



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Συνεργασία επίγειου και εναέριου
ρομπότ προς αποδοτική αυτόνομη
εξερεύνηση αγνώστου χώρου στα
πλαίσια του διαγωνισμού
RoboCup-Rescue

Εκπόνηση:

Κλειώ Μπαζεβάνη
ΑΕΜ:7939

Βασίλειος Τσουβάλας
ΑΕΜ:8040

Επίβλεψη:

Λουκάς Πέτρου
Αναπ. Καθηγητής

Εμανουήλ Τσαρδούλιας
Μεταδιδακτορικός Ερευνητής

Θεσσαλονίκη, Μάρτιος 2018

EΓΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης. Θα θέλαμε να ευχαριστήσουμε θερμά τον επιβλέποντα καθηγητή μας, κ. Λουκά Πέτρου για την εμπιστοσύνη που μας έδειξε, αναθέτοντάς μας την παρούσα διπλωματική εργασία.

Ένα ιδιαίτερο ευχαριστώ οφείλουμε στον Μεταδιδακτορικό Ερευνητή Δρ. Εμμανουήλ Τσαρδούλια για την άριστη συνεργασία, την αμέριστη υπομονή και την καθοδήγησή του καθ'όλη τη διάρκεια της διπλωματικής μας εργασίας.

Τέλος θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας για τη στήριξη και την ανοχή που μας έδειξαν όλο αυτό το διάστημα.

ΠΕΡΙΛΗΨΗ

Η ρομποτική αποτελεί ένα διεπιστημονικό κλάδο της μηχανικής η οποία εμπεριέχει έννοιες των κλάδων των Μηχανολόγων Μηχανικών, των Ηλεκτρολόγων Μηχανικών και της Πληροφορικής Επιστήμης. Σκοπός του κλάδου αυτού είναι να αναπτύξει τεχνολογία η οποία να προστατεύει και να βελτιώνει την ποιότητα ζωής των ανθρώπων, υπακούοντας σε εντολές και χωρίς να βλάπτει το ανθρώπινο ον, όπως αρχικά διατυπώθηκε από τον συγγραφέα Ισαάκ Ασίμωφ το 1942.

Όπως οι άνθρωποι, έτσι και τα ρομπότ δρουν πιο αποδοτικά όταν συμμετέχουν ως μέλη σε μία ομάδας. Αυτό όμως δεν αναφεί το γεγονός ότι για να δουλέψει αποτελεσματικά μια ομάδα πρέπει να υπάρχει σωστή επικοινωνία αλλά και σωστή συνεργασία μεταξύ των μελών της ομάδας. Το πρόβλημα με το οποίο ασχολείται η παρούσα διπλωματική εργασία είναι η συνεργασία ενός εναέριου και ενός επίγειου ρομποτικού πράκτορα με σκοπό την εξερεύνηση και χαρτογράφηση ενός άγνωστου χώρου. Η εφαρμογή που αναπτύχθηκε περιγράφεται αναλυτικά παρακάτω.

Στο πρώτο στάδιο, μόλις δοθεί το σήμα ότι και οι δύο πράκτορες είναι έτοιμοι, ο εναέριος πράκτορας απογειώνεται και ζεκινάει την αναζήτηση του επίγειου πράκτορα στον χώρο. Μόλις τον εντοπίσει, ο εναέριος πράκτορας γνωρίζει τη θέση του μέσα στο χάρτη του επίγειου. Στο δεύτερο στάδιο, το οποίο αποτελεί το βασικό κοινωνίας της υλοποίησης, ο επίγειος πράκτορας δίνει εντολή στον εναέριο να κατευθυνθεί προς τα όρια του εξερευνημένου μέχρι τώρα χώρου, να λάβει φωτογραφίες με την κάμερα του, η οποία βρίσκεται τοποθετημένη στο κάτω μέρος του εναέριου πράκτορα, και μέσω μίας μεθόδου αντιστοίχισης να διευρυνθεί ο χάρτης. Εν συνεχείᾳ, ο επίγειος πράκτορας επιλέγει κάποιους υποψήφιους στόχους από τον διευρυμένο χάρτη και στέλνει τον εναέριο πράκτορα να τους αξιολογήσει. Αφού ο εναέριος πράκτορας επισκεψθεί όλους τους υποψήφιους στόχους, λαμβάνονται υπόψιν οι δύο αξιολογήσεις (μία του επίγειου και μία του εναέριου πράκτορα) και επιλέγεται ο βέλτιστος στόχος. Στην συνέχεια ο επίγειος πράκτορας κατευθύνεται προς τον στόχο αυτό. Η βασική αυτή διαδικασία επαναλαμβάνεται μέχρι να εξερευνηθεί ή να καλυφθεί πλήρως ο χώρος.

ΠΕΡΙΛΗΨΗ

Για την υλοποίηση και την οπτικοποίηση των προσομοιώσεων χρησιμοποιήθηκαν ο 3D προσομοιωτής RViz και ο 3D προσομοιωτής Gazebo. Για τη διεξαγωγή των προσομοιώσεων δημιουργήθηκαν τα αντίστοιχα μοντέλα στους προσομοιωτές και σχεδιάστηκαν οι χώροι προς εξερεύνηση, οι οποίοι πληρούν τις προϋποθέσεις του διαγωνισμού RoboCup. Οι προσομοιώσεις που διεξήχθησαν αποτελούνται από δύο φάσεις. Κατά την πρώτη φάση η εξερεύνηση πραγματοποιήθηκε από τον επίγειο ρομποτικό πράκτορα ο οποίος δρούσε μόνο με βάση τα δεδομένα από τους αισθητήρες του, ενώ κατά τη δεύτερη φάση η εξερεύνηση γινόταν από τον επίγειο ρομποτικό πράκτορα σε συνεργασία των εναέριο ρομποτικό πράκτορα.

ABSTRACT

Collaboration of ground and aerial robots for efficient autonomous exploration of unknown space under the conditions of the RoboCup-Rescue contest

Robotics is an interdisciplinary branch of engineering that incorporates concepts of Mechanical Engineering, Electrical Engineering and Computer Science. The purpose of this industry is to develop technology that protects and improves the quality of life of people, obeys orders without harming the human being, as originally formulated by the writer Isaac Asimov in 1942.

Like humans, robots work more efficiently when they are members of a group. However, this does not negate the fact that in order for a team to work effectively, must exist good communication and good cooperation between the team members. The problem that this diploma thesis deals with is the collaboration of an Unmanned Aerial Vehicle and an Unmanned Ground Vehicle with the purpose of exploring and mapping an unknown space. The developed application is described in detail below.

In the first stage, once both agents are ready, the aerial agent takes off and starts searching for the ground agent. Once identified, the aerial agent knows its position in the map of the ground agent. In the second stage, which is the basic part of the implementation, the ground agent instructs the aerial agent to navigate to the boundaries of the space that explored so far, capture photos with its camera, and a mapping method is used to expand the map. Subsequently, the ground agent selects some candidate targets from the extended map and sends the aerial agent to evaluate them. When the aerial agent visits all targets, the two evaluations (one of the ground and one aerial agent) are performed and the optimal target is selected. At the final stage, the ground agent is directed to this target. This basic process is repeated until the space is fully explored or covered.

ABSTRACT

The 3D simulator RViz and the 3D simulator Gazebo were used to implement and visualize the simulations. For the simulations, the corresponding models were added in the simulators and three different maps, which meet the requirements of RoboCup, were designed. The simulations consist of two phases. In the first phase the exploration was carried out by the ground robotic agent, which only used the data from its sensors, while in the second phase, the exploration was performed by the ground robotic agent in collaboration with the aerial robotic agent.

K. Baxevani, V. Tsouvalas

Aristotle University of Thessaloniki

Dept. Electrical and Computer Engineering

March 2018

Περιεχόμενα

1 Εισαγωγή	1
1.1 Περιγραφή του προβλήματος	1
1.1.1 Απαιτήσεις του Προβλήματος	1
1.1.2 Υποθέσεις και Περιορισμοί του Προβλήματος	2
1.2 Σκοπός-Συνεισφορά της Διπλωματικής Εργασίας	2
1.3 Διάρθρωση της Αναφοράς	2
2 Επισκόπηση Ερευνητικής Περιοχής	4
2.1 Ανίχνευση σχετικής ύφεσης των UGV/UAV	4
2.2 Ολοκληρωμένες Τλοποιήσεις	5
3 Θεωρητικό Υπόβαθρο	
- Εργαλεία	8
3.1 ROS - Robot Operating System	8
3.1.1 Βασικές έννοιες του ROS	8
3.1.2 Πωκέτα του ROS	9
3.2 GAZEBO	10
3.3 RViz	11
3.4 OpenCV	12
3.5 SLAM - Simultaneous Localization and Mapping	13
3.6 OGM - Occupancy Grid Map	13
3.7 Αλγόριθμος Brushfire	14
3.8 PID Controller	14
3.9 Αλγόριθμος A*	15
3.10 Uniform Sampling Path Planning	17
4 Τλοποίηση	18
4.1 Τροποποίηση Ρομποτικών Μοντέλων στο GAZEBO	20
4.2 Επικοινωνία των δύο Προστόρων	21
4.2.1 Join Tree Service	21
4.2.2 Limit Exploration Service	22
4.2.3 Drone Control Service	22

4.2.4	Potential Targets Evaluation Service	24
4.3	Πακέτο του Drone Controller	25
4.4	Πακέτο του Turtlebot Controller	28
4.4.1	Localization	28
4.4.2	Drone Control	29
4.4.3	Limit Exploration	33
4.4.4	Potential Targets Evaluation	34
4.5	Πακέτο του Turtlebot Controller	38
4.5.1	Turtlebot Control	41
4.5.2	Turtlebot Navigation	41
4.5.3	Turtlebot Perception	46
4.5.4	Target Selection	46
4.5.5	Path Planning	50
4.6	TF Frame Tree Controller	50
5	Πειράματα	52
5.1	Δημιουργία πειραματικών χώρων στην πλατφόρμα προσομοιώσεων GAZEBO	53
5.2	Διεξαγωγή Πειραμάτων	55
5.3	Αποτελέσματα Πειραμάτων	57
5.3.1	Εξερεύνηση μόνο με το Turtlebot	58
5.3.2	Εξερεύνηση και με τους δύο πρόσκτορες με το Drone να εξερευνά τα όρια του OGM	61
5.3.3	Εξερεύνηση με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων	64
5.4	Συγκρίσεις Αποτελεσμάτων των Πειραμάτων	67
5.4.1	Σύγκριση Πειραματικών Κατηγοριών I και II	67
5.4.2	Σύγκριση Πειραματικών Κατηγοριών I και III	69
6	Συμπεράσματα - Σχολιασμός	71
6.1	Συμπεράσματα	71
6.2	Προβλήματα	72
6.3	Μελλοντική Μελέτη	72

Κατάλογος Σχημάτων

1	Turtlebot - AR Drone 2.0	11
2	Ενδεικτικό στιγμιότυπο από τα πειράματα στην πλατφόρμα RViz	11
3	Ενδεικτικό παράδειγμα OGM χάρτη που εξήχθη από τα πειράματα	14
4	Γραφική απεικόνιση ενός PID ελεγκτή	15
5	Εύρεση βέλτιστου μονοπατιού με χρήση του αλγορίθμου A*	15
6	Βασική Δομή Πακέτων της Υλοποίησης	18
7	Συνοπτικό Διάγραμμα Ροής Υλοποίησης	19
8	Ενδεικτικό Στιγμιότυπο Τροποποιημένων Μοντέλων των Ρομποτικών Πρακτόρων στο Gazebo	20
9	Διάγραμμα Ροής Κλήσης την Υπηρεσίας Join Tree	21
10	Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρακτόρων για την Λήψη Πληροφοριών για τα Όρια του OGM	23
11	Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρακτόρων για την εκτέλεση βασικών λειτουργιών του Drone	23
12	Μήνυμα που δημιουργήθηκε για την αποστολή της αξιολόγησης των πιθανών στόχων από το Drone	24
13	Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρακτόρων για την Αξιολόγηση των Πιθανών Στόχων από το Drone	25
14	Βασική Δομή Πακέτων του Drone Contoller	26
15	Βασικό Διάγραμμα Ροής του Drone	27
16	Διάγραμμα Ροής παρακολούθησης Ar Marker από το Drone	31
17	Διάγραμμα Ροής Εξερεύνησης Ορίων του χάρτη από το Drone	33
18	Διάγραμμα Ροής Αξιολόγησης Πιθανών Στόχων από το Drone	35
19	Γραφική Απεικόνιση Κατανομής Βαρών για την Αξιολόγησης Πιθανών Στόχων από το Drone	36
20	Βασική Δομή Πακέτων του Turtlebot Contoller	39
21	Βασικό Διάγραμμα Ροής του Turtlebot	40
22	Γραφική Αναπαράσταση Υπολογισμού Ταχυτήτων για την Αποφυγή Εμπόδιων του Turtlebot	41
23	Ενδεικτικό Στιγμιότυπο του αποτελέσματος ενίσχυσης του OGM από την Λειτουργία της Εξερεύνησης των Ορίων του από το Drone	44

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

24	Γραφική Αναπαράσταση Υπολογισμού Ταχυτήτων για την προσέγγιση ενός στόχου από το Turtlebot	44
25	Διάγραμμα Ροής που ακολουθείται για την επιλογή στόχου από το Turtlebot	45
26	Γραφική Αναπαράσταση Επικοινωνίας μεταξύ Turtlebot Perception και CRSM SLAM για την παροχή του χάρτη στον Turtlebot Controller	46
27	Διάγραμμα Ροής της διαδικασίας επιλογής στόχου του Turtlebot	49
28	Γραφική Αναπαράσταση Επικοινωνίας μεταξύ Turtlebot Controller και OGM Path Planner για την δημιουργία Μονοπατιού προς τον Στόχο	50
29	Γραφική Απεικόνιση των δύο TF Frame Tree πρίν την κλήση της υπηρεσίας Join Tree Service από τον Drone Controller	51
30	Γραφική Απεικόνιση του ενωμένου TF Frame Tree μετά την κλήση της Join Tree Service από τον Drone Controller	51
31	Διαγράμματα Ποσοστού Εξερεύνησης και Κάλυψης με τον Χρόνο του Χώρου $100m^2$ από το Turtlebot	53
32	Κατόψιες των χώρων που σχεδιάστηκαν για την υλοποίηση των προσομοιώσεων	54
33	Εικόνες που εξήγιασαν μετά την ολοκλήρωση των πειραμάτων από το εργαλείο Rviz	55
34	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 1 μόνο από το UGV	58
35	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 2 μόνο από το UGV	59
36	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 μόνο από το UGV	60
37	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 1 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM	61
38	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 2 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM	62
39	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM	63
40	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 1 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους	64
41	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 2 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους	65
42	Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους	66

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

43	Συγκριτικό Ραβδόγραμμα Πειραμάτων για τις Κατηγορίες Πειραμάτων I και II	68
44	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)	75
45	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη (Full Coverage)	76
46	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)	76
47	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)	77
48	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη (Full Coverage)	77
49	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)	78
50	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)	78
51	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη (Full Coverage)	79
52	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)	79
53	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την εξερεύνηση (Full Exploration)	80
54	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη(Full Coverage)	81
55	Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)	81
56	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την εξερεύνηση (Full Exploration)	82
57	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη (Full Coverage)	82
58	Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)	83
59	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την εξερεύνηση (Full Exploration)	83

KATALOGOS SXHMATON

60	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη (Full Coverage)	84
61	Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constraint)	84

Κατάλογος Πινάκων

1	Πίνακας Αντιστοίχησης Εντολής του Control Drone Service με Βασική Λειτουργία του Drone	32
2	Πίνακας Βαρών για τα Κόστη της διαδικασίας του Target Selection του Turtlebot	56
3	Πίνακας Συγχριτικών Αποτελεσμάτων για τις Κατηγορίες Πειραμάτων I και II	67
4	Πίνακας Συγχριτικών Αποτελεσμάτων για τις Κατηγορίες Πειραμάτων I και III	69

Κατάλογος Αλγορίθμων

1	A* Algorithm	16
2	Uniform Sampling	17
3	Drone Localization Algorithm	28
4	Move Drone Forward by Function	29
5	Drone Marker Follower Algorithm	30
6	Drone Limit Exploration Algorithm	34
7	Drone Potential Targets Evaluation Algorithm	37
8	Turtlebot OGM Limit Points Calculation Algorithm	42
9	Turtlebot Image Merging Algorithm	43

Ακρωνύμια Εγγράφου

UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
ROS	Robot Operating System
OGM	Occupancy Grid Map
AR marker	Augmented Reality Marker
RPC	Remote Procedure Call
OOI	Object of Interest
PID controller	Proportional Integral Derivative controller
SLAM	Simultaneous Localization and Mapping
CRSM SLAM	Critical Rays Scan Match SLAM
OpenCV	Open Source Computer Vision Library
PRM	Probabilistic RoadMap
FOV	Field of View

1

Εισαγωγή

Ένα ρομπότ (robot) πρόκειται για έναν πράκτορα εφοδιασμένο με αισθητήρες και τελεστές, ο οποίος μπορεί να προγραμματιστεί για να παρουσιάσει αυτόνομα νοημοσύνη. Με τον όρο πράκτορα εννοούμε μία αυτόνομη οντότητα που προσπαθεί να επιτύχει ένα στόχο και ενεργεί σε ένα περιβάλλον. Συγκεκριμένα, ο κλάδος της ρομποτικής αναπτύχθηκε με σκοπό την ανάπτυξη τεχνολογίας η οποία θα υπακούει στις Αρχές οι οποίες διατυπώθηκαν πρώτη φορά από τον συγγραφέα Ισαάκ Ασίμωφ το 1942. Σύμφωνα με τις αρχές αυτές η τεχνολογία αυτή πρέπει να προστατεύει και να βελτιώνει την ποιότητα ζωής των ανθρώπων, υπακούοντας σε εντολές και χωρίς να βλάπτει το ανθρώπινο ον. Τα ρομπότ κατηγοριοποιούνται ανάλογα με τη λειτουργία τους, όπως βιομηχανικά, οικιακά, ιατρικά, στρατιωτικά, ερευνητικά κλπ.

Παρόλο που στην αρχή, για την υλοποίηση των ρομποτικών εργασιών χρησιμοποιούνταν, κατά μεγάλο ποσοστό, επίγειοι ρομποτικοί πράκτορες, τα τελευταία χρόνια οι εναέριοι πράκτορες έχουν κάνει την εμφάνιση τους. Με την εισαγωγή των εναέριων πρακτόρων διευρήνθηκαν οι δυνατότητες των ρομποτικών εφαρμογών, καθώς ο βαθμός ελευθερίας κινήσεων αυξήθηκε. Στην παρούσα διπλωματική διερευνάται αν η συνεργασία ενός επίγειου και ενός εναέριου ρομποτικού πράκτορα με σκοπό την εξερεύνηση ενός άγνωστου χώρου εκτελείται γρηγορότερα σε σχέση με την περίπτωση όπου η εξερεύνηση αυτή εκτελείται μόνο από το επίγειο ρομπότ.

1.1 Περιγραφή του προβλήματος

Το πρόβλημα το οποίο καλείται να αντιμετωπίσει η παρούσα διπλωματική αποτελείται από τρία σκέλη. Το πρώτο σκέλος αποτελεί η δημιουργία μίας καλώς ορισμένης διαδικασίας η οποία επιτρέπει την επικοινωνία και τη συνεργασία των δύο ρομποτικών πρακτόρων. Το δεύτερο σκέλος του προβλήματος είναι η σωστή εξαγωγή συμπερασμάτων από τον εναέριο πράκτορα. Ο εναέριος πράκτορας χρησιμοποιώντας τις διαδοχικές λήψεις από την κάμερα που βρίσκεται στο κάτω μέρος του, πρέπει να εξάγει τα δικά του συμπεράσματα για το χώρο με βάση τα οποία θα μπορεί να επηρεάζει την επιλογή μονοπατιού του επίγειου ρομποτικού πράκτορα. Το τρίτο και τελευταίο σκέλος του προβλήματος είναι η βέλτιστη εξερεύνηση του άγνωστου χώρου από τον επίγειο πράκτορα χρησιμοποιώντας τα δεδομένα από τους αισθητήρες που διαθέτει και τις πληροφορίες που δέχεται από τον εναέριο πράκτορα. Στην συνέχεια θα γίνει μια περιγραφή των απαιτήσεων που πρέπει να πληρούνται ώστε να επιτευχθεί ο στόχος της διπλωματικής εργασίας, όπως και των περιορισμών και υποθέσεων που έγιναν κατά την κατασκευή των άλγορίθμων και την διεξαγωγή των πειραμάτων.

1.1.1 Απαιτήσεις του Προβλήματος

Οι απαιτήσεις του προβλήματος είναι οι παρακάτω:

- Η λειτουργία του όλου συστήματος πρέπει να μην απαιτεί εξωτερικό χρήστη, δηλαδή η ομάδα ρομποτικών πρακτόρων να είναι αυτόνομη.
- Θα πρέπει να υλοποιηθεί το απαραίτητο λογισμικό για κάθε ρομπότ για την επίτευξη των επιμέρους διαδικασιών του.

- Θα πρέπει να δημιουργηθεί το κατάλληλο δίκτυο επικοινωνίας ώστε τα δύο ρομπότ να μπορούν να ανταλλάζουν πληροφορίες μεταξύ τους.

1.1.2 Υποθέσεις και Περιορισμοί του Προβλήματος

Οι υποθέσεις / περιορισμοί της διπλωματικής εργασίας είναι οι παρακάτω:

- Ο χώρος στον οποίο κινούνται τα ρομπότ είναι στατικός.
- Υποθέτουμε ότι στο επίγειο ρομπότ έχει τοποθετηθεί ένας AR Marker ώστε να γίνει ο εντοπισμός του από την κάμερα που διαθέτει ο εναέριος ρομποτικός πράκτορας.
- Τα εμπόδια του χώρου θεωρήθηκαν ότι είχαν κόκκινο χρώμα έτσι ώστε να γίνονται εύκολα αντιληπτά από την κάμερα του εναέριου ρομπότ.
- Ο εναέριος ρομποτικός πράκτορας θεωρήθηκε ότι πετάει σε συγκεκριμένο ύψος (5 m) πάνω από οποιοδήποτε εμπόδιο.
- Τα εμπόδια έχουν συγκεκριμένο ύψος (1.5 m) όπως στο διαγωνισμό RoboCup-Rescue.

1.2 Σκοπός-Συνεισφορά της Διπλωματικής Εργασίας

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι να επιτύχει την αποτελεσματική συνεργασία ενός εναέριου και ενός επίγειου ρομποτικού πράκτορα κατά τη διεργασία εξερεύνησης άγνωστου χώρου, και να ερευνήσει κατά πόσο η συνεργασία αυτή βελτιώνει την ποιότητα της εξερεύνησης και το χρόνο που απαιτείται για να ολοκληρωθεί.

Συγκεκριμένα διερευνήθηκε ο χρόνος που χρειάστηκε και ο αριθμός των στόχων που απαιτήθηκε για να δημιουργηθεί ο χάρτης ενός άγνωστου χώρου, αρχικά στην περίπτωση όπου ο επίγειος πράκτορας δρούσε μόνος του και στην συνέχεια στην περίπτωση που ο επίγειος πράκτορας συνεργάστηκε με έναν εναέριο πράκτορα.

Μετά την υλοποίηση της διπλωματικής αυτής μπορούν να εξαχθούν κάποια συμπεράσματα τα οποία προσδιορίζουν κάποιες περιπτώσεις που είναι ωφέλιμο να επιλέγεται μία ομάδα από επίγειους και εναέριους ρομποτικούς πράκτορες, και κάποιες που προκύπτει το αντίθετο αποτέλεσμα. Οι περιπτώσεις αυτές αφορούν τόσο τα χαρακτηριστικά του χώρου (το μέγεθός του, την πυκνότητα των εμποδίων), όσο και το σκοπό της εξερεύνησης (πλήρη κάλυψη ή πλήρη εξερεύνηση).

1.3 Διάρθρωση της Αναφοράς

Η διάρθρωση της παρούσας διπλωματικής είναι η εξής:

- **Κεφάλαιο 2:** Παρατίθεται η ανασκόπηση της ερευνητικής περιοχής αναφορικά με τα αντικείμενα τα οποία μελετήθηκαν για την υλοποίηση της παρούσας διπλωματικής.
- **Κεφάλαιο 3:** Παρατίθενται όλα τα εργαλεία αλλά και το θεωρητικό υπόβαθρο που ήταν απαραίτητα για την υλοποίηση της παρούσας διπλωματικής. Ειδικότερα τα εργαλεία περιλαμβάνουν τα εξής:
 - **ROS:** Πλατφόρμα λογισμικού ρομποτικών εφαρμογών.

- **Gazebo**: 3D προσομοιωτής ρομπότ
 - **RViz**: 3D προσομοιωτής ρομπότ
 - **OpenCV**: Βιβλιοθήκη για Computer Vision
 - **SLAM**: Πρόβλημα εντοπισμού θέσης και χαρτογράφησης άγνωστου χώρου
 - **OGM**: Είδος απεικόνισης χάρτη
 - **PID controller**: Ελεγκτής θέσης και ταχύτητας
 - **Αλγόριθμος A***: Αλγόριθμος εύρεσης βέλτιστου μονοπατιού
 - **Uniform Sampling**: Είδος δειγματοληψίας
- **Κεφάλαιο 4**: Περιγράφεται η όλη διαδικασία υλοποίησης της εργασίας. Συγκεκριμένα παρουσιάζεται η λογική πάνω στην οποία βασίστηκε η υλοποίηση και επεξηγούνται τα κομμάτια του προγραμματιστικού κώδικα που χρειάστηκε να γραφτούν. Τα βασικά αυτά κομμάτια είναι τα παρακάτω:
 - **Επικοινωνία των δύο πρακτόρων**
 - **Μέθοδοι του Drone Controller**
 - **Μέθοδοι του Turtlebot Controller**
 - **Κεφάλαιο 5**: Το κεφάλαιο αυτό αφορά τα πειράματα που διεξήχθησαν με σκοπό την αξιολόγηση των μεθόδων που χρησιμοποιήθηκαν.
 - **Κεφάλαιο 6**: Τέλος, το κεφάλαιο αυτό περιέχει σχόλια που αφορούν τόσο την υλοποίηση η οποία περιγράφηκε στα παραπάνω κεφάλαια όσο και τα αποτελέσματα που προέκυψαν από τα πειράματα που υλοποιήθηκαν. Τέλος αναφέρονται και κάποιες προτάσεις για κάποια μελλοντική επέκταση και βελτίωση της υλοποίησης αυτής.

2

Επισκόπηση Ερευνητικής Περιοχής

Η χρήση ετερογενών ρομπότ για την διεκπεραίωση διαφόρων εργασιών αν και παρέχει πολλά πλεονεκτήματα σε σχέση με την χρήση μόνο ενός ρομπότ, κρύβει και πολλές προκλήσεις. Μια εξ αυτών αποτελεί η συνεργασιακή στρατηγική που θα ακολουθηθεί ώστε να γίνεται καλύτερη η αξιοποίηση των δυνατοτήτων των δύο ρομπότ για την επίτευξη της εργασίας που έχουν να πραγματοποιήσουν. Το μεγαλύτερο όμως πρόβλημα που εμφανίζεται στην χρήση πολλών ρομπότ, και διογκώνεται με την χρήση ετερογενών ρομπότ είναι η συγχώνευση των πληροφοριών, που παρέχουν μεμονωμένα τα ρομπότ, σε μία κοινή πληροφορία, η οποία χρησιμοποιείται για την κατασκευή ενός συνολικού χάρτη. Με τον τρόπο αυτό, τα ρομπότ είναι ικανά να κινηθούν αυτόνομα στον χώρο. Η σωστή κατασκευή του χάρτη αυτού είναι ζωτικής σημασίας αφού χωρίς αυτήν δεν είναι ικανή η επίτευξη περαιτέρω εργασιών.

2.1 Ανίχνευση σχετικής θέσης των UGV/UAV

Μια από τις βασικότερες μεθόδους για την συγχώνευση των πληροφοριών των δύο ρομπότ είναι η αναγνώριση της σχετικής θέσης των δύο ρομπότ. Με τον τρόπο αυτό οι πληροφορίες του ενος ρομπότ μπορούν να μετασχηματιστούν κατάλληλα για να χρησιμοποιηθούν από το άλλο.

Στα [1],[2] χρησιμοποιήθηκαν Color Markers και AR Markers που είχαν τοποθετηθεί πάνω στο UGV, ώστε το UAV μέσω κάμερας να εντοπίζει τόσο την θέση όσο και τον προσανατολισμό του. Με τον τρόπο αυτό παρέχονται σχετικές πληροφορίες για τον χώρο γύρω του, με τα AR Markers να παρέχουν καλύτερα αποτελέσματα, χυρίως λογω των προβλημάτων ανίχνευσης των Color Markers σε περιβάλλον χαμηλής σκίασης και της πιθανής σύγχυσης που μπορεί να εμφανιστεί αν κάποιο αντικείμενο στον χώρο εμφανίσει παρόμοιους χρωματισμούς.

Στο [3] υλοποιείται ένα σύστημα στο οποίο το UAV είναι τοποθετημένο πάνω στο UGV και ενεργοποιείται όταν το UGV απαιτήσει πληροφορίες για τον χώρο γύρω του. Στην συνέχεια το UAV αφού αποστέλει αρκετές πληροφορίες για τον χώρο προσγειώνεται ξανά επάνω στο UGV, απαλείφοντας έτσι τα προβλήματα που προκαλούνται λόγω της μικρής διάρκειας πτήσης των εναέριων ρομπότ. Και εδώ ο εντοπισμός της θέσης του UGV έγινε με την χρήση AR Markers.

Στο [4] προτιμήθηκε μια διαφορετική προσέγγιση, όπου τώρα το UGV ανιχνεύει το UAV. Το UGV μεσω κάμερας εντοπίζει το UAV, το οποίο έχει εξοπλιστεί με 4 Led υψηλής ισχύος στο κάτω μέρος του. Τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν έδειξαν ότι το σύστημα ήταν ικανό να ανιχνεύσει το UAV σε ποικίλες καιρικές συνθήκες.

Στο [5] χρησιμοποιήθηκε το Petri Net Plans Framework για να σχεδιαστεί ένα σύστημα προσγείωσης του UAV πάνω στο UGV, το οποίο όμως μπορεί να διαχειριστεί την περίπτωση σφάλματος λόγω κάποιας διακοπής (π.χ. κάποιο δυναμικό εμπόδιο κάνει αδύνατη την προσγείωση). Ακόμα ένα πλεονέκτημα του συστήματος που αναπτύχθηκε είναι το γεγονός ότι το UGV εξακολουθεί να προσεγγίζει τον στόχο που του έχει ανατεθεί κατά την προσγείωση, κινούμενο βέβαια με μικρότερες ταχύτητες, για να κάνει την προσγείωση του UAV ευκολότερη. Τα πειράματα που διεξάχθηκαν κάνοντας χρήση του ROS και του V-REP έδειξαν ότι το σύστημα λειτουργεί ακόμα και σε περιπτώσεις αναπάντεχων διακοπών.

Στο [6] αναπτύσσεται μια μέθοδος για την ανίχνευση ενός αντικείμενου και από τα δύο ρομπότ ανάμεσα σε παρόμοια αντικείμενα, κανένας η οπτική γωνία που βλέπουν το αντικείμενο το καθιστούν διαφορετικό. Η μέθοδος αυτή απαιτεί το UAV να βρίσκεται πάνω από το αντικείμενο που θέλουμε να ταυτοποιήσει και από τα δύο ρομπότ. Κάνοντας χρήση της μεθόδου Multiple Triangle Representations Matching (MTRM), η οποία κάνει χρήση μια σειράς από μετασχηματισμούς της εικόνας που αποστέλλεται από το UAV (translation, rotation and scaling), ταυτίζει το αντικείμενο που βλέπει με ένα από τα αντικείμενα που υπάρχουν στον οπτικό πεδίο της κάμερας του UGV (Geometric Relation-based Triangle Representation). Η μέθοδος αυτή υπολογίζει έναν συντελεστή μοναδικότητας για κάθε αντικείμενο που βλέπει το UGV και επιλέγεται το αντικείμενο με την μεγαλύτερη μοναδικότητα.

Στο [7] χρησιμοποιούνται οι τεχνικές Image Denoising (median filtering,morphological denoising) , όπου η εικόνα υφίσταται έναν πολυωνυμικό μετασχηματισμό ώστε να ληφθούν οι πραγματικές αποστάσεις των σημείων που απεικονίζονται. Στην υλοποίηση αυτήν, τα εμπόδια προσεγγίζονται ως πολύγωνα και χρησιμοποιείται μία υβριδική μέθοδος για τον εντοπισμό των εμποδίων που βασίζεται στον συνδυασμό του Color Segmentation και Connected Domain Detection. Τέλος, με βάση την τελική πληροφορία που λαμβάνεται από την εικόνα του UAV κατασκευάζεται ο χάρτης στον οποίο θα βασιστεί το Path Planning και δημιουργείται ένας αλγόριθμος που βασίζεται στον συνδυασμό ενός γενετικού αλγορίθμου και Local Rolling Optimization (LRO) βελτιώνοντας έτσι τον υπολογιστικό χρόνο που απαιτείται.

2.2 Ολοκληρωμένες Υλοποιήσεις

Τηράχουν ήδη πολλές υλοποιήσεις συστημάτων που χρησιμοποιούν UGV και UAV για την επίτευξη διαφόρων στόχων. Μια συνηθισμένη στρατηγική εξερεύνησης του χώρου είναι το UAV να πετάει πάνω από το UGV ώστε να του παρέχει πληροφορίες για την καλύτερη επιλογή στόχων Target Selection αλλά και μονοπατιού (Path Planning) .

Στο [8] υπήρξε μια πρώτη υλοποίηση ενός συστήματος για την εξερεύνηση ενός χώρου που αποτελείται από στατικά εμπόδια. Μια ομάδα από 38 UGV και 3 UAV, με προκαθορισμένους ρόλους, χαρτογράφησαν έναν χώρο 3 km^2 . Η ομάδα των UGV κινούνταν αποφεύγοντας εμπόδια κάνοντας χρήση ενός δυναμικού πεδίου γύρω από αυτά. Τα UAV ενεργοποιούνταν όταν σχημάτιζαν μια ευθεία γραμμή που χαρτογραφούσε τον χώρο κινούμενο προς μια σταθερή κατεύθυνση με ίδια ταχύτητα. Τα UAV ενεργοποιούνται από ένα σήμα από την ομάδα των UGV κάθε φορά που βρίσκονται σε μια κοντινή απόσταση από αυτά.

Στο [9] ένα υλοποιήθηκε ένα σύστημα που αποτελείται από 3 ομάδες ρομπότ (Aerial Team,Ground Team and Back-up team) για αποστολές αναζήτησης και διάσωσης. Αρχικά αποστέλλεται η ομάδα των UGV ώστε να μαζέψουν πληροφορίες και να εκτέλεσουν SLAM. Παράλληλα αποστέλλεται και μια ομάδα που αποτελείται από ένα UGV και ένα UAV, η οποία ακολουθεί την προηγούμενη ομάδα και ψάχνει τον χώρο που έχει χαρτογραφηθεί για πιθανούς στόχους. Όταν βρεθεί ένας στόχος τότε η ομάδα των UGV πλησιάζει τον στόχο για να πραγματοποιήσει την διάσωση. Η Back-up team ενεργοποιείται όταν η ομάδα των UGV ζητήσει περισσότερη βοήθεια. Τα πειράματα που διεξήχθησαν έδειξαν ότι το σύστημα χρειάστηκε τον μισό χρόνο από ότι χρειαζόταν μια ομάδα UGV.

