



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ  
ΧΑΡΤΟΓΡΑΦΗΣΗΣ ΧΩΡΟΥ ΜΕ ΤΡΟΧΟΦΟΡΟ  
ΡΟΜΠΟΤ ΚΑΙ ΑΙΣΘΗΤΗΡΑ LASER

ΧΑΡΕΛΛΗΣ ΓΕΩΡΓΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΦΟΥΡΛΑΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής

Λαμία Ιούλιος 2024





ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ  
ΧΑΡΤΟΓΡΑΦΗΣΗΣ ΧΩΡΟΥ ΜΕ ΤΡΟΧΟΦΟΡΟ  
ΡΟΜΠΟΤ ΚΑΙ ΑΙΣΘΗΤΗΡΑ LASER

ΧΑΡΕΛΛΗΣ ΓΕΩΡΓΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΦΟΥΡΛΑΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής

Λαμία Ιούλιος 2024





UNIVERSITY OF  
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

# IMPLEMENTATION OF MAPPING ALGORITHMS WITH WHEELED ROBOT AND LASER SENSOR

CHARELLIS GEORGIOS

FINAL THESIS

ADVISOR

FOURLAS GEORGIOS  
Professor

Lamia July 2024



«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις <sup>(1)</sup>, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 19 / 08 /2024

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»







## ΠΕΡΙΛΗΨΗ

---

Στη παρούσα πτυχιακή εργασία, με την χρήση του ROS (Robot Operating System) παρουσιάζεται η ανάπτυξη και η εφαρμογή αλγορίθμων χαρτογράφησης (Gmapping) και αυτόνομης πλοήγησης σε δισδιάστατο χώρο, με τη χρήση του τροχοφόρου ρομποτικού οχήματος Pioneer 3-AT το οποίο διαθέτει δύο βαθμούς ελευθερίας κίνησης και του αισθητήρα λέιζερ SICK LMS-500-20000 PRO.

Με τη χρήση του συστήματος Adaptive Monte Carlo Localization (AMCL), που είναι ένα πιθανολογικό σύστημα εντοπισμού για ένα ρομπότ που κινείται σε 2D, γίνεται υπολογισμός της θέσης και του προσανατολισμού του ρομπότ σε έναν δεδομένο γνωστό χάρτη χρησιμοποιώντας έναν σαρωτή λέιζερ 2D. Για τις ανάγκες της πτυχιακής χρησιμοποιήθηκε ο χώρος του εργαστηρίου Ρομποτικής & Αυτομάτου Ελέγχου του τμήματος Πληροφορικής & Τηλεπικοινωνιών, για τον οποίο δημιουργήθηκε ο σχετικός χάρτης. Κατά την υλοποίηση της εργασίας τέθηκαν διάφορα σημεία στα οποία κλήθηκε το ρομποτικό όχημα να πάει αυτόνομα με τη χρήση του πακέτου Move Base του ROS όπου για κάθε διαδρομή δημιουργεί ένα μονοπάτι στο επίπεδο (Path Planning) και το όχημα το ακολουθεί προκειμένου να φτάσει στον στόχο του.

Εν κατακλείδι, στην εργασία αναγράφονται πλήρως και αναλυτικά οι υλοποιήσεις των εργαλείων και του εξοπλισμού που χρησιμοποιήθηκαν, αλλά και των πακέτων που αξιοποιήθηκαν.

## ABSTRACT

---

In this thesis, the development and application of mapping algorithms (Gmapping) and autonomous navigation in two-dimensional space is presented. This was accomplished with the use of ROS (Robot Operating System) and the Pioneer 3-AT wheeled robotic vehicle which has two degrees of free movement and the SICK LMS-500-20000 PRO laser sensor.

Using Adaptive Monte Carlo Localization (AMCL), which is a probabilistic localization system for a 2D moving robot, the position and orientation of the robot on a given known map is calculated with the help of a 2D laser scanner. For the needs of the project, the space of the Robotics & Automatic Control laboratory of the Informatics & Telecommunications department was used and the relevant map was created. During the implementation of the work, several points were set where the robotic vehicle was asked to go autonomously using the Move Base package of ROS. For each path a route (Path Planning) was created and the vehicle followed it in order to reach its set point.

In conclusion, the essay describes fully and in detail the implementations of the tools and equipment used, as well as the packages needed.



## Table of Contents

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| ΠΕΡΙΛΗΨΗ .....                                                        | I         |
| ABSTRACT .....                                                        | II        |
| <b>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ .....</b>                                      | <b>2</b>  |
| 1.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ .....                                      | 2         |
| 1.2 ΕΙΣΑΓΩΓΗ ΣΤΟ ROS .....                                            | 2         |
| 1.3 ΤΙ ΕΙΝΑΙ ΤΟ PIONEER 3-AT .....                                    | 3         |
| 1.4 ΤΙ ΕΙΝΑΙ ΤΟ AMR (AUTONOMOUS MOBILE ROBOT) .....                   | 4         |
| 1.5 ΤΙ ΕΙΝΑΙ ΤΟ SICK LMS500-20000 PRO .....                           | 5         |
| 1.5.1 SICK LMS500 LASER SCANNER .....                                 | 5         |
| 1.5.2 MULTI-ECHO TECHNOLOGY .....                                     | 5         |
| <b>ΚΕΦΑΛΑΙΟ 2 ΕΞΟΠΛΙΣΜΟΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ .....</b>          | <b>8</b>  |
| 2.1 ΕΞΟΠΛΙΣΜΟΣ .....                                                  | 8         |
| 2.1.1 ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ .....                                       | 8         |
| 2.1.1 ΣΥΝΔΕΣΗ ΤΟΥ ΑΙΣΘΗΤΗΡΑ ΜΕ ΤΟ PIONEER 3-AT .....                  | 8         |
| 2.2 ΠΑΚΕΤΑ(ROSARIA) ΓΙΑ ΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟ PIONEER 3-AT .....      | 9         |
| 2.3 ΠΑΚΕΤΑ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ .....                                    | 11        |
| 2.3.1 ΠΑΚΕΤΑ ΓΙΑ ΤΟ LASER SICK LMS-500 .....                          | 11        |
| 2.3.2 ΠΑΚΕΤΑ GMAPPING ΚΑΙ SLAM-GMAPPING .....                         | 11        |
| 2.3.3 ΠΑΚΕΤΟ AMCL .....                                               | 13        |
| 2.3.4 ΠΑΚΕΤΟ MOVE_BASE .....                                          | 15        |
| <b>ΚΕΦΑΛΑΙΟ 3 ΟΔΗΓΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ .....</b>             | <b>18</b> |
| 3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΛΕΙΤΟΥΡΓΙΚΟΥ ΚΑΙ ΤΟΥ ROS .....                        | 18        |
| 3.1.1 ΕΓΚΑΤΑΣΤΑΣΗ ROS NOETIC .....                                    | 19        |
| 3.1.2 ΔΗΜΙΟΥΡΓΙΑ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ROS NOETIC .....                       | 19        |
| 3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΕΡΓΑΛΕΙΩΝ .....     | 20        |
| 3.2.1 ROSARIA, (LIBRARY) ARIACODA ΚΑΙ ROSARIA_CLIENT .....            | 20        |
| 3.2.3 SICK LMS500-20000 .....                                         | 21        |
| 3.2.2 ΠΑΚΕΤΑ PIONEER 3-AT .....                                       | 22        |
| 3.2.3 ΠΑΚΕΤΑ SLAM-GMAPPING .....                                      | 23        |
| 3.3 ΟΛΟΚΛΗΡΩΣΗ ΥΛΟΠΟΙΗΣΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ AMCL .....                  | 24        |
| <b>ΚΕΦΑΛΑΙΟ 4 ΚΩΔΙΚΑΣ .....</b>                                       | <b>27</b> |
| 4.1 Ο ΚΩΔΙΚΑΣ ΜΕ ΠΡΟΣΘΗΚΗ ΤΟΥ XACRO ΑΡΧΕΙΟΥ ΤΟΥ ΡΟΜΠΟΤ .....          | 27        |
| 4.2 ΑΡΧΕΙΑ ΚΩΔΙΚΑ ΓΙΑ ΤΟ ΡΟΜΠΟΤ .....                                 | 28        |
| 4.2.1 ΤΑ LAUNCH ΑΡΧΕΙΑ ΤΟΥ ΡΟΜΠΟΤ ΓΙΑ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΣΤΟ RVIZ ..... | 28        |
| 4.3 ΤΟ LAUNCH ΑΡΧΕΙΟ ΓΙΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΧΑΡΤΟΓΡΑΦΗΣΗΣ .....         | 29        |

|                                                                                                                            |                  |
|----------------------------------------------------------------------------------------------------------------------------|------------------|
| <b>4.4 ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ LAUNCH ΑΡΧΕΙΟΥ ΓΙΑ ΤΟΝ ΑΙΣΘΗΤΗΡΑ SICK LMS500-20000 .....</b>                                     | <b>30</b>        |
| <b>4.5 ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ LAUNCH ΑΡΧΕΙΟΥ AMCL_DEMO ΤΟ ΟΠΟΙΟ ΣΤΗ ΣΥΝΕΧΕΙΑ ΣΗΚΩΝΕΙ ΤΟ AMCL.LAUNCH ΚΑΙ ΤΟ MOVE_BASE .....</b> | <b>31</b>        |
| 4.5.1 ΤΟ ΑΡΧΕΙΟ AMCL .....                                                                                                 | 32               |
| 4.5.2 ΤΟ ΑΡΧΕΙΟ MOVE BASE.....                                                                                             | 33               |
| <b>4.6 ΤΑ CONFIG ΑΡΧΕΙΑ ΓΙΑ ΛΕΙΤΟΥΡΓΕΙΑ ΤΟΥ AMCL, MOVE_BASE.....</b>                                                       | <b>33</b>        |
| 4.6.1 ΤΟ ΑΡΧΕΙΟ COSTMAP_GLOBAL_STATIC.....                                                                                 | 33               |
| 4.6.2 ΤΟ ΑΡΧΕΙΟ COSTMAP_COMMON.....                                                                                        | 34               |
| 4.6.3 ΤΟ ΑΡΧΕΙΟ COSTMAP_LOCAL.....                                                                                         | 34               |
| 4.6.4 ΤΟ ΑΡΧΕΙΟ COSTMAP_GLOBAL_LASER.....                                                                                  | 34               |
| 4.6.5 ΤΟ ΑΡΧΕΙΟ PLANNER .....                                                                                              | 35               |
| <b>4.7 ΤΑ ΑΡΧΕΙΑ ΤΟΥ ΧΑΡΤΗ ΠΟΥ ΔΗΜΙΟΥΡΓΗΣΑ .....</b>                                                                       | <b>36</b>        |
| 4.7.1 ΤΟ LAUNCH ΑΡΧΕΙΟ ΤΟΥ ΧΑΡΤΗ ΠΟΥ ΔΗΜΙΟΥΡΓΗΣΑ .....                                                                     | 36               |
| 4.7.2 ΤΟ ΑΡΧΕΙΟ .YAML ΤΟΥ ΧΑΡΤΗ .....                                                                                      | 36               |
| 4.7.3 ΤΟ ΑΡΧΕΙΟ .PGM (ΦΩΤΟΓΡΑΦΙΑ) ΤΟΥ ΧΑΡΤΗ .....                                                                          | 36               |
| <br><b><u>ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u></b>                                                                             | <b><u>37</u></b> |
| <br><b><u>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u></b>                                                                                        | <b><u>38</u></b> |

# ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ

---

## 1.1 Εισαγωγή στη ρομποτική

---

Η ρομποτική είναι ένας νέος κλάδος που προέρχεται από την τεχνολογική ανάπτυξης της επιστήμης των υπολογιστών αλλά και της μηχανικής και της ηλεκτρονικής, ο οποίος ασχολείται με την σχεδίαση και τη λειτουργία διάφορων μηχανικών εξαρτημάτων, που αποτελούν το ρομπότ σε διάφορες διατάξεις. Τα οποία, ελέγχονται και διαχειρίζονται από τον ειδικευόμενο, μέσω ενός λειτουργικού συστήματος (ROS). Σκοπός της ανάπτυξης αυτού του κλάδου είναι η αυτοματοποίηση ορισμένων διεργασιών κυρίως στις βιομηχανίες μαζικής παραγωγής όπου γίνεται αντικατάσταση του ανθρώπινου δυναμικού με εξειδικευμένες μηχανές που εκτελούν μια προκαθορισμένη σειρά διεργασιών. Επίσης, προσφέρει αύξηση κέρδους, βελτιωμένη ποιότητα και μεγαλύτερο έλεγχο των μέσων παραγωγής.

Παρόλα αυτά προσφέρει και άλλα πολλαπλά οφέλη και σε άλλους τομείς όπως η ιατρική, η εξερεύνηση του διαστήματος, έρευνα και διάσωση του περιβάλλοντος και στις αγροτικές και οικιακές εργασίες. Καθώς με την έρευνα και την δημιουργία μέσων και εργαλείων συντελεί και προσφέρει μεθόδους και μέσα για αναζήτηση, έρευνα και αντιμετώπιση διαφόρων προβλημάτων όπως αναφέρεται στα παραπάνω ζητήματα.[1]

## 1.2 Εισαγωγή στο ROS

---

Το ROS (Robot Operating System) παρά το όνομα του δεν είναι ένα λειτουργικό σύστημα αλλά είναι ένα SDK (Software Development Kit) δηλαδή, ένα κιτ ανάπτυξης λογισμικού ανοιχτού κώδικα για εφαρμογές ρομποτικής. Το ROS προσφέρει μια τυπική πλατφόρμα λογισμικού σε προγραμματιστές σε όλες τις βιομηχανίες που θα τους μεταφέρει από την έρευνα στην ανάπτυξη και την κατασκευή πρωτοτύπων.[2]

Για περισσότερα από 10 χρόνια, το ROS έχει δημιουργήσει μια τεράστια ομάδα λογισμικού για τη ρομποτική, δημιουργώντας μια παγκόσμια κοινότητα εκατομμυρίων προγραμματιστών και χρηστών που συμβάλλουν και βελτιώνουν αυτό το λογισμικό. Το ROS αναπτύσσεται από την κοινότητα αλλά και για την ίδια. Καθώς, θα είναι εκείνη η κάτοχος της όχι μόνο για το κοντινό παρών αλλά και για το μακρινό μέλλον.

Το ROS βασίζεται σε όλη τη βιομηχανία της ρομποτικής. Είναι ο κανόνας για τη διδασκαλία της ρομποτικής. Είναι η βάση για τις περισσότερες έρευνες στη ρομποτική, από έργα ενός μαθητή έως συνεργασίες πολλών ιδρυμάτων και διαγωνισμούς μεγάλης κλίμακας. Και είναι μέσα σε ρομπότ που τρέχουν σε παραγωγή σε όλο τον κόσμο σήμερα.

Το ROS παρέχει τα εργαλεία, τις βιβλιοθήκες και τις δυνατότητες που χρειάζεται κάθε προγραμματιστής για να αναπτύξει τις εφαρμογές ρομποτικής του, με ελευθερία και να το ενσωματώσει εύκολα στο υπάρχον υλικό του.[3]

Στον πυρήνα του, το ROS παρέχει ένα σύστημα μετάδοσης μηνυμάτων, που συχνά ονομάζεται "μεσαίο λογισμικό" ή "διοχέτευση". Η επικοινωνία είναι μια από τις πρώτες ανάγκες που προκύπτουν κατά την εφαρμογή μιας νέας εφαρμογής ρομπότ ή πραγματικά οποιουδήποτε συστήματος λογισμικού που θα αλληλεπιδράσει με το υλικό.

Για να δημιουργήσετε εφαρμογές αποτελεσματικά, χρειάζεστε καλά εργαλεία προγραμματιστή. Το ROS τα διαθέτει, όπως: εκκίνηση, ενδοσκόπηση, εντοπισμός σφαλμάτων, οπτικοποίηση,

σχεδίαση, καταγραφή και αναπαραγωγή. Αυτά τα εργαλεία επιταχύνουν την πρόοδο της ομάδας προγραμματιστή σας και μπορούν να συμπεριληφθούν στο προϊόν σας όταν το αποστέλλετε.

