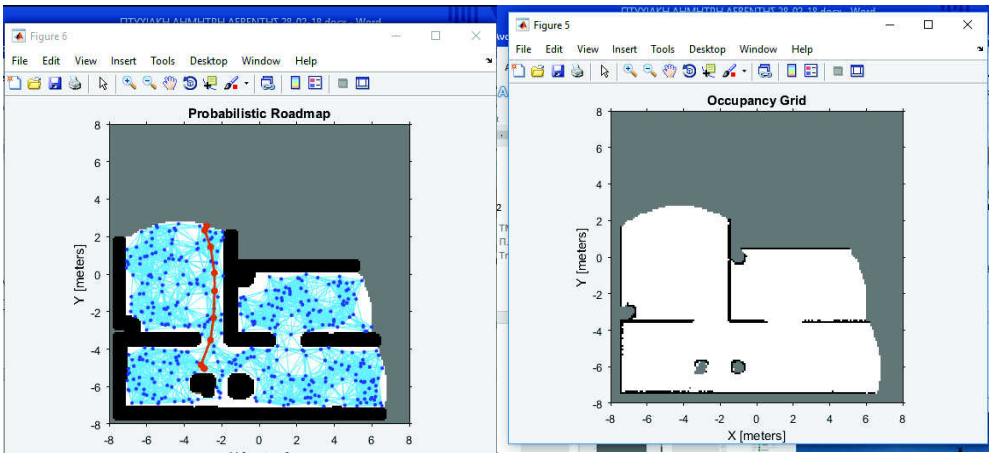
Εικόνα 48 Αναγνώριση Χάρτη με τη βοήθεια κώδικα μέσω του Matlab



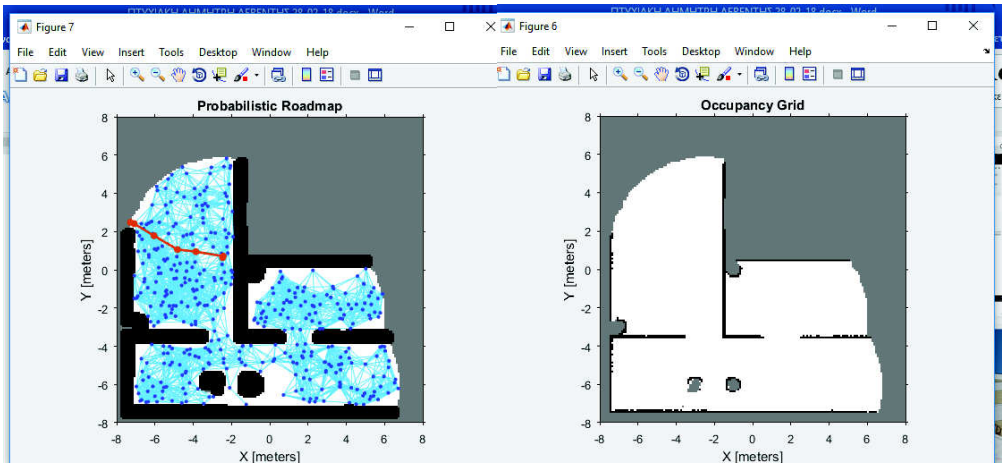
Εικόνα 49 Το Matlab τρέχοντας τις εντολές του ενώνει τις τελειες και δημιουργει νέα διαδρομη



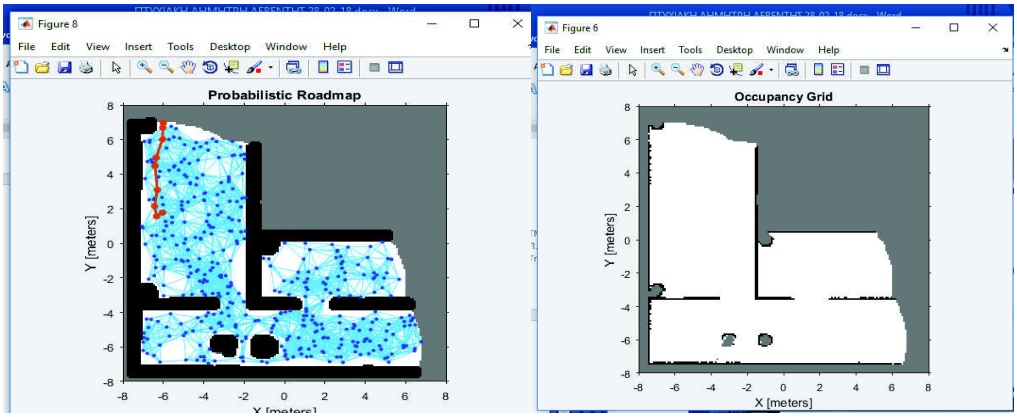
Εικόνα 50 Ο χάρτης αναθεωρείαι κάθε φορά που αποθηκεύεται ένα κομματι χάρτη στην δεξια εικόνα



