



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πειραματική υλοποίηση αλγορίθμου ταυτόχρονης χωροθέτησης και κατασκευής χάρτη υποβρυχίου ρομποτικού οχήματος με χρήση οπτικών ορόσημων

ΚΑΡΑΒΑΣ ΕΥΑΓΓΕΛΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΚΑΡΡΑΣ ΓΕΩΡΓΙΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Ιούνιος 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πειραματική υλοποίηση αλγορίθμου ταυτόχρονης
χωροθέτησης και κατασκευής χάρτη υποβρυχίου
ρομποτικού οχήματος με χρήση οπτικών ορόσημων

ΚΑΡΑΒΑΣ ΕΥΑΓΓΕΛΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΚΑΡΡΑΣ ΓΕΩΡΓΙΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Ιούνιος 2023



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Experimental implementation of simultaneous
localization and mapping algorithm of an
underwater robotic vehicle using visual
landmarks

EVANGELOS KARAVAS

FINAL THESIS

ADVISOR

GEORGE KARRAS

ASSISTANT PROFESSOR

Lamia June 2023

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις⁽¹⁾, 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 παραφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

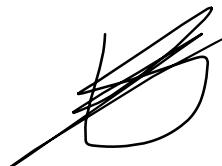
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και 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: 13/07/2023

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

Η παρούσα πτυχιακή εργασία εξετάζει το θέμα της ταυτόχρονης χωροθέτησης και κατασκευής χάρτη (Simultaneous Localization and Mapping) για υποβρύχιο ρομποτικό όχημα (Blue Rov 2, Blue Robotics) με χρήση κάμερας. Αρχικά, παρουσιάζεται μια εισαγωγή στη ρομποτική και στον τομέα των υποβρυχίων ρομπότ. Αναλύονται οι προκλήσεις που συναντά ένα ρομποτικό υποβρύχιο στην εκτέλεση της ταυτόχρονης χωροθέτησης και κατασκευής χάρτη, όπως η αβεβαιότητα της αισθητηριακής πληροφορίας και η περιορισμένη επικοινωνία με τον εξωτερικό κόσμο. Επίσης, αναφέρεται η βιβλιογραφία αλλά και πιθανές μελλοντικές αναπτύξεις που θα μπορούσαν να ακολουθήσουν. Στη συνέχεια, γίνεται μια περιγραφή του εξοπλισμού που χρησιμοποιήθηκε, όπως και μια εισαγωγή για την κατανόηση του ROS (Robot Operating System) και πώς το λογισμικό και τα εργαλεία του βοηθούν στο να δημιουργούνται εφαρμογές πάνω σε ρομποτικά οχήματα. Παρουσιάζονται επιπλέον, τα πακέτα διασύνδεσης του υποβρυχίου με το ROS καθώς και η λειτουργία και η χρήση τους. Ακολούθως, παρουσιάζονται πακέτα και αλγόριθμοι που έχουν αναπτυχθεί για την υποβρύχια χωροθέτηση Visual SLAM. Αναφέρονται οι προσεγγίσεις βασισμένες σε οπτικούς αισθητήρες (κάμερες), όπως και πληροφορίες για τα πακέτα που χρησιμοποιήθηκαν μέσω του ROS όπως το Aruco Detection και Fiducial Slam. Στη συνέχεια, παρουσιάζονται τα αποτελέσματα που προέκυψαν από την πειραματική εφαρμογή Visual SLAM του συστήματος Blue Rov 2 σε προσομοίωση αλλά και στο πραγματικό υποβρύχιο όχημα στη πισίνα του Πανεπιστημίου Θεσσαλίας. Τέλος, αναλύονται οι μετρήσεις της ακρίβειας του συστήματος και παρουσιάζονται οι χαρτογραφημένες περιοχές της πισίνας.

ABSTRACT

This thesis examines the topic of Simultaneous Localization and Mapping (SLAM) for an underwater robotic vehicle (Blue Rov 2, Blue Robotics) using a camera. First, an introduction to robotics and the field of underwater robots is presented. The challenges faced by an underwater robotic vehicle in performing simultaneous positioning and mapping are analyzed, such as the uncertainty of sensory information and limited communication with the outside world. In addition, the bibliography is mentioned as well as possible future developments that could follow. Then there is a description of the equipment used, as well as an introduction to understanding ROS (Robot Operating System) and how its software and tools help to create applications on robotic vehicles. In addition, the underwater vehicle interface packages with the ROS are presented as well as their operation and use. Next, packages and algorithms developed for underwater Visual SLAM positioning are presented. Approaches based on optical sensors (cameras) are reported, as well as information on packages used through ROS such as Aruco Detection and Fiducial Slam. Then, the results obtained from the experimental Visual SLAM application of the Blue Rov 2 system in simulation and in the real underwater vehicle in the pool of the University of Thessaly are presented. Finally, the system accuracy measurements are analyzed and the mapped areas of the pool are presented.

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	II
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	2
(ΥΠΟΚΕΦΑΛΑΙΟ 1.1) ΡΟΜΠΟΤΙΚΗ.....	2
(ΕΝΟΤΗΤΑ 1.1.Α) ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ	2
(ΕΝΟΤΗΤΑ 1.1.Β) ΥΠΟΒΡΥΧΙΑ ΡΟΜΠΟΤΙΚΑ ΟΧΗΜΑΤΑ (ROVs).....	3
(ΥΠΟΚΕΦΑΛΑΙΟ 1.2) ΠΡΟΒΛΗΜΑ.....	3
(ΥΠΟΚΕΦΑΛΑΙΟ 1.3) ΒΙΒΛΙΟΓΡΑΦΙΑ	4
(ΥΠΟΚΕΦΑΛΑΙΟ 1.4) ΤΙ ΑΚΟΛΟΥΘΕΙ	4
(ΥΠΟΚΕΦΑΛΑΙΟ 1.5) ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	5
ΚΕΦΑΛΑΙΟ 2 ΠΕΡΙΓΡΑΦΗ.....	6
(ΥΠΟΚΕΦΑΛΑΙΟ 2.1) BLUE ROV 2	6
(ΕΝΟΤΗΤΑ 2.1.Α) BLUE ROV 2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	6
(ΕΝΟΤΗΤΑ 2.1.Β) BLUE ROV 2 ΕΙΚΟΝΕΣ.....	7
(ΥΠΟΚΕΦΑΛΑΙΟ 2.2) ΚΑΜΕΡΑ SONY PS3	8
(ΕΝΟΤΗΤΑ 2.2.Α) ΠΕΡΙΓΡΑΦΗ ΚΑΜΕΡΑΣ	8
(ΕΝΟΤΗΤΑ 2.2.Β) ΕΙΚΟΝΕΣ ΤΗΣ ΚΑΜΕΡΑΣ.....	8
(ΥΠΟΚΕΦΑΛΑΙΟ 2.3) ΠΙΣΙΝΑ	11
(ΕΝΟΤΗΤΑ 2.3.Α) ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΠΙΣΙΝΑΣ.....	11
(ΕΝΟΤΗΤΑ 2.3.Β) ΕΙΚΟΝΕΣ ΤΗΣ ΠΙΣΙΝΑΣ.....	11
(ΥΠΟΚΕΦΑΛΑΙΟ 2.4) ROS (ROBOT OPERATING SYSTEM).....	15
(ΕΝΟΤΗΤΑ 2.4.Α) ΕΙΣΑΓΩΓΗ ΣΤΟ ROS	15
(ΕΝΟΤΗΤΑ 2.4.Β) ΕΙΣΑΓΩΓΗ ΣΤΑ TRANSFORMATIONS	16
(ΕΝΟΤΗΤΑ 2.4.Γ) ΕΙΣΑΓΩΓΗ ΣΤΟ RQT	17
(ΕΝΟΤΗΤΑ 2.4.Ε) ΕΙΣΑΓΩΓΗ ΣΤΟ RVIZ	18
(ΥΠΟΚΕΦΑΛΑΙΟ 2.5) BLUE ROV 2 και ROS	19
(ΕΝΟΤΗΤΑ 2.5.Α) ΠΑΚΕΤΟ ΣΥΝΔΕΣΗΣ BLUE ROV 2 ΣΕ ROS	19
(ΕΝΟΤΗΤΑ 2.5.Β) ΔΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΑΚΕΤΟΥ «BLUEROV_ROS_PLAYGROUND»	19
(ΕΝΟΤΗΤΑ 2.5.Γ) ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ «BLUEROV_ROS_PLAYGROUND».....	20
ΚΕΦΑΛΑΙΟ 3 SIMULTANEOUS LOCALIZATION AND MAPPING	22
(ΥΠΟΚΕΦΑΛΑΙΟ 3.1) VISUAL SLAM	22
(ΥΠΟΚΕΦΑΛΑΙΟ 3.2) FIDUCIAL SLAM	22
(ΕΝΟΤΗΤΑ 3.2.Α) ΕΙΣΑΓΩΓΗ ΣΤΑ FIDUCIAL MARKERS.....	22
(ΕΝΟΤΗΤΑ 3.2.Β) ΠΑΚΕΤΟ FIDUCIALS ΣΕ ROS	23
(ΕΝΟΤΗΤΑ 3.2.Γ) ΔΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΑΚΕΤΟΥ FIDUCIALS.....	24
(ΕΝΟΤΗΤΑ 3.2.Δ) ΧΡΗΣΗ ΤΟΥ ΠΑΚΕΤΟΥ FIDUCIALS.....	24
(ΕΝΟΤΗΤΑ 3.2.Ε) ΚΩΔΙΚΑΣ ΚΑΙ ΠΑΡΑΜΕΤΡΟΙ	25

ΚΕΦΑΛΑΙΟ 4 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	28
(ΥΠΟΚΕΦΑΛΑΙΟ 4.1) ΠΕΙΡΑΜΑ ΣΕ ΠΡΟΣΟΜΟΙΩΣΗ	29
(ΕΝΟΤΗΤΑ 4.1.Α) ΕΙΣΑΓΩΓΗ ΣΤΟ GAZEBO.....	29
(ΕΝΟΤΗΤΑ 4.1.Β) ΔΙΕΞΑΓΩΓΗ ΠΕΙΡΑΜΑΤΟΣ ΣΕ ΠΡΟΣΟΜΟΙΩΣΗ	29
(ΥΠΟΚΕΦΑΛΑΙΟ 4.2) ΠΕΙΡΑΜΑ ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ ΡΟΜΠΟΤΙΚΗΣ ΚΑΙ ΑΥΤΟΜΑΤΙΣΜΟΥ	34
(ΥΠΟΚΕΦΑΛΑΙΟ 4.3) ΣΥΜΠΕΡΑΣΜΑΤΑ	40
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	41

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

(Υποκεφάλαιο 1.1) ΡΟΜΠΟΤΙΚΗ

(Ενότητα 1.1.a) Εισαγωγή στη Ρομποτική

Η ρομποτική [1] είναι ένας τομέας της τεχνολογίας που ασχολείται με τον σχεδιασμό, την κατασκευή, τη λειτουργία και την επικοινωνία των ρομπότ. Ένα ρομπότ είναι ένα μηχανικό σύστημα που μπορεί να εκτελέσει καθήκοντα ή εργασίες με αυτονομία ή με ελάχιστη ανθρώπινη παρέμβαση. Η ρομποτική συνδυάζει πολλούς διαφορετικούς τομείς, όπως η μηχανική, η ηλεκτρονική, η πληροφορική, η τεχνητή νοημοσύνη και ο αυτόματος έλεγχος. Οι εφαρμογές της ρομποτικής καλύπτουν πολλούς τομείς, όπως η βιομηχανία, η υγεία, η εξερεύνηση διαστήματος, η αυτόνομη οδήγηση, η αεροναυπηγική, η ψυχαγωγία και πολλοί άλλοι. Αυτές οι τεχνολογίες χρησιμοποιούνται για την ανάπτυξη μηχανών που θα μπορούν να λειτουργούν ως υποκατάστατα του ανθρώπου.

Η ιδέα της δημιουργίας μηχανών που θα βασίζονται στην αυτονομία και τη λήψη αποφάσεων ανεξάρτητα από την ανθρώπινη παρουσία ξεκίνησε πολλά χρόνια πριν, ωστόσο δε γνώρισε πραγματική άνθηση μέχρι τον 20ο αιώνα. Η θεμελιώδης αρχή στη χρήση των ρομπότ, είναι να εξυπηρετούν τον άνθρωπο στην καθημερινότητά του. Τα τελευταία χρόνια, η έρευνα στη χρήση των ρομπότ έχει βελτιώσει σημαντικά τη ζωή του ανθρώπου και τον έχει οδηγήσει σε καινοτομίες.

Η Ρομποτική αναπτύσσει μηχανές που μπορούν να υποκαταστήσουν τον άνθρωπο και να αναπαράγουν ανθρώπινες ενέργειες. Τα ρομπότ μπορούν να χρησιμοποιηθούν σε πολλές καταστάσεις για πολλούς σκοπούς, αλλά σήμερα πολλά χρησιμοποιούνται σε επικίνδυνα περιβάλλοντα (συμπεριλαμβανομένης της επιθεώρησης ραδιενεργών υλικών, ανίχνευσης και απενεργοποίησης βομβών), σε διαδικασίες κατασκευής ή όπου οι άνθρωποι δεν μπορούν να επιβιώσουν (όπως στο διάστημα ή κάτω από τη θάλασσα). Τα ρομπότ μπορούν να πάρουν οποιαδήποτε μορφή, αλλά μερικά είναι φτιαγμένα για να μοιάζουν με ανθρώπους στην εμφάνιση. Αυτό υποστηρίζεται ότι βοηθά στην αποδοχή των ρομπότ σε ορισμένες αντιγραφικές συμπεριφορές που συνήθως εκτελούνται από ανθρώπους.

Ένα ρομπότ πρέπει να είναι ευέλικτο και να μπορεί να προσαρμόζεται στο περιβάλλον του. Αυτό γίνεται μέσω της χρήσης αισθητήρων και ενός συστήματος ελέγχου. Οι αισθητήρες συλλέγουν πληροφορίες από το περιβάλλον, όπως θερμοκρασία ή εικόνες, ενώ το σύστημα ελέγχου αποφασίζει και ρυθμίζει τη συμπεριφορά του ρομπότ.

Το σύστημα ελέγχου μπορεί να χωριστεί σε δύο κατηγορίες, ανάλογα με τον τρόπο λειτουργίας του. Η πρώτη κατηγορία είναι η ανοικτού βρόχου ελέγχου, όπου οι εντολές εκτελούνται βάσει των εισόδων από τους αισθητήρες, αλλά δεν υπάρχει ανατροφοδότηση για τον έλεγχο της συμπεριφοράς του ρομπότ. Η δεύτερη κατηγορία είναι ο κλειστού βρόχου έλεγχος, όπου οι αισθητήρες συλλέγουν πληροφορίες για την κατάσταση του ρομπότ και αυτές οι πληροφορίες χρησιμοποιούνται για να προσδιορίσουν και να ρυθμίσουν τις εντολές και τη συμπεριφορά του.

(Ενότητα 1.1.8) Υποβρύχια Ρομποτικά Οχήματα (ROVs)

Στις μέρες μας υπάρχουν όλο και πιο πολλά τηλεχειριζόμενα καταδυόμενα οχήματα που χρησιμοποιούνται αρκετά συχνά. Μια τηλεχειριζόμενη υποβρύχια ρομποτική μονάδα αποτελείται από το καταδυόμενο ρομποτικό όχημα, την μονάδα ελέγχου στην επιφάνεια και το καλώδιο που συνδέει το δυο προηγούμενα. Ο πλοήγηση γίνεται από χειριστή, μέσω της μονάδας ελέγχου, η παρουσία του οποίου είναι απαραίτητη και συνεχής για να μπορέσει το σκάφος να εκτελέσει την οποιαδήποτε αποστολή. Τα αυτόνομα υποβρύχια ρομποτικά οχήματα (AUVs) [2] σχεδιάστηκαν για να λειτουργούν αυτόνομα χωρίς την ανάγκη ανθρώπινης παρέμβασης. Τα AUVs χρησιμοποιούνται για μια ποικιλία αποστολών, όπως εξερεύνηση υποβρυχίων περιοχών, μελέτη των ωκεανών, περιβαλλοντική παρακολούθηση, ανεύρεση και ανάκτηση αντικειμένων, επισκευές υποβρυχίων καλωδίων και πολλές άλλες εφαρμογές. Τα AUVs συνήθως είναι εξοπλισμένα με διάφορους αισθητήρες, όπως ακουστικούς αισθητήρες (sonar), οπτικούς αισθητήρες (κάμερες), αισθητήρες μέτρησης φυσικών παραμέτρων (θερμοκρασία, αλατότητα, πίεση) και άλλους αισθητήρες που είναι απαραίτητοι για την επίτευξη των αποστολών τους.

Επιπλέον, τα AUVs διαθέτουν συστήματα πλοήγησης και ελέγχου που τους επιτρέπουν να πλοηγούνται αυτόνομα και να παρακολουθούν το περιβάλλον τους. Η τεχνολογία που χρησιμοποιείται στα AUVs συνεχώς εξελίσσεται, με την εισαγωγή νέων και προηγμένων αισθητήρων, τη βελτίωση της αυτονομίας, την αύξηση της ταχύτητας και τη βελτίωση της ακρίβειας της πλοήγησης. Επίσης, τα AUVs μπορούν να συνεργαστούν σε ομάδες για να επιτελέσουν συντονισμένες αποστολές και να ανταλλάξουν πληροφορίες.

Συνολικά, τα AUVs αντιπροσωπεύουν ένα σημαντικό πεδίο της υποβρύχιας ρομποτικής και παρέχουν μια αξιόπιστη και αποδοτική λύση για την εξερεύνηση και την ανάκτηση πληροφοριών από το υποβρύχιο περιβάλλον χωρίς τον συνεχή έλεγχο και καθοδήγηση από τον χειριστή.

(Υποκεφάλαιο 1.2) ΠΡΟΒΛΗΜΑ

Σήμερα λιγότερο από το 20% του υποβρύχιου κόσμου έχει εξερευνηθεί. Αυτό συμβαίνει λόγω των αντίξων συνθηκών επιβίωσης της ανθρώπινης ζωής σε τέτοια περιβάλλοντα. Πιο συγκεκριμένα ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζει ο άνθρωπος για την εξερεύνηση του βυθού είναι οι τοξικές συνθήκες που υπάρχουν υποβρυχίως (όπως η τοξικότητα του οξυγόνου, νόσος των δυτών κ.α.). Γι' αυτό, η αποστολή ενός δύτη σε μεγάλα υποβρύχια βάθη είναι πολύ επικίνδυνη και ακριβή διαδικασία. Υπάρχει μεγάλο εύρος αποστολών στον βυθό, όπως η αντικατάσταση ή η επιδιόρθωση υποβρύχιων καλωδίων, η χαρτογράφηση του ωκεανού κ.α. Στην παρούσα πτυχιακή εργασία υλοποιείται κατάλληλος αλγορίθμος για την ταυτόχρονη χωροθέτηση και κατασκευή χάρτη (Simultaneous Localization and Mapping) με χρήση κάμερας και οπτικών οροσήμων, για το υποβρύχιο ρομποτικό όχημα (Blue Rov 2, Blue Robotics) του Εργαστηρίου Ρομποτικής & Αυτομάτου Ελέγχου, του Τμήματος Πληροφορικής & Τηλεπικοινωνιών του Πανεπιστημίου Θεσσαλίας.

