
Προσομοίωση του Ρομποτικού Οχήματος Εξωτερικού Χώρου reDevil



**ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ**

Σχολή Μηχανικών Παραγωγής και Διοίκησης

Ελισάβετ Ηλιοπούλου

Επιβλέπων Καθηγητής: Τσουρβελούδης Νικόλαος

Χανιά, 2018

Περιεχόμενα

Περιεχόμενα	3
Κεφάλαιο 1 : Εισαγωγή	4
1.1 Ρομποτικά οχήματα εξωτερικού χώρου	4
1.2 Ρομποτικό όχημα ATRV-mini	4
1.3 Ανακαίνιση του ρομποτικού οχήματος (reDevil)	6
1.4 Διατύπωση προβλήματος	6
1.5 Ρομποτική ανοιχτού κώδικα	6
1.6 ROS (Robot Operating System)	7
1.7 Πακέτα ROS (Open Source ROS Packages)	9
1.8 Πλεονεκτήματα προγραμματισμού ρομπότ με ROS	11
1.9 Ρομποτική προσομοίωση	12
1.10 Γραφικό περιβάλλον Rviz	12
1.11 Προσομοιωτής Gazebo	13
1.12 Σχέση Gazebo και ROS	14
Κεφάλαιο 2. Προσομοίωση	15
2.1 Αναλυτική περιγραφή του Gazebo	15
2.2 Σχεδίαση τριδιάστατου μοντέλου	18
2.3 Προσθήκη εμφυτευμάτων κώδικα (Plugins)	21
2.3.1 Αισθητήρες Υπερήχων	21
2.3.2 IMU	24
2.3.3 Συσκευή Εικονοληψίας	26
2.3.4 Global Positioning System (GPS)	26
2.3.5 Κίνηση με Ολίσθηση Τροχών (Skid Steering)	28
2.4 Προγραμματισμός ρομποτικού ελεγκτή	31
2.4.1 Έλεγχος του οχήματος με PID Controllers	31
2.4.2 Πλοήγηση με τη χρήση Τηλεχειρισμού (Keyboard Teleoperation)	33
Κεφάλαιο 3: Ρομποτικά οχήματα εξωτερικού χώρου διαθέσιμα στο Gazebo.	38
Κεφάλαιο 4: Συμπεράσματα	43
Βιβλιογραφία	45

Κεφάλαιο 1 : Εισαγωγή

1.1 Ρομποτικά οχήματα εξωτερικού χώρου

Τα κινητά ρομπότ είναι σύγχρονες “έξυπνες” κατασκευές οι οποίες έχουν τη δυνατότητα να μετακινούνται και να αλληλεπιδρούν με το περιβάλλον τους.

Τα έντροχα ρομπότ εξωτερικού χώρου (ρομποτικά οχήματα) εμφανίζονται και χρησιμοποιούνται σε διάφορους κλάδους όπως, στην ψυχαγωγία, στην εκτέλεση ορισμένων οικιακών εργασιών, σε στρατιωτικές αποστολές, σε αποστολές αναζήτησης εγκλωβισμένων από σεισμούς σε σημεία στα οποία δεν μπορεί να φθάσει άνθρωπος, σε εξέταση ή και ανατίναξη εκρηκτικών μηχανισμών, ή για εκπαιδευτικούς και ερευνητικούς σκοπούς. Αποτελούν επίσης το επίκεντρο μεγάλης έρευνας η οποία πραγματοποιείται στα ερευνητικά τμήματα διάφορων εταιρειών, και σχεδόν όλα τα μεγάλα πανεπιστήμια έχουν ένα ή περισσότερα εργαστήρια που επικεντρώνονται στην έρευνα και την κατασκευή ρομποτικών οχημάτων.

1.2 Ρομποτικό όχημα ATRV-mini

Το ρομπότ ATRV-mini είναι ένα έντροχο ρομπότ σχεδιασμένο για χρήση σε εξωτερικούς χώρους. Διαθέτει τέσσερις τροχούς και δύο κινητήρες συνδεδεμένους σε κάθε ζεύγος τροχών.



Εικόνα 1. 1: ATRV - mini

Τα τεχνικά χαρακτηριστικά του ATRV- mini παρουσιάζονται στον Πίνακα 1 [13].

Διαστάσεις	62x53x45 cm
Απόσταση από το έδαφος	7.62 cm
Βάρος	38.6 kg
Ταχύτητα	0 – 1.5 m/sec
Κίνηση	4 τροχοί
Γωνία στροφής	Μηδενική (γύρω απο το κέντρο)
Τροχοί	25.4 mm
Σύστημα ελέγχου	RWI rFLEX System
Κινητήρες	Δύο 0.10 HP, 12V DC
Υπολογιστής	Pentium III EBX
Λογισμικό	RWI Mobility Robot Infrastructure
I/O	Ethernet, Rs-232, Joystick
Αισθητήρες	24 Sonar (6 σε κάθε μία απο τις τέσσερις γωνίες) Camera (Sony EVI D30) GPS receiver IMU Ασύρματη επικοινωνία

Πίνακας 1: Τεχνικά χαρακτηριστικά από το εγχειρίδιο χρήσης του ATRV - mini.

1.3 Ανακαίνιση του ρομποτικού οχήματος (reDevil)

Το 2017 το όχημα ανακαινίσθηκε και μετονομάσθηκε σε reDevil. Για την κίνηση του οχήματος χρησιμοποιούνται [16]:

-) δύο κινητήρες 24V
-) κύκλωμα ελέγχου κινητήρα Sabertooth 2x12A για την επικοινωνία του ελεγκτή με τους κινητήρες.

Χρησιμοποιεί τους παρακάτω αισθητήρες:

-) αισθητήρες υπερήχων
-) κάμερα
-) GPS
-) IMU

Ως πηγή ισχύος χρησιμοποιήθηκαν δύο μπαταρίες των 12V σε σειρά.

1.4 Διατύπωση προβλήματος

Κρίθηκε απαραίτητη η προσομοίωση του ρομπότ εξωτερικού χώρου ATRV- mini ώστε να γίνουν πειραματισμοί πάνω στα δημιουργούμενα πακέτα και λογισμικά, για την αποφυγή φθοράς λόγω επανειλημμένης χρήσης του πραγματικού ρομπότ και ατυχημάτων λόγω σφαλμάτων ή κακού χειρισμού. Η εφαρμογή των δημιουργούμενων πακέτων πάνω στο προσομοιωμένο όχημα, μας δίνει τη δυνατότητα να κάνουμε μια ολοκληρωμένη έρευνα στην πράξη.

1.5 Ρομποτική ανοιχτού κώδικα

Η ρομποτική ανοιχτού κώδικα (OSR) υποστηρίζει την ανάπτυξη, τη διανομή και την υιοθέτηση ανοιχτού λογισμικού και υλικού για χρήση στη ρομποτική έρευνα, την εκπαίδευση και την ανάπτυξη προϊόντων. Αυτός ο κλάδος της ρομποτικής χρησιμοποιεί

δωρεάν λογισμικό ανοιχτού κώδικα παρέχοντας σχέδια, μοντέλα και αρχεία κώδικα σε άλλους χρήστες [19].

1.6 ROS (Robot Operating System)

Το ROS (Robot Operating System) είναι ένα ευέλικτο σύστημα για τη σύνταξη λογισμικού ρομπότ [1]. Είναι ένα meta-operating σύστημα, δηλαδή δεν παρέχει μόνο τις τυπικές υπηρεσίες λειτουργικού συστήματος, αλλά και λειτουργίες υψηλού επιπέδου (ασύγχρονες και σύγχρονες κλήσεις, κεντρικές βάσεις δεδομένων και συστήματα διαμόρφωσης ρομπότ κ.λπ.). Έχει ευρεία εφαρμογή στο χώρο της ρομποτικής γιατί είναι ένα λογισμικό ανοιχτού κώδικα (open source).

Το ROS πρωτοεμφανίστηκε στο Πανεπιστήμιο του Στάνφορντ πριν η ρομποτική εταιρεία Willow Garage αναπτύξει επίσημα το ROS το 2007. Το ROS παρείχε υπηρεσίες βασικού λειτουργικού συστήματος και προσέφερε βασικούς ελεγκτές για χαρτογράφηση, εντοπισμό, πλοήγηση, σε έναν αυξανόμενο αριθμό ερευνητών. Το 2013, μια ανεξάρτητη μη κερδοσκοπική οργάνωση που ονομάζεται Open Source Robotics Foundation (OSRF) ανέλαβε τη διαχείριση του ROS, για να στηρίξει τη σωστή ανάπτυξη λογισμικού ανοιχτού κώδικα για χρήση στη ρομποτική έρευνα, και στην ανάπτυξη προϊόντων.

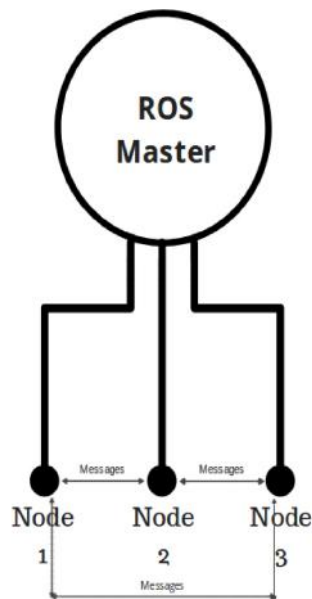
Πρόκειται για μια συλλογή εργαλείων λογισμικού και βιβλιοθηκών που χρησιμοποιούνται από προγραμματιστές ρομπότ για την ανάπτυξη εφαρμογών. Το ROS δεν είναι λειτουργικό σύστημα, αλλά χρησιμοποιεί το περιβάλλον χρήστη του λειτουργικού συστήματος, το σύστημα αρχείων, το σύστημα διαχείρισης εργασιών και τα βοηθητικά προγράμματα προγραμματισμού. Το πιο διαδεδομένο λειτουργικό σύστημα για το ROS είναι το Linux, ακολουθούμενο από Java ή MacOS.

Προσφέρει την υπηρεσία επικοινωνίας μεταξύ εργασιών που εκτελούνται για τη λειτουργία του ρομπότ, με τη χρήση μηνυμάτων (Publish/Subscribe) η οποία επιτρέπει στο Gazebo να επικοινωνεί με εργασίες (nodes) μέσω των θεμάτων εργασιών (topics). Το κάθε θέμα έχει συγκεκριμένο τύπο, ανάλογα με το είδος της εργασίας και του μηνύματος. Στην Εικόνα 1. 2 βλέπουμε τη λίστα των ενεργών εργασιών, καθώς το προσομοιωμένο όχημα έχει τεθεί σε λειτουργία.

```
/clock
/cmd_vel
/fix
/fix/position/parameter_descriptions
/fix/position/parameter_updates
/fix/status/parameter_descriptions
/fix/status/parameter_updates
/fix/velocity/parameter_descriptions
/fix/velocity/parameter_updates
/fix_velocity
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/imu_data
/joint_states
/odom
/robot/camera1/camera_info
/robot/camera1/image_raw
/robot/camera1/image_raw/compressed
/robot/camera1/image_raw/compressed/parameter_descriptions
/robot/camera1/image_raw/compressed/parameter_updates
/robot/camera1/image_raw/compressedDepth
/robot/camera1/image_raw/compressedDepth/parameter_descriptions
/robot/camera1/image_raw/compressedDepth/parameter_updates
/robot/camera1/image_raw/theora
/robot/camera1/image_raw/theora/parameter_descriptions
/robot/camera1/image_raw/theora/parameter_updates
/robot/camera1/parameter_descriptions
/robot/camera1/parameter_updates
/rosout
/rosout_agg
/sensor/scnar1
/sensor/scnar1/parameter_descriptions
/sensor/scnar1/parameter_updates
/sensor/scnar2
/sensor/scnar2/parameter_descriptions
/sensor/scnar2/parameter_updates
/sensor/scnar3
/sensor/scnar3/parameter_descriptions
/sensor/scnar3/parameter_updates
/sensor/scnar4
/sensor/scnar4/parameter_descriptions
/sensor/scnar4/parameter_updates
/tf
```

Εικόνα 1. 2: Λίστα των ενεργών εργασιών

Η διαχείριση των εργασιών και των θεμάτων εργασιών γίνεται από το ROS Master, το κεντρικό σύστημα, το οποίο επιτρέπει στις εργασίες να επικοινωνούν μεταξύ τους, μέσω μηνυμάτων. Το σχεδιάγραμμα της επικοινωνίας μεταξύ των εργασιών και του ROS Master φαίνεται στην Εικόνα 1.3.

**Εικόνα 1. 3: ROS Master**

Ένα σημαντικό πλεονέκτημα του ROS είναι η δυνατότητα επιλογής γλώσσας προγραμματισμού, για την ανάπτυξη του πηγαίου κώδικα. Οι διαθέσιμες γλώσσες προγραμματισμού είναι οι C++, Python και Lisp, ενώ υπάρχουν πειραματικές βιβλιοθήκες σε Java και Lua. Επίσης, καθώς η δημιουργία πηγαίου κώδικα για ρομποτικές εφαρμογές μπορεί να γίνει αρκετά δύσκολη, η χρήση ήδη υπαρχόντων πακέτων, που προσφέρει η κοινότητα του ROS, διευκολύνει πολύ τη διαδικασία. Για την υλοποίηση της διαδικασίας ελέγχου και κίνησης του οχήματος, χρησιμοποιήθηκαν διάφορα πακέτα που προσφέρονται από το ROS.

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε το ROS για τον έλεγχο του ρομποτικού οχήματος ATRV-mini στο περιβάλλον προσομοίωσης Gazebo, το οποίο θα αναλυθεί παρακάτω.

1.7 Πακέτα ROS (Open Source ROS Packages)

Όπως αναφέρθηκε παραπάνω, το ROS επιτρέπει τη χρήση διαφόρων πακέτων ανοιχτού κώδικα, για ρομποτικές εφαρμογές. Τα πακέτα είναι βιβλιοθήκες (libraries), αρχεία κώδικα (scripts), configuration files, και άλλα. Σε περιπτώσεις εμπορικών

εφαρμογών ο χρήστης είναι υποχρεωμένος να πάρει έγκριση από τον αντίστοιχο δημιουργό του κάθε πακέτου. Χάρης σε αυτά τα πακέτα μπορούμε να χρησιμοποιήσουμε διάφορα εργαλεία, χωρίς να συντάξουμε όλον τον κώδικα από την αρχή.