Εν κατακλείδι, ο στόχος του έργου ROS είναι να ανεβάζει συνεχώς τον πήχη σε ό,τι θεωρείται δεδομένο, και έτσι να μειώνει το εμπόδιο εισόδου στην κατασκευή εφαρμογών ρομπότ. Οποιοσδήποτε έχει μια καλή ιδέα για ένα χρήσιμο (ή διασκεδαστικό ή ενδιαφέρον) ρομπότ θα πρέπει να είναι σε θέση να κάνει αυτή την ιδέα πραγματικότητα, χωρίς να χρειάζεται να κατανοήσει τα πάντα σχετικά με το υποκείμενο υλικό και λογισμικό.[2]



Εικόνα 1, Εικόνα περιγραφής του ROS σύμφωνα με τα παραπάνω (<https://www.ros.org/blog/ecosystem/>)

### 1.3 Τι είναι το Pioneer 3-AT

Το Pioneer 3-AT είναι ένα ρομπότ γενικής χρήσης με τέσσερις τροχούς, με πολλές δυνατότητες συνδεσιμότητας με αρκετό εξοπλισμό, όπως αισθητήρες, ενσωματωμένο υπολογιστή, αυτόνομο λογισμικό πλοήγησης, γάντζο και πολλά άλλα. Το συγκεκριμένο μοντέλο χρησιμοποιεί τα προγράμματα οδήγησης "rosaria", "rosarni" ή "p2os".[4]

Το Pioneer 3-AT είναι ιδανικό για λειτουργία παντός εδάφους ή εργαστηριακό πειραματισμό ανήκει στην οικογένεια του (AMR). Και συνοδεύεται από μία μπαταρία, διακόπτη έκτακτης ανάγκης, κωδικοποιητές τροχών και μικροελεγκτή το υλικό λογισμικού ARCOS, καθώς και προηγμένο πακέτο ανάπτυξης λογισμικού ρομποτικής για φορητές συσκευές Pioneer SDK.[5]

Τα ερευνητικά ρομπότ Pioneer είναι τα πιο δημοφιλή έξυπνα κινητά στον κόσμο ρομπότ για εκπαίδευση και έρευνα. Η ευελιξία, η αξιοπιστία τους και η ανθεκτικότητα τα έχει κάνει την προτεινόμενη πλατφόρμα για προηγμένα έξυπνα ρομπότ. Τα Pioneers είναι συναρμολογημένα, προσαρμόσιμα, αναβαθμίζονται πάρα πολύ εύκολα με πληθώρα εργαλείων και αρκετά ανθεκτικά για να αντέξουν για χρόνια εργαστηριακής χρήσης και εκπαιδευτικής χρήσης.[5]



Εικόνα 2, Pioneer 3-AT model

(<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.generationrobots.com%2Fmedia%2FPioneer3AT-P3AT-RevA->



[datasheet.pdf&psig=AOvVaw1Ed7Qn8V4CsRihAcRy1MVy&ust=1718215653168000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCJiPtCS1IYDFQAAAAAdAAAAABAE\)](https://www.cyberbotics.com/doc/guide/pioneer-3at?version=R2021a#pioneer-3-at-model)

Αυτό που το κάνει κατάλληλο για αυτή τη χρήση είναι ότι χρησιμοποιείται για την έρευνα και την κατασκευή πρωτότυπων εφαρμογών που περιλαμβάνουν χαρτογράφηση, πλοήγηση, παρακολούθηση, αναγνώριση και άλλες συμπεριφορές. Αυτό το μοντέλο περιλαμβάνει υποστήριξη για 4 κινητήρες και 16 αισθητήρες σόναρ (8 προς τα εμπρός, 8 προς τα πίσω) για μετρήσεις εγγύτητας.[6]

Ο παρακάτω πίνακας παρουσιάζει μερικά από τα χαρακτηριστικά του ρομπότ σχετικά με το ύψος, το βάρος και τις διαστάσεις του μοντέλου που έχω χρησιμοποιήσει.

| Characteristics             | Values    |
|-----------------------------|-----------|
| Length                      | 508 mm    |
| Width                       | 497 mm    |
| Height                      | 277 mm    |
| Weight                      | 12 kg     |
| Max. forward/backward speed | 0.7 m/s   |
| Rotation Speed              | 140°/s    |
| Traversable Step            | 10 cm     |
| Traversable Gap             | 15 cm     |
| Battery Voltage             | 12V       |
| Maximum battery use         | 3 Battery |

Πίνακας 1, Μερικά χαρακτηριστικά του ρομπότ Pioneer 3-AT (<https://www.cyberbotics.com/doc/guide/pioneer-3at?version=R2021a#pioneer-3-at-model>)( <https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf>)

## 1.4 Τι είναι το AMR (Autonomous Mobile Robot)

Τα αυτόνομα φορητά ρομπότ, ή AMR (Autonomous Mobile Robot), είναι μια κατηγορία ρομπότ που έχουν σχεδιαστεί για κίνηση και μεταφορά αντικειμένων στο χώρο. Μπορούν να πλοηγηθούν δυναμικά, προσαρμόζοντας τη διαδρομή τους με βάση τα δεδομένα σε πραγματικό χρόνο που συλλέγουν από το περιβάλλον τους, χωρίς την παραμικρή επέμβαση από τον χειριστή. Συνήθως, αυτό επιτυγχάνεται μέσω μιας σειράς εξελιγμένων ενσωματωμένων αισθητήρων, υπολογιστών και χαρτών, που επιτρέπουν στα AMR να κατανοούν και να ερμηνεύουν το περιβάλλον τους προκειμένου να δημιουργήσουν μια εναλλακτική διαδρομή εάν εμφανιστεί ένα νέο εμπόδιο ώστε να φέρνουν εις πέρας τη διαδρομή που τους έχει ανατεθεί.[7][8]

Σε ένα περιβάλλον αποθήκης, αυτές οι εξελιγμένες τεχνολογίες συνεργάζονται με το λογισμικό εκτέλεσης αποθήκης WES(Warehouse Execution Software), προκειμένου να παραχωρήσουν στους AMR αυξημένη ευελιξία για να δημιουργήσουν τις δικές τους διαδρομές μεταξύ τοποθεσιών εντός

μιας αποθήκης ή εγκατάστασης, εντοπίζοντας και αποφεύγοντας τα εμπόδια και κάνοντας επαναδρομολόγηση όταν χρειάζεται.[7][8]

## 1.5 Τι είναι το SICK LMS500-20000 PRO

---

### 1.5.1 Sick LMS500 laser scanner

---

Ο σαρωτής λέιζερ Sick LMS500 είναι ένας αισθητήρας λέιζερ (μετρητής απόστασης) με μεσαίο εύρος ακόμη και κάτω από δύσκολες κλιματολογικές συνθήκες χάρη στην τεχνολογία Multi-echo technology που διαθέτει. Το συμπαγές περίβλημα με βαθμολογία IP 67 με ενσωματωμένη θέρμανση καθιστά το LMS500 έναν ανθεκτικό αισθητήρα λέιζερ κατάλληλο για πολλά περιβάλλοντα με σοβαρούς περιορισμούς. Ο συγκεκριμένος αισθητήρας έχει εύρος κάλυψης 190° όπως φαίνεται και παρακάτω στα χαρακτηριστικά αλλά και στην (Εικόνα 6).[9]

Τα κύρια πλεονεκτήματα του σαρωτή λέιζερ Sick LMS500 είναι η χαμηλή κατανάλωση ενέργειας, η γρήγορη επεξεργασία σήματος, πολλές εισόδους και εξόδους και τέλος, η δυνατότητα συγχρονισμού πολλών αισθητήρων μαζί. Η μειωμένη κατανάλωση ενέργειας καθώς και οι έξυπνες λειτουργίες αυτοελέγχου του εξασφαλίζουν εξαιρετικά χαμηλό συνολικό κόστος ιδιοκτησίας. Ο LMS500 διαθέτει την καλύτερη αναλογία τιμής-απόδοσης στην αγορά σε αυτήν την κατηγορία αισθητήρων. [9][10][11]



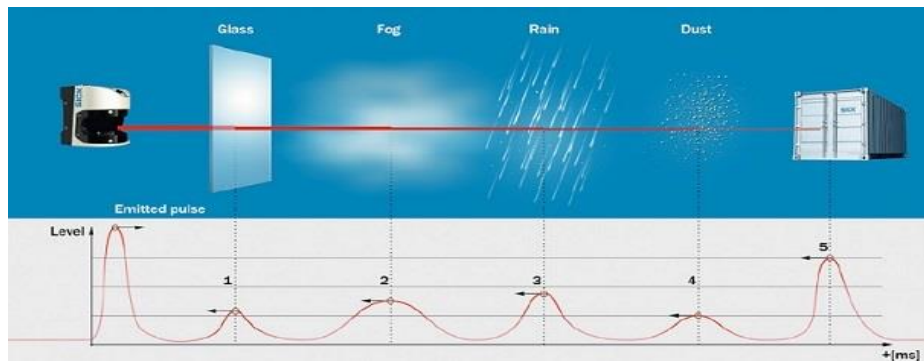
Εικόνα 3, Laser SICK LMS500-20000 PRO

(<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.sick.com%2Fus%2Fen%2Fcatalog%2Fproducts%2Flidar-and-radar-sensors%2Flidar-sensors%2Fflms5xx%2Fflms500-20000-pro%2Fp%2Fp216241&psig=AOvVaw3Uz3Weg01FEik3i85VRWbX&ust=1718287139633000&source=images&cd=vfe&opi=89978449&ved=0CBiQjRxqFwoTCICt48qc1oYDFQAAAAAdAAAAABao>)

### 1.5.2 Multi-echo technology

---

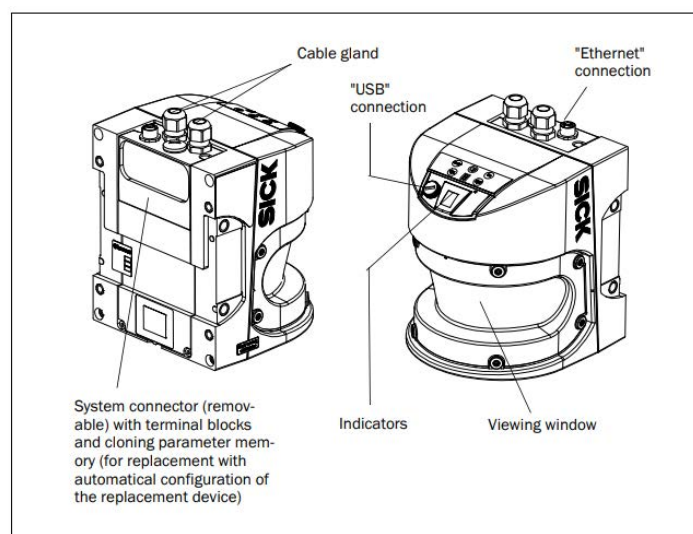
Το περιβάλλον και ο καιρός μπορεί να προκαλέσουν παρεμβολές. Η ομίχλη, η σκόνη, η βροχή και ακόμη και ένα παράθυρο είναι όλα πηγές παρεμβολών για έναν σαρωτή λέιζερ. Χάρη στην αποκλειστική τεχνολογία Multi-echo technology, οι σαρωτές λέιζερ της Sick προσφέρουν μεγαλύτερη αντοχή στις παρεμβολές. Μπορούν να λάβουν πολλαπλές πληροφορίες οπτικού σήματος μιας εκπομπής λέιζερ, που τους επιτρέπει να διακρίνουν την πραγματική μέτρηση από την παρεμβολή, όπως φαίνεται στη παρακάτω Εικόνα 4.[9][10][11]



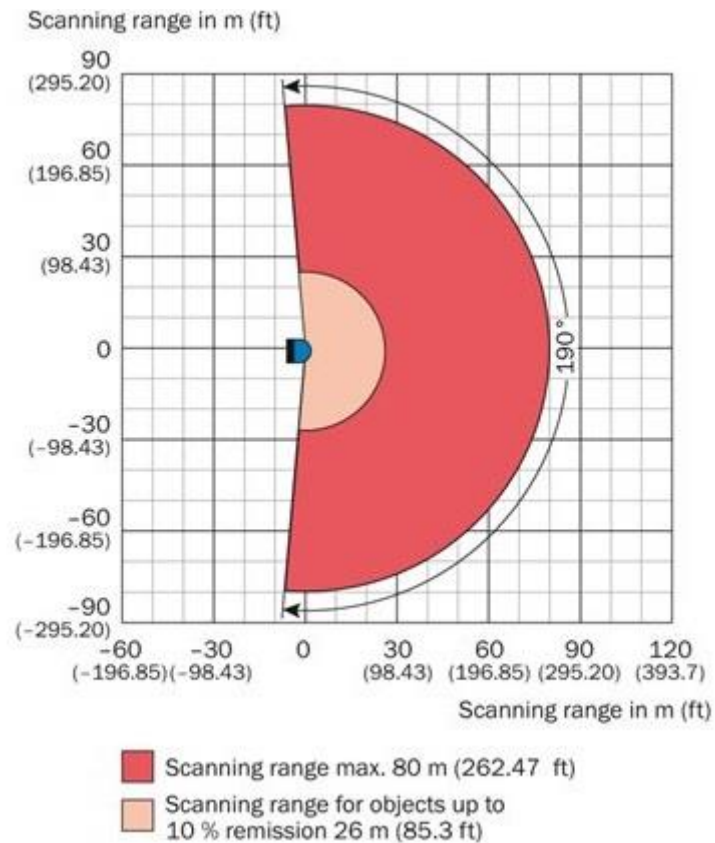
Εικόνα 4, Παρουσίαση της Multi-echo τεχνολογίας(<https://www.generationrobots.com/en/401697-sick-lms500-20000-pro-hr-indoor-laser-scanner.html>)

Παρακάτω παρουσιάζονται κάποια από τα σημαντικά χαρακτηριστικά του αισθητήρα. Ποιο συγκεκριμένα: τα ηλεκτρολογικά, τα μηχανολογικά χαρακτηριστικά αλλά και τα χαρακτηριστικά της συνδέσεως Ethernet από την οποία δεχόμαστε πληροφορίες για τις μετρήσεις του αισθητήρα λέιζερ:[9][10]

- Mechanical and electrical data
  - o Electrical connection: 1 x system connector with screw terminal block
  - o Supply voltage: 24 V DC
  - o Power: 22 W
  - o Scanning angle: 190 °
  - o Enclosure rating: IP 65 (EN 60529, paragraph 14.2.7)
  - o Weight: 3.7 kg
  - o Dimensions: 160 mm × 155 mm × 185 mm
  - o Fog correction: No
  - o Operating range: 0 m ... 80 m
  - o Max. range with 10 % reflectivity: 26 m
- Ethernet
  - o Function (Ethernet): Host
  - o Data transmission speed (Ethernet): 10/100 Mbits/s
  - o Protocol (Ethernet): TCP/IP, OPC [9][10]



Εικόνα 5, Περιφερειακές επεξηγήσεις του Laser SICK LMS500-20000 PRO ([https://www.sick.com/media/docs/4/14/514/Operating\\_instructions\\_Laser\\_Measurement\\_Sensors\\_of\\_the\\_LMS5xx\\_Product\\_Family\\_en\\_IM0037514.PDF](https://www.sick.com/media/docs/4/14/514/Operating_instructions_Laser_Measurement_Sensors_of_the_LMS5xx_Product_Family_en_IM0037514.PDF))



Εικόνα 6, Εύρος κάλυψης μετρήσεων του αισθητήρα SICK LMS500-20000 PRO

(<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.generationrobots.com%2Fen%2F401697-sick-lms500-20000-pro-hr-indoor-laser-scanner.html&psig=AOvVaw252qksxk9NtJYEI2wM5sai&ust=1718287701301000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCLjD19ae1oYDFQAAAAAdAAAAABAZ>)

# ΚΕΦΑΛΑΙΟ 2 ΕΞΟΠΛΙΣΜΟΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

## 2.1 Εξοπλισμός

### 2.1.1 Λειτουργικό Σύστημα

Αρχικά για την χρήση του ROS θα εγκαταστήσουμε το λειτουργικό σύστημα Linux και πιο συγκεκριμένα τα UBUNTU 20.04.6 LTS (Focal Fossa). Το λειτουργικό σύστημα είναι ανοιχτού κώδικα για υπολογιστές και τροφοδοτεί εκατομμύρια υπολογιστές και φορητούς υπολογιστές σε όλο τον κόσμο.[12]

Πολλοί από τους μεγαλύτερους κατασκευαστές υπολογιστών στον κόσμο πιστοποιούν τους φορητούς υπολογιστές και τους επιτραπέζιους υπολογιστές τους για το UBUNTU, από ηλεκτρονικούς φορητούς υπολογιστές έως σταθμούς εργασίας υψηλής τεχνολογίας. Το πιστοποιημένο υλικό UBUNTU έχει περάσει την εκτεταμένη διαδικασία δοκιμών και αναθεώρησης, διασφαλίζοντας ότι το UBUNTU λειτουργεί σωστά και υπάρχουν και μεγάλες εταιρίες να το επαληθεύουν.[12]

Αυτό που κάνει ιδανικά τα UBUNTU για αυτή την υλοποίηση της εργασίας αλλά και για κάθε υλοποίηση είναι ότι είναι ένα ελαφρύ λειτουργικό. Το οποίο είναι αρκετά αναγκαίο γιατί η υλοποίηση έχει ανάγκη για τιμές πραγματικού χρόνου δηλαδή τιμές και διοχετεύσεις που θα δίνει ή θα λαμβάνει το ROS μέσω των διοχετεύσεων του UBUNTU για την ομαλή και στιγμιαία συνδεσιμότητα.[12]

Partners on board



Εικόνα 7, Εταιρίες υποστήριξης και συντάιροι για τα Ubuntu (<https://ubuntu.com/certified>)

### 2.1.1 Σύνδεση του αισθητήρα με το Pioneer 3-AT

Ο αισθητήρας SICK LMS500 σύμφωνα με τη βιβλιογραφία έχει ως κύρια τροφοδοσία τα 24V και το τροχοφόρο ρομπότ έχει τα 12V όπως παρουσιάζονται στα χαρακτηριστικά του αισθητήρα παραπάνω (κάτω από την Εικόνα 4) και στον Πίνακα 1 αντίστοιχα. Για την τροφοδοσία του αισθητήρα από το ρομπότ προκειμένου να δώσει μετρήσεις για την υλοποίηση της εργασίας θα πρέπει να γίνουν κάποιες αλλαγές σε σχέση με το βασικό μοντέλο τροφοδοσίας του αισθητήρα. Χρησιμοποιήθηκε ένας μετατροπέας τάσης ο οποίος είχε ως είσοδο τροφοδοσίας από τη μητρική πλακέτα του Pioneer 3-AT και παρουσιάζεται παρακάτω στην Εικόνα 8. Στη συνέχεια χρειαζόταν ρύθμιζα μέσω του ποτενσιόμετρου που φαίνεται στην Εικόνα 8 με τη χρήση του πολύμετρου ώστε να γίνει σωστά η ρύθμιση στα 24 Volt και έπειτα στην έξοδο συνδεόντουσαν οι είσοδοι του αισθητήρα. [5][9][10]



Εικόνα 8, Μετατροπέας Τάσης

([## 2.2 Πακέτα\(RosAria\) για την επικοινωνία με το Pioneer 3-AT](https://www.google.com/aclk?sa=l&ai=DChcSEwi7id2akNmGAXWoi4MHHaeRJYYABAKGgJIZg&gclid=Cj0KCQjwsaqzBhDdARIsAK2gqne5Z2ca55OMx5xr16oTcW_rxtPQmgaUtIBYFmClpHyQ8P-C99zVIlkaAg0YEALw_wcB&sig=AOD64_2SK5k2WdVAS7BnS8aKuJWQUN7tJQ&ctype=5&q=&ved=2ahUKEwiikdiakNmGAXWph_0HHbw9AdkQ9aACKAB6BAgGEF8&adurl=) </a></p>
</div>
<div data-bbox=)