Ένα από τα προβλήματα χαρτογράφησης με τη βοήθεια οπτικών οροσήμων είναι η μελέτη και η υλοποίηση κατάλληλου αλγορίθμου ώστε η εκτίμηση θέσης και η δημιουργία χάρτη να είναι

ακριβής και αξιόπιστη. Προβλήματα όπως η αστάθεια των αισθητήρων, η ασυμβατότητα μεταξύ των αισθητήρων ή η ανεπαρκής ποιότητα των μετρήσεων μπορούν να επηρεάσουν την ακρίβεια και την αξιοπιστία του αλγορίθμου. Επίσης, Οι αλγόριθμοι SLAM απαιτούν σημαντικούς υπολογιστικούς πόρους, καθώς πρέπει να εκτελούν επαναλαμβανόμενους υπολογισμούς για την εκτίμηση της θέσης και του χάρτη. Αυτό μπορεί να οδηγήσει σε προβλήματα υπερφόρτωσης του υπολογιστικού συστήματος ή σε κατώτερη αναμενόμενη απόδοση. Επιπλέον, η αδυναμία χρήσης GPS κάνει την διαδικασία αποστολής ενός ROV κάτω από το νερό πολύ πιο δύσκολη από ότι στην επιφάνεια.

Αυτά είναι μερικά από τα προβλήματα που πρέπει να αντιμετωπιστούν για τη σωστή υλοποίηση του SLAM. Οι προκλήσεις αυτές αντιμετωπίζονται με τη χρήση βελτιστοποιημένων αλγορίθμων, προηγμένων αισθητήρων και τον κατάλληλο προσδιορισμό των παραμέτρων των καμερών που χρησιμοποιεί το ρομπότ.

(Υποκεφάλαιο 1.3) Βιβλιογραφία

Έχουν δημιουργηθεί πολλά υποβρύχια ρομποτικά οχήματα (ROVs) με σκοπό την εξερεύνηση του ωκεανού, όπως και ένα μεγάλο υπόβαθρο έρευνας που σχετίζεται με αυτό. Οι αλγόριθμοι που έχουν αναπτυχθεί, μπορούν να αντιμετωπίσουν εκπληκτικά μεγάλα προβλήματα στη χαρτογράφηση και χωροθέτηση. Έρευνες, προσέγγισαν να διατηρήσουν έναν ενιαίο χάρτη κατά τη λειτουργία της χαρτογράφησης και να ενημερώνει το τρέχον σχέδιο όταν το SLAM χάνει την παρακολούθηση των χαρακτηριστικών. Με αποτέλεσμα, τα πειράματα στο νερό να δείχνουν τα πλεονεκτήματα της προσέγγισης αυτής που χρησιμοποιείται σε ένα AUV. Επίσης, άλλες έρευνες έχουν προτείνει αλγόριθμο SLAM με στόχο την ταχύτερη επεξεργασία μεγάλου όγκου δεδομένων που προκαλείτε από μεγάλη ποσότητα δεδομένων εικόνας, δίνοντας προτάσεις για καλύτερη αναπαράσταση χαρακτηριστικών του περιβάλλοντος, συγχώνευση δεδομένων αισθητήρων για την καταστολή της συσσώρευσης σφαλμάτων και τη δημιουργία ενός ακριβούς χάρτη [4]. Άλλες προσεγγίσεις τεχνικών έχουν εμπνευστεί από χαρακτηριστικά ζώων για την ανάπτυξη νέων μεθόδων και αλγορίθμων στο πλαίσιο του SLAM [5]. Πιο συγκεκριμένα, η εργασία αυτή πρότεινε έναν αλγόριθμο, που βασίζεται στον εντοπισμό που εκτελείται από νευρικές δομές που βρίσκονται σε θηλαστικά όπως οι νυχτερίδες, και αναμένεται να είναι παρόμοια με εκείνα που χρησιμοποιούνται από τα δελφίνια, και οι ερευνητές βασίστηκαν σε προσεγγίσεις SLAM οι οποίες μοντελοποίησαν μηχανισμούς πλοϊγησης εγκεφάλου αρουραίων (RatSLAM).

Υπάρχει μια μεγάλη ποικιλία προσεγγίσεων SLAM που διατίθενται στην κοινότητα της ρομποτικής. Έχουν προταθεί πολλοί αλγόριθμοι και αποτελεσματικοί τρόποι για την ελαχιστοποίηση σφαλμάτων, αύξηση ακρίβειας της θέσης και προσανατολισμού του ρομπότ, και μια συνολική προσέγγιση βελτιστοποίησης των χαρτών.

Επιπλέον, πολλοί οργανισμοί που ασχολούνται ενεργά με τον κόσμο του ωκεανού, έχουν δημοσιοποιήσει πακέτα οργανωμένου κώδικα για την ευκολότερη χρήση τους και την ώθηση στην ανάπτυξη νέων τεχνικών χωροθέτησης και χαρτογράφησης.

(Υποκεφάλαιο 1.4) Τι Ακολουθεί

Υπάρχει μεγάλο εύρος ανάπτυξης στον τομέα της ρομποτικής και πιο συγκεκριμένα στον τομέα των υποβρυχίων. Η πτυχιακή εργασία με θέμα την υλοποίηση αλγορίθμου SLAM για υποβρύχιο ρομποτικό όχημα με χρήση οπτικών ορόσημων δεν έχει τελειοποιηθεί, και ένα από τα προβλήματα ήταν η σταθερότητα του υποβρυχίου. Η κίνηση του υποβρύχιου οχήματος Blue Rover 2 δεν είναι ομαλή όσο αφορά τη προσθήκη της κάμερας για την προβολή του βυθού, και οι μετρήσεις της κάμερας μπορούν να οδηγήσουν σε ελαφρά διαφορετικές σκηνές κατά τη δημιουργία του χάρτη. Για το πρόβλημα αυτό αφαιρέθηκε ένα βαρίδιο από το δεξί μέρος του υποβρυχίου το οποίο βοήθησε στη σταθεροποίηση του, αλλά χρειάζεται παραπάνω σταθεροποίηση. Επίσης, υπάρχουν πολλοί τρόποι υλοποίησης αλγορίθμων χαρτογράφησης και χωροθέτησης όπου μπορούν να ερευνηθούν μελλοντικά. Για παράδειγμα, η ανάπτυξη αλγορίθμου SLAM με χρήση υποβρύχιων ραντάρ (sonar) για μια ευρύτερη λειτουργία τέτοιων αλγορίθμων σε πιο σκοτεινά περιβάλλοντα όπως σε μεγάλο βάθος νερού (1000m), ή σπήλαια που δεν τα διαπερνά το φως του ήλιου.

(Υποκεφάλαιο 1.5) Δομή της Εργασίας

Η παρούσα πτυχιακή εργασία αποτελείται από 4 κεφάλαια:

- 1^ο Κεφάλαιο: Γίνεται η εισαγωγή στην εργασία, περιγράφεται ο σκοπός της, το πρόβλημα που εξετάζουμε όπως και μια βιβλιογραφική επισκόπηση. Τέλος, αναφέρεται τι μπορεί να ακολουθήσει μετά από αυτήν την εργασία και οι διάφοροι τρόποι υλοποίησης αλγορίθμων χαρτογράφησης και χωροθέτησης.
- 2^ο Κεφάλαιο: Περιγραφή του υποβρύχιου ρομποτικού οχήματος όπως και των αλλαγών πάνω σε αυτό (προσθήκη κάμερας προβολής του βυθού), παρουσίαση εικόνων του ρομπότ, της κάμερας και της πισίνας για την πειραματική υλοποίηση του αλγορίθμου.
- 3^ο Κεφάλαιο: Πληροφορίες σχετικά με το Visual SLAM και περεταίρω επεξήγηση για το πακέτο Fiducial SLAM και τον κώδικα. Επίσης γίνεται αναφορά στις αλλαγές στα πακέτα που χρησιμοποιήθηκαν και αλλαγή παραμέτρων για την καλύτερη λειτουργία στο συγκεκριμένο υποβρύχιο ρομποτικό όχημα.
- 4^ο Κεφάλαιο: Προβολή των αποτελεσμάτων της πειραματικής υλοποίησης του αλγορίθμου στο εργαστήριο ρομποτικής αλλά και σε προσομοίωση (Gazebo), και παρουσίαση αποδοτικότητας και ακρίβειας των αποτελεσμάτων.

ΚΕΦΑΛΑΙΟ 2 Περιγραφή

(Υποκεφάλαιο 2.1) Blue Rov 2

(Ενότητα 2.1.a) Blue Rov 2 Χαρακτηριστικά

Το υποβρύχιο Blue ROV 2 [6] είναι ένα αυτόνομο ρομποτικό υποβρύχιο που έχει σχεδιαστεί για εξερεύνηση και εργασίες υποθαλάσσιας ρομποτικής. Αναπτύχθηκε από την εταιρεία Blue Robotics [7], που ειδικεύεται στην κατασκευή προσιτών υποβρυχίων ρομπότ.

Το BlueROV2 διαθέτει μια ανθεκτική κατασκευή και είναι σχεδιασμένο για να λειτουργεί σε βαθιά νερά έως και 300 μέτρα. Έχει μήκος περίπου μισού μέτρου και βάρος περίπου 18 κιλών όπως φαίνεται και στην Εικόνα 1. Είναι εξοπλισμένο με ισχυρούς ηλεκτρικούς κινητήρες και ενσωματωμένες μπαταρίες που του παρέχουν μεγάλη αυτονομία και ευελιξία.

Το BlueROV2 (Εικόνα 2, Εικόνα 3) διαθέτει ενσωματωμένο σύστημα αισθητήρων που του επιτρέπει να ανιχνεύει την περιβάλλουσα ατμόσφαιρα, τη θερμοκρασία του νερού και της παραμέτρης πίσης, διαθέτει κάμερες υψηλής ανάλυσης για την καταγραφή εικόνων και βίντεο κατά τη διάρκεια των αποστολών. Ένα από τα κύρια χαρακτηριστικά του BlueROV2 είναι η δυνατότητα του για αυτόνομη πλοήγηση και έλεγχο. Με τη χρήση ειδικού λογισμικού και την υποστήριξη του ROS, το BlueROV2 μπορεί να εκτελέσει αποστολές αυτόνομα μέσω περίπλοκων κινήσεων υποθαλάσσιας πτηζης.

Επίσης, το υποβρύχιο αυτό χρησιμοποιείται σε δημόσιους τομείς, όπως η υποθαλάσσια έρευνα, η επισκευή και συντήρηση υποθαλάσσιων υποδομών, η εξερεύνηση υποθαλάσσιων τοπίων και αρχαιολογικών ευρημάτων. Το σχετικά χαμηλό κόστος και η ευελιξία του καθιστούν προσιτό για ερευνητές, εκπαιδευτικά ιδρύματα και ερασιτεχνική ρομποτική. Συνολικά, το υποβρύχιο BlueROV2 είναι ένα ευέλικτο και προσιτό ρομποτικό υποβρύχιο που προσφέρει ευκαιρίες για υποθαλάσσια έρευνα και ανακάλυψη και αποτελεί ένα από τα πιο δημοφιλή υποβρύχια ρτηζότητα της ρομποτικής.

ΟΝΟΜΑ: Blue Rov 2

ΚΑΤΑΣΚΕΥΑΣΤΗΣ: Blue Robotics

ΔΙΑΣΤΑΣΕΙΣ: 457 mm x 338 mm x 254 mm

ΚΙΝΗΤΗΡΕΣ: 6 Blue Robotics T200

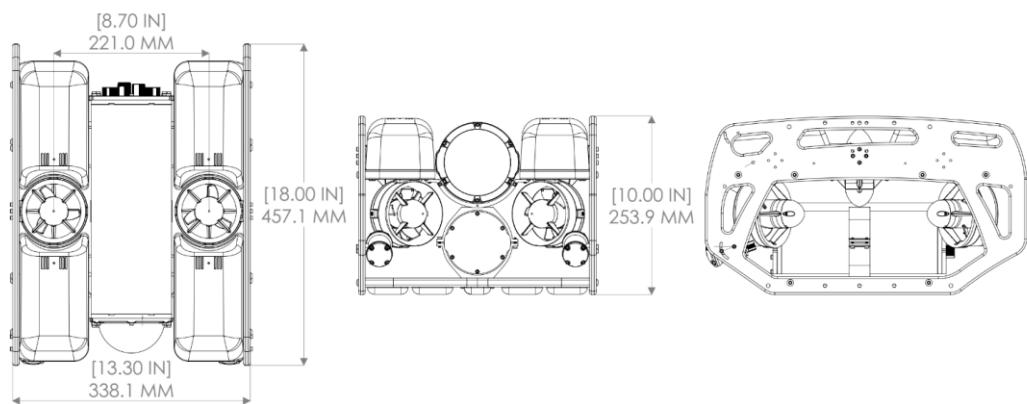
ΒΑΘΟΣ ΚΑΤΑΔΥΣΗΣ: 300m

ΒΑΡΟΣ: 12kg

ΑΙΣΘΗΤΗΡΕΣ: IMU, Compass, Internal Pressure, Pressure/Depth and Temperature Sensor (external), Leak Sensor, Current and Voltage Sensing, Camera

ΔΙΕΥΘΥΝΣΗ ΑΓΟΡΑΣ: <https://bluerobotics.com/store/rov/bluerov2/>

(Ενότητα 2.1.8) Blue Rov 2 Εικόνες



Εικ. 1 (Οι διαστάσεις του υποβρύχιου ρομποτικού οχήματος (AUV) Blue Rov 2)



Εικ. 2 (Το υποβρύχιο Blue Rov 2)



Εικ. 3 (Blue Rov 2 από διαφορετικές γωνίες θέασης)