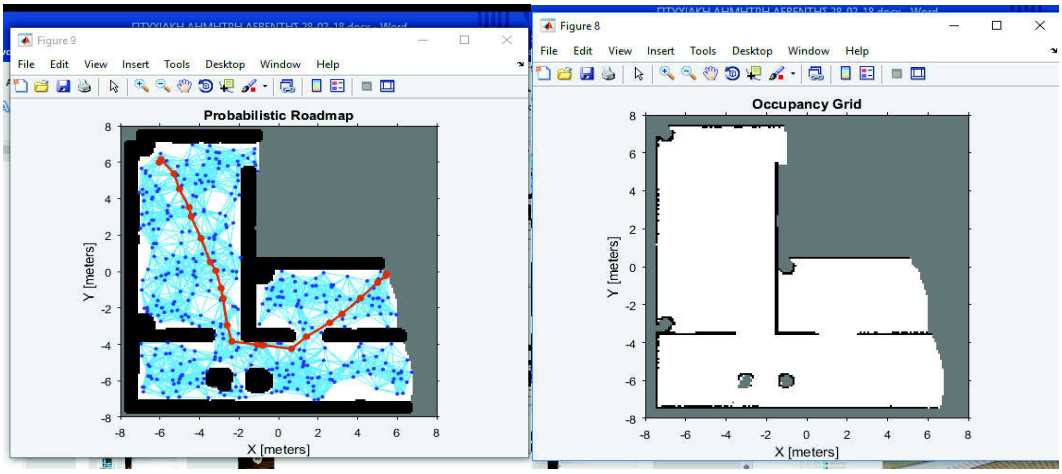
Εικόνα 51 Ο αλγόριθμος που τρέχει και επεξεργάζεται νέα δεδομένα



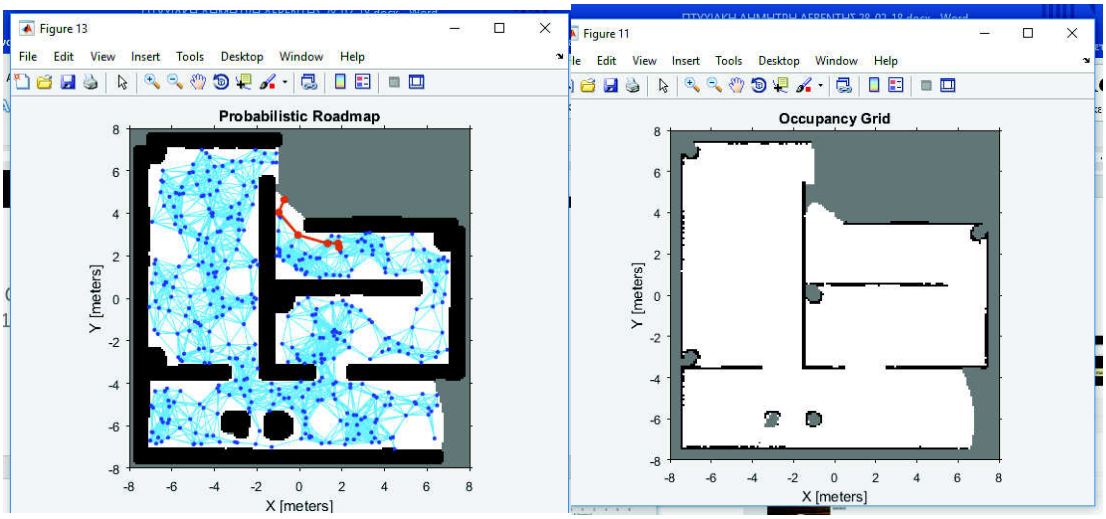
Εικόνα 52Το omnirob αποφεύγει και ορισμένα εμπόδια εκτός από τους τοιχους του χαρτη