Στο [10] οι Michael, Shen και άλλοι, χρησιμοποίησαν ένα σύστημα πολλαπλών πρακτόρων με σκοπό την εξερεύνηση κτιρίου, το οποίο έφερε διάφορες καταστροφές εξαιτίας σεισμού. Το σύστημα που χρησιμοποιήθηκε αποτελείται από τρεις πράκτορες, δύο επίγειους (Kenaf,Quince) και έναν εναέριο (Pelican Quardrotor). Αρχικά εισέρχεται στο χώρο το πρώτο επίγειο (Kenaf) και κάνει εξερεύνηση στο χώρο, αναγνωρίζοντας σημεία τα οποία δεν ήταν προσβάσιμα. Μετά την ολοκλήρωση της εξερεύνησης από το Kenaf, εισέρχεται στο χώρο το δεύτερο επίγειο (Quince), το οποίο διαθέτει μία πλατφόρμα με

σκοπό να μεταφέρει και τον εναέριο πράκτορα, και κατευθύνεται προς τα σημεία του χάρτη που έχουν χαρακτηριστεί ως μη προσβάσιμα. Στη συνέχεια ελευθερώνεται το Pelican Quardrotor ώστε να εξερευνήσει τα σημεία αυτά.

Στο [11] οι Butzke και άλλοι χρησιμοποιώσαν ένα ρομποτικό σύστημα αποτελούμενο από έναν επίγειο και από έναν εναέριο πράκτορα. Όσον αφορά τον εντοπισμό της θέσης του ρομπότ (Localization) προτιμήθηκε καθένας πράκτορας να είναι πλήρως ικανός και να εκτελεί το δικό του SLAM. Αυτό το σύστημα SLAM περιέχει ένα 2-D χάρτη, ώστε να προσδιορίζονται οι θέσεις x,y,θ του πράκτορα. Επιλέχθηκε 2-D χάρτης καθώς παρέχει επαρκή ακρίβεια με σημαντικά χαμηλότερη υπολογιστική ισχύ σε σύγχριση με κάποιο 3-D χάρτη. Στη συνέχεια, για την επικοινωνία μεταξύ των πράκτορων (Communication), κάθε πράκτορας χρησιμοποιούσε την ατομική του μονάδα roscore. Η αποστολή τυπικών μηνυμάτων ROS στους πράκτορες πραγματοποιείται μέσα από το πακέτο ROSCON. Με το τρόπο αυτό σε περίπτωση που εκτελείται το λογισμικό της επικοινωνίας σε άλλο σύστημα, όταν πρέπει μόνο να γίνει ανανέωση στα links του ROSCON πακέτου. Για την πλοήγηση των πράκτορων (Navigation) ο κάθε πράκτορας είναι υπεύθυνος για τη δική του πλοήγηση προς τον εκάστοτε στόχο. Αφού λαμβάνεται κάποιος στόχος, το σύστημα που υπάρχει στον κάθε πράκτορα εκτελεί μία αναζήτηση στο χάρτη των εμποδίων, ώστε να δημιουργεί μία τροχιά προς το στόχο. Για την εύρεση αυτή των στόχων-αντικειμένων (Object Detection), επιλέχθηκε ο κάθε πράκτορας να εκτελεί μία ανάλυση του βίντεο, που παίρνει σαν feedback, ώστε να ανιχνεύει πιθανά OOI που υπάρχουν στο χώρο. Μόλις ανιχνεύεται κάποιο OOI σταματά την κίνηση του, αποστέλλει μία φωτογραφία του OOI στο χρήστη και περιμένει μία απάντηση (Αποδοχή/Απόρριψη), ώστε να συνεχίσει. Για τη συγχώνευση των χαρτών των δύο πράκτορων (Map Merging), προτιμήθηκε αρχικά να υπολογίζεται ο μετασχηματισμός των αρχικών θέσεων μεταξύ των δύο πράκτορων ώστε να τοποθετούνται σωστά τα δεδομένα στον κοινό χάρτη. Τα δεδομένα αυτά χαρακτηρίζονται ως, είτε εμπόδια, είτε ελεύθερος χώρος και απεικονίζονται ως ένα grid που αναπαριστά τα κελιά που είναι εμπόδια, τα κελιά που είναι ελεύθερα και τα κελιά τα οποία δεν έχουν εξερευνηθεί ακόμη.

Στο [12] οι Saska, Krajnik και Preucill ασχολήθηκαν με τη χαρτογράφηση (Mapping) και την πλοήγηση (Navigation) επίγειων και εναέριων ρομπότ. Για τη χαρτογράφηση από το ΥΓΤ αρχικά το ρομπότ καθιδηγείται από ένα χρήστη και δημιουργεί ένα χάρτη. Ο χάρτης αυτός έχει τη μορφή μιας ακολουθίας από επιμέρους τμήματα. Τα τμήματα αυτά περιγράφονται από την αρχική κατεύθυνση α του ρομπότ, όπως δίνεται από την πυξίδα του, το μήκος (s) που μετράται με οδομετρία και από ένα σύνολο ορόσημων (L). Το σύνολο των ορόσημων αποτελείται από κάποιες όψεις που γίνονται αντιληπτές από τις φωτογραφίες που τραβήχτηκαν από την μπροστινή κάμερα του ρομπότ, χρησιμοποιώντας τη μέθοδο SURF (Speeded Up Robust Features). Κάθε τέτοιο ορόσημο περιγράφεται από τις συντεταγμένες της εικόνας και από την απόσταση του ρομπότ από το τμήμα της αρχής H πληροφορία τότε μπορεί να χρησιμοποιηθεί από το σύστημα της πλοήγησης. Κατα τη διάρκεια της αυτόνομης πλοήγησης, το ρομπότ έχει σταθερή γραμμική ταχύτητα μέχρι η υπολογισμένη, μέσω οδομετρίας, απόσταση του ρομπότ και του τμήματος αρχής να είναι ίση με το συνολικό μήκος του τμήματος αυτού. Η γωνιακή ταχύτητα του υπολογίζεται από τις θέσεις που έχουν ήδη χαρτογραφηθεί, και από τα ορόσημα που έχουν εντοπισθεί. Ο εντοπισμός της θέσης (Navigation) του εναέριου πράκτορα είναι βασισμένος σε δύο μεθόδους. Η πρώτη βασίζεται στο drone dead-reckoning σύστημα το οποίο υπολογίζει την ταχύτητα του UAV με βάση την οπτική ροή των εικόνων της κάμερας που βρίσκεται στο κάτω τμήμα του ρομπότ. Η δεύτερη μέθοδος ψάχνει στην εικόνα της κάμερας στο κάτω μέρος του ρομπότ, τα χρωματισμένα μοτίβα που είναι τοποθετημένα στο ελικοδρόμιο του. Έτσι το μοτίβο προσδιορίζει την αρχή και τον προσανατολισμό του συστήματος συντεταγμένων του UAV. Το μειονέκτημα της μεθόδου αυτής είναι ότι απαιτείται πολύ καλή χρωματική αντιστοίχιση, και επίσης υπάρχει η πιθανότητα ανίχνευσης κάποιου μοτίβου το οποίο μοιάζει με το επιθυμητό αλλά δεν είναι. Η πλοήγηση του UAV διαχωρίζεται σε τέσσερις φάσεις: απογείωση, πτήση, επιστροφή στο ελικοδρόμιο και προσγείωση. Κατά

την απογείωση το UAV κρατιέται πάνω από το ελικοδρόμιο, κρατάει δηλαδή σταθερά την κατεύθυνσή του και αυξάνει μόνο το ύψος του. Μόλις φτάσει στο επιμυητό υψόμετρο τότε οι συντεταγμένες x,y,z τροφοδοτούν τους ελεγκτές των συντεταγμένων στροφής και το UAV ξεκινά να κινείται προς τον στόχο του. Μόλις φτάσει στον στόχο του τότε τραβά φωτογραφίες με τις κάμερες και ορίζει ως τον επόμενο στόχο του το σημείο (0,0, z) για να επιστρέψει. Μόλις ανιχνεύσει το μοτίβο του ελικοδρομίου ‘χλειδώνει’ τη θέση του πάνω από το κέντρο του ελικοδρομίου και σταδιακά μειώνει το ύψος του. Μόλις φτάσει σε ύψος 0.3 m από το ελικοδρόμιο απενεργοποιεί τους κινητήρες του και προσγειώνεται.

Στο [13] οι Chae, Yeol Kim και Tae Hong μελετούν ένα σύστημα ενός επίγειου και ενός εναέριου πράκτορα το οποίο διαχειρίζεται ένας χρήστης που βρίσκεται τουλάχιστον 5 km μακριά. Κατά τη διάρκεια της ελεγχόμενης καυνοδήγησης του UGV, διάφοροι αισθητήρες όπως LIDAR και RADAR αναπαράγουν το περιβάλλον ώστε να μπορεί να χρησιμοποιηθεί η πληροφορία αυτή από το χρήστη. Κατά τη διάρκεια της πτήσης του UAV εξατίας της πιθανότητας πρόσκρουσης, χρησιμοποιεί μία συνάρτηση η οποία περιέχει από πριν το μονοπάτι το οποίο θα διανυθεί. Κατά τη διάρκεια κίνησης και τα δύο ρουμπότ μπορούν να ανιχνεύσουν κινούμενα αντικείμενα και να ειδοποιήσουν τον χρήστη. Η ανίχνευση τέτοιων αντικειμένων γίνεται χρησιμοποιώντας θερμικές κάμερες και η μέθοδος που χρησιμοποιείται για αυτό μπορεί να διασπαστεί σε τρεις φάσεις: διαδικασία ελέγχου, ταυτοποίηση κινούμενου αντικειμένου και διαμόρφωση επιλογών ανίχνευσης. Η διαδικασία ελέγχου διανέτει συναρτήσεις για τον ορισμό των συνδέσεων επικοινωνίας, λαμβάνει εικόνες για την ανίχνευση κινούμενων αντικειμένων και επιστρέφει το αποτέλεσμα. Η ταυτοποίηση του κινούμενου αντικειμένου είναι μία μέθοδος διαχωρισμού της πληροφορίας του background της εικόνας και της βασικής πληροφορίας του αντικειμένου. Η διαμόρφωση επιλογών ανίχνευσης λαμβάνει υπόψιν τις παραμέτρους ρυθμίσεων, όπως το ελάχιστο μέγεθος των κινούμενων αντικειμένων, την ταχύτητα με την οποία κινούνται και το βαθμό ευαισθησίας τους.

Στο [14] οι Guerin, Guinand και άλλοι μελέτησαν την περίπτωση όπου ένας εναέριος πράκτορας καυνοδηγεί επίγειους πράκτορες ώστε να πραγματοποίησουν μεταφορά φορτίων για βιομηχανικούς σκοπούς. Στο UGV που θεωρείται αρχηγός δίνονται οι πληροφορίες ώστε να πλοιηγείται μέσα στο χώρο περνώντας από σημεία τα οποία έχουν προσδιοριστεί από το χρήστη. Τα υπόλοιπα UGV θεωρούνται οι ακόλουθοι και συνήθως μεταφέρουν βαρύ φορτίο. Για την πλοιήγηση τους θεωρούν τη θέση του αρχηγού ως στόχο και έτσι τον ακολουθούν κρατώντας μία απόσταση ασφαλείας. Ο ρόλος του UAV είναι, με βάση μίας κάμερας εντοπισμού χρωμάτων που διανέτει, να παρέχει συνολική κάλυψη του χώρου και να βοηθά έτσι τον αρχηγό των UGV να αποφεύγει τα εμπόδια. Πετάει σε συγκεκριμένο ύψος φροντίζοντας ο αρχηγός των UGV να παραμένει πάντα στο κέντρο της εικόνας. Τα σημεία της τροχιάς σημειώνονται πάνω στην εικόνα από το χρήστη και καυνοδηγούν τον αρχηγό.

3

Θεωρητικό Υπόβαθρο - Εργαλεία

Στο παρόν κεφάλαιο όμως παρατεθούν όλα τα εργαλεία που χρησιμοποιήθηκαν για την παρούσα διπλωματική. Επιπλέον όμως παρατεθεί το θεωρητικό υπόβαθρο στο οποίο βασίστηκε η υλοποίηση της εργασίας, όπως αυτή όμως περιγραφεί στο τέταρτο κεφάλαιο.

3.1 ROS - Robot Operating System

Το ROS¹ αποτελεί ένα σύνολο από "software tools & methodologies" τα οποία χρησιμοποιούνται για την υλοποίηση ρομποτικών εφαρμογών. Παρόλο που το ROS δεν αποτελεί ένα λειτουργικό σύστημα, παρέχει παρόμοιες λειτουργίες σε ένα ετερογενές σύνολο υπολογιστών. Το ROS ξεκίνησε το 2007 από το Stanford Artificial Intelligence Laboratory και μετά από 10 χρόνια το ίδρυμα Open Source Robotics Foundation το οποίο κατέχει τα πνευματικά δικαιώματα το μετονόμασε σε Open Robotics.

Το ROS παρέχει ένα μεγάλο αριθμό λειτουργιών όπως hardware abstraction, χαρητηριστικού επιπέδου έλεγχο συσκευών, ανταλλαγή μηνυμάτων μεταξύ των διεργασιών αλλά και διαχείριση πακέτων. Οι διεργασίες που υλοποιούνται στο ROS χρησιμοποιούν αρχιτεκτονική με γράφους όπου η διαχείριση γίνεται μέσω διαφόρων κόμβων. Το ROS εκτελείται συνήθως σε περιβάλλον Linux ή Mac OS και οι γλώσσες που χρησιμοποιούνται είναι κυρίως η C++ και η Python. Μέσα σε αυτά τα 10 χρόνια έχουν υπάρξει διάφορες εκδόσεις του ROS όπως οι Fuerte, Indigo, Kinetic με πιο πρόσφατη την έκδοση Lunar. Στην παρούσα διπλωματική η έκδοση που χρησιμοποιήθηκε είναι η Indigo.

3.1.1 Βασικές έννοιες του ROS

Τα δομικά στοιχεία του ROS αποτελούν:

- **Nodes:** Ένα Node πρόκειται για μία διαδικασία η οποία εκτελεί υπολογισμούς. Μέσα σε ένα σύστημα είναι αναγκαίος ο συνδυασμός ενός πλήθους από nodes τα οποία επικοινωνούν μεταξύ τους ώστε να υλοποιηθεί μία διεργασία. Η χρήση τους παρέχει ποικίλα πλεονεκτήματα, όπως η μείωση της πολυπλοκότητας του κώδικα που χρησιμοποιείται και η μεγαλύτερη ανοχή λαθών, αφού τα λάθη που προκύπτουν απομονώνονται από τα επιμέρους nodes.
- **Messages:** Οι πληροφορίες και τα δεδομένα αποστέλλονται μεταξύ των κόμβων με τη μορφή μηνυμάτων. Ένα μήνυμα διαχωρίζεται σε δύο μέρη, το δυναμικό και το σταθερό. Το δυναμικό κομμάτι αποτελείται από τα δεδομένα που πρόκειται να σταλούν, δηλαδή συνεχώς μεταβάλλονται, ενώ το σταθερό κομμάτι είναι ίδιο για κάθε μήνυμα, και περιέχει όλες εκείνες τις πληροφορίες που είναι απαραίτητες για την αποκαδικοποίηση και κατανόηση του εκάστοτε μηνύματος.

¹<http://wiki.ros.org/ROS/Introduction>

- **Topics:** Τα Topics αποτελούν τους διαύλους, μέσω των οποίων μεταφέρονται τα διάφορα μηνύματα και επικοινωνούν οι κόμβοι (nodes) ενός συστήματος. Το μεγάλο πλεονέκτημα που παρέχουν είναι ότι στο ίδιο topic υπάρχει η δυνατότητα να εγγράψουν και να διαβάσουν παραπάνω από ένας κόμβοι.
- **Services:** Το μοντέλο της εγγραφής/ανάγνωσης δεν είναι κατάλληλο για RPC επικοινωνία η οποία είναι απαραίτητη για τα διανεμημένα συστήματα. Για το λόγο αυτό χρησιμοποιούμε τα ROS services μέσω των οποίων το μοντέλο επικοινωνίας που χρησιμοποιείται είναι της μορφής ερώτησης/απάντησης και για την υλοποίηση χρησιμοποιείται ένα ζεύγος μηνυμάτων (ένα μήνυμα για την ερώτηση και ένα για την απάντηση).
- **Actions:** Έχουν την ίδια μορφή με τα Services, με τη διαφορά ότι αντί για αίτηση και απάντηση αποτελούνται από στόχο goal και αποτέλεσμα result, καθώς επίσης και από ένα επιπλέον πεδίο, την ανατροφοδότηση feedback. Ένα action ζητά με το στόχο μία ενέργεια και επιστρέφει το κατάλληλο αποτέλεσμα όταν αυτή ολοκληρωθεί. Με το επιπλέον πεδίο όμως, την ανατροφοδότηση, παρέχει ενημερώσεις για την εξέλιξη της ενέργειας αλλά και τη δυνατότητα ακύρωσης αυτής.

3.1.2 Πακέτα του ROS

Στη διπλωματική αυτή τα πακέτα που χρησιμοποιήθηκαν στο ROS είναι:

- **tf**²: Είναι ένα πακέτο που επιτρέπει στον χρήστη να παρακολουθεί πολλαπλά συστήματα συντεταγμένων με την πάροδο του χρόνου. Ένα tf διατηρεί τη σχέση μεταξύ των συστημάτων συντεταγμένων σε μια δομή δέντρου ρυθμισμένη με το χρόνο και επιτρέπει στον χρήστη να μετασχηματίζει σημεία, διανύσματα μεταξύ οποιωνδήποτε δύο συστημάτων συντεταγμένων σε οποιοδήποτε επιθυμητό χρονικό σημείο.
- **tum_simulator**³: Το πακέτο αυτό περιέχει τον κώδικα μέσω του οποίου είναι εφικτή η προσομοίωση το AR.Drone 2.0. Οι αισθητήρες, οι κάμερες καθώς και το ίδιο το drone έχουν μοντελοποιηθεί ώστε να πλησιάζουν την πραγματικότητα. Αυτό δίνει τη δυνατότητα εκτέλεσης κώδικα που προορίζεται για το πραγματικό drone σε προσομοίωση του Gazebo, έτσι ώστε να γίνουν γρηγορότερα η δοκιμή (testing) και η αποσφαλμάτωση (debugging) του.
- **ardrone_autonomy**⁴: Το πακέτο αυτό είναι ένα πρόγραμμα οδήγησης (driver), το οποίο μεταφράζει τις εντολές που στέλνονται από το laptop σε εντολές που καταλαβαίνει και εκτελεί το drone. Επιπρόσθετα, μέσω του προγράμματος αυτού γίνονται διαθέσιμα τα δεδομένα των αισθητήρων του drone στους διάφορους κόμβους που τα απαιτούν.
- **ar_pose**⁵: Το πακέτο αυτό λαμβάνει εικόνες από usb_camera topics και μέσω της βιβλιοθήκης ARToolkit δημοσιεύει το μετασχηματισμό της θέσης μεταξύ της κάμερας και των AR Markers που εντοπίζει σε αυτά.

²<http://wiki.ros.org/tf>

³http://wiki.ros.org/tum_simulator

⁴http://wiki.ros.org/ardrone_autonomy

⁵http://wiki.ros.org/ar_pose

- **art_ogmpp**⁶: Μέσω του πακέτου αυτού παρέχεται ένας αλγόριθμος σχεδίασης μονοπατιού χρησιμοποιώντας τον αλγόριθμο A* και Uniform decomposition PRM. Αυτός ο σχεδιασμός μονοπατιού παρέχεται μέσω μίας υπηρεσίας του ROS με το όνομα OGM Path Planning.

Μερικά από τα πακέτα προκαθορισμένων μηνυμάτων προς γενική χρήση, τα οποία χρησιμοποιήθηκαν στην υλοποίηση είναι τα ακόλουθα.

- **geometry_msgs**: Το πακέτο αυτό παρέχει μηνύματα για κοινά γεωμετρικά πρωτότυπα όπως σημεία, διανύσματα και θέσεις. Αυτά τα αρχέτυπα έχουν σχεδιαστεί για να παρέχουν έναν κοινό τύπο δεδομένων και να διευκολύνουν τη διαλειτουργικότητα σε όλο το σύστημα.
- **sensor_msgs**: Μέσω αυτού του πακέτου ορίζονται τα μηνύματα που αφορούν αισθητήρες που χρησιμοποιούνται, όπως για παράδειγμα κάμερα, αισθητήρας laser κλπ.
- **visualization_msgs**: Αποτελείται από ένα σύνολο μηνυμάτων που χρησιμοποιούνται από πακέτα υψηλότερου επιπέδου, όπως το rviz, τα οποία ασχολούνται με συγκεκριμένα δεδομένα απεικόνισης.
- **nav_msgs**: Στο πακέτο αυτό καθορίζονται τα μηνύματα που είναι απαραίτητα για την πλοιήγηση, όπως MapMetaData, OccupancyGrid, Odometry, Path ή το service GetMap.

3.2 GAZEBO

Ένα ακόμη εργαλείο που χρησιμοποιήθηκε για τις προσομοιώσεις είναι η open source πλατφόρμα 3D ρομποτικών προσομοιώσεων GAZEBO⁷. Το GAZEBO αποτελεί έναν Open source προσομοιωτή ο οποίος μπορεί να εκτελεστεί είτε σε συνδυασμό με το ROS, είτε ως standalone εφαρμογή και προσφέρει τη δυνατότητα προσομοίωσης με ακρίβεια και αποτελεσματικότητα των πολλαπλών ρομπότ σε πολύπλοκα εσωτερικά και εξωτερικά περιβάλλοντα.

Για την διεξαγωγή των πειραμάτων χρησιμοποιήθηκε η έκδοση Gazebo 2.2.7, διότι στην έκδοση αυτή είναι συμβατά και τα δύο μοντέλα των ρομποτικών πρακτόρων που επιλέχθηκαν. Τα μοντέλα αυτά χρειάστηκε να τροποποιηθούν για την υλοποίηση και επιπλέον σχεδιάστηκαν εξ ολοκλήρου οι χώροι που εκτελέστηκαν τα πειράματα. Θα γίνει εκτενέστερη περιγραφή αυτών στο πέμπτο κεφάλαιο της υλοποίησης.

Ο ρομποτικός πράκτορας εδάφους που επιλέχθηκε για την υλοποίηση της διπλωματικής αυτής ονομάζεται Turtlebot⁸. Το Turtlebot πρόκειται για ένα ρομποτικό πράκτορα χαμηλού κόστους με open source λογισμικό ο οποίος μπορεί να χρησιμοποιηθεί τόσο για ερευνητικά όσο και για προσωπικά project όπως χαρτογράφηση χώρου, λήψη εικόνων κλπ.

Ο εναέριος ρομποτικός πράκτορας που επιλέχθηκε ώστε να συνεργαστεί με το Turtlebot ονομάζεται Parrot AR Drone 2.0⁹. Στην αρχή σχεδιάστηκε για παιχνίδια σε περιβάλλον εικονικής πραγματικότητας. Ο έλεγχος του γίνεται μέσω smart phone ή από tablet PC. Όμως παρά τον αρχικό σχεδιασμό του, ως παιχνίδι υψηλής τεχνολογίας, το drone κέρδισε το ενδιαφέρον πανεπιστημίων αλλά και ερευνητικών κέντρων και σήμερα χρησιμοποιείται σε πολλές έρευνες στον κλάδο της ρομποτικής. Παρακάτω στο Σχήμα 1 φαίνονται δύο ενδεικτικές φωτογραφίες των δύο ρομποτικών πρακτόρων

⁶https://github.com/etsardou/intelligent_robot_systems_2016/blob/master/documentation

⁷<http://wiki.ros.org/gazebo>

⁸<http://www.turtlebot.com/>

⁹https://en.wikipedia.org/wiki/Parrot_AR.Drone

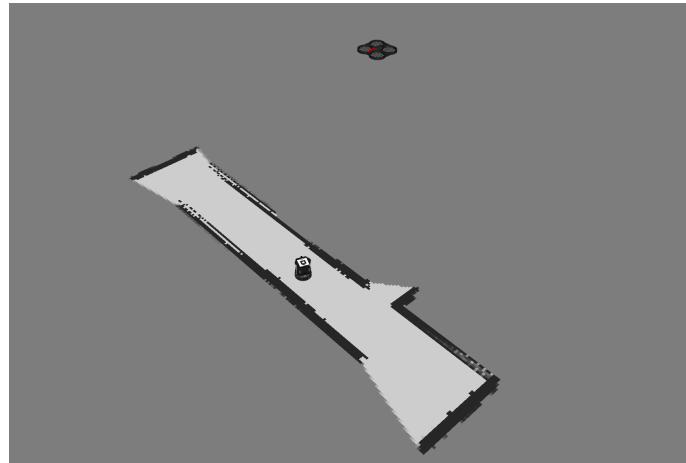


Σχήμα 1: Turtlebot - AR Drone 2.0

3.3 RViz

Το RViz¹⁰ είναι ένα τρισδιάστατο περιβάλλον απεικόνισης το οποίο επιτρέπει την εισαγωγή και των συνδυασμό δεδομένων αισθητήρων, μοντέλων ρομπότ και άλλων τρισδιάστατων δεδομένων σε μία ενιαία απεικόνιση.

Καθώς η προσομοίωση με το Gazebo απαιτεί μεγάλη υπολογιστική ισχύ για την οπτικοποίηση της διεξαγωγής της χαρτογράφησης του χώρου, χρησιμοποιήθηκε ο προσομοιωτής RViz με το Gazebo να εκτελείται χωρίς το γραφικό περιβάλλον του χρήστη. Στο εργαλείο αυτό εμφανίζονται τα μοντέλα των δύο ρομποτικών πρωτότορων καυνώς επίσης και ποικίλες επιπλέον παράμετροι που εκτελούνται κατά τη διάρκεια της προσομοίωσης.



Σχήμα 2: Ενδεικτικό στιγμιότυπο από τα πειράματα στην πλατφόρμα RViz

¹⁰<http://wiki.ros.org/rviz>

3.4 OpenCV

Η OpenCV¹¹ αποτελεί μία open source βιβλιοθήκη η οποία χρησιμοποιείται για computer vision και machine learning. Η βιβλιοθήκη αυτή δημιουργήθηκε με σκοπό να παρέχει μία κοινή δομή για εφαρμογές computer vision και να επιταχύνει τη χρήση της αντίληψη των μηχανών σε εμπορικά προϊόντα.

Η OpenCV έχει περισσότερους από 2500 βελτιστοποιημένους αλγορίθμους, ο οποίος περιλαμβάνει ένα πλήρες σύνολο κλασικών και σύγχρονων υπολογιστικών αλγορίθμων computer vision και machine learning. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την ανίχνευση και αναγνώριση προσώπων, τον εντοπισμό αντικειμένων, την ταξινόμηση των ανθρώπινων ενεργειών σε βιντεο, την παρακολούθηση κινήσεων κάμερας, την παρακολούθηση κινούμενων αντικειμένων, την εξαγωγή τρισδιάστατων μοντέλων αντικειμένων, την παραγωγή πινάκων 3D από στερεοφωνικές κάμερες, εικόνα μιας ολόκληρης σκηνής. Επιπρόσθετα μέσω αυτής υπάρχει η δυνατότητα εύρεσης παρόμοιων εικόνων από μια βάση δεδομένων εικόνων, αφαίρεσης των κόκκινων ματιών από κάποιες εικόνες που που έχουν τραβηγτεί χρησιμοποιώντας το φλας, την παρακολούθηση των κινήσεων των ματιών, την αναγνώριση κάποιου τοπίου καθώς και η δημιουργία δεικτών για το σχεδιασμό τοπίων εικονικής πραγματικότητας κλπ.

Τέλος, η OpenCV μπορεί να χρησιμοποιηθεί σε συνδυασμό με πολλές γλώσσες προγραμματισμού υψηλού επιπέδου, όπως C, C++, Python, Java, Matlab κλπ. και είναι διαθέσιμη για τα λειτουργικά συστήματα Windows, Linux, Android και Mac OS.

Οι δύο συναρτήσεις που χρησιμοποιούνται στην παρούσα υλοποίηση εμφανίζονται παρακάτω.

Η συνάρτηση ProjectPoint¹² της βιβλιοθήκης OpenCV προβάλει σημεία του τρισδιάστατου χώρου στο επίπεδο μίας εικόνας. Τα ορίσματα της είναι:

- ένας πίνακας με τις συντεταγμένες των τρισδιάστατων σημείων που θα προβληθούν στην εικόνα
- οι δύο πίνακες μετασχηματισμού θέσης και στροφής των τρισδιάστατων σημείων σε σχέση με την κάμερα
- ένας πίνακας με τα χαρακτηριστικά της κάμερας
- οι παράμετροι παραμόρφωσης της κάμερας
- οι αναλογίες απεικόνισης της εικόνας

Μετά την εκτέλεση της συνάρτησης επιστρέφεται ο πίνακας με τις μετασχηματισμένες συντεταγμένες των τρισδιάστατων σημείων ως προς το σύστημα συντεταγμένων της κάμερας.

Η συνάρτηση cornerHarris¹³ της βιβλιοθήκης OpenCV χρησιμοποιείται ώστε να εντοπίζονται οι γωνίες σε μία δεδομένη εικόνα. Τα ορίσματα της είναι:

- η εικόνα η οποία θα αναλυθεί για τον εντοπισμό γωνιών
- η διάσταση του μπλοκ μέσα στο οποίο εκτελείται η αναζήτηση
- ένας πίνακας με τα χαρακτηριστικά της κάμερας
- δύο μαθηματικοί παράμετροι οι οποίοι χρησιμοποιούνται για την εκτέλεση του αλγορίθμου

¹¹<https://opencv.org/about.html>

¹²https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#projectpoints

¹³https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html#cornerharris

3.5 SLAM - Simultaneous Localization and Mapping

Το πρόβλημα του SLAM είναι ο συνδυασμός του εντοπισμού θέσης ενός ρομπότ σε ένα χώρο και της χαρτογράφησης του χώρου αυτού. Το ρομπότ δεν ξέρει την θέση του, ούτε την θέση των εμποδίων ωλά πρέπει να εκτιμήσει και τα δύο. Για να εκτελεστεί σωστά η εκτίμηση των θέσεων του οχήματος και να κατασκευαστεί ένας πιστός με την πραγματικότητα χάρτης, χρειάζεται να ακολουθήσουμε τεχνικές SLAM. Για την αναπαράσταση του χώρου, υπάρχουν δύο είδη SLAM, Volumetric και feature-based SLAM. Στο Volumetric SLAM κάθε κελί της αναπαράστασης (2D ή 3D) αναπαριστά ένα συγκεκριμένο τμήμα του χώρου και την πιθανότητα κατάληψή του, ενώ στο feature-based SLAM αναπαρίστανται μόνο τα landmarks των οποίων την εκτίμηση θέσης διατηρούμε. Στην παρούσα διπλωματική εργασία επιλέχθηκε το Volumetric SLAM, καθώς το Turtlebot ήταν εξοπλισμένο με αισθητήρα μέτρησης απόστασης Lidar.

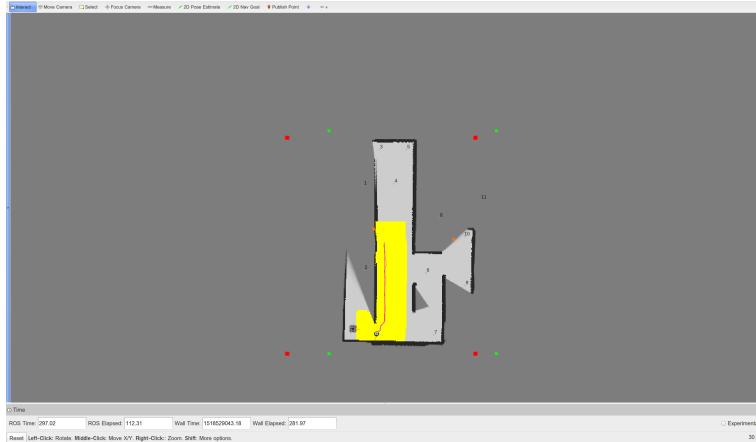
Το CRSIM SLAM¹⁴ αποτελεί μία βελτιωμένη μέθοδο SLAM όπου τα βασικά της χαρακτηριστικά είναι:

- Χρησιμοποιεί scan-to-map αντιστοίχηση αντί για τη συνηθισμένη αντιστοίχιση scan-to-scan, με στόχο τη μείωση της συσσώρευσης θορύβου.
- Η αντιστοίχιση σάρωσης πραγματοποιείται μέσω ενός Random Restart Hill Climbing algorithm (RRHC)
- Μόνο οι κρίσιμες ακτίνες συμμετέχουν στο RRHC. Ως κρίσιμες σημειώνονται οι ακτίνες οι οποίες περιέχουν περισσότερη χωρική πληροφορία από τις άλλες.
- Η ενημέρωση του χάρτη εκτελείται με δυναμικό τρόπο, ανάλογα με την τρέχουσα περιβαλλοντική δομή.

3.6 OGM - Occupancy Grid Map

Ο χάρτης αυτός είναι το αποτέλεσμα μετά το τέλος της εκτέλεσης κάποιου SLAM αλγορίθμου. Πρόκειται για ένα χάρτη ο οποίος αποτελείται από τετραγωνικά κελιά ίδιου μεγέθους. Κάθε κελί αποτελεί μία θέση στο χάρτη και χαρακτηρίζεται από τρεις τιμές. Το κελί μπορεί να έχει τιμή μεταξύ 0 και 1 ανάλογα με την πιθανότητα κατάληψης ή όχι της θέσης αυτής από κάποιο εμπόδιο. Η τιμή 0 αντιστοιχεί σε κελί όπου το ρομπότ γνωρίζει ότι η θέση αυτή αντιστοιχεί σε ελεύθερο χώρο, η τιμή 1 αντιστοιχεί στο κελί αν στη θέση βρίσκεται κάποιο εμπόδιο και η τιμή 0.5 αντιστοιχεί σε όλα τα κελιά που δεν έχουν εξερευνηθεί ακόμη. Παρακάτω στο Σχήμα 3 φαίνεται ο OGM που δημιουργήθηκε κατά την εκτέλεση ενός πειραμάτων.

¹⁴http://wiki.ros.org/crsm_slam



Σχήμα 3: Ενδεικτικό παράδειγμα OGM χάρτη που εξήχθη από τα πειράματα

3.7 Αλγόριθμος Brushfire

Ο αλγόριθμος Brushfire¹⁵ πρόκειται για έναν αλγόριθμο εξάπλωσης σε κάποιο γράφο ή grid. Συγκεκριμένα, χρησιμοποιείται για τη δημιουργία δυναμικών πεδίων μέσα σε κάποιο χώρο. Τα βήματα από τα οποία αποτελείται ο αλγόριθμος αυτός είναι:

- Εκκίνηση από ένα set σημείων
- Εξάπλωση στους γείτονες υπό συνθήκη
- Ανάθεση τιμής στους γείτονες
- Τερματισμός εξάπλωσης υπό συνθήκη

3.8 PID Controller

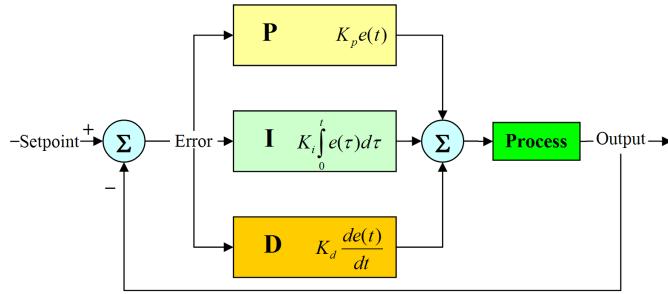
Ένα σημαντικό μέρος της εργασίας αποτέλεσε ο σχεδιασμός και η υλοποίηση ενός PID¹⁶ ελεγκτή θέσης. Η μαθηματική εξίσωση του ελεγκτή αυτού είναι:

$$u(t) = u_p(t) + u_i(t) + u_d(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3.1)$$

Ο όρος u_p αποτελεί τον αναλογικό (proportional) όρο, και αντίστοιχα οι όροι u_i , u_d αποτελούν τον ολοκληρωτικό (integral) και τον διαφορικό (derivative) όρο. Οι όροι K_p , K_i , K_d είναι μη-αρνητικές μεταβλητές οι οποίες ονομάζονται κέρδη του ελεγκτή, ενώ ο όρος $e(t)$ αντιπροσωπεύει το σφάλμα μεταξύ της πραγματικής και της επιθυμητής τιμής του μεγέθους που μετράται.