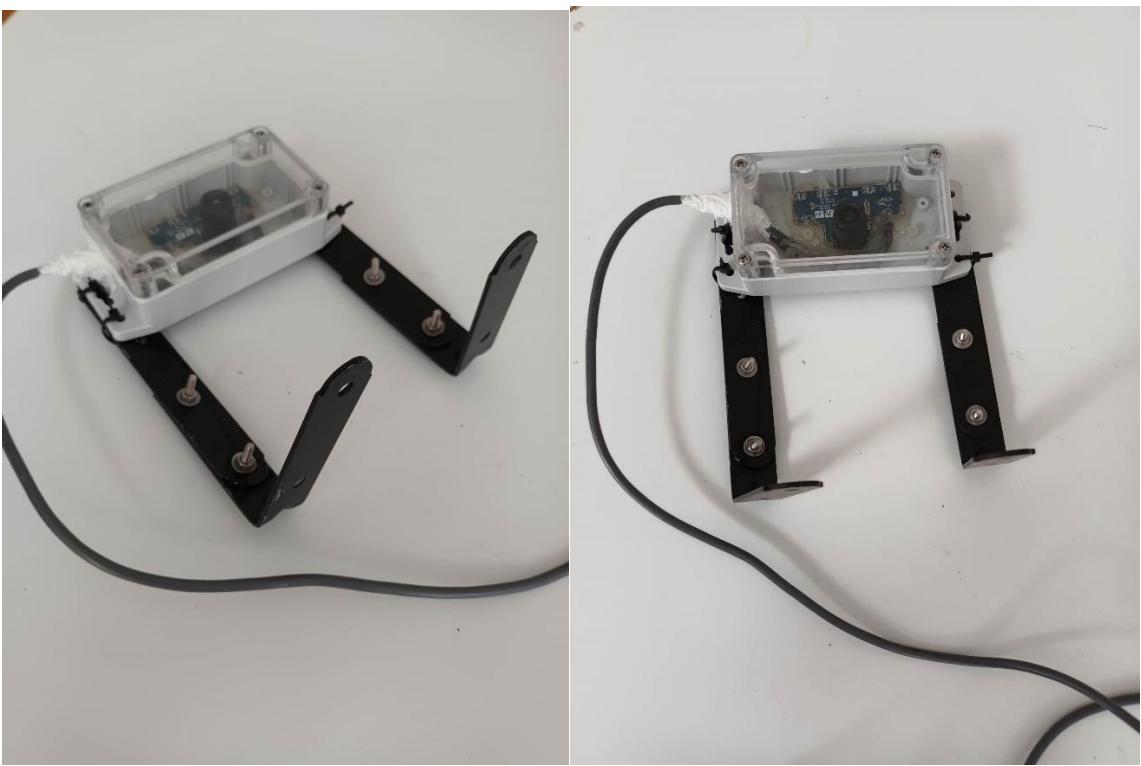
(Υποκεφάλαιο 2.2) Κάμερα SONY PS3

(Ενότητα 2.2.a) Περιγραφή Κάμερας

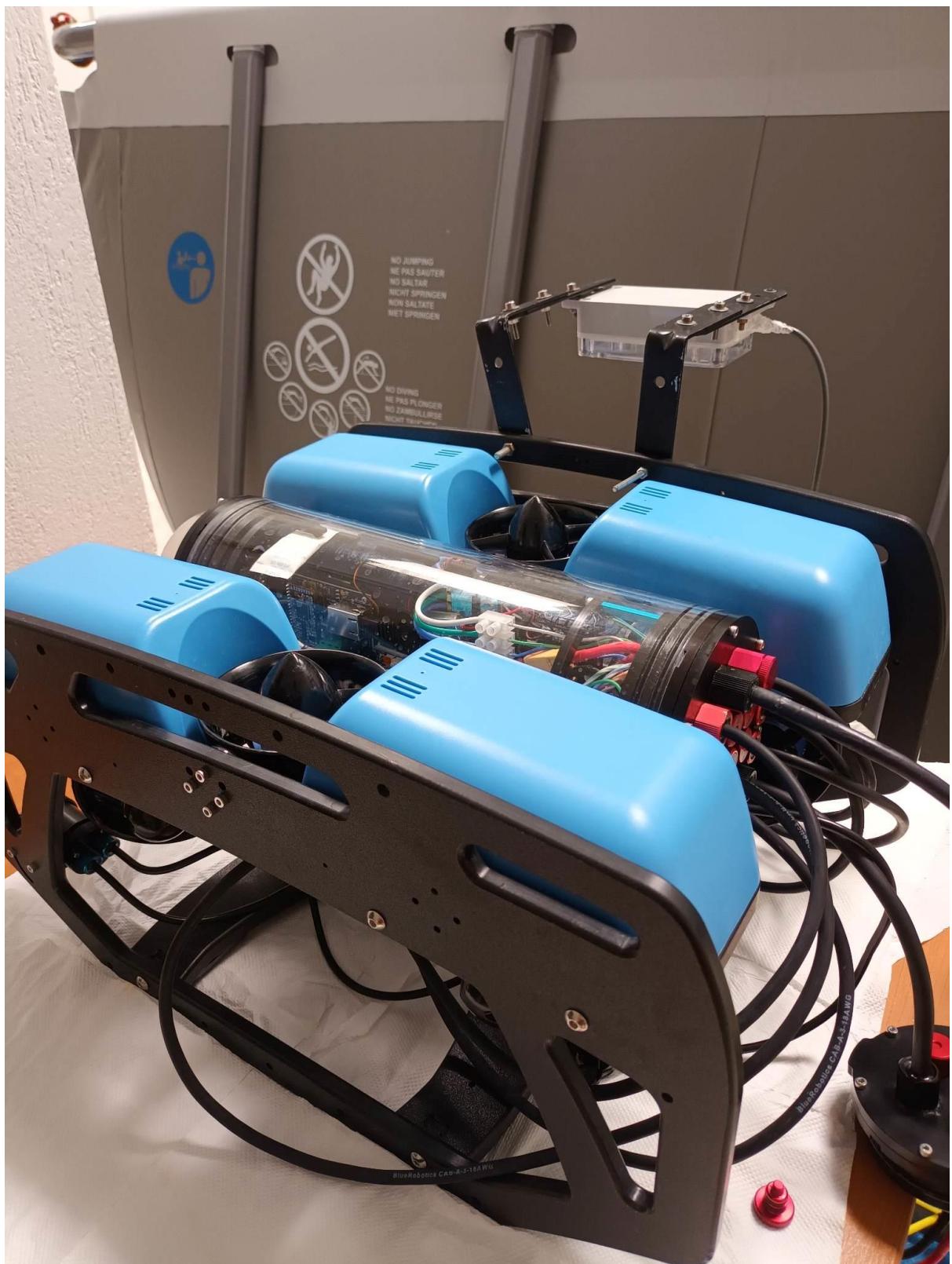
Για την χαρτογράφηση του βυθού της πισίνας χρησιμοποιήθηκε μια κάμερα SONY-PS3 (Εικόνα 4) η οποία τοποθετήθηκε στα δεξί μέρος του ρομπότ, με μια μεταλλική βάση η οποία βιδώθηκε πάνω στο ρομπότ όπως φαίνεται στις Εικόνα 5 και Εικόνα 6. Επίσης, για την προστασία της κάμερας από το νερό χρησιμοποιήθηκε ένα αδιάβροχο προστατευτικό κάλυμμα το οποίο είχε δημιουργηθεί από προηγούμενη υλοποίηση αλγορίθμου ανίχνευσης οπτικών δεικτών στο εργαστήριο Ρομποτικής του Πανεπιστημίου Θεσσαλίας. Για τη σωστή τοποθέτηση της κάμερας ήταν απαραίτητη η αφαίρεση ενός βαριδίου από το υποβρύχιο, για την ισορροπημένη πλευστότητα και ρύθμιση του συνολικού βάρους.

Η κάμερα έχει ξεχωριστό καλώδιο το οποίο συνδέεται απευθείας με τον ηλεκτρονικό υπολογιστή στον οποίο συνδέεται και το υποβρύχιο ρομπότ. Το καλώδιο της κάμερας έχει μήκος 10 μέτρα όπου στην περίπτωση του πειράματος δεν περιόριζε τις κινήσεις του υποβρυχίου (το μήκος της πισίνας είναι 6 μέτρα).

(Ενότητα 2.2.β) Εικόνες της Κάμερας



Εικ. 4 (Κάμερα SONY-PS3 σε μια προστατευτική θήκη για χρήση κάτω από το νερό)



Εικ. 5 (Η κάμερα «SONY-PS3» ενσωματωμένη στο ρομποτικό όχημα Blue Rov 2)



Εικ. 6 (Ενσωματωμένη κάμερα στο υποβρύχιο από διαφορετικές γωνίες θέασης)

(Υποκεφάλαιο 2.3) Πισίνα

(Ενότητα 2.3.a) Περιγραφή της πισίνας

Η πισίνα που χρησιμοποιήθηκε για την διεξαγωγή του πειράματος είναι από το εργαστήριο Ρομποτικής και έχουν τοποθετηθεί στον βυθό της πισίνας 171 κατάλληλοι οπτικοί δείκτες (Aruco Markers) περιμέτρου 10cm x 10cm και μικρότεροι δείκτες 5cm x 5cm ώστε ο χάρτης να μην είναι στατικός και επίπεδος, αλλά δημιουργώντας την αντίληψη ότι μερικοί δείκτες βρίσκονται σε μεγαλύτερο βάθος. Η πισίνα όπως φαίνεται και στην Εικόνα 7 και Εικόνα 8 έχει 6 μέτρα μήκος, 3 μέτρα πλάτος, και 1,5 μέτρα βάθος πράγμα που καθιστά δύσκολη την τοποθέτηση της κάμερας λόγω του περιορισμού βάθους της πισίνας. Γι' αυτόν τον λόγο η κάμερα τοποθετήθηκε στο δεξί μέρος του ρομπότ και όχι στο κέντρο του, δίνοντας μεγαλύτερο οπτικό πεδίο στον φακό της κάμερας κάνοντας την χαρτογράφηση των δεικτών ευκολότερη και ταχύτερη (Εικόνα 9).

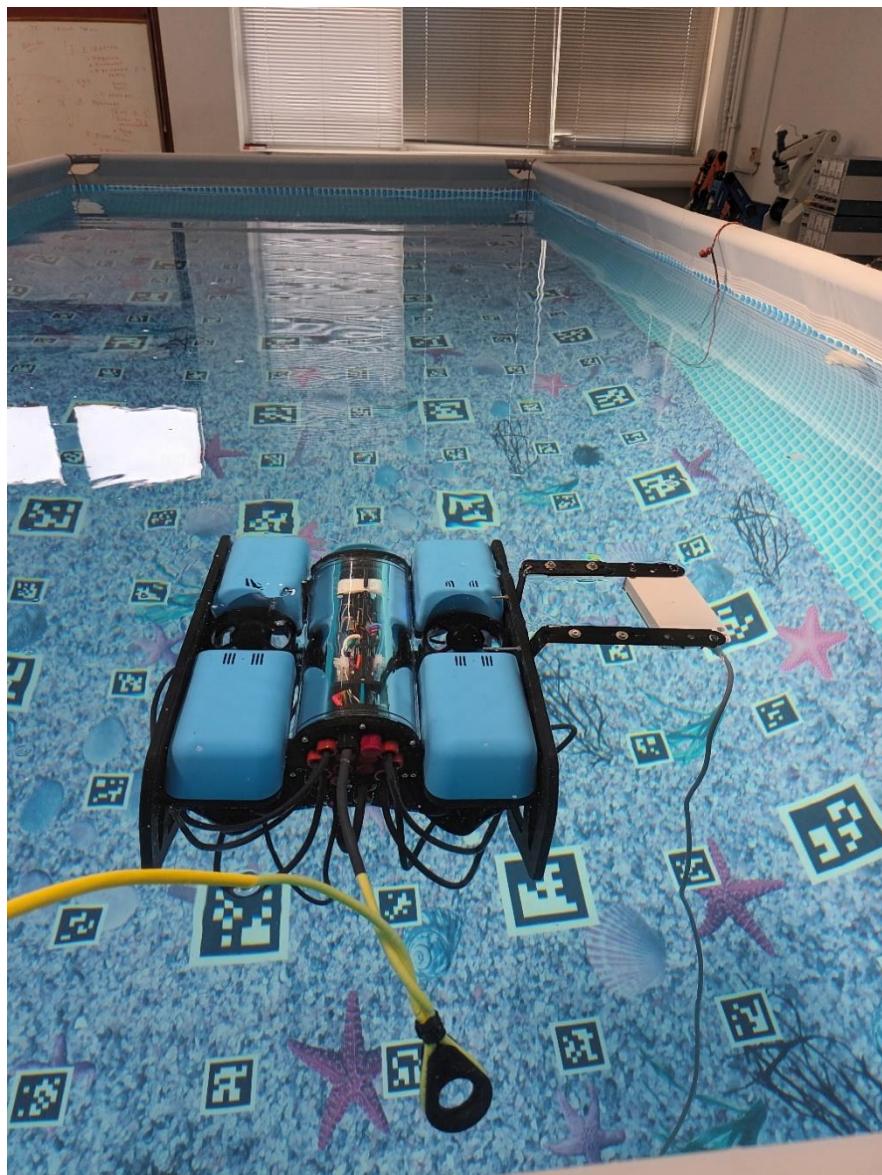
(Ενότητα 2.3.b) Εικόνες της πισίνας



Εικ. 7 (Πισίνα Πανεπιστημίου Θεσσαλίας με διαστάσεις: 6m x 3m x 1,5m)



Εικ. 8 (Πισίνα Πανεπιστημίου Θεσσαλίας Τμήματος Πληροφορικής και Τηλεποικινωνιών)



Εικ. 9 (To Blue Rov 2 βυθισμένο στη πισίνα)

(Υποκεφάλαιο 2.4) ROS (Robot Operating System)

(Ενότητα 2.4.a) Εισαγωγή στο ROS

Το ROS (Robot Operating System) [8] είναι ένα λογισμικό ανοιχτού κώδικα που αποτελεί μια ευέλικτη και επαναχρησιμοποιήσιμη πλατφόρμα για την ανάπτυξη ρομποτικών εφαρμογών. Αρχικά αναπτύχθηκε από το Willow Garage [9], ένα ερευνητικό εργαστήριο, και αργότερα παραχωρήθηκε στο Open Source Robotics Foundation (OSRF) [10] που συνεχίζει την ανάπτυξη και τη συντήρησή του.

Το ROS παρέχει ένα πλήθος εργαλείων, βιβλιοθηκών και βασικών λειτουργιών που επιτρέπουν την ανάπτυξη, τη δοκιμή και την εκτέλεση ρομποτικών συστημάτων. Ο στόχος του ROS είναι να διευκολύνει την ανταλλαγή πληροφοριών μεταξύ διάφορων υποσυστημάτων του ρομπότ και να προωθήσει την επαναχρησιμοποίηση του κώδικα και των αλγορίθμων.

Παρόλο που το ROS περιέχει τη λέξη "Operating System" (λειτουργικό σύστημα), δεν είναι ένα πραγματικό λειτουργικό σύστημα, αλλά ένα λογισμικό πλαίσιο που εκτελείται πάνω από ένα υπάρχον λειτουργικό σύστημα (όπως το Linux).

Οι βασικές αρχές του ROS περιλαμβάνουν:

- **Master:** Το ROS Master παρέχει καταχώριση ονόματος και αναζήτηση στο υπόλοιπο Υπολογιστικό Γράφημα. Χωρίς το Master, οι κόμβοι δεν θα μπορούσαν να βρουν ο ένας τον άλλον, να ανταλλάξουν μηνύματα ή να καλέσουν υπηρεσίες.
- **Κόμβους (Nodes):** Το ROS είναι μια δικτυακή αρχιτεκτονική που βασίζεται στην έννοια των κόμβων. Κάθε κόμβος αντιπροσωπεύει ένα ανεξάρτητο υποσύστημα που εκτελεί συγκεκριμένες λειτουργίες και επικοινωνεί με άλλους κόμβους μέσω μηνυμάτων.
- **Μηνύματα (Messages):** Οι κόμβοι ανταλλάσσουν πληροφορίες μεταξύ τους μέσω μηνυμάτων. Το ROS παρέχει ένα σύστημα ορισμού μηνυμάτων που επιτρέπει τη μεταφορά δεδομένων, όπως αισθητήρες εισόδου, πληροφορίες κατάστασης, εντολές ελέγχου κ.λπ.
- **Θέματα (Topics):** Τα θέματα είναι μηχανισμοί δημοσίευσης/συνδρομής που επιτρέπουν την ανταλλαγή μηνυμάτων μεταξύ πολλαπλών κόμβων. Ένας κόμβος μπορεί να δημοσιεύει πληροφορίες σε ένα θέμα, ενώ άλλοι κόμβοι μπορούν να ζητήσουν πληροφορίες από αυτό το θέμα.
- **Υπηρεσίες (Services):** Οι υπηρεσίες επιτρέπουν στους κόμβους να καλούν και να αποκτούν πρόσβαση σε λειτουργίες άλλων κόμβων. Αντί για την ασύγχρονη ανταλλαγή μηνυμάτων όπως στα θέματα, οι υπηρεσίες προσφέρουν σύγχρονη διεπαφή κλήσης/απόκρισης.
- **Πακέτα (Packages):** Το ROS οργανώνει τον κώδικα, τις βιβλιοθήκες και τις εξαρτήσεις του σε πακέτα. Ένα πακέτο περιέχει τον κώδικα και τα αρχεία πόρων για ένα συγκεκριμένο λειτουργικό σύνολο.
- **Bags:** Τα bags είναι μια μορφή αποθήκευσης και αναπαραγωγής δεδομένων μηνυμάτων ROS. Τα bags είναι ένας σημαντικός μηχανισμός για την αποθήκευση δεδομένων, όπως δεδομένα αισθητήρων, που μπορεί να είναι δύσκολο να συλλεχθούν αλλά είναι απαραίτητα για την ανάπτυξη και τη δοκιμή αλγορίθμων.

Το ROS παρέχει επίσης εργαλεία για την οπτικοποίηση δεδομένων, τον χειρισμό μεγάλων ποσοτήτων δεδομένων, την παρακολούθηση κατάστασης του συστήματος, την αποσφαλμάτωση και πολλά άλλα. Επιπλέον, υπάρχουν πολλές κοινότητες και πακέτα τρίτων που προσθέτουν επιπλέον λειτουργικότητα και υποστήριξη για διάφορους τομείς της ρομποτικής.

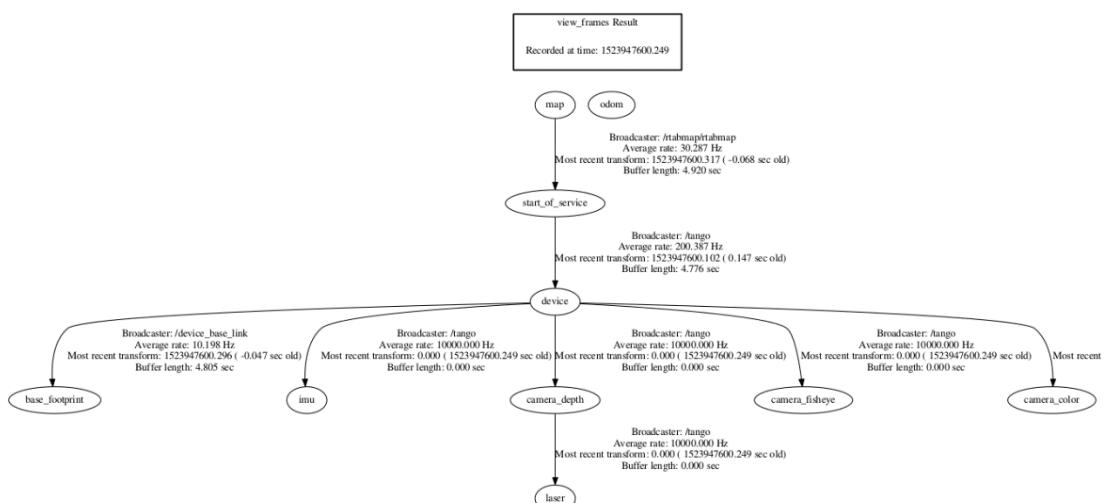
Το ROS χρησιμοποιείται ευρέως στην ακαδημαϊκή έρευνα και τη βιομηχανία για την ανάπτυξη ρομποτικών συστημάτων. Η ευελιξία του, μαζί με την πληθώρα των διαθέσιμων εργαλείων και βιβλιοθηκών, καθιστούν το ROS ένα ισχυρό εργαλείο για την ανάπτυξη και την επίλυση προβλημάτων στον τομέα της ρομποτικής.

(Ενότητα 2.4.β) Εισαγωγή στα Transformations

Τα ρομποτικά οχήματα αλληλεπιδρούν με τον πραγματικό κόσμο, γι' αυτό είναι απαραίτητες οι συντεταγμένες που θα περιγράφουν τη θέση των εξαρτημάτων κι αισθητήρων στο ρομποτικό όχημα, και οι θέσεις των αντικειμένων με τα οποία θα αλληλεπιδρά το ρομπότ. Μέσω του ROS, αυτές οι συντεταγμένες ορίζονται σε ένα σύστημα coordinate frames και ονομάζονται Transformations (TF) [11].

Για τη λήψη πληροφοριών από το σύστημα TF είναι αναγκαία η γνώση του με ποιον τρόπο συνδέονται όλες οι θέσεις των εξαρτημάτων. Δηλαδή, ένα ρομποτικό σύστημα έχει συνήθως πολλά τρισδιάστατα πλαίσια συντεταγμένων που αλλάζουν με την πάροδο του χρόνου, όπως ένα παγκόσμιο πλαίσιο, πλαίσιο βάσης, πλαίσιο λαβής, πλαίσιο κεφαλής κ.λπ. Το tf παρακολουθεί όλα αυτά τα πλαίσια με την πάροδο του χρόνου και επιτρέπει στον χρήστη να δει που βρίσκεται το κάθε πλαίσιο σε σχέση με ένα άλλο, ή την τρέχουσα θέση του πλαισίου ή του ρομπότ σε σχέση με τον χάρτη.

Στην παρακάτω εικόνα (Εικόνα 10) φαίνεται ένα γενικό σχέδιο μετασχηματισμών σε μορφή δέντρου (transformations tree) :



Εικ. 10 (Ένα δέντρο μετασχηματισμών «Transformations Tree»)

Βλέποντας την εικόνα των μετασχηματισμών φαίνεται ότι στη βάση του δέντρου υπάρχει το πλαίσιο του χάρτη (map) το οποίο συνδέεται με την εκκίνηση της υπηρεσίας (start_of_service) την οποία καλείται να ολοκληρώσει το ρομποτικό όχημα (device). Έπειτα, φαίνονται στο δέντρο όλα τα πλαίσια των αισθητήρων που χρησιμοποιεί το ρομπότ (imu, camera_depth, camera_fisheye, camera_color) καθώς και τα "αποτυπώματα" του ρομπότ (base_footprint). Τέλος, από τον αισθητήρα camera_depth υπάρχει το πλαίσιο laser. Πέρα από την σειρά των πλαισίων που φαίνονται μέσα από το δέντρο, φαίνεται επίσης το ποιος κόμβος στέλνει τις πληροφορίες μετασχηματισμού, τη χρονική διάρκεια του κάθε πλαισίου, αλλά και τη συχνότητα αποστολής μηνυμάτων του κάθε πλαισίου.

(Ενότητα 2.4.γ) Εισαγωγή στο RQT

Το RQT (ROS Qt) [12] είναι ένα πακέτο λογισμικού του ROS που παρέχει ένα γραφικό περιβάλλον χρήστη (GUI) για την ανάπτυξη, την εκτέλεση και τον έλεγχο ρομποτικών εφαρμογών. Βασίζεται στο Qt framework, ένα εργαλείο ανάπτυξης λογισμικού με γραφική διεπαφή χρήστη.