Το ROSARIA είναι μια γέφυρα διασύνδεσης μεταξύ του ROS και της βιβλιοθήκης Aria (Advanced Robotics Interface for Applications) της MobileRobots. Η βιβλιοθήκη Aria, είναι μια βιβλιοθήκη C++ που προσφέρει επικοινωνία με το ROS μέσω διοχετεύσεων, υποστηρίζει όλα τα ρομπότ της Adept MobileRobots, MobileRobots Inc.. Στην παρούσα υλοποίηση, χρησιμοποιείται για την επικοινωνία με το ρομπότ Pioneer 3-AT. Λειτουργώντας ως κόμβος, το ROSARIA δημοσιεύει topics που περιλαμβάνουν δεδομένα από τον ενσωματωμένο ελεγκτή του ρομπότ (ARCOS), τα οποία προέρχονται από την βιβλιοθήκη Aria. Με αυτό τον τρόπο, επιτρέπει στους χρήστες να καθορίζουν την ταχύτητα, την επιτάχυνση είτε γραμμική είτε στροφική και άλλες εντολές για το ρομπότ μέσω της Aria. Η βιβλιοθήκη Aria όμως δεν βρίσκεται διαθέσιμη λόγω της παλαιότητας της, γι' αυτό στην παρούσα εργασία, δούλεψα και εγκατέστησα την βιβλιοθήκη Aria Coda η οποία είναι βασισμένη στην παλαιότερη βιβλιοθήκη (Aria) στην οποία οδηγήθηκα μέσω των πηγών. Μέσω της RosAria, δημιουργούνται topics με πρόθεμα /RosAria/.... Τα οποία μπορούν να αντικατασταθούν με άλλα ονόματα αν επιθυμούμε. Τα topics είναι κανάλια επικοινωνίας τα οποία δίνουν τη δυνατότητα σε άλλους κόμβους /node να επικοινωνήσουν μεταξύ τους στέλνοντας ή διαβάζοντας μηνύματα. Οι παράμετροι χρησιμοποιούνται για τη διαμόρφωση της συμπεριφοράς του κόμβου RosAria κατά την αρχική εκκίνηση του κόμβου, όπως ο τρόπος σύνδεσης με τον ελεγκτή ρομπότ. Μπορούν να οριστούν στη γραμμή εντολών κατά την εκτέλεση του roslaunch ή σε ένα αρχείο εκκίνησης κατά τη χρήση του roslaunch. Οι παράμετροι αποθηκεύονται από τον διακομιστή παραμέτρων ROS. Τα ROS services από την άλλη είναι κανάλια μέσω των οποίων άλλοι κόμβοι μπορούν να καλέσουν συγκεκριμένα συμβάντα στο ρομπότ μέσω του ROSARIA. [13][14]

| Topic Name            | Type                   | Description                        |
|-----------------------|------------------------|------------------------------------|
| /RosAria/cmd_vel      | geometry_msgs/Twist    | Λαμβάνει νέες εντολές ταχύτητας.   |
| /RosAria/pose         | nav_msgs/Odometry      | Δημοσιεύει πληροφορίες οδομετρίας. |
| /RosAria/bumper_state | rosaria/BumperState    | Δημοσιεύει καταστάσεις bumper.     |
| /RosAria/sonar        | sensor_msgs/PointCloud | Δημοσιεύει αναγνώσεις sonar.       |

|                                  |                         |                                                                         |
|----------------------------------|-------------------------|-------------------------------------------------------------------------|
| /RosAria/sonar_pointcloud2       | sensor_msgs/PointCloud2 | Δημοσιεύει αναγνώσεις sonar ως τύπο sensor_msgs/PointCloud2.            |
| /RosAria/battery_state_of_charge | std_msgs/Float32        | Δημοσιεύει την ποσοστιαία κατάσταση φόρτισης της μπαταρίας.             |
| /RosAria/battery_voltage         | std_msgs/Float64        | Δημοσιεύει τη μέτρηση τάσης της μπαταρίας.                              |
| /RosAria/battery_recharge_state  | std_msgs/Int8           | Δημοσιεύει την κατάσταση επαναφόρτισης.                                 |
| /RosAria/motors_state            | std_msgs/Bool           | Υποδεικνύει εάν οι κινητήρες είναι ενεργοποιημένοι ή απενεργοποιημένοι. |

Πίνακας 2, Τα topic του RosAria (<https://wiki.ros.org/ROSARIA#RosAria>)

| Service Name            | Type           | Description                   |
|-------------------------|----------------|-------------------------------|
| /RosAria/enable_motors  | std_srvs/Empty | Ενεργοποιεί τους κινητήρες.   |
| /RosAria/disable_motors | std_srvs/Empty | Απενεργοποιεί τους κινητήρες. |

Πίνακας 3, Τα Services του RosAria (<https://wiki.ros.org/ROSARIA#RosAria>)

| Parameter Name      | Type   | Description                                    |
|---------------------|--------|------------------------------------------------|
| port                | string | Η σειριακή θύρα του ρομπότ.                    |
| baud                | int    | Το ρυθμό baud για τη σειριακή επικοινωνία.     |
| depug_aria          | bool   | Ενεργοποιεί τη λειτουργία αποσφαλμάτωσης ARIA. |
| aria_log_filename   | string | Το αρχείο καταγραφής ARIA.                     |
| publish_aria_lasers | bool   | Δημοσιεύει δεδομένα laser από το ARIA.         |
| cmd_vel_timeout     | float  | Το χρονικό όριο για τις εντολές cmd_vel.       |
| odom_frame          | string | Το όνομα για το tf της οδομετρίας.             |
| base_link_frame     | string | Το όνομα για το tf της βάσης.                  |
| bumpers_frame       | string | Το όνομα για το tf των bumper.                 |
| sonar_frame         | string | Το όνομα για το tf των sonar.                  |

Πίνακας 4, Οι Parameters του RosAria (<https://wiki.ros.org/ROSARIA#RosAria>)



## 2.3 Πακέτα για την Υλοποίηση

### 2.3.1 Πακέτα για το Laser SICK Lms-500

Το συγκεκριμένο περιέχει πακέτα του ROS για σαρωτές λέιζερ SICK. Αυτό το πρόγραμμα οδήγησης παρέχει τα δεδομένα μέτρησης ως δεδομένα PointCloud2 και LaserScan για όλα. Με την βοήθεια τους παρέχουν τα ακατέργαστα δεδομένα για κάθε επίπεδο σάρωσης σε διαφορετικό πλαίσιο συντεταγμένων. Αλλά δεν υπάρχει μετασχηματισμός συντεταγμένων μεταξύ των επιπέδων σάρωσης, που να μπορούν να περιγραφούν με μηνύματα μετασχηματισμού tf, επομένως δεν δημοσιεύονται μηνύματα tf και θα πρέπει να γίνουν οι απαραίτητες αλλαγές για τον καθορισμό των μετασχηματισμών οι οποίοι γίνονται στην συνέχεια (στο Κεφάλαιο 3) αλλά και άλλων μετασχηματισμών του κώδικα για την ομαλή και την ορθή λειτουργία του.[10][15]

### 2.3.2 Πακέτα Gmapping και SLAM-Gmapping

Αρχικά, αυτό το πακέτο περιέχει ένα περιτύλιγμα ROS για το Gmapping του OpenSlam. Το πακέτο Gmapping παρέχει SLAM που βασίζεται σε λέιζερ (Simultaneous Localization and Mapping), ως κόμβος του ROS που ονομάζεται slam\_gmapping. Χρησιμοποιώντας το slam\_gmapping, μπορείτε να δημιουργήσετε έναν δισδιάστατο χάρτη κατοχής πλέγματος (όπως μια κάτοψη κτιρίου) από δεδομένα λέιζερ και πόζας που συλλέγονται από ένα κινητό ρομπότ.[16]

Για να χρησιμοποιήσετε το slam\_gmapping, χρειάζεστε ένα κινητό ρομπότ που παρέχει δεδομένα οδομετρίας και να έχει έναν μετρητή απόστασης (αισθητήρα λέιζερ) σε σταθερή θέση με τις απαραίτητες μεταρρυθμίσεις και συσχετίσεις με το κέντρο του ρομπότ ή με κάποιο άλλο μέρος του. Ο κόμβος slam\_gmapping θα προσπαθήσει να μετατρέψει κάθε εισερχόμενη σάρωση στο πλαίσιο odom (odometry) tf. [17]

Επομένως, Ο κόμβος slam\_gmapping δέχεται μηνύματα sensor\_msgs/LaserScan και δημιουργεί έναν χάρτη (nav\_msgs/OccupancyGrid). Ο χάρτης μπορεί να ανακτηθεί μέσω ενός θέματος ή μιας υπηρεσίας ROS.[17]

Services:

| Services    | Type            | Description                                                 |
|-------------|-----------------|-------------------------------------------------------------|
| dynamic_map | nav_msgs/GetMap | Κλήση αυτής της υπηρεσίας για λήψη των δεδομένων του χάρτη. |

Πίνακας 5, services ([http://wiki.ros.org/gmapping#slam\\_gmapping](http://wiki.ros.org/gmapping#slam_gmapping))

Subscribed topics:

| Topic | Type                  | Description                                                                                         |
|-------|-----------------------|-----------------------------------------------------------------------------------------------------|
| /tf   | tf/tfMessage          | Μετασχηματισμοί απαραίτητοι για τη συσχέτιση των πλαισίων για το λέιζερ, τη βάση και την οδομετρία. |
| /scan | sensor_msgs/LaserScan | Σαρώσεις λέιζερ για τη δημιουργία του χάρτη.                                                        |

Πίνακας 6, subscribe topics ([http://wiki.ros.org/gmapping#slam\\_gmapping](http://wiki.ros.org/gmapping#slam_gmapping))



Published topics:

| Topic         | Type                   | Description                                                                                                    |
|---------------|------------------------|----------------------------------------------------------------------------------------------------------------|
| /map_metadata | nav_msgs/MapMetaData   | Λήψη των δεδομένων του χάρτη από αυτό το topic, το οποίο είναι λανθάνον και ενημερώνεται περιοδικά.            |
| /map          | nav_msgs/OccupancyGrid | Λήψη των δεδομένων του χάρτη από αυτό το topic, το οποίο είναι λανθάνον και ενημερώνεται περιοδικά.            |
| ~entropy      | std_msgs/Float64       | Εκτίμηση της εντροπίας της κατανομής της θέσης του ρομπότ (υψηλότερη τιμή υποδεικνύει μεγαλύτερη αβεβαιότητα). |

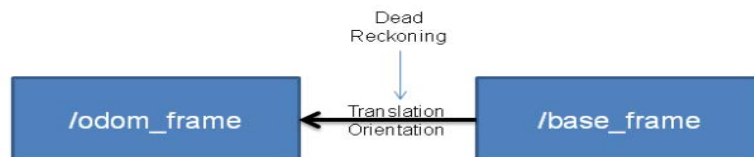
Πίνακας 7, publish topics ([http://wiki.ros.org/gmapping#slam\\_gmapping](http://wiki.ros.org/gmapping#slam_gmapping))

Οι μετασχηματισμοί που πρέπει να γίνουν είναι από το /scan στο /base\_link και θα πρέπει να είναι στατικές μεταβλητές θέσης. Επίσης, μεταδίδεται περιοδικά από ένα robot\_state\_publisher ή ένα tf static\_transform\_publisher. Από το base\_link στο odom το οποίο γίνεται αυτόματα αλλά αξίζει να τονιστεί, καθώς παρέχεται από το σύστημα οδομετρίας του ρομπότ και στην συγκεκριμένη περίπτωση. Τέλος, ο μετασχηματισμός που παρέχεται από την ενέργεια του πακέτου είναι από το /map στο /odom όπου γίνεται η εκτίμηση της τρέχουσας θέσης του ρομπότ μέσα στο πλαίσιο του χάρτη.[17]

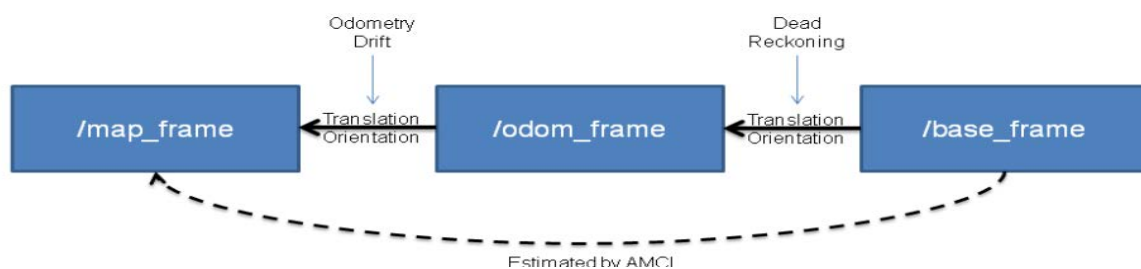
### 2.3.3 Πακέτο AMCL

Το amcl λαμβάνει έναν χάρτη που βασίζεται σε λέιζερ, σαρώνει με λέιζερ και μετασχηματίζει μηνύματα, και εξάγει εκτιμήσεις. Κατά την εκκίνηση, το amcl προετοιμάζει το φίλτρο σωματιδίων του σύμφωνα με τις παραμέτρους που παρέχονται. Επίσης, πρέπει να οριστούν οι παράμετροι και η αρχική κατάσταση φίλτρου για την σωστή λειτουργία του. Καθώς αν δε γίνει αυτό θα έχουμε ένα σύννεφο σωματιδίων μεσαίου μεγέθους με κέντρο περίπου το (0,0,0) το οποίο δεν είναι σωστό γιατί είναι μια αυθαίρετη τιμή και επηρεάζει σε μεγάλο βαθμό το σύστημα μας.[18]

Odometry Localization



AMCL Map Localization



Εικόνα 9, Πως μεταφέρονται τα δεδομένα μεταξύ των frames (<http://wiki.ros.org/amcl>)

Subscribed topics:

| Topic name                | Type                                                 | Description                                                                                                                             |
|---------------------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>/scan</code>        | <code>sensor_msgs/LaserScan</code>                   | Λαμβάνει δεδομένα από σαρωτή λέιζερ.                                                                                                    |
| <code>/tf</code>          | <code>tf/tfMessage</code>                            | Λαμβάνει μετασχηματισμούς συντεταγμένων.                                                                                                |
| <code>/initialpose</code> | <code>geometry_msgs/PoseWithCovarianceStamped</code> | Λαμβάνει την αρχική θέση και την συνδιακύμανση για την αρχικοποίηση του φίλτρου σωματιδίων.                                             |
| <code>/map</code>         | <code>nav_msgs/OccupancyGrid</code>                  | Λαμβάνει τον χάρτη με κυψελίδες πληρότητας όταν έχει οριστεί η παράμετρος <code>use_map_topic</code> , για εντοπισμό με βάση το λέιζερ. |

Πίνακας 8, Subscribed Topics για το AMCL ( <http://wiki.ros.org/amcl> )

Published topics:

| Topic Name     | Type                                    | Description                                                                                                                    |
|----------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| /amcl_pose     | geometry_msgs/PoseWithCovarianceStamped | Δημοσιεύει την εκτιμώμενη θέση του ρομπότ στον χάρτη, μαζί με τη διακύμανση.                                                   |
| /particlecloud | geometry_msgs/PoseArray                 | Δημοσιεύει το σύνολο των εκτιμήσεων θέσης (σωματίδια) που διατηρούνται από το φίλτρο.                                          |
| /tf            | tf/tfMessage                            | Δημοσιεύει τον μετασχηματισμό από το odom (ο οποίος μπορεί να επαναχαρτογραφηθεί μέσω της παραμέτρου ~odom_frame_id) στον map. |

Πίνακας 9, Published topics για το AMCL (<http://wiki.ros.org/amcl>)

Services:

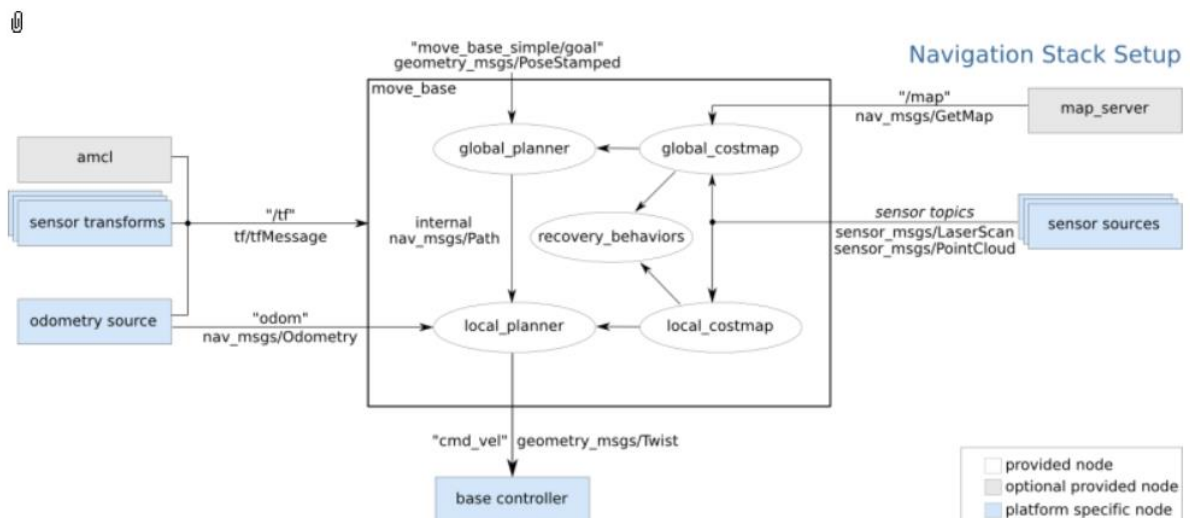
| Service Name            | Type            | Description                                                                                                                                                         |
|-------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| global_localization     | std_srvs/Empty  | Εκκινεί το global localization, διασπείροντας τα σωματίδια τυχαία στον χάρτη                                                                                        |
| request_nomotion_update | std_srvs/Empty  | Χειροκίνητη ενημέρωση και δημοσίευση ενημερωμένων σωματιδίων.                                                                                                       |
| set_map                 | nav_msgs/SetMap | Χειροκίνητη ενημέρωση και δημοσίευση ενημερωμένων σωματιδίων.                                                                                                       |
| static_map              | nav_msgs/GetMap | Ο κόμβος AMCL καλεί αυτή την υπηρεσία για να λάβει τον χάρτη που χρησιμοποιείται για εντοπισμό με βάση το λέιζερ. Η εκκίνηση αναστέλλεται μέχρι να ληφθεί ο χάρτης. |

Πίνακας 10, Services του AMCL (<http://wiki.ros.org/amcl>)

Κατά την παραλαβή της πρώτης σάρωσης λέιζερ, το amcl αναζητά τον μετασχηματισμό μεταξύ του πλαισίου του λέιζερ και του πλαισίου βάσης (~base\_frame\_id) και τον ασφαρίζει για πάντα. Άρα το amcl δεν μπορεί να χειριστεί ένα λέιζερ που κινείται σε σχέση με τη βάση.