Η δομή του ROS απαρτίζεται από πακέτα. Όλα αυτά βοηθούν ώστε να δημιουργηθούν τα εμφυτεύματα (plugins), τα οποία μας δίνουν διάφορες δυνατότητες και εργαλεία για τον έλεγχο και την κίνηση του ρομποτικού οχήματος, καθώς και για σύνθετες συμπεριφορές του. Τα ήδη υπάρχοντα πακέτα μπορούμε να τα βρούμε στα αποθετήρια (repositories), που είναι διαδικτυακοί χώροι αποθήκευσης. Σαφώς είναι ελεύθερη η δημιουργία εκ νέου πακέτων, ή η πρόταση για βελτίωση των υπαρχόντων [15].

Ένα πακέτο ROS έχει την παρακάτω δομή:

- **include/**όνομα πακέτου: Εδώ περιέχονται **header files** όταν το πακέτο περιγράφεται σε γλώσσα C++. Τα αρχεία αυτά περιγράφουν τα εμφυτεύματα κώδικα που χρειάστηκαν για τις λειτουργίες του ρομπότ.
- **src/** όνομα πακέτου: Αρχεία **py**, όταν το πακέτο περιγράφεται σε γλώσσα Python.
- **msg/** όνομα αρχείου μηνυμάτων: Περιέχει μηνύματα που χρειάζονται για τη δημοσίευση (publish) από το ROS σε κάποιο θέμα μιας ενεργής εργασίας.
- **scripts/** όνομα εκτελέσιμων αρχείων κώδικα. Στα αρχεία αυτά περιγράφονται οι λειτουργίες των διαφόρων εργασιών που χρειάστηκαν για τον έλεγχο του οχήματος.
- **package.xml**: Πρέπει να περιλαμβάνεται σε οποιοδήποτε πακέτο. Αυτό το αρχείο ορίζει ιδιότητες σχετικά με το πακέτο, όπως το όνομα του πακέτου, η έκδοση, οι συγγραφείς, και τα dependencies.
- **CMakeLists.txt**: Κάθε πακέτο πρέπει να περιέχει ένα ή περισσότερα αρχεία CMakeLists.txt. Περιγράφουν τον τρόπο κατασκευής του πακέτου και το σημείο εγκατάστασης του.

1.8 Πλεονεκτήματα προγραμματισμού ρομπότ με ROS

Παλαιότερα, ο προγραμματισμός ρομπότ ήταν πιο δύσκολος, χρονοβόρος, και χωρίς όλες τις δυνατότητες που προσφέρονται σήμερα. Ο χρήστης, αγόραζε ένα νέο ρομπότ και ξόδευε αμέτρητες ώρες για να υπολογίσει πώς να το μετακινήσει. Έχοντας δουλέψει αρκετά ώστε το λογισμικό και οι δυνατότητες της έρευνας να αρχίσουν να αναπτύσσονται, προέκυπταν νέες ιδέες, και σκέψεις για πρόσθετες δυνατότητες και πειράματα. Δυστυχώς, ήταν συχνά αδύνατη η επέκταση πέρα από τις αρχικές δυνατότητες του ρομποτικού συστήματος, και ο πυρήνας του λειτουργικού συστήματος δεν ήταν ανοικτός σε τροποποιήσεις. Τότε χρειαζόταν είτε η άδεια του κατασκευαστή για να χορηγήσει πρόσβαση στο λογισμικό του, είτε η εκκίνηση από μια νέα ρομποτική πλατφόρμα ή σύστημα. Έτσι, κάθε φορά που ένας ερευνητής ήθελε να δοκιμάσει κάτι καινούριο, σπαταλούσε χρόνο και προσπάθεια, επιβραδύνοντας την καινοτομία στον τομέα.

Τα τελευταία χρόνια, ο προγραμματισμός ρομπότ έχει γίνει πιο φιλικός για τους χρήστες. Τα συστήματα ανοιχτού κώδικα άλλαξαν το προσκήνιο για τους ερευνητές αντιμετωπίζοντας ένα σημαντικό εμπόδιο στην ανάπτυξη της ρομποτικής.

Αρχικά, χρησιμοποιούνταν δύο συστήματα για εμφάνιση μηνυμάτων πραγματικού χρόνου και ταξινόμηση δεδομένων, τα οποία αν και χρησιμοποιούνται ακόμα, δεν έχουν αρκετό διαθέσιμο λογισμικό. Αυτά είναι τα Lightweight Communications and Marshalling (LCM) και Yet Another Robot Platform (YARP). Το LCM δημιουργήθηκε το 2006 για την ομάδα MIT DARPA Urban Challenge, και από τότε έχει χρησιμοποιηθεί σε πολλές ρομποτικές εφαρμογές και αυτόματα συστήματα, στα πεδία της έρευνας αλλά και της παραγωγής. Το YARP δημιουργήθηκε το 2002, και είναι ένα πακέτο λογισμικού ανοιχτού κώδικα, σε γλώσσα C++, για την διασύνδεση αισθητήρων, επεξεργαστών και ενεργοποιητών.

Το ROS (Robot Operating System) είναι ένα ανοιχτού κώδικα, εγκεκριμένο από την BSD, ευέλικτο σύστημα για τη σύνταξη λογισμικού ρομπότ που προσφέρει στους χρήστες έναν τρόπο να δημιουργήσουν γρήγορα, να διατηρήσουν και να επεκτείνουν τις δυνατότητες των ρομπότ τους. Αυτή η συλλογή εργαλείων και βιβλιοθηκών παρέχει υπηρεσίες λειτουργικού συστήματος που επιτρέπουν τον έλεγχο υλικού (hardware) και προσομοίωσης. Σε αντίθεση με τα περισσότερα ιδιόκτητα λογισμικά, το ROS επιτρέπει στους ερευνητές να αναπτύξουν εμφυτεύματα κώδικα και συμπληρωματικούς ελεγκτές για ένα ευρύ φάσμα εργασιών. Ο συλλογικός χαρακτήρας της κοινότητας είναι ζωτικής σημασίας, καθώς τα μέλη ειδικεύονται σε διαφορετικούς τομείς και δημοσιεύουν

πακέτα και μοντέλα για κοινή χρήση. Στο συμβατικό έλεγχο ενός ρομπότ, η έλλειψη υποστήριξης από τον κατασκευαστή για την αντιμετώπιση προβλημάτων δημιουργούσε πρόβλημα και επιβράδυνε την ανάπτυξη του λογισμικού. Στο σύγχρονο έλεγχο ρομπότ, υπάρχει υποστήριξη από μια ενεργή και συνεργατική κοινότητα.

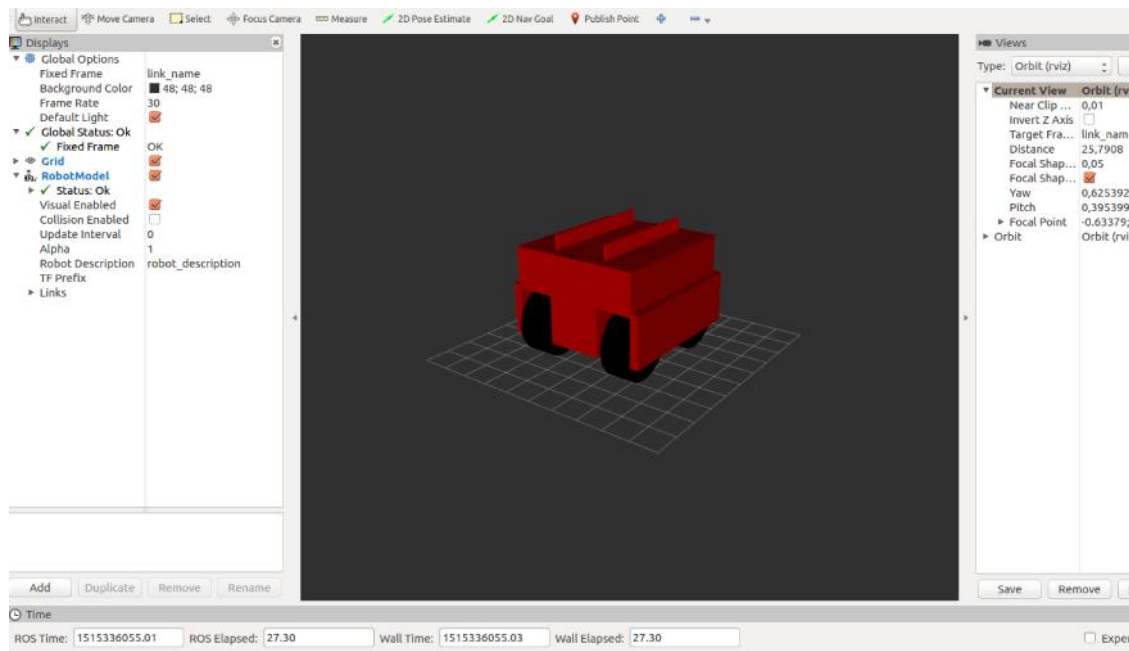
Ακόμα, το ROS επιτρέπει στους ερευνητές να κατασκευάσουν πιο σύνθετα ρομπότ σε πολύ μικρότερο χρονικό διάστημα. Τα τελευταία χρόνια οι βιβλιοθήκες με τα διαθέσιμα ρομπότ έχουν επεκταθεί. Συνδέοντας το υλικό (hardware) με το λογισμικό (software) με απλοποιημένο τρόπο, το ROS παρέχει πλοήγηση, ανίχνευση εμποδίων και τηλεχειρισμό. Προσφέρει ευκολία για τους ερευνητές να εξοπλίζουν γρήγορα τα ρομπότ με βασικές λειτουργίες, ώστε να μπορούν να επικεντρωθούν σε πιο ουσιαστικό πειραματικό έργο. Για παράδειγμα, κάποια ρομπότ έχουν βοηθήσει τετραπληγικά άτομα να διερευνήσουν ξανά τον κόσμο τους, να εκτελούν οικιακές δουλειές, έχουν βοηθήσει αστροναύτες στην εκτέλεση πειραμάτων στο διάστημα, και πολλά άλλα.

1.9 Ρομποτική προσομοίωση

Η ρομποτική προσομοίωση είναι η φωτορεαλιστική αναπαράσταση ενός ρομπότ με την προσομοίωση των φυσικών ιδιοτήτων του περιβάλλοντος, με την υλοποίηση λογισμικού πραγματικού χρόνου, προσομοιωμένες μετρήσεις των αισθητήρων, και δυνατότητες επικοινωνίας και καταγραφής δεδομένων. Επιτρέπει δοκιμές και πειραματισμούς, σε ρομποτικές εφαρμογές, που στην πράξη θα ήταν ακριβοί και χρονοβόροι. Επιπλέον, επιτρέπεται η ανάπτυξη λογισμικού πάνω σε αυτές τις εφαρμογές, από κοινότητες προγραμματιστών, χωρίς να υπάρχει ο κίνδυνος καταστροφής τους.

1.10 Γραφικό περιβάλλον Rviz

Ο τριδιάστατος οπτικοποιητής Rviz είναι ένα γραφικό εργαλείο του ROS για την εμφάνιση δεδομένων αισθητήρων και πληροφοριών κατάστασης του προσομοιωμένου ρομπότ. Υπάρχει η δυνατότητα, να εμφανίσουμε σε πραγματικό χρόνο τιμές αισθητήρων που προέρχονται από το ROS, όπως δεδομένα κάμερας, μετρήσεις απόστασης υπέρυθρων, δεδομένα σόναρ και πολλά άλλα. Στην Εικόνα 1.4 φαίνεται το μοντέλο reDevil στο περιβάλλον του Rviz.



Εικόνα 1. 4: Το reDevil στο περιβάλλον του Rviz

1.11 Προσομοιωτής Gazebo

Το Gazebo είναι ένα γραφικό περιβάλλον, το οποίο προσφέρει τη δυνατότητα τριδιάστατης προσομοίωσης. Λειτουργεί σε περιβάλλον Linux και υποστηρίζεται από το ROS, ενώ είναι ανοιχτού κώδικα. Το Gazebo έχει τη δυνατότητα προσομοίωσης πολύπλοκων ρομποτικών σεναρίων σε εσωτερικά και εξωτερικά περιβάλλοντα [21].

Τυπικές χρήσεις του Gazebo περιλαμβάνουν:

- ⌋ δοκιμές αλγορίθμων,
- ⌋ σχεδιασμό ρομποτικών οχημάτων, βραχιόνων, ανθρωποειδών ρομπότ, κτηρίων, αισθητήρων και άλλων,
- ⌋ εκτέλεση δοκιμών παλινδρόμησης με ρεαλιστικά σενάρια.

Μερικά βασικά χαρακτηριστικά του Gazebo περιλαμβάνουν:

- ⌋ προσομοίωση φυσικών ιδιοτήτων
- ⌋ πλούσια βιβλιοθήκη μοντέλων ρομπότ και χώρων

-) μεγάλη ποικιλία αισθητήρων
-) φιλικές προς το χρήστη προγραμματιστικές και γραφικές διεπαφές

Το Gazebo, έχει σχεδιαστεί για να αναπαράγει με ακρίβεια τα δυναμικά περιβάλλοντα που μπορεί να αντιμετωπίσει ένα ρομπότ. Όλα τα προσομοιωμένα αντικείμενα έχουν μάζα, ταχύτητα, τριβή και πολλές άλλες ιδιότητες που τους επιτρέπουν να συμπεριφέρονται ρεαλιστικά όταν ωθούνται, τραβιούνται, χτυπιούνται ή μεταφέρονται. Αυτές οι ενέργειες μπορούν να χρησιμοποιηθούν ως αναπόσπαστα μέρη ενός πειράματος. Τα ίδια τα ρομπότ είναι δυναμικές δομές που αποτελούνται από άκαμπτα σώματα που συνδέονται μέσω αρθρώσεων. Οι δυνάμεις, τόσο γωνιακές όσο και γραμμικές, μπορούν να εφαρμοστούν στις επιφάνειες και τις αρθρώσεις για να δημιουργήσουν μετακίνηση και αλληλεπίδραση με το περιβάλλον. Ο κόσμος στο γραφικό περιβάλλον του Gazebo μπορεί να συμπεριλάβει δρόμους, κτήρια, γρασίδι και άλλα αντικείμενα που έχουν δημιουργηθεί από τους χρήστες του. Είναι επίσης δυνατόν να ελεγχθούν, συνθήκες φωτισμού, συντελεστές τριβής, και άλλα. Ακόμα, το Gazebo διαθέτει μια ενεργή ηλεκτρονική βάση, με χρήστες που εξελίσσουν συνεχώς τα πακέτα για να καλύψουν τις μεταβαλλόμενες ανάγκες στον κόσμο των ρομπότ.