Το RQT παρέχει πολλά πρόσθετα plugins που επιτρέπουν την οπτικοποίηση, τον έλεγχο και την ανάλυση των δεδομένων που σχετίζονται με τα ρομπότ. Μερικές από τις λειτουργίες του περιλαμβάνουν:

- **rqt_graph:** Απεικονίζει γραφικά το γράφο των κόμβων (nodes) και των θεμάτων (topics) του ROS, προσφέροντας μια οπτική αναπαράσταση της δομής του ρομποτικού συστήματος.
- **rqt_console:** Εμφανίζει τα μηνύματα καταγραφής (log messages) που παράγονται από τους κόμβους του ROS, δίνοντας τη δυνατότητα παρακολούθησης των γεγονότων και την αντιμετώπιση προβλημάτων.
- **rqt_plot:** Επιτρέπει την απεικόνιση και την ανάλυση δεδομένων αισθητήρων ή άλλων μετρήσεων σε μορφή γραφημάτων.
- **rqt_bag:** Επιτρέπει την αναπαραγωγή, την επεξεργασία και την ανάλυση των εγγραφών (recordings) του ROS, γνωστών και ως "τσάντες" (bags), που περιέχουν δεδομένα αισθητήρων και άλλες πληροφορίες του ρομπότ.
-

Το RQT παρέχει ένα εύχρηστο περιβάλλον για την ανάπτυξη και την αποσφαλμάτωση ρομποτικών εφαρμογών, παρέχοντας έναν οπτικό τρόπο αλληλεπίδρασης με το ROS.

(Ενότητα 2.4.ε) Εισαγωγή στο Rviz

Το rviz [13] είναι ένα από τα βασικά εργαλεία που παρέχει το ROS (Robot Operating System) για την απεικόνιση και τον οπτικό έλεγχο ρομποτικών δεδομένων. Είναι ένα περιβάλλον που επιτρέπει την οπτικοποίηση των αισθητήρων, των αντικειμένων και των δεδομένων του ρομπότ, καθώς και τον έλεγχο και την αλληλεπίδραση με αυτά.

Οι βασικές λειτουργίες του rviz περιλαμβάνουν:

Οπτικοποίηση αισθητήρων: Μπορείτε να εμφανίσετε δεδομένα από διάφορους αισθητήρες όπως κάμερες, αισθητήρες βάθους, λέιζερ και πυξίδες. Τα δεδομένα αυτά εμφανίζονται ως αναπαραστάσεις του περιβάλλοντος, και μπορούν να διαμορφωθούν ανάλογα τις ανάγκες του χρήστη.

- Οπτικοποίηση χάρτη: Μπορεί να εμφανιστεί ο χάρτης του περιβάλλοντος που έχει δημιουργηθεί από τεχνικές όπως το SLAM (Simultaneous Localization and Mapping). Ο χάρτης εμφανίζεται ως αναπαράσταση του περιβάλλοντος με εμπόδια, διαδρόμους και άλλα στοιχεία.

Οπτικοποίηση ρομποτικών μοντέλων: Εισάγονται τα μοντέλα των ρομπότ ή και των αντικειμένων που αλληλεπιδρούν με αυτά, και να γίνει έλεγχος της θέσης, του προσανατολισμού, και της κίνησης των μοντέλων.

- Έλεγχος του ρομπότ : Μέσω του rviz μπορούν να δοθούν εντολές κίνησης σε ένα ρομπότ και να παρακολουθείτε η κίνηση του στο περιβάλλον. Υπάρχει επίσης, η δυνατότητα διαμόρφωσης της ταχύτητας και του ελέγχου του ρομπότ.

Συνολικά, το rviz είναι ένα ισχυρό εργαλείο για την οπτικοποίηση και τον έλεγχο ρομποτικών δεδομένων και μοντέλων. Χρησιμοποιείται ευρέως στον χώρο της ρομποτικής για την ανάπτυξη, την επίδειξη και την αξιολόγηση ρομποτικών εφαρμογών.

(Υποκεφάλαιο 2.5) Blue Rov 2 και ROS

(Ενότητα 2.5.a) Πακέτο Σύνδεσης Blue Rov 2 σε ROS

Για τη σωστή σύνδεση του υποβρυχίου με το ROS χρησιμοποιήθηκε το πακέτο «bluerov_ros_playground» [14] το οποίο είναι ένα πακέτο ανοικτού κώδικα και δημιουργήθηκε για το υποβρύχιο ρομποτικό όχημα Blue Rov 2. Το πακέτο αυτό, παρέχει τη δυνατότητα για τον έλεγχο, την ανάπτυξη και την προσομοίωση του υποβρυχίου σε περιβάλλον ROS, και δημιουργήθηκε από τον Patrick José Pereira [15] ως βοήθεια ενσωμάτωσης του Blue Rov 2 με το ROS.

Με το πακέτο «bluerov_ros_playground» ελέγχεται η κίνηση του υποβρυχίου ρομποτικού οχήματος, όπως και η λήψη δεδομένων από τους αισθητήρες και η προβολή της εμπρόσθιας κάμερας. Επίσης, υποστηρίζεται η χρήση του ρομπότ σε προσομοίωση, όπου αυτό επέτρεψε την αρχική δοκιμή και εκτέλεση του λογισμικού χωρίς την πραγματική παρουσία του ρομπότ.

(Ενότητα 2.5.b) Λειτουργία Του Πακέτου «bluerov_ros_playground»

Το πακέτο είναι πλήρως λειτουργικό για μια πληθώρα υποβρύχιων ρομποτικών οχημάτων και όχι μόνο για το Blue Rov 2. Παρόλα αυτά, είναι σημαντικό να γίνεται η σωστή αξιοποίηση του πακέτου ανοικτού κώδικα, ώστε να μην υπάρχει βλάβη ή καταστροφή του υποβρύχιου ρομποτικού οχήματος. Γι' αυτό, προτείνεται η αρχική χρήση του πακέτου «bluerov_ros_playground» να λειτουργήσει πειραματικά σε προσομοίωση για την αποφυγή βλαβών ή δυσλειτουργιών του πακέτου και την συνήθεια λειτουργίας του ρομπότ από τον χειριστή.

Για την λειτουργία χρειάζονται:

- ένα λειτουργικό περιβάλλον για το ROS,
- η σωστή εγκατάσταση του πακέτου «bluerov_ros_playground»
- Το πακέτο MAVROS [16] για τη σύνδεση πληροφοριών από το ρομπότ στο ROS
- Το πακέτο freefloating_gazebo [17] για τη σύνδεση προσομοίωσης (υποβρυχίου και του ROS)

(Ενότητα 2.5.γ) Χρήση Του Πακέτου «bluerov_ros_playground»

Για την λειτουργία του πακέτου ανοικτού κώδικα «bluerov_ros_playground» θα πρέπει το υποβρύχιο ρομπότ blue rov 2 να είναι συνδεδεμένο με τον ηλεκτρονικό υπολογιστή και να είναι πλήρως λειτουργικό. Έπειτα, χρησιμοποιώντας την παρακάτω εντολή ο χρήστης μπορεί να χρησιμοποιήσει πλήρως το υποβρύχιο όχημα μέσω τηλεχειριστηρίου όπως και να βλέπει πληροφορίες για την κατάσταση του (μπαταρία, πίεση του νερού, οδομετρία κ.α.):

```
$ roslaunch bluerov_ros_playground bluerov_node.launch
```

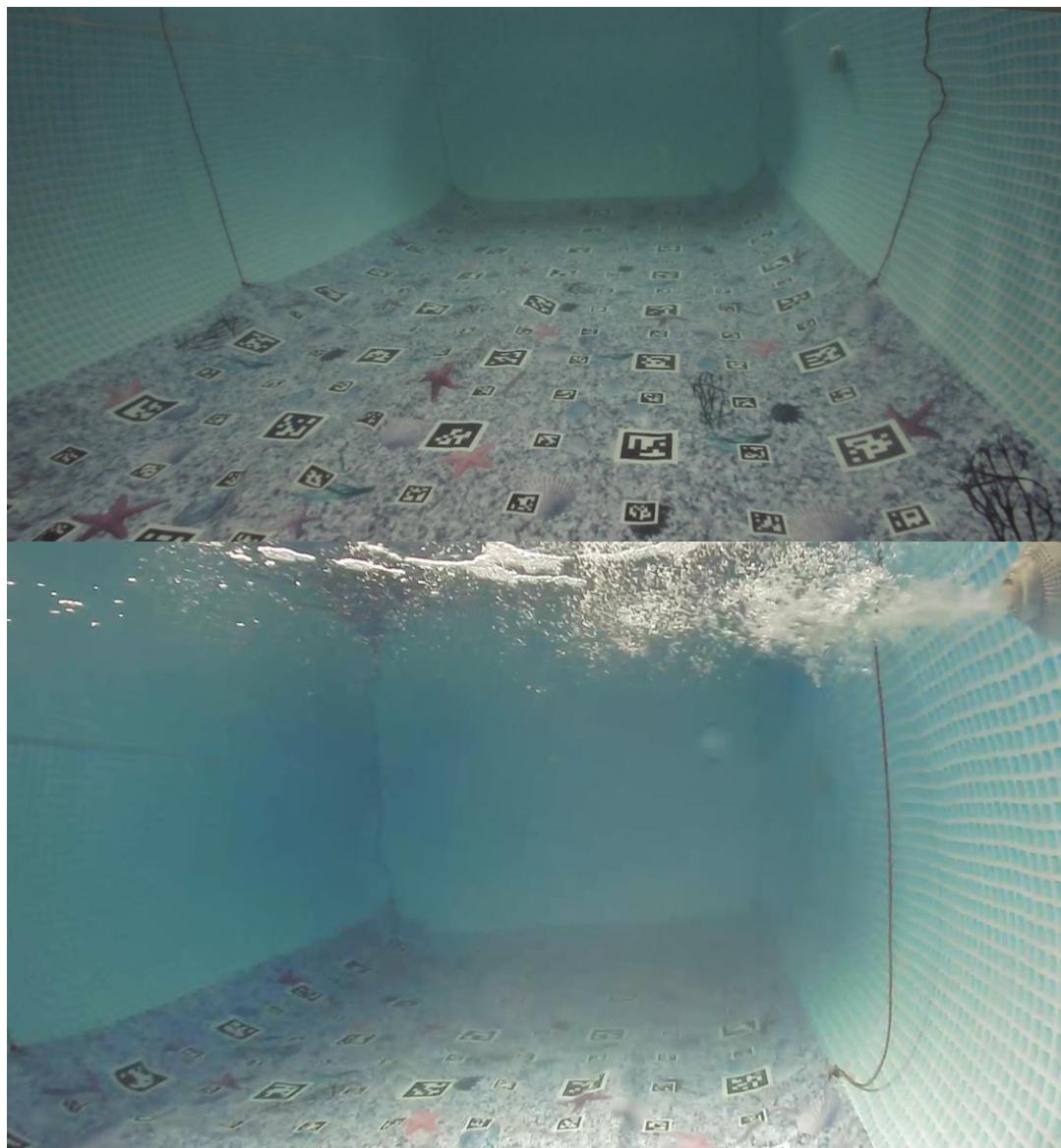
Στη συνέχεια εκτελώντας την εντολή «*rostopic list*» φαίνεται η λίστα με όλες τις απαραίτητες πληροφορίες (Εικόνα 11):

- /BlueRov2/arm - std_msgs/Bool
- /BlueRov2/battery - sensor_msgs/BatteryState
- /BlueRov2/camera/image_raw - sensor_msgs/Image
- /BlueRov2/imu/data - sensor_msgs/Imu
- /BlueRov2/mode/set - std_msgs/String
- /BlueRov2/odometry - nav_msgs/Odometry
- /BlueRov2/rc_channel1/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel2/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel3/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel4/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel5/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel6/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel7/set_pwm - std_msgs/UInt16
- /BlueRov2/rc_channel8/set_pwm - std_msgs/UInt16
- /BlueRov2/servo1/set_pwm - std_msgs/UInt16
- /BlueRov2/servo2/set_pwm - std_msgs/UInt16
- /BlueRov2/servo3/set_pwm - std_msgs/UInt16
- /BlueRov2/servo4/set_pwm - std_msgs/UInt16
- /BlueRov2/servo5/set_pwm - std_msgs/UInt16
- /BlueRov2/servo6/set_pwm - std_msgs/UInt16
- /BlueRov2/servo7/set_pwm - std_msgs/UInt16
- /BlueRov2/servo8/set_pwm - std_msgs/UInt16
- /BlueRov2/setpoint_velocity/cmd_vel - geometry_msgs/TwistStamped
- /BlueRov2/state - std_msgs/String
- /rosout - rosgraph_msgs/Log
- /rosout_agg - rosgraph_msgs/Log

Εικ. 11 (Πληροφορίες για το υποβρύχιο Blue Rov 2 μέσω του ROS)

Κι εκτελώντας την παρακάτω εντολή γίνεται η προβολή της εμπρόσθιας κάμερας όπως φαίνεται στην Εικόνα 12:

```
$ rosrun image_view image_view image:=/BlueRov2/camera/image_raw
```



Εικ. 12 (Προβολή εικόνας εμπρόσθιας κάμερας Blue Rov 2 μέσα στη πισίνα)

ΚΕΦΑΛΑΙΟ 3 Simultaneous Localization and Mapping

(Υποκεφάλαιο 3.1) Visual SLAM

Η μέθοδος Visual SLAM [18] αναφέρεται στην χωροθέτηση και χαρτογράφηση με τη χρήση μιας ή περισσότερων καμερών, η οποία είναι προτιμητέα από τη χρήση άλλων αισθητήρων χάρη στο ελαφρύτερο βάρος, το χαμηλότερο κόστος και την πλουσιότερη παρουσίαση του περιβάλλοντος.

Η γνώση για το Visual SLAM είναι πολύτιμη για ρομποτικές εφαρμογές όπως στην αυτόνομη οδήγηση αυτοκινήτων, στην υποβρύχια ρομποτική, στην αυτόνομη πτήση drones και σε άλλες εφαρμογές που χρειάζεται αλληλεπίδραση με το περιβάλλον τους. Συνδυάζει την οπτική αντίληψη και τον υπολογισμό της θέσης για να ανιχνεύει τις κινήσεις του ρομπότ και να υπολογίζει την τοποθεσία του και να δημιουργεί τον χάρτη. Η βασική ιδέα του Visual SLAM είναι ότι το ρομπότ χρησιμοποιεί μια ή περισσότερες κάμερες για να ανιχνεύσει τα χαρακτηριστικά του περιβάλλοντος, όπως γωνίες, γραμμές και σημεία ενδιαφέροντος. Αυτά τα χαρακτηριστικά χρησιμοποιούνται για την εκτίμηση της θέσης και της κίνησης του ρομπότ, καθώς και για τη δημιουργία ενός χάρτη του περιβάλλοντος.

Η διαδικασία περιλαμβάνει την εξαγωγή χαρακτηριστικών από τις εικόνες, την σύγκριση των εικόνων για διαφορετικά χαρακτηριστικά κάθε στιγμάτου της εικόνας, την εκτίμηση της θέσης και της κίνησης του ρομπότ και τέλος την ενσωμάτωση όλων των πληροφοριών σε έναν χάρτη. Η διαδικασία επαναλαμβάνεται συνεχώς όσο το ρομπότ κινείται στον χώρο και λειτουργεί αυτόματα.

Υπάρχουν πολλές έρευνες και μελέτες για το Visual SLAM [19], όπου εξετάζουν διάφορες τεχνικές, συγκρίνοντας την απόδοση και την ακρίβεια. Σε μερικές από τις μελέτες, χρησιμοποιήθηκαν πολλαπλές κάμερες σε πολλούς πράκτορες για την ταχύτερη δημιουργία του χάρτη και τη λειτουργία σε στατικά και δυναμικά περιβάλλοντα, αν και αυτή η μέθοδος λειτουργεί σε πραγματικό χρόνο (real-time), απαιτεί μεγάλη υπολογιστική δύναμη. Άλλη έρευνα ωστόσο, με τη χρήση κάμερας σε συνδυασμό με αισθητήρες έδειξε πως το σύστημα τους μπορεί να διαχειριστεί συνθήκες όπου ήταν αδύνατο για απλά συμβατικά μοντέλα SLAM, με μειονέκτημα τον συγχρονισμό των υπολοίπων αισθητήρων και της κάμερας.

(Υποκεφάλαιο 3.2) Fiducial Slam

(Ενότητα 3.2.a) Εισαγωγή στα Fiducial Markers

Οι εφαρμογές που χρησιμοποιούν δείκτες κωδικών (γραμμωτοί κώδικες, κωδικοί QR, Aruco Markers κ.λπ.) έχουν γίνει πανταχού παρόντες μετά τη δημοτικότητα τέτοιων κωδικών και την ανάγκη μας να συνδέσουμε φυσικά αντικείμενα με αντίστοιχα ψηφιακά αναγνωριστικά. Μαθαίνοντας πώς να εφαρμοστεί ένας από αυτούς τους αλγόριθμους, αποκτάται μια βαθύτερη κατανόηση των παραμέτρων που σχετίζονται με το σχεδιασμό αυτών των markers, των αλγορίθμων που τους ανιχνεύουν, αλλά και τη χρησιμότητα τους σε ρομποτικές εφαρμογές.

Μια εκτίμηση θέσης του ρομπότ που βασίζεται σε δεδομένα οπτικών αισθητήρων είναι ένα βασικό χαρακτηριστικό σε πολλές ρομποτικές εφαρμογές, όπως ο εντοπισμός [20], η πλοήγηση του ρομπότ, το SLAM και άλλες. Αυτή η διαδικασία βασίζεται στην εύρεση αντιστοιχιών μεταξύ σημείων χαρακτηριστικών σε πραγματικό περιβάλλον και της προβολής τους σε μια δισδιάστατη εικόνα (μια κάμερα). Ως εκ τούτου, τα συνθετικά Markers χρησιμοποιούνται συνήθως για την εξαγωγή κατάλληλων σημείων χαρακτηριστικών [21].

Το προκαθορισμένο μέγεθος και οι αναλογίες ενός Fiducial Marker επιτρέπουν την εξαγωγή μιας σχετικής περιστροφής της κάμερας και την εκτίμηση της απόστασης από την κάμερα ως προς τον Marker. Μία από τις πιο δημοφιλείς προσεγγίσεις είναι η χρήση δυαδικών δεικτών τετράγωνου σχήματος [22] (παραδείγματα δεικτών Aruco φαίνονται στην Εικόνα 13) [23].