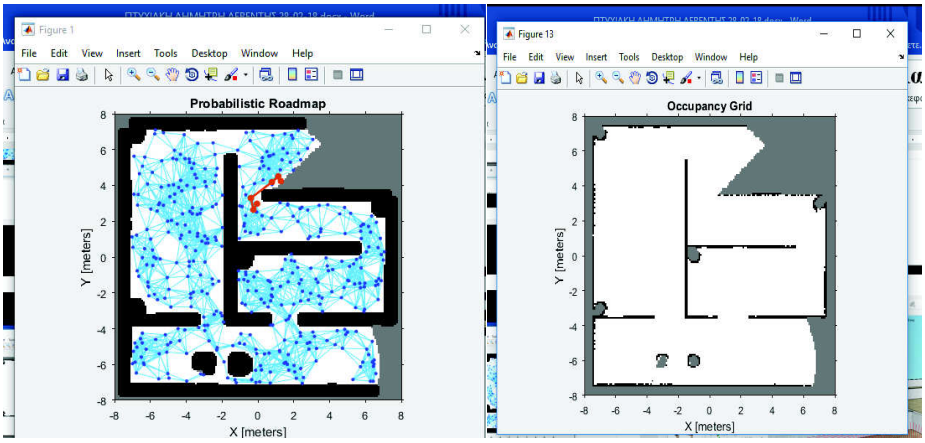
Εικόνα 53



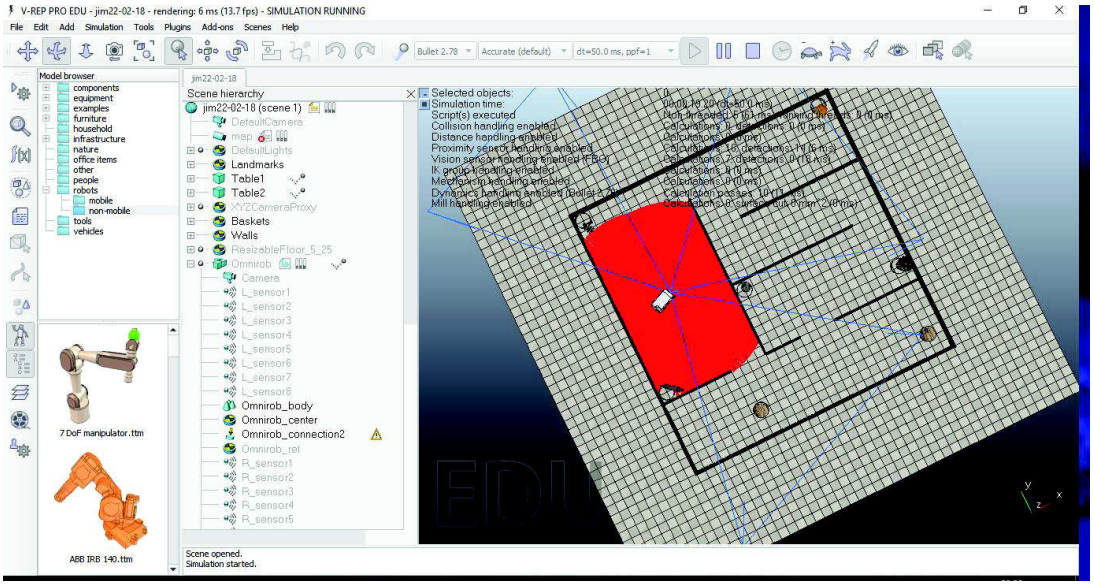
Εικόνα 54



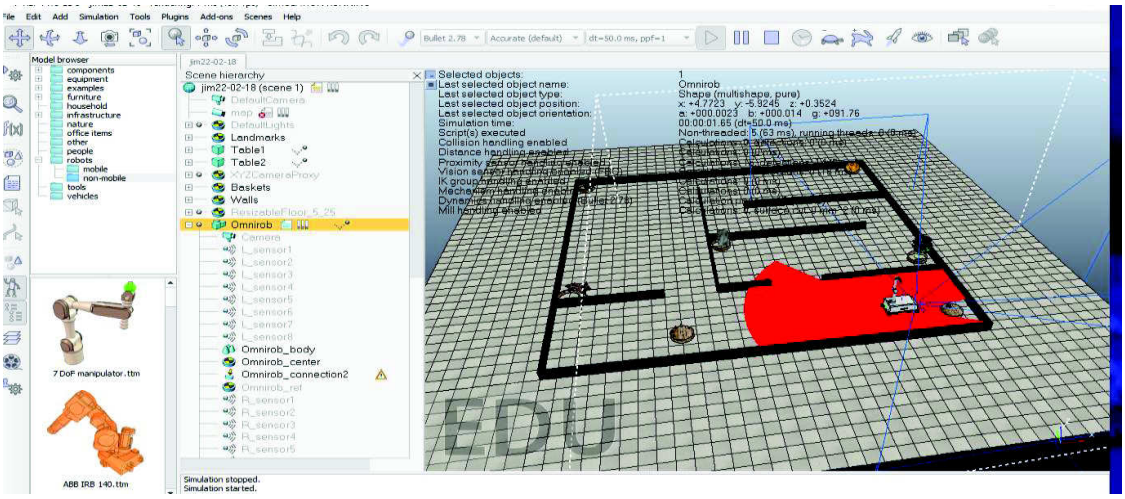
Εικόνα 55



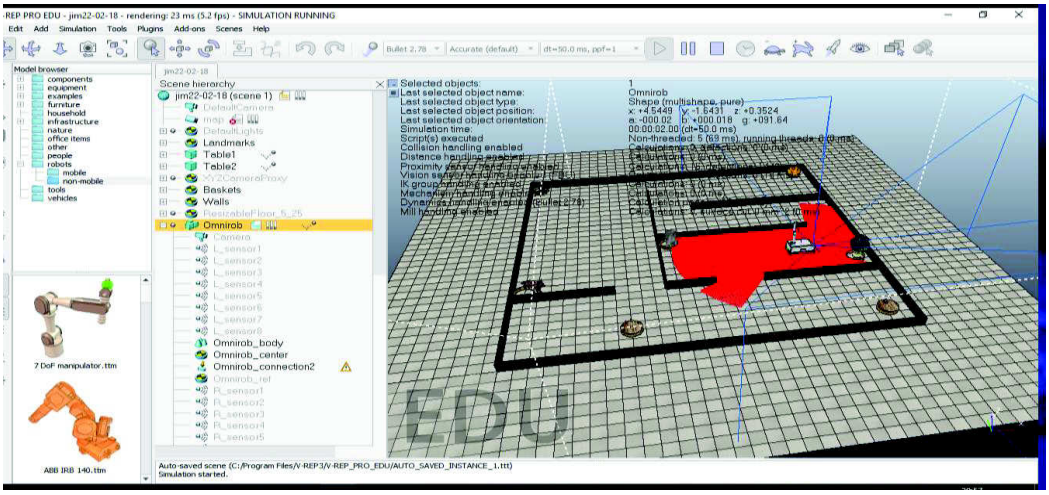
Εικόνα 56



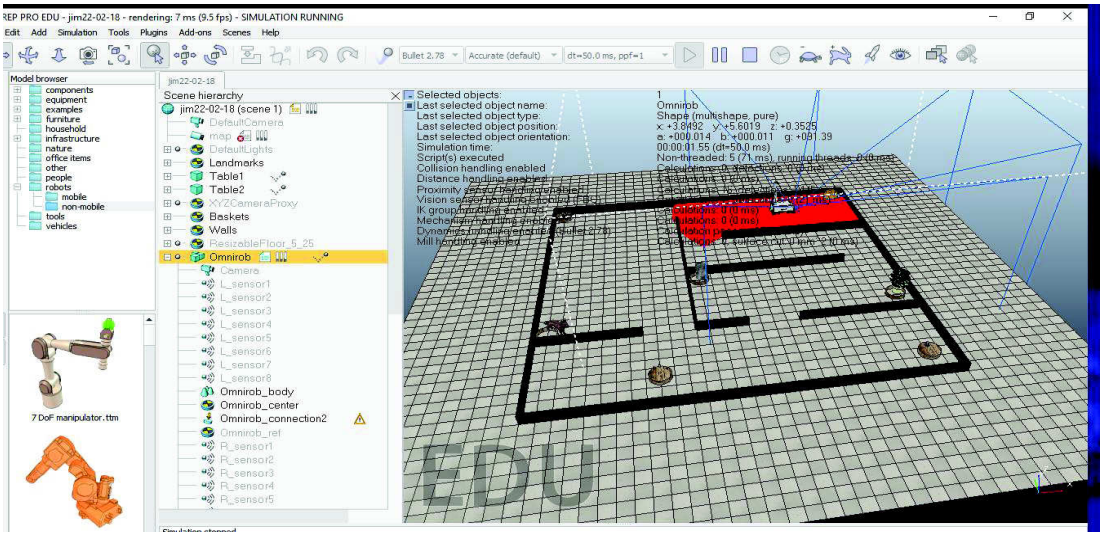
Εικόνα 57 Εικόνα του kuka αφου έχει ξεκινήσει την αναγνωριση χάρτη με τη μέθοδο Monte Carlo



Εικόνα 58 Εικόνα του omnirob αφου έχει κάνει αναγνώριση το περιβάλλον αξιολογεί τη διαδρομή που θα ακολουθήσει



Εικόνα 59 Το omnirob έχει μπει με επιτυχία εντός του λαβύρηνθου.



Εικόνα 60 Το omnirob ολοκληρώνει της πλοήγηση στο χάρτη χώρις να χρειαστεί να τον αποθηκευσει το χάρτη κάνοντας χρήση Μαρκοβιανών Μεθόδων και Στατιστικης με τη βοήθεια του Matlab

Κεφάλαιο 5

Συμπεράσματα – Προοπτικές Διερεύνησης Πλοήγησης Ρομποτικών Συστημάτων

Το λογισμικό ros Indigo είναι ένα λογισμικό ανοιχτού κώδικα πρώτα από όλα το οποίο μπορεί να τρέξει εκτός από το ελεύθερο λογισμικό Linux και οποιοδήποτε άλλο λογισμικό από τα γνωστά εμπορικά Mac OS Windows. Αυτό δίνει τη δυνατότητα να συνδυάσει ένα πλήρες φάσμα δυνατοτήτων και μελλοντική του εξελιξιμότητα.

5.1 Πλεονεκτήματα χρήσης προσομοίωσης

Η προσομοίωση χρησιμοποιείται για να προγραμματίσουμε εξατομικευμένες εφαρμογές για ένα ρομπότ χωρίς την αναγκαία ύπαρξή του. Κύριο πλεονέκτημα είναι ότι οι περισσότερες εφαρμογές μπορούν να μεταφερθούν σε ένα ρομποτικό σύστημα χωρίς σημαντικές αλλαγές. Η αγορά, η αναμονή της παραλαβής και συναρμολόγησης μπορεί να είναι χρονοβόρα διαδικασία