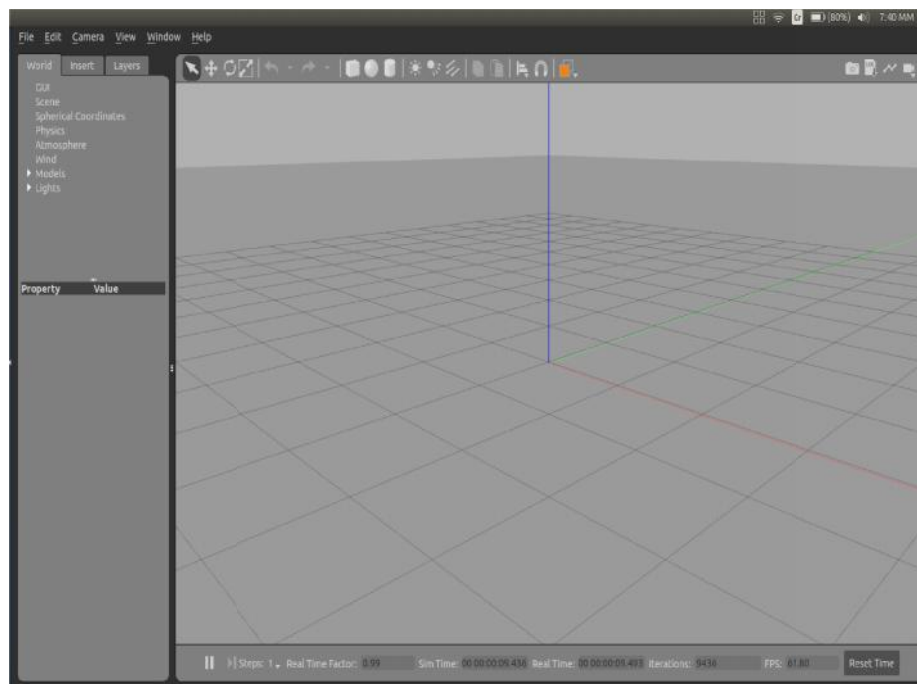
1.12 Σχέση Gazebo και ROS

Στα πλαίσια της ρομποτικής προσομοίωσης και του χρησιμοποιούμενου λογισμικού, το ROS συνεργάζεται με το Gazebo. Ένα από τα κύρια πλεονεκτήματα της σύνδεσης του Gazebo με το ROS, είναι η συνεχής ροή δεδομένων πραγματικού χρόνου, και η ταυτόχρονη γραφική αναπαράσταση των εργασιών του ρομπότ. Επιπρόσθετα, η δυνατότητα χρήσης διερμηνευμένου κώδικα (scripts) μας δίνει πολλές δυνατότητες κίνησης του ρομπότ και λειτουργίας του προσομοιωτή. Ένα σημαντικό εργαλείο στην επικοινωνία μεταξύ του Gazebo και του ROS, είναι το `roslaunch gazebo`. Σε επόμενο κεφάλαιο θα αναλυθούν οι ιδιότητες και η χρήση του.

Κεφάλαιο 2. Προσομοίωση

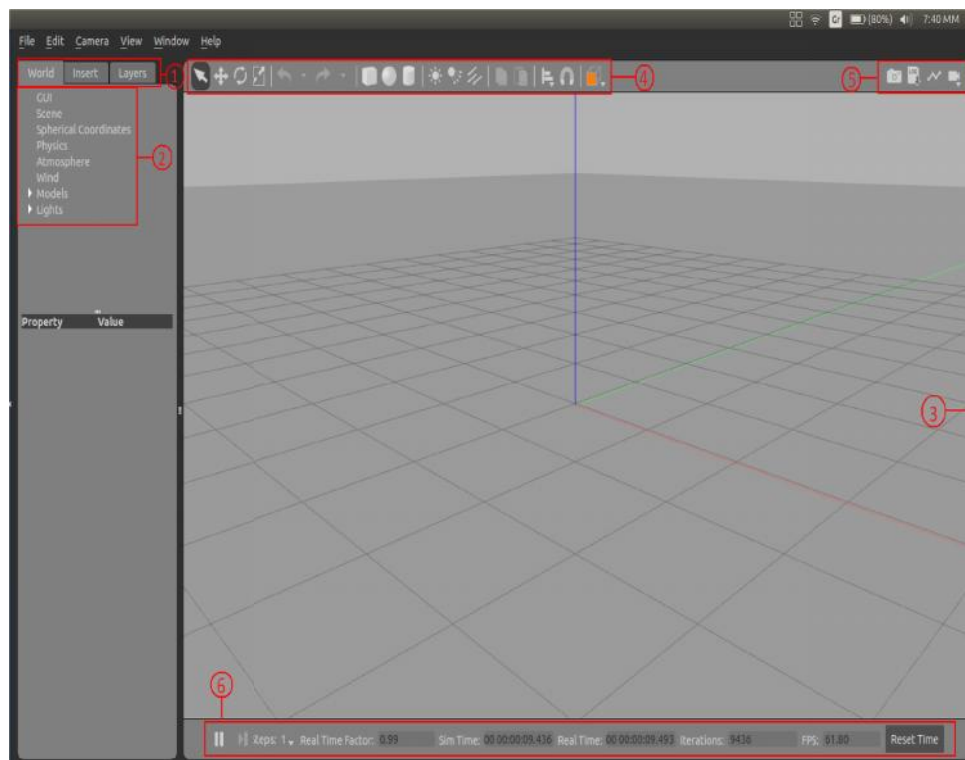
2.1 Αναλυτική περιγραφή του Gazebo.

Στην Εικόνα 2.1 βλέπουμε το γραφικό περιβάλλον, στο οποίο πραγματοποιήθηκαν οι δοκιμές του οχήματος.



Εικόνα 2. 1: Γραφικό περιβάλλον του Gazebo

Στην Εικόνα 2.2 βλέπουμε στο αριστερό πάνελ του παραθύρου (1) τις καρτέλες World, Insert, και Layers. Στην καρτέλα World (2) φαίνονται οι διάφορες επιλογές προσομοίωσης, όπως φυσικές ιδιότητες, χρησιμοποιούμενα μοντέλα, τρέχον περιβάλλον, και σφαιρικές συντεταγμένες. Στην καρτέλα Insert βρίσκονται τα διάφορα ήδη υπάρχοντα μοντέλα που μπορούμε να εντάξουμε στο γραφικό περιβάλλον και να χρησιμοποιήσουμε στην προσομοίωση, καθώς και τα μοντέλα που έχουμε δημιουργήσει. Η τρίτη καρτέλα του πάνελ, είναι το Layers που οργανώνει και εμφανίζει τις διαφορετικές ομάδες (στρώματα) απεικόνισης που είναι διαθέσιμες στη προσομοίωση. Ένα στρώμα μπορεί να περιέχει ένα ή περισσότερα μοντέλα. Αυτή είναι μια προαιρετική δυνατότητα, και στην υπάρχουσα προσομοίωση είναι κενή, καθώς μας απασχολεί μόνο το ρομποτικό όχημα.

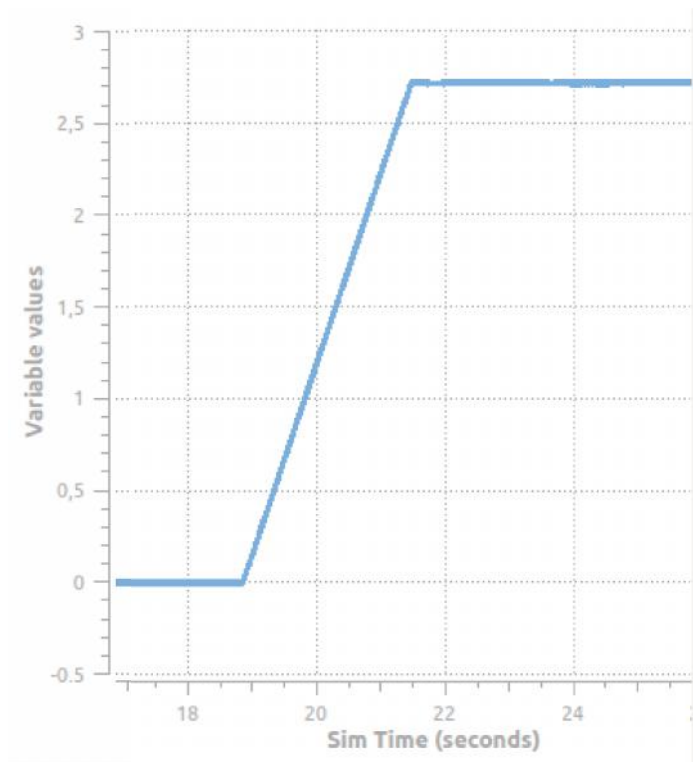


Εικόνα 2. 2: Αναλυτική περιγραφή του Gazebo

Στα δεξιά του παραθύρου, φαίνεται το δεξί πάνελ (3), το οποίο είναι κρυμμένο από προεπιλογή. Μπορεί να χρησιμοποιηθεί για την αλληλεπίδραση μας με τα κινητά μέρη ενός επιλεγμένου μοντέλου (τις αρθρώσεις). Εάν δεν υπάρχουν επιλεγμένα μοντέλα στη σκηνή, ο πίνακας δεν εμφανίζει καμία πληροφορία.

Επιπλέον, το γραφικό περιβάλλον, διαθέτει δύο γραμμές εργαλείων και ένα μενού. Η μία γραμμή εργαλείων (4), βρίσκεται στο πάνω μέρος της οθόνης, και περιλαμβάνει επιλογές μετακίνησης, περιστροφής, προοπτικής και μεγέθυνσης του μοντέλου, καθώς και ένταξη τριών βασικών τριδιάστατων σχημάτων στο περιβάλλον.

Επιπλέον, μας δίνεται η δυνατότητα κατασκευής γραφήματος των ενεργών εργασιών προς τον χρόνο προσομοίωσης, με την επιλογή Create Plot, που βρίσκεται στο δεξιό μέρος της γραμμής εργαλείων (5). Μπορούμε να επιλέξουμε δύο ή παραπάνω παραμέτρους από τη λίστα, ανάμεσα στα θέματα εργασιών (topics), εξαρτήματα του μοντέλου, καθώς και φυσικές ιδιότητες. Στη συνέχεια με την επιλογή Export, εξάγουμε το γράφημα σε μορφή αρχείου pdf ή csv. Παρακάτω (Εικόνα 2.3) βλέπουμε το γράφημα της γραμμικής ταχύτητας του οχήματος προς το χρόνο προσομοίωσης.



Εικόνα 2. 3: Γραφική αναπαράσταση της γραμμικής ταχύτητας σε συνάρτηση με το χρόνο

Οι αυξομειώσεις στο γράφημα, είναι οι πειραματισμοί στην ταχύτητα και την κατεύθυνση του οχήματος. Αυτές οι δύο παράμετροι θα αναλυθούν σε επόμενο κεφάλαιο.

Η δεύτερη γραμμή εργαλείων (6) της Εικόνας 2.2, βρίσκεται στο κάτω μέρος της οθόνης. Μας δείχνει το χρόνο προσομοίωσης, τον πραγματικό χρόνο, και τις επαναλήψεις που έχουν γίνει. Διαθέτει μία επιλογή παύσης της κίνησης της προσομοίωσης, και μία επιλογή επανεκκίνησης του χρόνου.

Το μενού στο ανώτερο μέρος της οθόνης, προσφέρει βασικά εργαλεία του αρχείου, όπως αποθήκευση, επιλογές εμφάνισης, εργασίες κόμβων, επανεκκίνηση του κόσμου και των θέσεων των μοντέλων, και άλλα. Μια από τις βασικές επιλογές του μενού, είναι τα εργαλεία Model Editor και Building Editor. Το Building Editor, όπως προκύπτει και από το όνομα του είναι κυρίως για τη δημιουργία κτηρίων. Στην περίπτωση δημιουργίας ρομποτικού οχήματος, χρησιμοποιείται το Model Editor. Ο χρήστης εισάγει στο γραφικό περιβάλλον είτε ένα από τα τρία βασικά τριδιάστατα

σχήματα, διαμορφώνοντας τα ανάλογα με τις προτιμήσεις του, είτε κάποιο από τα έτοιμα μοντέλα από την καρτέλα Insert στο αριστερό μέρος της οθόνης. Ταυτόχρονα, το Gazebo δημιουργεί ένα αρχείο κώδικα SDF (Simulation Description Format), δίνοντας μας τη δυνατότητα να επεξεργαστούμε το μοντέλο και εκτός γραφικού περιβάλλοντος, με τη βοήθεια κάποιου συντάκτη.

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε μία διαφορετική μορφή αρχείου, ώστε να υπάρχει πλήρης συνεργασία μεταξύ του ROS και του Gazebo, δηλαδή η σωστή λειτουργία των εμφυτευμάτων (plugins) και η ανταλλαγή μηνυμάτων μεταξύ των εργασιών και του ROS Master. Η σχεδίαση του μοντέλου αναλύεται παρακάτω.