¹⁵https://www.cs.cmu.edu/~motionplanning/lecture/Chap4-Potential-Field_- howie.pdf

¹⁶https://en.wikipedia.org/wiki/PID_controller



Σχήμα 4: Γραφική απεικόνιση ενός PID ελεγκτή

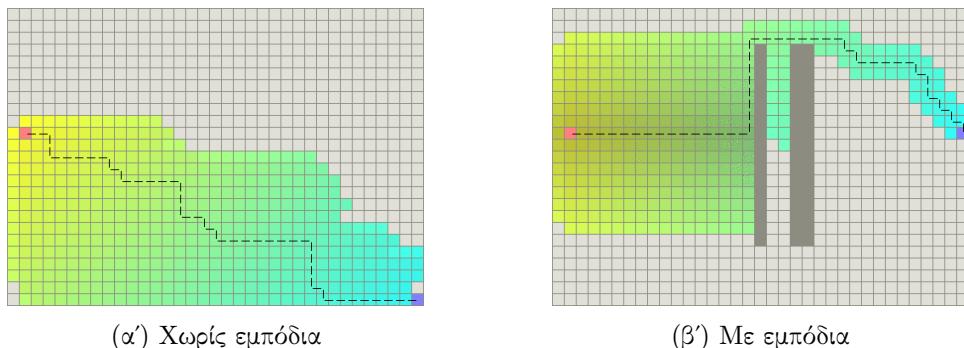
3.9 Αλγόριθμος A*

Ο αλγόριθμος αναζήτησης βέλτιστου μονοπατιού που χρησιμοποιείται από το UGV είναι ο A* αλγόριθμος. Ο συγκεκριμένος αλγόριθμος αποτελεί τη βελτιωμένη έκδοση του αλγορίθμου Dijkstra και προκύπτει από το συνδυασμό των δύο ειδών αλγορίθμων αναζήτησης μονοπατιού, τον Depth First αλγόριθμο και τον Best First αλγόριθμο. Το βασικό στοιχείο αυτού του αλγορίθμου είναι ότι αποφέύγει να επεκταθεί σε μονοπάτια με υψηλό κόστος βάρους.

Συγκεκριμένα, ο A* επιλέγει τη διαδρομή που ελαχιστοποιεί την εξίσωση:

$$f(n) = g(n) + h(n) \quad (3.2)$$

όπου n είναι ο τελευταίος κόμβος στο μονοπάτι, g(n) είναι το κόστος της διαδρομής από τον αρχικό κόμβο μέχρι τον n και h(n) είναι μια heuristic που υπολογίζει το κόστος της φθηνότερης διαδρομής από τον κόμβο n στον στόχο. Για να βρει ο αλγόριθμος τη συντομότερη διαδρομή, θα πρέπει η heuristic συνάρτηση να είναι παραδεκτή (admissible), πράγμα που σημαίνει ότι ποτέ δεν υπερεκτιμά το πραγματικό κόστος προκειμένου να πάει στον πλησιέστερο κόμβο - στόχο. Παρακάτω παρατίθενται δύο ενδεικτικά σχήματα στο Σχήμα 5για την εξάπλωση του A* καθώς και ο ψευδοχώδικας του αλγορίθμου αυτού.



Σχήμα 5: Εύρεση βέλτιστου μονοπατιού με χρήση του αλγορίθμου A*

Algorithm 1 A* Algorithm

```

1: Put  $n_{start}$  in the OPEN list with  $f(n_{start}) = h(n_{start})$ 
2: while the OPEN list is not empty do
3:   Take from the OPEN list the node  $n_{current}$  with the lowest  $f(n_{current}) = g(n_{current}) + h(n_{current})$ 
4:   if  $n_{current} = n_{goal}$  then
5:     EXIT
6:   end if
7:   Generate each state  $n_{successor}$  that comes after  $n_{current}$ 
8:   for each  $n_{successor}$ , of  $n_{current}$  do
9:     Set successor_current_cost =  $g(n_{current}) + w(n_{current}, n_{successor})$ 
10:    if  $n_{successor}$  is in the OPEN list then
11:      if  $n_{successor} \leq$  successor_current_cost
12:        continue (to line 20)
13:      end if
14:    else if  $n_{successor}$  is in the CLOSED list then
15:      if  $n_{successor} \leq$  successor_current_cost then
16:        continue (to line 20)
17:      end if
18:    Move  $n_{successor}$  from CLOSED list to the OPEN list
19:    else
20:      Add  $n_{successor}$  to the OPEN list
21:      Set  $h(n_{successor})$  to be the heuristic distance to  $n_{goal}$ 
22:    end if
23:     $g(n_{successor}) =$  successor_current_cost
24:    Set the parent of  $n_{successor}$  to  $n_{current}$ 
25:  end for
26:  Add  $n_{current}$  to the CLOSED list
27:  Remove  $n_{current}$  from OPEN list
28: end while
29: if  $n_{current} \neq n_{goal}$  then
30:   the OPEN list is empty
31:   return failure
32: end if
```

3.10 Uniform Sampling Path Planning

To Uniform Sampling Path Planning πρόκειται για μία υλοποίηση της μεθοδολογίας PRM - Probabilistic RoadMaps η οποία χρησιμοποιείται για τη δημιουργία ενός μονοπατιού μέσα σε ένα χώρο. Το χαρακτηριστικό της είναι ότι τα δείγματα είναι ομοιόμορφα κατανεμημένα μέσα στον ελεύθερο χώρο. Η υλοποίηση της μεθόδου αυτής απεικονίζεται με τη μορφή φευδοκώδικα παρακάτω και περιγράφεται αναλυτικά στην συνέχεια.

Algorithm 2 Uniform Sampling

```

1: for  $i = x_{min}$ , while  $i < x_{max}$ ,  $i = i + STEP$  do
2:   for  $i = y_{min}$ , while  $i < y_{max}$ ,  $i = i + STEP$  do
3:     if  $[i, j]$  in free space then
4:        $Add[i, j]$  in free space
5:       if  $[i - STEP, j]$  in graph then
6:         Connect  $[i, j], [i - STEP, j]$ 
7:       end if
8:       if  $[i - STEP, j - STEP]$  in graph then
9:         Connect  $[i, j], [i - STEP, j - STEP]$ 
10:      end if
11:      if  $[i, j - STEP]$  in graph then
12:        Connect  $[i, j], [i, j - STEP]$ 
13:      end if
14:      if  $[i - STEP, j + STEP]$  in graph then
15:        Connect  $[i, j], [i - STEP, j + STEP]$ 
16:      end if
17:    end if
18:  end for
19: end for

```

Κατά το πρώτο βήμα εκτελείται ομοιόμορφη δειγματοληψία σε όλο το διαθέσιμο χώρο. Εν συνεχείᾳ, αποκλείονται τα σημεία τα οποία καλύπτονται από κάποιο εμπόδιο και τοποθετούνται το αρχικό και το τελικό σημείο του μονοπατιού. Για κάθε σημείο που έχει προκύψει από τη δειγματοληψία ενώνεται με όλους τους γείτονες του που βρίσκονται σε ελεύθερο χώρο και έτσι δημιουργείται ένα δίκτυο σημείων. Τέλος εκτελείται ο αλγόριθμος αναζήτησης A* που περιγράφηκε παραπάνω ο οποίος επιλέγει το βέλτιστο μονοπάτι.

Τα βασικά πλεονεκτήματα της δειγματοληψίας αυτής είναι:

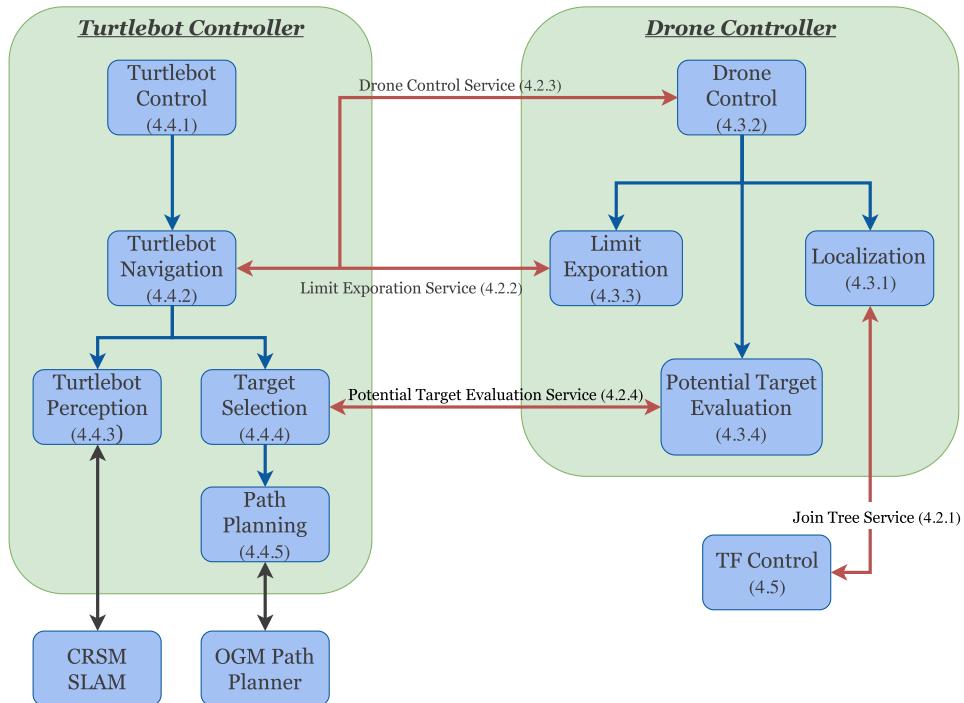
- Χαμηλοί χρόνοι εκτέλεσης
- Μεγάλο ποσοστό επιτυχίας
- Σταθερή μέθοδος
- Το μήκος του μονοπατιού είναι σχεδόν βέλτιστο

4

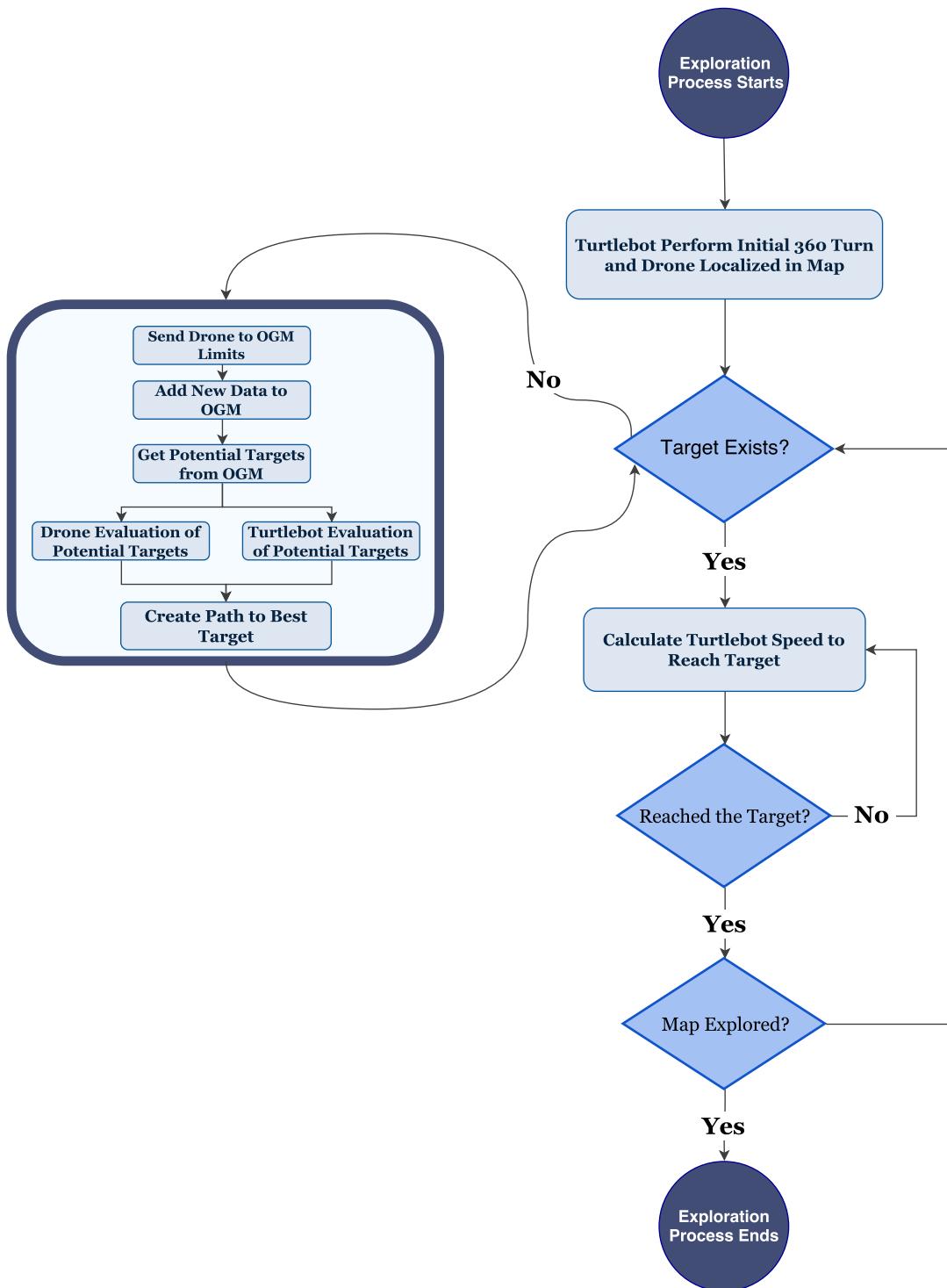
Τυλοποίηση

Στον παρόν κεφάλαιο θα περιγραφούν αναλυτικά όλες οι διαδικασίες που υλοποιήθηκαν με σκοπό την επίτευξη του στόχου της διπλωματικής αυτής, όπως τέθηκε στο πρώτο κεφάλαιο. Αρχικά θα αναφερθεί η τροποποίηση των ρομποτικών μοντέλων που πραγματοποιήθηκε και στην συνέχεια του κεφαλαίου θα γίνει αναλυτική περιγραφή των δύο Controller καθώς και του τρόπου επικοινωνίας μεταξύ τους.

Η βασική δομή της αρχιτεκτονικής του συστήματος καθώς και μια συνοπτική περιγραφή λειτουργίας αυτού παρουσιάζονται στα Σχήματα 6 και 7 αντίστοιχα.



Σχήμα 6: Βασική Δομή Πακέτων της Τυλοποίησης



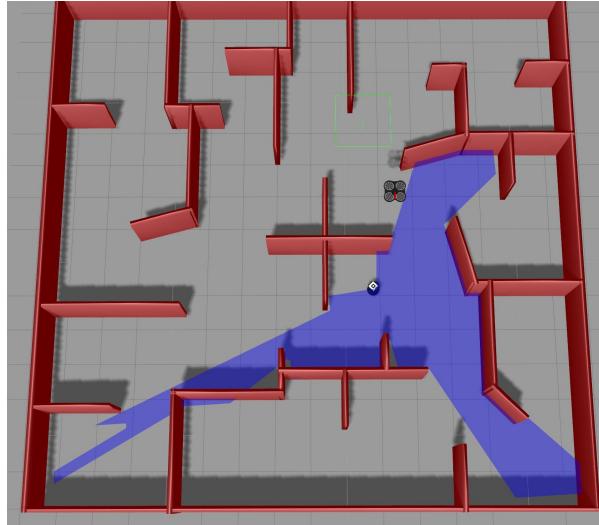
Σχήμα 7: Συνοπτικό Διάγραμμα Ροής Υλοποίησης

4.1 Τροποποίηση Ρομποτικών Μοντέλων στο GAZEBO

Για την εκτέλεση του CRSIM SLAM από το Turtlebot ήταν απαραίτητη η κατάλληλη ενσωμάτωση του δισδιάστατου αισθητήρα Hokuyo LIDAR στο προσομοιωτικό του μοντέλο στο GAZEBO. Επιπλέον, για τον εντοπισμό του επίγειο ρομποτικού πράκτορα από τον εναέριο, ήταν αναγκαία η τοποθέτηση ενός AR Marker στην επάνω επιφάνεια του Turtlebot. Για την ενσωμάτωση του μοντέλου του AR Marker σε αυτό του Turtlebot δημιουργήθηκαν τα παρακάτω XML αρχεία.

- **kobuki_hexagons_hokuyo_tag.urdf.xacro:** Το αρχείο αυτό δημιουργήθηκε στην τοποθεσία turtlebot_description/robots και περιέχει αναφορές στα ξεχωριστά αρχεία που συντελούν στην πλήρη περιγραφή του μοντέλου που χρησιμοποιήθηκε για την εφαρμογή της μεθόδου CRSIM SLAM και την τοποθέτηση του AR Marker στο GAZEBO.
- **hokuyo.urdf.xacro:** Το αρχείο αυτό δημιουργήθηκε στην τοποθεσία “turtlebot_description/urdf/sensors” και περιλαμβάνει την περιγραφή του αισθητήρα Hokuyo για το μοντέλο που εφαρμόζει τη μέθοδο CRSIM SLAM. Σε αυτό το αρχείο ορίζονται οι τεχνικές παράμετροι για τη λειτουργία του αισθητήρα, οι σημαντικότερες των οποίων είναι ο αριθμός των ακτίνων που παράγονται, η ελάχιστη και μέγιστη τιμή του εύρους τους (τόξο κύκλου) σε ακτίνια, καθώς και η ελάχιστη και μέγιστη τιμή του μήκους εκπομπής μιας ακτίνας. Το εύρος των ακτίνων του αισθητήρα για το συγκεκριμένο μοντέλο ορίστηκε ίσο με 260 μοίρες.
- **ar_tag.urdf.xacro:** Το αρχείο αυτό δημιουργήθηκε στην τοποθεσία “turtlebot_description/urdf/tags” το οποίο περιέχει την περιγραφή του AR Marker.

Τέλος τροποποιήθηκε το μοντέλο του turtlebot στο GAZEBO (turtlebot_description/urdf/turtlebot_gazebo.urdf.xacro) προσθέτοντας τους κατάλληλους συνδέσμους ώστε να ενσωματωθούν τα παραπάνω.



Σχήμα 8: Ενδεικτικό Στιγμιότυπο Τροποποιημένων Μοντέλων των Ρομποτικών Πρακτόρων στο Gazebo

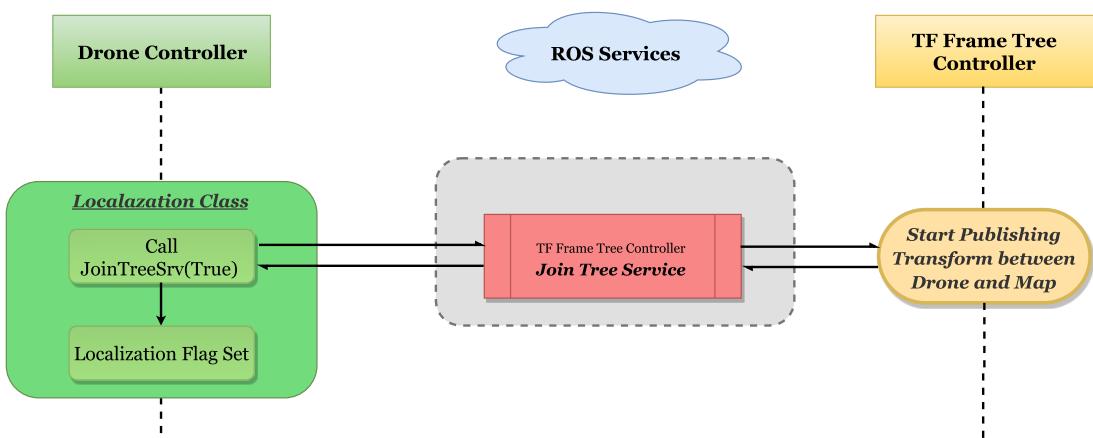
4.2 Επικοινωνία των δύο Πρακτόρων

Για την συνεργασία των δύο πρακτόρων είναι απαραίτητη η χρήση των κατάλληλων ROS Topics και ROS Services ώστε να μπορούν να ανταλλάσσουν πληροφορίες μεταξύ τους. Ο τρόπος με τον οποίο ανταλλάζουν πληροφορίες τα δύο ρομπότ θα πρέπει να γίνει με τέτοιον τρόπο ώστε η διεργασία που εκτελεί ο κάθε πράκτορας ξεχωριστά την συγκεκριμένη χρονική στιγμή, να διακόπτεται όσο τον δυνατόν λιγότερο. Με τον τρόπο αυτό η συνεργασία των δύο πρακτόρων πραγματοποιείται με τον πιο αποδοτικό τρόπο. Στην συνέχεια της ενότητας παρουσιάζονται εκτενέστερα τα μηνύματα καθώς και οι υπηρεσίες που δημιουργήθηκαν για την επίτευξη της επικοινωνίας που αναλύθηκε παραπάνω.

4.2.1 Join Tree Service

Στην έναρξη της προσομοίωσης οι δύο πράκτορες έχουν δύο ξεχωριστά συστήματα συντεταγμένων. Με τον εντοπισμό της θέσης του Drone ως πρός το χάρτη, γίνεται δυνατή η σύνδεση των συστημάτων αυτών. Για τον λόγο αυτό δημιουργήθηκε η εν λόγω υπηρεσία. Η δημιουργία της υπηρεσίας αυτής γίνεται σε ένα ανεξάρτητο ROS Node που είναι υπεύθυνο για την ένωση των δύο συστημάτων συντεταγμένων με τον χάρτη, όπως θα παρουσιαστεί λεπτομερώς στο τέλος του κεφαλαίου. Μέσω της ενεργοποίησης αυτής γίνεται αναμετάδοση της θέσης και του προσανατολισμού του συστήματος συντεταγμένων του πράκτορα στο σύστημα συντεταγμένων του χάρτη (σταθερό σύστημα συντεταγμένων).

Ο Drone Controller, που θα παρουσιαστεί στην συνέχεια, καλεί την υπηρεσία μέσω της κλάσης που υλοποιεί τον εντοπισμό της θέσης του πράκτορα στον χώρο (Localization Class), εφόσον έχει εντοπιστεί το Ar Marker που βρίσκεται στην επάνω επιφάνεια του Turtlebot. Η θέση και ο προσανατολισμός επομένως του Drone ως προς τον χάρτη γίνονται διαθέσιμα την στιγμή που βρίσκεται το Ar Marker, αφού η σχετική θέση των πρακτόρων είναι πλέον γνωστή και η θέση του Tutlebot στον χάρτη είναι διαθέσιμη (ο χάρτης κατασκευάζεται ως προς το Tutlebot). Τέλος, αναφέρεται ότι η υπηρεσία αυτή επιστρέφει έναν Boolean (True/False) ανάλογα με το αν η σύνδεση των δύο δέντρων ήταν επιτυχής ή όχι (αντίστοιχα). Για την καλύτερη κατανόηση της όλης διαδικασίας παρουσιάζεται σε μορφή διαγράμματος ροής στο Σχήμα 9.



Σχήμα 9: Διάγραμμα Ροής Κλήσης την Υπηρεσίας Join Tree

4.2.2 Limit Exploration Service

Όπως έχει αναφερθεί και στον Κεφάλαιο 1, το Drone θα παρέχει πληροφορίες για τα όρια του OGM. Για την επίτευξη αυτής της λειτουργίας κατασκευάστηκε η συγκεκριμένη υπηρεσία καθώς επίσης και ένα ROS Topic για την αποστολή των πληροφοριών που συλλέχθηκαν (εικόνα από τα όρια του χάρτη).

Όσον άφορα την δημιουργία της υπηρεσίας, αυτή δημιουργείται μέσα στην κλάση Limit Exploration του Drone Controller και δέχεται ως ορίσματα 2 αριθμούς κινητής υποδιαστολής διπλής ακρίβειας 64-bit (Float64) που αντιστοιχούν στις συντεταγμένες (x, y) του σημείου που πρέπει να εξερευνηθεί. Αν το σημείο που λαμβάνει η υπηρεσία είναι έγκυρο (αντιστοιχεί σε σημείο του χάρτη) τότε η υπηρεσία επιστρέφει αληθή τιμή (True), αφού το Drone ταξιδέψει σε αυτό, ειδάλλως επιστρέφεται λανθάνουσα (False).

Όπως αναφέρθηκε και προηγουμένως, η αποστολή της πληροφορίας από τα όρια του χάρτη αποστέλλεται στο Turtlebot Controller μέσω ενός ROS Topic. Με τον τρόπο αυτό η επικοινωνία των δυο πρακτόρων για την επίτευξη τόσο της διαδικασίας εξερεύνησης των ορίων του χάρτη, όσο και της αποστολής της πληροφορίας που λήφθηκαν για αυτά, γίνεται έτσι ώστε να επηρεάζεται στον μικρότερο δυνατό βαθμό η ροή εκτέλεσης που ακολουθεί ο Turtlebot Controller.

Η συνολική διαδικασία επομένως για την εξερεύνηση των ορίων του OGM περιγράφεται ως εξής:

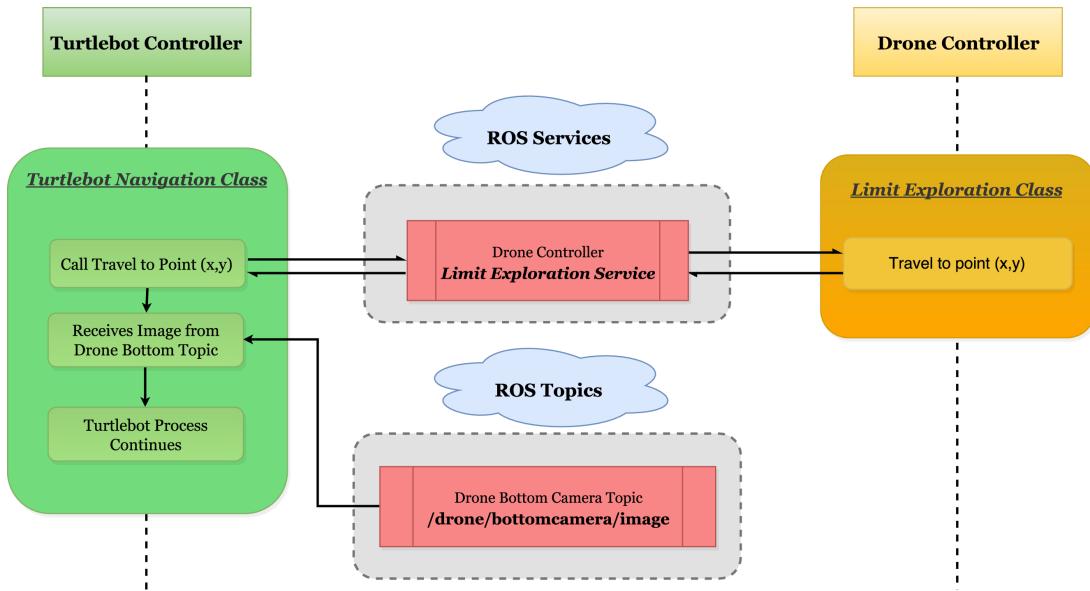
- (A) Το Turtlebot ενημερώνει το Drone ότι απαιτείται η εξερεύνηση ενός σημείου στον χάρτη μεσώ της κλήσης της υπηρεσίας Limit Exploration.
- (B) Η υπηρεσία ενεργοποιεί την αντίστοιχη σημαία στον Drone Controller για την επίτευξη της διαδικασίας, εφόσον το σημείο είναι αποδεκτό, ειδάλλως η υπηρεσία τερματίζεται επιστρέφοντας False.
- (C) Το Drone ξεκινάει να ταξιδεύει στο σημείο ενδιαφέροντος. Με την άριξη στο σημείο που ζητήθηκε, η υπηρεσία τερματίζεται ενημερώνοντας τον Turtlebot Controller μέσω της επιστρέφουσας τιμής True ότι πραγματοποιήθηκε η ενέργεια που ζητήθηκε.
- (D) Το Turtlebot αφού ενημερώθηκε για την επιτυχή ολοκλήρωση της διεργασίας, αναμένει τις πληροφορίες για την λήψη των πληροφοριών που ζήτησε στο αντίστοιχο ROS Topic, και συνεχίζει την εκτέλεση των διεργασιών του.

Μια σχηματική απεικόνιση της όλης διαδικασίας που αναλύθηκε φαίνεται και εποπτικά στο Σχήμα 10.

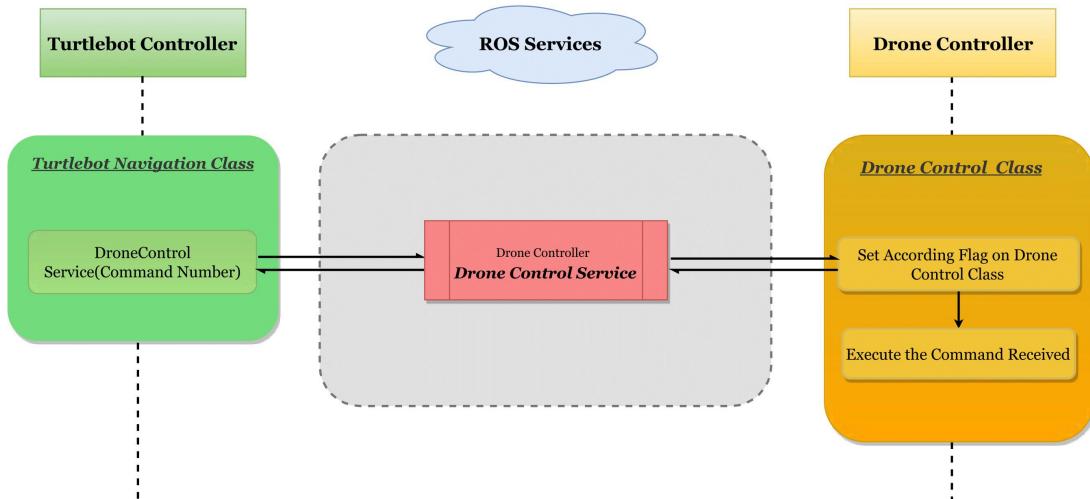
4.2.3 Drone Control Service

Ο σκοπός που δημιουργήθηκε η συγκεκριμένη υπηρεσία είναι για να καταστεί δυνατός ο έλεγχος του Drone από το ROS Node του Turtlebot Controller, που θα αναλυθεί σε επόμενη ενότητα του κεφαλαίου. Μέσω της υπηρεσίας αυτής ο Turtlebot Controller μπορεί να δώσει εντολή στο Drone να απογειωθεί, να προσγειωθεί, να ακολουθήσει την κίνηση του Turtlebot (με την παρακολούθηση του Ar Marker) καθώς και να αυξομειώσει το ύψος πτήσης του.¹⁷ Η διαδικασία αποστολής μιας εντολής εκτέλεσης κάποιας βασικής λειτουργίας του Drone από τον Turtlebot Controller παρουσιάζεται και γραφικά στο Σχήμα 11.

¹⁷Για περισσότερες λεπτομέρειες ανατρέξτε στον Πίνακα 1



Σχήμα 10: Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρωτόρων για την Λήψη Πληροφοριών για τα Όρια του OGM



Σχήμα 11: Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρωτόρων για την εκτέλεση βασικών λειτουργιών του Drone

4.2.4 Potential Targets Evaluation Service

Ένας περαιτέρω τρόπος υποβοήθησης του Turtlebot, που αναφέρθηκε και στο Κεφάλαιο 1, είναι η αξιολόγηση των πιθανών στόχων που έχουν επιλεγεί από το Drone. Η υλοποίηση της λειτουργίας αυτής χρειάστηκε και εδώ την δημιουργία μιας υπηρεσίας και ενός ROS Topic για την αποστολή των αποτελεσμάτων αξιολόγησης. Επιπροσθέτως ήταν απαραίτητη η κατασκευή ενός μηνύματος (ROS Message) για επιστροφή της αξιολόγησης στο Turtlebot. Η δομή του μηνύματος φαίνεται αναλυτικά στο Σχήμα 12.



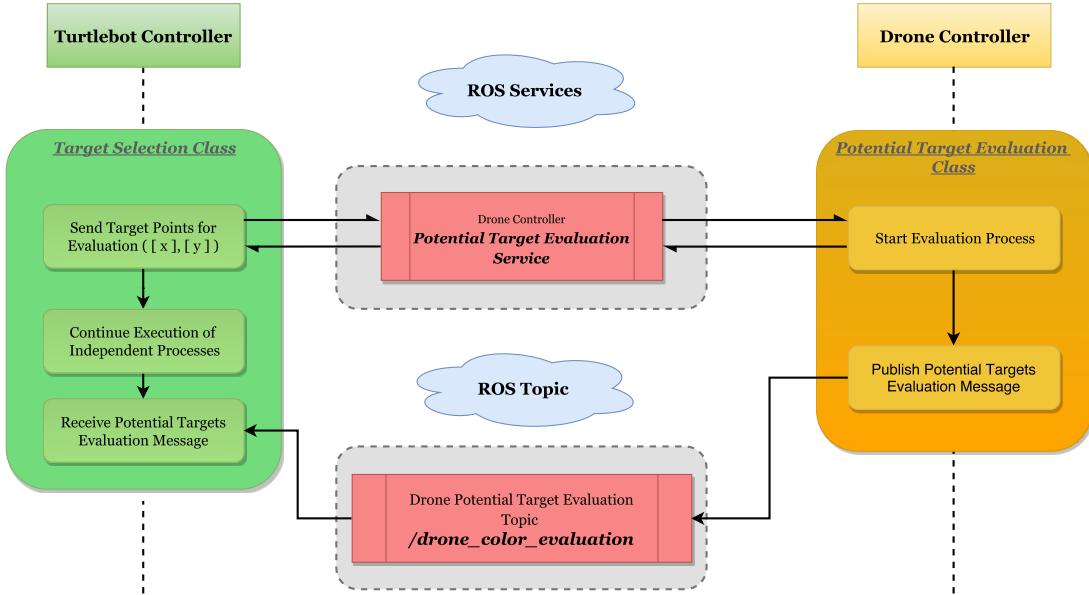
Σχήμα 12: Μήνυμα που δημιουργήθηκε για την αποστολή της αξιολόγησης των πιθανών στόχων από το Drone

Η υπηρεσία αυτή δέχεται ως ορίσματα δύο μονοδιάστατους πινάκες που τα στοιχεία αυτών ανά ζεύγη αντιστοιχούν στις συντεταγμένες (x, y) των σημείων που πρέπει να αξιολογηθούν. Αν τα σημεία που λήφθηκαν περιέχουν έγκυρες τιμές (αντιστοιχούν σε σημεία του χάρτη), τότε η υπηρεσία επιστρέφει αληθή τιμή (True), ειδάλλως επιστρέφεται λανθάνουσα (False). Για τους λογούς που αναλύθηκαν και στην περίπτωση της υπηρεσίας Limit Exploration που μελετήθηκε στην προηγούμενη υποενότητα, τα αποτελέσματα της αξιολόγησης αποστέλλονται στο Turtlebot Controller μέσω ενός ROS Topic.

Για την αξιολόγηση των πιθανών στόχων από το Drone ακολουθεί τα παρακάτω βήματα:

- Αποστολή των σημείων που φέρουν υπό αξιολόγηση από το Turtlebot στο Drone, μεσω της αλήσης της υπηρεσίας Potential Targets Evaluation.
- Η υπηρεσία ενεργοποιεί την αντίστοιχη σημαία στον Drone Controller για την επίτευξη της διαδικασίας αξιολόγησης και στη συνέχεια επιστρέφει αν υπηρεσία ενεργοποιήθηκε επιτυχώς ή όχι σε μορφή True-False στον Turtlebot Controller.
- Το Drone ξεκινάει να αξιολογεί τα σημεία αυτά κατασκευάζοντας τον Πινάκα Αξιολόγησης (στην μορφή του μηνύματος του Σχήματος 12) και θα τον αποστέλλει μεσω ενός ROS Topic.
- Το Turtlebot ενημερώθηκε για την επιτυχή έναρξη της διεργασίας και περιμένει από το αντίστοιχο ROS Topic για την λήψη των αποτελεσμάτων αξιολόγησης, ενώ παράλληλα συνεχίζει την εκτέλεση των ανεξάρτητων διεργασιών του.