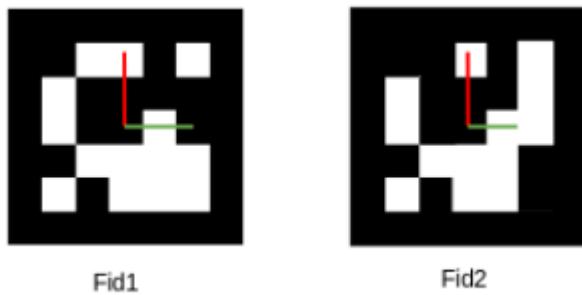


Εικ. 13 (Μια βιβλιοθήκη οπτικών οροσήμων «Fiducial Markers»)

Ένα Fiducial Marker τοποθετείται σε μια επιφάνεια. Στη συνέχεια, κατά τη διάρκεια μιας διαδικασίας κίνησης και εντοπισμού, ένα AUV πρέπει να ανιχνεύει συνεχώς έναν δείκτη (Marker) και να κινείται ανάλογα. Έτσι, ο δείκτης πρέπει να ανιχνεύεται καθ' όλη τη διάρκεια της διαδικασίας χαρτογράφησης και χωροθέτησης, πράγμα που σημαίνει ότι θα πρέπει να παραμένει όλη την ώρα σε γωνία θέασης της κάμερας. Αυτή η απαίτηση δεν είναι δύσκολο να ικανοποιηθεί με τη χρήση μιας ομάδας από Fiducial Markers. Έτσι, ένα μεγάλο σύνολο από δείκτες είναι ευκολότερο να χαρτογραφηθούν, αλλά από την άλλη πλευρά, ένα πολύ μεγάλο σύνολο δεικτών χρειάζεται αρκετά παραπάνω χρόνο για να χαρτογραφηθεί.

(Ενότητα 3.2.8) Πακέτο Fiducials σε ROS

Σε αυτήν την εργασία χρησιμοποιήθηκε το πακέτο Fiducials [24]. Αυτό το πακέτο παρέχει ένα σύστημα που επιτρέπει σε ένα ρομπότ να προσδιορίζει τη θέση και τον προσανατολισμό του παρακολουθώντας έναν αριθμό από Fiducial Markers που είναι τοποθετημένοι στο περιβάλλον του ρομπότ. Αρχικά, καθορίζεται η θέση ενός δείκτη ή αυτό καθορίζεται αυτόματα. Μετά, δημιουργείται ένας χάρτης (με τη μορφή αρχείου 6 DOF θέσεων) παρατηρώντας ζεύγη δεικτών (Markers) (Εικόνα 14) και προσδιορίζοντας τη τοποθεσία και την περιστροφή μεταξύ τους.



Εικ. 14 (Προβολή δείκτη θέσεως και προσανατολισμού ενός Fiducial Marker)

(Ενότητα 3.2.γ) Λειτουργία του πακέτου Fiducials

Το σύστημα εντοπισμού της Ubiquity Robotics χρησιμοποιεί έναν αριθμό δεικτών γνωστού μεγέθους για να προσδιορίσει τη θέση του ρομπότ. Η ανίχνευση των δεικτών γίνεται από τον κόμβο (node) aruco_detect [25]. Για κάθε δείκτη που είναι ορατός στην εικόνα, παράγεται ένα σύνολο κορυφών σε συντεταγμένες εικόνας. Δεδομένου ότι οι παράμετροι της κάμερας και το μέγεθος του δείκτη είναι γνωστά, μπορεί να εκτιμηθεί η θέση του δείκτη σε σχέση με την κάμερα.

Οι συντεταγμένες της εικόνας (x, y) κάθε κορυφής αντιστοιχούν σε μια ακτίνα από την κάμερα. Ο κώδικας εκτίμησης θέσης λύνει ένα σύνολο γραμμικών εξισώσεων για τον προσδιορισμό των συντεταγμένων του κόσμου (X, Y, Z) για καθεμία από τις κορυφές. Από αυτό, λαμβάνουμε τον μετασχηματισμό του συστήματος συντεταγμένων του fiducial στο σύστημα συντεταγμένων της κάμερας T_{fid_cam} . Αυτό αντιπροσωπεύει τη θέση του δείκτη στο σύστημα συντεταγμένων της κάμερας. Στο παρακάτω διάγραμμα, φαίνονται δύο fiducials, fid1 και fid2. Εάν το Fid1 βρίσκεται σε μια γνωστή θέση στον κόσμο, T_{map_fid1} και γνωρίζουμε τους μετασχηματισμούς δείκτη σε κάμερα και για τους δύο δείκτες, μπορούμε να υπολογίσουμε τη στάση του Fid2 ως εξής:

$$T_{map_fid2} = T_{map_fid1} * T_{cam_fid2} * T_{fid1_cam} \quad [26]$$

Με αυτόν τον τρόπο, ο χάρτης δημιουργείται καθώς παρατηρούνται περισσότερα ζεύγη δεικτών. Πολλαπλές παρατηρήσεις συνδυάζονται με στάθμιση, για να παραχθεί μια εκτίμηση της θέσης κάθε δείκτη, αλλά και του ρομπότ.

(Ενότητα 3.2.δ) Χρήση του πακέτου Fiducials

Για την χρήση του πακέτου Fiducials ROS θα πρέπει να υπάρχει:

- ένα λειτουργικό περιβάλλον για το ROS,
- να εγκατασταθούν όλα τα απαραίτητα πακέτα
- η χρήση κάμερας
- η θέση και ο προσανατολισμός της κάμερας σε σχέση με το ρομπότ (στατικός μετασχηματισμός)

Θα χρειαστούν επίσης οι παρακάτω εντολές:

Για την δημιουργία των Markers:

```
rosrun aruco_detect create_markers.py 100 112 fiducials.pdf
```

Μόλις εκτυπωθούν, μπορούν να επικολληθούν στο περιβάλλον. Δεν χρειάζεται να τοποθετηθούν σε κάποιο συγκεκριμένο μοτίβο, αλλά η πυκνότητα πρέπει να είναι τέτοια ώστε δύο ή περισσότερα Markers να μπορούν να φαίνονται από την κάμερα στο ρομπότ, ώστε να μπορεί να κατασκευαστεί ο χάρτης.

Στη συνέχεια, θα πρέπει να λειτουργούν δύο nodes, το aruco_detect το οποίο διαχειρίζεται τον εντοπισμό των δεικτών και το fiducial_slam το οποίο συνδυάζει τις εκτιμήσεις θέσεως και χτίζει τον χάρτη και κάνει μια εκτίμηση της θέσης του ρομπότ. Ο χάρτης δημιουργείται σε μορφή αρχείου (Text document) και αποθηκεύεται αυτόματα.

Υπάρχουν έτοιμα αρχεία και για τους δύο κόμβους:

```
roslaunch aruco_detect aruco_detect.launch  
roslaunch fiducial_slam fiducial_slam.launch
```

Υπάρχει επίσης έτοιμο αρχείο οπτικοποίησης του χάρτη στο Rviz. Με αυτό θα εμφανιστεί ο χάρτης καθώς χτίζεται όπως και η τρέχουσα προβολή της κάμερας:

```
roslaunch fiducial_slam fiducial_rviz.launch
```

(Ενότητα 3.2.ε) Κώδικας και Παράμετροι

Στην παρούσα εργασία, το πακέτο Fiducials είχε μερικές από τις πιο σημαντικές αλλαγές στις παραμέτρους του κώδικα, γιατί τα χαρακτηριστικά της κάμερας, του ρομπότ, και το μέγεθος των δεικτών παίζουν σημαντικό ρόλο για σωστή και αξιόπιστη λειτουργία του αλγορίθμου.

Το αρχείο εκκίνησης του κόμβου *aruco_detect.launch* για την αναγνώριση των δεικτών από την κάμερα (Εικόνα 15) :

```

<!-- Run the aruco_detect node
-->
<launch>
  <!-- namespace for camera input /bluerov2/bluerov2/cameraback/camera_image-->
  <arg name="camera" default="/usb_cam"/>
  <arg name="image" default="/image_raw"/>
  <arg name="transport" default="compressed"/>
  <arg name="fiducial_len" default="0.1"/>
  <arg name="dictionary" default="3"/>
  <arg name="do_pose_estimation" default="true"/>
  <!-- If vis_msgs set to true, pose estimation will be published with ROS standard vision_msgs -->
  <arg name="vis_msgs" default="false"/>
  <arg name="ignore_fiducials" default="" />
  <arg name="fiducial_len_override" default="" />
  <arg name="verbose" default="false"/>

  <node pkg="aruco_detect" name="aruco_detect"
    type="aruco_detect" output="screen" respawn="false">
    <param name="image_transport" value="$(arg transport)"/>
    <param name="publish_images" value="true" />
    <param name="fiducial_len" value="$(arg fiducial_len)"/>
    <param name="dictionary" value="$(arg dictionary)"/>
    <param name="do_pose_estimation" value="$(arg do_pose_estimation)"/>
    <param name="vis_msgs" value="$(arg vis_msgs)"/>
    <param name="ignore_fiducials" value="$(arg ignore_fiducials)"/>
    <param name="fiducial_len_override" value="$(arg fiducial_len_override)"/>
    <param name="verbose" value="$(arg verbose)"/>
    <remap from="camera/compressed"
      to="$(arg camera)/$(arg image)/$(arg transport)"/>
    <remap from="camera_info" to="$(arg camera)/camera_info"/>
  </node>
</launch>

```

Εικ. 15 (Αρχείο εκκίνησης κόμβου aruco_detect)

Όπως φαίνεται και στο παραπάνω αρχείο εκκίνησης του κόμβου πρέπει να ορίσουμε τις παραμέτρους:

- Τις πληροφορίες της κάμερας (`<arg name="camera" default="/usb_cam"/>`)
- Της εικόνας (`<arg name="image" default="/image_raw"/>`)
- Του μεγέθους των δεικτών (`<arg name="fiducial_len" default="0.1"/>`)
- Τη σωστή βιβλιοθήκη δεικτών όπου στην προκειμένη περίπτωση χρησιμοποιήθηκε η βιβλιοθήκη 3 (`<arg name="dictionary" default="3"/>`)

Μετά, το αρχείο εκκίνησης `fiducial_slam.launch` για την δημιουργία χάρτη και εκτίμηση της θέσης του ρομπότ (Εικόνα 16) :

```

fiducial_slam.launch
~/catkin_ws/src/fiducials/fiducial_slam/launch

<!--
  Run the fiducial_slam node
-->
<launch>
  <arg name="camera" default="head_camera"/>
  <arg name="map_frame" default="map"/>
  <arg name="odom_frame" default="odom"/>
  <arg name="base_frame" default="base_link_frd"/>
  <arg name="publish_tf" default="true"/>
  <arg name="tf_publish_interval" default="1.0"/>
  <arg name="future_date_transforms" default="0.1"/>
  <arg name="publish_6dof_pose" default="false"/>
  <arg name="systematic_error" default="0.01"/>
  <arg name="covariance_diagonal" default="True"/>

  <node type="fiducial_slam" pkg="fiducial_slam" output="screen" name="fiducial_slam">
    <param name="map_file" value="$(env HOME)/.ros/slam/map.txt" />
    <param name="map_frame" value="$(arg map_frame)" />
    <param name="odom_frame" value="$(arg odom_frame)" />
    <param name="base_frame" value="$(arg base_frame)" />
    <param name="publish_tf" value="$(arg publish_tf)" />
    <param name="tf_publish_interval" value="$(arg tf_publish_interval)" />
    <param name="future_date_transforms" value="$(arg future_date_transforms)" />
    <param name="publish_6dof_pose" value="$(arg publish_6dof_pose)" />
    <param name="sum_error_in_quadrature" value="true"/>
    <rosparam param="covariance_diagonal" subst_value="True">$(arg covariance_diagonal)</rosparam>
    <remap from="/camera_info" to="$(arg camera)/camera_info"/>
  </node>
</launch>

```

Εικ. 16 (Αρχείο εκκίνησης κόμβου fiducial_slam)

Το οποίο χρειάστηκε αλλαγές ως προς:

- Τη κάμερα (`<arg name="camera" default="head_camera"/>`)
- Την αποθήκευση του χάρτη (`<arg name="map_frame" default="map"/>`)
- Την οδομετρία (`<arg name="odom_frame" default="odom"/>`)
- Τη βάση του ρομπότ (`<arg name="base_frame" default="base_link_frd"/>`)

Και τέλος, το αρχείο εκκίνησης Fiducial_rviz.launch για την οπτικοποίηση του χάρτη όσο ακόμα κατασκευάζεται αλλά και τον προσανατολισμό του ρομπότ (Εικόνα 17):

```

fiducial_rviz.launch
~/catkin_ws/src/Fiducials/fiducial_slam/launch

<?xml version="1.0"?>

<launch>
  <node pkg="rviz" type="rviz" name="$(anon rviz)" args="-d $(find fiducial_slam)/fiducials.rviz"/>
    <arg name="model" default="$(find bluerov2_description)/robots/bluerov2_default.xacro"/>
    <arg name="rvizconfig" default="$(find bluerov_ros_playground)/model/model_preset.rviz" />
    <!-- <node name="rviz" pkg="rviz" type="rviz" required="true" args="-d $(arg rvizconfig)"/>-->
    <param name="robot_description" command="$(find xacro)/xacro.py --inorder $(arg model)"/>

    <param name="publish_frequency" type="double" value="10.0" />
  |
</launch>

```

XML ▾ Tab Width: 8 ▾ Ln 15, Col 5 ▾ INS

Εικ. 17. (Αρχείο εκκίνησης κόμβου fiducial_rviz)

Στο οποίο προστέθηκε η επιλογή οπτικοποίησης του ρομπότ από το Rviz κάνοντας ευκολότερη την κατανόηση του προσανατολισμού του υποβρυχίου ρομπότ. Για την οπτικοποίηση χρησιμοποιήθηκε έτοιμο 3D αρχείο του υποβρυχίου το οποίο βρίσκεται στο πακέτο bluerov_ros_playground.

Με τις εξής εντολές:

```

<arg name="model" default="$(find bluerov2_description)/robots/bluerov2_default.xacro"/>

<arg name="rvizconfig" default="$(find bluerov_ros_playground)/model/model_preset.rviz"/>

```

ΚΕΦΑΛΑΙΟ 4 Αποτελέσματα

(Υποκεφάλαιο 4.1) Πείραμα σε Προσομοίωση

Για τη σωστή υλοποίηση του πειράματος και την αποφυγή βλαβών ή και την καταστροφή του υποβρύχιου ρομποτικού οχήματος υλοποιήθηκε ο αλγόριθμος χωροθέτησης και χαρτογράφησης σε προσομοίωση. Δηλαδή, χρησιμοποιώντας την εφαρμογή Gazebo [27], σε σύνδεση με το ROS, και την χρήση πακέτων προσομοίωσης ενός θαλάσσιου κόσμου, το υποβρύχιο ρομπότ Blue Rover 2 κατάφερε να ανιχνεύει τους δείκτες, να υπολογίζει την θέση του ως προς αυτούς, και να δημιουργεί έναν χάρτη.

(Ενότητα 4.1.α) Εισαγωγή στο Gazebo

Πιο συγκεκριμένα, το Gazebo είναι ένας προσομοιωτής ρομποτικών περιβαλλόντων. Αποτελεί ένα ισχυρό εργαλείο που επιτρέπει τη δημιουργία, την εκτέλεση και τον έλεγχο 3D περιβαλλόντων για ρομποτικές εφαρμογές. Ο σκοπός του Gazebo είναι να παρέχει ένα ρεαλιστικό περιβάλλον για τη δοκιμή, την εκτίμηση και την ανάπτυξη ρομποτικών συστημάτων.

Οι βασικές λειτουργίες και χαρακτηριστικά του Gazebo περιλαμβάνουν:

- **Φυσική προσομοίωση:** Το Gazebo υποστηρίζει ρεαλιστική φυσική προσομοίωση για τα αντικείμενα και το περιβάλλον. Αυτό σημαίνει ότι μπορεί να προσομοιώσει τη βαρύτητα, την τριβή, την ανάγκη για αισθητήρες και άλλες φυσικές ιδιότητες των ρομπότ.
- **Δυναμικά μοντέλα ρομπότ:** Μέσω του Gazebo δημιουργούνται ακριβή μοντέλα ρομπότ με τις δομές, τις αρθρώσεις και τους αισθητήρες που τα χαρακτηρίζουν.
- **Περιβάλλοντα προσομοίωσης:** Με το Gazebo δημιουργούνται προσαρμοσμένα 3D περιβάλλοντα για την προσομοίωση των ρομποτικών εφαρμογών. Υπάρχει η δυνατότητα προσθήκης αντικειμένων, εμποδίων και άλλων στοιχείων για την αναπαράσταση πραγματικών συνθηκών.
- **Αλληλεπίδραση αντικειμένων:** Το Gazebo επιτρέπει την αλληλεπίδραση μεταξύ των αντικειμένων στο περιβάλλον.
- **Αποσφαλμάτωση:** Το Gazebo παρέχει εργαλεία αποσφαλμάτωσης για την εύρεση και την επίλυση προβλημάτων κατά την προσομοίωση.

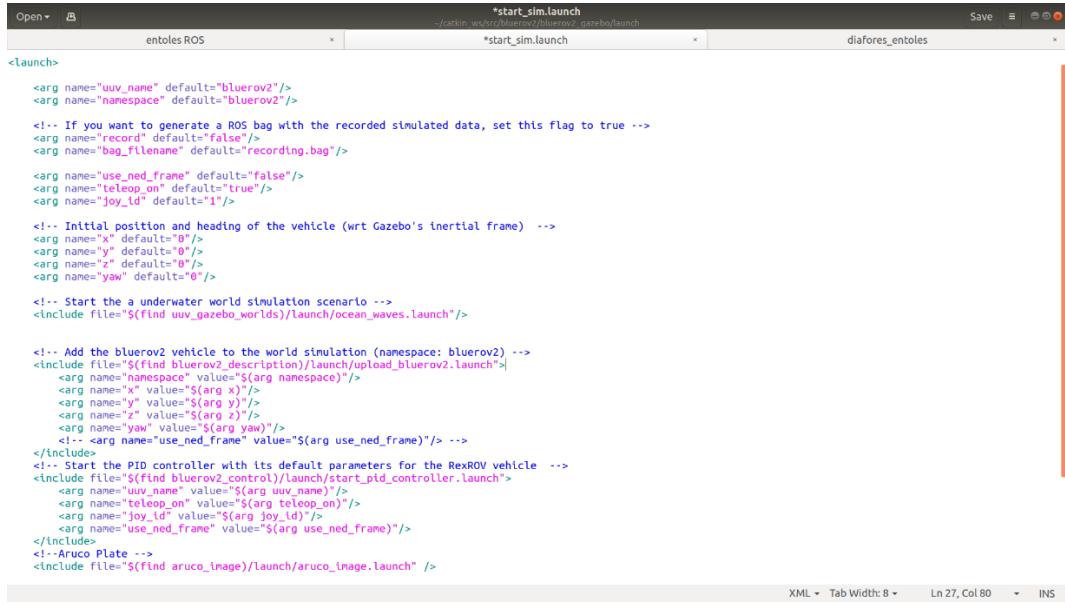
Συνολικά, το Gazebo είναι ένα ισχυρό εργαλείο που επιτρέπει την ανάπτυξη, την εκτέλεση και τον έλεγχο προηγμένων ρομποτικών συστημάτων σε εικονικά περιβάλλοντα. Χρησιμοποιείται ευρέως από ερευνητές, αναπτυξιακές ομάδες και φοιτητές για την προσομοίωση, την ανάπτυξη και την εκτίμηση των ρομποτικών εφαρμογών.