2.2 Σχεδίαση τριδιάστατου μοντέλου

Για τη σχεδίαση του τριδιάστατου μοντέλου χρησιμοποιήθηκε κώδικας URDF (Universal Robotic Description Format) [11]. Ο κώδικας URDF είναι μια μορφή αρχείου XML που χρησιμοποιείται στο ROS για να περιγράψει όλα τα γεωμετρικά χαρακτηριστικά ενός ρομπότ. Για να χρησιμοποιηθεί ένα αρχείο URDF στο Gazebo, πρέπει να προστεθούν κάποιες ετικέτες που σχετίζονται με την προσομοίωση. Το URDF ορίζει τις κινηματικές και δυναμικές ιδιότητες ενός μοντέλου [12].

Η δομή του μοντέλου που αναπτύχθηκε για την προσομοίωση του reDevil, έχει την εξής μορφή:

) Περιγραφή των εξαρτημάτων, <links>.

Κάθε στοιχείο <link> πρέπει να ακολουθείται από ετικέτες που αφορούν

- ο την αδράνεια <inertial>...</inertial>,
- ο τη γεωμετρία <geometry>...</geometry>,
- ο τη μάζα <mass value="..." />.

) Προσδιορισμός των κόμβων/αρθρώσεων <joints>.

Κάθε ετικέτα <joint name=... type=...> αποτελείται από

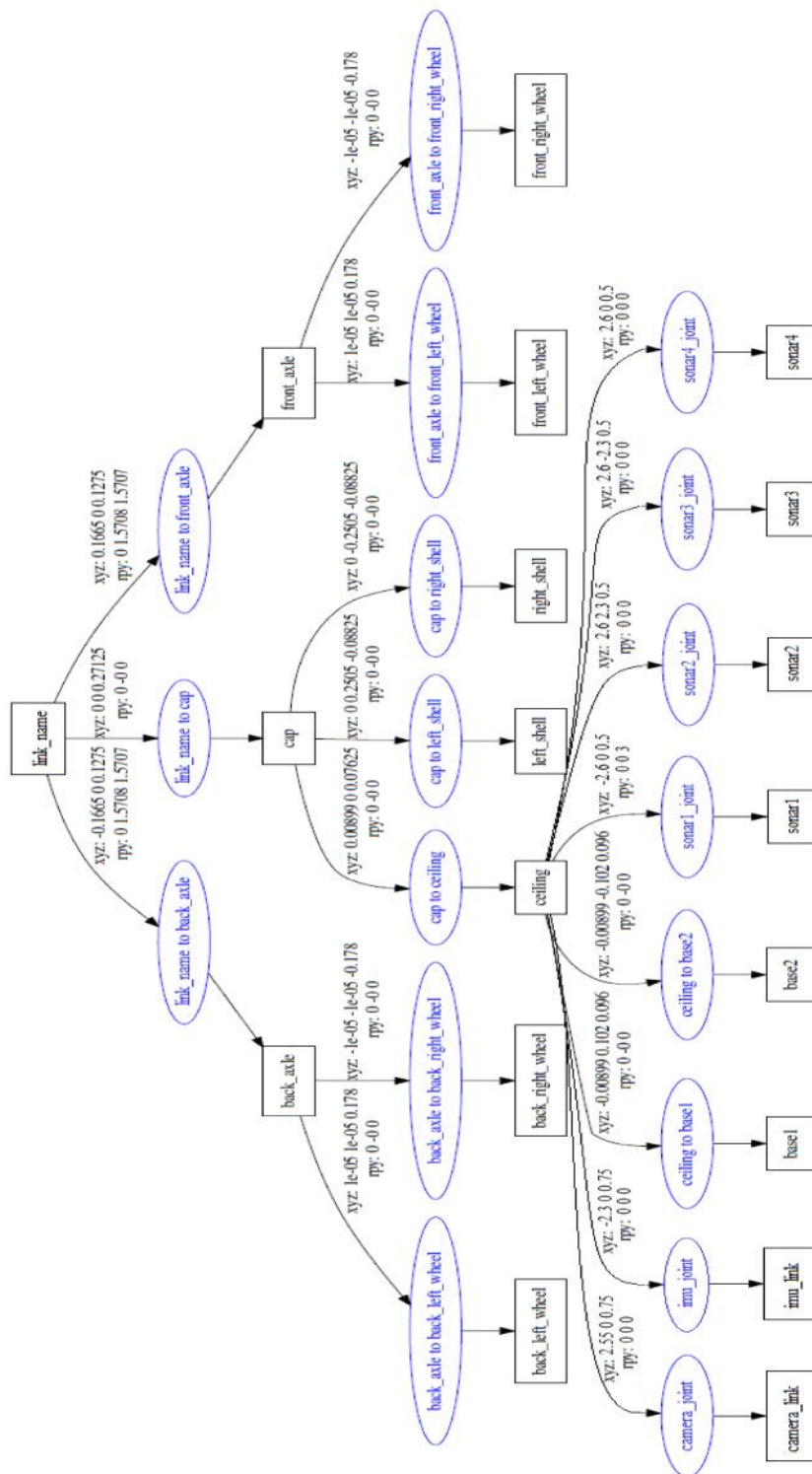
- ο το κύριο εξάρτημα γονέα <parent link="..." />
- ο το παιδί <child link="..." />
- ο τη θέση τους στο σύστημα συντεταγμένων <origin rpy="..." xyz="..." />

- ο τον άξονα εφαρμογής <axis xyz="..." />
- ο και τα όρια (μόνο για τις περιστρεφόμενες αρθρώσεις) <limit effort="..." lower="..." upper="..." velocity="..." />

Κατά αυτόν τον τρόπο σχεδιάστηκαν τα παρακάτω εξαρτήματα.

-) Το σασί του ρομποτικού οχήματος
-) Οι δύο οριζόντιοι άξονες των τροχών
-) Το κάλυμμα του άξονα και των τροχών
-) Τέσσερις τροχοί
-) Οροφή του οχήματος
-) Δύο πλαϊνά καλύμματα των τροχών
-) Δύο βάσεις στην κορυφή του οχήματος, για την στήριξη πρόσθετων εξαρτημάτων
-) Τέσσερις αισθητήρες υπερήχων (Sonar Sensors)
-) IMU
-) Κάμερα

Στην Εικόνα 2.4 φαίνεται η δομή του δέντρου που αποτελείται από αυτά, και τους αντίστοιχους κόμβους ανά ζεύγος εξαρτημάτων.



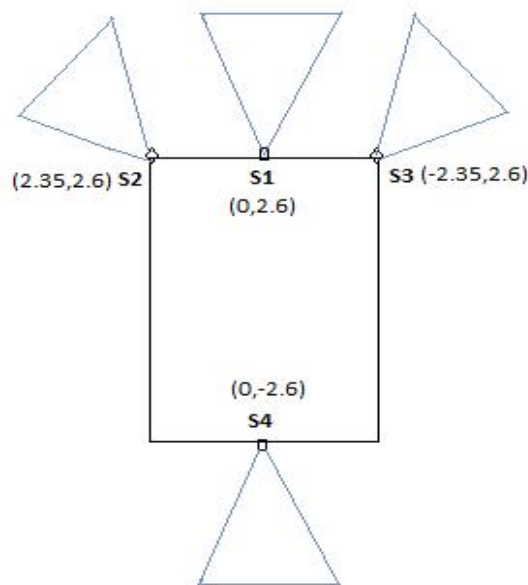
Εικόνα 2. 4: Δέντρο προσομοιωμένων εξαρτημάτων του μοντέλου

2.3 Προσθήκη εμφυτευμάτων κώδικα (Plugins)

Τα εμφυτεύματα κώδικα (plugins) είναι βιβλιοθήκες που δίνουν σε ένα URDF μοντέλο μεγαλύτερη λειτουργικότητα και προσομοιώνουν τα δομικά στοιχεία του ρομπότ, όπως για παράδειγμα τους αισθητήρες, και χαρακτηρίζονται από την κατάληξη .so . Χρησιμοποιούνται για την λειτουργία κάποιων εξαρτημάτων του ρομπότ που επικοινωνούν με το ROS καθώς και κάποιων λειτουργιών που δεν φαίνονται στη μοντελοποίηση ή τρέχουν στο παρασκήνιο, όπως το σύστημα κίνησης του οχήματος. Κατά αυτόν τον τρόπο προγραμματίστηκαν οι αισθητήρες και κινητήρες που θα αναλυθούν παρακάτω [23].

2.3.1 Αισθητήρες Υπερήχων

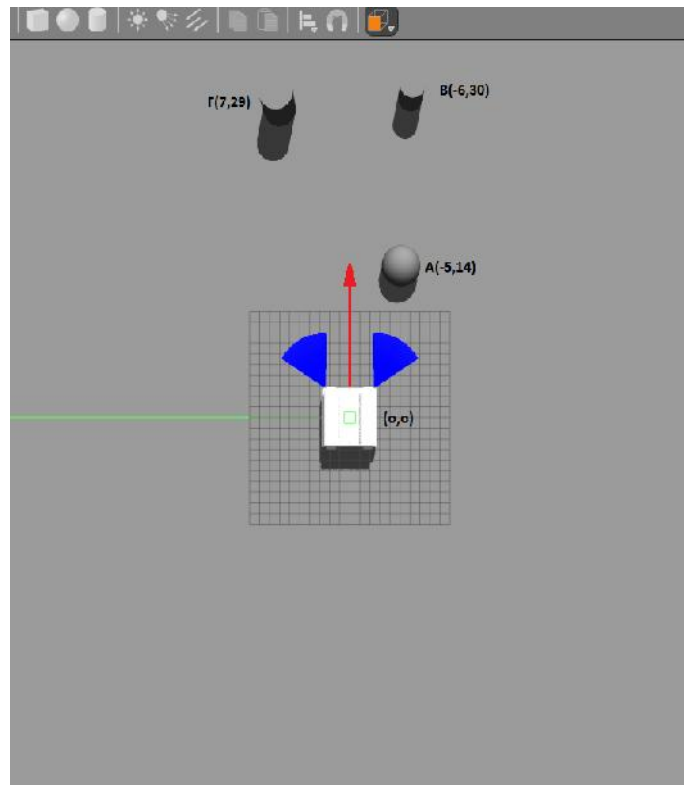
Ένας αισθητήρας υπερήχων υπολογίζει την απόσταση ενός εμποδίου, με τη βοήθεια της αντανάκλασης ενός υπερηχητικού σήματος. Χρησιμοποιεί το χρόνο που κάνει το επιστρεφόμενο σήμα να καλύψει την απόσταση, ώστε να εκτιμήσει την απόσταση του από το εμπόδιο αυτό. Όπως προαναφέρθηκε, χρησιμοποιήθηκαν τέσσερις αισθητήρες υπερήχων [10]. Στην Εικόνα 2.5 φαίνεται το σχέδιο της τοποθέτησης των αισθητήρων στο όχημα.



Εικόνα 2. 5: Σχέδιο τοποθέτησης αισθητήρων.

Χρησιμοποιήθηκε μοντέλο Sonar Sensor από προϋπάρχον πακέτο. Το συγκεκριμένο μοντέλο διαθέτει απόσταση μέτρησης από 0.2m έως και 5m. Για την πλήρη προσομοίωση του αισθητήρα, χρειάστηκε η προσθήκη ενός αρχείου περιγραφής εξαρτημάτων σύνθετης γεωμετρίας (mesh) που βρίσκεται στις βιβλιοθήκες του Gazebo, με τα εξωτερικά χαρακτηριστικά του. Οι μετρήσεις του αισθητήρα είναι ορατές από το αντίστοιχο θέμα (topic) του ROS μέσω της εντολής `rostopic echo`, ή από το περιβάλλον του Gazebo.

Για την πειραματική δοκιμή της λειτουργίας των προσομοιωμένων αισθητήρων, τοποθετήθηκαν εμπόδια δεξιά και αριστερά από το όχημα. Στην Εικόνα 2.6 φαίνεται η τοποθέτηση των εμποδίων στο γραφικό περιβάλλον του Gazebo, καθώς και το φάσμα ακτινών των αισθητήρων S2 και S3.



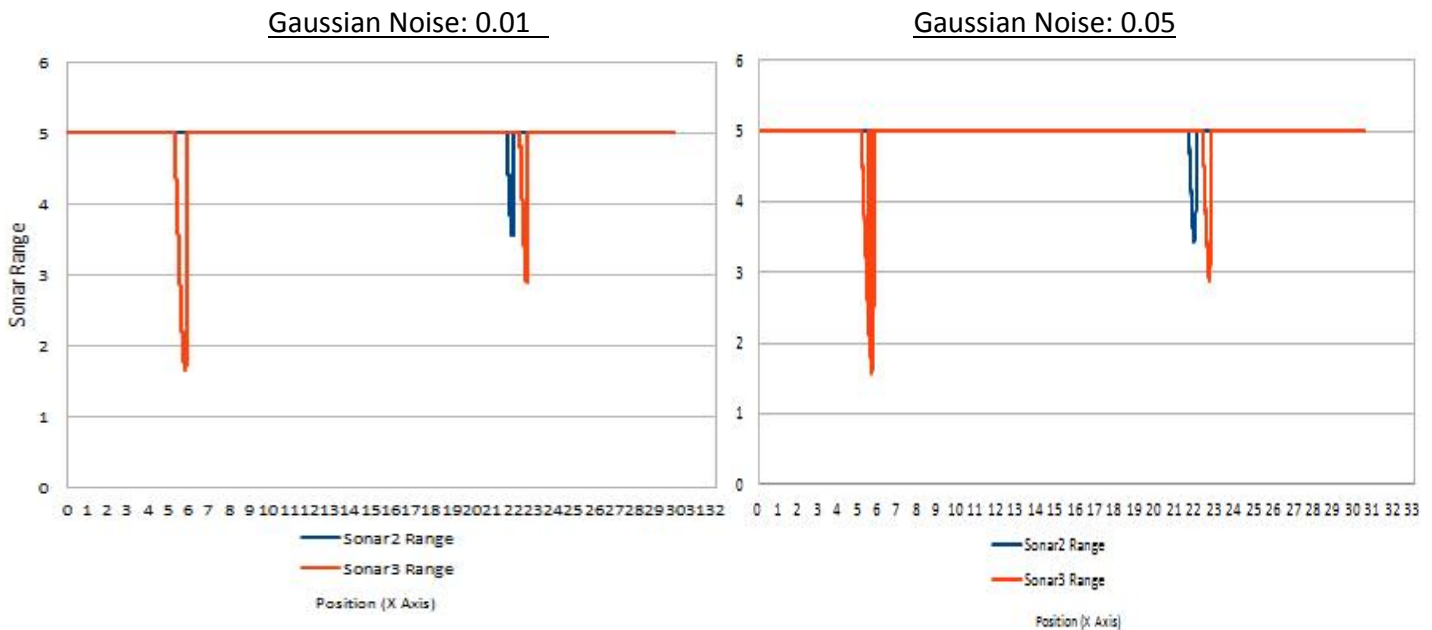
Εικόνα 2.6: Τοποθέτηση των εμποδίων γύρω από το όχημα.