ενώ η προσομοίωση λόγω του ανοιχτού λογισμικού το μόνο που χρειάζεται είναι η εγκατάσταση και η εκμάθηση του. Με την προσομοίωση, μπορεί ο χρήστης να έχει τα αποτελέσματα μίας παραμετροποίησης που έχει κάνει στο ρομποτικό σύστημα σε πολύ μικρό χρονικό διάστημα. Επιπλέον σημαντικός παράγοντας είναι το χαμηλό κόστος . Η έρευνα λοιπόν, προχωράει χωρίς την ανάγκη χρηματοδότησης. Με αυτό τον τρόπο υπάρχει δυνατότητα επαναλαμβανόμενων πειραμάτων χωρίς τις υλικές δεσμεύσεις που θα υπήρχαν σε πραγματικές συνθήκες υλοποίησης. Δυστυχώς παρά τα πλεονεκτήματα του, ακόμα και τα καλύτερα λογισμικά δεν μπορούν να προσομοιώσουν τέλεια την εφαρμογή σε πραγματικές συνθήκες λόγω των πολύπλευρων διαταραχών που μπορεί να αντιμετωπίσει το εκάστοτε σύστημα.

5.2 Μειονεκτήματα

Βέβαια το αρνητικό με τα λογισμικά Ανοιχτού Κώδικα είναι ότι υπάρχουν πολλές νομικές προκλήσεις. Οι εταιρείες που διαχειρίζονται προϊόντα ανοιχτού κώδικα όπως η ros έχουν κάποια δυσκολία να εξασφαλίσουν τα εμπορικά τους σήματα. Για παράδειγμα, το πεδίο εφαρμογής των εικασιών «υπονοούμενων αδειών» παραμένει ασαφές και μπορεί να θέσει σε κίνδυνο την ικανότητα μιας επιχείρησης να κατοχυρώνει διπλώματα ευρεσιτεχνίας με παραγωγές με λογισμικό ανοικτής πηγής. Ένα άλλο παράδειγμα είναι η περίπτωση εταιρειών που προσφέρουν πρόσθετα για αγορά. οι κάτοχοι άδειας χρήσης που κάνουν προσθήκες στον κώδικα ανοιχτού κώδικα που

είναι παρόμοιοι με εκείνους που αγοράζονται ενδέχεται να έχουν ασυλία από τα κοστούμια ευρεσιτεχνίας.