(Ενότητα 4.1.β) Διεξαγωγή πειράματος σε προσομοίωση

Γι' αυτή την εργασία χρησιμοποιήθηκε ο κόσμος προσομοίωσης ocean_waves από το πακέτο «sun_simulator» [28] και η προσομοίωση του υποβρυχίου Blue Rov 2 από το πακέτο «bluerov_ros_playground». Μετά, προστέθηκαν στον κόσμο 4 βασικοί Aruco Markers της ίδιας οικογενείας σε απόσταση 5 μέτρα κάτω από το ρομπότ.

Για την διευκόλυνση λειτουργίας της προσομοίωσης δημιουργήθηκε ένα αρχείο εκκίνησης το οποίο περιλαμβάνει τον κόσμο, το υποβρύχιο Blue Rov 2, τη λειτουργία του μέσω τηλεχειριστηρίου και τους δείκτες (Εικόνα 18, Εικόνα 19, Εικόνα 20).

```
roslaunch bluerov2_gazebo start_sim.launch
```



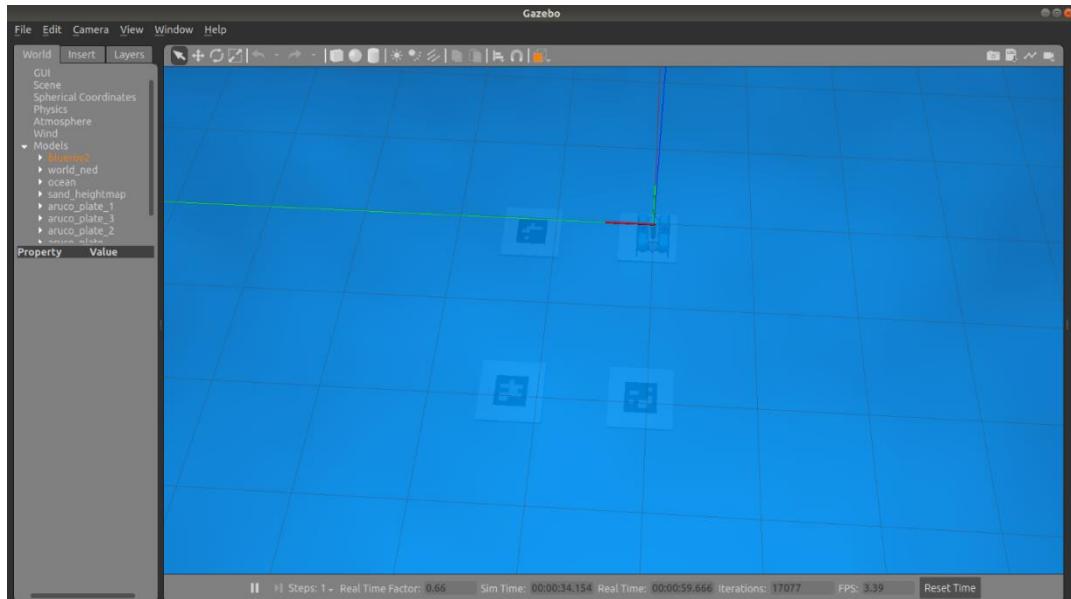
```

Open ▾  Save
start_sim.launch
entoles ROS      start_sim.launch
Save  ⌂  ⌂  ⌂
<launch>
  <arg name="uvv_name" default="bluerov2"/>
  <arg name="namespace" default="bluerov2"/>
  <!-- If you want to generate a ROS bag with the recorded simulated data, set this flag to true -->
  <arg name="record" default="false"/>
  <arg name="bag_filename" default="recording.bag"/>
  <arg name="use_ned_frame" default="false"/>
  <arg name="teleop_on" default="true"/>
  <arg name="joy_id" default="1"/>
  <!-- Initial position and heading of the vehicle (wrt Gazebo's inertial frame) -->
  <arg name="x" default="0"/>
  <arg name="y" default="0"/>
  <arg name="z" default="0"/>
  <arg name="yaw" default="0"/>
  <!-- Start the underwater world simulation scenario -->
  <include file="$(find uuv_gazebo_worlds)/launch/ocean_waves.launch"/>

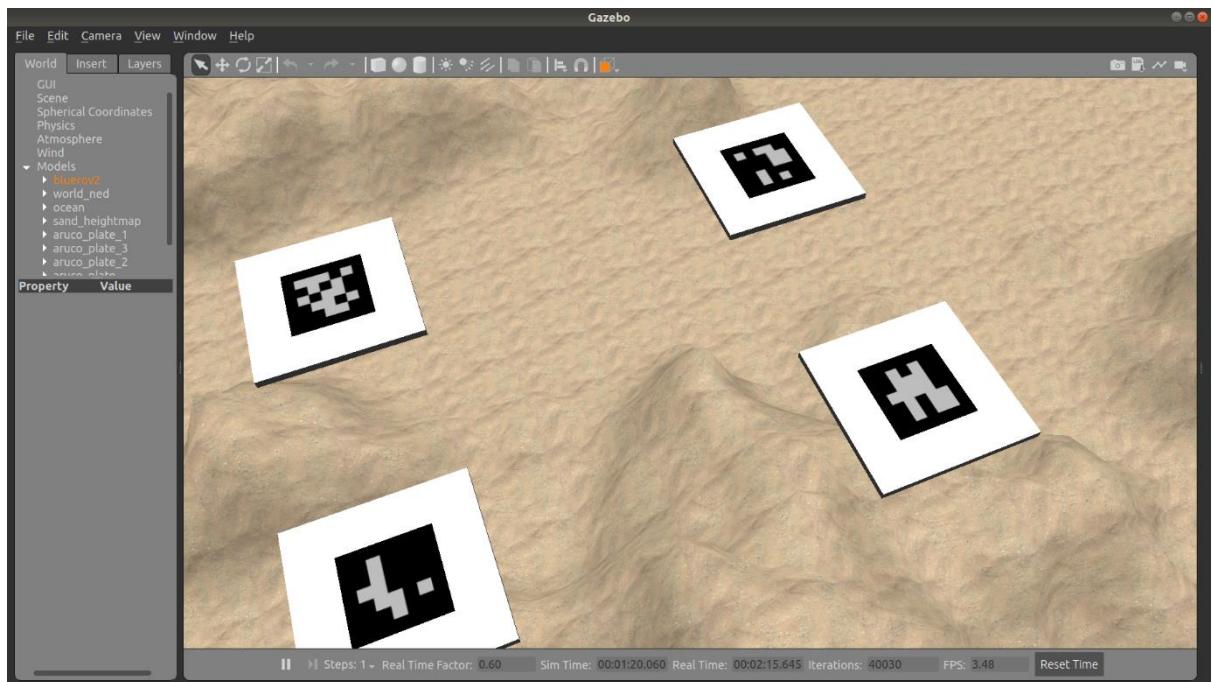
  <!-- Add the bluerov2 vehicle to the world simulation (namespace: bluerov2) -->
  <include file="$(find bluerov2_description)/launch/upload_bluerov2.launch">
    <arg name="namespace" value="$(arg namespace)"/>
    <arg name="x" value="$(arg x)"/>
    <arg name="y" value="$(arg y)"/>
    <arg name="z" value="$(arg z)"/>
    <arg name="yaw" value="$(arg yaw)"/>
    <!-- arg name="use_ned_frame" value="$(arg use_ned_frame)"/> ...
  </include>
  <!-- Start the PID controller with its default parameters for the RexROV vehicle -->
  <include file="$(find bluerov2_control)/launch/start_pid_controller.launch">
    <arg name="uvv_name" value="$(arg uvv_name)"/>
    <arg name="teleop_on" value="$(arg teleop_on)"/>
    <arg name="joy_id" value="$(arg joy_id)"/>
    <arg name="use_ned_frame" value="$(arg use_ned_frame)"/>
  </include>
  <!--Aruco Plate -->
  <include file="$(find aruco_image)/launch/aruco_image.launch" />

```

Εικ. 18 (Αρχείο εκκίνησης κόμβου προσομοίωσης)



Εικ. 19 (Υποβρύχιος κόσμος προσομοίωσης)



Εικ. 20 (Fiducial Markers του κόσμου προσσομοίωσης)

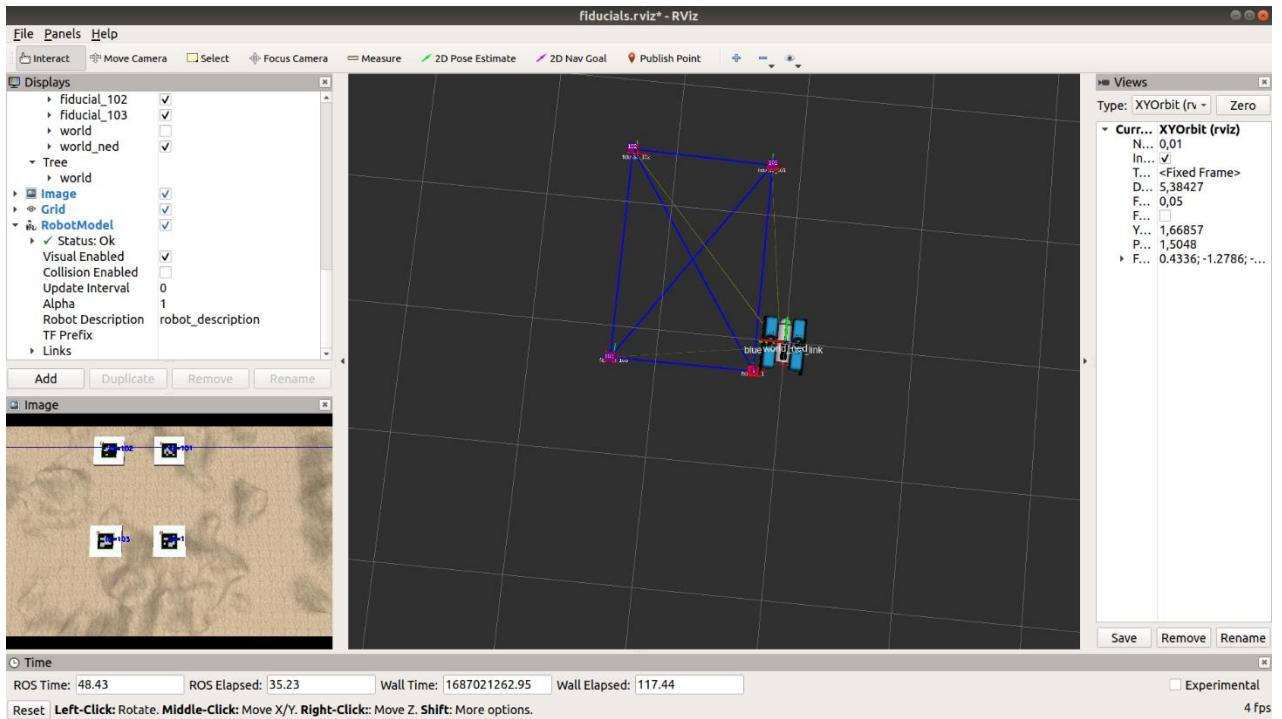
Στη συνέχεια χρησιμοποιήθηκαν τα αρχεία εκκίνησης:

- *roslaunch aruco_detect aruco_detect.launch*
- *roslaunch fiducial_slam fiducial_slam.launch*

Και τέλος η εντολή για την οπτικοποίηση του χάρτη και της κάμερας του ρομπότ μέσω rviz

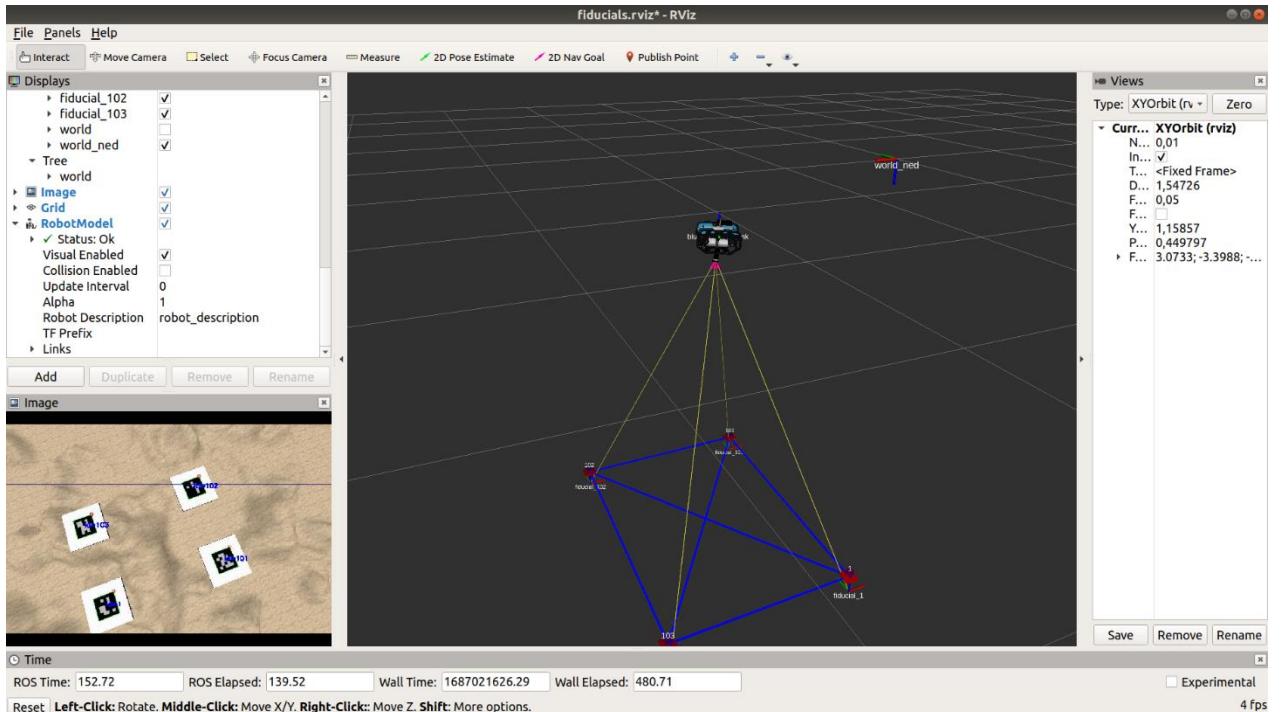
- *roslaunch fiducial_slam fiducial_slam.rviz*

Στο αρχείο φαίνεται κάτω αριστερά η κάμερα του ρομπότ μαζί με την ανίχνευση των δεικτών και στο κέντρο η χαρτογράφηση των δεικτών και η χωροθέτηση του υποβρυχίου ρομπότ (Το ρομπότ βρίσκεται πάνω από τον δείκτη 1) (Εικόνα 21).



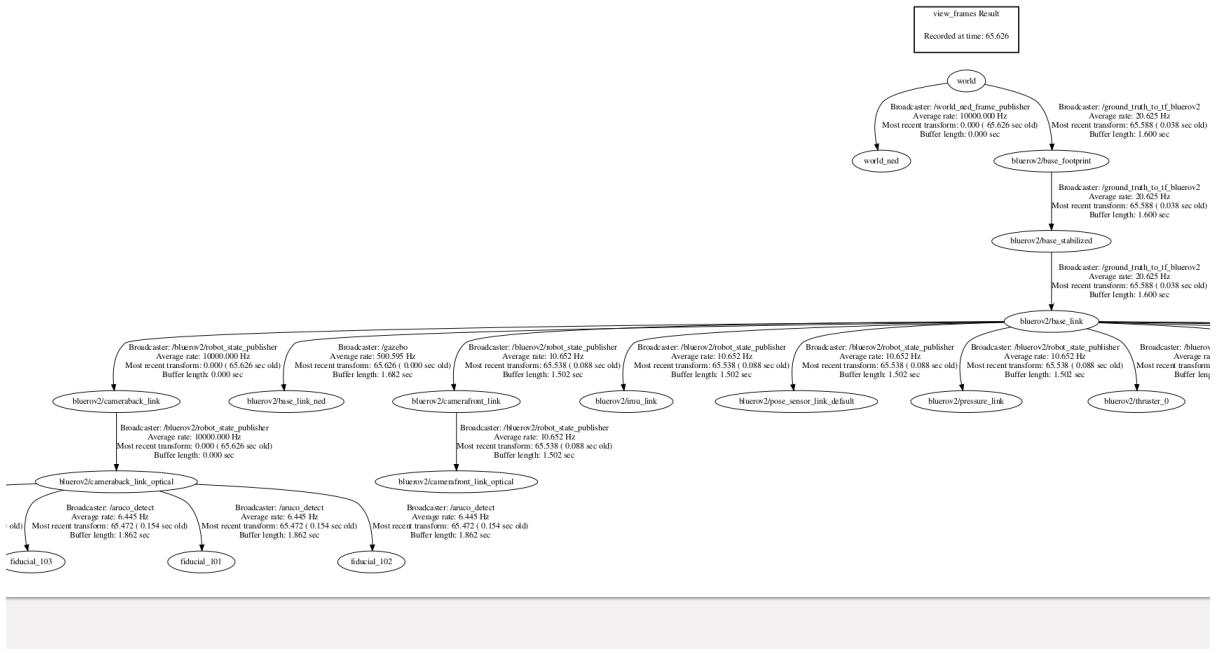
Εικ. 21 (Οπτικοποίηση της κάμερας και της χαρτογράφησης των Fiducial Markers)

Στη συνέχεια, τερματίστηκαν τα έτοιμα αρχείο *aruco_detect* και *fiducial_slam* και μετακινήθηκε το υποβρύχιο στον χώρο. Όταν, εκκινήθηκαν τα αρχεία πάλι, φαίνεται πως και το ρομπότ μετακινήθηκε στον χάρτη (Το ρομπότ πλέον βρίσκεται στο κέντρο των δεικτών) (Εικόνα 22).