Εποπτικά η διαδικασία που αναλύθηκε στα παραπάνω βήματα παρουσιάζεται στο Σχήμα 13.

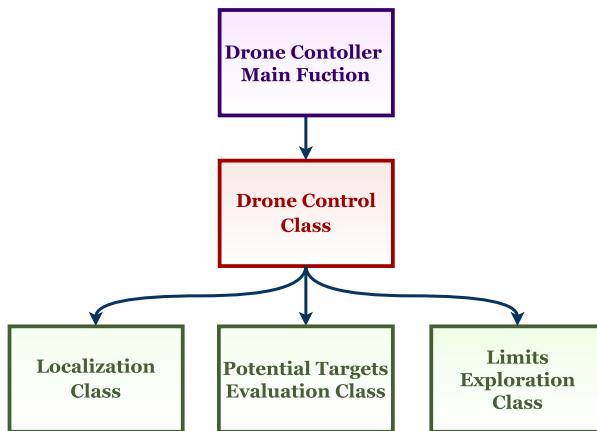


Σχήμα 13: Γραφική Αναπαράσταση Επικοινωνίας των Δυο Πρακτόρων για την Αξιολόγηση των Πιθανών Στόχων από το Drone

4.3 Πακέτο του Drone Controller

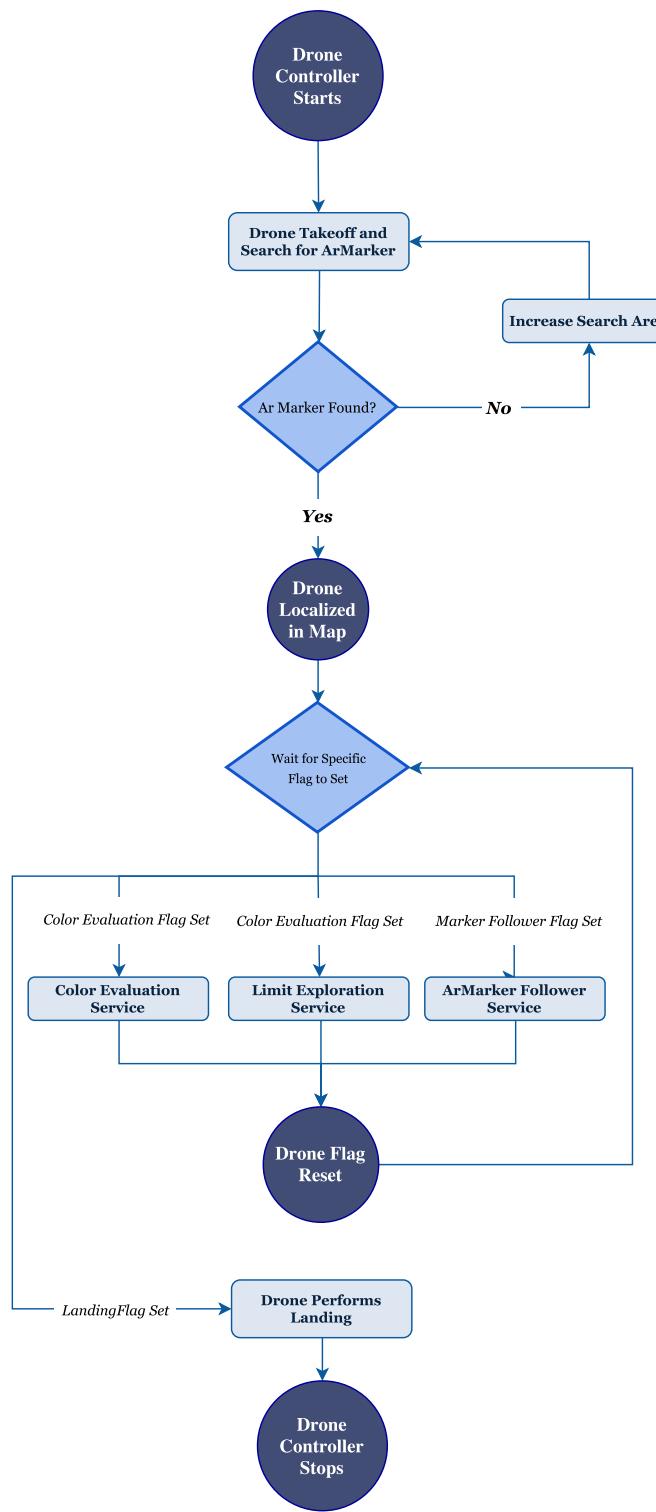
Το πακέτο του Drone Controller που υλοποιήθηκε αποτελείται από τέσσερα υποπακέτα, καθένα από τα οποία αποτελεί μια ξεχωριστή κλάση όπως αυτήν ορίζεται στον δομημένο προγραμματισμό της Python. Περιληπτικά, τα υποπακέτα αυτά απεικονίζονται γραφικά στο Σχήμα 14 και είναι τα εξής:

- **Drone Control:** Πακέτο Υλοποίησης Βασικών Κινήσεων του Drone.
- **Potential Targets Evaluation:** Πακέτο που αξιολογεί τους υποψήφιους στόχους που έχουν επιλεγεί μέσω της διαδικασίας του Target Selection του Turtlebot.
- **Limit Exploration:** Πακέτο για την εξερεύνηση των ορίων του χάρτη που έχει ήδη κατασκευαστεί από το Turtlebot.
- **Localization:** Πακέτο υπεύθυνο για τον εντοπισμό της θέσης του Drone στον χάρτη.



Σχήμα 14: Βασική Δομή Πακέτων του Drone Contoller

Στο Σχήμα 15 εμφανίζεται το βασικό διάγραμμα ροής του Drone Controller. Με την έναρξη του Drone Controller, το Drone απογειώνεται σε ένα προκαθορισμένο ύψος ώστε να αρχίσει να εξερευνεί τον χώρο για το Ar Marker. Ακολουθεί μια αναζήτηση στον χώρο μέχρι το Drone να βρεί το Ar Marker και να εντοπίσει την θέση του ως προς τον χάρτη που έχει κατασκευαστεί. Εν συνεχεία, στην βασική λειτουργία του Drone Controller, με την βοήθεια των ROS Service, ενεργοποιείται η αντίστοιχη σημαία και το Drone εκτελεί την αντίστοιχη ενέργεια. Στην συνέχεια της ενότητας θα αναλυθεί εκτενέστερα η υλοποίηση κάθε υποπακέτου που αναφέρθηκε προηγουμένως.



Σχήμα 15: Βασικό Διάγραμμα Ροής του Drone

4.4 Πακέτο του Turtlebot Controller

4.4.1 Localization

Όπως έχει ήδη αναφερθεί, στην χλάση αυτή υπάρχουν οι κατάλληλες συναρτήσεις για την αναζήτηση και την εύρεση του Ar Marker. Το Ar Marker έχει τοποθετηθεί στην πάνω επιφάνεια του Turtlebot με κατάλληλη φορά έτσι ώστε οι δύο πράκτορες να ευθυγραμμιστούν και να μπορούν να κινηθούν παράλληλα. Έτσι όταν εντοπιστεί το Ar Marker καθίσταται δυνατός ο προσδιορισμός της θέσης του Drone ως προς τον χάρτη. Στην συνέχεια, με την βοήθεια της *Join TF Tree Service* πραγματοποιείται η ένωση του Drone TF Frame Tree με το Frame του χάρτη που έχει κατασκευαστεί από το Turtlebot. Με τον τρόπο αυτό το Drone έχει πλέον την δυνατότητα να κινείται στον χάρτη. Η παραπάνω διαδικασία παρατίθεται και σε μορφή ψευδοκώδικα στον παρακάτω πίνακα.

Algorithm 3 Drone Localization Algorithm

```

1: procedure DRONE LOCALIZATION
2:   DroneLocalizationFlag  $\leftarrow$  False
3:   ArMarkerFlag  $\leftarrow$  False
4:   JointTreeFlag  $\leftarrow$  False
5:   SquareSearchSize  $\leftarrow$  0
6:   while ArMarkerFlag do
7:     SquareSearchSize  $\leftarrow$  2 Meters
8:     for i = 1, i++, while i < 3 do
9:       Move Drone Forward by SquareSearchSize
10:      Rotate Drone by 90° Clockwise
11:      if ArMarkerFlag then
12:        break
13:      end if
14:    end for
15:   end while
16:   Call JoinTreeService
17:   while JointTreeFlag do
18:     pass
19:   end while
20:   DroneLocalizationFlag  $\leftarrow$  True

```

Η γραμμική όσο και η περιστροφική κίνηση του Drone γίνεται μέσω δύο απλών συναρτήσεων οι οποίες μετράνε τον χρόνο έχει παρέλθει και υπολογίζουν σύμφωνα με την ταχύτητα που έχει δοθεί στο Drone την εκτιμώμενη θέση στην οποία βρίσκεται. Παρακάτω φαίνεται ένας ψευδοκώδικας αυτής της υλοποίησης για την γραμμική κίνηση κατά μια απόσταση. (Ακριβώς η ίδια λογική ακολουθείται και για την περιστροφή του Drone, μόνο που σε εκείνη την περίπτωση αντί για γραμμική ταχύτητα στέλνεται η γωνιακή ταχύτητα στο ROS Topic που είναι υπεύθυνο για την ταχύτητα που θα δοθεί στο Drone.)

Algorithm 4 Move Drone Forward by Function

```

1: function MOVE DRONE FORWARD(Distance)
2:   StartTime  $\leftarrow$  0
3:   DistanceTraveled  $\leftarrow$  0
4:   while DistanceTraveled  $<$  Distance do
5:     SendTwistMessageToDroneSpeedTopic
6:     DistanceTraveled  $\leftarrow$  LinearSpeed * (Time – StartTime)
7:   end while
8:   return True

```

Για να υπάρχει η δυνατότητα να γίνεται έλεγχος αν έχει εντοπιστεί το Ar Marker στον χώρο, δημιουργήθηκε ένας ROS Timer που καλεί μια απλή συνάρτηση μέσω της οποίας ελέγχει αν υπάρχει μετασχηματισμός από το TF Frame της κάτω κάμερας του Drone, στο TF Frame του Ar Marker. Αναλόγως ενεργοποιεί την σημαία του Ar Marker για να ενημερώσει την παραπάνω διαδικασία ώστε να γίνει η κλήση της *Join TF Tree Service*.

4.4.2 Drone Control

Η κλάση αυτή αποτελεί την βασική κλάση που αρχικοποιείται από το Drone κατά την έναρξη της λειτουργίας του ROS Node που ανατίθεται στον Drone Controller. Μέσω αυτής της κλάσης αρχικοποιούνται τα υπόλοιπα υποπακέτα τα οποία χρησιμοποιούνται στο πακέτο του Drone Controller. Επιπλέον, στη συγκεκριμένη κλάση υπάρχουν οι συναρτήσεις για την υλοποίηση των βασικών λειτουργιών του Drone καθώς επίσης και η δημιουργία ενός ROS Service που αντίστοιχα με την εντολή που λαμβάνει εκτελεί την αντίστοιχη λειτουργία. Οι βασικές αυτές λειτουργίες είναι τέσσερις και είναι οι εξής:

1. Take-Off
2. Land
3. Marker Follower
4. Fly to Height

Οι δύο πρώτες συναρτήσεις για την απογείωση και την προσγείωση του Drone εκτελούνται από δύο απλές συναρτήσεις που εκδίδουν ένα κενό μήνυμα στο αντίστοιχο ROS Topic του Drone. Η επίτευξη της υπόλοιπης διαδικασίας έχει υλοποιηθεί στο πακέτο του Ardrone Autonomy που αναφέρθηκε στο Κεφάλαιο 3.

Η διαδικασία παρακολούθησης του Ar Marker επιτεύχθηκε με την υλοποίηση ενός PID θέσης καθώς επίσης και από τις αντίστοιχες συναρτήσεις για την επανεύρεση του Ar Marker σε περίπτωση που αυτός έχει χαθεί. Ακόμα, για τον έλεγχο της ύπαρξης του Ar Marker χρησιμοποιήθηκε και ένας ROS Timer που ελέγχει τον χρόνο από την τελευταία φορά που εκδόθηκε μήνυμα στο αντίστοιχο ROS Topic του Ar Marker, όπως αυτό αναφέρθηκε στο πακέτο του Ar Tools στο Κεφάλαιο 3. Η διαδικασία που εκτελείται επισυνάπτεται σε μορφή ψευδοχώδικα στην συνέχεια.

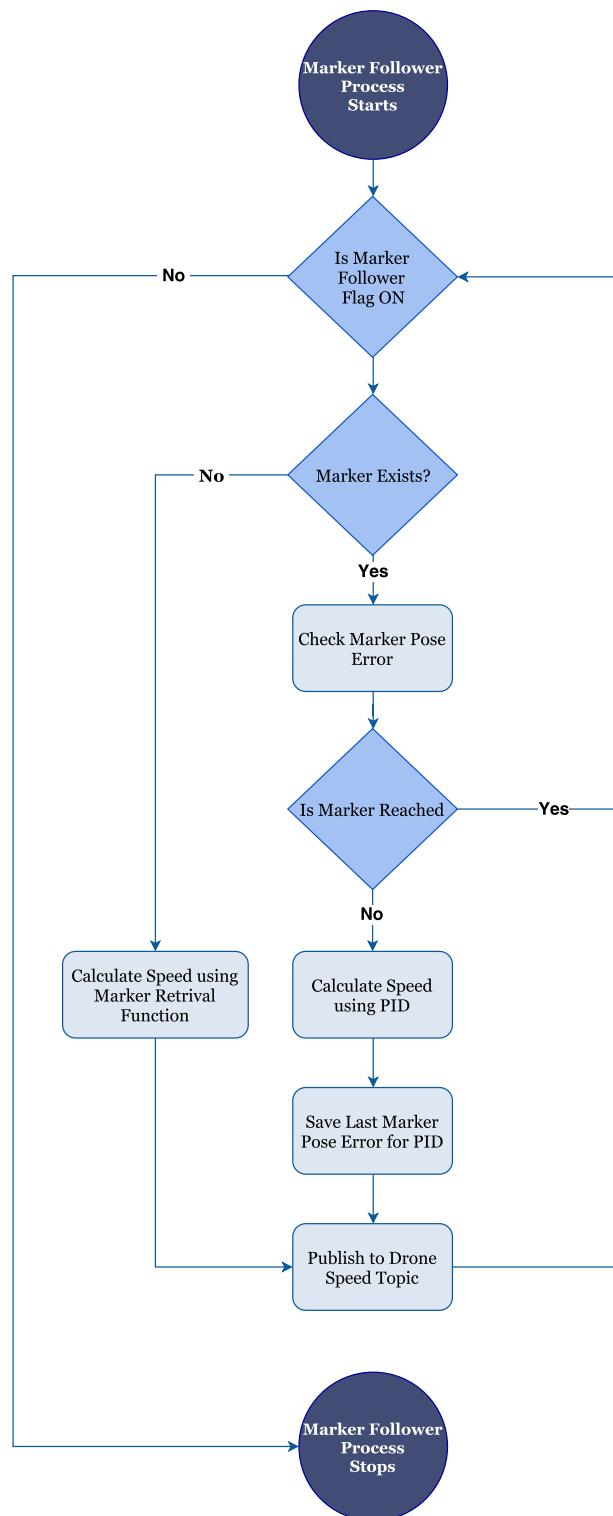
Algorithm 5 Drone Marker Follower Algorithm

```

1: procedure DRONE MARKER FOLLOWER
2:   FirstRunFlag  $\leftarrow$  False
3:   PoseError  $\leftarrow$  0
4:   LastPoseError  $\leftarrow$  0
5:   while MarkerFollowerFlag do
6:     if MarkerExist then
7:       if MarkerReached() then
8:         continue
9:       else
10:        PoseError  $\leftarrow$  GetPoseError()
11:        if FirstRunFlag then
12:          FirstRunFlag  $\leftarrow$  True
13:        else
14:          Speed  $\leftarrow$  PID(PoseError, LastPoseError)
15:          Publish Speed to Drone Speed Topic
16:        end if
17:        LastPoseError  $\leftarrow$  PoseError
18:      end if
19:    else:
20:      Speed  $\leftarrow$  MarkerRetrival()
21:      Publish Speed to Drone Speed Topic
22:    end if
23:  end while

```

Στην περίπτωση που χαθεί ο Ar Marker καλείται μια συνάρτηση επανεύρεσης του που εκτελείται έως ότου ξαναβρεθεί. Στη συνάρτηση αυτή το Drone αρχίζει να πετάει προς τα πάνω και να περιστρέφεται μέχρι ο Ar Marker να βρίσκεται στο οπικό πεδίο της κάμερας του. Η σχεδίαση της συνάρτησης βασίστηκε στο γεγονός ότι ο Ar Marker χάθηκε στιγμιαία όμως βρίσκεται σε κοντινή απόσταση από το Drone και δεν αποτελεί μια συνάρτηση αναζήτησης του Ar Marker στον χώρο, όπως αυτήν που υλοποιήθηκε στο υποπακέτο του Localization. Για την καλύτερη κατανόηση της συνολική διαδικασία, αυτή παρουσιάζεται και σε μορφή διαγράμματος στο Σχήμα 16.



Σχήμα 16: Διάγραμμα Ροής παρακολούθησης Ar Marker από το Drone

Όσον αφορά την τέταρτη λειτουργία που πραγματοποιείται, αυτή εκτελείται μέσω μιας συνάρτησης που αρχικά διαβάζει το ύψος του Drone από το ROS Topic, το οποίο περιέχει το ύψος στο οποίο πετάει το Drone, και την συνέχεια εκτελεί τον PID θέσης που αναφέρθηκε παραπάνω για να φτάσει στον επιθυμητό ύψος.

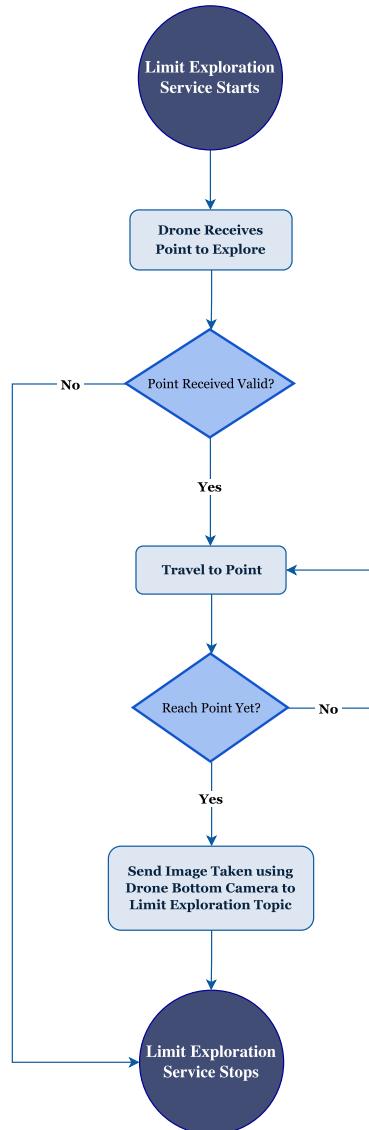
Για τον καλύτερο έλεγχο του Drone από άλλα ROS Nodes και στην περίπτωσή μας από το Turtlebot, δημιουργήθηκε το ROS Service Control Drone μέσω του οποίου μπορεί να δοθεί εντολή στο Drone για να εκτελέσει μια από τις 4 βασικές λειτουργίες που παρουσιάστηκαν. Χρησιμοποιώντας την υπηρεσία αυτή, ανάλογα με την εντολή που λαμβάνεται, ενεργοποιείται η αντίστοιχη σημαία και εκτελείται η κατάλληλη ενέργεια. Ακόμα υπάρχει μια εντολή που ελέγχει αν το Drone είναι σε θέση να ακολουθήσει το Ar Marker. Η υπηρεσία αυτή επιστρέφει μία σειρά χαρακτήρων ανάλογα με το αν η λειτουργία που ζητήθηκε ήταν δυνατόν να πραγματοποιηθεί ή όχι. Πιο συγκεκριμένα οι εντολές που λαμβάνονται φαίνονται στον Πίνακα 1.

Drone Control Service Input Command to Drone Procedure Matching			
Αριθμός	Λειτουργία του Drone	Success	Fail
1	Marker Follower (Start)	Done	Not Ready
2	Marker Follower (Stop)	Done	Not Ready
3	Fly Drone (Up)	Done	Not Ready
4	Fly Drone (Down)	Done	Not Ready
5	Take-Off	Done	Already Flying
6	Land	Done	Already Landed
7	Check for Ar Marker Existance	No Marker	Failed
Any Other Number	-	Invalid	Invalid

Πίνακας 1: Πίνακας Αντιστοίχησης Εντολής του Control Drone Service με Βασική Λειτουργία του Drone

4.4.3 Limit Exploration

Στην χλάση αυτή πραγματοποιείται η διαδικασία αποστολής του Drone στα όρια του χάρτη που έχει κατασκευαστεί μέχρι στιγμής ώστε να προστεθούν περισσότερες πληροφορίες για αυτά στο OGM. Με τον τρόπο αυτό το Turtlebot μπορεί να πλοηγηθεί σε σημεία πέρα από τον ήδη γνωστό χάρτη με απώτερο σκοπό να βελτιστοποιηθεί η διαδικασία εξερεύνησης με την βοήθεια του Drone. Έτσι, όπως είναι φανερό, χρειάζεται η δημιουργία του κατάλληλου ROS Service ώστε το Drone να μπορεί να ταξιδέψει στα σημεία αυτά καθώς και το κατάλληλο ROS Topic για την μεταφορά της εικόνας που λήφθηκε στον Controller του Turtlebot για περαιτέρω επεξεργασία. Επιπλέον στην χλάση αυτή περιλαμβάνονται και όλες οι συναρτήσεις που απαιτούνται για την διεκπεραίωση της διαδικασίας που αναφέρθηκε. Η διαδικασία αυτή γίνεται περισσότερο κατανοητή από το διάγραμμα ροής που φαίνεται στο Σχήμα 17.



Σχήμα 17: Διάγραμμα Ροής Εξερεύνησης Ορίων του χάρτη από το Drone

Για την μεταφορά του Drone στα οριακά σημεία που έχουν επιλεγεί για να εξερευνηθούν χρησιμοποιήθηκε ένας PID θέσης, όπως αυτός παρουσιάστηκε στο Κεφάλαιο 3. Ως είσοδοι στον PID λαμβάνονται η διαφορά της θέσης του Drone από το οριακό σημείο, καθώς και το προηγούμενο σφάλμα θέσης από αυτό.

Για να μπορεί να ενεργοποιηθεί η διαδικασία που αναλύθηκε παραπάνω χρησιμοποιήθηκε το Limit Exploration Service. Μέσω της υπηρεσίας αυτής αποστέλλονται τα σημεία που πρέπει να ταξιδέψει το Drone και ακολουθείται η διαδικασία που παρουσιάζεται στο Σχήμα 17. Εάν η διαδικασία πραγματοποιηθεί επιτυχώς, η τιμή που επιστρέφεται είναι αληθής, διαφορετικά η υπηρεσία επιστρέφει λανθάνουσα τιμή. Η διαδικασία εξερεύνησης των ορίων του χάρτη παρουσιάζεται και σε μορφή ψευδοχώδικα παραχάτω.

Algorithm 6 Drone Limit Exploration Algorithm

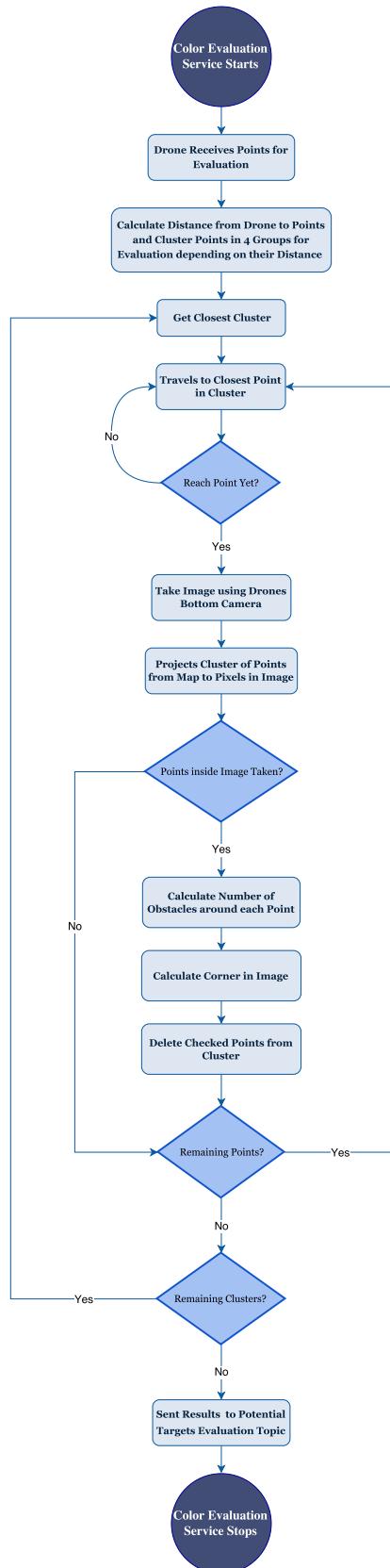
```

1: procedure DRONE LIMIT EXPLORATION(LimitPoint)
2:   if LimitPoint is Valid then
3:     PoseError  $\leftarrow$  0
4:     LastPoseError  $\leftarrow$  0
5:     FirstRunFlag  $\leftarrow$  False
6:     while GetPoseError()  $\leq$  0.25 do
7:       PoseError  $\leftarrow$  GetPoseError()
8:       if FirstRunFlag then
9:         FirstRunFlag  $\leftarrow$  True
10:      else:
11:        Speed  $\leftarrow$  PID(PoseError, LastPoseError)
12:        Publish Speed to Drone Speed Topic
13:      end if
14:      LastPoseError  $\leftarrow$  PoseError
15:    end while
16:    return True
17:  else:
18:    return False
19:  end if
```

4.4.4 Potential Targets Evaluation

Η συγκεκριμένη κλάση περιλαμβάνει τις απαραίτητες συναρτήσεις για την αξιολόγηση των απαιτούμενων στόχων καθώς επίσης και την υπηρεσία Potential Target Evaluation Service από την οποία λαμβάνονται οι συντεταγμένες των υποψήφιων στόχων που θα αξιολογηθούν από το Drone. Η διαδικασία αξιολόγησης των στόχων βασίζεται στο Σχήμα 18.

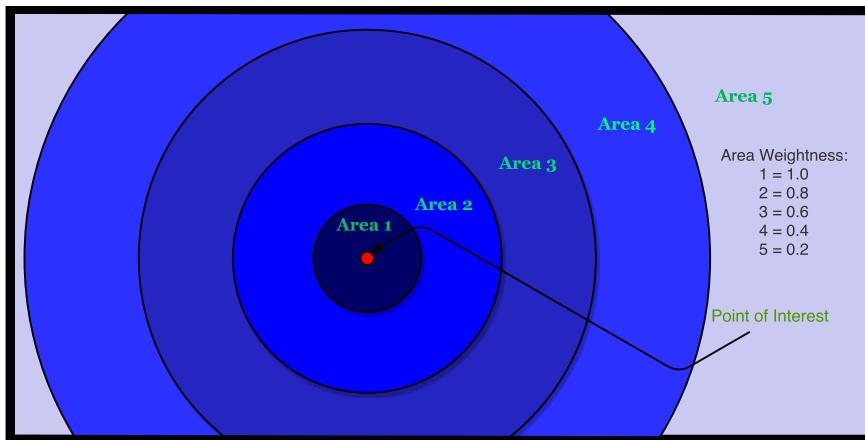
Όσον αφορά την ομαδοποίηση των συντεταγμένων των στόχων υπό αξιολόγηση έγινε ώστε να βελτιστοποιηθεί ο χρόνος που χρειάζεται για να αξιολογήσει το Drone όλους τους στόχους ταξιδεύοντας κατά την μικρότερη δυνατή απόσταση. Η ομαδοποίηση έγινε σε 4 ομάδες ανάλογα με την θέση των στόχων στον χάρτη.



Σχήμα 18: Διάγραμμα Ροής Αξιολόγησης Πινακών Στόχων από τoDrone

Για την αντιστοίχιση των σημείων του χώρου σε Pixels στον χάρτη χρησιμοποιήθηκε η βιβλιοθήκη του OpenCV και συγκεκριμένα η συνάρτηση ProjectPoints(). Η συνάρτηση όπως αναφέρθηκε στο Κεφάλαιο 3, δέχεται ως ορίσματα τις συντεταγμένες των σημείων ως προς το σύστημα συντεταγμένων της κάμερας και μετασχηματίζει τα σημεία αυτά σε συντεταγμένες σε μορφή (x, y) Pixel που αντιστοιχούν στην εικόνα. Εάν οι συντεταγμένες των Pixel που υπολογίστηκαν είναι εντός των ορίων των διαστάσεων της εικόνας που λήφθηκε από το Drone (640×320), τότε το σημείο ενδιαφέροντος βρίσκεται εντός της φωτογραφίας που τραβήχτηκε και επομένως είναι δυνατόν να αξιολογηθεί. Ο μετασχηματισμός των σημείων από το σύστημα συντεταγμένων του χάρτη στο σύστημα συντεταγμένων της κάμερας υπολογίστηκε λαμβάνοντας την σχετική θέση της κάμερας ως προς τον χάρτη την στιγμή της λήψης της φωτογραφίας από το ROS TF Frame Tree.

Η διαδικασία αξιολόγησης των στόχων μέσω της κάμερας του Drone γίνεται με την μέτρηση της πυκνότητας των εμποδίων γύρω από τον εκάστοτε στόχο. Επομένως, αφού έχει εντοπιστεί το σημείο ενδιαφέροντος στην συνέχεια εξερευνάται ο χώρος γύρω από αυτό ώστε να υπολογιστεί ένας συντελεστής της πυκνότητας εμποδίων.



Σχήμα 19: Γραφική Απεικόνιση Κατανομής Βαρών για την Αξιολόγησης Πιθανών Στόχων από το Drone

Καθώς η απόσταση του σημείου ενδιαφέροντος από τα εμπόδια έχει σημασία για την αξιολόγηση του σημείου, επιλέχθηκε διαφορετική βαρύτητα στην καταμέτρηση των κοντινότερων εμποδίων από οτι στα εμπόδια που βρίσκονται στα όρια της εικόνας. Τα βάρη κάθε περιοχής επιλέχθηκαν μεταξύ του διαστήματος $[0.2, 1]$, ξεκινώντας από την πιο απομακρυσμένη περιοχή έως το σημείο ενδιαφέροντος αντίστοιχα. Στην γραφική αναπαράσταση του Σχήματος 19 φαίνεται και σχηματικά ο διαχωρισμός των περιοχών αυτών.

Για την εύρεση γωνιών στην εικόνα χρησιμοποιήθηκε η συνάρτηση cornerHarris() της βιβλιοθήκης OpenCV, οπού υλοποιεί τον αλγόριθμο του Harris Corner Detection. Για να χρησιμοποιηθεί ο αλγόριθμος αυτός, η εικόνα του Drone μετασχηματίζεται σε ασπρόμαυρη κάνοντας χρήση μιας μάσκας. Η μάσκα αυτή μετασχηματίζει τα κόκκινα σημεία που θεωρούνται εμπόδια, όπως αυτό ειπώθηκε στο Κεφάλαιο 1, σε μαύρα και οποιοισδήποτε άλλο σημείο σε λευκό υποδηλώνοντας τον ελεύθερο χώρο. Επιπλέον, για να βρεθεί με ακρίβεια ο αριθμός των γωνιών που υπάρχουν στην εικόνα, αφού αυτό μπορεί να περιέρχει πολλές φορές απότομες εναλλαγές λόγω της μερικής ορατότητας κάποιου αντικείμενου με αποτέλεσμα ο αλγόριθμος επιστρέφει λανθασμένες τιμές, χρατήθηκαν μόνο οι τιμές των γωνιών που δεν βρίσκονται στα όρια της εικόνας. Τελός, καθώς πολλά από τα σημεία που επιστρέφει η συνάρτηση αντιστοιχούν στην ίδια γωνιά, έγινε

μια ομαδοποίηση των κοντινών σημείων έτσι ώστε να οριστεί εν τέλη ο ακριβής αριθμός γωνιών στην εικόνα. Η διαδικασία αξιολόγησης των πιθανών στόχων παρατίθεται και σε μορφή ψευδοκώδικα παρακάτω.

Algorithm 7 Drone Potential Targets Evaluation Algorithm

```

1: procedure DRONE POTENTIAL TARGETS EVALUATION
2:   ColorEvaluationArray  $\leftarrow \{\}$ 
3:   CornerEvaluationArray  $\leftarrow \{\}$ 
4:   EvaluationImage  $\leftarrow$  DroneBottomCameraTopic()
5:   Apply Mask to EvaluationImage
6:   for all PixelPoint in Cluster do
7:     Corners  $\leftarrow 0$ 
8:     OccupiedPixels  $\leftarrow 0$ 
9:     if PixelPoint  $\leq$  ImageLimits then
10:      for i = 1, i++, while i < ImageWidth do
11:        for j = 1, j++, while j < ImageHeight do
12:          if EvaluationImage[ i ][ j ] inside Area1 then
13:            Weight  $\leftarrow 1.0$ 
14:          else if EvaluationImage[ i ][ j ] inside Area2 then
15:            Weight  $\leftarrow 0.8$ 
16:          else if EvaluationImage[ i ][ j ] inside Area3 then
17:            Weight  $\leftarrow 0.6$ 
18:          else if EvaluationImage[ i ][ j ] inside Area4 then
19:            Weight  $\leftarrow 0.4$ 
20:          else
21:            Weight  $\leftarrow 0.2$ 
22:          end if
23:          OccupiedPixels  $\leftarrow$  OccupiedPixels + Weight
24:        end for
25:      end for
26:      CornersArray  $\leftarrow$  HarrisCornerDetection(EvaluationImage)
27:      for all Points in CornersArray do
28:        if Point near ImageLimits then
29:          Delete Point
30:        end if
31:      end for
32:      DifferenceArray  $\leftarrow \mathbf{D}(\text{Corners})$ 
33:      for all distance in DifferenceArray do
34:        if distance  $\geq$  Threshold then
35:          Corners++
36:        end if
37:      end for
38:    end if
39:    ColorEvaluationArray  $\leftarrow$  OccupiedPixels
40:    CornerEvaluationArray  $\leftarrow$  Corners

```

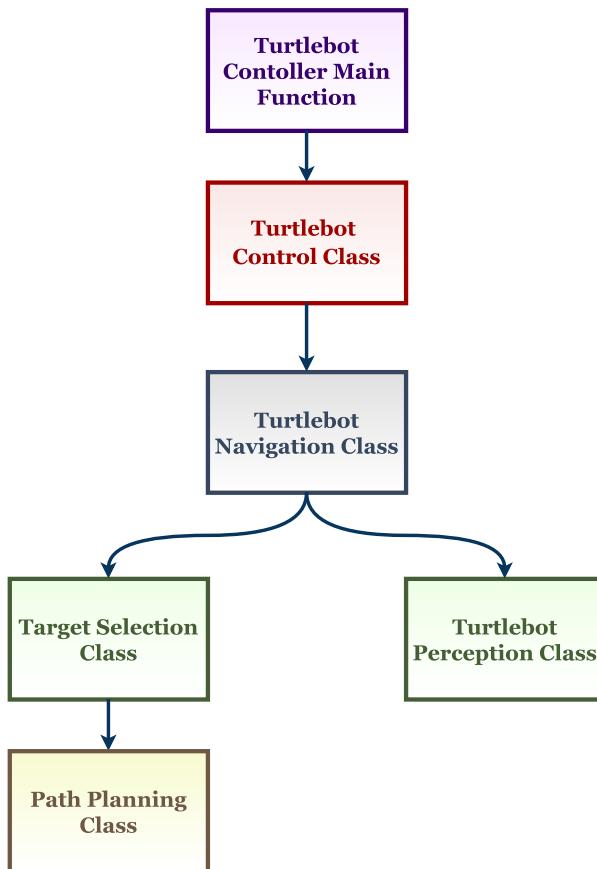
Όλη η διαδικασία της αξιολόγησης των στόχων γίνεται με την χλήση της υπηρεσίας Potential Target Evaluation Service. Όταν σταλθεί ένα αίτημα σε αυτήν την υπηρεσία, λαμβάνονται οι θέσεις των στόχων υπό αξιολόγηση ως προς το σύστημα συντεταγμένων του χάρτη. Αν ο τύπος των δεδομένων είναι σωστός, η σημαία εκκίνησης της διαδικασίας αξιολόγησης στόχων ενεργοποιείται (Color Evaluation Flag) και γίνεται επιτυχής έξοδος από την υπηρεσία. Εφόσον η σημαία έχει ενεργοποιηθεί, ξεκινάει η διαδικασία αξιολόγησης, όπως φαίνεται και στο διάγραμμα στο Σχήμα 18. Με τον τρόπο αυτό, όταν σταλεί μήνυμα για την αξιολόγηση των πιθανών στόχων στο Drone, η ροή του προγράμματος που κάλεσε την διαδικασία αξιολόγησης δεν σταματάει έως ότου ολοκληρωθεί η διαδικασία αξιολόγησης, αλλά περιμένει την λήψη των αποτελεσμάτων από το κατάλληλο ROS Topic, ενώ παράλληλα συνεχίζει την εκτέλεση του προγράμματος του.