Το παρακάτω σχέδιο δείχνει τη διαφορά μεταξύ εντοπισμού με χρήση οδομετρίας και amcl. Κατά τη λειτουργία το amcl εκτιμά τον μετασχηματισμό του βασικού πλαισίου (~base\_frame\_id) σε σχέση με το καθολικό πλαίσιο (~global\_frame\_id) αλλά δημοσιεύει μόνο τον μετασχηματισμό μεταξύ του καθολικού πλαισίου και του πλαισίου οδομετρίας (~odom\_frame\_id).[18]

### 2.3.4 Πακέτο Move\_base



Εικόνα 10, Διάγραμμα επικοινωνιών του Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

Ο κόμβος move\_base παρέχει μια διεπαφή ROS για διαμόρφωση, εκτέλεση και αλληλεπίδραση με τη στοίβα πλοήγησης σε ένα ρομπότ. Μια προβολή υψηλού επιπέδου του κόμβου move\_base και της αλληλεπίδρασής του με άλλα στοιχεία φαίνεται παραπάνω. Το μπλε ποικίλλει ανάλογα με την πλατφόρμα του ρομπότ, το γκρι είναι προαιρετικό αλλά παρέχεται για όλα τα συστήματα και οι λευκοί κόμβοι απαιτούνται αλλά παρέχονται και για όλα τα συστήματα.[19]

Subscribe topics:

| Topic name            | Type                              | Description                                                                                             |
|-----------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------|
| /move_base/goal       | move_base_msgs/MoveBaseActionGoal | Στόχος για το move_base να επιδιώξει να πάει στο σημείο.                                                |
| move_base/cancel      | actionlib_msgs/GoalID             | Αίτημα για ακύρωση συγκεκριμένου στόχου.                                                                |
| move_base_simple/goal | geometry_msgs/PoseStamped         | Μη-δράση διεπαφή για χρήστες που δεν ενδιαφέρονται για την παρακολούθηση της εκτέλεσης των στόχων τους. |

Πίνακας 11, Subscribed Topics για το Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

Published topics:

| Topic Name         | Type                                  | Description                                                                         |
|--------------------|---------------------------------------|-------------------------------------------------------------------------------------|
| move_base/feedback | move_base_msgs/MoveBaseActionFeedback | Παρέχει την τρέχουσα θέση της θέσης στον κόσμο.                                     |
| move_base/status   | actionlib_msgs/GoalStatusArray        | Παρέχει πληροφορίες κατάστασης για τους στόχους που στέλνονται στη δράση move_base. |

|                  |                                     |                                                            |
|------------------|-------------------------------------|------------------------------------------------------------|
| move_base/result | move_base_msgs/MoveBaseActionResult | Το αποτέλεσμα είναι κενό για τη δράση move_base.           |
| move_base/result | geometry_msgs/Twist                 | Ροή εντολών ταχύτητας που προορίζεται για εκτέλεση κίνησης |

Πίνακας 12, Published Topics για το Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

Services:

| Services name       | Type           | Description                                                                                                                                                                                                                          |
|---------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| make_plan           | clear_costmaps | Επιτρέπει σε έναν εξωτερικό χρήστη να ζητήσει ένα σχέδιο προς μια συγκεκριμένη θέση από το move_base χωρίς να εκτελέσει το σχέδιο.                                                                                                   |
| clear_unknown_space | std_srvs/Empty | Επιτρέπει σε έναν εξωτερικό χρήστη να ζητήσει από το move_base να καθαρίσει τον άγνωστο χώρο γύρω από το ρομπότ. Χρήσιμο όταν το move_base έχει σταματήσει τα costmaps για μεγάλο διάστημα και στη συνέχεια ξεκινά σε νέα τοποθεσία. |
| clear_costmaps      | std_srvs/Empty | Επιτρέπει σε έναν εξωτερικό χρήστη να καθαρίσει τα εμπόδια στα costmaps του move_base. Πρέπει να χρησιμοποιείται με προσοχή καθώς μπορεί να προκαλέσει σύγκρουση του ρομπότ με αντικείμενα.                                          |

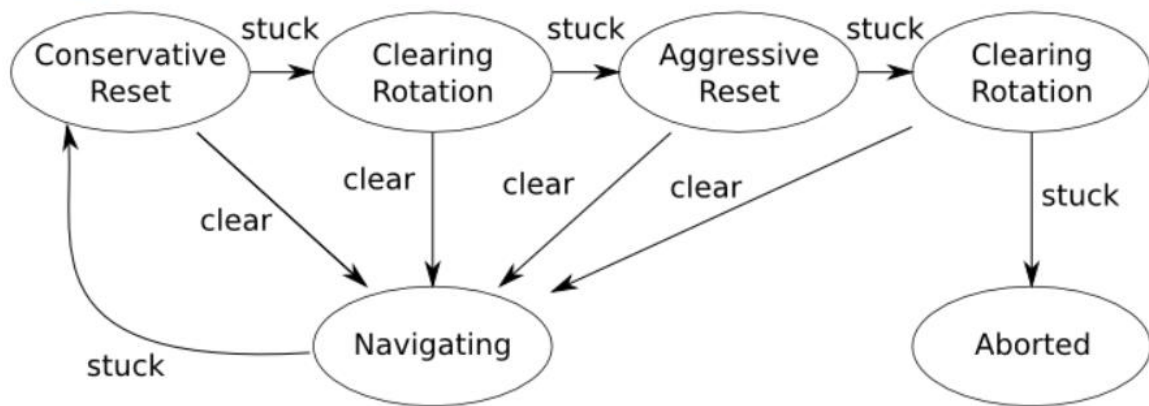
Πίνακας 13, Services του Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

Βασικά Parameters:

| Parameter Name      | Type   | Default Value                           | Description                                                                                                                                                             |
|---------------------|--------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| base_global_planner | string | navfn/NavfnROS                          | Το όνομα του plugin για τον παγκόσμιο προγραμματιστή που θα χρησιμοποιηθεί με το move_base. Το plugin πρέπει να συμμορφώνεται με τη διεπαφή nav_core::BaseGlobalPlanner |
| base_local_planner  | string | base_local_planner/TrajectoryPlannerROS | Το όνομα του plugin για τον τοπικό προγραμματιστή που θα χρησιμοποιηθεί με το move_base. Το plugin πρέπει να συμμορφώνεται με τη διεπαφή nav_core::BaseLocalPlanner.    |

Πίνακας 14, Βασικά Parameters του Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

### move\_base Default Recovery Behaviors



Εικόνα 11, Οι προεπιλεγμένες συμπεριφορές ανάκαμψης της Move\_base ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base))

Η εκτέλεση του κόμβου move\_base σε ένα ρομπότ που έχει διαμορφωθεί σωστά (δείτε την τεκμηρίωση της στοίβας πλοήγησης για περισσότερες λεπτομέρειες) έχει ως αποτέλεσμα ένα ρομπότ που θα επιχειρήσει να επιτύχει μια διαδρομή με τη βάση του προς μια κατεύθυνση που καθορίζεται από τον χρήστη. Ο κόμβος move\_base θα φτάσει τελικά σε αυτήν την θέση του στόχου του ή θα δώσει σήμα αποτυχίας προς τον χρήστη. Ο κόμβος move\_base μπορεί προαιρετικά να εκτελεί συμπεριφορές ανάκτησης όταν το ρομπότ αντιλαμβάνεται ότι έχει κολλήσει. Από προεπιλογή, ο κόμβος move\_base θα κάνει τις ακόλουθες ενέργειες για να προσπαθήσει να εκκαθαρίσει χώρο:

Πρώτα, τα εμπόδια έξω από τον καθορισμένο χάρτη (που έχει δώσει ο χρήστης) θα αφαιρεθούν. Στη συνέχεια, αν είναι δυνατόν, το ρομπότ θα πραγματοποιήσει μια περιστροφή χωρίς να αλλάξει θέση για να καθαρίσει τον χώρο. Αν αυτό αποτύχει, το ρομπότ θα καθαρίσει πιο επιθετικά τον χάρτη του, αφαιρώντας όλα τα εμπόδια γύρω από την συγκεκριμένη περιοχή. Έπειτα θα ακολουθηθεί άλλη μια περιστροφή επί τόπου (για επαναπροσδιορισμό των εμποδίων). Αν όλα αυτά αποτύχουν, το ρομπότ θα θεωρήσει ότι ο στόχος του είναι ανέφικτος και θα ειδοποιήσει τον χρήστη ότι έχει ακυρωθεί. Αυτές οι συμπεριφορές ανάκτησης μπορούν να ρυθμιστούν χρησιμοποιώντας την παράμετρο recovery\_behaviors και να απενεργοποιηθούν χρησιμοποιώντας την παράμετρο recovery\_behavior\_enabled. [19]



## ΚΕΦΑΛΑΙΟ 3 ΟΔΗΓΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ

### 3.1 Εγκατάσταση λειτουργικού και του ROS



Εικόνα 12, Το ρομπότ Pioneer 3-AT με τον αισθητήρα SICK LMS500 που χρησιμοποιήσα για την υλοποίηση της εργασίας

Αρχικά για την υλοποίηση θα χρειαστεί να γίνει εγκατάσταση λειτουργικού συστήματος (UBUNTU LINUX 20.04.6) και των βασικών πακέτων ROS, η έκδοση η οποία υποστηρίζεται από τη δική μας έκδοση LINUX. Παρουσιάζονται αναλυτικά παρακάτω οι οδηγίες και οι σύνδεσμοι όπου ακολούθησα για την εγκατάσταση και την υλοποίηση.

Οπότε έγινε εγκατάσταση:

- UBUNTU LINUX 20.04.6 LTS (Focal Fossa) (<https://releases.ubuntu.com/focal/>)
- ROS Noetic (<http://wiki.ros.org/noetic>)



Εικόνα 13,14, Οι εκδόσεις του UBUNTU και του ROS αντίστοιχα

([https://www.google.com/url?sa=i&url=https%3A%2F%2Fruncloud.io%2Fblog%2Fubuntu20&psig=AOvVaw1HliDhMg\\_nN\\_Y\\_4MR4Fkai&ust=1718985310644000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCLC71cDF6oYDFQAAAAAdAAAAABAK](https://www.google.com/url?sa=i&url=https%3A%2F%2Fruncloud.io%2Fblog%2Fubuntu20&psig=AOvVaw1HliDhMg_nN_Y_4MR4Fkai&ust=1718985310644000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCLC71cDF6oYDFQAAAAAdAAAAABAK))  
(<https://www.google.com/imgres?q=ROS%20noetic&imgurl=https%3A%2F%2Fros.org%2Fimgs%2Fros-noetic->

[ninemys.svg&imgrefurl=https%3A%2F%2Fros.org%2F&docid=mHvQSIHxB47PTM&tbnid=OMjlaDu5EdLbyM&vet=12ahUKewjy7K3yx-qGAxV-if0HHV1OC48QM3oECGgQAA..i&w=228&h=353&hcb=2&ved=2ahUKewjy7K3yx-qGAxV-if0HHV1OC48QM3oECGgQAA](https://ninemys.svg&imgrefurl=https%3A%2F%2Fros.org%2F&docid=mHvQSIHxB47PTM&tbnid=OMjlaDu5EdLbyM&vet=12ahUKewjy7K3yx-qGAxV-if0HHV1OC48QM3oECGgQAA..i&w=228&h=353&hcb=2&ved=2ahUKewjy7K3yx-qGAxV-if0HHV1OC48QM3oECGgQAA)

### 3.1.1 Εγκατάσταση ROS Noetic

---

Για την εγκατάσταση του ROS θα χρειαστεί να τρέξουμε ορισμένες γραμμές κώδικα στο terminal του συστήματος που στήσαμε, οπότε ακολουθούν οι γραμμές κώδικα και μερικές πληροφορίες για αυτά:

- `sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'`  
(Ρυθμίζει τον υπολογιστή ώστε να δέχεται λογισμικό από το packages.ros.org)
- `sudo apt install curl`
- `curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -`
- `sudo apt update` (βεβαιώνει ότι το ευρετήριο του πακέτου Debian είναι ενημερωμένο)
- `sudo apt install ros-noetic-desktop-full` (Desktop-Full Install και περιέχει επιπλέον προσομοιωτές 2D και 3D και πακέτα αντίληψης 2D και 3D. Ένα από αυτά είναι και το πρόγραμμα rviz που χρησιμοποιώ κατά βάση στην υλοποίηση.)
- `apt search ros-noetic` (Για να δούμε αν εγκαταστάθηκε η έκδοση του ROS αλλά και για την αναζήτηση πακέτων ROS)
- `source /opt/ros/noetic/setup.bash` (Πρέπει να διοχετευθεί αυτό το σενάριο σε κάθε τερματικό bash στο οποίο χρησιμοποιείτε ROS. Έπειτα πρέπει να ξανατρέξει αυτή η εντολή εντός του περιβάλλοντος που θα φτιαχτεί) [20]

### 3.1.2 Δημιουργία περιβάλλοντος ROS Noetic

---

Για τη δημιουργία του περιβάλλοντος ROS θα πρέπει να τρέξουν πάλι ορισμένες γραμμές κώδικα στο terminal:

- `mkdir -p ~/catkin_ws/src` (Δημιουργία του φακέλου catkin\_ws και του υποφακέλου /src)
- `cd ~/catkin_ws/` (μεταφορά στο φάκελο)
- `catkin_make` (Εκτελώντας το για πρώτη φορά στον χώρο εργασίας σας, θα δημιουργήσει έναν σύνδεσμο CMakeLists.txt στον φάκελο 'src'.)

Επιπλέον, εάν κοιτάξετε στον τρέχοντα κατάλογο σας, θα πρέπει τώρα να έχετε έναν φάκελο «build» και «devel». Μέσα στο φάκελο 'devel' μπορείτε να δείτε ότι υπάρχουν τώρα πολλά αρχεία setup.\*sh. Η προμήθεια οποιουδήποτε από αυτά τα αρχεία θα επικαλύψει αυτόν τον χώρο εργασίας πάνω από το περιβάλλον σας.

- `source devel/setup.bash` (Για να βεβαιωθείτε ότι ο χώρος εργασίας σας επικαλύπτεται σωστά από το σενάριο εγκατάστασης)
- `echo $ROS_PACKAGE_PATH` (βεβαιωθείτε ότι η μεταβλητή περιβάλλοντος ROS\_PACKAGE\_PATH περιλαμβάνει τον κατάλογο στον οποίο βρίσκεστε.) [21]



## 3.2 Εγκατάσταση και επεξεργασία των υπολογιστικών εργαλείων

---

### 3.2.1 RosAria, (library) AriaCoda και rosaria\_client

---

Αρχικά για τον χειρισμό και την επικοινωνία του συστήματος μας με το ρομπότ θα χρειαστεί η εγκατάσταση του υπολογιστικού εργαλείου RosAria καθώς και μια βιβλιοθήκη (Aria). Αλλά όπως προανέφερα θα χρειαστεί να εγκαταστήσουμε την βιβλιοθήκη AriaCoda γιατί η άλλη δεν υποστηρίζεται πλέον. Οπότε ακολουθούν τα βήματα για την εγκατάσταση και οδηγίες:

RosAria:

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/amor-ros-pkg/rosaria.git` (Επειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd ..` (Μεταβαίνουμε μια θέση πίσω στο working space που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [23]

AriaCoda:

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/reedhedges/AriaCoda.git` (Επειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd ..` (Μεταβαίνουμε μια θέση πίσω στο working space που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [22]

rosaria\_client:

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/pengtang/rosaria\_client.git` (Επειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd ..` (Μεταβαίνουμε μια θέση πίσω στο working space που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [24]

Έπειτα για να τρέξει το RosAria θα πρέπει να γραφεί η εντολή:

- `roscore` (Για να σηκώσουμε τον Master)
- `roslaunch rosaria RosAria` (Για να τρέξει το RosAria γράφω την εντολή σε νέο terminal, όπου θα επιφέρει επικοινωνία με το ρομπότ)

Σημείωση: Θα πρέπει να γίνει έλεγχος της εισόδου USB, by default η τιμή είναι `/dev/ttyUSB0`. [23]

- `roslaunch rosaria_client interface` (Σε νέο terminal τρέχω την εντολή, για να δώσω τις κατάλληλες εντολές όπως φαίνεται στο terminal που βρίσκεται κάτω από τα άλλα δύο στην Εικόνα 14.1 παρακάτω).

Στην Εικόνα 14.1 υπάρχουν οι εξής επιλογές του interface

1. Πήγαινε αριστερά για 3 δευτερόλεπτα
2. Περιστρέψου όπως το ρολόι (γύρω από την θέση σου)
3. Περιστρέψου ανάποδα από τη φορά του ρολογιού (γύρω από την θέση σου)
4. Τηλεχειρισμός (Δίνεται δυνατότητα αυξομείωσης της γραμμικής και της γωνιακής ταχύτητας)

5. Εμφάνισε την θέση σου
6. Ενεργοποίηση κινητήρων
7. (q) για έξοδο

Σημείωση: Το `rosaria_client` χρησιμοποιήθηκε για τον τηλεχειρισμό (δηλαδή της εντολής 6) του ρομπότ ώστε να γίνει η χαρτογράφηση.[24]

The screenshot shows three terminal windows. The top-left window shows the successful launch of `roslaunch` for the `gcharellis-Strix-GL504GM-GL504GM:11311/` package. The top-right window shows several error messages related to parsing arguments for the `ArConfig` and `ArRobotParams` packages, indicating invalid MTX sonar board IDs. The bottom window shows the `rosaria_client` interface, which displays the current state of the robot (e.g., battery voltage, bumper state) and provides a menu of commands to interact with the robot.