Εικ. 22 (Οπτικοποίηση χαρτογράφησης των Fiducial Markers μετά από επανεκκίνηση των αρχείων)

Επίσης, το δέντρο μετασχηματισμών (Transformations Tree) της προσομοίωσης (Εικόνα 23):



Εικ. 23 (Δέντρο μετασχηματισμών πειράματος προσομοίωσης)

(Υποκεφάλαιο 4.2) Πείραμα στο εργαστήριο Ρομποτικής και Αυτομάτου Ελέγχου

Το πείραμα υλοποίησης αλγορίθμου SLAM σε υποβρύχιο ρομποτικό όχημα Blue Rov 2 έλαβε θέση στο εργαστήριο Ρομποτικής και Αυτομάτου Ελέγχου, του Τμήματος Πληροφορικής & Τηλεπικοινωνιών, του Πανεπιστημίου Θεσσαλίας, το εαρινό εξάμηνο έτος 2023. Για την υλοποίηση έπρεπε να λειτουργήσουν οι εντολές :

- rosrun bluerov_ros_playground bluerovnode.launch
(Για τη λειτουργία του ρομπότ μέσω του ROS)
- rosrun aruco_detect aruco_detect.launch
(Για την ανίχνευση των οπτικών οροσήμων)
- rosrun fiducial_slam fiducial_slam.launch
(Για τη δημιουργία του χάρτη και της χωροθέτησης)
- rosrun fiducial_slam fiducial_rviz.launch
(Οπτικοποίηση του χάρτη και της θέσης του ρομπότ)

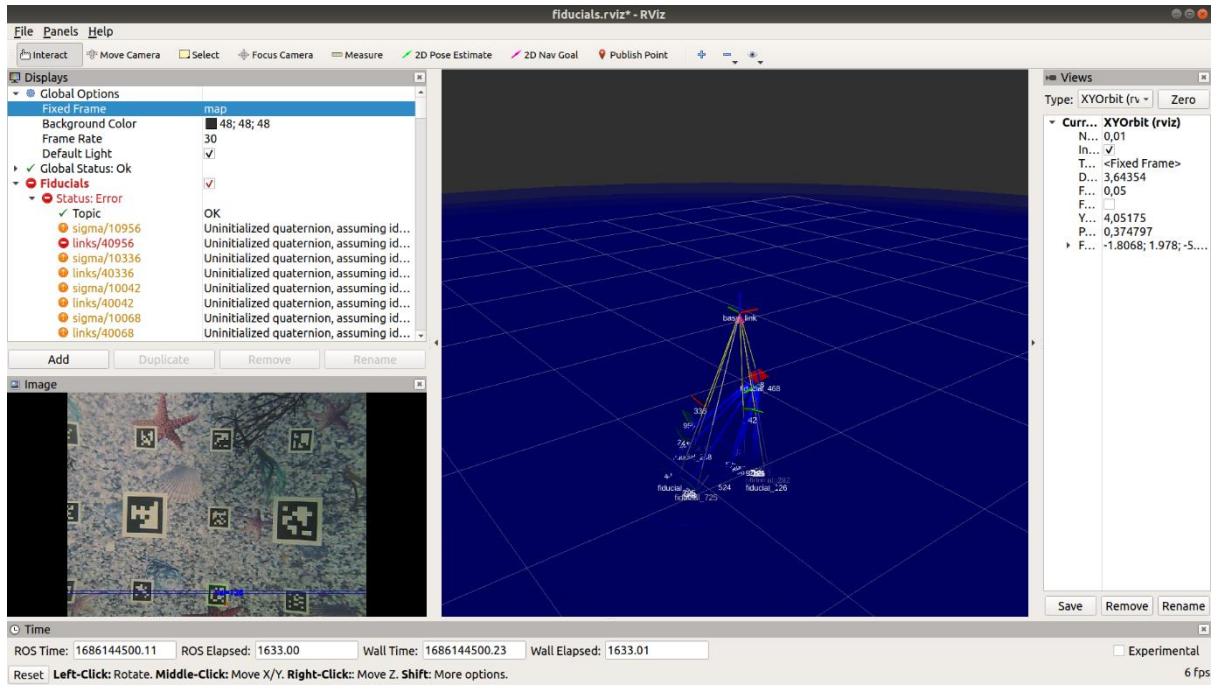
Το ρομποτικό όχημα τοποθετήθηκε μέσα στην πισίνα όπου και ξεκίνησε η ανίχνευση των οροσήμων (Aruco Markers). Στη συνέχεια, μετακινήθηκε μέσω τηλεχειριστηρίου με αργή και σταθερή κίνηση, για την ακριβή ανίχνευση των οροσήμων, όπως και τη πλήρη δημιουργία του χάρτη. Επίσης, κατά τη διάρκεια της χαρτογράφησης είναι απαραίτητο να εμφανίζονται πάντα τουλάχιστον δύο ορόσημα στην προβολή της κάμερας για τη χωροθέτηση του υποβρύχιου οχήματος.

Στην οπτικοποίηση του χάρτη (map), φαίνεται το υποβρύχιο όχημα (base_link) το οποίο δείχνει ποια ορόσημα φαίνονται από την κάμερα εκείνη τη χρονική στιγμή όπως και τον προσανατολισμό του υποβρυχίου.

Στις παρακάτω εικόνες (Εικόνα 24, Εικόνα 25) φαίνονται τα αρχικά στάδια χαρτογράφησης :

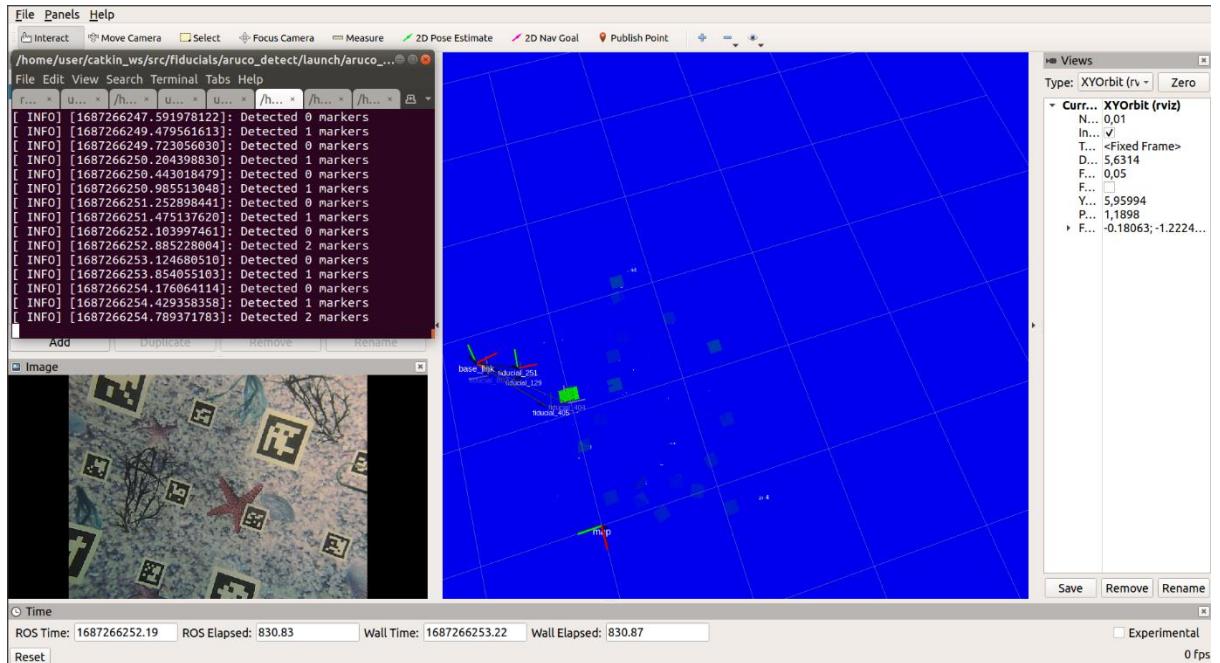


Εικ. 24 (Blue Rov 2 AUV κατά τη διάρκεια του πειράματος (εκκίνηση χαρτογράφησης))



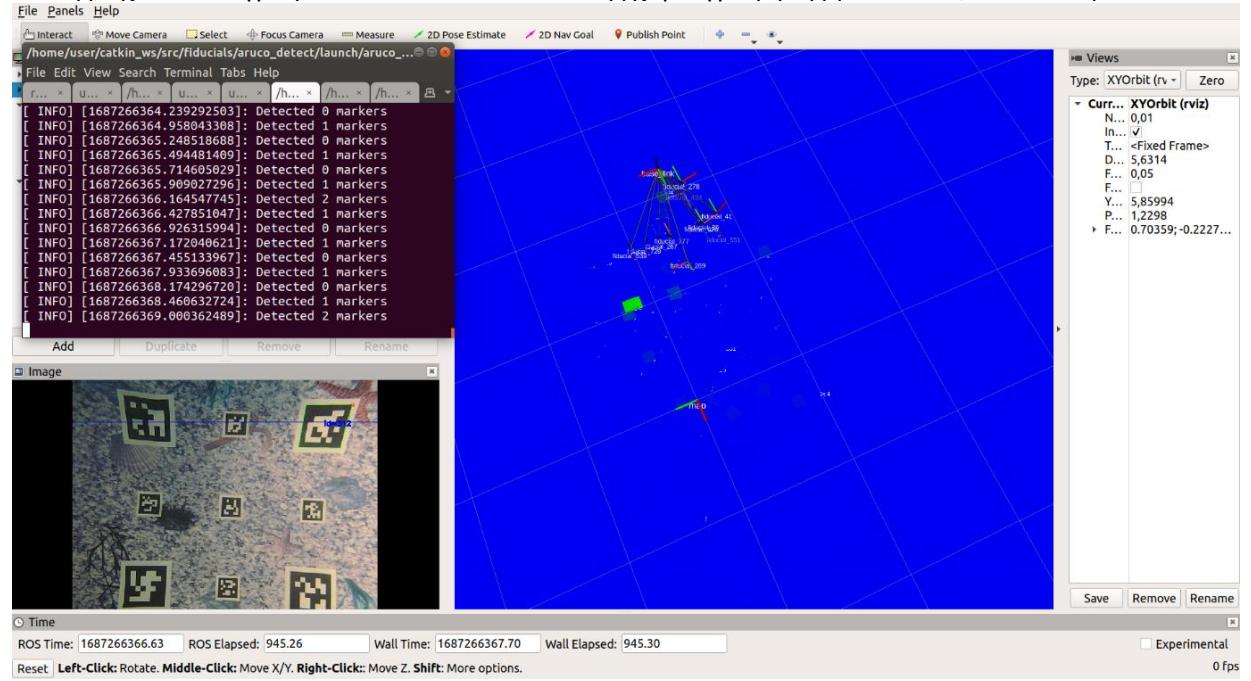
Εικ. 25 (Προβολή διαδικασίας χαρτογράφησης στα αρχικά στάδια)

Κατά τη διαδικασία χαρτογράφησης λήφθηκαν στιγμότυπα όπου έχει χαρτογραφηθεί ο μισός χάρτης (Εικόνα 26, Εικόνα 27) :

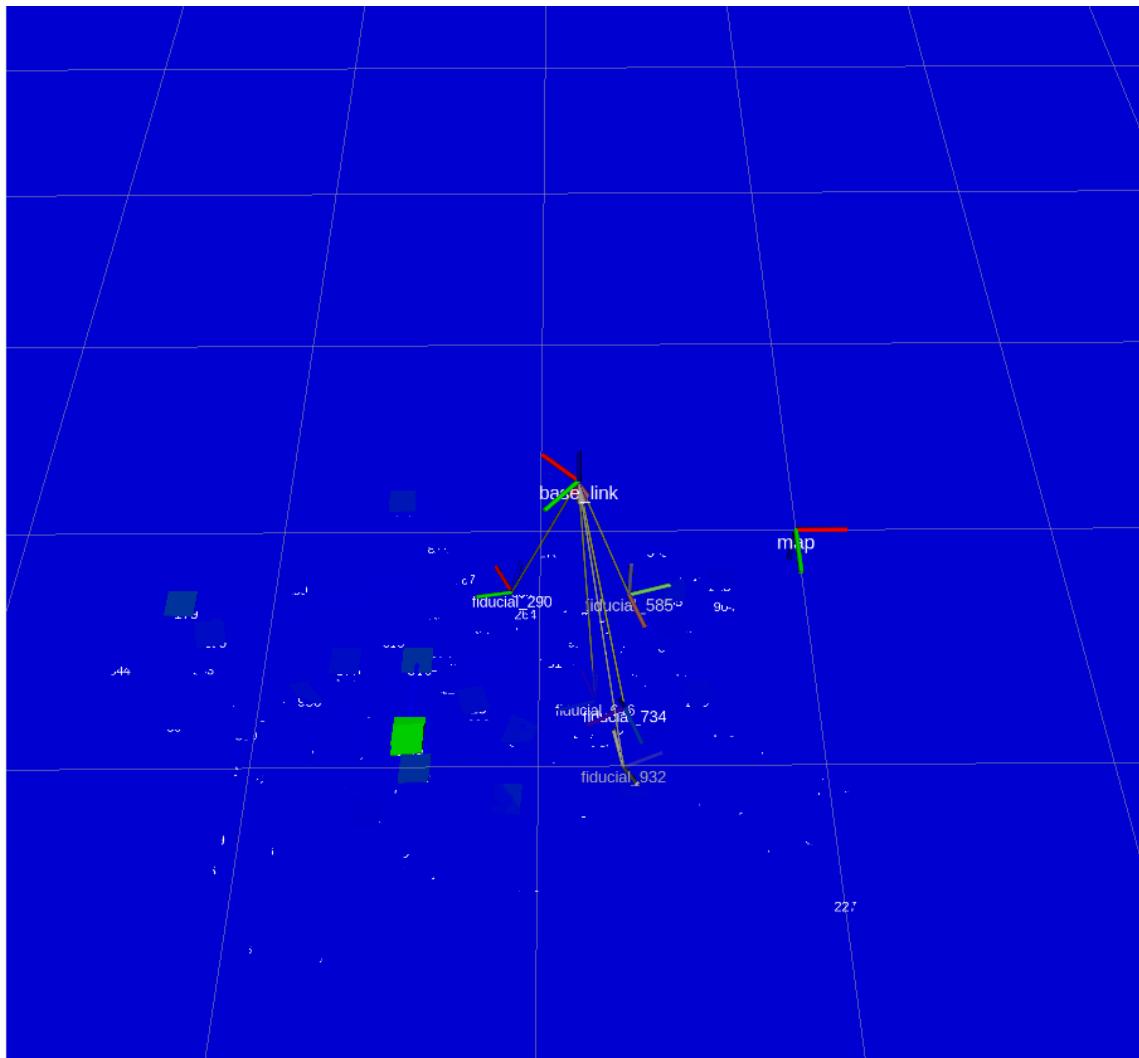


Εικ. 26 (Προβολή διαδικασίας χαρτογράφησης, όπου έχει χαρτογραφηθεί ο μισός περύπου χάρτης (87 Fiducial Markers))

Επίσης, έχουν καταγραφεί και τα τελικά στάδια της χαρτογράφησης (Εικόνα 27, Εικόνα 28) :

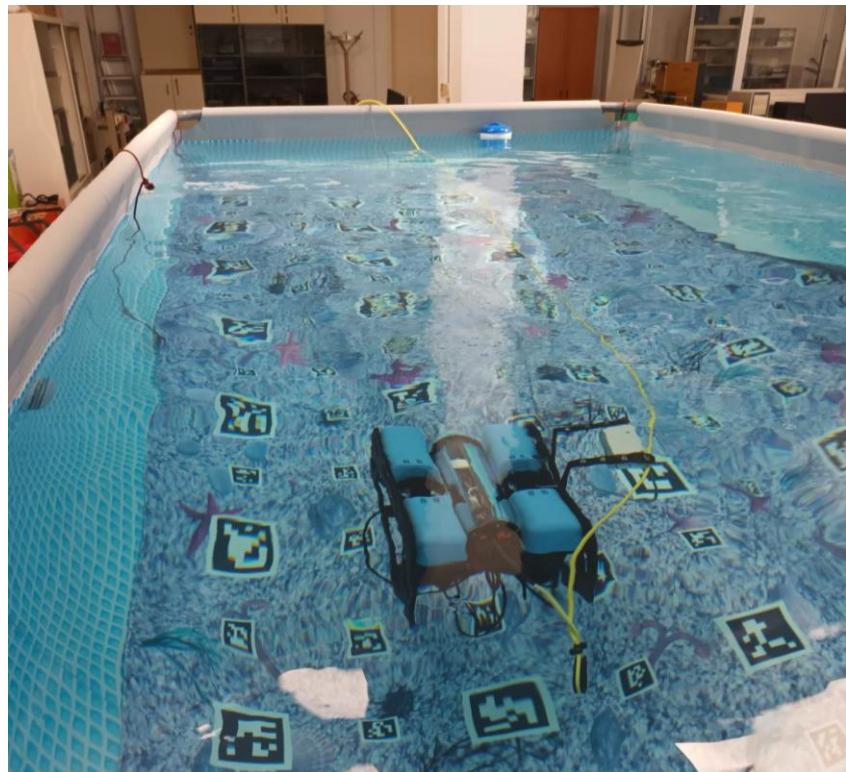


Εικ. 27 (Προβολή διαδικασίας χαρτογράφησης στα τελικά στάδια του χάρτη)

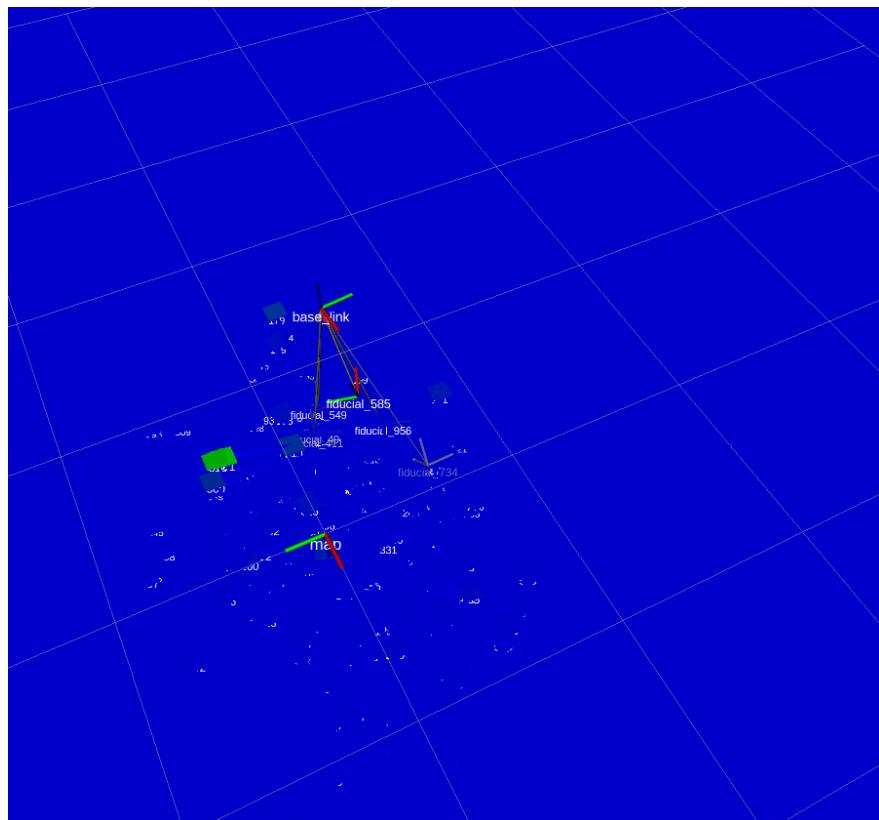


Εικ. 28 (Προβολή ολοκληρωμένου χάρτη)

Στη συνέχεια, τερματίστηκαν οι κόμβοι του πακέτου Fiducials και μετακινήθηκε το ρομπότ στη πισίνα. Όταν οι κόμβοι εκκινήθηκαν πάλι, το ρομποτικό όχημα ανίχνευσε τα ορόσημα και ανανέωσε τον χάρτη. Έτσι, όπως φαίνεται από τις εικόνες (Εικόνα 29, Εικόνα 30) το ρομπότ βρίσκεται στην απέναντι μεριά της πισίνας με διαφορετικό προσανατολισμό.