Η δομή της κοινότητας ανοιχτού κώδικα προϋποθέτει ότι τα άτομα διαθέτουν τεχνογνωσία στον προγραμματισμό, προκειμένου να συμμετάσχουν σε ανοιχτές αλλαγές κώδικα και ανταλλαγές. Τα άτομα που ενδιαφέρονται να στηρίξουν το κίνημα του ανοιχτού κώδικα ενδέχεται να μην διαθέτουν αυτό το σύνολο δεξιοτήτων, αλλά υπάρχουν και πολλοί άλλοι τρόποι συμβολής.

Οι προγραμματιστές και οι προγραμματιστές αποτελούν ένα μεγάλο ποσοστό της κοινότητας ανοικτού πηγαίου κώδικα και η επιδιωκόμενη τεχνική υποστήριξη ή / και τεκμηρίωση ενδέχεται να μην είναι χρήσιμες ή σαφείς για χρήστες λογισμικού ανοικτού κώδικα .

Η δομή της κοινότητας ανοιχτού κώδικα είναι μια δομή που περιλαμβάνει συνεισφορές πολλών προγραμματιστών και προγραμματιστών. το λογισμικό που παράγεται με αυτόν τον τρόπο μπορεί να μην έχει τυποποίηση και συμβατότητα με διάφορες εφαρμογές και δυνατότητες υπολογιστών.

Η παραγωγή μπορεί να είναι πολύ περιορισμένη. Οι προγραμματιστές που δημιουργούν λογισμικό ανοιχτού κώδικα μπορούν συχνά να επιστρέψουν την προσοχή τους αλλού πολύ γρήγορα. Αυτό ανοίγει την πόρτα για πολλά προγράμματα και εφαρμογές γεμάτες σφάλματα εκεί έξω. Επειδή κανείς δεν πληρώνεται για να το δημιουργήσει, πολλά έργα δεν ολοκληρώνονται ποτέ.

Στη βιομηχανία ανοιχτού κώδικα, ο χρήστης αποφασίζει την ποιότητα του λογισμικού. Ένας χρήστης πρέπει να μάθει ανεξάρτητα τις δεξιότητες της δημιουργίας λογισμικού και στη συνέχεια να κάνει τους κατάλληλους προσδιορισμούς για την ποιότητα και τις δυνατότητες

Οι βιβλιοθηκονόμοι ενδέχεται να μην είναι ικανοί να αναλάβουν αυτή τη νέα ευθύνη των τεχνολογιών .Δεν υπάρχει εγγύηση ότι η ανάπτυξη θα συμβεί. Δεν είναι γνωστό εάν ένα έργο ανοιχτού κώδικα θα καταστεί χρήσιμο, ειδικά όταν ένα έργο ξεκινά χωρίς σημαντική υποστήριξη από έναν ή περισσότερους οργανισμούς. Ακόμα και αν το έργο φτάσει σε ένα χρηστικό στάδιο, είναι πιθανό το έργο να πεθάνει εάν δεν υπάρχει επαρκής χρηματοδότηση ή ενδιαφέρον για αυτό. Είναι μερικές φορές δύσκολο να γνωρίζουμε ότι υπάρχει ένα έργο και η τρέχουσα κατάσταση του. Ειδικά για έργα ανοιχτού κώδικα χωρίς σημαντική υποστήριξη, δεν υπάρχει μεγάλη διαφήμιση στο λογισμικό ανοιχτού κώδικα. Το ποσό υποστήριξης για ένα έργο ανοιχτού κώδικα ποικίλλει σε μεγάλο βαθμό. Η διαθέσιμη υποστήριξη για το λογισμικό ανοιχτού κώδικα είναι κατά κύριο λόγο αυτοτροφοδοτούμενες συζητήσεις στο Διαδίκτυο, που μερικές φορές συντονίζονται από μια κεντρική ομάδα συνεργατών. Το ποσό της τεκμηρίωσης ή των οδηγών για ένα έργο ανοιχτού κώδικα ποικίλλει επίσης σημαντικά. Τα πιο δημοφιλή ή υποστηριζόμενα από την εταιρεία έργα

έχουν συχνά λεπτομερέστερη και συντηρημένη τεκμηρίωση. Ωστόσο, καθώς τα έργα ανοιχτού κώδικα αλλάζουν τακτικά, η τεκμηρίωση μπορεί εύκολα να ξεπεραστεί.

Δεν υπάρχει εγγύηση για ενημερώσεις. Παρόλο που το λογισμικό ανοιχτού κώδικα είναι διαθέσιμο σε όλους δωρεάν, δεν είναι εξασφαλισμένες οι τακτικές ενημερώσεις, δεδομένου ότι οι χρήστες δεν πληρώνουν για τη χρήση του.