Το όχημα έχει αρχική θέση την $(0,0)$. Ξεκινάμε να το κινούμε πάνω στον άξονα των x με θετική φορά. Η κίνηση του οχήματος πραγματοποιήθηκε με χρήση Τηλεχειρισμού, που θα αναλυθεί σε επόμενη ενότητα. Καθώς κινείται μακριά από τα εμπόδια, η εμβέλεια των αισθητήρων παραμένουν στην αρχική τους τιμή, δηλαδή τα 5m. Όταν το όχημα φτάσει στη θέση $(5.1,0)$, γίνεται η πρώτη ανίχνευση εμποδίου από τον δεξιό αισθητήρα (S3). Καθώς το όχημα κινείται δίπλα στο εμπόδιο, η ελάχιστη απόσταση στην οποία θα βρεθεί από αυτό, είναι τα 1.9m. Στη θέση $(5.7,0)$ του οχήματος, ο αισθητήρας σταματάει να ανιχνεύει το εμπόδιο, καθώς το όχημα έχει περάσει με ασφάλεια δίπλα από αυτό.

Προχωρώντας, και φτάνοντας στη θέση $(21.6,0)$ ο αριστερός αισθητήρας (S2) ανιχνεύει το εμπόδιο που βρίσκεται στη θέση $(7,29)$. Η ελάχιστη απόσταση στην οποία θα βρεθεί το όχημα από το εμπόδιο είναι τα 3.45m, όταν το όχημα βρεθεί στη θέση $(21.9,0)$. Στη θέση $(21.99,0)$, ο αισθητήρας σταματάει να ανιχνεύει το εμπόδιο που βρίσκεται στα αριστερά του οχήματος.

Ταυτόχρονα με τις μετρήσεις του αριστερού αισθητήρα, καθώς το όχημα βρίσκεται στη θέση $(22.3,0)$, ο δεξιός αισθητήρας (S3) ανιχνεύει άλλο ένα εμπόδιο το οποίο βρίσκεται στη θέση $(-6,30)$. Ενώ το όχημα κινείται δίπλα σε αυτό, η ελάχιστη απόσταση μεταξύ τους είναι 2.88m στη θέση $(22.6,0)$ του οχήματος. Ο δεξιός αισθητήρας καταγράφει την τελευταία μέτρηση, όταν το όχημα βρεθεί στη θέση $(22.7,0)$.

Τα δεδομένα των ακριανών αισθητήρων S2 και S3 καθώς το όχημα κινείται στον άξονα των x παρουσιάζονται στην Εικόνα 2.7. Πραγματοποιήθηκαν δύο μετρήσεις με Γκαουσιανό θόρυβο 0.01 και 0.05. Ο Γκαουσιανός θόρυβος προστίθεται στο φάσμα κάθε δέσμης των αισθητήρων. Μετά την προσθήκη θορύβου, η προκύπτουσα εμβέλεια συσφίγγεται μεταξύ της ελάχιστης και της μέγιστης εμβέλειας του αισθητήρα. Το αποτέλεσμα από τη αυξομείωση του θορύβου φαίνεται στην καμπύλη του γραφήματος.

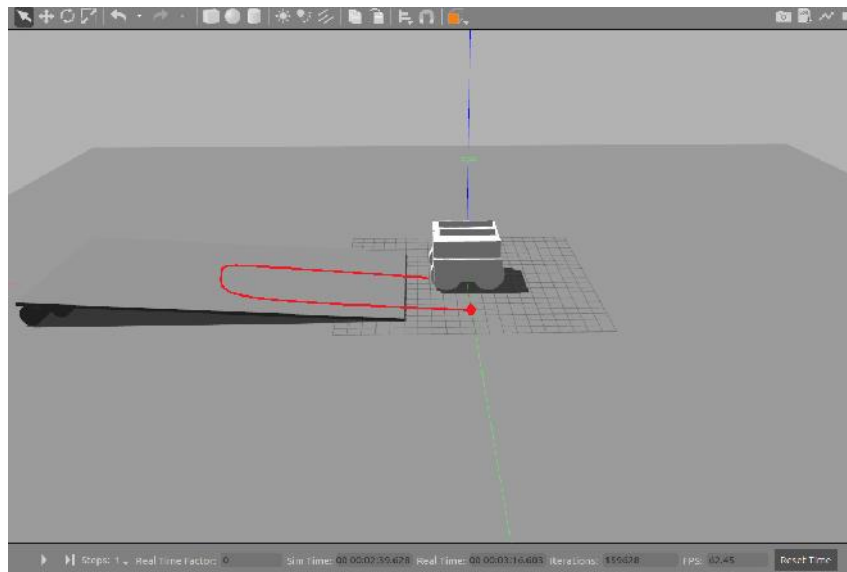


Εικόνα 2. 7: Γράφημα καταγεγραμμένων δεδομένων με Γκαουσιανό θόρυβο 0.01 (αριστερά) και 0.05 (δεξιά) .

2.3.2 IMU

Προσομοιώνει έναν αδρανειακό αισθητήρα προσανατολισμού. Μια μονάδα μέτρησης αδρανείας (IMU) είναι μια ηλεκτρονική συσκευή που, χρησιμοποιώντας ένα συνδυασμό επιταχυνσιόμετρων, γυροσκοπίων και μαγνητομέτρων υπολογίζει τον προσανατολισμό του σώματος στον τριδιάστατο χώρο. Η IMU που χρησιμοποιείται έχει ρυθμό ανανέωσης δεδομένων 100 φορές το δευτερόλεπτο [7]. Τα δεδομένα των μετρήσεων που πραγματοποιεί στην προσομοίωση μπορούμε να τα δούμε στο αντίστοιχο θέμα /imu, ή στο περιβάλλον του Gazebo από το παράθυρο του Topic Visualization.

Για τον έλεγχο λειτουργίας του αισθητήρα IMU, αυτή τη φορά χρησιμοποιήθηκε ανηφορική επιφάνεια πάνω στην οποία οδηγήθηκε το όχημα. Στη συγκεκριμένη διαδρομή μετρήθηκαν τρεις παράμετροι, η περιστροφή γύρω από τον άξονα x (roll), η περιστροφή γύρω από τον άξονα y (pitch) και η περιστροφή γύρω από τον άξονα z (yaw), που περιγράφουν πλήρως τις περιστροφικές κινήσεις ενός σώματος. Στην Εικόνα 2.8 βλέπουμε τη διαδρομή που ακολούθησε το όχημα για την καταγραφή των πειραματικών δεδομένων.

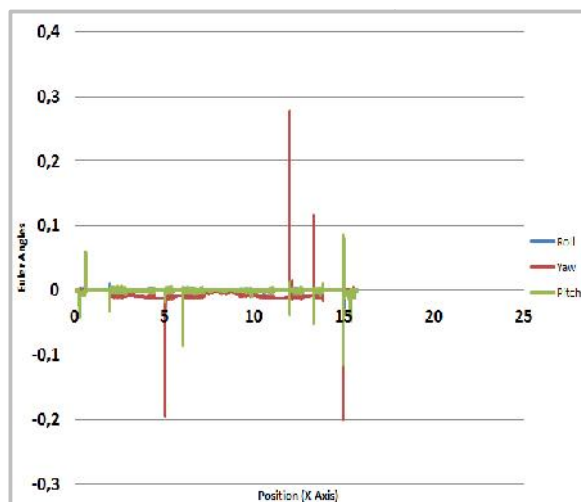


Εικόνα 2. 8: Διαδρομή του οχήματος για την καταγραφή των πειραματικών δεδομένων.

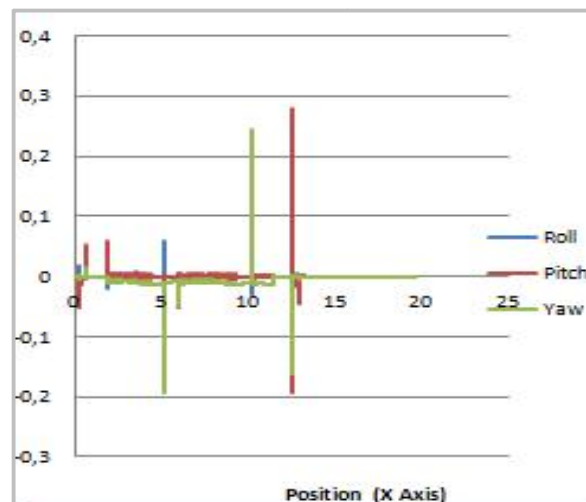
Η αρχική θέση του οχήματος είναι η $(0,0)$. Πραγματοποιεί ευθεία πορεία πάνω στον άξονα των x , μέχρι το σημείο $(19.5,0)$ όπου εκτελεί αριστερή επιτόπια στροφή και συνεχίζει την πορεία προς τα πίσω.

Οι μετρήσεις που πραγματοποιήθηκαν κατά την κίνηση του οχήματος στον άξονα των x καταγράφονται στα γραφήματα της Εικόνας 2.9.

Gaussian Noise: 0.00



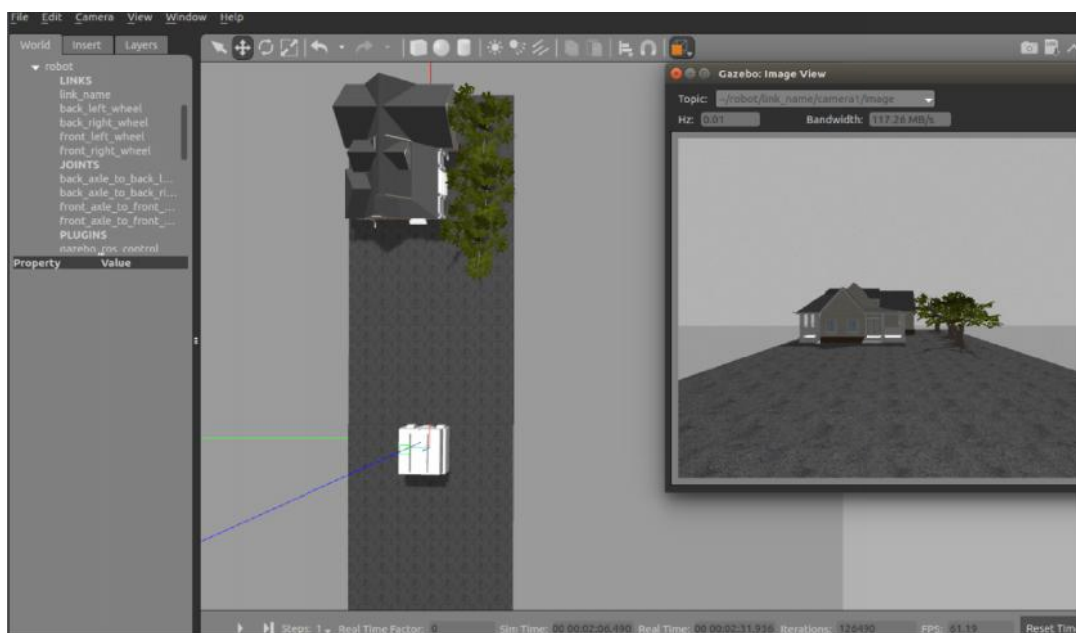
Gaussian Noise: 0.05



Εικόνα 2. 9: Γράφημα καταγεγραμμένων δεδομένων με Γκαουσιανό θόρυβο 0.00 (αριστερά) και 0.05 (δεξιά).

2.3.3 Συσκευή Εικονοληψίας

Το προσομοιωμένο όχημα reDevil περιλαμβάνει μια RGB κάμερα [9]. Το Gazebo διαθέτει παράθυρο μέσω του Topic Visualization, που μας δείχνει το κάδρο που κατέγραψε η συσκευή εικονοληψίας την οποία έχουμε τοποθετήσει στο όχημα. Η ανανέωση της εικόνας γίνεται 30 φορές το δευτερόλεπτο, που σημαίνει ότι έχουμε πλήρη έλεγχο της κίνησης του περιβάλλοντος που βρίσκεται στο μπροστινό μέρος του οχήματος. Στην Εικόνα 2.10 φαίνεται η οπτική καταγραφή στο παράθυρο του Topic Visualization του Gazebo.