Η αποστολή των αποτελεσμάτων της αξιολόγησης γίνεται μεσώ ενός ROS Topic, το οποίο αποστέλλει ένα ROS Message που περιεχεί τους δύο πίνακες με το Color και το Corner Evaluation των πιθανών στόχων, όπως αναφέρθηκε και στην υποενότητα 4.2.4 του κεφαλαίου.

4.5 Πακέτο του Turtlebot Controller

Το πακέτο του Turtlebot Controller αποτελείται από 5 υποπακέτα, καθένα από τα οποία αποτελεί μια ξεχωριστή χλάση, όπως αυτή ορίζεται στον δομημένο προγραμματισμό της Python. Θα πρέπει να σημειώθει ότι η βασική δομή του πακέτου αυτού, βασίστηκε στην διδακτορική εργασία του ερευνητή Εμμανουήλ Τσαρδούλια [16]. Περιληπτικά, τα υποπακέτα που περιέχονται στον Turtlebot Controller είναι τα εξής:

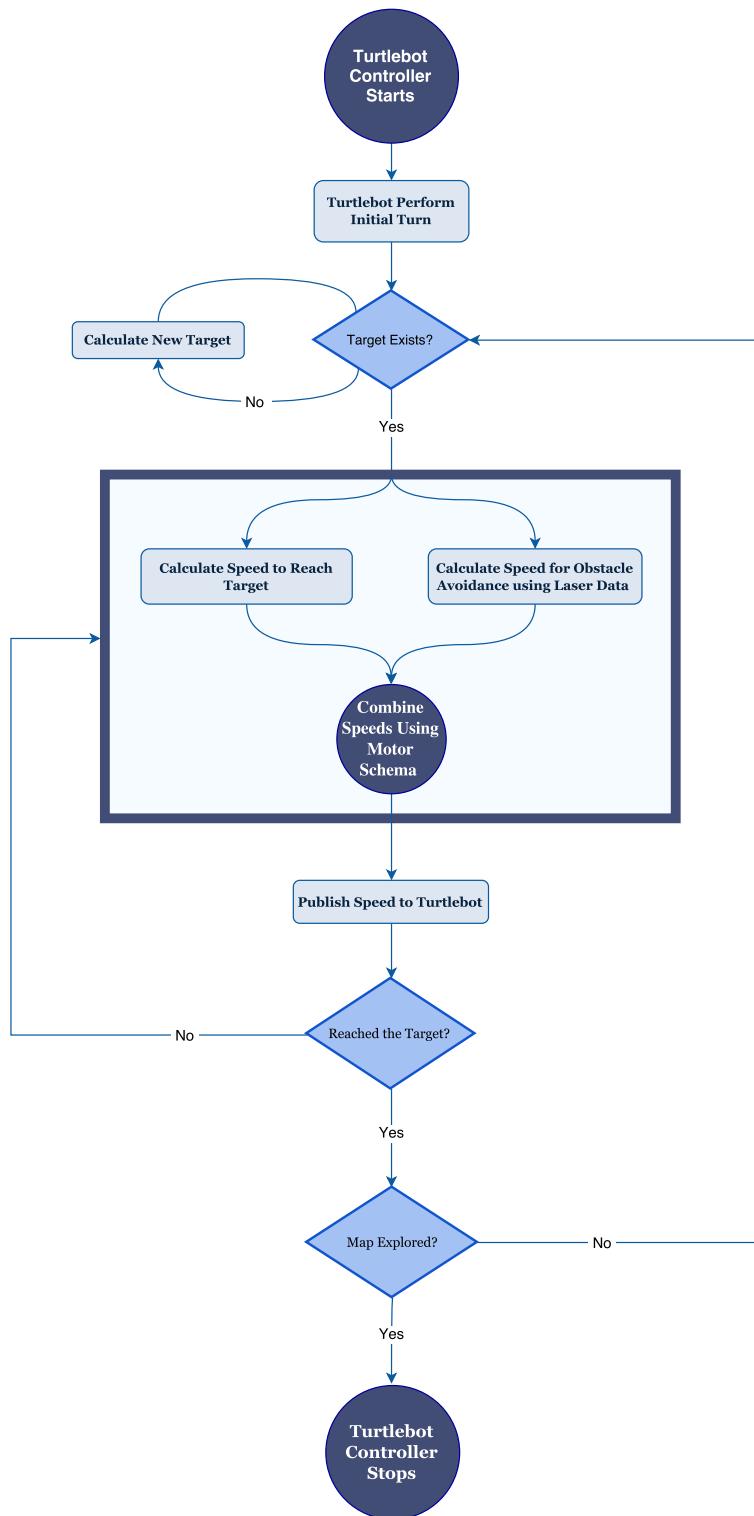
- **Turtlebot Control:** Πακέτο Ανάθεσης Ταχυτήτων στο Turtlebot.
- **Turtlebot Navigation:** Πακέτο υπεύθυνο για την πλοήγηση του Turtlebot στον χώρο.
- **Turtlebot Perception:** Πακέτο το οποίο διαβάζει τον χάρτη που κατασκευάζεται από το CRSM SLAM, την θέση του Turtlebot καθώς επίσης ανανεώνει και οπτικοποιεί τον χάρτη κάλυψης (Coverage Map).
- **Target Selection:** Πακέτο στο οποίο ανατίθεται η επιλογή του στόχου καθώς και η δημιουργία μονοπατιού προς αυτόν.
- **Path Planning:** Πακέτο που επικοινωνεί με την υπηρεσία OGM Path Planner για την επιστροφή του μονοπατιού μεταξύ δύο σημείων.



Σχήμα 20: Βασική Δομή Πωλέτων του Turtlebot Contoller

Με την έναρξη του Turtlebot Controller, το Turtlebot εκτελεί μια αρχική περιστροφή 360° ενώ παράλληλα διαβάζει και αρχικοποιεί τον χάρτη από το ROS Topic στο οποίο τον δημοσιεύει το CRSIM SLAM. Στην συνέχεια ακολουθεί μια επαναληπτική διαδικασία επιλογής στόχου και πλοιογήσης προς αυτόν μέσω των ταχυτήτων που ανατίθενται. Οι ταχύτητες αυτές είναι επηρεασμένες τόσο από το Lidar όσο και από την μέτρηση της απόστασης του Turtlebot από τους εκάστοτε στόχους. Η επαναληπτική αυτή διαδικασία ολοκληρώνεται με εξερεύνηση όλου του χώρου, όπου και ο Turtlebot Controller τερματίζει. Η παραπάνω διαδικασία παρουσιάζεται στο Σχήμα 21, όπου εμφανίζεται το βασικό διάγραμμα ροής του Turtlebot Controller.

Στην συνέχεια της ενότητας θα αναλυθεί εκτενέστερα η υλοποίηση κάθε υποπακέτου που τροποποιήθηκε, κάθισς επίσης θα γίνει και μια συνοπτική περιγράφη των υποπακέτων που χρησιμοποιήθηκαν αυτούσια για την καλύτερη κατανόηση του αναγνώστη.

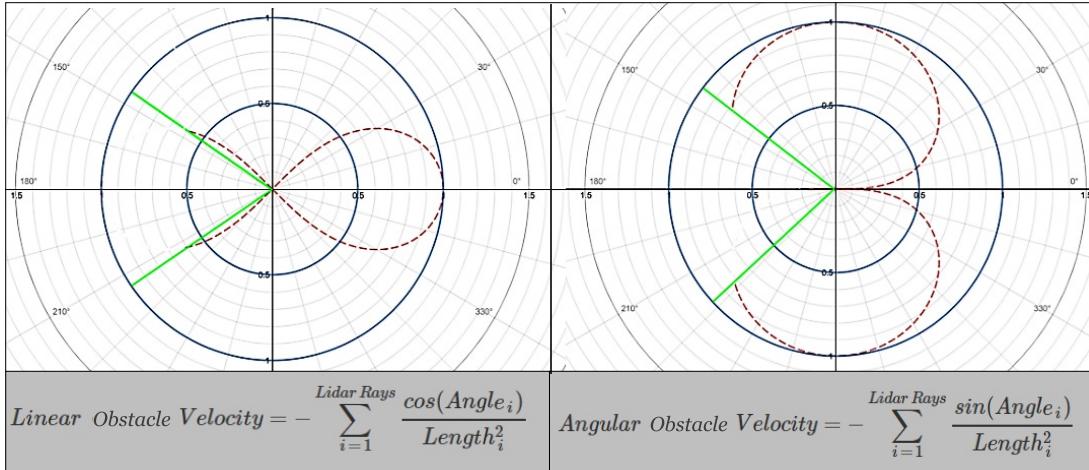


Σχήμα 21: Βασικό Διάγραμμα Ροής του Turtlebot

4.5.1 Turtlebot Control

Το πακέτο αυτό αποτελεί την κλάση που αρχικοποιείται από το ROS Node, το οποίο ανατίθεται στον Turtlebot Controller. Μέσω της κλάσης αυτής αρχικοποιούνται και όλες οι υπόλοιπες κλάσεις που περιέχονται στο πακέτο του Turtlebot Controller.

Επιπροσθέτως, στην κλάση αυτήν γίνεται υπολογισμός της ταχύτητας αποφυγής εμποδίων ώστε το Turtlebot ώστε να μπορεί να περιπλανηθεί στον χώρο αποφεύγοντας οποιαδήποτε εμπόδια. Αυτό επιτυγχάνεται με την βοήθεια ενός ROS Timer και χρησιμοποιώντας τα δεδομένα από το Lidar του Turtlebot. Καθώς κάποιες ακτίνες του Lidar έχουν μεγαλυτερη αξία από κάποιες άλλες, οι τιμές τους πολλαπλασιαστήκαν με μια συνάρτηση, η τιμή της οποίας μεταβάλλεται με την γωνία της εκάστοτε ακτίνας. Η αξιοποίηση των δεδομένων του Lidar για τον προσδιορισμό των ζητούμενων ταχυτήτων απεικονίζεται γραφικά στο Σχήμα 22, όπου με κόκκινη διακεκομένη γραμμή παρουσιάζεται η τιμη της συνάρτησης για το εύρως γωνιών του Lidar.



Σχήμα 22: Γραφική Αναπαράσταση Υπολογισμού Ταχυτήτων για την Αποφυγή Εμποδίων του Turtlebot

Θα πρέπει να σημειωθεί ότι η συναρτήση υπολογισμού των ταχυτήτων αποφυγής εμποδίων πάρθηκε από την αρχική δομή του Controller και αναφέρθηκε συνοπτικά για την καλύτερη κατανόηση της συνολικής λειτουργίας.

4.5.2 Turtlebot Navigation

Η κλάση αυτή αποτελεί ουσιαστικά την κυριότερη κλάση του Turtlebot Controller, αφού περιλαμβάνει τις βασικές συναρτήσεις για την πλοήγηση του Turtlebot στον χώρο καθώς επίσης καλεί και τις κατάλληλες συναρτήσεις του Target Selection για την επιλογή των στόχων και την κατασκευή μονοπατιού προς αυτόν.

Η διαδικασία επιλογής στόχου παρουσιάζεται στο διάγραμμα ροής του Σχήματος 25. Η διαδικασία ξεκινά με την λήψη του OGM καθώς και του χάρτη κάλυψης που βρίσκονται στην κλάση Turtlebot Perception. Στην συνέχεια, η διαδικασία επιλογής στόχου διαχωρίζεται ανάλογα με το αν επιλεγεί να σταλεί το Drone στα όρια του εξερευνημένου χάρτη. Στην περίπτωση αυτή εκτελείται η διεργασία ενίσχυσης του OGM που θα αναλυθεί στην επόμενη παράγραφο. Διαφορετικά καλείται η συνάρτηση επιλογής στόχου της κλάσης Target Selection, η οποία επιστρέφει το μονοπάτι προς τον στόχο και ύα παρουσιάστει

στην αντίστοιχη υποενότητα. Τέλος, το μονοπάτι που έχει κατασκευαστεί χωρίζεται σε υποστόχους και η διαδικασία ολοκληρώνεται ξεκινώντας έτσι την κίνηση του Turtlebot προς τον στόχο.

Η διεργασία ενίσχυσης του OGM αποστέλλοντας το Drone στα όρια του χάρτη ακολουθεί τα εξής βήματα:

- (A) Υπολογίζονται τα σημεία του OGM τα οποία θα εξερευνήσει το Drone.
- (B) Το Drone αποστέλλεται διαδοχικά στα σημεία που υπολογίστηκαν με χριτήριο την απόστασή του από αυτά.
- (C) Εφόσον το Drone φτάσει στο εκάστοτε σημείο και ληφθεί η εικόνα από την κάτω κάμερα του Drone εκτελείται η συγχώνευση των πληροφοριών της εικόνας με τον OGM.
- (D) Όταν το Drone εξερευνήσει όλα τα σημεία η διεργασία ολοκληρώνεται και ακολουθεί η κλήση της συνάρτησης επιλογής στόχου και κατασκευής μονοπατιού που αναφέρθηκε προηγουμένως.

Για την εύρεση των σημείων στα οποία θα σταλεί το Drone ακολουθείται ο παρακάτω αλγόριθμος.

Algorithm 8 Turtlebot OGM Limit Points Calculation Algorithm

```

1: procedure OGM Limit Points Calculation(OGM, MaxNumberofPoints)
2:   LimitPointArray  $\leftarrow \{\}$ 
3:   SearchStep  $\leftarrow \frac{2*(OGMWidth+OGMHeight)}{MaxNumberofPoints}$ 
4:
5:   Comment: Search in Y-Axis
6:   for i = 1, step = SearchStep, while i < OGMWidth do
7:     Find Minimum and Maximum Explored Point
8:     LimitPointArray  $\leftarrow$  Minimum
9:     if |Maximum – Minimum|  $\geq$  Threshold then
10:      LimitPointArray  $\leftarrow$  Maximum
11:    end if
12:   end for
13:
14:   Comment: Search in X-Axis
15:   for i = 1, step = SearchStep, while i < OGMHeight do
16:     Find Minimum and Maximum Explored Point
17:     LimitPointArray  $\leftarrow$  Minimum
18:     if |Maximum – Minimum|  $\geq$  Threshold then
19:       LimitPointArray  $\leftarrow$  Maximum
20:     end if
21:   end for

```

Ο OGM αποθηκεύεται ως ένας δισδιάστατος πίνακας, όπου κάθε κελί του πίνακα αποτελεί και ένα σημείο του χάρτη. Η αναζήτηση των μεγίστων και ελαχίστων εξερευνημένων σημείων που αναφέρεται στις γραμμές 8 και 17 γίνεται με αναζήτηση των αντίστοιχων στη-

λών και γραμμών του πίνακα για τα οριακά εξερευνημένα κελιά. Η αποστολή του Drone στα σημεία ενδιαφέροντος γίνεται με την χρήση της υπηρεσίας Limit Exploration Service Service, όπώς έχει αναλυθεί στην αντίστοιχη υπόενοτητα του Κεφαλαίου. Όσον αφορά το κομμάτι της συγχώνευσης της νέας πληροφορίας από την κάμερα του Drone στον OGM διερευνήθηκαν διάφορες στρατηγικές. Η λογική της στρατηγικής που ακολουθήθηκε βασίζεται στην προϋπόθεση ότι απαιτείται μικρή ακρίβεια για την πληροφορία που θα προστεθεί στον OGM αφού αυτός θα χρησιμοποιηθεί μόνο για την δημιουργία στόχων πέρα από τα οριακά σημεία του εξερευνημένου χάρτη, και δεν αποτελεί την τελική του μορφή, ο οποίος εξάγεται από το CRSM SLAM. Ακόμα, όπως έχει αναφερθεί στο Κεφάλαιο 1, κάθε κόκκινο Pixel θεωρείται εμπόδιο, ενώ τα υπόλοιπα Pixels ελεύθερα σημεία. Έτσι γνωρίζοντας την θέση και τον προσανατολισμό του Drone στον χάρτη κατά την διάρκεια λήψης της φωτογραφίας, ο αλγόριθμος συγχώνευσης των πληροφοριών της εικόνας του Drone στον OGM εμφανίζεται στον παρακάτω πίνακα.

Algorithm 9 Turtlebot Image Merging Algorithm

```

1: procedure DRONE IMAGE AND OGM DATA MERG-
   ING(OGM, Image, CenterofImageinOGMx, CenterofImageinOGMy, Droneθ)
2:   Resize Image to Match OGM Pixel Resolution
3:   for i = 1, i++, while i < ImageWidth do
4:     for j = 1, j++, while j < ImageHeight do
5:       x = CenterofImageinOGMx + i · cos (Droneθ) - j · sin (Droneθ)
6:       y = CenterofImageinOGMy + i · sin (Droneθ) + j · cos (Droneθ)
7:       if OGM[ x ][ y ] is Unknown then
8:         if Image[ i ][ j ] is Red then
9:           OGM[ x ][ y ] ← 100
10:        else
11:          OGM[ x ][ y ] ← 0
12:        end if
13:      end if
14:    end for
15:  end for

```

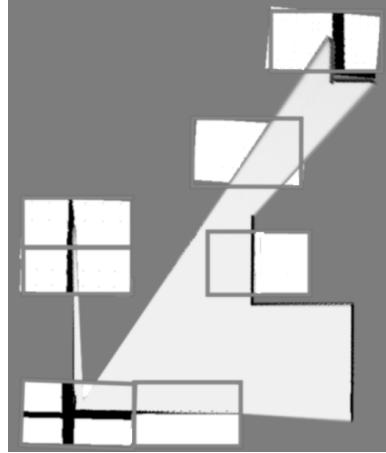
Όσον αφορά το μέγεθος χλιμάκωσης της εικόνας του Drone, εφόσον αυτό πετάει σε προκαθορισμένο ύψος, μπορεί να υπολογιστεί γνωρίζοντας την εστιακή απόσταση του φακού και την γωνία θέασης της κάμερας του Drone. Το μέγεθος χλιμάκωσης έγινε έτσι ώστε ένα Pixel της εικόνας να αντιστοιχεί σε ίδια απόσταση με ένα κελί του OGM (ρεσολουτιον). Ο τρόπος υπολογίσμου των νέων διαστάσεων της είκονας γνωρίζοντας το ύψος πτήσης του Drone και του FOV της κάμερας γίνεται μέσω της εξίσωσης:

$$Image[width, height] = \frac{\frac{2\pi}{360^\circ}}{Res} \cdot FOV[width, height]$$

όπου το Res πρόκειται για το resolution του OGM και το FOV για το οπτικό πεδίο της κάμερας

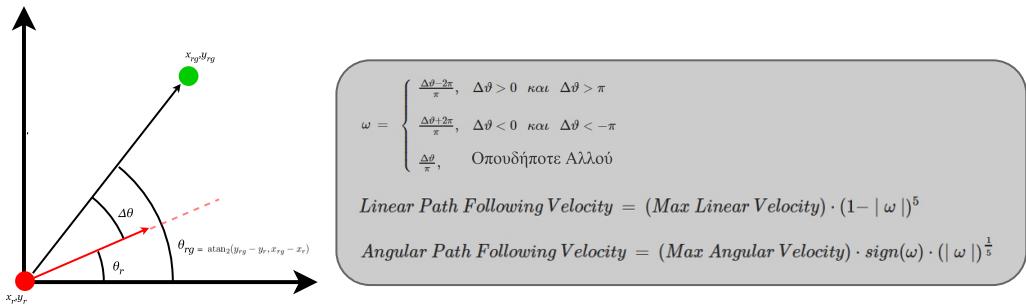
Ακόμα, η διαδικασία αντιστοίχισης των Pixel της εικόνας σε αυτά του OGM, που πραγματοποιείται στις γραμμές 5 και 6 του αλγορίθμου, αποτελείται από μια μετατόπιση και μια περιστροφή ενός δισδιάστατου σημείου. Η μετατόπιση καυνώς και η περιστροφή είναι γνωστά αφού γνωρίζουμε ότι το κέντρο της εικόνας είναι το σημείο στο οποίο στάλθηκε

το Drone και η γωνία στην οποία βρίσκεται το Drone, ως προς τον χάρτη, μπορεί να ληφθεί μεσώ του ROS TF Frame Tree.



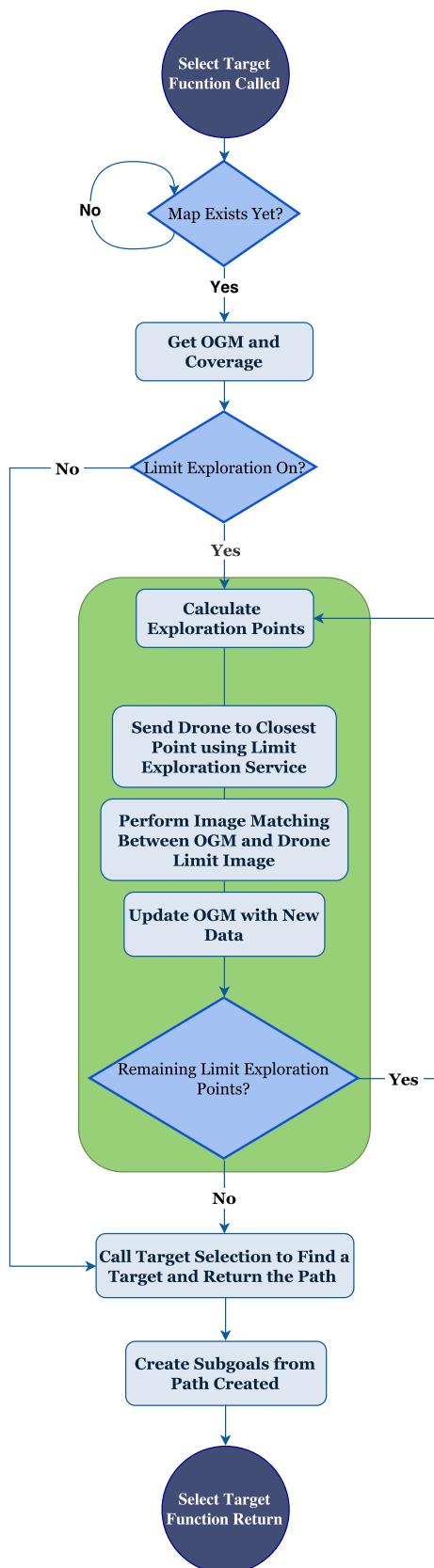
Σχήμα 23: Ενδεικτικό Στιγμιότυπο του αποτελέσματος ενίσχυσης του OGM από την λειτουργία της Εξερεύνησης των Ορίων του από το Drone

Πέρα από την παραπάνω διαδικασία πλοϊγησης του Turtlebot στον χώρο, στην κλάση αυτή υλοποιείται και η συνάρτηση δημιουργίας της κατάλληλης ταχύτητας για την προσέγγιση του εκάστοτε στόχου. Η συνάρτηση αυτή υπολογίζει το σφάλμα θέσης και προσανατολισμού του Turtlebot, αφού γνωρίζει την θέση και τον προσανατολισμό αυτού. Έχοντας το σφάλμα αυτό, οι ταχύτητες υπολογίζονται όπως φαίνονται στο Σχήμα 24.



Σχήμα 24: Γραφική Αναπαράσταση Υπολογισμού Ταχυτήτων για την προσέγγιση ενός στόχου από το Turtlebot

Τέλος, στην κλάση αυτήν περιλαμβάνεται και μια συνάρτηση που ελέγχει αν το Turtlebot έχει φτάσει στον υποστόχο που του έχει ανατεθεί. Έφοσον η απόσταση μεταξύ τους είναι μικρότερη από ένα όριο, το Turtlebot θεωρείται ότι έχει φτάσει επιτυχώς στον υποστόχο. Εάν ο υποστόχος που προσεγγίστηκε αποτελεί και τον τελευταίο του μονοπατιού που κατασκευάστηκε το Turtlebot ταξιδεύει στον τελικό στόχο. Θέτοντας την κατάλληλη σημαία, ώστε να ενημερωθεί ο Turtlebot Controller ότι θα πρέπει να ξεκινήσει και πάλι η διαδικασία επιλογής στόχου. Για τον τακτικό έλεγχο της επιτυχής προσέγγισης του στόχου που έχει ανατεθεί στο Turtlebot, η συνάρτηση αυτή καλείται με την χρήση ενός ROS Timer. Θα πρέπει να σημειωθεί ότι οι δύο τελευταίες συναρτήσεις δεν υλοποιήθηκαν αλλά πάρθηκαν αυτούσιες από την αρχική δομή του Controller και αναφέρονται για την καλύτερη κατανόηση της συνολικής λειτουργίας.

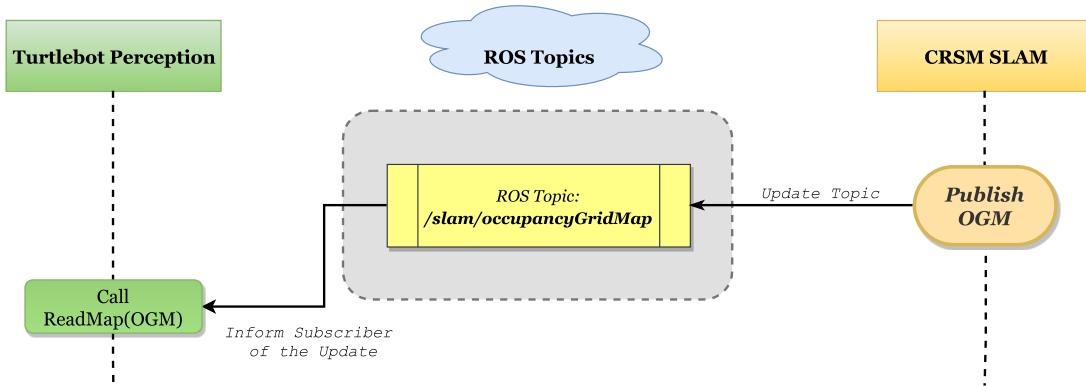


Σχήμα 25: Διάγραμμα Ροής που ακολουθείται για την επιλογή στόχου από το Turtlebot

4.5.3 Turtlebot Perception

Η κλάση Turtlebot Perception περιέχει τις κατάλληλες συναρτήσεις για να παρέχει στον Turtlebot Controller ανά πάσα στιγμή τον OGM, την θέση του Turtlebot καθώς και τον χάρτη κάλυψης.

Η επικοινωνία με το CRSM SLAM για την παροχή του OGM στον Turtlebot Controller παρουσιάζεται γραφικά στο Σχήμα26. Πέρα από την ενημέρωση του OGM, η συνάρτηση δημιουργεί και ανανεώνει επιπλέον τον χάρτη κάλυψης τον οποίο και στην συνέχεια αποστέλλει σε ένα ROS Topic για να χρησιμοποιηθεί από οποιδήποτε ROS Node τον απαιτήσει.



Σχήμα 26: Γραφική Αναπαράσταση Επικοινωνίας μεταξύ Turtlebot Perception και CRSM SLAM για την παροχή του χάρτη στον Turtlebot Controller

Η παροχή της πληροφορίας της θέσης του Turtlebot γίνεται μέσω ενός ROS Timer. Η συνάρτηση διαβάζει από το ROS TF Frame Tree την θέση και τον προσανατολισμό του Turtlebot ως προς το Frame του χάρτη, και εν συνεχεία αποθηκεύει την θέση του σε μορφή (x, y, θ) συντεταγμένων.

Το υποπακέτο αυτό πάρθηκε αυτούσιο από την αρχική δομή του Controller και παρουσιάστηκε για την καλύτερη κατανόηση της συνολικής λειτουργίας.

4.5.4 Target Selection

Στην κλάση του Target Selection γίνεται η επιλογή του στόχου στον οποίο θα κινηθεί το Turtlebot καθώς και η κατασκευή του μονοπατιού προς αυτόν. Η επιλογή αυτή βασίζεται σε μια Cost Based στρατηγική, όπου αυθορίζονται διαφορετικά κόστη για την εξαγωγή του καλύτερου δυνατού στόχου. Τα κόστη αυτά είναι τα εξής:

1. Κόστος Απόστασης από Εμπόδια
2. Κόστος Απόστασης από Turtlebot
3. Κόστος από την διαδικασία Color Evaluation
4. Κόστος από την διαδικασία Corner Evaluation

Το πρώτο κόστος βρίσκεται με την μέτρηση της ευκλείδειας απόστασης μεταξύ του στόχου και του κοντινότερου εμποδίου, ενώ το δεύτερο μεταξύ του στόχου και του Turtlebot. Τα δύο τελευταία κόστη αφορούν την αξιολόγηση του Drone και οι τιμές που παίρνουν έχουν αναλυθεί στην ενότητα του Drone Controller. Οι συναρτήσεις υπολογισμού για τα δύο πρώτα κόστη παρουσιάζονται παρακάτω.

$$w_{DistanceObstacle} = \frac{1}{8} \sum_{i=1}^8 D_i$$

όπου D_i η απόσταση του i-οστού σημείου του μονοπατιού από το πλησιέστερο εμπόδιο

$$w_{DistanceRobot} = \sum_{i=1}^{PathSize-1} D_{i,i+1}$$

όπου D_i η απόσταση του i-οστού σημείου του μονοπατιού από τη θέση του ρομπότ

Τα κόστη αυτά έχουν διαφορετική βαρύτητα, αναλόγως με τον σχοπό της εξερεύνησης, επομένως θα πρέπει να πολλαπλασιαστούν πριν ανθροιστούν με τα κατάλληλα βάρη. Η μεταβολή των βαρών για τα διάφορα κόστη γίνεται με την κλήση μίας συνάρτησης η οποία θέτει τα κόστη σε κατάλληλες τιμές σύμφωνα με τον στόχο της εξερεύνησης και τον χρόνο που έχει περάσει από την έναρξη της διαδικασίας. Επομένως η τελική συνάρτηση για την αξιολόγηση των στόχων είναι:

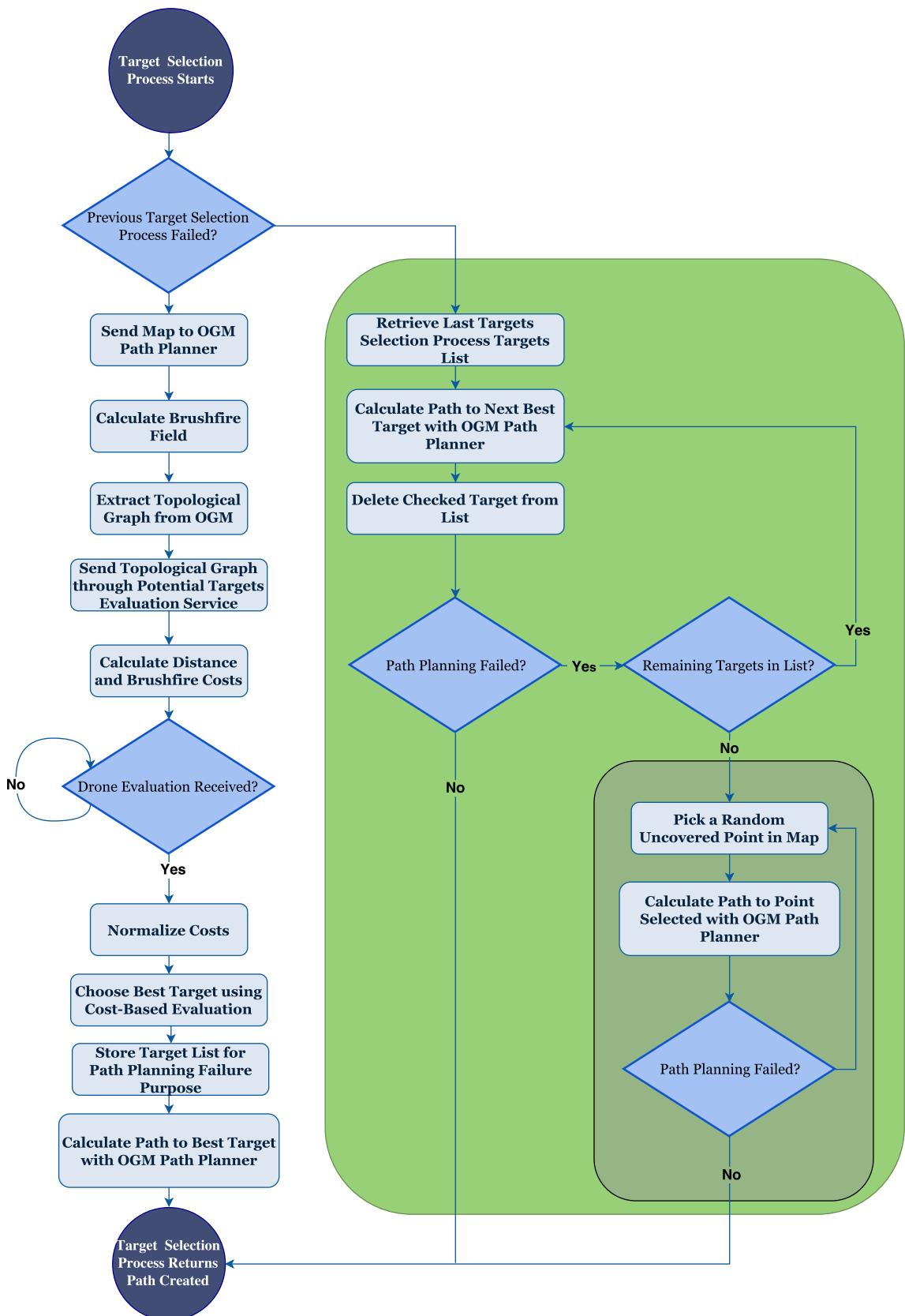
$$\begin{aligned} EvaluationOfTarget = & Coef_1 \cdot w_{DistanceObstacle} + Coef_2 \cdot w_{DistanceRobot} \\ & + Coef_3 \cdot w_{ColorEvaluation} + Coef_4 \cdot w_{CornerEvaluation} \end{aligned} \quad (4.1)$$

Περισσότερες πληροφορίες για τις τιμές αυτές θα αναφερθούν στο κεφάλαιο των περιμέτρων.