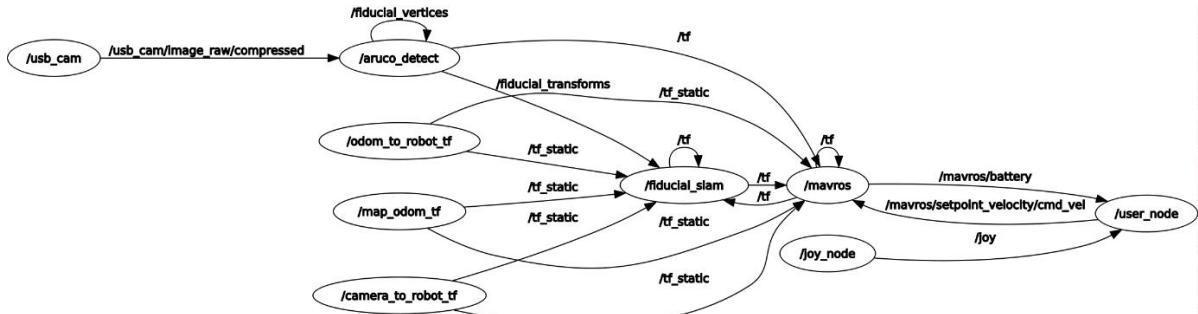


Εικ. 29 (Το υποβρύχιο Blue Rov 2 στην απέναντι μεριά της πισίνας)



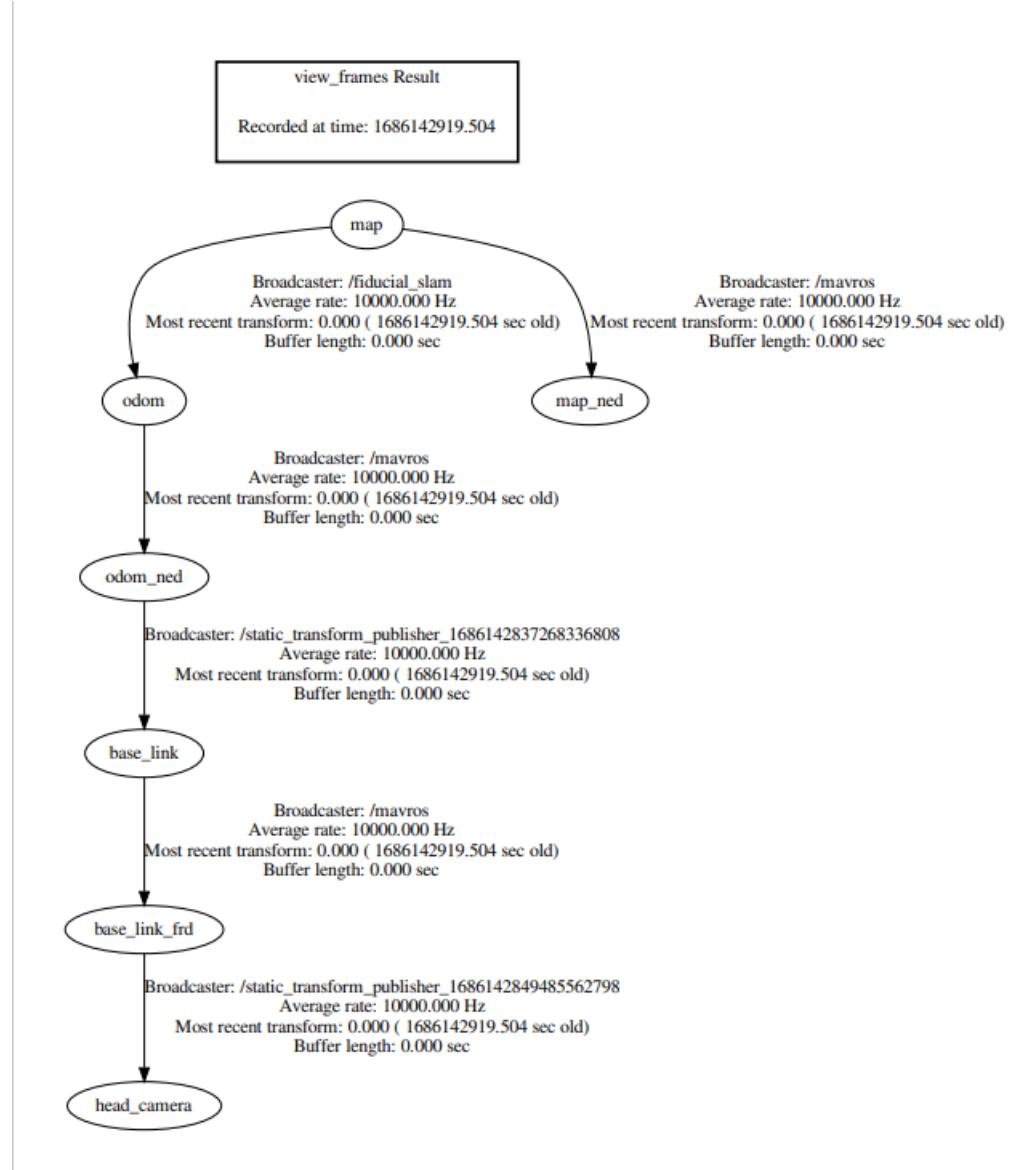
Εικ. 30 (Ο χάρτης μετά την επανεκκίνηση των κόμβων)

Στη συνέχεια, όλοι οι κόμβοι (nodes) και τα θέματα (topics) που χρειάστηκαν για τη λειτουργία του ρομποτικού οχήματος και της κάμερας, σε μορφή γραφήματος (Εικόνα 31) :



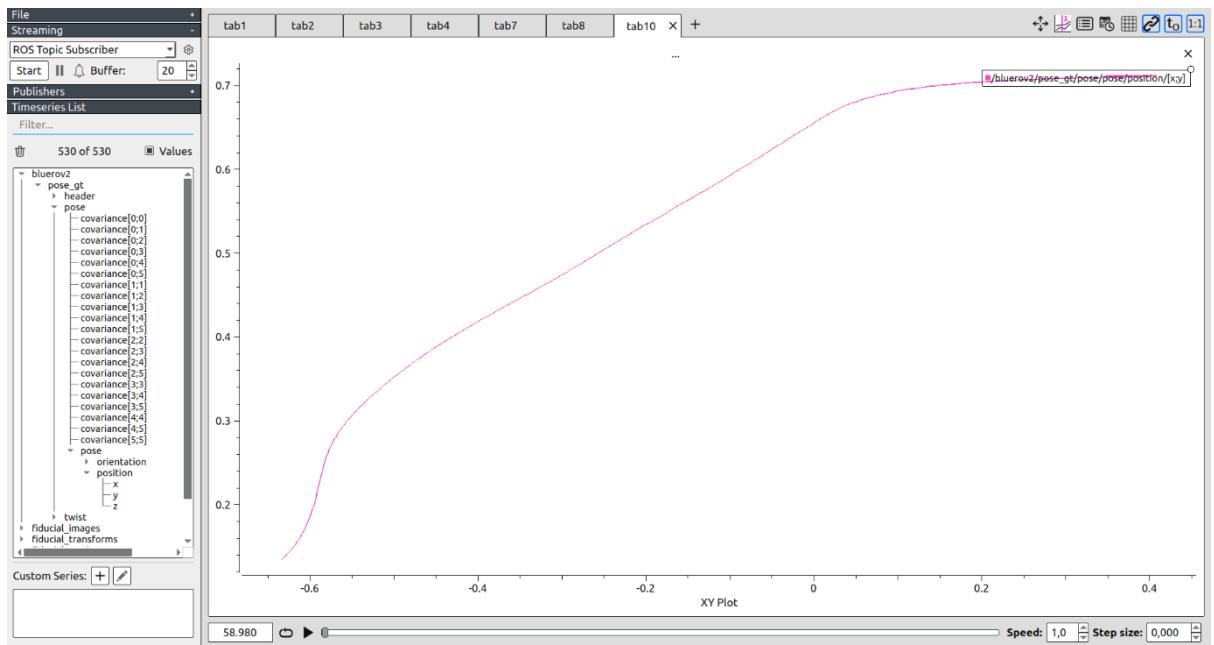
Εικ. 31 (Rqt Graph των πακέτων κατά τη διεξαγωγή του πειράματος)

Τέλος, το δέντρο μετασχηματισμών (Transformations Tree) (Εικόνα 32):



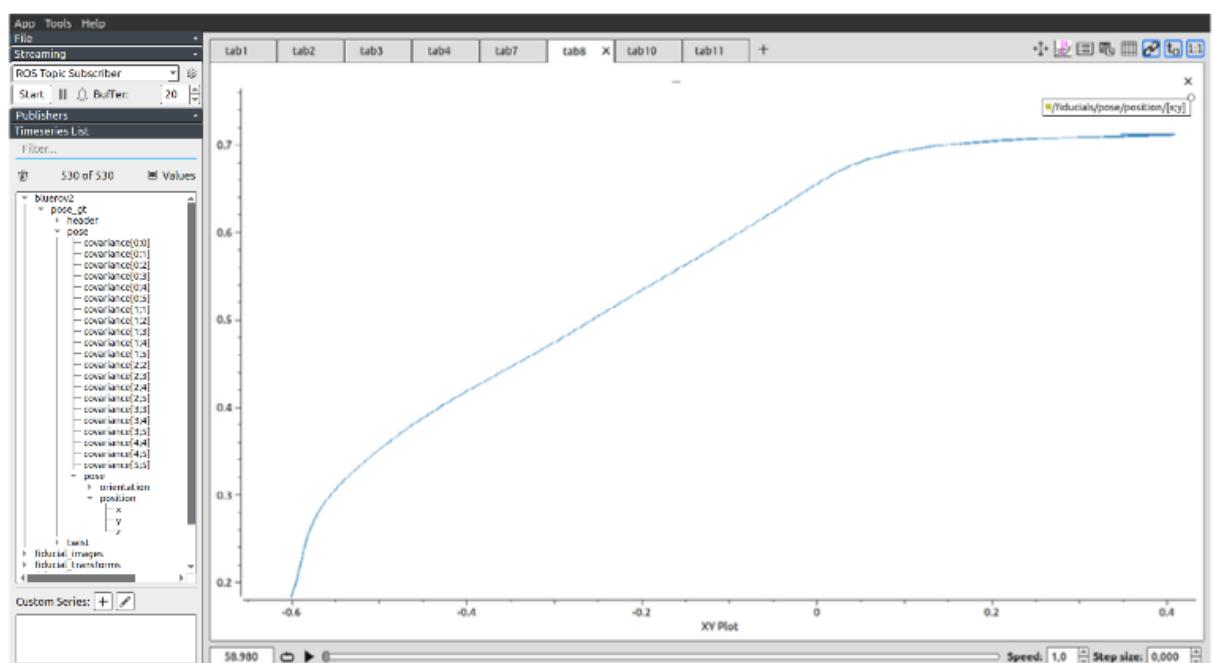
Εικ. 32 (Δέντρο μετασχηματισμών πειραματικής υλοποίησης αλγορίθμου SLAM)

Εμφάνιση στιγμιότυπου (Εικόνα 33) δεδομένων αισθητήρα οδομετρίας (pose_gt), για την κίνηση του Blue Rov 2, σε άξονες (x, y) κατά τη διάρκεια πειράματος :



Εικ. 33 (Στιγμιότυπο γραφήματος κίνησης του ROV σε άξονες x,y κατά τη διάρκεια πειράματος από το topic pose_gt)

Εμφάνιση στιγμιότυπου (Εικόνα 34) δεδομένων οδομετρίας του ROV από το θέμα fiducial_pose για τη θέση του ρομπότ:



Εικ. 34 (Στιγμιότυπο γραφήματος κίνησης του ROV σε άξονες x,y κατά τη διάρκεια πειράματος από το topic fiducial_pose)

(Υποκεφάλαιο 4.3) Συμπεράσματα

Κατά τη διάρκεια του πειράματος παρατηρήθηκε πως όταν το ROV ανιχνεύει παραπάνω από 2 οπτικά ορόσημα, ο αλγόριθμος Fiducial Slam είναι πολύ πιο ακριβής ως προς τη χαρτογράφηση και τη χωροθέτηση του υποβρυχίου, κι αυτό ευθύνεται ως προς την αυξημένη βεβαιότητα που λαμβάνει ο αλγόριθμος από την ύπαρξη των παραπάνω οπτικών οροσήμων. Έτσι, στην αρχή του πειράματος το ROV ήταν απαραίτητο να κινείται αργά, αλλά όσο ο χάρτης των οροσήμων αυξανόταν, η χωροθέτηση γινόταν ταχύτερα και επέτρεπε μεγαλύτερη ελευθερία κίνησης στο ROV.

Επιπλέον, η πλαϊνή κάμερα ήταν ασταθής λόγω της κίνησης του ROV, με αποτέλεσμα η χαρτογράφηση να μην είναι πάντα ακριβής, αλλά με μικρή πιθανότητα λάθους προσανατολισμού των οροσήμων. Για την επίλυση αυτού του ζητήματος, χρειάστηκε το ρομποτικό όχημα να ξαναπεράσει από αυτά τα ορόσημα, με σκοπό την ανίχνευση τους και των γειτονικών οροσήμων, και την επιδιόρθωση του λάθους προσανατολισμού.

Με το πέρας υλοποίησης των παραπάνω, διεξήχθη πείραμα, αρχικά σε προσομοίωση και στη συνέχεια στο εργαστήριο Ρομποτικής και Αυτομάτου Ελέγχου, του Πανεπιστημίου Θεσσαλίας, ένας αλγόριθμος χαρτογράφησης και χωροθέτησης υποβρυχίου οχήματος με χρήση οπτικών οροσήμων (Fiducial Markers). Η επίτευξη των πειραμάτων δείχνει ότι το Blue Rov 2 σε συνδυασμό με το πακέτο Fiducial Slam έχει τη δυνατότητα να χαρτογραφεί το περιβάλλον του, αλλά και να έχει επίγνωση της θέσης του ως προς τον χάρτη, γεγονός που θα είναι πολύ χρήσιμο σε μελλοντικά πειράματα που θα πραγματοποιηθούν στο Blue Rov 2.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ρομποτική

<https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE>
<https://bigblue.academy/gr/ti-einai-i-rompotiki>
<https://builtin.com/robotics>

[2] Υποβρύχια Ρομποτικά Οχήματα

https://en.wikipedia.org/wiki/Autonomous_underwater_vehicle
<https://www.sciencedirect.com/topics/earth-and-planetary-sciences/autonomous-underwater-vehicle>

Έρευνες σχετικές με τη πτυχιακή εργασία:

- [3] <https://www.sciencedirect.com/science/article/pii/S2405896321015068>
- [4] <https://www.frontiersin.org/articles/10.3389/fnbot.2021.801956/full>
- [5] https://www.researchgate.net/publication/286724450_Underwater_SLA_M_Challenges_state_of_the_art_algorithms_and_a_new_biologically-inspired_approach

[6] Blue Rov 2: Ένα υποβρύχιο ρομποτικό όχημα από την Blue Robotics

<https://bluerobotics.com/store/rov/bluerov2/>
<https://en.wikipedia.org/wiki/Robotics>
<http://users.sch.gr/jenyk/index.php/robotics>

[7] Blue Robotics: Εταιρεία σχεδίασης υποβρυχίων ρομποτ

<https://bluerobotics.com/>

[8] ROS (Robot Operating System):

<https://www.ros.org/>
https://en.wikipedia.org/wiki/Robot_Operating_System
<https://wiki.ros.org/ROS/Introduction>

[9] Willow garage

https://en.wikipedia.org/wiki/Willow_Garage

[10] Open Robotics

https://en.wikipedia.org/wiki/Open_Robotics

[11] ROS Transformations

<http://wiki.ros.org/tf/Overview/Transformations>
<https://linklab-uva.github.io/autonomousracing/assets/files/ROS-tf.pdf>

[12] RQT Framework

<http://wiki.ros.org/rqt>
<https://roboticsbackend.com/rqt-graph-visualize-and-debug-your-ros-graph/>

- [13] Rviz 3D visualization tool for ROS.
<http://wiki.ros.org/rviz>
- [14] ROS Package «bluerov_ros_playground»
https://github.com/patrickelectric/bluerov_ros_playground
- [15] Δημιουργός πακέτου βοήθειας ενσωμάτωσης του BlueRov με το ROS
https://github.com/patrickelectric/bluerov_ros_playground
- [16] MAVROS communication Package for ROS
<http://wiki.ros.org/mavros>
- [17] Freefloating_gazebo Water Simulation Package for ROS
https://github.com/freefloating-gazebo/freefloating_gazebo
- [18] Visual SLAM
<https://cvg.cit.tum.de/research/vslam>
<https://www.sciencedirect.com/science/article/pii/S0957417422010156>
-
- [19] Visual SLAM approaches
<https://www.mdpi.com/1424-8220/22/23/9297>
<https://ipsjcv.springeropen.com/articles/10.1186/s41074-017-0027-2>
- [20] A. Babinec, L. Jurišica, P. Hubinsky, and F. Duchon, "Visual localization of mobile robot using artificial markers", *Procedia Engineering*, vol. 96, pp. 1-9, 2014.
<https://www.sciencedirect.com/science/article/pii/S1877705814031427>
- [21] Detection of Aruco Markers
https://docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html
http://wiki.ros.org/aruco_detect
- [22] A. Zakiev, K. Shabalina, T. Tsoy, and E. Magid, "Pilot virtual experiments on aruco and artag systems comparison for fiducial marker rotation resistance", *Proceedings of 14th International Conference on Electromechanics and Robotics Zavalishin's Readings*, pp. 455-464, 2020.
https://www.researchgate.net/publication/335503059_Pilot_Virtual_Experiments_on_ArUco_and_ArTag_Systems_Comparison_for_Fiducial_Marker_Rotation_Resistance
- [23] Embedded Aruco: a novel approach for high precision UAV landing
<https://ieeexplore.ieee.org/abstract/document/9438855>
- [24] Fiducial ROS Package
<http://wiki.ros.org/fiducials>
- [25] ARUCO_DETECT node
http://wiki.ros.org/aruco_detect

[26] Εντοπισμός Fiducial Marker σε σχέση με την κάμερα
http://wiki.ros.org/aruco_detect

[27] Gazebo
<https://gazebosim.org/home>
https://en.wikipedia.org/wiki/Gazebo_simulator

[28] UUV Simulator
<https://uuvsimulator.github.io/>