Εικόνα 14.1, Στο screenshot παρουσιάζονται τα terminals με τις εντολές που έχω τρέξει πάνω αριστερά το `roscore` δίπλα το `RosArria` και από κάτω το interface

### 3.2.3 SICK lms500-20000

Πακέτα που περιέχουν το URDF του Laser (Αλλά και πολλών άλλων αισθητήρων της εταιρίας SICK) καθώς και τα `.launch` αρχεία για τη λειτουργία τους πάνω στα ROS και την ενσωμάτωση των μετρήσεων τους από διάφορους αλγορίθμους όπως αυτών των αλγορίθμων που χρησιμοποιούνται σε αυτή την πτυχιακή εργασία. Ο τρόπος εγκατάστασης των πακέτων αναφέρεται παρακάτω και έπειτα και οι αλλαγές που έγιναν για την σωστή επικοινωνία και ενσωμάτωση του αισθητήρα που χρησιμοποιώ.

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/SICKAG/sick_scan.git` (Έπειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd ..` (Μεταβαίνουμε μια θέση πίσω στο working space που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [26]

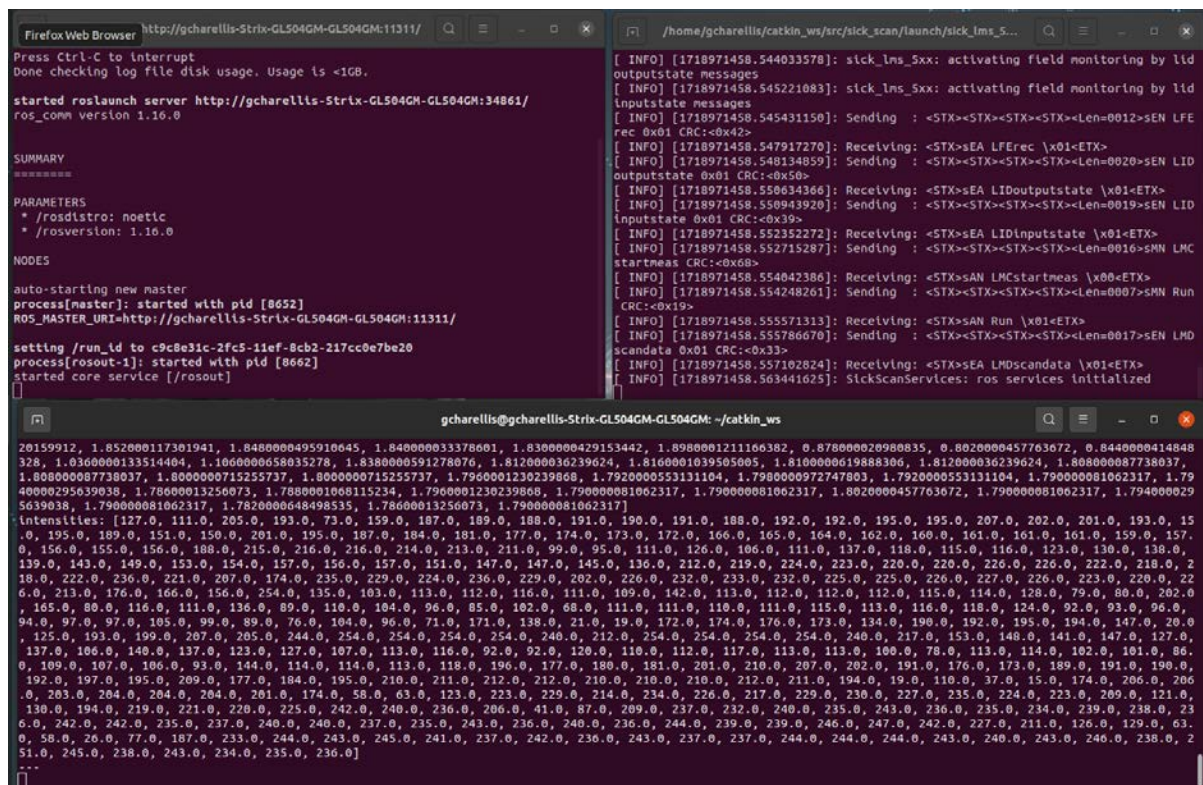
Αξίζει να τονιστεί ότι θα πρέπει να αλλαχτεί η IP που έχει by default σε 169.254.110.78 και να γίνει ο απαραίτητος στατικός μετασχηματισμός από το `base_link` στο `laser` με τιμές 0.2 0 0.41 0 0 0 συνεπάγονται με την πραγματικότητα και να αλλάξουμε αν επιθυμούμε το όνομα του topic όπου θα

γίνονται διοχετεύσεις ώστε να λαμβάνουμε τις μετρήσεις σε αυτό. Όπως φαίνεται στο υποκεφάλαιο 4.4 στο αρχείο `sick_lms_5xx.launch`

Για να λάβουμε μετρήσεις θα πρέπει να συνδέσουμε την θύρα ethernet και να φτιάξουμε μια νέα σύνδεση με IP 169.254.110.78 και SUBMASK 255.55.0.0 έπειτα στο νέο τερματικό να γραφεί:

- `roslaunch sick_scan sick_lms_5xx.launch` (Το `launch` αρχείο του αισθητήρα)
- Καλό θα είναι για τη δοκιμή να τρέξουμε σε άλλο terminal την εντολή `roscore`

Σημείωση: Για τον έλεγχο αν δίνονται τιμές στο `topic` θα πρέπει σε νέο terminal να τρέξουμε την εντολή `rostopic echo` και το όνομα του `topic` αν είναι σωστό θα πρέπει να έχουμε κάτι σαν αυτό που φαίνεται παρακάτω στην Εικόνα 15.



The screenshot shows a terminal window with two panes. The top pane displays the output of the `roslaunch sick_scan sick_lms_5xx.launch` command, showing the launch of the `sick_scan` package and the initialization of the `sick_lms_5xx` node. The bottom pane shows the output of the `rostopic echo` command, displaying a large list of laser range-finding data points in a structured format.

Εικόνα 15, Πάνω αριστερά είναι το terminal με το `roscore` και πάνω δεξιά είναι το `.launch` αρχείο για την εκκίνηση του αισθητήρα και κάτω οι μετρήσεις του laser.

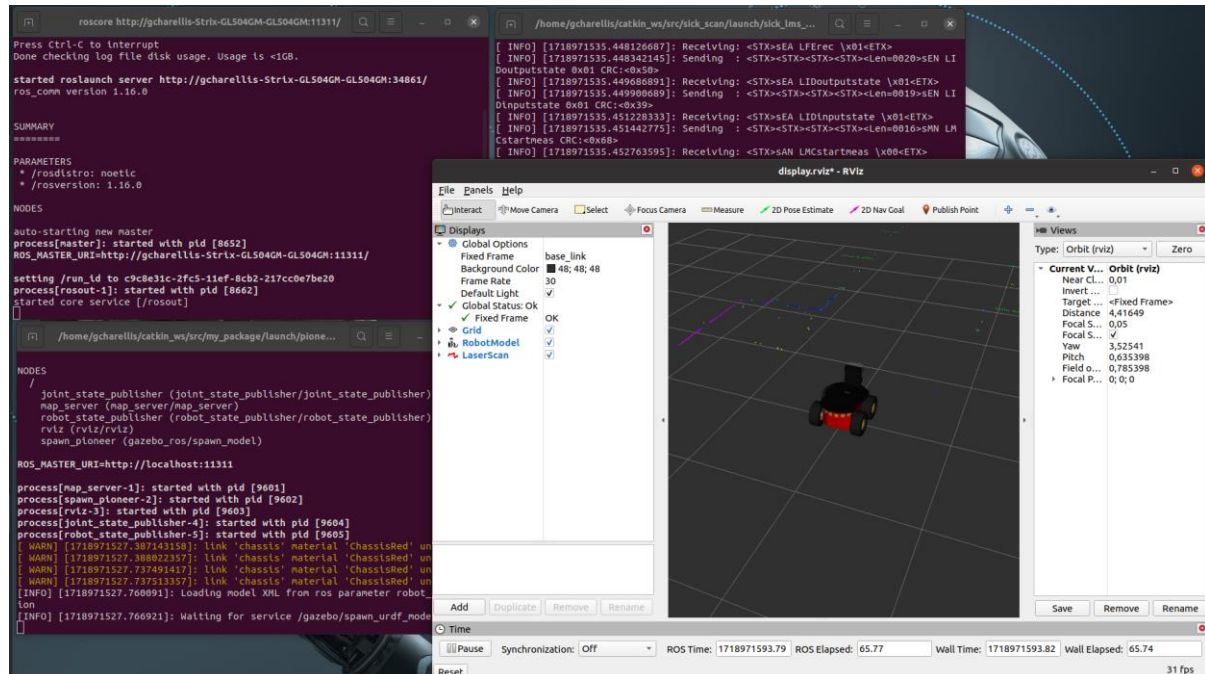
### 3.2.2 Πακέτα Pioneer 3-AT

Πακέτα που περιέχουν το μοντέλο του ρομπότ σε σασρο μορφή αλλά και τα πακέτα για το `amcl` και το `move base` του ρομπότ Pioneer 3-AT :

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/Gastd/p3at_tutorial.git` (Έπειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd..` (Μεταβαίνουμε μια θέση πίσω στο working space που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [25]



Με βάση αυτό, δημιούργησα το δικό μου πακέτο για να σηκώνει το μοντέλο του ρομπότ και τον χάρτη όπως φαίνεται στο αρχείο κώδικα `pioneer3at.launch` της υποενοτήτας 4.2.1. Δημιούργησα ένα νέο αρχείο `.launch` για το χάρτη που δημιουργώ στην συνέχεια της υποενοτήτας 3.2.3. Επίσης έγιναν κάποιες αλλαγές στο αρχείο `pioneer3at_body.xacro` ώστε να γίνει η ενσωμάτωση του μοντέλου αισθητήρα. Οι αλλαγές παρουσιάζονται στο υποκεφάλαιο 4.1. Αλλά παρουσιάζονται και σε όλο το υποκεφάλαιο 4.2 και τα αρχεία κώδικα από τα οποία εξαρτάται το `.launch` αρχείο `pioneer3at.launch`.



Εικόνα 16, Η εικόνα του ρομπότ μετά την ενσωμάτωση του μοντέλου και με τις μετρήσεις του Laser

Η υλοποίηση του παραδείγματος της Εικόνας 16:

- `roscore` (Για να σηκώσουμε τον Master)
  - `roslaunch rosaria RosAria` (Για να τρέξει το RosAria γράφω την εντολή σε νέο terminal, όπου θα επιφέρει επικοινωνία με το ρομπότ)
- Σημείωση: Θα πρέπει να γίνει έλεγχος της εισόδου USB, by default η τιμή είναι `/dev/ttyUSB0`. [23]
- `roslaunch rosaria_client interface` (Σε νέο terminal τρέχω την εντολή, για να δώσω τις κατάλληλες εντολές όπως φαίνεται στο terminal που βρίσκεται κάτω από τα άλλα δύο στην Εικόνα 14.1 παραπάνω).
  - `roslaunch sick_scan sick_lms_5xx.launch` (Το launch αρχείο του αισθητήρα)
  - `roslaunch my_package pioneer3at.launch` (Το launch αρχείο για το ρομπότ που σηκώνει το μοντέλο του)
  - `roslaunch rviz rviz` (τρέχουμε την εντολή αυτή για να ανοίξει το rviz και προσθέτουμε ότι θέλουμε από ορίσματα και topics)
  -

### 3.2.3 Πακέτα Slam-Gmapping

Πακέτα που κατά την εκτέλεση τους, δημιουργούν έναν νέο χάρτη για την εγκατάσταση θα πρέπει να γίνουν τα παρακάτω βήματα:

- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/ros-perception/slam_gmapping.git` (Επειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)

- `cd..` (Μεταβαίνουμε μια θέση πίσω στο `working space` που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε) [27]

Για την εκτέλεση του `slam-gmapping` θα πρέπει παράλληλα να τρέξουμε τα αρχεία για το `RosAria`, `rosaria_client`, τον αισθητήρα και το `slam-gmapping` και έπειτα να αποθηκεύσουμε τον χάρτη. Ακολουθούν τα βήματα:

- `roscore` (Για να σηκώσουμε τον Master)
- `roslaunch rosaria RosAria` (Για να τρέξει το `RosAria` γράφω την εντολή σε νέο terminal, όπου θα επιφέρει επικοινωνία με το ρομπότ)
- `roslaunch rosaria_client interface` (Σε νέο terminal τρέχω την εντολή, για να δώσω τις κατάλληλες εντολές όπως φαίνεται στο terminal που βρίσκεται κάτω από τα άλλα δύο στην Εικόνα 14.1 παραπάνω).
- `roslaunch sick_scan sick_lms_5xx.launch` (To launch αρχείο του αισθητήρα)
- `roslaunch my_package pioneer3at.launch` (To launch αρχείο για το ρομπότ που σηκώνει το μοντέλο του)
- `roslaunch rviz rviz` (τρέχουμε την εντολή αυτή για να ανοίξει το `rviz` και προσθέτουμε ότι θέλουμε από ορίσματα και topics)
- `roslaunch slam_gmapping slam_gmapping_pr2.launch` (To launch αρχείο για την χαρτογράφηση)
- Κάνουμε ενεργοποίηση των κινητήρων (εντολή 6 στο `rosaria client`) και έπειτα επιλέγουμε τον τηλεχειρισμό (εντολή 4 στο `rosaria client`)
- Έπειτα κάνουμε μια βόλτα το ρομπότ σε όλο τον χώρο για να γίνει η χαρτογράφηση
- `roslaunch map_server map_server -f <name of map>` (Για την αποθήκευση του χάρτη τρέχουμε αυτή την εντολή)

### 3.3 Ολοκλήρωση υλοποίησης με τη χρήση της AMCL

Για την εγκατάσταση της `AMCL` και της `Move_base` θα ακολουθήσουμε ορισμένα βήματα:

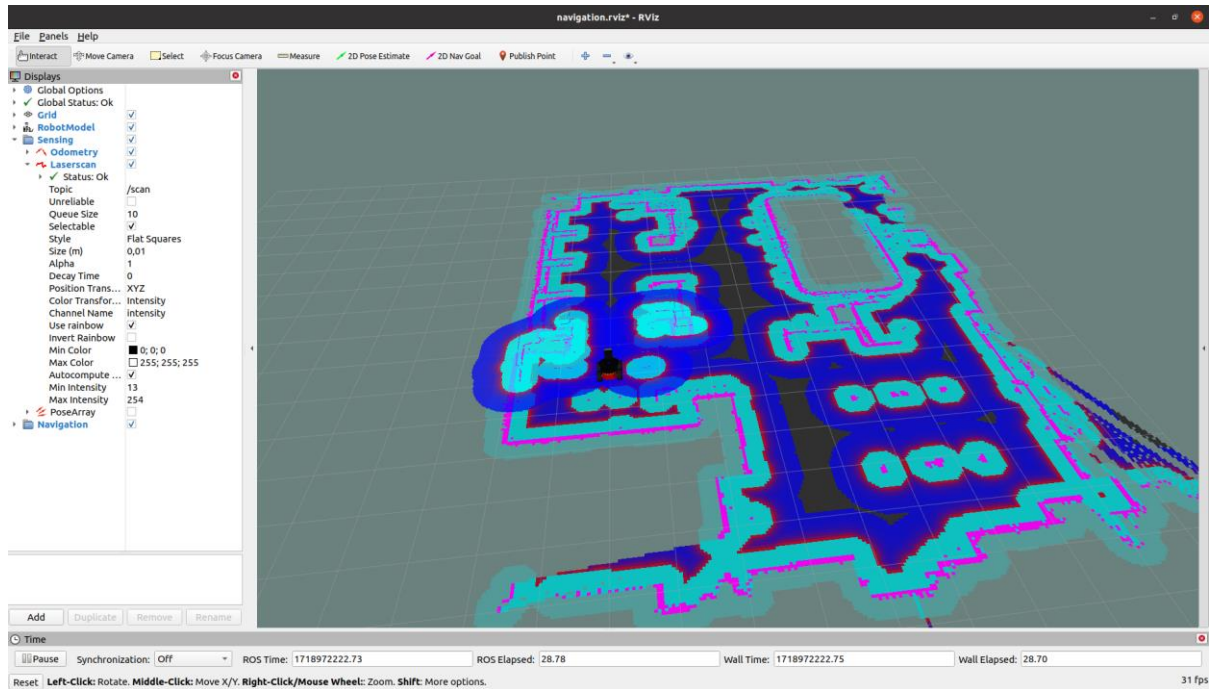
- `cd ~/catkin_ws/src` (Μεταβαίνουμε στον φάκελο για την εγκατάσταση)
- `git clone https://github.com/strawlab/navigation.git` (Έπειτα γίνεται εγκατάσταση των υπολογιστικών από την διεύθυνση του Github)
- `cd..` (Μεταβαίνουμε μια θέση πίσω στο `working space` που έχουμε φτιάξει)
- `source devel/setup.bash` (Κάνουμε διευθυνσιοδότηση της πηγής)
- `catkin_make` (Κάνουμε build τα πακέτα που εγκαταστήσαμε)

Επεξεργασία των αρχείων της `AMCL`, της `Move_base` και των `config` αρχείων του κώδικα καθώς και μεταφορά στο φάκελο αρχείων `my_package`. Επίσης χρειάστηκαν μικρο-αλλαγές στον κώδικα όπως οι αλλαγές των διευθύνσεων και οι ονομασίες των topics που χρησιμοποιούνται από τον κώδικα και από τα άλλα αρχεία ώστε να υπάρχει σωστή επικοινωνία και ορθή διοχέτευση τιμών για την αποτελεσματικότητα του αλγορίθμου και ως επί των πλείστων η ορθή υλοποίηση της εργασίας. Παρουσιάζονται στα υποκεφάλαια 4.5 και 4.6 για τα αρχεία `AMCL`, `Move_base` και τα `config` αρχεία αντίστοιχα. Τα `config` αρχεία αποτελούν την κύρια πηγή του αλγορίθμου και για την υπολογισμό της θέσης των στοιχείων (μετρήσεις απόστασης θέσης και ορίων ή εμποδίων) σχετικά με την θέση του στο χάρτη αλλά και του χάρτη με τα εμπόδια ή τα όρια του χάρτη και τη δυναμική αναγνώριση εμποδίων.