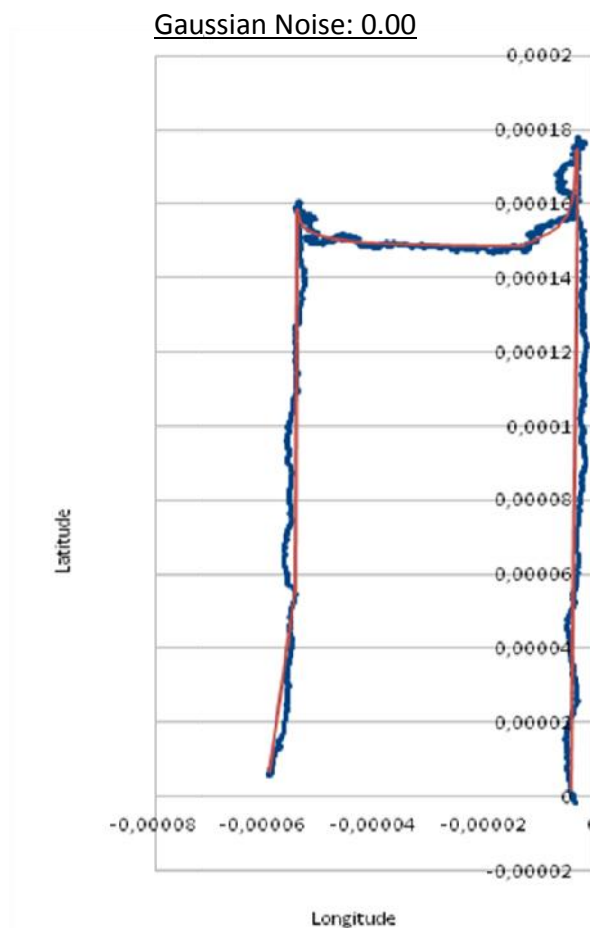
Εικόνα 2. 10: Λήψη της κάμερας στο περιβάλλον του Gazebo

2.3.4 Global Positioning System (GPS)

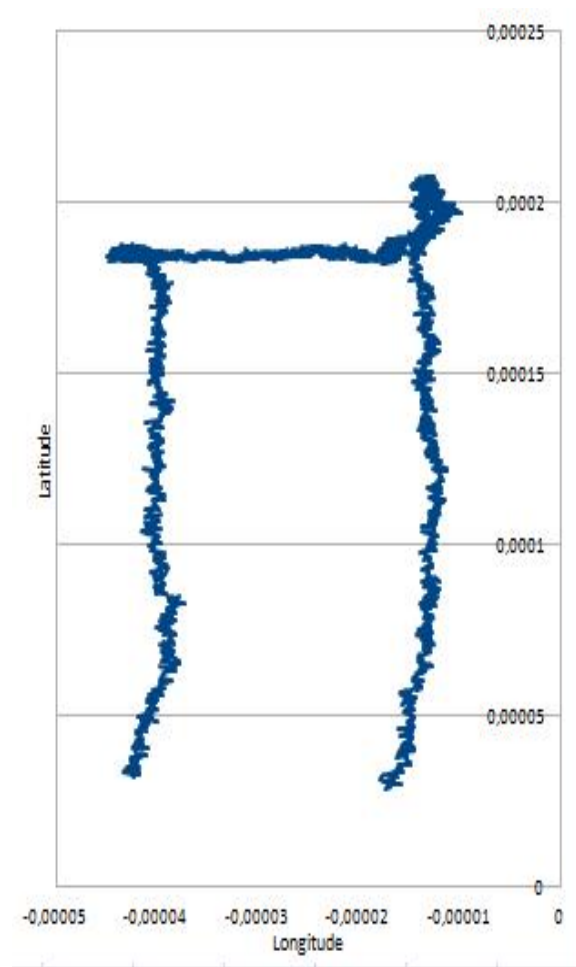
Το Global Positioning System ή Παγκόσμιο Σύστημα Στιγματοθέτησης, είναι σύστημα εντοπισμού της παγκόσμιας γεωγραφικής θέσης, ακίνητου ή κινούμενου στοιχείου το οποίο βασίζεται σε ένα πλέγμα δορυφόρων. Οι δέκτες GPS παρέχουν ακριβείς πληροφορίες για τη θέση τους σε γεωγραφικό πλάτος και μήκος, για το υψόμετρο, την ταχύτητα και την κατεύθυνση κατά την οποία κινείται το σώμα. Επίσης, μέσω ειδικού λογισμικού χαρτογράφησης μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές σε χάρτη [22].

Η χρήση του GPS στο Gazebo μας δίνει τη θέση του ρομπότ, με τη βοήθεια των γεωγραφικών του συντεταγμένων [6]. Οι γεωγραφικές συντεταγμένες ενός σώματος είναι το γεωγραφικό μήκος (Longitude) και το γεωγραφικό πλάτος (Latitude). Η ανανέωση των μετρήσεων γίνεται 10 φορές το δευτερόλεπτο, γεγονός που μας προσφέρει μεγάλη ακρίβεια. Μέσω του θέματος /fix μπορούμε να δούμε τα δεδομένα των μετρήσεων στην οθόνη.

Για τον έλεγχο του GPS χρησιμοποιήθηκε η ίδια κατασκευή που βοήθησε στη λήψη μετρήσεων του IMU όπως απεικονίζεται στην Εικόνα 2.8. Στην Εικόνα 2.11, βλέπουμε τη διαδρομή που πραγματοποίησε το όχημα, όπως περιγράφεται από τα Longitude και Latitude. Με μπλε χρώμα φαίνεται η διαδρομή από τα καταγεγραμμένα δεδομένα που μας έδωσε το ROS, ενώ με ροζ χρώμα φαίνεται η θεωρητική διαδρομή που έπρεπε να καταγραφεί.



Εικόνα 2. 11: Γράφημα καταγεγραμμένης διαδρομής με Γκαουσιανό θόρυβο 0.00.

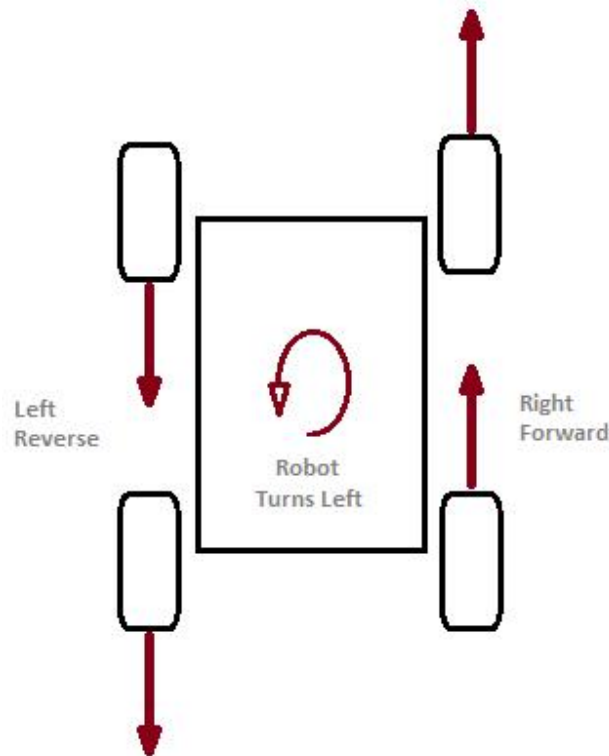
Gaussian Noise 0.05

Εικόνα 2. 12: Γράφημα καταγεγραμμένης διαδρομής με Γκαουσιανό θόρυβο 0.05.

2.3.5 Κίνηση με Ολίσθηση Τροχών (Skid Steering)

Το σύστημα κίνησης με ολίσθηση των τροχών χρησιμοποιείται ευρέως σε χωματουργικά μηχανήματα ή άρματα μάχης, αλλά επίσης και σε οχήματα με τέσσερις ή έξι τροχούς. Σε αυτά τα οχήματα, οι τροχοί σε κάθε πλευρά μπορούν να κινούνται με διαφορετικές ταχύτητες. Ανάλογα με τη φορά της κίνησης που θέλουμε να δώσουμε στο όχημα, οι τροχοί μπορούν να κινηθούν προς τα εμπρός και προς τα πίσω (όλοι οι τροχοί σε μια πλευρά κινούνται με την ίδια ταχύτητα), ενώ υπάρχει ένας κινητήρας σε κάθε πλευρά [8][14].

Στην Εικόνα 2.13 οι τροχοί στη δεξιά πλευρά κινούνται προς τα εμπρός και οι τροχοί στην αριστερή πλευρά κινούνται αντίστροφα με την ίδια ταχύτητα. Το αποτέλεσμα είναι μια περιστροφή αριστερόστροφα γύρω από το κέντρο του οχήματος.

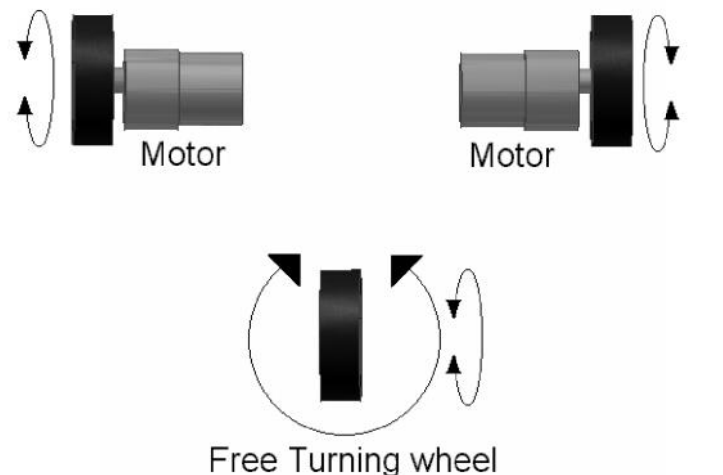


Εικόνα 2. 13: Αναπαράσταση της κίνησης των τροχών κατά τη στρέψη του οχήματος

Η χρήση τροχών προσφέρει ευκολία στο χειρισμό των ρομποτικών οχημάτων αλλά και απλότητα στην κατασκευή τους. Σε κάποια ρομπότ, χρησιμοποιείται ένα είδος οδήγησης που ονομάζεται διαφορική οδήγηση (Drive). Σε αυτή την περίπτωση, τοποθετούνται οι μπροστινοί τροχοί και ένας τρίτος τροχός στο πίσω μέρος για να παλινδρομεί συνεχώς, αποτρέποντας την πτώση του ρομπότ. Για να διατηρήσει ένα ρομπότ ευθεία πορεία, θα πρέπει να έχει ταυτόχρονη κίνηση και των δύο μπροστινών τροχών με ίδια ταχύτητα και φορά. Για να εκτελέσει περιστροφή, ένα κοινό σύστημα κινεί και τους δύο τροχούς, ώστε το όχημα να αλλάξει κατεύθυνση. Η διαφορική οδήγηση χαρακτηρίζεται από την ανεξάρτητη κίνηση των δυο τροχών με κοινό άξονα περιστροφής. Αν και οι δύο τροχοί γυρίσουν με την ίδια ταχύτητα σε αντίθετες

κατευθύνσεις, το ρομπότ θα περιστραφεί γύρω από το κεντρικό σημείο του άξονα του. Διαφορετικά, ανάλογα με την ταχύτητα περιστροφής και την κατεύθυνση της, το κέντρο περιστροφής μπορεί να πέσει οπουδήποτε στη γραμμή που ορίζεται από τα δύο σημεία επαφής των τροχών. Για παράδειγμα, αν θέλουμε να στρίψουμε το ρομπότ αριστερά, θα πρέπει να περιστραφεί ο δεξιός τροχός με μεγαλύτερη ταχύτητα από τον αριστερό.

Στην Εικόνα 2.14 βλέπουμε την κίνηση των τροχών που περιγράφεται παραπάνω.



Εικόνα 2. 14: Κίνηση των τροχών κατά τη διαφορική οδήγηση

Το Skid Steer Drive είναι στενά συνδεδεμένο με το διαφορικό σύστημα οδήγησης, αντικαθιστώντας τον τρίτο τροχό με επιπλέον τροχούς κίνησης. Όταν ασχολούμαστε με ρομποτικά οχήματα με τέσσερις ή έξι τροχούς, χρησιμοποιούμε οδήγηση με ολίσθηση των τροχών (Skid Steering), ενώ στην περίπτωση των τριών τροχών χρησιμοποιούμε διαφορική οδήγηση (Differential Drive). Αυτό, συμβαίνει γιατί θα ήταν δύσκολη η πορεία του οχήματος με τέσσερις ή έξι τροχούς σε μία ευθεία γραμμή, χρησιμοποιώντας διαφορική οδήγηση, διότι το πίσω μέρος του οχήματος δεν θα ελεγχόταν σωστά.

Για την υλοποίηση του Skid Steering στο ρομποτικό όχημα, χρειάστηκε να προσθέσουμε έναν εμφυτευμένο κώδικα από τις βιβλιοθήκες του Gazebo. Το Skid Steer Drive controller plugin, είναι ίσως το πιο σημαντικό για τη λειτουργία του ρομπότ, και τους μετέπειτα πειραματισμούς. Η λειτουργία του θα αναλυθεί στο επόμενο κεφάλαιο.

2.4 Προγραμματισμός ρομποτικού ελεγκτή

2.4.1 Έλεγχος του οχήματος με PID Controllers

Ένας PID ελεγκτής ή τριπλός ελεγκτής είναι ένας μηχανισμός ανάδρασης βρόχου ελέγχου που χρησιμοποιείται σε συστήματα ελέγχου και σε εφαρμογές που απαιτούν συνεχή διαμορφωμένο έλεγχο. Ένας PID ελεγκτής υπολογίζει συνεχώς μια error value $e(t)$, και διαμορφώνει τις τιμές του με στόχο να την μηδενίσει. Τα αρχικά PID προέρχονται από τις λέξεις Proportional, Integral, Derivative που στα ελληνικά σημαίνουν Αναλογικοί, Ολοκληρωτικοί, Διαφορικοί. Στην πράξη οι ελεγκτές PID συναντώνται συχνά, όπως για παράδειγμα στον έλεγχο ταχύτητας σε ένα αυτοκίνητο. Οι εξωτερικές επιδράσεις όπως οι κλίσεις θα προκαλούσαν αλλαγές ταχύτητας, όμως ο οδηγός έχει τη δυνατότητα να αλλάξει την επιθυμητή καθορισμένη ταχύτητα. Ο αλγόριθμος PID επαναφέρει την πραγματική ταχύτητα στην επιθυμητή ταχύτητα με τον βέλτιστο τρόπο, ελέγχοντας την ισχύ εξόδου του κινητήρα [24].