Η διαδικασία της επιλογής στόχου καθώς και της κατασκευής μονοπατιού φαίνονται σχηματικά στο διάγραμμα ροής του Σχήματος 27. Η διαδικασία ξεκινά με την αποστολή του OGM στον OGM Path Planner, με την κλήση της κατάλληλης συνάρτησης που βρίσκεται στην κλάση Path Planning, ώστε να μπορεί να κατασκευαστεί στην συνέχεια ένα μονοπάτι προς τον στόχο που θα επιλεγεί. Ακολουθεί η κατασκευή του πεδίου Brushfire και η εξαγωγή του τοπολογικού γράφου, ο οποίος αποτελείται από τους υποφήριους στόχους του χώρου. Οι αλγορίθμοι για τον υπολογισμό του Brushfire καθώς και του τοπολογικού γράφου πάρθηκαν από την αρχική δομή του πακέτου που χρησιμοποιήθηκε. Στην συνέχεια χρησιμοποιώντας την υπηρεσία Potential Targets Evaluation αποστέλλονται οι πιθανοί στόχοι του τοπολογικού γράφου στον Drone Controller. Αφού γίνει ο υπολογισμός του κόστους απόστασης από το Turtlebot και του κόστους απόστασης από τα εμπόδια (τοπολογικού κόστους), χρησιμοποιώντας το πεδίο Brushfire που υπολογίστηκε προηγουμένως, καθώς και την θέση του Turtlebot την δεδομένη χρονική στιγμή, γίνεται εκκίνηση μίας διαδικασίας αναμονής μέχρι να πραγματοποιηθεί η λήψη της αξιολόγησης των στόχων από τις διαδικασίες που περιγράφηκαν στην αντίστοιχη υποενότητα του Drone Controller. Για την ορθή διεξαγωγή συμπερασμάτων κατά την αξιολόγηση των στόχων, τα 4 κόστη που έχουν βρεθεί κανονικοποιούνται στο διάστημα [0, 255] και γίνεται η επιλογή του καλύτερου στόχου με τον τρόπο που αναφέρθηκε στη αρχή της υποενότητας. Τέλος, καλώντας την υπηρεσία κατασκευής του μονοπατιού που βρίσκεται στην κλάση του Path Planning γίνεται η δημιουργία του μονοπατιού που στην συνέχεια επιστρέφεται από την διαδικασία.

Στην περίπτωση όπου η κατασκευή του μονοπατιού ως προς τον στόχο είναι ανεπιτυχής, ξεκινάει μια διαφορετική διαδικασία επιλογής στόχου. Η διαδικασία αυτή ξεκινά με την επανάληψη της αρχικής διεργασίας, μόνο που αυτήν την φορά έχει ενεργοποιηθεί η σημαία εναλλακτικής επιλογής στόχου. Κατά το αρχικό Path Planning έχει πραγματοποιηθεί η αποθήκευση της λίστας των υποφήριων στόχων με την κατάλληλη σειρά προτεραιότητας. Έτσι στην εναλλακτική διαδικασία επιλογής στόχου γίνεται έλεγχος όλων των υποφήριων στόχων για την δυνατότητα κατασκευής μονοπατιού προς αυτούς διαδοχικά. Εάν βρεθεί μονοπάτι προς κάποιον από αυτούς, η επαναληπτική διαδικασία τερματίζει και

η συνολική διεργασία επιλογής στόχου επιστρέφει το μονοπάτι που κατασκευάστηκε. Σε διαφορετική περίπτωση, αρχίζει μια διαδικασία τυχαίας επιλογής ενός σημείου που δεν έχει καλυφθεί από το Turtlebot και ελέγχεται η δυνατότητα κατασκευής μονοπατιού. Μόλις βρεθεί κάποιο σημείο που πραγματοποιεί τα παραπάνω χριτήρια, η διεργασία τερματίζει επιστρέφοντας και πάλι το μονοπάτι.

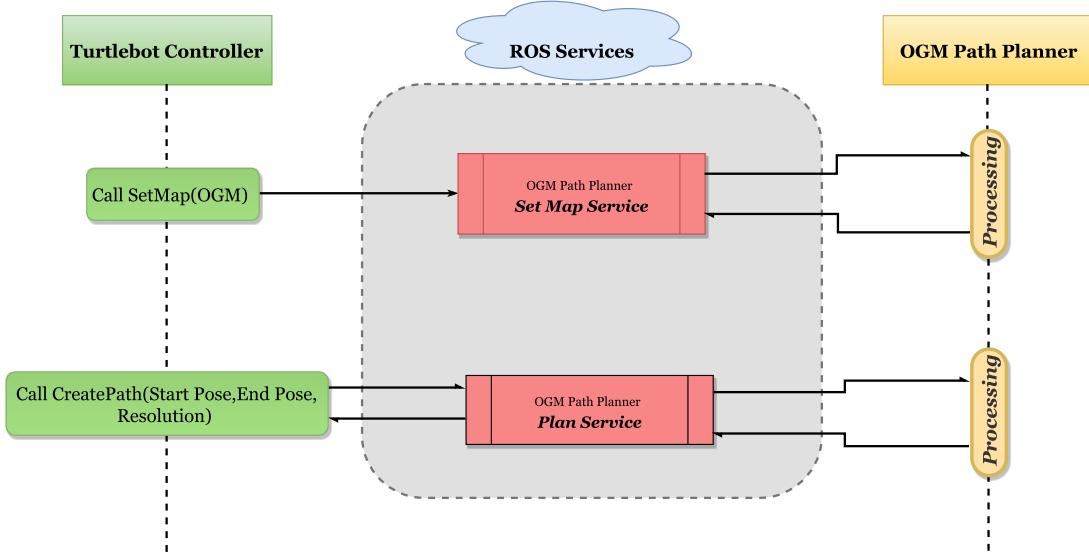


Σχήμα 27: Διάγραμμα Ροής της διαδικασίας επιλογής στόχου του Turtlebot

4.5.5 Path Planning

Όπως έχει αναφερθεί και στο Κεφάλαιο 3, για την κατασκευή του μονοπατιού χρησιμοποιήθηκε ο OGM Path Planner. Η επικοινωνία του Turtlebot Controller με τον OGM Path Planner πραγματοποιείται μέσω της συγκεκριμένη κλάση. Η κατασκευή του μονοπατιού πραγματοποιείται χρησιμοποιώντας δύο υπηρεσίες (ROS Services) που παρέχονται από τον OGM Path Planner.

Η πρώτη υπηρεσία αφορά την αποστολή του χάρτη στον οποίο θέλουμε να δημιουργήσουμε το μονοπάτι στον OGM Path Planner, ενώ η δεύτερη υπηρεσία την δημιουργία μονοπατιού μεταξύ των δύο σημείων που ζητήθηκαν στον χάρτη που διαθέτει ο OGM Path Planner την δεδομένη χρονική στιγμή. Στην κλάση αυτήν υπάρχουν επομένως δύο συαρτήσεις για την ανταλλαγή των δεδομένων μεταξύ των δύο ROS Nodes. Ο τρόπος επικοινωνίας παρουσιάζεται γραφικά στο Σχήμα 28.



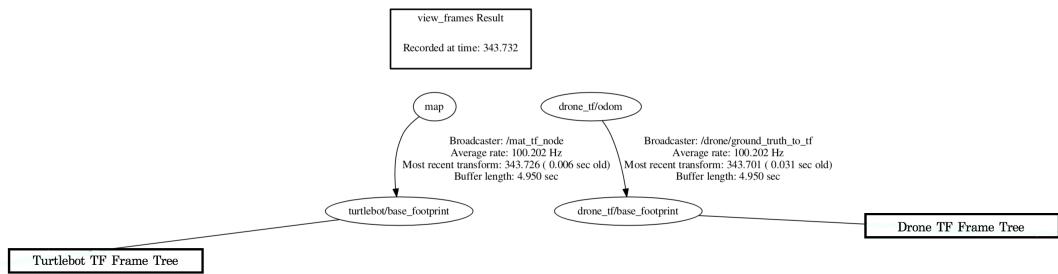
Σχήμα 28: Γραφική Αναπαράσταση Επικοινωνίας μεταξύ Turtlebot Controller και OGM Path Planner για την δημιουργία Μονοπατιού προς τον Στόχο

Το υποπακέτο αυτό πάρθηκε αυτούσιο από την αρχική δομή του Controller και παρουσιάστηκε για την καλύτερη κατανόηση της συνολικής λειτουργίας.

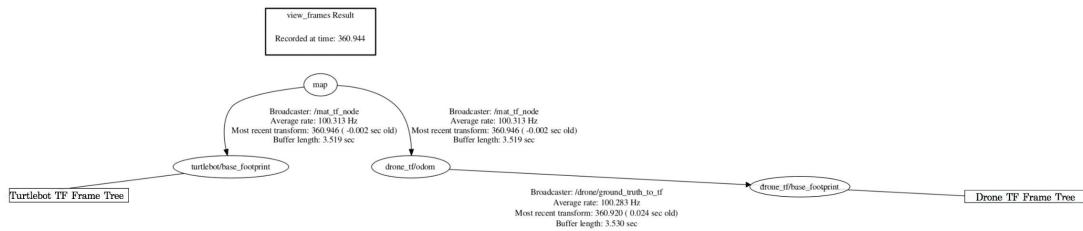
4.6 TF Frame Tree Controller

Στην έναρξη της προσομοίωσης, όπως έχουμε ήδη αναφέρει οι δύο πράκτορες έχουν ξεχωριστά συστήματα συντεταγμένων. Η ένωση αυτών γίνεται με την κλήση της υπηρεσίας Join Tree Service, όπως έχει αναφερθεί στην αρχή του κεφαλαίου αυτού. Ο TF Frame Tree Controller αποτελεί ουσιαστικά ένα ROS Node που κάνει διαθέσιμη την παραπάνω λειτουργία στα υπόλοιπα ROS Nodes. Ο λόγος της δημιουργίας της υπηρεσίας αυτής σε ανεξάρτητο ROS Node είναι για τον καλύτερο έλεγχο της σύνδεσης των δύο TF Tree.

Ο TF Frame Tree Controller αποτελεί ένα script σε python που ενεργοποιεί την κατάλληλη συνάρτηση για την ένωση του Drone TF Tree με το Frame του χάρτη μόλις γίνει η κλήση της υπηρεσίας Join Tree Service.



Σχήμα 29: Γραφική Απεικόνιση των δύο TF Frame Tree πρίν την κλήση της υπηρεσίας Join Tree Service από τον Drone Controller



Σχήμα 30: Γραφική Απεικόνιση του ενωμένου TF Frame Tree μετά την κλήση της Join Tree Service από τον Drone Controller

5

Πειράματα

Στο παρόν κεφάλαιο πραγματοποιείται η εκτενής περιγραφή όλων των προσομοιώσεων που εκτελέσθηκαν για τις ανάγκες της παρούσας διπλωματικής εργασίας. Στο σημείο αυτό, πρέπει να σημειωθεί ότι τα πειράματα έγιναν σε ένα Laptop, το οποίο διαθέτει επεξεργαστή Intel Core i7, τεσσάρων πυρήνων, συχνότητα λειτουργίας 2.4 GHz, μνήμη RAM 8 GB και λειτουργικό σύστημα Linux, Ubuntu 14.04 Thustly Tahr. Για την αποτελεσματικότερη κατανόηση των πειραμάτων παραθέτονται παρακάτω διάφορα στιγμιότυπα από την εκτέλεση αυτών, καθώς και τα διαγράμματα που προέκυψαν αφού έγινε η επεξεργασία των δεδομένων με τη βοήθεια του υπολογιστικού εργαλείου Matlab. Τα διαγράμματα που θα παρουσιαστούν θα απεικονίζουν το ποσοστό του εξερευνημένου χώρου καθώς και το ποσοστό κάλυψης του χώρου με την πάροδο του χρόνου. Η μέτρηση των δύο αυτών ποσοστών αποτελεί ένα βασικό τρόπο σύγκρισης αν η διαδικασία εξερεύνησης βελτιώθηκε με την βοήθεια του εναέριου πράκτορα με σκοπό την ασφαλή διεξαγωγή κάποιων συμπερασμάτων. Ακόμα, για λόγους που θα εξηγηθούν στην συνέχεια, τα ίδια ποσοστά μετρήθηκαν συναρτήσει του αριθμού των στόχων που επισκέφθηκε το Turtlebot κατά τη εξερεύνηση. Με τον τρόπο αυτό θα εξαχθούν και κάποια επιπλέον συμπεράσματα σχετικά με την επίτευξη, ή όχι, βελτίωσης στην επιλογή των στόχων που επιλέγονται τελικά από το Turtlebot. Τέλος, πρέπει να σημειωθεί ότι σε όλα τα πειράματα η γραμμική και η γωνιακή ταχύτητα των ρομποτικών πρακτόρων είναι 0.3 m/s, 0.3 rad/s αντίστοιχα.

Συνολικά εκτελέσθηκαν 54 διαφορετικά πειράματα.

- Όπως αναφέρθηκε και στα παραπάνω κεφάλαια σχεδιάστηκαν **3** διαφορετικοί χώροι για να υλοποιηθούν τα πειράματα. Ο πρώτος χώρος αποτελείται κυρίως από διαφόρου μήκους διαδρόμους με διάφορα πλάτη ο καθένας. Στον δεύτερο και στον τρίτο χώρο τα εμπόδια που είναι τοποθετημένα ποικίλουν τόσο σε μέγεθος όσο και σε σχήμα. Η βασική διαφορά των δύο χώρων είναι ότι στον δεύτερο χώρο τα εμπόδια είναι πυκνώς τοποθετημένα δημιουργώντας μικρότερους υποχώρους προς εξερεύνηση-κάλυψη, σε αντίθεση με τον τρίτο, όπου οι επιμέρους υποχώροι είναι ευρύτεροι. Το ύψος των εμποδίων και των τοίχων είναι 1.5 m, καθώς επιλέχθηκε για την υλοποίηση οι χώροι να πληρούν τις προϋποθέσεις του RoboCup¹⁸.
- Τα δύο ρομπότ συνεργάστηκαν για **3** διαφορετικούς στόχους οι οποίοι είναι η πλήρης εξερεύνηση, η πλήρης κάλυψη και η πλήρης κάλυψη σε περιορισμένο χρόνο ενός χώρου. Πλήρη Εξερεύνηση σε ένα χώρο πραγματοποιείται όταν όλα τα σημεία του χώρου έχουν εντοπιστεί από το LIDAR του Turtlebot, ενώ πλήρη κάλυψη θεωρείται όταν το Turtlebot έχει βρεθεί σε ακτίνα 1.25 m από όλα τα σημεία του χώρου αυτού. Τέλος, ο τρίτος σκοπός της πλήρους κάλυψης σε περιορισμένο χρόνο αποτελείται από δύο φάσεις. Στην πρώτη φάση, η οποία διαρκεί 7 λεπτά, εκτελείται μία λογική δυναμικής προσέγγισης της πλήρους κάλυψης, όπου οι συντελεστές της συνάρτησης επιλογής στόχου μεταβάλλονται με το χρόνο, ενώ στη δεύτερη φάση εκτελείται η μέθοδος της πλήρους εξερεύνησης όπως παρουσιάζεται παραπάνω.
- Οι στρατηγικές συνεργασίας που υλοποιήθηκαν είναι **3**. Αρχικά το Turtlebot δρούσε μόνο του και η επιλογή του επόμενου στόχου γινόταν με βάση τις πληροφορίες από τους αισθητήρες του. Στη δεύτερη στρατηγική, οι δύο πράκτορες συνεργάστηκαν, με το drone να εξερευνεί τα όρια του γνωστού χάρτη και να προτείνει στο

¹⁸<http://www.roboocup.org/>

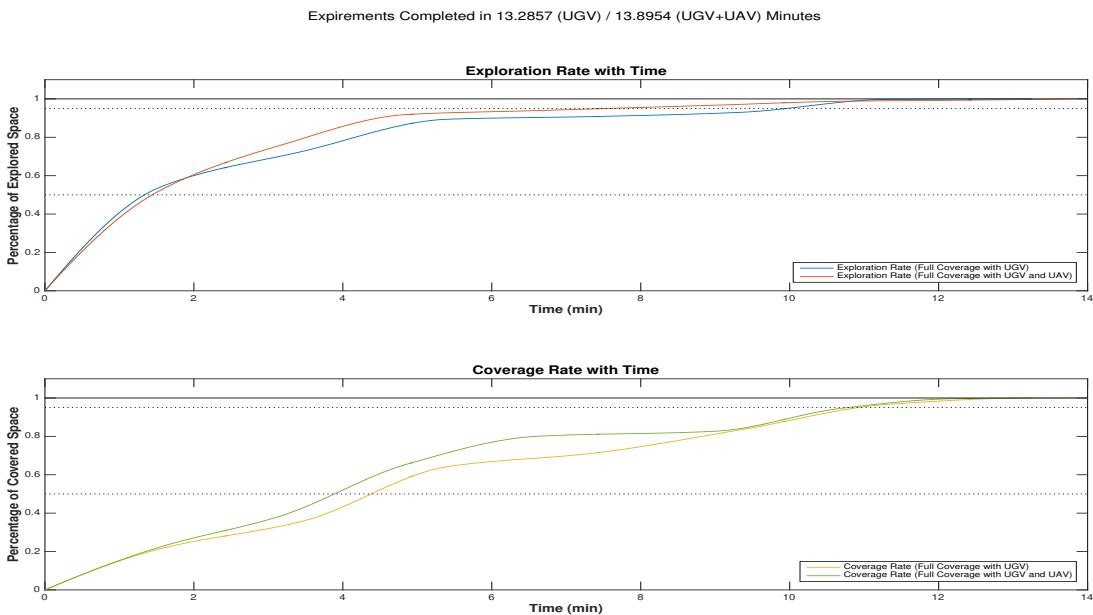
Turtlebot στόχους εκεί. Τέλος, οι δύο πράκτορες συνεργάστηκαν, με το drone να εκτελεί εξερεύνηση στα όρια του OGM και να αξιολογεί τους υποψήφιους στόχους χρησιμοποιώντας την κάμερα του και υπολογίζοντας την πυκνότητα των εμποδίων γύρω από τον εκάστοτε στόχο επηρεάζοντας την απόφαση του Turtlebot.

- Τέλος, πρέπει να σημειωθεί ότι κάθε σενάριο εκτελέστηκε από 2 φορές για την καλύτερη ακρίβεια των αποτελεσμάτων.

5.1 Δημιουργία πειραματικών χώρων στην πλατφόρμα προσομοιώσεων GAZEBO

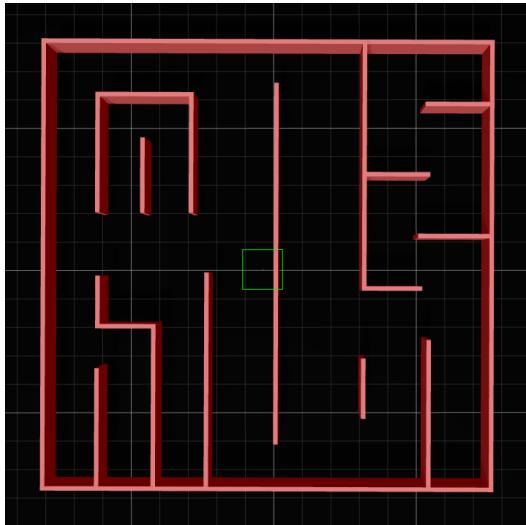
Για τις προσομειώσεις που υλοποιήθηκαν, σχεδιάστηκαν διαφορετικοί χώροι για κάθε πείραμα. Επειδή υπήρχε η ανάγκη οι χώροι να περιέχουν χρώματα, ώστε ο εναέριος πράκτορας να πραγματοποιεί την διαδικασία αξιολόγησης των στόχων όπως αυτό ειπώθηκε στο Κεφάλαιο 1, οι χώροι δημιουργήθηκαν στο πρόγραμμα Gazebo 7 κάνοντας χρήση του εργαλείου Model Editor και εν συνεχεία εισήχθησαν στο Gazebo 2 ως έτοιμα μοντέλα. Για την υλοποίηση επιλέχθηκε το χρώμα κόκκινο για τους τοίχους και τα εμπόδια.

Μία πρώτη προσπάθεια διεξαγωγής πειραμάτων έγινε σε χώρο διαστάσεων $100 m^2$. Κατά την διεξαγωγή των πειραμάτων παρατηρήθηκε ότι ο χώρος ήταν αρκετά μικρός, ώστε η εξερεύνηση χρησιμοποιώντας μόνο το Turtlebot να είναι το ίδιο αποδοτική με την εξερεύνηση με την συνεργασία του με το Drone. Αυτό συμβαίνει αφού η επιλογή του μη βέλτιστου δυνατού στόχου δεν επιβραδύνει σε μεγάλο βαθμό τη διάρκεια της εξερεύνησης. Το Turtlebot έχει να κινηθεί σε μικρό σχετικά χώρο και επομένως η επιλογή ενός λάθος μονοπατιού που πιθανώς να οδηγήσει σε κάποιο αδιέξοδο δε θα επηρεάσει σε μεγάλο βαθμό τη ροή της εξερεύνησης. Στην συνέχεια παρατίθενται διαγράμματα από την διεξαγωγή ενός από τα πειράματα που πραγματοποιήθηκαν σε χώρο διαστάσεων $100m^2$ τόσο με το Turtlebot μόνο του, όσο και με την συνεργασία του Drone. Όπως φαίνεται στο Σχήμα 31, η διάρκεια της πλήρους κάλυψης όλου του χώρου πραγματοποιείται και στις δύο περιπτώσεις σε περίπου 13 λεπτά.

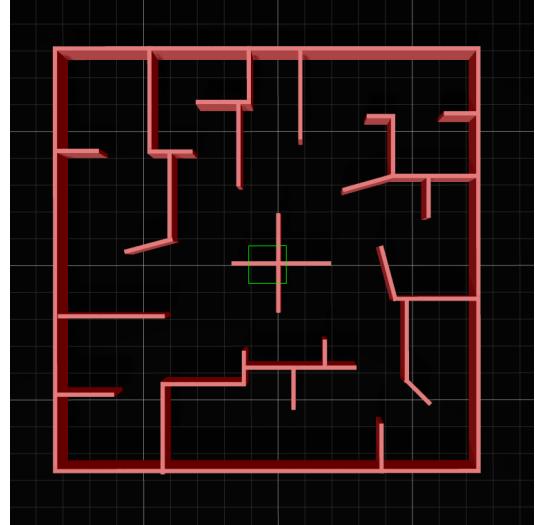


Σχήμα 31: Διαγράμματα Ποσοστού Εξερεύνησης και Κάλυψης με τον Χρόνο του Χώρου $100m^2$ από το Turtlebot

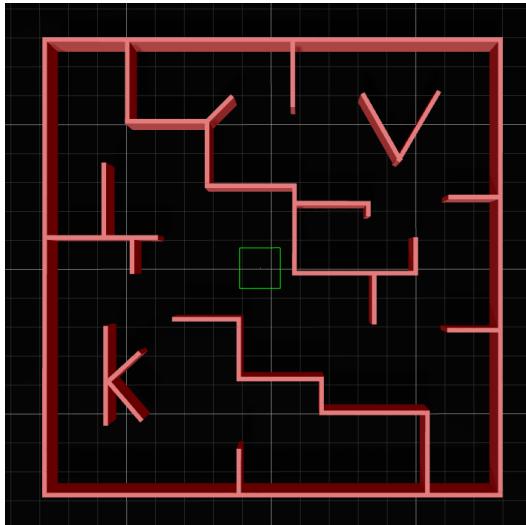
Για τον παραπάνω λόγο επιλέχθηκε να γίνει η κατασκευή μεγαλύτερων χώρων διαστάσεων περίπου $225 m^2$. Οι κατόψεις των χώρων που χρησιμοποιήθηκαν τελικώς, και θώς και του μικρότερου χώρου των $100 m^2$, φαίνονται παρακάτω. Όπως μπορεί να παρατηρηθεί είναι τρεις χώροι διαφορετικού τύπου.



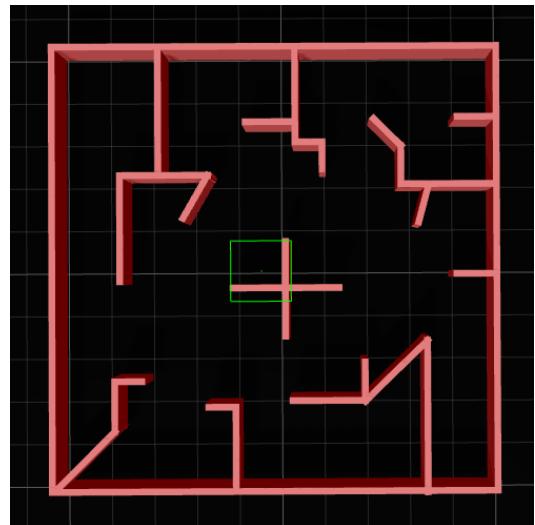
(α') Κάτοψη Πρώτου Πειραματικού Χώρου



(β') Κάτοψη Δεύτερου Πειραματικού Χώρου



(γ') Κάτοψη Τρίτου Πειραματικού Χώρου

(δ') Κάτοψη Μικρού Πειραματικού Χώρου
 $100 m^2$

Σχήμα 32: Κατόψεις των χώρων που σχεδιάστηκαν για την υλοποίηση των προσομοιώσεων

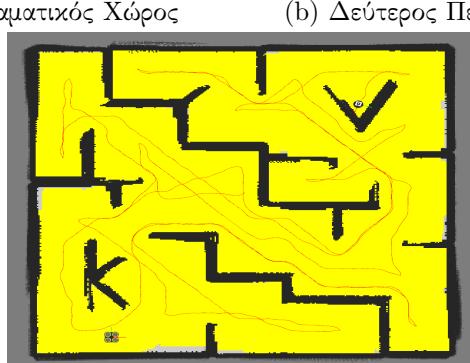
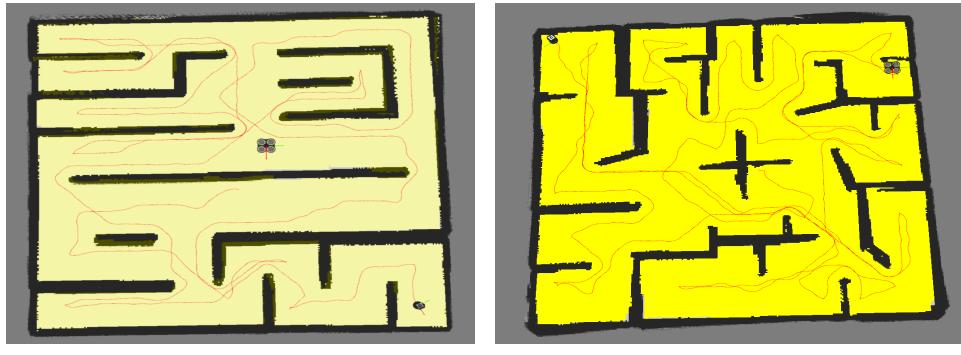
5.2 Διεξαγωγή Πειραμάτων

Για την μέτρηση της αποδοτικότητας της συνεργασίας που υλοποιήθηκε μεταξύ των δύο πρακτόρων διεξάχθηκαν πειράματα στους 3 χώρους που παρουσιάστηκαν στην προηγούμενη ενότητα. Για την ορθότερη διεξαγωγή συμπερασμάτων κάθε πείραμα εκτελέστηκε από δύο φορές και στην συνέχεια τα δεδομένα των πειραμάτων αμφούστηκαν για παρθούντελικά οι μέσοι όροι που θα εμφανιστούν στην επόμενη ενότητα. Σε αυτό το σημείο πρέπει να σημειωθεί ότι σε κάθε χώρο το Turtlebot ξεκινούσε την κίνηση του από συγκεκριμένο σημείο του εκάστοτε χώρου, αφού αρχικά είχε κάνει μία περιστροφή 360 μοιρών ώστε να συλλέξει πληροφορίες για την πρώτη επιλογή στόχου.

Τα πειράματα που πραγματοποιήθηκαν στους 3 χώρους χωρίζονται στις εξής 3 κατηγορίες όπως περιγράφηκαν και παραπάνω:

- (I) Εξερεύνηση μόνο με το Turtlebot.
- (II) Εξερεύνηση με το Drone να εξερευνά τα όρια του OGM.
- (III) Εξερεύνηση με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων.

Ο λόγος κατηγοριοποίησης των πειραμάτων γίνεται ώστε να μπορέσει να διερευνηθεί η απόδοση κάθε μιας από τις υπηρεσίες υποβοήθησης που παρέχει το Drone στον τρόπο εξερεύνησης.



(c) Τρίτος Πειραματικός Χώρος

Σχήμα 33: Εικόνες που εξήχθησαν μετά την ολοκλήρωση των πειραμάτων από το εργαλείο Rviz

Ακόμα, όπως έχει ήδη αναφερθεί και στην υποενότητα 4.4.4 του τέταρτου κεφαλαίου, ανάλογα με τον σκοπό της συνεργασίας ωστε πρέπει να δούσονταν τα αντίστοιχα βάρη για το κάθε κόστος που υπολογίστηκε. Ο παρακάτω πινάκας παρουσιάζει τα βάρη ($Coe f_i$), που χρησιμοποιήθηκαν για κάθε μια κατηγορία πειραμάτων που διεξήχθει σύμφωνα με τον στόχο της εξερεύνησης, όπως περιγράφηκε στην Εξίσωση 4.1. Οι τιμές των βαρών αυτών προσδιορίστηκαν με τη μέθοδο δοκιμών.

Target Selection Costs Weight Assignment Table					
Σκοπός Συνεργασίας	Κατηγορία Πειράματος	Κόστος Απόστασης από Εμπόδια ($Coe f_1$)	Κόστος Απόστασης από Turtlebot ($Coe f_2$)	Κόστος Color Evaluation ($Coe f_3$)	Κόστος Corner Evaluation ($Coe f_4$)
Full Exploration	I	1.2	2.0	-	-
	II	1.2	2.0	-	-
	III	1.2	2.0	4.0	1.0
Full Coverage	I	1.0	-8.0	-	-
	II	1.0	-8.0	-	-
	III	1.0	-8.0	2.0	1.0
Full Coverage Under Time Constraint	I	1.0 → 1.2	-8.0 → 2.0	-	-
	II	1.0 → 1.2	-8.0 → 2.0	-	-
	III	1.0 → 1.2	-8.0 → 2.0	2.0 → 4.0	1.0 → 1.0

Table 2: Πίνακας Βαρών για τα Κόστη της διαδικασίας του Target Selection του Turtlebot

Full Exploration

Όταν η εξερεύνηση εκτελείται μόνο από το Turtlebot και έχει ως σκοπό την πλήρη εξερεύνηση του χώρου, τότε το κόστος της απόστασης του στόχου από τη θέση του ρομπότ ωστε πειδρά θετικά με μεγαλύτερη βαρύτητα από ότι το κόστος της απόστασης του στόχου από τα εμπόδια κατά την αξιολόγηση. Ο λόγος για τον οποίο έγινε αυτή η επιλογή είναι επειδή θέλουμε το Turtlebot να επιλέγει στόχους οι οποίοι είναι μακριά, ώστε να εξερευνηθεί ο χάρτης γρήγορότερα. Η ίδια συνθήκη ισχύει και στην περίπτωση όπου η εξερεύνηση γίνεται και από τους δύο πράκτορες, με τον εναέριο πράκτορα να εξερευνά τα όρια του OGM, ώστε να γίνει γρήγορη εξερεύνηση του χώρου. Τέλος, στην περίπτωση όπου συμμετέχουν και οι δύο πράκτορες στην εξερεύνηση, με τον εναέριο πράκτορα να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων, το κόστος του Color Evaluation έχει τη μεγαλύτερη βαρύτητα, ενώ η βαρύτητα του κόστους του Corner Evaluation είναι η μικρότερη. Επιπλέον, το κόστος της απόστασης του στόχου από τη θέση του ρομπότ και από τα εμπόδια έχει την ίδια βαρύτητα με τις προηγούμενες περιπτώσεις. Η λογική αυτή επιλέχθηκε καθώς για τη γρήγορη εξερεύνηση του χώρου είναι σημαντικό να επιλέγουμε στόχους που αντιστοιχούν σε περιοχές με χαμηλή πυκνότητα εμποδίων. Το κόστος της πυκνότητας των γωνιών δεν επηρεάζει σημαντικά την αξιολόγηση καθώς είναι ένας παράγοντας από τον οποίο δε μπορούν να εξαχθούν πολλά συμπεράσματα.

Full Coverage

Αρχικά, στην περίπτωση όπου εκτελείται εξερεύνηση με σκοπό την πλήρη κάλυψη του χώρου μόνο από τον επίγειο πράκτορα, το κόστος απόστασης του στόχου από τα εμπόδια επιδρά θετικά με μικρή βαρύτητα, ενώ η απόσταση του στόχου από το Turtlebot έχει αρνητική επιδραση με μεγάλη βαρύτητα. Επειδή ο σκοπός είναι η πλήρης κάλυψη ο επίγειος πράκτορας πρέπει να επιλέγει στόχους οι οποίοι βρίσκονται πιο κοντά του ώστε να ολοκληρώνεται η κάλυψη μίας περιοχής πριν προχωρήσει στην επόμενη. Η απόσταση του στόχου από τα εμπόδια δεν έχει μεγάλη σημασία για την πλήρη κάλυψη του χώρου. Ακριβώς η ίδια επιλογή έγινε και στην περίπτωση όπου η εξερεύνηση πραγματοποιείται και από τους δύο πράκτορες, με τον εναέριο πράκτορα να εξερευνά τα όρια του OGM. Στην τρίτη περίπτωση όπου συνεργάζονται και οι δύο πράκτορες, με τον εναέριο πράκτορα να εξερευνάει τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων, οι βαρύτητες του κόστους της απόστασης του στόχου από τα εμπόδια και από το Turtlebot είναι οι ίδιες με τις δύο προηγούμενες περιπτώσεις. Επιπλέον, προστίθενται τόσο το κόστος του Color Evaluation πολλαπλασιασμένο με μία, μέτριας τιμής, θετική βαρύτητα, όσο και το κόστος του Corner Evaluation, το οποίο διατηρεί και σε αυτό το σενάριο εξερεύνησης την ίδια χαμηλή θετική βαρύτητα.

Full Coverage under Time Constraint

Το τελευταίο είδος εξερεύνησης είναι η πλήρης κάλυψη του χώρου κάτω από χρονικούς περιορισμούς. Σε αυτή την περίπτωση, οι βαρύτητες που πολλαπλασιάζονται με τα κόστη, μεταβάλλονται σύμφωνα με το χρόνο μέχρι το πέρασμα των 7 λεπτών, ενώ στη συνέχεια διατηρούν τη μέγιστη τιμή τους. Όπως φαίνεται και στον παραπάνω πίνακα, και στα τρία είδη εξερεύνησης οι συντελεστές που αφορούν τα κόστη αρχικά έχουν τις τιμές που επιλέχθηκαν στην περίπτωση της πλήρης κάλυψης του χώρου, ενώ μετά το πέρας των 7 λεπτών διατηρούν σταθερά τις τιμές που χρησιμοποιήθηκαν για την εξερεύνηση του χώρου. Μέσα στο διάστημα των 7 πρώτων λεπτών οι τιμές των συντελεστών αλλάζουν δυναμικά, μέσω μίας συνάρτησης, ανάλογα με το χρόνο που έχει περάσει.