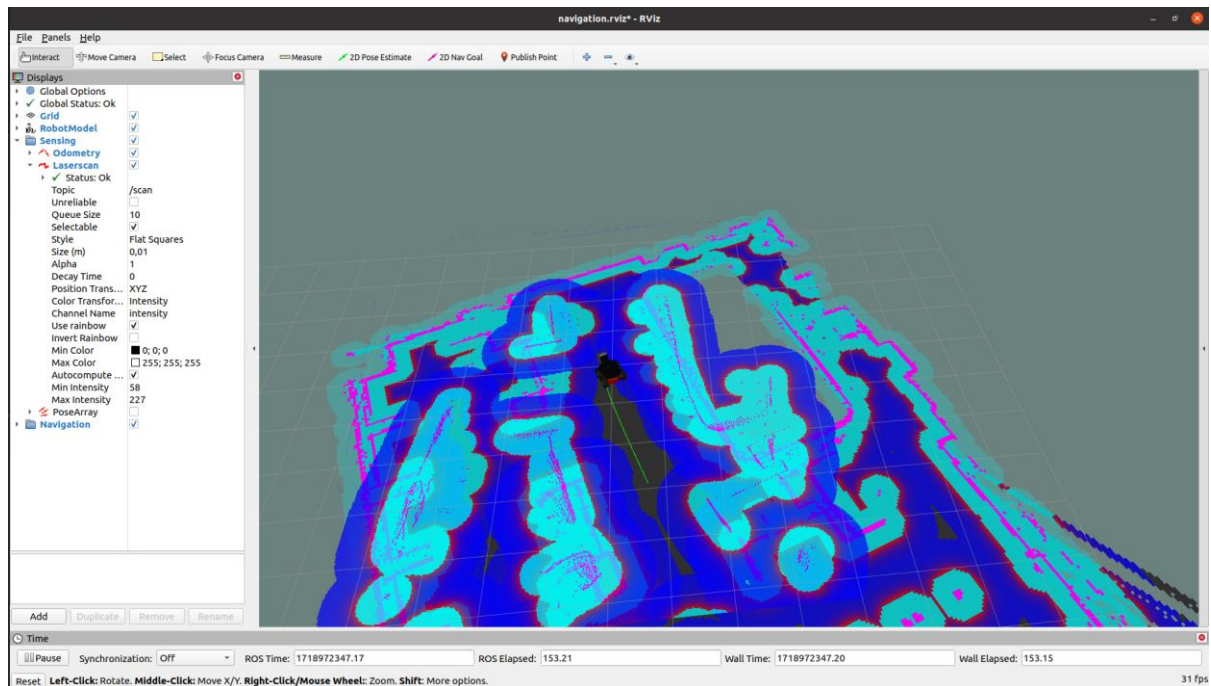
Έπειτα για την εκτέλεση της εργασίας θα ακολουθήσουμε τα παρακάτω βήματα:

- `roscore` (Για να σηκώσουμε τον Master)

- `roslaunch rosaria RosAria` (Για να τρέξει το RosAria γράφω την εντολή σε νέο terminal, όπου θα επιφέρει επικοινωνία με το ρομπότ)
- `roslaunch sick_scan sick_lms_5xx.launch` (To launch αρχείο του αισθητήρα)
- `roslaunch my_package pioneer3at.launch` (To launch αρχείο για το ρομπότ που σηκώνει το μοντέλο του)
- `roslaunch my_package amcl_demo.launch` (To .launch αρχείο του για την υλοποίηση της AMCL και της Move\_base )



Εικόνα 17, Έτσι ξεκινάει η υλοποίηση αν ακολουθήσουμε τα παραπάνω βήματα (τελικό στάδιο εργασίας)



Εικόνα 18, Εκτέλεση κίνησης με την χρήση της AMCL και της Move\_base

Πατώντας στο ροζ βελάκι 2D Nav Goal επιλέγουμε την δυνατότητα επιλογής θέσης και έπειτα επιλέγουμε το σημείο στο χάρτη (εντός του χάρτη αλλιώς δε θα γίνει κάποια ενέργεια) ώστε να του

δείξουμε που πρέπει να πάει το ρομποτικό μας μοντέλο. Να σημειωθεί ότι κατά την επιλογή του σημείου στο χάρτη θα πρέπει να κρατήσουμε παρατεταμένα για να επιλέξουμε την κατεύθυνση που θέλουμε να έχει (κουνώντας το ποντίκι). Επιπλέον, έχουμε τη δυνατότητα να το ξεκινήσουμε από όποιο σημείο θέλουμε επιλέγοντας τη 2D Pose Estimate και ακολουθώντας τις παραπάνω οδηγίες.

## ΚΕΦΑΛΑΙΟ 4 ΚΩΔΙΚΑΣ

### 4.1 Ο κώδικας με προσθήκη του xacro αρχείου του ρομπότ

pioneer3at\_body.xacro

```
414 <!-- sick_lms_5xx Laser -->
415 <joint name="sick_lms_5xx" type="fixed">
416   <axis xyz="0 1 0" />
417   <origin xyz="0.2 0 0.41" rpy="0 0 0"/>
418   <parent link="base_link"/>
419   <child link="sick_lms_5xx"/>
420 </joint>
421 <link name="sick_lms_5xx">
422   <collision>
423     <origin xyz="0 0 0" rpy="0 0 0"/>
424     <geometry>
425       <box size="0.01 0.01 0.01"/>
426     </geometry>
427   </collision>
428   <visual>
429     <origin xyz="0 0 0" rpy="0 0 0"/>
430     <geometry>
431       <mesh filename="package://my_package/meshes/LMS5xx_small.dae"/>
432     </geometry>
433   </visual>
434   <inertial>
435     <mass value="1e-5" />
436     <origin xyz="0 0 0" rpy="0 0 0"/>
437     <inertia ixx="1e-6" ixy="0" ixz="0" iyy="1e-6" iyz="0" izz="1e-6" />
438   </inertial>
439 </link>

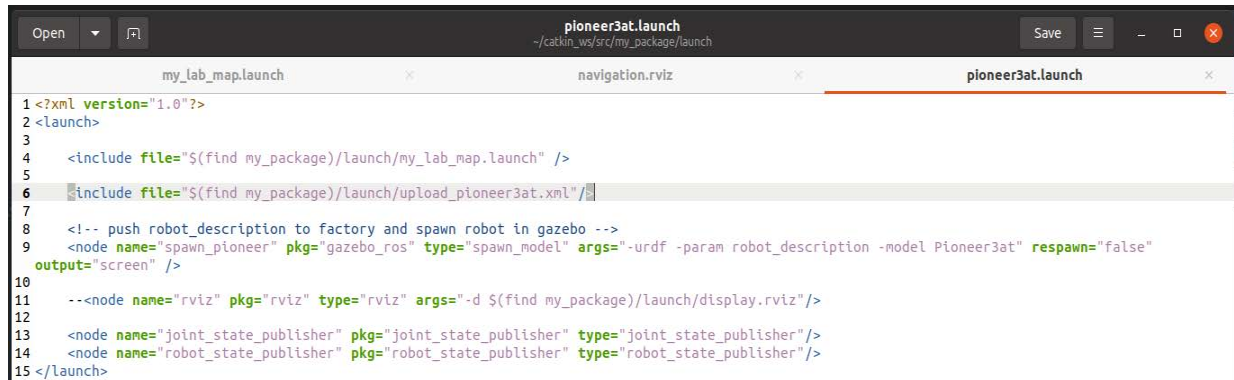
440 <gazebo reference="sick_lms_5xx">
441   <sensor type="ray" name="sick_lms_5xx">
442     <pose>0 0 0 0 0 0</pose>
443     <visualize>true</visualize>
444     <update_rate>40</update_rate>
445     <ray>
446       <scan>
447         <horizontal>
448           <samples>1080</samples>
449           <resolution>1</resolution>
450           <min_angle>-2.3559777</min_angle>
451           <max_angle>2.3559777</max_angle>
452         </horizontal>
453       </scan>
454       <range>
455         <min>0.10</min>
456         <max>30.0</max>
457         <resolution>0.01</resolution>
458       </range>
459       <noise>
460         <type>gaussian</type>
461         <!-- Noise parameters based on published spec for Hokuyo laser
462              achieving "+-30mm" accuracy at range < 10m. A mean of 0.0m and
463              stddev of 0.01m will put 99.7% of samples within 0.03m of the true
464              reading. -->
465         <mean>0.0</mean>
466         <stddev>0.01</stddev>
467       </noise>
468     </ray>
469     <plugin name="sick_lms_5xx" filename="libgazebo_ros_laser.so">
470       <topicName>/scan</topicName>
471       <frameName>sick_lms_5xx</frameName>
472     </plugin>
473   </sensor>
474 </gazebo>
475
```



## 4.2 Αρχεία κώδικα για το ρομπότ

### 4.2.1 Τα launch αρχεία του ρομπότ για την εμφάνιση του στο rviz

pioneer3at.launch



```
1 <?xml version="1.0"?>
2 <launch>
3
4   <include file="$(find my_package)/launch/my_lab_map.launch" />
5
6   <include file="$(find my_package)/launch/upload_pioneer3at.xml" />
7
8   <!-- push robot_description to factory and spawn robot in gazebo -->
9   <node name="spawn_pioneer" pkg="gazebo_ros" type="spawn_model" args="-urdf -param robot_description -model Pioneer3at" respawn="false"
10  output="screen" />
11
12   <!-- node name="rviz" pkg="rviz" type="rviz" args="-d $(find my_package)/launch/display.rviz" />
13
14   <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
15   <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
16 </launch>
```

upload\_pioneer3at.xml



```
1 <?xml version="1.0"?>
2 <launch>
3   <!-- send table.xml to param server -->
4   <param name="robot_description" command="$(find xacro)/xacro $(find my_package)/robots/p3at.xacro" />
5
6 </launch>
7
```

p3at.xacro



```
1 <?xml version="1.0"?>
2
3 <robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="p3at_tutorial">
4
5   <xacro:include filename="$(find my_package)/robots/pioneer3at_body.xacro" />
6
7   <!--<xacro:include filename="$(find pr2_description)/urdf/common.xacro" />--
8   <xacro:p3at_front suffix="left" reflect="1"/>
9   <xacro:p3at_front suffix="right" reflect="-1"/>
10  <xacro:p3at_back suffix="left" reflect="1"/>
11  <xacro:p3at_back suffix="right" reflect="-1"/>
12
13  <xacro:p3at_gazebo_ext/>
14
15
16 </robot>
```

## 4.3 Το Launch αρχείο για την εκτέλεση της χαρτογράφησης

Slam\_gmapping\_pr2.launch

```
Open  slam_gmapping_pr2.launch
~/catkin_ws/src/slam_gmapping/gmapping/launch

1 <?xml version="1.0"?>
2 <launch>
3   <param name="use_sim_time" value="true"/>
4   <node pkg="gmapping" type="slam_gmapping" name="slam_gmapping" output="screen">
5     <remap from="cloud" to="base_scan"/>
6     <param name="map_update_interval" value="5.0"/>
7     <param name="maxUrange" value="16.0"/>
8     <param name="sigma" value="0.05"/>
9     <param name="kernelSize" value="1"/>
10    <param name="lstep" value="0.05"/>
11    <param name="astep" value="0.05"/>
12    <param name="iterations" value="5"/>
13    <param name="lsigma" value="0.075"/>
14    <param name="ogain" value="3.0"/>
15    <param name="lskip" value="0"/>
16    <param name="srr" value="0.1"/>
17    <param name="srt" value="0.2"/>
18    <param name="str" value="0.1"/>
19    <param name="stt" value="0.2"/>
20    <param name="linearUpdate" value="1.0"/>
21    <param name="angularUpdate" value="0.5"/>
22    <param name="temporalUpdate" value="3.0"/>
23    <param name="resampleThreshold" value="0.5"/>
24    <param name="particles" value="30"/>
25    <param name="xmin" value="-50.0"/>
26    <param name="ymin" value="-50.0"/>
27    <param name="xmax" value="50.0"/>
28    <param name="ymax" value="50.0"/>
29    <param name="delta" value="0.05"/>
30    <param name="llsamplerange" value="0.01"/>
31    <param name="llsamplerstep" value="0.01"/>
32    <param name="lasamplerange" value="0.005"/>
33    <param name="lasamplerstep" value="0.005"/>
34  </node>
35 </launch>
```

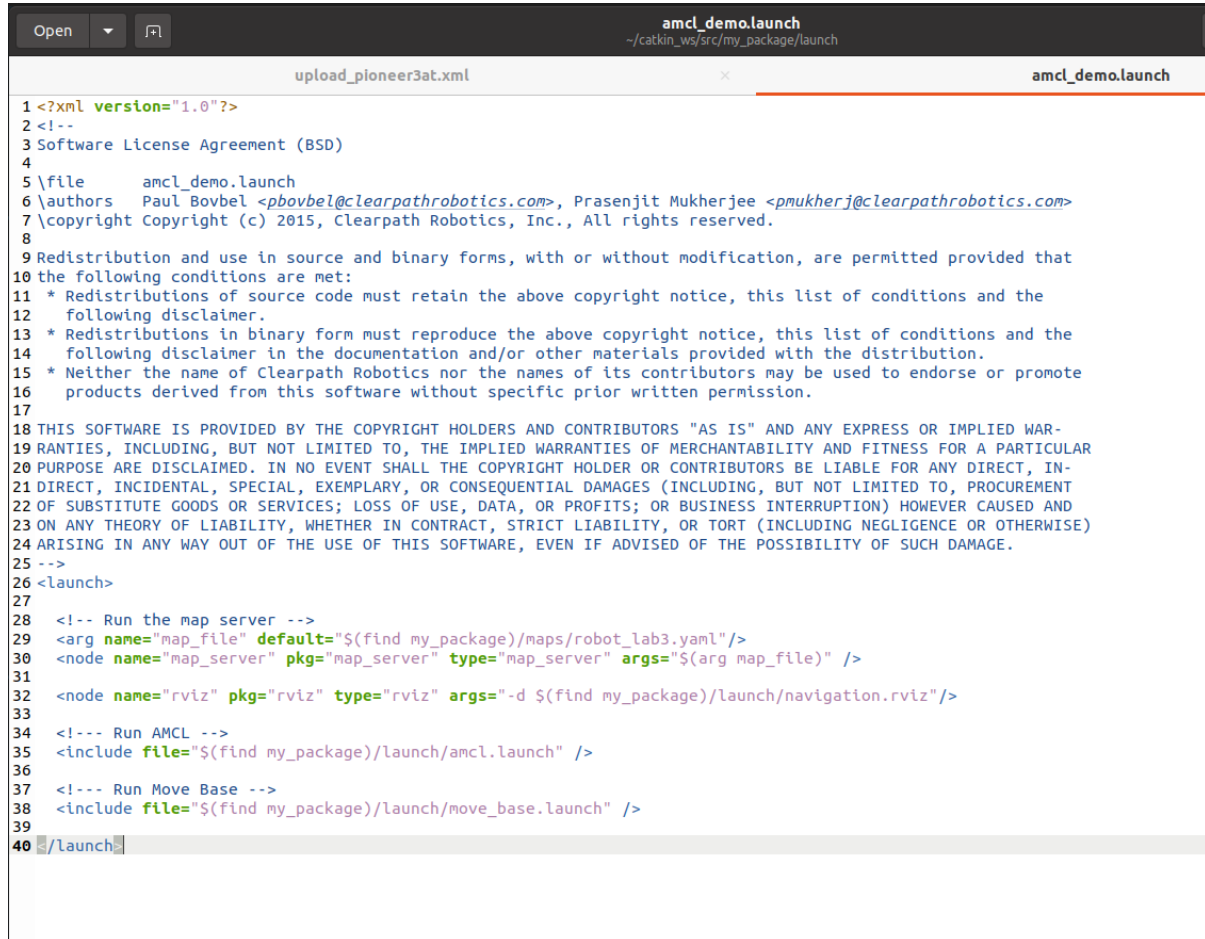
## 4.4 Το περιεχόμενο του Launch αρχείου για τον αισθητήρα SICK lms500-20000

sick\_lms\_5xx.launch

```
Open  *sick_lms_5xx.launch
~/catkin_ws/src/sick_scan/launch
0 <!--
1 <!-- start and stop angle is given in [rad] -->
2 <!--
3 Check IP-address, if you scanner is not found after roslaunch.
4 -->
5 <launch>
6 <!--<arg name="hostname" default="192.168.0.1"/>-->
7 <arg name="hostname" default="169.254.110.78"/>
8 <node pkg="tf2_ros" type="static_transform_publisher" name="base_link_to_scan" args="0.2 0 0.41 0 0 0 base_link scan" />
9 <!-- launch file for the lms_5xx series -->
10 <node name="sick_lms_5xx" pkg="sick_scan" type="sick_generic_caller" respawn="false" output="screen">
11 <!--<param name="robot_description" command="$(find xacro)/xacro.py '$(find sick_scan)/urdf/example.urdf.xacro'" />-->
12
13 <!-- default values: -->
14 <!--
15 <param name="min_ang" type="double" value="-1.658062789" />
16 <param name="max_ang" type="double" value="1.658062789" />
17 <param name="intensity" type="bool" value="True" />
18 <param name="skip" type="int" value="0" />
19 <param name="frame_id" type="str" value="laser" />
20 <param name="time_offset" type="double" value="-0.001" />
21 <param name="publish_datagram" type="bool" value="False" />
22 <param name="subscribe_datagram" type="bool" value="false" />
23 <param name="device_number" type="int" value="0" />
24 <param name="range_min" type="double" value="0.05" />
25 -->
26 <param name="frame_id" type="str" value="sick_lms_5xx" />
27 <param name="scanner_type" type="string" value="sick_lms_5xx" />
28 <!-- -95° -->
29 <param name="min_ang" type="double" value="-1.658062789" />
30 <!-- 95° -->
31 <param name="max_ang" type="double" value="1.658062789" />
32 <param name="range_max" type="double" value="100.0" />
33 <param name="intensity" type="bool" value="true" />
34 <param name="intensity_resolution_16bit" type="bool" value="false" />
35 <param name="hostname" type="string" value="$(arg hostname)" />
36 <param name="port" type="string" value="2112" />
37 <param name="timelimit" type="int" value="5" />
38 <param name="use_binary_protocol" type="bool" value="true" />
39 <!-- possible angular resolutions are 1,0.6667,0.5,0.3333,0.25,0.1667
40 possible scan frequencies are 25,35,50,75,100 Hz
41 Make sure to check in the manual whether the desired scanning frequency
42 and angular resolution combination is available.
43 The parameters scan_freq and ang_res must always be used together.
44 <param name="ang_res" type="double" value="0.1667" />
45 <param name="scan_freq" type="double" value="25" />
46 -->
47 <param name="start_services" type="bool" value="True" /> <!-- start ros service for cola commands -->
48 <param name="activate_lferec" type="bool" value="True" /> <!-- activate field monitoring by lferec messages -->
49 <param name="activate_lidoutputstate" type="bool" value="True" /> <!-- activate field monitoring by lidoutputstate messages -->
50 <param name="activate_lidinputstate" type="bool" value="True" /> <!-- activate field monitoring by lidinputstate messages -->
51 <param name="min_intensity" type="double" value="0.0" /> <!-- Set range of LaserScan messages to infinity, if intensity < min_intensity (default: 0) -->
52
53 </node>
54 </launch>
```