Πέραν των προφανών ζημιών προς τη θεωρητική επιτυχία του λογισμικού ανοιχτού κώδικα, υπάρχουν διάφοροι παράγοντες που συμβάλλουν στην έλλειψη μακροπρόθεσμης επιτυχίας σε έργα ανοιχτού κώδικα. Ένα από τα πιο προφανή μειονεκτήματα είναι ότι χωρίς αγορά άδειας πνευματικών δικαιωμάτων/αμοιβή , υπάρχει ελάχιστο οικονομικό κίνητρο για έναν προγραμματιστή να εμπλακεί με ένα έργο στην πρώτη θέση ή για να συνεχίσει την ανάπτυξη και υποστήριξη μετά την απελευθέρωση του αρχικού προϊόντος. Αυτό οδηγεί σε αμέτρητα παραδείγματα λογισμικού που αναμένεται για πάντα να καταδικάζονται σε εκδόσεις beta και σε μη υποστηριζόμενα πρότυπα προϊόντα μοντέλου. Με τις δωρεές ως μοναδική πηγή εισοδήματος για ένα έργο ανοικτού κώδικα (και με άδεια GPL), δεν υπάρχει σχεδόν καμία βεβαιότητα στο μέλλον του έργου απλώς και μόνο λόγω εγκατάλειψης του έργου, καθιστώντας την κακή επιλογή για κάθε είδους εφαρμογή στην οποία το μέλλον οι εκδόσεις, η υποστήριξη και ένα μακροπρόθεσμο σχέδιο θα ήταν απαραίτητες, όπως συμβαίνει με το μεγαλύτερο μέρος του επιχειρησιακού λογισμικού.

Οι οργανισμοί με επιχειρηματικές συμφωνίες εξακολουθούν να καταβάλλουν συμφωνίες παραχώρησης αδειών ακόμα και αν επιλέξουν να χρησιμοποιούν εναλλακτικό λογισμικό ανοιχτού κώδικα. Ως εκ τούτου, πολλές οργανώσεις είναι απίθανο να εξετάσουν τη χρήση εναλλακτικών προϊόντων. Ως μέθοδος εξοικονόμησης κόστους που χρησιμοποιεί τα προϊόντα της Microsoft, πολλές μεγάλες εταιρείες χρησιμοποιούν επιχειρηματικές συμφωνίες και συνεπώς καταβάλλουν ενιαία αμοιβή αδειών χρήσης τεχνολογίας πληροφορικής, με χαμηλότερο κόστος ανά προϊόν. "Οι οργανισμοί με ΕΑ που ενδιαφέρονται για εναλλακτικά προϊόντα μπορούν να επωφεληθούν από το σενάριο γεφύρωσης κενών, αλλά μόνο αφού εγκαταλείψουν το Microsoft Office από την ΕΑ τους κατά την επόμενη ανανέωση και την τελική πραγματικότητα" [38].

- Ros
- Matlab
- Gazebo

Η προσομοίωση ρομπότ είναι ένα βασικό εργαλείο στην εργαλειοθήκη κάθε ρομποτικό εργαλείο. Ένας καλά σχεδιασμένος προσομοιωτής καθιστά δυνατή την ταχεία δοκιμή αλγορίθμων, σχεδιασμό ρομπότ, εκτέλεση δοκιμών παλινδρόμησης, και εκπαίδευση συστήματος AI χρησιμοποιώντας ρεαλιστικά σενάρια. Το Gazebo προσφέρει τη δυνατότητα προσομοίωσης με ακρίβεια και αποτελεσματικότητα των πληθυσμών ρομπότ σε πολύπλοκα εσωτερικά και εξωτερικά περιβάλλοντα. Στα χέρια σας υπάρχει ένας ισχυρός μηχανισμός φυσικής, γραφικά υψηλής ποιότητας και βολικές προγραμματικές και γραφικές διεπαφές. Το καλύτερο από όλα, το Gazebo είναι δωρεάν με μια ζωντανή κοινότητα.

- **Vrep**

5.3 Μελλοντική Χρήση

Το ρομποτάκι Turtlebot και Kuka επιδέχεται βελτιώσεις και επεκτάσεις στο κώδικα καθώς το παραπάνω project αποτελεί ένα απλό παράδειγμα για την περιγραφή μερικών χαρακτηριστικών του VREP του Gazebo. Σαν επέκταση λοιπόν θα μπορούσαμε να τοποθετήσουμε κάποια εμπόδια πάνω σε ένα μεγαλύτερο χάρτη προσομοίωσης με περισσότερα αντικείμενα και το ρομπότ να μπορεί να τα εντοπίζει εγκαίρως και να τα αποφεύγει ή αφού έρθει σε σύγκρουση με αυτά να κάνει όπισθεν και να τα αποφεύγει συνεχίζοντας τη πορεία του. Ακόμη μπορούμε να κάνουμε χρήση περισσότερων αντικειμένων και να δούμε πως συμπεριφέρονται, πως λειτουργούν και σε τι μας εξυπηρετεί να τα χρησιμοποιήσουμε.

Κεφάλαιο 6

Βιβλιογραφία

ΒΙΒΛΙΑ

1. Bradski, G. & Kaehler, A. '*Learning OpenCV*'. O' Reilly, 2008.

2. Anis Koubaa "Robot Operating System (ROS): The Complete Reference, Τόμος 1"Springer International Publisching Switzerland,2016

3. Anis Koubaa "Robot Operating System (ROS): The Complete Reference, Τόμος 2"Springer International Publisching AG, part of Springer Nature 2018.

4. Anis Koubaa,Hachemi Bennaceur,Imen Chaari,Sahar Trigui,Adel Ammar, Mohamed Foued Sriti, Maram Alajlan, Imar Cheikhrouhou,Yasir Javed "Robot Path Planning and Cooperation,Foundations, Algorithms and Experimentations"Springer International Publisching Switzerland,2018

5. Anis Koubaa "Robot Operating System (ROS): The Complete Reference, Τόμος 2" Springer International Publisching AG, part of Springer Nature 2018.