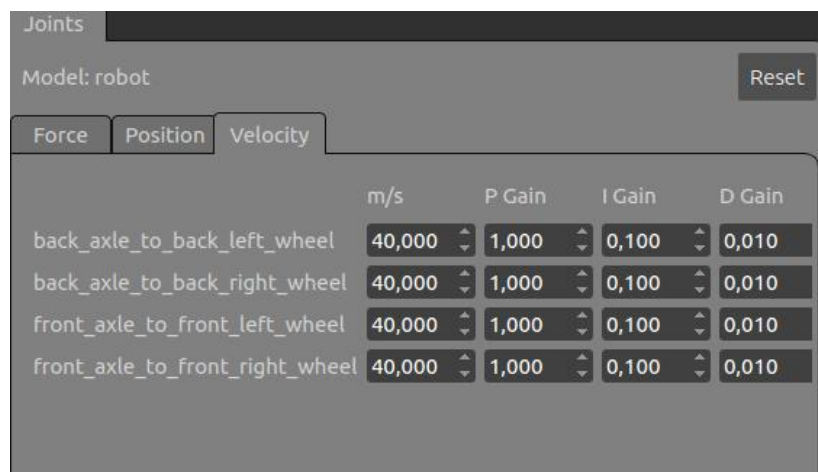
Η συνάρτηση μεταφοράς του PID ελεγκτή είναι η εξής:

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s}$$

όπου,

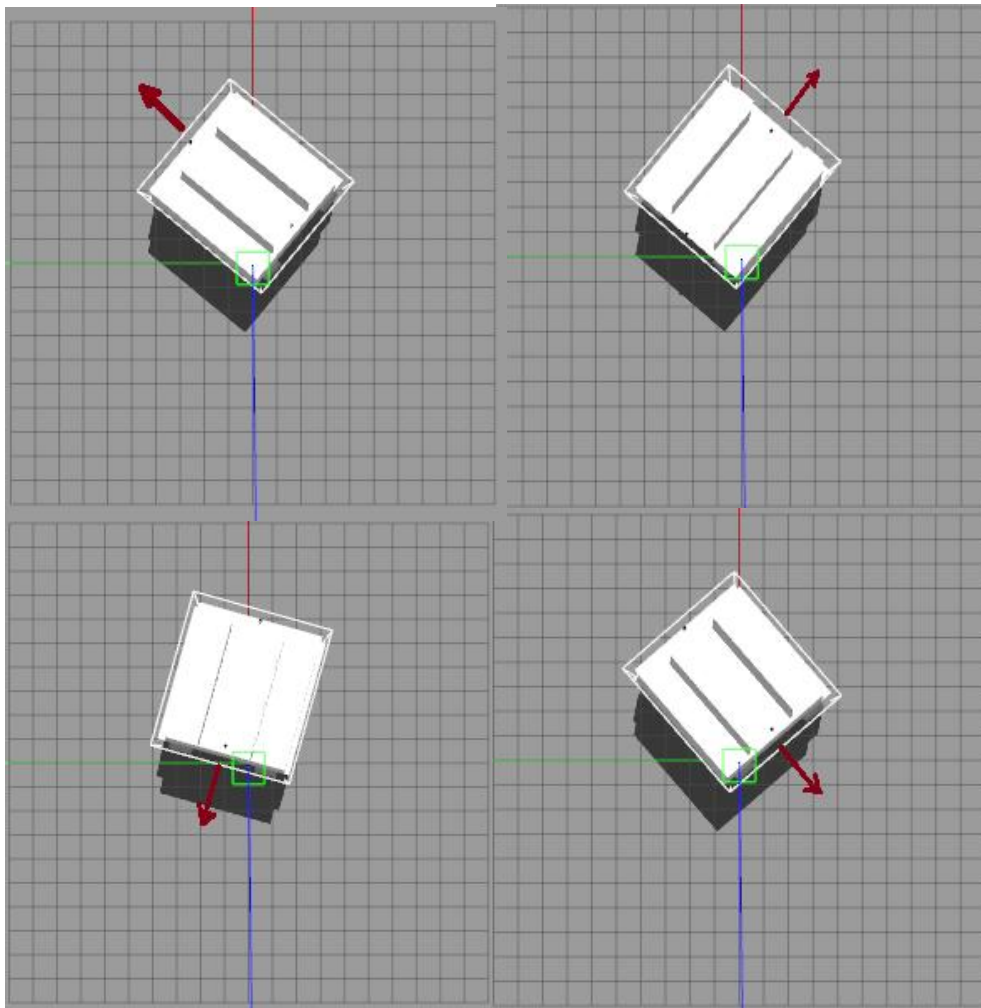
-) K_p = Αναλογικό Κέρδος (Proportional Gain)
-) K_I = Ολοκληρωτικό Κέρδος (Integral Gain)
-) K_D = Διαφορικό Κέρδος (Differential Gain)

Στο περιβάλλον του Gazebo μπορούμε να χειριστούμε τα P, I, D για την κάθε άρθρωση που δίνει κίνηση στους τροχούς. Στην περίπτωση μας, εμφανίζονται οι αρθρώσεις του οχήματος, και τρεις καρτέλες Force, Position, και Velocity. Έχουμε δηλαδή τη δυνατότητα, να ασκήσουμε δύναμη, να περιστρέψουμε τις αρθρώσεις και να δώσουμε ταχύτητα μέσω του PID ελεγκτή. Στην Εικόνα 2.15 φαίνεται το χειριστήριο αυτό.



Εικόνα 2. 15:PID ελεγκτής στο Gazebo

Επιλέγοντας τις κατάλληλες τιμές της ταχύτητας και των P, I, D, και με τη βοήθεια του Skid Steer Drive Plugin του οχήματος μπορούμε να του δώσουμε κίνηση και κατεύθυνση. Οι τροχοί περιστρέφονται, και ανάλογα με την ταχύτητα του καθενός το όχημα έχει τη δυνατότητα να κινηθεί. Στην Εικόνα 2.16 φαίνεται το παράδειγμα που περιγράφεται στην Εικόνα 2.13. Έχοντας εισάγει ίδιες τιμές της ταχύτητας στους τέσσερις τροχούς, και αντίθετη φορά, το όχημα εκτελεί αριστερόστροφη περιστροφή γύρω από τον άξονα του.



Εικόνα 2. 16: Θέσεις του οχήματος προς τέσσερις κατευθύνσεις καθώς το όχημα περιστρέφεται γύρω από τον κάθετο άξονα.

2.4.2 Πλοήγηση με τη χρήση Τηλεχειρισμού (Keyboard Teleoperation)

Για την πλοήγηση του ρομπότ με τη χρήση του Keyboard Teleoperation [20] χρειάζεται και πάλι το Skid Steer Drive controller plugin που προαναφέρθηκε. Το Keyboard Teleoperation είναι ένα πακέτο που δημοσιεύει (publish) μηνύματα τύπου Twist στο θέμα `cmd_vel`. Ο ελεγκτής Skid Steer Drive ακούει τις εντολές που στέλνονται σε αυτό το θέμα και κινεί το ρομπότ. Για να είναι αυτό εφικτό, ο χρήστης στέλνει εντολές από το πληκτρολόγιο πατώντας συγκεκριμένα πλήκτρα, δίνοντας την κατεύθυνση και τη φορά, καθώς και αυξομειώσεις στην ταχύτητα. Στην Εικόνα 2.17,

φαίνεται το παράθυρο του τερματικού μέσω του οποίου στέλνουμε τις εντολές κίνησης.

```
Reading from the keyboard and Publishing to Twist!
-----
Moving around:
  u      i      o
  j      k      l
  m      ,      .

For Holonomic mode (strafing), hold down the shift key:
-----
  U      I      O
  J      K      L
  M      <      >

t : up (+z)
b : down (-z)

anything else : stop

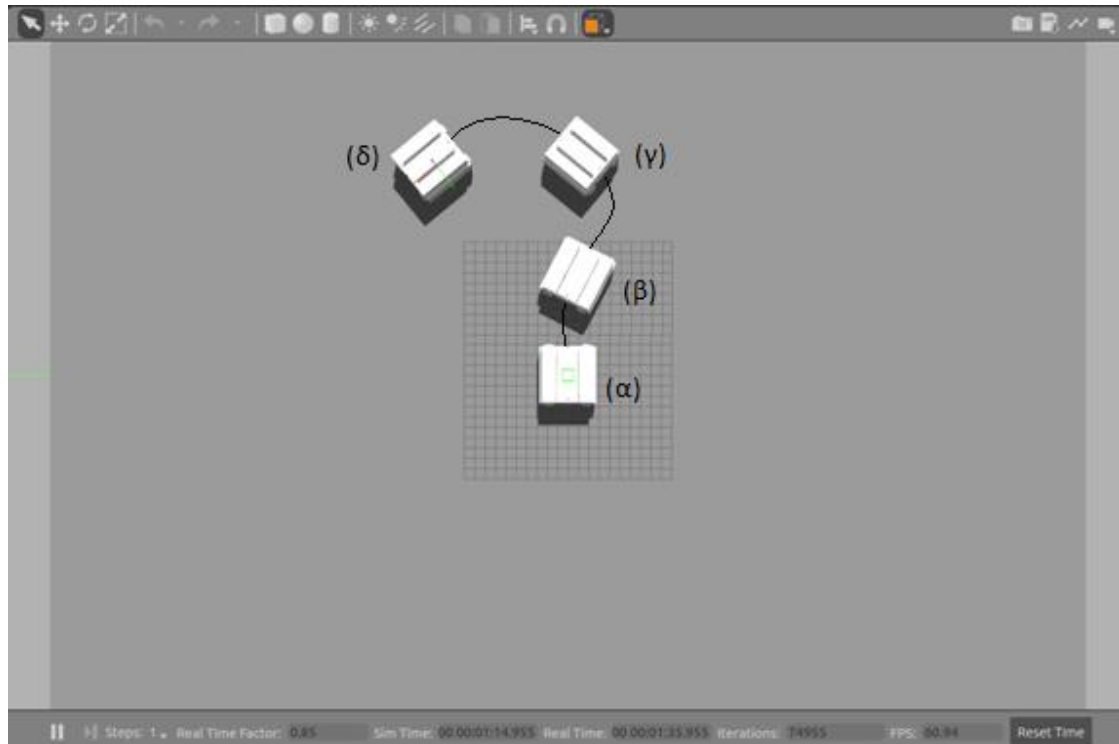
q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5      turn 1
```

Εικόνα 2. 17: Χειριστήριο του Keyboard Teleoperation

Στην Εικόνα 2.18 βλέπουμε τη διαδρομή που ακολούθησε το όχημα με τη χρήση του Τηλεχειρισμού (Keyboard Teleoperation). Για να πραγματοποιηθεί η συγκεκριμένη διαδρομή, το όχημα ξεκίνησε από το σημείο (0,0) στη θέση (α) της Εικόνας 2.18. Αρχικά, πατώντας το πλήκτρο «j» το όχημα εκτελεί ευθεία κίνηση. Στη συνέχεια με το πλήκτρο «u» εκτελεί ανοιχτή στροφή δεξιά για να βρεθεί στη θέση (β). Για να βρεθεί το όχημα στη θέση (γ), ακολουθεί ευθεία πορεία με τη χρήση του πλήκτρου «j», και ύστερα πατώντας το πλήκτρο «,» εκτελεί αριστερή στροφή γύρω από τον άξονα του μέχρι να φτάσει στη σωστή θέση. Για να φτάσει το όχημα στη θέση (δ), δίνουμε εντολή με το πλήκτρο «.» να κάνει ανοιχτή στροφή αριστερά. Τέλος, ακινητοποιούμε το όχημα με τη χρήση του πλήκτρου «k».



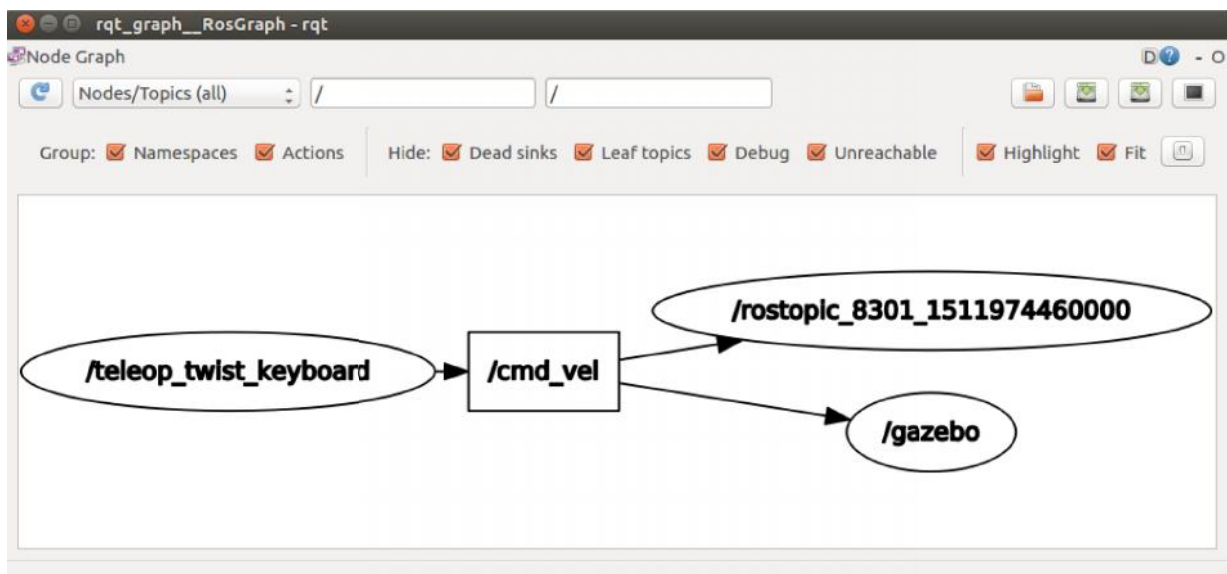
Εικόνα 2. 18: Η διαδρομή που ακολούθησε το όχημα λαμβάνοντας τις εντολές του Keyboard Teleoperation.