## 4.5 Το περιεχόμενο του launch αρχείου amcl\_demo το οποίο στη συνέχεια σηκώνει το amcl.launch και το move\_base

amcl\_demo.launch:

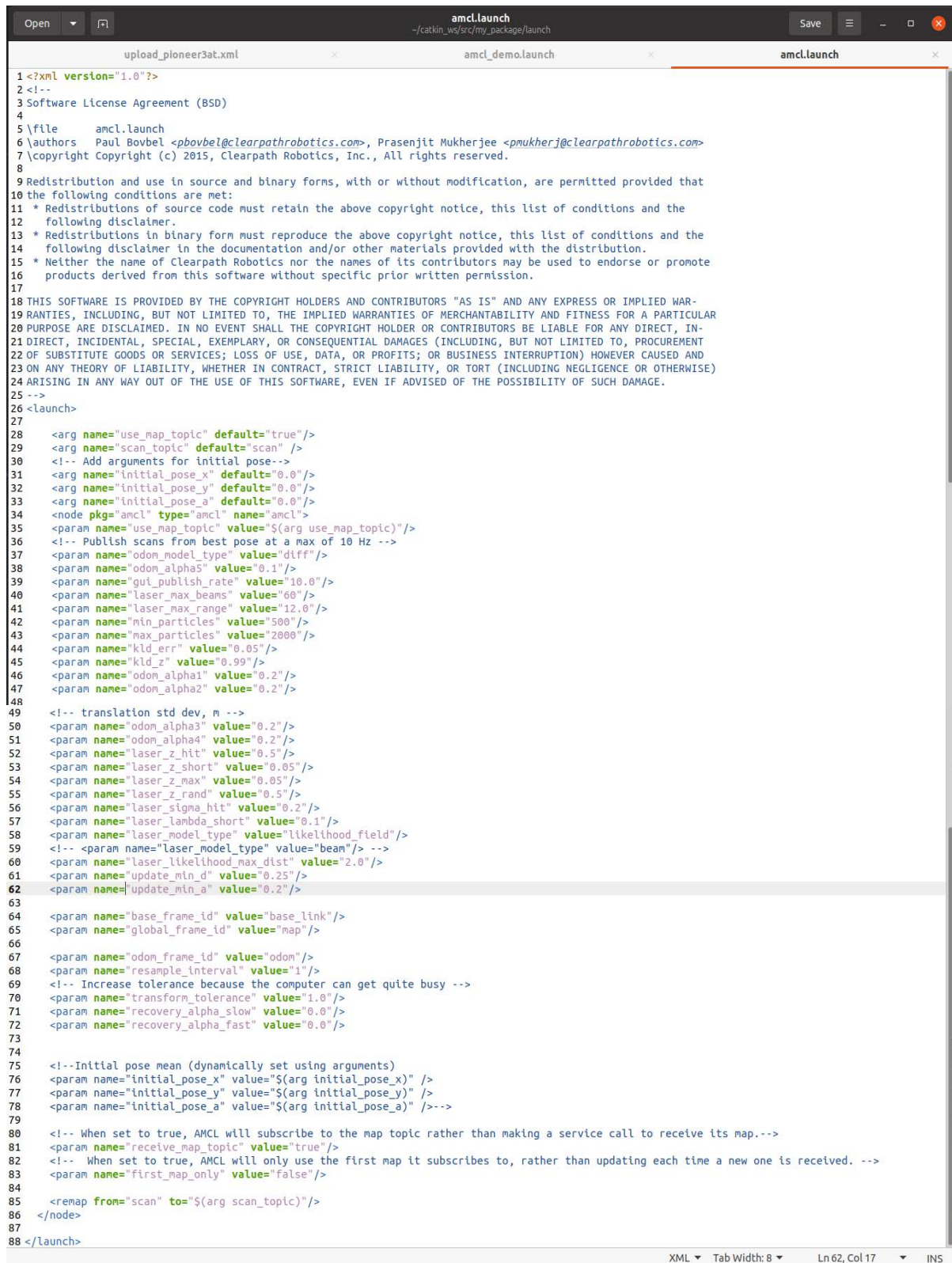


```
1 <?xml version="1.0"?>
2 <!--
3 Software License Agreement (BSD)
4
5 \file      amcl_demo.launch
6 \authors   Paul Bovbel <pbovbel@clearpathrobotics.com>, Prasenjit Mukherjee <pmukherj@clearpathrobotics.com>
7 \copyright Copyright (c) 2015, Clearpath Robotics, Inc., All rights reserved.
8
9 Redistribution and use in source and binary forms, with or without modification, are permitted provided that
10 the following conditions are met:
11 * Redistributions of source code must retain the above copyright notice, this list of conditions and the
12   following disclaimer.
13 * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
14   following disclaimer in the documentation and/or other materials provided with the distribution.
15 * Neither the name of Clearpath Robotics nor the names of its contributors may be used to endorse or promote
16   products derived from this software without specific prior written permission.
17
18 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WAR-
19 RANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
20 PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, IN-
21 DIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
22 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
23 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
24 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
25 -->
26 <launch>
27
28 <!-- Run the map server -->
29 <arg name="map_file" default="$(find my_package)/maps/robot_lab3.yaml"/>
30 <node name="map_server" pkg="map_server" type="map_server" args="$(arg map_file)" />
31
32 <node name="rviz" pkg="rviz" type="rviz" args="-d $(find my_package)/launch/navigation.rviz"/>
33
34 <!-- Run AMCL -->
35 <include file="$(find my_package)/launch/amcl.launch" />
36
37 <!-- Run Move Base -->
38 <include file="$(find my_package)/launch/move_base.launch" />
39
40 </launch>
```



## 4.5.1 Το αρχείο amcl

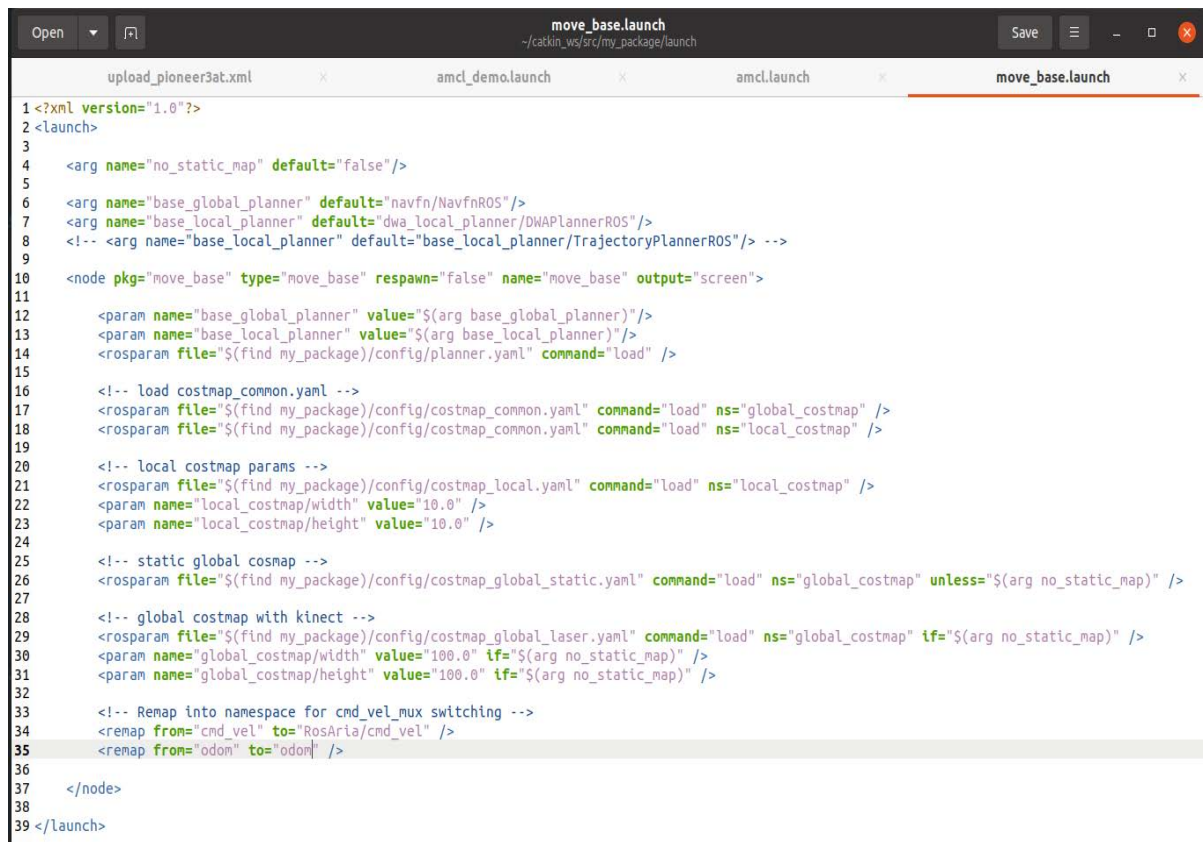
amcl.launch



```
1 <?xml version="1.0"?>
2 <!--
3 Software License Agreement (BSD)
4
5 \file      amcl.launch
6 \authors   Paul Bovbel <pbovbel@clearpathrobotics.com>, Prasenjit Mukherjee <pmukherj@clearpathrobotics.com>
7 \copyright Copyright (c) 2015, Clearpath Robotics, Inc., All rights reserved.
8
9 Redistribution and use in source and binary forms, with or without modification, are permitted provided that
10 the following conditions are met:
11 * Redistributions of source code must retain the above copyright notice, this list of conditions and the
12   following disclaimer.
13 * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
14   following disclaimer in the documentation and/or other materials provided with the distribution.
15 * Neither the name of Clearpath Robotics nor the names of its contributors may be used to endorse or promote
16   products derived from this software without specific prior written permission.
17
18 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WAR-
19 RANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
20 PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, IN-
21 DIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
22 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
23 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
24 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
25 -->
26 <launch>
27
28   <arg name="use_map_topic" default="true"/>
29   <arg name="scan_topic" default="scan" />
30   <!-- Add arguments for initial pose -->
31   <arg name="initial_pose_x" default="0.0"/>
32   <arg name="initial_pose_y" default="0.0"/>
33   <arg name="initial_pose_a" default="0.0"/>
34   <node pkg="amcl" type="amcl" name="amcl">
35     <param name="use_map_topic" value="$(arg use_map_topic)"/>
36     <!-- Publish scans from best pose at a max of 10 Hz -->
37     <param name="odom_model_type" value="dfft"/>
38     <param name="odom_alpha5" value="0.1"/>
39     <param name="gui_publish_rate" value="10.0"/>
40     <param name="laser_max_beams" value="60"/>
41     <param name="laser_max_range" value="12.0"/>
42     <param name="min_particles" value="500"/>
43     <param name="max_particles" value="2000"/>
44     <param name="kld_err" value="0.05"/>
45     <param name="kld_z" value="0.99"/>
46     <param name="odom_alpha1" value="0.2"/>
47     <param name="odom_alpha2" value="0.2"/>
48
49     <!-- translation std dev, m -->
50     <param name="odom_alpha3" value="0.2"/>
51     <param name="odom_alpha4" value="0.2"/>
52     <param name="laser_z_hit" value="0.5"/>
53     <param name="laser_z_short" value="0.05"/>
54     <param name="laser_z_max" value="0.05"/>
55     <param name="laser_z_rand" value="0.5"/>
56     <param name="laser_sigma_hit" value="0.2"/>
57     <param name="laser_lambda_short" value="0.1"/>
58     <param name="laser_model_type" value="likelihood_field"/>
59     <!-- <param name="laser_model_type" value="beam"/> -->
60     <param name="laser_likelihood_max_dist" value="2.0"/>
61     <param name="update_min_d" value="0.25"/>
62     <param name="update_min_a" value="0.2"/>
63
64     <param name="base_frame_id" value="base_link"/>
65     <param name="global_frame_id" value="map"/>
66
67     <param name="odom_frame_id" value="odom"/>
68     <param name="resample_interval" value="1"/>
69     <!-- Increase tolerance because the computer can get quite busy -->
70     <param name="transform_tolerance" value="1.0"/>
71     <param name="recovery_alpha_slow" value="0.0"/>
72     <param name="recovery_alpha_fast" value="0.0"/>
73
74     <!-- Initial pose mean (dynamically set using arguments) -->
75     <param name="initial_pose_x" value="$(arg initial_pose_x)"/>
76     <param name="initial_pose_y" value="$(arg initial_pose_y)"/>
77     <param name="initial_pose_a" value="$(arg initial_pose_a)"/>
78
79     <!-- When set to true, AMCL will subscribe to the map topic rather than making a service call to receive its map. -->
80     <param name="receive_map_topic" value="true"/>
81     <!-- When set to true, AMCL will only use the first map it subscribes to, rather than updating each time a new one is received. -->
82     <param name="first_map_only" value="false"/>
83
84     <remap from="scan" to="$(arg scan_topic)"/>
85   </node>
86 </launch>
```

## 4.5.2 Το αρχείο move base

move\_base.launch



```
1 <?xml version="1.0"?>
2 <launch>
3
4   <arg name="no_static_map" default="false"/>
5
6   <arg name="base_global_planner" default="navfn/NavfnROS"/>
7   <arg name="base_local_planner" default="dwa_local_planner/DWAPlanerROS"/>
8   <!-- <arg name="base_local_planner" default="base_local_planner/TrajectoryPlannerROS"/> -->
9
10  <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
11
12    <param name="base_global_planner" value="$(arg base_global_planner)"/>
13    <param name="base_local_planner" value="$(arg base_local_planner)"/>
14    <rosparam file="$(find my_package)/config/planner.yaml" command="load" />
15
16    <!-- load costmap_common.yaml -->
17    <rosparam file="$(find my_package)/config/costmap_common.yaml" command="load" ns="global_costmap" />
18    <rosparam file="$(find my_package)/config/costmap_common.yaml" command="load" ns="local_costmap" />
19
20    <!-- local costmap params -->
21    <rosparam file="$(find my_package)/config/costmap_local.yaml" command="load" ns="local_costmap" />
22    <param name="local_costmap/width" value="10.0" />
23    <param name="local_costmap/height" value="10.0" />
24
25    <!-- static global cosmap -->
26    <rosparam file="$(find my_package)/config/costmap_global_static.yaml" command="load" ns="global_costmap" unless="$(arg no_static_map)" />
27
28    <!-- global costmap with kinect -->
29    <rosparam file="$(find my_package)/config/costmap_global_laser.yaml" command="load" ns="global_costmap" if="$(arg no_static_map)" />
30    <param name="global_costmap/width" value="100.0" if="$(arg no_static_map)" />
31    <param name="global_costmap/height" value="100.0" if="$(arg no_static_map)" />
32
33    <!-- Remap into namespace for cmd_vel_mux switching -->
34    <remap from="cmd_vel" to="RosAria/cmd_vel" />
35    <remap from="odom" to="odom" />
36
37  </node>
38
39 </launch>
```

## 4.6 Τα config αρχεία για λειτουργία του AMCL, Move\_base

### 4.6.1 Το αρχείο costmap\_global\_static

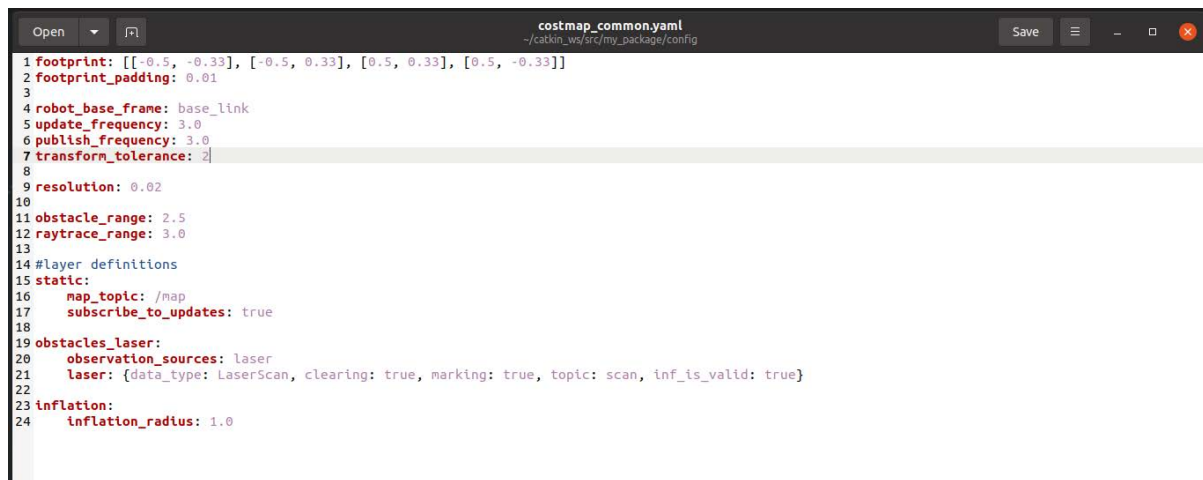
Costmap\_global\_static.yaml



```
1 global_frame: map
2 rolling_window: false
3 track_unknown_space: true
4
5 plugins:
6 - {name: static, type: "costmap_2d::StaticLayer"}
7 - {name: inflation, type: "costmap_2d::InflationLayer"}
```