6. Morgan Quigley, Brian Gerkey, William D. Smart "Programming Robots with ROS: A Practical Introduction to the Robot Operating System" Published O'Reilly Media, Inc,2015.

7. Wyatt Newman "A Systematic Approach to Learning Robot Programming with ROS" CRC Press, 2017

8. Lentin Joseph, Jonathan Cacace "Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System, 2nd Edition" Packt Publishing Ltd, 2018

9. Lentin Joseph" ROS Robotics Projects" Packt Publishing Ltd, 2017

10. Carol Fairchild, Dr. Thomas L. Harman "ROS Robotics By Example: Learning to control wheeled, limbed, and flying robots using ROS Kinetic Kame" Packt Publishing Ltd,2017

ΑΡΘΡΑ

11. K. Karteek Reddy,K.Praveen "PATH PLANNING USING VREP" International Journal of Research in Engineering and Technology, Department of Mechanical Engineering, M.V.G.R. College of Engineering, JNTU (Kakinada Vizianagaram, Andhra Pradesh, India,2013 προσπελάστηκε στις 10/06/2018

12. Lucas Nogueira" Comparative Analysis Between Gazebo and V-REP Robotic Simulators" Universidade de Campinas School of Electrical and Computer Engineering, 2014. προσπελάστηκε στις 10/06/2018

13. Travis Deyle http://www.hizook.com/ OmniRob: Kuka Robotics' Foray into Omnidirectional Mobile Manipulation Platforms προσπελάστηκε στις 10/06/2018

ΕΡΓΑΣΙΕΣ – ΔΙΑΤΡΙΒΕΣ

14. Γ. Ν. Αντώνιος Τζες, «Αυτόνομη Πλοήγηση Ρομποτικών Συστημάτων με Τεχνικές Ενισχυτικής Μάθησης»Τμήμα Πληροφορικής Πανεπιστήμιο Ιωαννίνων. προσπελάστηκε στις 20/01/2018

15. Σωτήτης Χολέβας «Ρομποτική Πλοήγηση και καταγραφή δεδομένων βασιζόμενη σε κάμερα»ΤΕΙ Ηρακλείου.Σελ.8-11 https://apothesis.lib.teicrete.gr/bitstream/handle/11713/4169/choleva2011.pdf?sequence=1&isAllowed=y προσπελάστηκε στις 20/01/2018

16. Jason M. O'Kane " A Gentle Introduction to ROS" University of South Carolina Department of Computer Science and Engineering, 2016 προσπελάστηκε στις 10/06/2018

17. Coppelia Robotics «Virtual Robot Experimentation Platform Building a clean model tutorial»Coppelia Robotics Version 3.5.0 http://www.coppeliarobotics.com/helpFiles/ προσπελάστηκε στις 10/06/2018

ΠΗΓΕΣ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ

18. http://imm.demokritos.gr/platon/AEOAAUAC_OOIOO_AOOIIAOEOIIOO/aeoaauac_ooioo_aooiiaoeoiioo.html 12/5/18» προσπελάστηκε στις 20/01/2018

19. http://legacy.python.org/dev/peps/pep-0373/ προσπελάστηκε στις προσπελάστηκε στις 10/06/2018

20. http://gazebosim.org/tutorials προσπελάστηκε στις προσπελάστηκε στις 20/01/2018

21. http://wiki.ros.org/indigo#Installation προσπελάστηκε στις 10/06/2018

22. https://en.wikipedia.org/wiki/Robot_Operating_System προσπελάστηκε στις 10/06/2018

23. http://wiki.ros.org/indigo/Migration προσπελάστηκε στις 10/06/2018

24. http://wiki.ros.org/ROS/Tutori http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironmentals προσπελάστηκε στις 10/06/2018

25. http://wiki.ros.org/ROS/Tutorials/NavigatingTheFilesystem προσπελάστηκε στις 10/06/2018

26. http://wiki.ros.org/ROS/Tutorials/CreatingPackage προσπελάστηκε στις 10/06/2018

27. http://wiki.ros.org/ROS/Tutorials/BuildingPackages προσπελάστηκε στις 10/06/2018

28. http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics προσπελάστηκε στις 10/06/2018

29. http://wiki.ros.org/ROS/Tutorials/UnderstandingServicesParams προσπελάστηκε στις 10/06/2018

30. http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch προσπελάστηκε στις 10/06/2018

31. http://wiki.ros.org/ROS/Tutorials/UsingRosEd προσπελάστηκε στις 10/06/2018

32. http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv προσπελάστηκε στις 10/06/2018

33. http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29 προσπελάστηκε στις 10/06/2018

34. http://wiki.ros.org/ROS/Tutorials/ExaminingPublisherSubscriber

35. http://wiki.ros.org/ROS/Tutorials/ExaminingServiceClient προσπελάστηκε στις 10/06/2018

36. http://wiki.ros.org/ROS/Tutorials/ExaminingServiceClient προσπελάστηκε στις 10/06/2018

37. http://wiki.ros.org/ROS/Tutorials/Getting%20s http://wiki.ros.org/roswtftarted%20with%20roswtf προσπελάστηκε στις 10/06/2018

38. http://wiki.ros.org/roswtf/Plugins προσπελάστηκε στις 10/06/2018

39. http://learn.turtlebot.com/ προσπελάστηκε στις 10/06/2018

40. http://learn.turtlebot.com/2015/02/03/1/ προσπελάστηκε στις 10/06/2018

41. http://learn.turtlebot.com/20 http://learn.turtlebot.com/2015/02/03/3/15/02/03/2/ προσπελάστηκε στις 10/06/2018