5.3 Αποτελέσματα Πειραμάτων

Για να εξαχθούν συμπεράσματα για τον τρόπο συνεργασίας που πραγματοποιήθηκε στην ενότητα αυτή θα παρουσιαστούν τα διαγράμματα ποσοστού εξερεύνησης και κάλυψης του χώρου συνάρτησή του χρόνου, όπως αναφέρθηκαν στην αρχή του κεφαλαίου. Σε αυτό το σημείο πρέπει να αναφερθεί ότι ο πρώτος πειραματικός χώρος αποτελείται από διαδόμους διαφόρων διαστάσεων, με σκοπό να δοκιμαστεί η υλοποίηση και σε μία ιδιάζουσας μορφή χώρου, όπως οι διάδρομοι. Ο δεύτερος χώρος περιέχει εμπόδια διαφόρων μεγεθών τα οποία είναι πυκνώς τοποθετημένα, σε αντίθεση με τον τρίτο πειραματικό χώρο ο οποίος έχει μικρή περιεκτικότητα σε εμπόδια. Τέλος, σε όλα τα διαγράμματα

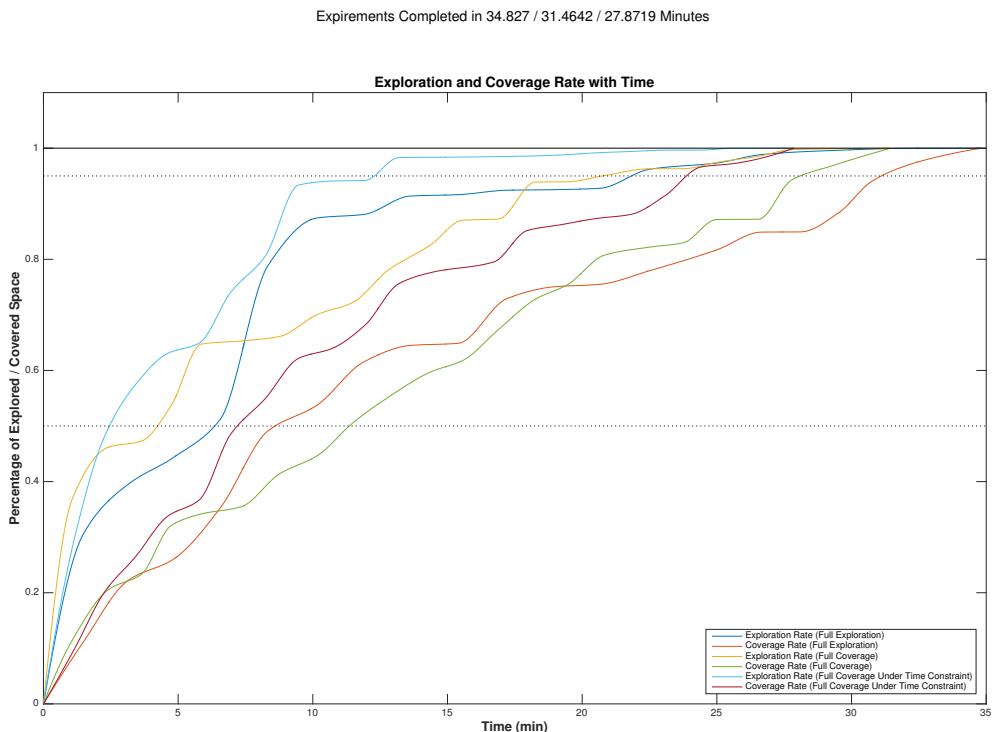
- η μπλε και η κοκκινη γραμμή αντιπροσωπεύουν τις καμπύλες εξερεύνησης και κάλυψης αντίστοιχα, για τα πειράματα με σκοπό την πλήρη εξερεύνηση (Full Exploration)
- η κίτρινη και η πράσινη γραμμή αντιπροσωπεύουν τις καμπύλες εξερεύνησης και κάλυψης αντίστοιχα, για τα πειράματα με σκοπό την πλήρη κάλυψη (Full Coverage)
- η γαλάζια και η μωβ γραμμή αντιπροσωπεύουν τις καμπύλες εξερεύνησης και κάλυψης αντίστοιχα, για τα πειράματα με σκοπό την πλήρη κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constraint)

5.3.1 Εξερεύνηση μόνο με το Turtlebot

Διεξήχθησαν συνολικά 18 πειράματα εξερεύνησης των 3 χώρων για την περίπτωση εξερεύνησης μόνο το Turtlebot. Τα διαγράμματα που προέκυψαν από τα πειράματα αυτά με την βοήθεια του Matlab παρουσιάζονται παρακάτω.

5.3.1.1 Πειράματα στον πρώτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερεύνημένης και της καλυμμένης περιοχής που αφορούν τον πρώτο χώρο, ενώ η εξερεύνηση εκτελείται μόνο από τον επίγειο ρομποτικό πράκτορα.

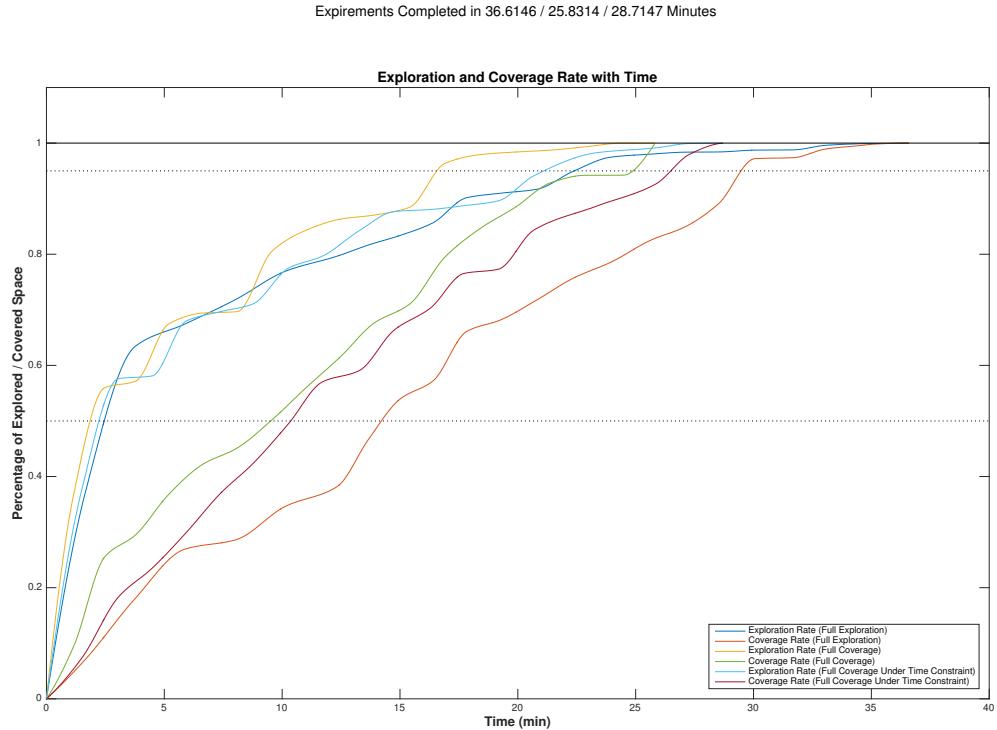


Σχήμα 34: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 1 μόνο από το UGV

Για τις καμπύλες που αφορούν την πλήρη εξερεύνηση (Full Exploration) πρέπει να επισημανθεί ότι παρατηρώντας την καμπύλη του ρυθμού εξερεύνησης προκύπτει ότι το Turtlebot σε μικρό χρονικό διάστημα (10 λεπτών) έχει εξερευνήσει το 90% του χώρου, ενώ από τις καμπύλες της μεθόδου πλήρους κάλυψης παρατηρείται η μικρή απόσταση μεταξύ της καμπύλης του ρυθμού εξερεύνησης και της καμπύλης του ρυθμού κάλυψης. Μόλις το Turtlebot εξερευνήσει μία νέα περιοχή, κινείται στην περιοχή αυτή μέχρι να την καλύψει. Παρόλα αυτά, στο τέλος της διαδικασίας φαίνεται ότι η επιλογή των στόχων δεν ήταν βέλτιστη, καθώς για μεγάλο χρονικό διάστημα το ποσοστό κάλυψης παραμένει σταυρό. Κάτι τέτοιο σημαίνει πως δεν είχαν καλυφθεί πλήρως κάποιοι υποχώροι. Στο τρίτο και τελευταίο ζεύγος καμπυλών της πλήρους κάλυψης σε περιορισμένο χρόνο είναι ορατό ότι πριν τα 7 λεπτά εκτελείται μία δυναμική προσέγγιση της πλήρους κάλυψης, ενώ στη συνέχεια ξεκινά η μέθοδος της πλήρους εξερεύνησης όπου σε σύντομο χρονικό διάστημα (4 λεπτών) το ποσοστό εξερευνημένης περιοχής αυξάνεται ραγδαίως από το 75% στο 95% χωρίς να συμβαίνει το ίδιο και στην καμπύλη της καλυμμένης περιοχής.

5.3.1.2 Πειράματα στο δεύτερο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυμμένης περιοχής που αφορούν το δεύτερο χώρο, ενώ η εξερεύνηση εκτελείται μόνο από τον επίγειο ρομποτικό πράκτορα.

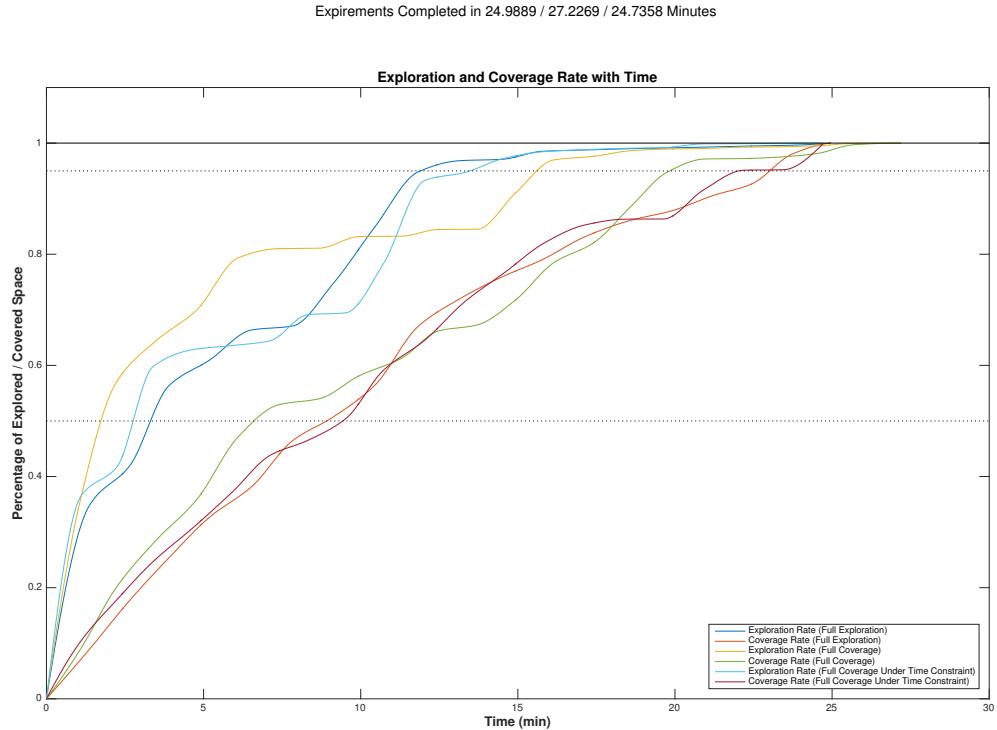


Σχήμα 35: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 2 μόνο από το UGV

Ο λόγος για τον οποίο στο πρώτο ζεύγος καμπυλών η καμπύλη του ρυθμού εξερεύνησης αυξάνεται απότομα μόνο στα πρώτα 3 λεπτά, είναι ότι ο χώρος αυτός έχει πυκνά εμπόδια. Έτσι, τα πυκνά εμπόδια δεν επιτρέπουν τις ακτίνες του LIDAR του Turtlebot να ταξιδέψουν μακριά, με αποτέλεσμα η καμπύλη αυτή να ακολουθεί πιο ομαλή πορεία. Για το δεύτερο ζεύγος της πλήρους κάλυψης μπορεί να εξαχθεί το συμπέρασμα ότι το Turtlebot εξερευνούσε μικρές περιοχές, τις οποίες κάλυπτε, πριν εξερευνήσει περισσότερες. Τέλος, όσον αφορά τις καμπύλες της πλήρους κάλυψης σε περιορισμένο χρόνο, από τη γραμμικότητα της καμπύλης του ρυθμού κάλυψης στα πρώτα λεπτά είναι φανερό ότι εκτελείται η λογική της δυναμική προσέγγιση της πλήρους κάλυψης. Μετά τα 7 λεπτά εκτελείται λογική πλήρους εξερεύνησης κάτι όμως που δεν είναι ορατό από την αντίστοιχη καμπύλη. Ο λόγος για τον οποίο συμβαίνει κάτι τέτοιο είναι είναι ο ίδιος που εξηγήθηκε λίγο παραπάνω όταν εκτελέστηκε η μέθοδος πλήρους εξερεύνησης.

5.3.1.3 Πειράματα στο τρίτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυψμένης περιοχής που αφορούν τον τρίτο χώρο, ενώ η εξερεύνηση εκτελείται μόνο από τον επίγειο ρομποτικό πράκτορα.



Σχήμα 36: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 μόνο από το UGV

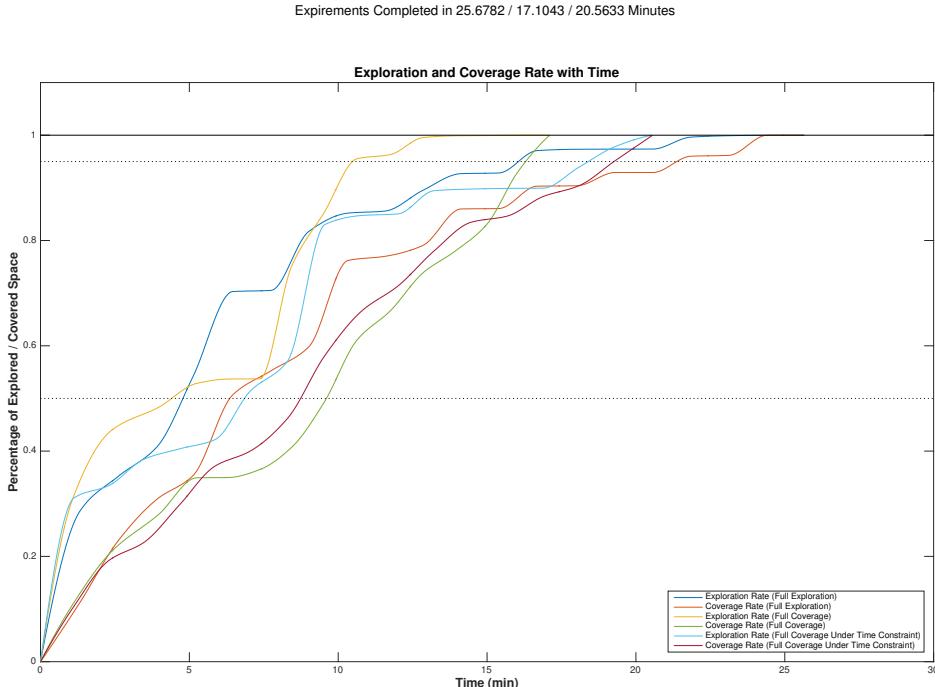
Από το πρώτο ζεύγος καμπυλών (πλήρη εξερεύνηση) είναι ορατό ότι το Turtlebot εξερευνεί επιτυχώς πάνω από το 95% του συνολικού χώρου σε λιγότερο από 15 λεπτά. Παρατηρώντας το δεύτερο ζεύγος (πλήρη κάλυψη) φαίνεται ότι οι δύο καμπύλες έχουν παρόμοιες κυματομορφές, αποτέλεσμα αναμενόμενο αναλογιζόμενο τη λογική της πλήρους κάλυψης η οποία αναφέρθηκε εκτενώς παραπάνω. Τέλος, παρατηρώντας το τρίτο ζεύγος καμπυλών και αυτού του χώρου, η δυναμική προσέγγιση της πλήρους κάλυψης είναι γραμμική στα πρώτα 7 λεπτά, ενώ μετά το πέρασμα των 7 λεπτών η καμπύλη εξερεύνησης αυξάνεται ραγδαίως μέσα σε 3 λεπτά από το 70% στο 95% χωρίς η καμπύλη κάλυψης να την ακολουθεί.

5.3.2 Εξερεύνηση και με τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM

Διεξάχθηκαν συνολικά 18 πειράματα εξερεύνησης των 3 χώρων για την περίπτωση εξερεύνησης και με τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM. Τα διαγράμματα που προέκυψαν από τα πειράματα αυτά με την βοήθεια του Matlab παρουσιάζονται παρακάτω.

5.3.2.1 Πειράματα στον πρώτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερεύνημένης και της καλυμμένης περιοχής που αφορούν τον πρώτο χώρο, ενώ η εξερεύνηση εκτελείται και με τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM.



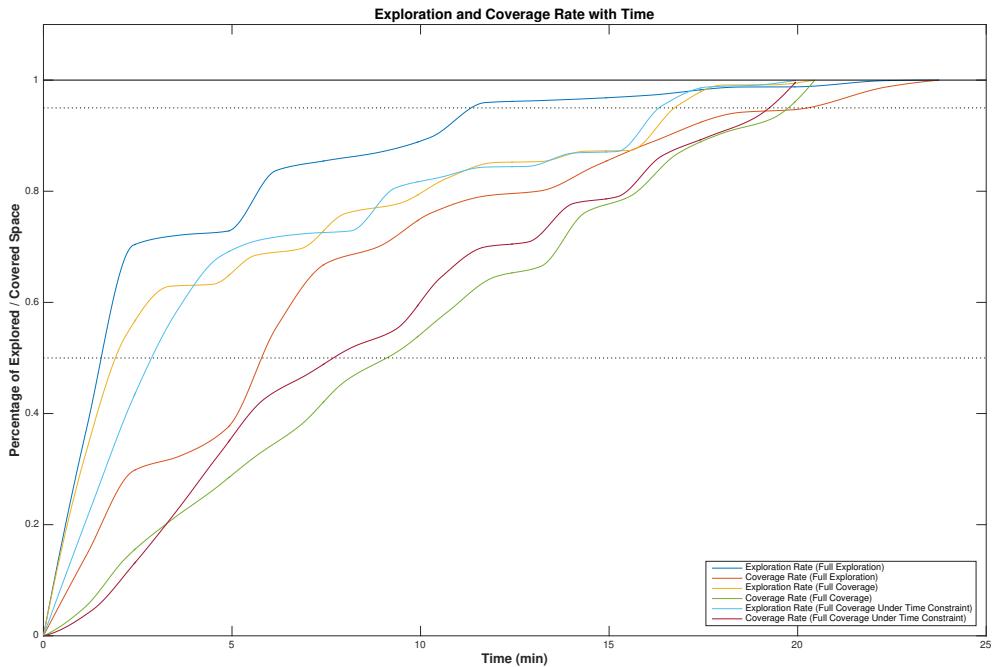
Σχήμα 37: Διαγράμματα Πειραμάτων εξερεύνησης του χώρο 1 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM

Από τις πρώτες δύο καπύλες της πλήρους εξερεύνησης φαίνεται ότι έχουν παρόμοια κυματομορφή. Ο λόγος για τον οποίο συμβαίνει κάτι τέτοιο είναι ότι για να προσεγγίσει το Turtlebot κάποιο μακρινό στόχο, τις περισσότερες φορές χρειάζεται να κινηθεί ανάμεσα στους διαδρόμους τους οποίους έχει ήδη εξερευνήσει, με αποτέλεσμα να τους καλύπτει. Στο δεύτερο ζεύγος παρατηρείται ότι το Turtlebot αρχικά εξερευνά μία μεγάλη περιοχή του χώρου την οποία καλύπτει, πριν προχωρήσει στην εξερεύνηση της επόμενης περιοχής, όπως αναφέρει η λογική της πλήρους κάλυψης. Τέλος όπως φαίνεται από τις δύο τελευταίες καμπύλες, μέχρι τα 7 λεπτά εκτελεί λογική δυναμικής προσέγγισης της πλήρους κάλυψης. Στα επόμενα 13.5 λεπτά του πειράματος, αρχικά το Turtlebot εξερευνεί μία μεγάλη περιοχή, αλλά στη συνέχεια οι μακρινοί στόχοι οι οποίοι επιλέγονται φαίνεται να συμβάλουν και στην αύξηση της πλήρους κάλυψης εξίσου.

5.3.2.2 Πειράματα στο δεύτερο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυμμένης περιοχής που αφορούν το δεύτερο χώρο, ενώ η εξερεύνηση εκτελείται και με τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM.

Expirements Completed in 23.7758 / 20.4567 / 19.9968 Minutes

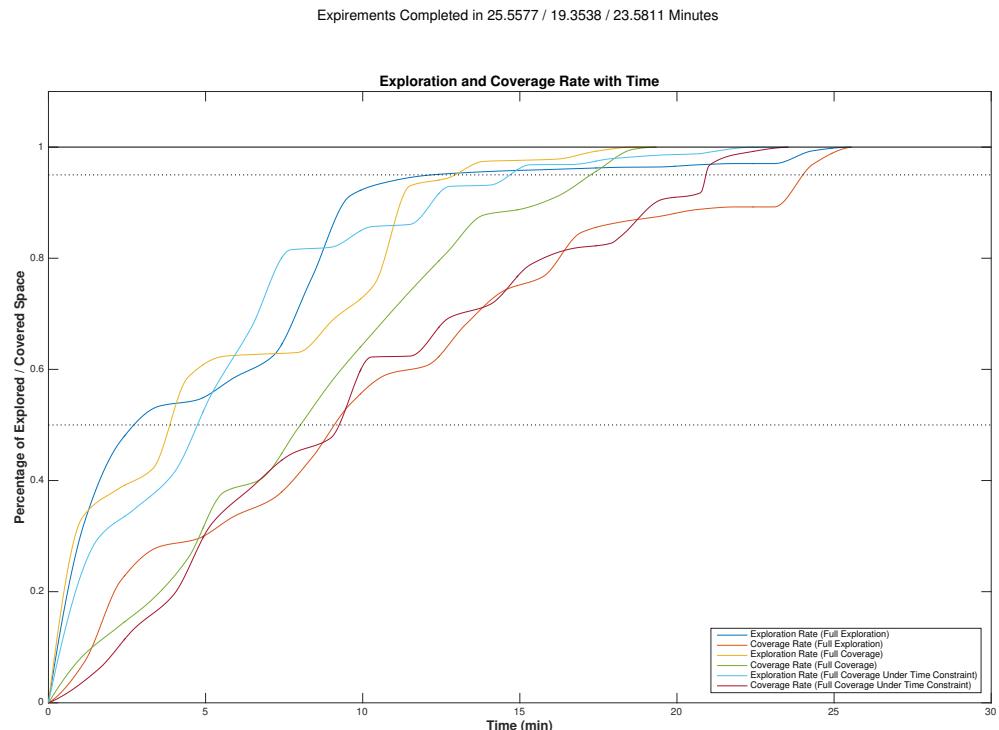


Σχήμα 38: Διαγράμματα Πειραμάτων εξερεύνησης του χώρο 2 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM

Παρατηρώντας το πρώτο ζεύγος, της πλήρους εξερεύνησης, στα πρώτα λεπτά το ποσοστό της εξερευνημένης περιοχής αυξάνεται ραγδαία ενώ στη συνέχεια αυτή τάση εξωμαλύνεται, με τις δύο καμπύλες να πλησιάζουν. Η εξερευνημένη περιοχή φτάνει στο 95% στα 12 λεπτά όπου και τελειώνει ο στόχος του πειράματος αυτού. Το ίδιο μοτίβο φαίνεται και στα άλλα δύο διαγράμματα. Από το δεύτερο ζεύγος είναι φανερό ότι πρόκειται για πλήρη κάλυψη, καθώς η καμπύλη του ρυθμού της κάλυψης αυξάνεται σχεδόν γραμμικά με το χρόνο. Τέλος, στο τρίτο ζεύγος καμπυλών, όπως έχει ήδη αναφερθεί, μέχρι τα 7 λεπτά η μέθοδος εξερεύνησης μεταβάλλεται δυναμικά από πλήρη κάλυψη σε πλήρη εξερεύνηση, ενώ μετά το πέρας του επταλέπτου χρησιμοποιείται η συνάρτηση για πλήρη εξερεύνηση.

5.3.2.3 Πειράματα στο τρίτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυψμένης περιοχής που αφορούν τον τρίτο χώρο, ενώ η εξερεύνηση εκτελείται και με τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM.



Σχήμα 39: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM

Παρατηρώντας το πρώτο ζεύγος, το οποίο αναφέρεται στην πλήρη εξερεύνηση του χώρου, είναι ορατό ότι μέσα στα 15 λεπτά έχει ολοκληρωθεί η πλήρης εξερεύνηση (ποσοστό εξερευνημένου χάρτη 95%). Στη συνέχεια χρειάζονται άλλα 15 λεπτά να εκτελεστεί να ολοκληρωθεί η πλήρης κάλυψη του χώρου αυτού, κάτι όμως που αναπαρίσταται για λόγους σύγχρισης που όχι φανούν στη συνέχεια. Ο σκοπός του πειράματος αυτού ολοκληρώνεται στα 15 λεπτά. Το δεύτερο ζεύγος καμπυλών αφορά την πλήρη κάλυψη του χώρου. Κατά τη διάρκεια αυτού του πειράματος επιλέγονται στόχοι οι οποίοι βρίσκονται κοντά στο Turtlebot και για αυτό η καμπύλη της εξερευνημένης περιοχής δεν απέχει πολύ από την καμπύλη της καλυψμένης περιοχής. Τέλος, στις δύο καμπύλες της υβριδικής μεθόδου μέχρι τα 7 λεπτά οι δύο καμπύλες δεν απέχουν, πράγμα αναμενόμενο καθώς ακολουθείται η δύναμική προσέγγιση της πλήρους κάλυψης, ενώ μετά τα 7 λεπτά εφαρμόζεται η λογική της πλήρους εξερεύνησης.

5.3.3 Εξερεύνηση με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων

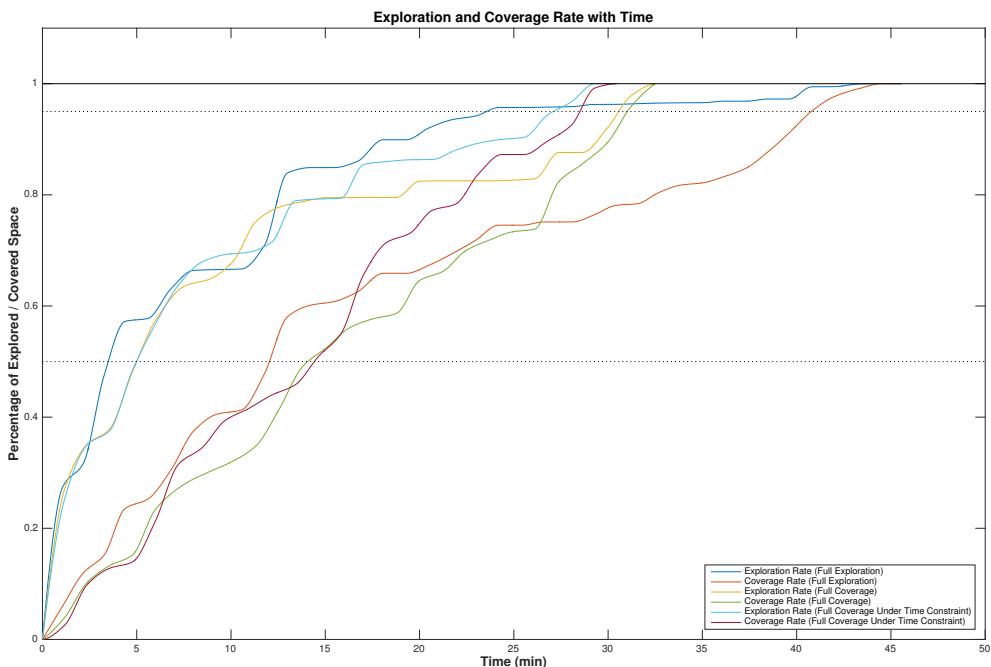
Σε αυτή μέθοδο συνεργασίας των δύο πρακτόρων οι χρόνοι ολοκλήρωσης ήταν μεγαλύτεροι από τις δύο προηγούμενες μεθόδους. Παρόλα αυτά το ενδιαφέρον σημείο είναι ότι σε αυτή τη μέθοδο οι στόχοι που επιλέγονται για τη εξερεύνηση είναι πιο ποιοτικοί. Μπορεί, λοιπόν, σε τέτοιου μεγέθους χώρους να μην έχει αποτέλεσμα, σε μεγάλους, όμως, χώρους όπου μία λάθος επιλογή στόχου θα κοστίσει περισσότερο, η διαφορά θα είναι αισθητή.

Διεξάχθηκαν συνολικά 18 πειράματα εξερεύνησης των 3 χώρων για την περίπτωση εξερεύνησης με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων. Τα διαγράμματα που προέκυψαν από τα πειράματα αυτά με την βοήθεια του Matlab παρουσιάζονται παρακάτω.

5.3.3.1 Πειράματα στον πρώτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερεύνημένης και της καλυμμένης περιοχής που αφορούν τον πρώτο χώρο, ενώ η εξερεύνηση πραγματοποιείται από τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων.

Expirements Completed in 45.5952 / 32.5186 / 30.4809 Minutes



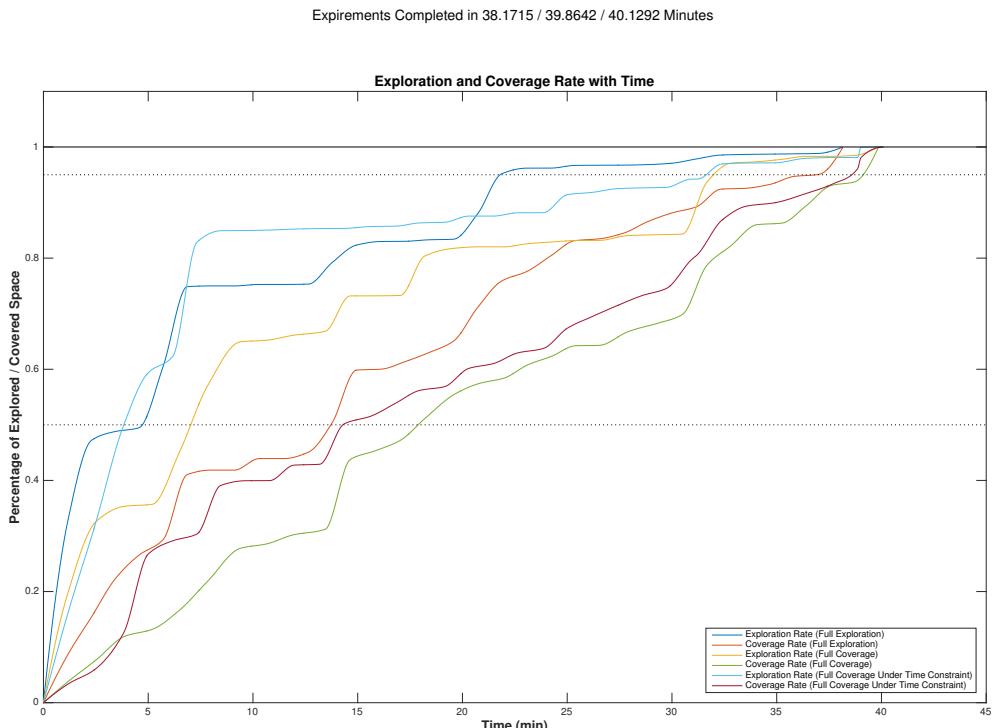
Σχήμα 40: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 1 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους

Το πρώτο ζεύγος καμπυλών αναφέρεται σε μέθοδο πλήρης εξερεύνησης και από την αντίστοιχη καμπύλη φαίνεται ότι έχει εξερευνήσει το 95% του χώρου στα 25 λεπτά, χρόνος παρόμοιος με αυτόν της αντίστοιχης εξερεύνησης από τον επίγειο πράκτορα όταν αυτός δρούσε μόνος του. Παρόλα αυτά, όπως έχει ήδη αναφερθεί, έχει γίνει πολύ πιο ποιοτική

επιλογή στόχων, όπως θα φανεί και στην συνέχεια. Ο αυξημένος χρόνος υλοποίησης είναι ορατός και από το δεύτερο ζεύγος όπου οι δύο καμπύλες ακολουθούν τη λογική της πλήρους κάλυψης. Τέλος, το τελευταίο ζεύγος, το οποίο αφορά τη πλήρη κάλυψη σε περιορισμένο χρόνο, στα πρώτα 7 λεπτά εκτελείται η λογική της δυναμικής προσέγγισης της πλήρους κάλυψης η οποία αντικατοπτρίζεται και στις αντίστοιχες καμπύλες, ενώ μετά το πέρασμα των 7 λεπτών παρατηρείται η ίδια συμπεριφορά με το αντίστοιχο διάγραμμα στη μέθοδο συνεργασίας του Drone και του Turtlebot με το UAV να εξερευνά τα όρια του OGM. Η αυτία είναι η ίδια, δηλαδή οι μακρινοί στόχοι οι οποίοι επιλέγονται φαίνεται να συμβάλουν και στην αύξηση της πλήρους κάλυψης εξίσου.

5.3.3.2 Πειράματα στο δεύτερο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυμμένης περιοχής που αφορούν το δεύτερο χώρο, ενώ η εξερεύνηση πραγματοποιείται από τους δύο πράκτορες, με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων.



Σχήμα 41: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 2 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους

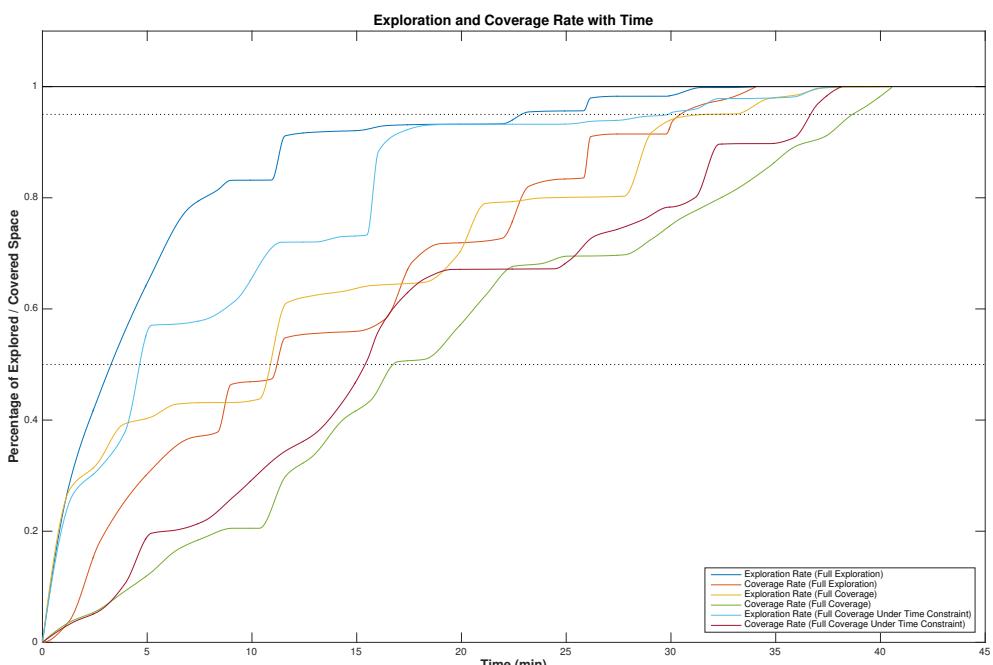
Ο χώρος αυτός αποτελείται από διαφόρου μεγέθους εμπόδια, πυκνώς τοποθετημένα. Παρατηρώντας τις πρώτες δύο καμπύλες, ο επίγειος πράκτορας εξερευνεί το 95% του χώρου σε 23 λεπτά, ενώ απαντούνται άλλα 15 λεπτά για την πλήρη κάλυψη, πράγμα αναμενόμενο για τη συγκεκριμένη λογική. Το δεύτερο ζεύγος της πλήρους κάλυψης απεικονίζεται αντιπροσωπευτικά η λογική αυτή, καθώς το Turtlebot εξερευνά και καλύπτει διαδοχικά τις διάφορες περιοχές μέσα στο χώρο. Τέλος, από το τρίτο και τελευταίο ζεύγος φαίνεται ότι στα πρώτα 7 λεπτά η δυναμική προσέγγιση της πλήρους κάλυψης εκτελεί ένα άλμα από το 60% στο 85% μόλις σε 3 λεπτά αλλά παραμένει σταθερή σε αυτό το ποσοστό για 18 λεπτά. Αυτό οφείλεται στην

μεγάλη περιεκτικότητα του χώρου σε εμπόδια. Συγκεκριμένα, οι ανεξερεύνητες περιοχές περιείχαν πολλά εμπόδια μεγάλη πράγμα που σημαίνει ότι η αξιολόγηση των στόχων που βρίσκονται κοντά σε αυτές τις περιοχές επιδρούσε αρνητικά στην επιλογή τους.