#### 4.6.2 Το αρχείο costmap\_common

##### costmap\_common.yaml



```
1 footprint: [[-0.5, -0.33], [-0.5, 0.33], [0.5, 0.33], [0.5, -0.33]]
2 footprint_padding: 0.01
3
4 robot_base_frame: base_link
5 update_frequency: 3.0
6 publish_frequency: 3.0
7 transform_tolerance: 2
8
9 resolution: 0.02
10
11 obstacle_range: 2.5
12 raytrace_range: 3.0
13
14 #Layer definitions
15 static:
16   map_topic: /map
17   subscribe_to_updates: true
18
19 obstacles_laser:
20   observation_sources: laser
21   laser: {data_type: LaserScan, clearing: true, marking: true, topic: scan, inf_is_valid: true}
22
23 inflation:
24   inflation_radius: 1.0
```

#### 4.6.3 Το αρχείο costmap\_local

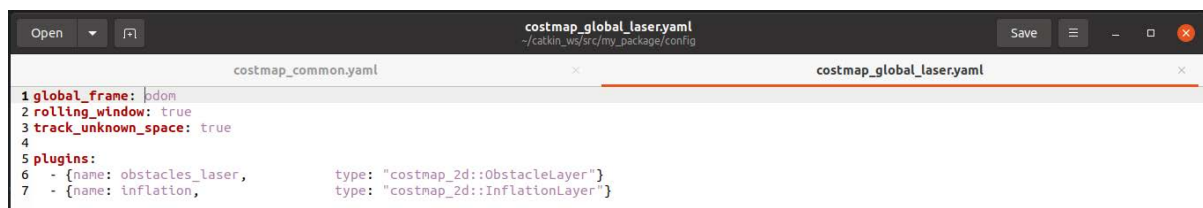
##### costmap\_local.yaml



```
1 global_frame: bdom
2 rolling_window: true
3
4 plugins:
5   - {name: obstacles_laser, type: "costmap_2d::ObstacleLayer"}
6   - {name: inflation, type: "costmap_2d::InflationLayer"}
```

#### 4.6.4 Το αρχείο costmap\_global\_laser

##### costmap\_global\_laser.yaml

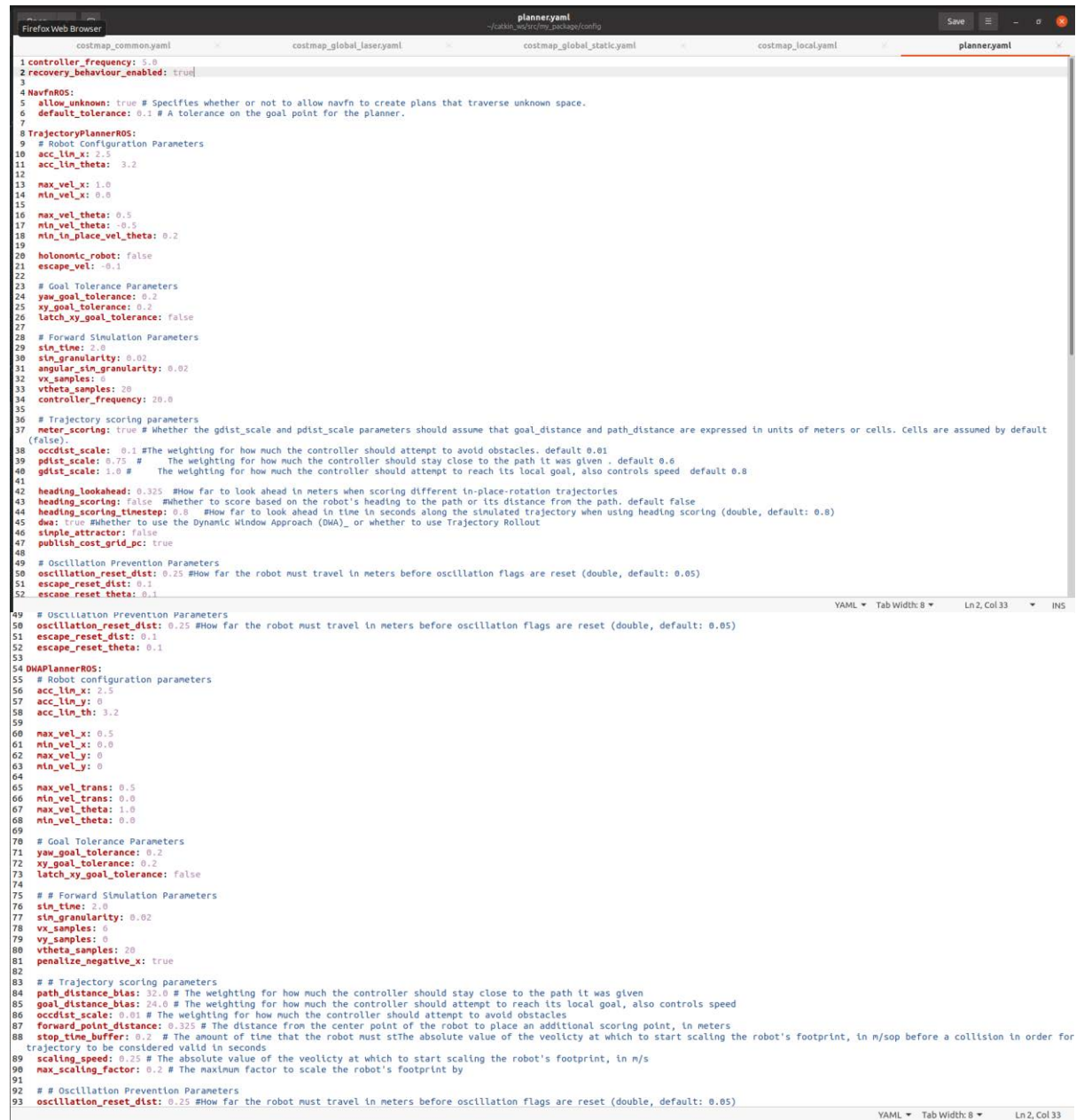


```
1 global_frame: bdom
2 rolling_window: true
3 track_unknown_space: true
4
5 plugins:
6   - {name: obstacles_laser, type: "costmap_2d::ObstacleLayer"}
7   - {name: inflation, type: "costmap_2d::InflationLayer"}
```



## 4.6.5 Το αρχείο planner

### planner.yaml



```
1 controller_frequency: 5.0
2 recovery_behaviour_enabled: true
3
4 NavfnROS:
5   allow_unknown: true # Specifies whether or not to allow navfn to create plans that traverse unknown space.
6   default_tolerance: 0.1 # A tolerance on the goal point for the planner.
7
8 TrajectoryPlannerROS:
9   # Robot Configuration Parameters
10  acc_lim_x: 2.5
11  acc_lim_theta: 3.2
12
13  max_vel_x: 1.0
14  min_vel_x: 0.0
15
16  max_vel_theta: 0.5
17  min_vel_theta: -0.5
18  min_in_place_vel_theta: 0.2
19
20  holonomic_robot: false
21  escape_vel: -0.1
22
23  # Goal Tolerance Parameters
24  yaw_goal_tolerance: 0.2
25  xy_goal_tolerance: 0.2
26  latch_xy_goal_tolerance: false
27
28  # Forward Simulation Parameters
29  sim_time: 2.0
30  sim_granularity: 0.02
31  angular_sim_granularity: 0.02
32  vx_samples: 6
33  vtheta_samples: 20
34  controller_frequency: 20.0
35
36  # Trajectory scoring parameters
37  meter_scoring: true # Whether the gdist_scale and pdist_scale parameters should assume that goal_distance and path_distance are expressed in units of meters or cells. Cells are assumed by default
38  (false).
39  occdist_scale: 0.1 #The weighting for how much the controller should attempt to avoid obstacles. default 0.01
40  pdist_scale: 0.75 # The weighting for how much the controller should stay close to the path it was given . default 0.6
41  gdist_scale: 1.0 # The weighting for how much the controller should attempt to reach its local goal, also controls speed default 0.8
42
43  heading_lookahead: 0.325 #How far to look ahead in meters when scoring different in-place/rotation trajectories
44  heading_scoring: false #Whether to score based on the robot's heading to the path or its distance from the path. default false
45  heading_scoring_timestep: 0.8 #How far to look ahead in time in seconds along the simulated trajectory when using heading scoring (double, default: 0.8)
46  dwa: true #Whether to use the Dynamic Window Approach (DWA), or whether to use Trajectory Rollout
47  simple_attractor: false
48  publish_cost_grid_pc: true
49
50  # Oscillation Prevention Parameters
51  oscillation_reset_dist: 0.1 #How far the robot must travel in meters before oscillation flags are reset (double, default: 0.05)
52  escape_reset_dist: 0.1
53  escape_reset_theta: 0.1
54
55  # Oscillation Prevention Parameters
56  oscillation_reset_dist: 0.25 #How far the robot must travel in meters before oscillation flags are reset (double, default: 0.05)
57  escape_reset_dist: 0.1
58  escape_reset_theta: 0.1
59
60 DWAPlannerROS:
61  # Robot configuration parameters
62  acc_lim_x: 2.5
63  acc_lim_y: 0
64  acc_lim_th: 3.2
65
66  max_vel_x: 0.5
67  min_vel_x: 0.0
68  max_vel_y: 0
69  min_vel_y: 0
70
71  max_vel_trans: 0.5
72  min_vel_trans: 0.0
73  max_vel_theta: 1.0
74  min_vel_theta: 0.0
75
76  # Goal Tolerance Parameters
77  yaw_goal_tolerance: 0.2
78  xy_goal_tolerance: 0.2
79  latch_xy_goal_tolerance: false
80
81  # Forward Simulation Parameters
82  sim_time: 2.0
83  sim_granularity: 0.02
84  vx_samples: 6
85  vy_samples: 0
86  vtheta_samples: 20
87  penalize_negative_x: true
88
89  # Trajectory scoring parameters
90  path_distance_bias: 32.0 # The weighting for how much the controller should stay close to the path it was given
91  goal_distance_bias: 24.0 # The weighting for how much the controller should attempt to reach its local goal, also controls speed
92  occdist_scale: 0.01 # The weighting for how much the controller should attempt to avoid obstacles
93  forward_point_distance: 0.325 # The distance from the center point of the robot to place an additional scoring point, in meters
94  stop_time_buffer: 0.2 # The amount of time that the robot must stay the absolute value of the velocity at which to start scaling the robot's footprint, in m/sop before a collision in order for
95  trajectory to be considered valid in seconds
96  scaling_speed: 0.25 # The absolute value of the velocity at which to start scaling the robot's footprint, in m/s
97  max_scaling_factor: 0.2 # The maximum factor to scale the robot's footprint by
98
99  # Oscillation Prevention Parameters
100 oscillation_reset_dist: 0.25 #How far the robot must travel in meters before oscillation flags are reset (double, default: 0.05)
```

## 4.7 Τα αρχεία του χάρτη που δημιουργήσα

### 4.7.1 Το launch αρχείο του χάρτη που δημιουργήσα

my\_lab\_map.launch

```
my_lab_map.launch
~/catkin_ws/src/my_package/launch

1 <?xml version="1.0"?>
2
3 <launch>
4   <arg name="map_file" value="$(find my_package)/maps/robot_lab3.yaml"> </arg>
5   <node pkg="map_server" type="map_server" name="map_server" args="$(arg map_file)">
6
7   </node>
8 </launch>
```

### 4.7.2 Το αρχείο .yaml του χάρτη

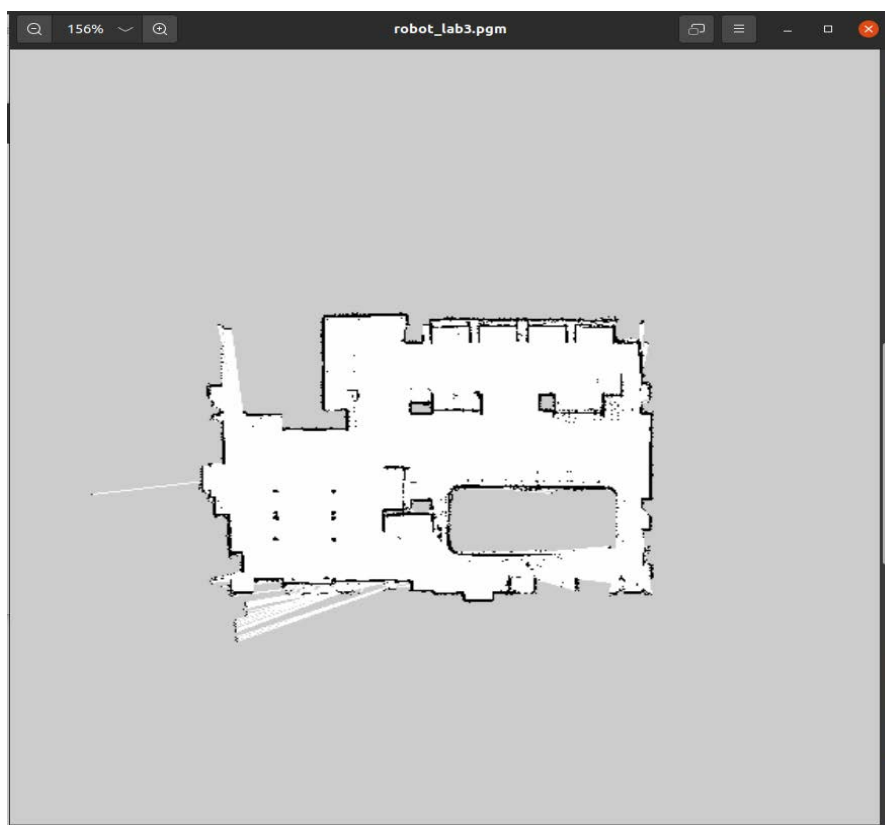
robot\_lab3.yaml

```
robot_lab3.yaml
~/catkin_ws/src/my_package/maps

1 image: robot_lab3.pgm
2 resolution: 0.050000
3 origin: [-50.000000, -50.000000, 0.000000]
4 negate: 0
5 occupied_thresh: 0.65
6 free_thresh: 0.196
7
```

### 4.7.3 Το αρχείο .pgm (φωτογραφία) του χάρτη

robot\_lab3.pgm



## ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα

---

Στη παρούσα υλοποίηση πραγματοποιείται μια χαρτογράφηση μέσω των αισθητήρων μετατόπισης που υπάρχουν στους τροχούς του τροχοφόρου (encoders) και μέσω του αισθητήρα laser (Sick lms 5xx). Αυτά τα εργαλεία έχουν ωφέλιμη, εύστοχη και ακριβής υλοποίηση και ίσως να έχει περιθώρια βελτίωσης αλλά οι αλλαγές που θα προσφέρουν στο σύστημα είναι μικρές σχετικά με το κόστος της αντικατάστασης των αισθητήρων ή κάποιας επιπλέον προσθήκης. Έπειτα επιτυγχάνεται αυτόνομη πλοήγηση στο χώρο η οποία επιζητά περιθώρια βελτίωσης καθώς μια αρκετά μεγάλη στιγμιαία επιτάχυνση δημιουργεί απώλειες και φθορές στο σύστημα. Γιατί οι μετρήσεις του αισθητήρων (καταγραφείς κίνησης encoders) έχουν την πιθανότητα να αποκλείσουν από τη σωστή μέτρηση στην περίπτωση που ολισθήσουν οι τροχοί και να δημιουργήσουν στρεβλώσεις στο σύστημα μας. Μια αναβάθμιση που μπορεί να φέρει είναι η βελτίωση του motion μοντέλου (planner) και των pid controller του ρομποτ. Επίσης, μπορεί να γίνει προσθήκη ενός αισθητήρα imu (γυροσκοπίου, επιταχυνσιόμετρου ) ώστε να προσφέρει μια ωφέλιμη αποσφαλμάτωση των μετρήσεων. Όπως παρατηρούμε είναι μια αρκετά πλήρης αλλά όχι και τέλεια υλοποίηση που μπορεί να φέρει εις πέρας τα καθήκοντα του με πολύ μικρές απώλειες όπως η ακριβής θέση του τροχοφόρου αλλά και το σημείο εκκίνησης του (αν ξεκινάει από άλλο σημείο εκκίνησης από αυτό που έγινε η καταγραφή του χάρτη).

## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

1. <https://el.wikipedia.org/wiki/Ρομποτική>
2. <https://www.ros.org/blog/ecosystem/>
3. <https://www.ros.org/blog/why-ros/>
4. <https://robots.ros.org/pioneer-3-at/>
5. <https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf>
6. <https://www.cyberbotics.com/doc/guide/pioneer-3at?version=R2021a#pioneer-3-at-model>
7. <https://www.conveyco.com/technology/autonomous-mobile-robots-amrs/>
8. <https://www.wevolver.com/article/understanding-amr-robots-a-comprehensive-guide>
9. <https://www.generationrobots.com/en/401697-sick-lms500-20000-pro-hr-indoor-laser-scanner.html>
10. [https://www.sick.com/media/docs/4/14/514/Operating\\_instructions\\_Laser\\_Measurement\\_Sensors\\_of\\_the\\_LMS5xx\\_Product\\_Family\\_en\\_IM0037514.PDF](https://www.sick.com/media/docs/4/14/514/Operating_instructions_Laser_Measurement_Sensors_of_the_LMS5xx_Product_Family_en_IM0037514.PDF)
11. <https://www.sick.com/be/en/catalog/products/lidar-and-radar-sensors/lidar-sensors/lms5xx/c/g179651>
12. <https://ubuntu.com/download/desktop>
13. <https://wiki.ros.org/ROSARIA#RosAria>
14. <https://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA>
15. [https://wiki.ros.org/sick\\_scan](https://wiki.ros.org/sick_scan)
16. <https://docs.ros.org/en/hydro/api/gmapping/html/index.html>
17. [http://wiki.ros.org/gmapping#slam\\_gmapping](http://wiki.ros.org/gmapping#slam_gmapping)
18. <http://wiki.ros.org/amcl>
19. [http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base)
20. <http://wiki.ros.org/noetic/Installation/Ubuntu>
21. <https://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>
22. <https://github.com/reedhedges/AriaCoda>
23. <https://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA>
24. [https://github.com/pengtang/rosaria\\_client](https://github.com/pengtang/rosaria_client)
25. [https://github.com/Gastd/p3at\\_tutorial](https://github.com/Gastd/p3at_tutorial)
26. [https://github.com/SICKAG/sick\\_scan](https://github.com/SICKAG/sick_scan)
27. [https://github.com/ros-perception/slam\\_gmapping](https://github.com/ros-perception/slam_gmapping)
28. <https://github.com/strawlab/navigation>