Ένα χρήσιμο εργαλείο κατά την κίνηση του οχήματος, είναι το θέμα `cmd_vel`. Η εντολή `rostopic echo cmd_vel` εκτυπώνει στην οθόνη τις αντίστοιχες ταχύτητες που δίνουμε μέσω του παραθύρου Τηλεχειρισμού, καθώς και την κίνηση στους αντίστοιχους άξονες. Στην Εικόνα 2.19 φαίνονται τα δεδομένα αυτά.

```
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: -10.5568883727
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 15.8630929717
---
linear:
  x: -10.5568883727
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```

Εικόνα 2. 19: Δεδομένα κατά την κίνηση.

Επιπλέον, μας δίνεται η δυνατότητα να δούμε τις εργασίες και τα θέματα που τρέχουν κατά τη διάρκεια της προσομοίωσης μέσω του `rqt_graph`. Κατά τη διάρκεια της συγκεκριμένης εργασίας, διακρίνουμε τα εξής (Εικόνα 2.21):



Εικόνα 2. 21: Rqt_graph κατά την κίνηση του οχήματος με το Keyboard Teleoperation

Όπως βλέπουμε η εργασία `teleop_twist_keyboard` δημοσιεύει στο θέμα `cmd_vel`, και το Gazebo λαμβάνει τα μηνύματα για να δώσει κίνηση στο ρομπότ. Επίσης στο πάνω δεξί μέρος εμφανίζεται το τρέχον `rostopic` που μας εμφανίζει τα μηνύματα στην οθόνη.

Κεφάλαιο 3: Ρομποτικά οχήματα εξωτερικού χώρου διαθέσιμα στο Gazebo.

-) Ένα παράδειγμα οχήματος εξωτερικού χώρου, που είναι διαθέσιμο στην open source κοινότητα και μπορεί να χρησιμοποιηθεί με το Gazebo και το ROS, είναι το HUSKY A200 της Clearpath Robotics. Πρόκειται για ένα ρομποτικό όχημα, που έχει σχεδιαστεί για να ευδοκιμεί σε σκληρές εξωτερικές συνθήκες. Διαθέτει ισχυρό 4x4 μηχανοκίνητο σύστημα μετάδοσης κίνησης, ελαστικά παντός εδάφους και ασφαλή απόσταση από το έδαφος.



Εικόνα 3. 1: HUSKY A200

Το HUSKY A200 είναι διαθέσιμο για προσομοίωση, και στα αισθητήρια όργανα του, περιλαμβάνονται IMU, GPS και μία Kinect camera.

-) Ένα ακόμα ρομπότ εξωτερικού χώρου είναι το Pioneer 3-AT της Mobile Robots. Το Pioneer 3-AT είναι ένα μικρό ρομπότ με τέσσερις τροχούς. Έχει σύστημα κίνησης με ολίσθηση τροχών (Skid Steer) και χρησιμοποιείται σε πολλούς εργαστηριακούς πειραματισμούς. Το Pioneer 3-AT συνοδεύεται από μια

μπαταρία, διακόπτη έκτακτης ανάγκης, κωδικοποιητές τροχών και μικροελεγκτή με firmware ARCOS, καθώς και το πακέτο ανάπτυξης λογισμικού ρομποτικής Pioneer SDK. Τα ρομπότ έρευνας της Pioneer είναι αρκετά δημοφιλή για την εκπαίδευση και την έρευνα [18].



Εικόνα 3. 2: Pioneer 3-AT

Διαθέτει αισθητήρια όργανα laser, καθώς και DGPS (Διαφορικό GPS). Το DGPS χρησιμοποιείται για τη βελτίωση της ακρίβειας του GPS. Χρησιμοποιούνται επίγειοι σταθμοί, των οποίων η θέση είναι ήδη γνωστή και υπολογίζουν το σφάλμα που εκπέμπουν συσκευές GPS, και διορθώνουν τους υπολογισμούς τους.

Προαιρετικά, διαθέτει 8 μπροστινούς αισθητήρες Sonar και 8 στο πίσω μέρος, που ανιχνεύουν εμπόδιο από τα 15 cm έως τα 7 m, καθώς και κάμερα και πυξίδα.



Εικόνα 3. 3: Pioneer 3-AT εξοπλισμένο με αισθητήρια όργανα

-) Ακόμα, άξια προσοχής είναι η προσομοίωση ενός μοντέλου Prius Hybrid [4]. Μέσω του Gazebo, μπορούμε να δούμε την προσομοίωση στην οθόνη. Το ROS επέτρεψε την ανάπτυξη της προσομοίωσης γρηγορότερα χρησιμοποιώντας τα υπάρχοντα λογισμικά και βιβλιοθήκες. Το γκάζι, το φρένο, το τιμόνι και το κιβώτιο του οχήματος ελέγχονται επίσης μέσω του ROS.



Εικόνα 3. 4: Prius Hybrid

Στο ίδιο το όχημα υπάρχει ένα σύστημα LiDAR στην οροφή, 8 υπερηχητικοί αισθητήρες, 4 κάμερες και 2 επίπεδα LiDAR. Η τεχνική LiDAR (Light Detection And Ranging) βασίζεται στην εκπομπή παλμικής ακτινοβολίας λέιζερ στην ατμόσφαιρα και ακολούθως, στην καταγραφή της οπισθοσκεδαζόμενης ακτινοβολίας λέιζερ [5].

-) Το Jackal UGV της Clearpath Robotics είναι ένα μικρό όχημα εξωτερικού χώρου, που κινείται με σύστημα ολίσθησης (Skid Steer) [3]. Διαθέτει ενσωματωμένο υπολογιστή, GPS και IMU πλήρως συμβατό με το ROS για αυτόνομη ικανότητα.

Συγχρονίζεται άμεσα με το RVIZ και το Gazebo. Μπορεί να εξοπλιστεί απο μια λίστα συμβατών αισθητήρων, κάμερας και άλλων εξαρτημάτων.



Εικόνα 3. 5: Jackal UGV

Το SUMMIT-XL είναι επίσης ρομποτικό όχημα εξωτερικού και εξωτερικού χώρου με ισχυρή μηχανική δομή, που κινείται και αυτό με σύστημα Skid Steer [2]. Βασίζεται σε 4 ηλεκτροκινητήρες υψηλής ισχύος. Κάθε τροχός ενσωματώνει έναν κινητήρα με κιβώτιο ταχυτήτων και κωδικοποιητή προαιρετικά. Διαθέτει οδόμετρο, τα δεδομένα του οποίου, βασίζονται στη χρήση των τεσσάρων κωδικοποιητών και ενός γωνιακού αισθητήρα υψηλής ακρίβειας που είναι τοποθετημένος μέσα στο όχημα. Μπορεί να πλοηγηθεί αυτόνομα ή μέσω τηλεχειρισμού, με τη βοήθεια κάμερας που μεταδίδει βίντεο σε πραγματικό χρόνο. Στα αισθητήρια όργανα συμπεριλαμβάνονται, ένας αισθητήρας λέιζερ Hokuyo και RTK-DGPS. Ο έλεγχος του οχήματος είναι βασισμένος στο ROS, ενώ είναι πλήρως συμβατό και το Gazebo.



Εικόνα 3. 6: SUMMIT-XL



Εικόνα 3. 7: SUMMIT-XL στο Gazebo

Κεφάλαιο 4: Συμπεράσματα

Πριν το 1980, που καθιερώθηκε ένας ευρέως αποδεκτός ορισμός για το τι είναι ένα βιομηχανικό ρομπότ, είχε τεθεί ήδη σε λειτουργία το πρώτο ρομπότ στη βιομηχανία. Μέχρι και σήμερα, έχουν γίνει τεράστια βήματα στην τεχνολογία των ρομποτικών οχημάτων, τα οποία συνεχίζουν με σημαντικές ανακαλύψεις, που βοηθούν σε διάφορους κλάδους της επιστήμης. Κάποιες από αυτές τις ανακαλύψεις έχουν εφαρμοστεί σε τομείς όπως, η ιατρική, η εξερεύνηση του διαστήματος, η διάσωση σε επικίνδυνα και άβυσσα μέρη, οι αγροτικές εφαρμογές, και άλλα [17].

Αρκετά χρόνια αργότερα από την εμφάνιση και τον προγραμματισμό των ρομποτικών οχημάτων, άρχισε να αναπτύσσεται η προσομοίωση. Η προσομοίωση είναι ένας πολύ σημαντικός τομέας, πλέον απαραίτητος, που βοηθά στην υλοποίηση πειραμάτων και ερευνών στην ρομποτική. Βοηθά στην εξάλειψη των σφαλμάτων που οδηγούν στην φθορά και καταστροφή των ρομποτικών οχημάτων, από τις συνεχείς δοκιμές τους στον πραγματικό κόσμο.

Το ρομποτικό όχημα reDevil (ATRV – Mini), δεν διαθέτει ήδη υπάρχον προσομοιωμένο μοντέλο για την ασφαλή υλοποίηση πειραμάτων. Παρόλα αυτά, διαθέτει σχετικά απλή δομή για τον τριδιάστατο σχεδιασμό του. Για τον προγραμματισμό και τον χειρισμό του, χρειάστηκε η αναζήτηση και η σύνταξη συμβατού λογισμικού και πακέτων, ώστε να γίνει δυνατή η μελλοντική του χρήση σε πειράματα και έρευνα.

Πιο συγκεκριμένα, η προσομοίωση του ρομποτικού οχήματος έγινε στον προσομοιωτή Gazebo, σε συνεργασία με τα παρεχόμενα πακέτα του ROS. Αυτό οδήγησε στην σύνταξη ενός λογισμικού για τον χειρισμό του οχήματος μέσα στο γραφικό περιβάλλον του Gazebo και τη καταγραφή δεδομένων που προέρχονται από τους προσομοιωμένους αισθητήρες, καθώς το όχημα κινείται στο εικονικό περιβάλλον.

Οι δυσκολίες που υπήρξαν κατά τον προγραμματισμό του οχήματος reDevil, αφορούσαν την συμβατότητα των χαρακτηριστικών του μοντέλου με προϋπάρχοντα πακέτα που διατίθενται από την open source κοινότητα, καθώς και η έλλειψη λογισμικού για σύγχρονα εξαρτήματα, όπως φωτοβολταϊκά πάνελ.

Στα μελλοντικά σχέδια για την ανάπτυξη του προσομοιωμένου μοντέλου reDevil, περιλαμβάνονται, η χρήση φωτοβολταϊκών πάνελ, η αυτόνομη πλοήγηση με

την ταυτόχρονη αποφυγή εμποδίων, καθώς και η χαρτογράφηση του περιβάλλοντος στο οποίο κινείται το όχημα.

Βιβλιογραφία

- [1] (Ηλεκτρονική πηγή) Robot Operating System (ROS), διαθέσιμο: <http://www.ros.org/>
- [2] (Ηλεκτρονική πηγή) SUMMIT-XL της Robotnik, διαθέσιμο: <https://www.robotnik.eu/mobile-robots/summit-xl/>
- [3] (Ηλεκτρονική πηγή) Jackal UGV της Clearpath Robotics, διαθέσιμο: <https://www.clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/>
- [4] (Ηλεκτρονική Πηγή) Simulated car demo, Prius Hybrid, διαθέσιμο: <https://www.osrfoundation.org/simulated-car-demo/>
- [5] (Ηλεκτρονική Πηγή) Vehicle and City Simulation, διαθέσιμο: http://gazebo.org/blog/car_sim
- [6] (Ηλεκτρονική Πηγή) GPS Sensor Plugin, διαθέσιμο: https://github.com/fsuarez6/labrob/tree/master/labrob_description
- [7] (Ηλεκτρονική Πηγή) IMU Plugin (GazeboRosImu) , διαθέσιμο: http://gazebo.org/tutorials?tut=ros_gzplugins
- [8] (Ηλεκτρονική Πηγή) Skid Steering Drive, διαθέσιμο: http://gazebo.org/tutorials?tut=ros_gzplugins
- [9] (Ηλεκτρονική Πηγή) Camera, διαθέσιμο: http://gazebo.org/tutorials?tut=ros_gzplugins
- [10] (Ηλεκτρονική Πηγή) Sonar Sensor, διαθέσιμο: https://github.com/tu-darmstadt-ros-pkg/hector_models/tree/groovy-devel/hector_sensors_description
- [11] (Ηλεκτρονική Πηγή) URDF in Gazebo, διαθέσιμο: http://gazebo.org/tutorials/?tut=ros_urdf
- [12] (Ηλεκτρονική Πηγή) Xacro, διαθέσιμο: <http://wiki.ros.org/xacro>
- [13] Οδυσσέας Μαυροματάκης, Ανακατασκευή ρομποτικού οχήματος ATRV-mini εξωτερικού χώρου, Μεταπτυχιακή εργασία, Τμήμα Μ.Π.Δ, Πολυτεχνείο Κρήτης, 2017
- [14] Ευαγγελία Μεριστούδη, Σχεδιασμός και Προσδιορισμός λειτουργικών τμημάτων ρομποτικού οχήματος εξωτερικού χώρου, Διπλωματική εργασία, Τμήμα Μ.Π.Δ, Πολυτεχνείο Κρήτης, 2016
- [15] Τσιλίβης Παναγιώτης, Αυτόνομη πλοήγηση μη επανδρωμένων αεροσκαφών, Διπλωματική εργασία, Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά, Τεχνολογικού Τομέα, 2016

[16] Ntegiannakis Theodosios Hardware and Software upgrade of the ATRV – Mini outdoor vehicle, M.Sc. Thesis, Production Engineering and Management, Technical University of Crete, 2017

[17] (Ηλεκτρονική Πηγή) Ρομποτική, Ιστορική αναδρομή, διαθέσιμο: <https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE>

[18] (Ηλεκτρονική Πηγή) Pioneer 3-AT, OMRON ADEPT MobileRobots, διαθέσιμο: <http://www.mobilerobots.com/ResearchRobots/P3AT.aspx>

[19] Ρομποτική ανοιχτού κώδικα, διαθέσιμο: <https://openhware.ellak.gr/2016/04/19/rompotiki-anichtou-kodika/>

[20] Rasika M. Kangutkar, Obstacle Avoidance and Path Planning for Smart Indoor Agents, Thesis, Rochester Institute of Technology RIT Scholar Works, 2017

[21] Modeling for Gazebo, A Design Guide for Proper Exporting from Solidworks for Gazebo Simulation

[22] Γεωγραφικές Συντεταγμένες, διαθέσιμο: https://en.wikipedia.org/wiki/Geographic_coordinate_system

[23] Gazebo plugins in ROS, διαθέσιμο: http://gazebosim.org/tutorials?tut=ros_gzplugins

[24] ΕΛΕΓΚΤΕΣ PID, διαθέσιμο: <http://gun.teipir.gr/DSAELAB/Ergastiriakes/pidtutorial.pdf>