42. http://learn.turtlebot.com/2015/02/03/3/ προσπελάστηκε στις 10/06/2018

43. http://learn.turtlebot.com/2015/02/03/4/ προσπελάστηκε στις 10/06/2018

44. http://learn.turtlebot.com/2015/02/03/5/ προσπελάστηκε στις 10/06/2018

45. http://learn.turtlebot.com/2015/02/03/6/ προσπελάστηκε στις 10/06/2018

46. http://learn.turtlebot.com/2015/02/03/7/ προσπελάστηκε στις 10/06/2018

47. http://learn.turtlebot.com/2015/02/03/8/ προσπελάστηκε στις 10/06/2018

48. http://learn.turtlebot.com/2015/02/03/9/ προσπελάστηκε στις 10/06/2018

49. http://learn.turtlebot.com/2015/02/03/10/ προσπελάστηκε στις 10/06/2018

50. https://github.com/markwsilliman/turtlebot/ προσπελάστηκε στις 10/06/2018

51. http://learn.turtlebot.com/2015/02/03/11/ προσπελάστηκε στις 10/06/2018

52. http://learn.turtlebot.com/2015/02/04/1/ προσπελάστηκε στις 10/06/2018

53. http://learn.turtlebot.com/2015/02/04/2/ προσπελάστηκε στις 10/06/2018

54. http://learn.turtlebot.com/2015/02/04/3/ προσπελάστηκε στις 10/06/2018

55. http://learn.turtlebot.com/2015/02/04/4/ προσπελάστηκε στις 10/06/2018

56. http://learn.turtlebot.com/2015/02/04/5/ προσπελάστηκε στις 10/06/2018

57. http://learn.turtlebot.com/2015/02/04/6/ προσπελάστηκε στις 10/06/2018

58. http://learn.turtlebot.com/2015/02/04/7/ προσπελάστηκε στις 10/06/2018

59. http://learn.turtlebot.com/2015/02/04/8/ προσπελάστηκε στις 10/06/2018

60. http://learn.turtlebot.com/2015/0 προσπελάστηκε στις 10/06/2018

61. http://learn.turtlebot.com/2015/02/10/99/2/04/9/ προσπελάστηκε στις 10/06/2018

62. http://learn.turtlebot.com/2015/02/10/99/ προσπελάστηκε στις 10/06/2018

63. http://wiki.ros.org/turtlebot_stage/Tutorials/indigo/Bring%20up%20TurtleBot%20in%20stage προσπελάστηκε στις 10/06/2018

64. http://wiki.ros.org/turtlebot_stage/Tutorials/indigo/Customizing%20the%20Stage%20Simulator προσπελάστηκε στις 10/06/2018

65. http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Gazebo%20Bringup%20Guide προσπελάστηκε στις 10/06/2018

66. http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Explore%20the%20Gazebo%20world προσπελάστηκε στις 10/06/2018

67. http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Make%20a%20map%20and%20navigate%20with%20it προσπελάστηκε στις 10/06/2018

68. http://wiki.ros.org/turtlebot/Tutorials/indigo/Customising%20the%20Turtle προσπελάστηκε στις 10/06/2018

69. http://wiki.ros.org/turtlebot/Tutorials/indigo/Create%20your%20First%20Rapp προσπελάστηκε στις 10/06/2018

70. http://wiki.ros.org/turtlebot/Tutorials/indigo/Create%20your%20First%20Interaction προσπελάστηκε στις 10/06/2018

71. http://wiki.ros.org/turtlebot/Tutorials/indigo/Adding%20New%203D%20Sensor προσπελάστηκε στις 10/06/2018

72. http://wiki.ros.org/turtlebot/Tutorials/indigo/Adding%20a%20lidar%20to%20the%20turtlebot%20using%20hector_models%20%28Hokuyo%20UTM-30LX%29 προσπελάστηκε στις 20/01/2018 προσπελάστηκε στις 10/06/2018

73. https://www.mathworks.com/ προσπελάστηκε στις 10/06/2018

74. https://www.mathworks.com/help/robotics/ref/robotics.prm-class.html?s_tid=srchtitle προσπελάστηκε στις 10/06/2018

75. https://www.mathworks.com/help/robotics/examples/path-planning-in-environments-of-different-complexity.html προσπελάστηκε στις 10/06/2018

76. https://www.mathworks.com/help/robotics/ug/probabilistic-roadmaps-prm.html προσπελάστηκε στις 10/06/2018
77. https://www.mathworks.com/help/robotics/ref/robotics.prm.findpath.html προσπελάστηκε στις 10/06/2018
78. https://www.mathworks.com/help/robotics/ref/robotics.prm.update.html προσπελάστηκε στις 10/06/2018
79. https://www.mathworks.com/help/robotics/ref/robotics.montecarlolocalization-system-object.html?searchHighlight=monte%20carlo%20robotics&s_tid=doc_srchtitle προσπελάστηκε στις 10/06/2018
80. https://www.mathworks.com/help/robotics/examples/localize-turtlebot-using-monte-carlo-localization.html προσπελάστηκε στις 10/06/2018
81. https://www.kuka.com/en-cn/products/mobility/mobile-robot-systems/kmr-iiwa/ προσπελάστηκε στις 10/06/2018
82. http://edurobotics.weebly.com/gammaepsilonnuiotakappa940-eta-rhoomicronmupiomicrontauiotakappa942.html προσπελάστηκε στις 10/06/2018