5.3.3.3 Πειράματα στο τρίτο χώρο

Στα παρακάτω διαγράμματα φαίνονται οι καμπύλες του ρυθμού αύξησης της εξερευνημένης και της καλυμμένης περιοχής που αφορούν τον τρίτο χώρο, ενώ η εξερεύνηση πραγματοποιείται από τους δύο πράκτορες με το Drone να εξερευνάει τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πιθανών στόχων.

Expirements Completed in 34.0651 / 40.5795 / 38.1619 Minutes



Σχήμα 42: Διαγράμματα Πειραμάτων εξερεύνησης του χώρου 3 από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους

Από το πρώτο ζεύγος είναι φανερό το απότομο άλμα της καμπύλης εξερεύνησης, όπου μέσα σε 10 λεπτά εξερευνήθηκε 85% του χώρου. Παρόλα αυτά κάτι τέτοιο δεν είναι περιέργο με δεδομένο ότι ο χώρος είναι ανοιχτός από εμπόδια. Στο δεύτερο ζεύγος, όπως και στα αντίστοιχα των υπόλοιπων χώρων, η λογική της πλήρους κάλυψης απεικονίζεται στις δύο καμπύλες, δηλαδή το Turtlebot εξερευνεί και καλύπτει διαδοχικά τις περιοχές του χώρου. Τέλος, όσον αναφορά το τρίτο ζεύγος μέχρι τα 7 λεπτά ακολουθείται η δυναμική προσέγγιση της πλήρους κάλυψης.

5.4 Συγκρίσεις Αποτελεσμάτων των Πειραμάτων

Στην ενότητα αυτή όμως γίνουν οι απαραίτητες συγκρίσεις, ώστε να μπορούν να εξαχθούν εύλογα συμπεράσματα. Για τη διευκόλυνση του αναγνώστη δημιουργήθηκαν συγχριτικά διαγράμματα τα οποία παρουσιάζονται στο παρότρημα αποτελεσμάτων και αναλύονται παρακάτω. Ο στόχος της σύγκρισης των τριών μεθόδων εξερεύνησης του χώρου είναι ο προσδιορισμός της χρηστικότητας της συνεργασίας των δύο ρομποτικών πρωτότορων ανάλογα με το χώρο και το σκοπό της εξερεύνησης.

5.4.1 Σύγκριση Πειραματικών Κατηγοριών I και II

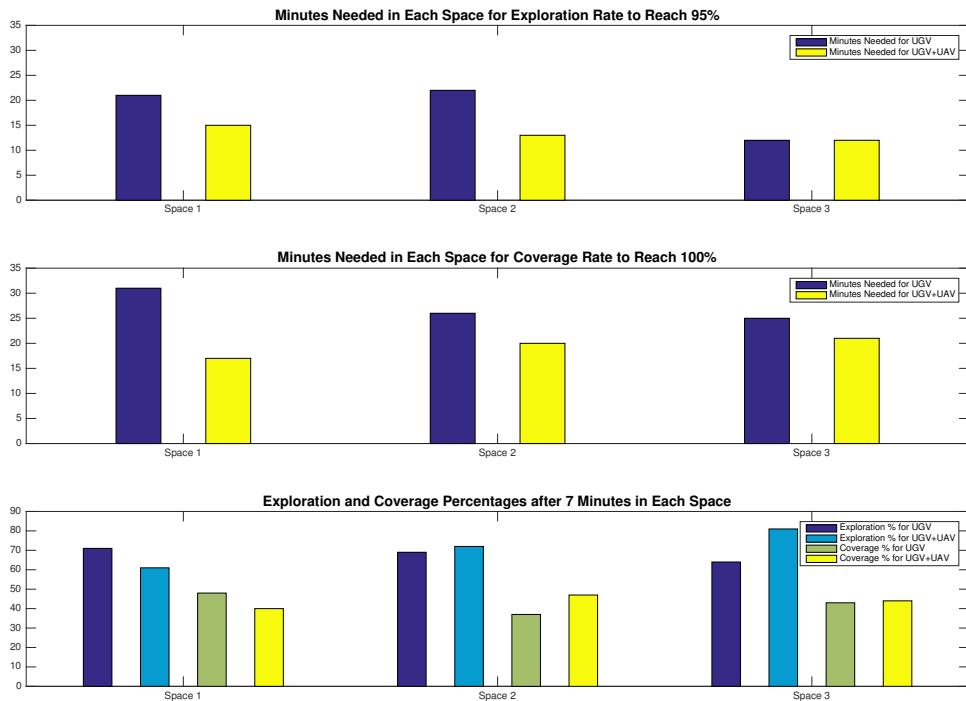
Ο παρακάτω πίνακας περιέχει τους χρόνους εκτέλεσης της εξερεύνησης για τους τρεις σκοπούς εξερεύνησης στους τρεις πειραματικούς χώρους για τις δύο κατηγορίες (UGV only (I)/ UAV - UGV with OGM exploration (II)). .

Πίνακας Πειραματικών Αποτελεσμάτων				
Σκοπός Συνεργασίας	Κατηγορία Πειράματος	Πειραματικός Χώρος 1	Πειραματικός Χώρος 2	Πειραματικός Χώρος 3
Full Exploration	I	21 λεπτά	22 λεπτά	12 λεπτά
	II	15 λεπτά	13 λεπτά	12 λεπτά
Full Coverage	I	12 λεπτά / 31 λεπτά	9.5 λεπτά / 26 λεπτά	7.5 λεπτά / 27 λεπτά
	II	9.5 λεπτά / 17 λεπτά	8.5 λεπτά / 20 λεπτά	8 λεπτά / 20 λεπτά
Full Coverage Under Time Constraint	I	71% / 48%	69% / 37%	64% / 43%
	II	61% / 40%	72% / 47%	81% / 44%

Πίνακας 3: Πίνακας Συγκριτικών Αποτελεσμάτων για τις Κατηγορίες Πειραμάτων I και II

Πρέπει να σημειωθεί ότι:

- οι τιμές για **Full Exploration** αντιστοιχούν στα λεπτά που το ποσοστό του εξερευνημένου χώρου είναι στο 95%.
- οι τιμές για **Full Coverage** αναφέρονται στα λεπτά που το ποσοστό του εξερευνημένου χώρου είναι στο 50% / 100%.
- οι τιμές για **Full Coverage Under Time Constraint** αναφέρονται στα ποσοστά του εξερευνημένου χώρου / κάλυψης στα 7 λέπτα.



Σχήμα 43: Συγκριτικό Ραβδόγραφμα Πειραμάτων για τις Κατηγορίες Πειραμάτων I και II

Στο πρώτο και στο δεύτερο ραβδόγραφμα του Σχήματος 43 φαίνονται οι χρόνοι που απαιτήθηκαν για την ολοκλήρωση της πλήρης εξερεύνησης και της πλήρης κάλυψης αντίστοιχα, για τους 3 πειραματικούς χώρους. Με μπλέ χρώμα απεικονίζονται οι χρόνοι που αφορούν την εξερεύνηση μόνο από το UGV, ενώ με κίτρινο, αφορούν τους χρόνους της εξερεύνησης από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM. Το τελευταίο ραβδόγραφμα απεικονίζει τα ποσοστά εξερευνημένου και καλυμμένου χώρου μετά το πέρας των 7 λεπτών. Με σκούρο και ανοιχτό μπλε φαίνονται τα ποσοστά του εξερευνημένου χώρου όταν η εξερεύνηση γίνεται μόνο με UGV και όταν η εξερεύνηση γίνεται από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM αντίστοιχα. Τέλος το πράσινο χρώμα αφορά το ποσοστό κάλυψης για εξερεύνηση μόνο με UGV, ενώ με κίτρινο, απεικονίζεται το ποσοστό κάλυψης που αφορά την εξερεύνηση που γίνεται από τους δύο πράκτορες, με το UAV να εξερευνά τα όρια του OGM.

Με βάση τα διάγραμμα 43,46,49 που φαίνονται στο παράρτημα των αποτελεσμάτων και σε συνδυασμό με τον παραπάνω πίνακα, όσον αφορά την πλήρη εξερεύνηση του χώρου (Full Exploration) η συνεργασία των ρομπότ είναι επιισυμητή κάτω από συγκεκριμένες συνθήκες. Αναλυτικότερα τόσο στην περίπτωση των διαδρόμων όσο και των πυκνών εμποδίων, η πλήρης εξερεύνηση του χώρου πραγματοποιήθηκε γρηγορότερα. Αντιθέτως στην περίπτωση των αραιών εμποδίων η συνεργασία των δύο ρομποτικών πρακτόρων δεν είναι πιο αποτελεσματική.

Προχωρώντας στα διαγράμματα 44,47,50, όσον αφορά την πλήρη κάλυψη του χώρου (Full Coverage) η συνεργασία των ρομπότ οδηγεί σε καλύτερα αποτελέσματα και στους τρεις χώρους. Πιο συγκεκριμένα, για τον πρώτο χώρο η δεύτερη μέθοδος με τη βοήθεια της κάμερας του Drone οδηγεί στην γρηγορότερη ολοκλήρωση της κάλυψης, πράγμα που οφείλεται στο γεγονός ότι τα εμπόδια του χώρου αποτελούνται από ευθείες γραμμές, το οποίο βοηθάει στη διαδικασία εξερεύνησης των ορίων. Όσον αφορά τους άλλους δύο χώρους αν και οι δύο καμπύλες κάλυψης (UGV only / UAV-UGV) έχουν τον ίδιο ρυθμό

μέχρι το ποσοστό κάλυψης περίπου 85%, στην περίπτωση της συνεργασίας η ολοκλήρωση της κάλυψης γίνεται ταχύτερα.

Σύμφωνα με τα διαγράμματα 45,48,51 η πλήρης κάλυψη του χώρου σε περιορισμένο χρόνο (Full Coverage under Time Constraint) με τη συνεργασία των ρομπότ, στο χρονικό διάστημα των 7 λεπτών σημείωσε βελτίωση μόνο στην περίπτωση των πυκνών εμποδίων. Στον πρώτο χώρο, ενώ αρχικά δεν υπήρξε κάποια βελτίωση, μετά το πέρασμα των 7 λεπτών η πλήρης κάλυψη του χώρου πραγματοποιήθηκε γρηγορότερα. Στον τρίτο χώρο δε παρατηρείται καμία βελτίωση, αφού οι διαδικασίες ολοκληρώνονται σχεδόν ταυτόχρονα και ο ρυθμός αύξησης των καμπυλών κάλυψης είναι περίπου ίδιος.

5.4.2 Σύγκριση Πειραματικών Κατηγοριών I και III

Όπως αναφέρθηκε στη περιγραφή των διαγραμμάτων, η μέθοδος συνεργασίας των δύο πρακτόρων, με το Drone να εξερευνά τα όρια του OGM και να πραγματοποιεί αξιολόγηση των πυθανών στόχων είναι πιο δαπανηρή χρονικά. Το ενδιαφέρον σημείο, όμως, είναι ότι η διαδικασία της εξερεύνησης του χώρου και για τους 3 σκοπούς αποτελείται από πολύ λιγότερους στόχους. Επομένως, θα μπορούσαν να εξαχθεί συμπεράσματα για το αν οι στόχοι που επιλέγονται με τη συγκεκριμένη μέθοδο είναι πιο ποιοτικοί. Παρακάτω φαίνεται συγκριτικός πίνακας μεταξύ των δύο μεθόδων (UGV only (I)/ UAV - UGV with Potential Target Evaluation (III)).

Πίνακας Πειραματικών Αποτελεσμάτων				
Σκοπός Εξερεύνησης	Κατηγορία Πειράματος	Πειραματικός Χώρος 1	Πειραματικός Χώρος 2	Πειραματικός Χώρος 3
Full Exploration	I	28 στόχοι / 19.5 λεπτά	33 στόχοι / 22 λεπτά	20 στόχοι / 12 λεπτά
	III	14 στόχοι / 21 λεπτά	13 στόχοι / 21 λεπτά	12 στόχοι / 14 λεπτά
Full Coverage	I	21 στόχοι / 39 στόχοι	18 στόχοι / 35 στόχοι	14 στόχοι / 36 στόχοι
	III	11 στόχοι / 30 στόχοι	14 στόχοι / 31 στόχοι	12 στόχοι / 29 στόχοι
Full Coverage Under Time Constraint	I	18 στόχοι → 80% / 74%	16 στόχοι → 69% / 37%	17 στόχοι → 64% / 43%
	III	6 στόχοι → 69% / 35%	6 στόχοι → 82% / 30%	7 στόχοι → 58% / 21%

Πίνακας 4: Πίνακας Συγκριτικών Αποτελεσμάτων για τις Κατηγορίες Πειραμάτων I και III

Πρέπει να σημειωθεί ότι:

- οι τιμές για **Full Exploration** αναφέρονται στον αριθμό των στόχων / στα λεπτά που το ποσοστό του εξερευνημένου χώρου είναι στο 95%.
- οι τιμές για **Full Coverage** αναφέρονται στον αριθμό των στόχων που το ποσοστό του εξερευνημένου χώρου είναι στο 50% / 100%.
- οι τιμές για **Full Coverage Under Time Constraint** αναφέρονται στον αριθμό των στόχων → και στα ποσοστά του εξερευνημένου χώρου / κάλυψης στα 7 λέπτα.

Με βάση τα διαγράμματα 52,55,58, η συγκεκριμένη μέθοδος συνεργασίας αναφορικά με την πλήρη εξερεύνηση του χώρου (Full Exploration), ήταν πιο αποδοτική. Αυτό είναι εμφανές τόσο από τον αριθμό των στόχων που επισκέψθηκε, όσο και από τα λεπτά που απαιτήθηκαν για την ολοκλήρωση της εξερεύνησης. Πιο αναλυτικά, στον πρώτο πειραματικό χώρο η διαδικασία ολοκληρώθηκε με το μισό αριθμό στόχων, ενώ στον δεύτερο, όπου τα εμπόδια είναι ακόμη πιο πυκνά, ο αριθμός των στόχων που χρειάστηκε, ήταν συγκριτικά, ακόμη μικρότερος. Στον τρίτο πειραματικό χώρο, του οποίου η περιεκτικότητα σε εμπόδια είναι χαμηλή, η διαφορά των στόχων που απαιτήθηκαν είναι μεν υπαρκτή, αλλά σε μικρότερο βαθμό. Το φανόμενο αυτό ήταν αναμενόμενο, καθώς η μέθοδος αξιολόγησης των πιθανών στόχων βασίζεται στην πυκνότητα των εμποδίων που είναι ορατά από την κάμερα του Drone.

Για τη μέθοδο της πλήρους κάλυψης (Full Coverage), όπως εμφανίζεται στα διαγράμματα 53,56,59, η συνεργασία των δύο ρομπότ δεν επέφερε κάποια βελτίωση για όλους τους χώρους. Συγκεκριμένα, η ιδιομορφία του πρώτου χώρου (περιλαμβάνει διαδρόμους) ωφέλησε τη συγκεκριμένη μέθοδο συνεργασίας ώστε να είναι και χρονικά αποδοτική. Στον χώρο αυτό τόσο η διαδικασία της εξερεύνησης των ορίων όσο και η διαδικασία αξιολόγησης των πιθανών στόχων από το Drone απέφεραν επιμψημητά αποτελέσματα. Για τους υπόλοιπους χώρους η μέθοδος δεν είχε τα αντίστοιχα αποτελέσματα, αφού χρειάστηκε εμφανώς μεγαλύτερο χρονικό διάστημα και ο αριθμός των στόχων δεν ήταν ικανός να καλύψει τη διαφορά αυτή.

Τέλος, για τη μέθοδο της πλήρους κάλυψης σε περιορισμένο χρόνο (Full Coverage under Time Constraint), παρατηρώντας τα διαγράμματα 54,57,60, τα αποτελέσματα ποικίλουν. Στον πρώτο χώρο κατά τη διάρκεια των 7 λεπτών δε παρατηρείται κάποια βελτίωση. Παρότι η εξερεύνηση διήρκεσε περίπου ίδια λεπτά, ο αριθμός των στόχων που χρειάστηκε ήταν μικρότερος. Έτσι προκύπτει το συμπέρασμα ότι στο χώρο αυτό η εξερεύνηση με τη μέθοδο αυτή οδήγησε σε καλύτερα αποτελέσματα. Στο δεύτερο χώρο η μέθοδος συνεργασίας επέφερε καλύτερα αποτελέσματα στα πρώτα 7 λεπτά, λόγω της πυκνότητας των εμποδίων. Στη συνέχεια, όμως, αφού ο σκοπός της εξερεύνησης άλλαξε σε full exploration, η εξερεύνηση άργησε να ολοκληρωθεί. Η αιτία της καθυστέρησης αυτής είναι ότι οι περιοχές που δεν ήταν εξερευνημένες είχαν μεγάλη περιεκτικότητα σε εμπόδια, με αποτέλεσμα η αξιολόγηση των στόχων κοντά στα σημεία αυτά από το Drone να συμβάλει αρνητικά στην επιλογή τους. Στον τρίτο και τελευταίο χώρο, μέχρι τα 7 λεπτά δε φάνηκε να λειτουργεί βελτιωτικά η μέθοδος αυτή. Αυτό συνεχίστηκε και μετά το πέρασμα των 7 λεπτών αφού ο αριθμός των εμποδίων δεν ήταν μεγάλος.

6

Συμπεράσματα - Σχολιασμός

6.1 Συμπεράσματα

Μετά την ολοκλήρωση των πειραμάτων μπορούν να εξαχθούν τα παρακάτω συμπεράσματα τα οποία αφορούν την επιλογή μίας πίο αποδοτικής μεθόδου εξερεύνησης του χώρου (UGV only/ UGV-UAV με εξερεύνηση στα όρια του χάρτη/ UGV-UAV με εξερεύνηση στα όρια του χάρτη και αξιολόγηση των υποψήφιων στόχων). Η επιλογή αυτή είναι διαφορετική αναλόγως των τύπο του χώρου και το σκοπό της εξερεύνησης. Συγκεκριμένα

- Για χώρους με αραιά εμπόδια
 - στους οποίους θα εκτελεστεί εξερεύνηση με σκοπό **Πλήρη Εξερεύνηση ή Πλήρη κάλυψη σε περιορισμένο χρόνο** η εξερεύνηση από τη συνεργασία των δύο ρομποτικών πρωτότορων δεν ήταν αποδοτική σε σχέση με τη εξερεύνηση από τον επίγειο πράκτορα, όταν δρούσε μόνος του.
 - στους οποίους θα εκτελεστεί εξερεύνηση με σκοπό την **Πλήρη Κάλυψη** η εξερεύνηση από τη συνεργασία των δύο πρωτότορων βελτίωσε το χρόνο ολοκλήρωσης σε σχέση με τη εξερεύνηση μόνο από τον επίγειο πράκτορα.
- Για χώρους με διαδρόμους
 - όπου πρέπει να πραγματοποιηθεί **Πλήρη Εξερεύνηση**, η συνεργασία μόνο με εξερεύνηση ορίων έχει χαμηλότερο χρόνο περάτωσης από ότι η εξερεύνηση μόνο με UGV, ενώ η συνεργασία με εξερεύνηση ορίων και αξιολόγηση υποψήφιων στόχων παρουσίασε ίδιους χρόνους με το UGV αλλά μείωσε τον αριθμό των στόχων.
 - όπου πρέπει να υλοποιηθεί **Πλήρη Κάλυψη**, η συνεργασία μόνο με εξερεύνηση των ορίων του OGM είναι πολύ γρήγορη. Επιπρόσθετα, παρόλο που η εξερεύνηση με εξερεύνηση ορίων και αξιολόγηση χρειάζεται ίδιο χρόνο περάτωσης, συγχριτικά με αυτόν της εξερεύνησης από τον ένα πράκτορα, αλλά εμφανώς λιγότερους στόχους.
 - όπου πρέπει να διεξαχθεί **Πλήρη Κάλυψη σε περιορισμένο χρόνο** η εξερεύνηση με εξερεύνηση ορίων OGM δε παρουσιάζει κάποια βελτίωση στη φάση της δυναμικής προσέγγισης της πλήρους κάλυψης (τα βάρη μεταβάλλονται με το χρόνο), ενώ στη φάση της πλήρους εξερεύνησης η εξερεύνηση ολοκληρώνεται γρηγορότερα.

- Σε χώρους με πυκνά εμπόδια

- για τη πραγματοποίηση **Πλήρους Εξερεύνησης** η συνεργασία μόνο με εξερεύνηση των ορίων του OGM μείωσε το χρόνο περάτωσης σε σχέση με τη εξερεύνηση μόνο με UGV, ενώ η συνεργασία με εξερεύνηση των ορίων OGM και αξιολόγηση υποψήφιων στόχων παρουσίασε ίδιους χρόνους με το UGV αλλά μείωσε τον αριθμό των στόχων.
- για την υλοποίηση **Πλήρους Κάλυψης**, η εξερεύνηση με εξερεύνηση των ορίων OGM επιτυγχάνει το μικρότερο χρόνο περάτωσης.
- με σκοπό **Πλήρη Κάλυψη σε περιορισμένο χρόνο**, η εξερεύνηση με εξερεύνηση των ορίων OGM είναι αυτή που εμφανίζει καλύτερα αποτελέσματα καθ'όλη τη διάρκεια της εξερεύνησης (πριν και μετά τα 7 λεπτά).

Αξίζει να σημειωθεί ότι στη συνεργασία των ρομπότ με εξερεύνηση του OGM και αξιολόγηση πυκνών στόχων, η βελτίωση του αριθμού των στόχων μπορεί στους συγκεκριμένους χώρους να μην επηρεάζουν σημαντικά την εξερεύνηση, αλλά σε χώρους μεγαλύτερου μεγέθους παίζουν σημαντικό ρόλο. Σε ένα μεγάλο χώρο οι ποιοτικοί στόχοι θα επιταχύνουν τη εξερεύνηση, καθώς θα αποφέγγονται καθυστερήσεις οι οποίες οφείλονται σε αδιέξοδα που μπορεί να βρεθεί ο επίγειος πράκτορας και από τα οποία θα χρειαστεί αρκετό χρόνο για να επιστρέψει.

6.2 Προβλήματα

Ένα πρόβλημα που προέκυψε κατά την πραγματοποίηση της υλοποίησης της παρούσας διπλωματικής είναι ότι στην έκδοση του προσομοιωτή Gazebo που ήταν συμβατή με την υλοποίηση μας δεν υπήρχε η δυνατότητα προσθήκης χρώματος. Για να επιλυθεί το παραπάνω πρόβλημα, η δημιουργία των χώρων πραγματοποιήθηκε στο πρόγραμμα Gazebo 7 όπου δίνεται η δυνατότητα προσθήκης χρώματος και στην συνεχεία εισήχθησαν στο Gazebo 2 ως μοντέλα. Ένα άλλο πρόβλημα που υπήρξε, αλλά το οποίο δεν αφορούσε άμεσα την υλοποίηση, ήταν ότι παρατηρούνταν μεγάλη καθυστέρηση στο σύστημα του υπολογιστή όταν εκτελούνταν η υπηρεσία του OGM Path Planning σε χώρους πολύ μεγάλων διαστάσεων ($10000m^2$). Για τον λόγο αυτό τα πειράματα διεξήχθησαν στους χώρους διαστάσεων που παρουσιάστηκαν στο Κεφάλαιο 5.

6.3 Μελλοντική Μελέτη

Με αφορμή το δεύτερο πρόβλημα που παρουσιάστηκε παραπάνω, προτείνεται η διερεύνηση της συνεργασία των δύο πρακτόρων που επιτεύχθηκε σε χώρους ακόμα μεγαλύτερων διαστάσεων, ώστε να γίνει μια εκτίμηση για την χρηστικότητα της συνεργασίας των πρακτόρων στην περίπτωση αυτή, αλλά και η υλοποίηση των πειραμάτων σε χώρους όπου το ύψος των τοίχων να είναι στα 2.5 μέτρα, όπως ισχύει στην πραγματικότητα. Ακόμα θα μπορούσε να διερευνηθεί η δυνατότητα χρήσης μιας ομάδας από εναέριους πράκτορες ώστε να επιτευχθούν ακόμα καλύτεροι χρόνοι διεκπεραίωσης των υπηρεσιών που προσφέρουν. Μια στρατηγική επίτευξης της παραπάνω ιδέας είναι η αποστολή κάθε εναερίου πράκτορα σε μια περιοχή του χάρτη, για την οποία θα είναι υπεύθυνος για την διεκπεραίωσης των υπηρεσιών που παρέχει η υλοποίηση μας. Τέλος, προτείνεται η προσαρμογή της εφαρμογής για τη χρήση της σε πραγματικά ρομπότ ώστε να μπορεί να δοκιμαστεί και σε πραγματικές συνθήκες. Σε αυτή την περίπτωση θα πρέπει να σημειωθεί ότι στα πειράματα που διεξήχθησαν δε μετρήθηκε ούτε λήφθηκαν υπόψιν οι καθυστερήσεις που προκλήθηκαν λόγω του δικτύου Wifi για την επικοινωνία των δύο πρακτόρων. Για το λόγο αυτό πριν την προσαρμογή θα πρέπει να μετρηθεί και να ενσωματωθεί και αυτό το σφάλμα.

Βιβλιογραφία

- [1] El Houssein Chouaib Harik, Francois Guerin, Frederic Guinand, Jean-Francois Brethe, Herve Pelvillain, Adel Zentout. *"Vision based target tracking using an un-manned aerial vehicle"*. 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO), June 30 2015 - July 2 2015
- [2] El Houssein Chouaib Harik, Frederic Guinand, Herve Pelvillain, Francois Guerin, Frederic Guinand, Jean-Francois Brethe. *"A Decentralized Interactive Architecture for Aerial and Ground Mobile Robots Cooperation"*. 2015 International Conference on Control, Automation and Robotics (ICCAR), 20-22 May 2015
- [3] Jae-Keun Lee, Hahmin Jung, Huosheng Hu and Dong Hun Kim. *"Collaborative Control of UAV/UGV"*. The 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2014), 12-15 November 2014
- [4] Luciano Cantelli, M. Lo Presti, Mangiameli Michele, C. D. Melita and Giovanni Muscato. *"Autonomous Cooperation Between UAV and UGV to Improve Navigation and Environmental Monitoring in Rough Environments"*. 10th International Symposium Humanitarian Demining (Hudem), 23-25 April 2013
- [5] Andrea Bertolaso, Masoume M. Raeissi, Alessandro Farinelli and Riccardo Muradore. *"Using Petri Net Plans for Modeling UAV-UGV Cooperative Landing"*. European Conference on Artificial Intelligence (ECAI), 29 August – 2 September 2016
- [6] Yifeng Cai and Sekiyama Kousuke. *"Geometric Relation Matching based Object Identification for UAV and UGV Cooperation"*. Technologies and Applications of Artificial Intelligence (TAAI), Conference , 20-22 November 2015
- [7] Jianqiang Li, Genqiang Deng, Chengwen Luo, Qiuzhen Lin, Qiao Yan and Zhong Ming. *"A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems"*. IEEE Transactions on Vehicular Technology (Volume: 65, Issue: 12, Dec. 2016), 1 November 2016
- [8] Herbert G. Tanner. *"Switched UAV-UGV Cooperation Scheme for Target Detection"*. IEEE International Conference on Robotics and Automation, 10-14 April 2007
- [9] Cai Luo, Andre Possani Espinosa, Danu Pranantha and Alessandro De Gloria. *"Multi-robot search and rescue team"*. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 1-5 November 2011
- [10] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi and Satoshi Tadokoro. *"Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots"*. Journal Field Robotics, v.29,(5),2012, Sept-Oct, pp. 832-841 Article, 2012
- [11] Jonathan Butzke, Andrew Dornbush, Maxim Likhachev. *"3-D Exploration with an Air-Ground Robotic System"*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 28 September-2 October 2015
- [12] Martin Saska, Tomas Krajnik, Libor Preucil. *"Cooperative μUAV-UGV autonomous indoor surveillance"*. The 9th International Multi-Conference on Systems, Signals and Devices, 2012
- [13] Heeseo Chae, Woo Yeol Kim and Ji Tae Hong. *"The Comparison of the Detecting Performance between the Ground and the Aerial Visual Analytics in the UGV-UAV Collaborative System"*. 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM) Banff, Alberta, Canada, July 12–15, 2016
- [14] El Houssein Chouaib Harik, Francois Guerin , Frederic Guinand,Jean-François

Brethe, Herve Pelvillain. "UAV-UGV Cooperation For Objects Transportation In An Industrial Area". 2015 IEEE International Conference on Industrial Technology (ICIT), March 17-19, 2015

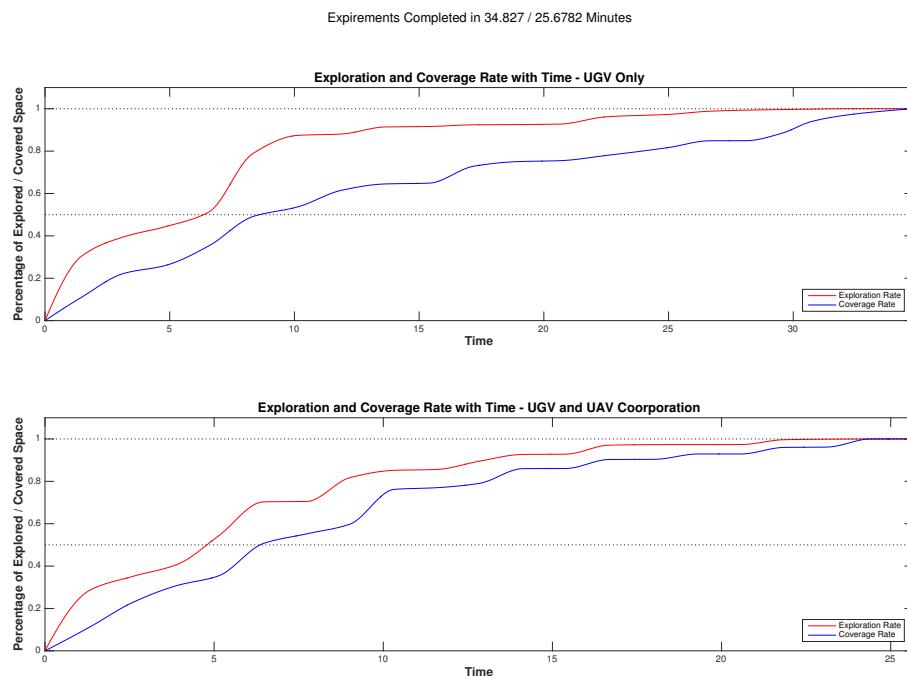
[15] El"od Pall, Koppány Mathe, Levente Tamas, Lucian Busoniu. "Railway Track Following with the AR.Drone Using Vanishing Point Detection". 2014 IEEE International Conference on Automation, Quality and Testing Robotics, May 22-24, 2014

[16] Emmanouil Tsardoulias. "Ενοποιημένο σύστημα ενός ή πολλαπλών ρομπότ για την πλοήγηση και εξερεύνηση σε αστικό περιβάλλον με σκοπό τον εντοπισμό ανθρώπινης ζωής (USAR)". April, 2013

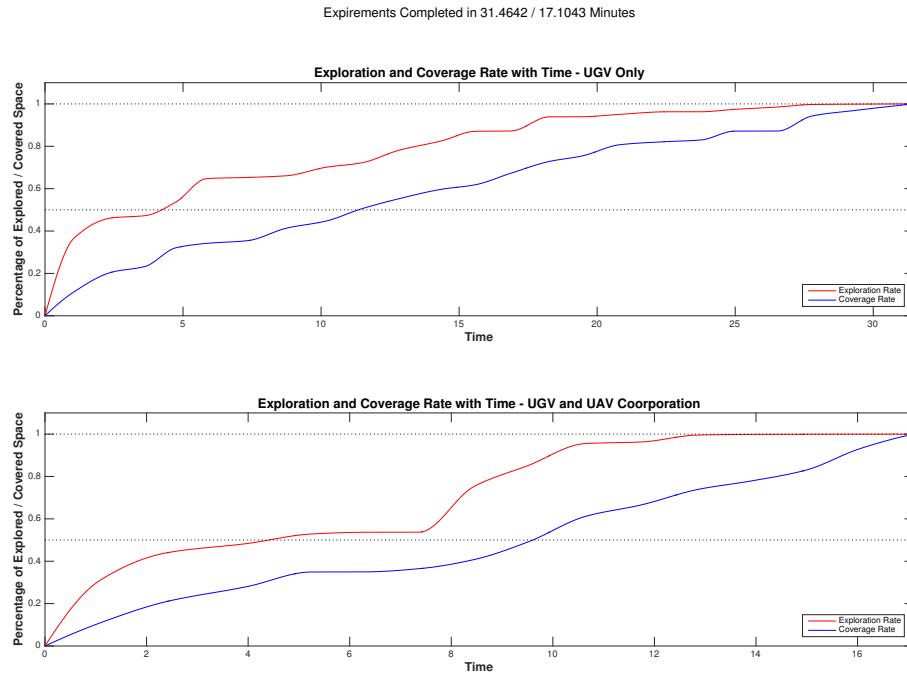
Παράρτημα Αποτελεσμάτων

Στο παρόν παράρτημα θα παρατεθούν κάποια επιπλέον συγχρητικά διαγράμματα που δημιουργήθηκαν μεταξύ της περίπτωσης εξερεύνησης μόνο από τον επίγειο ρομποτικό πράκτορα, της εξερεύνησης και από τους δύο πράκτορες με το Drone να εξερευνάει τα όρια του OGM και της εξερεύνησης από τους δύο πράκτορες με το Drone να εξερευνά τα όρια του OGM και να αξιολογού τους υποψήφιους στόχους.

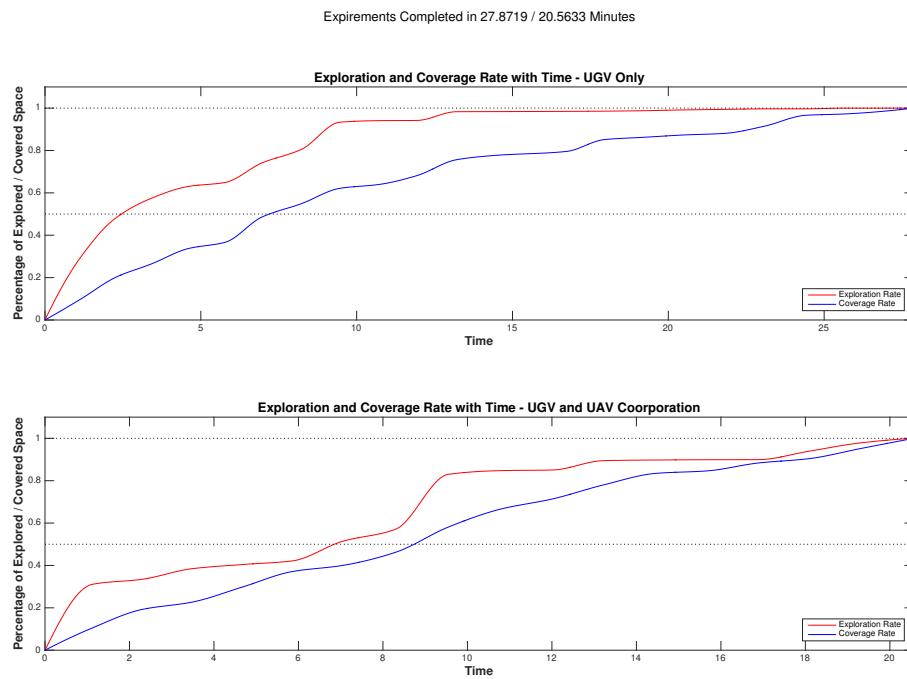
Συγκεκριμένα, παρακάτω παρουσιάζονται τα συγκρητικά διαγράμματα της εξερεύνησης μόνο από το UGV και της εξερεύνησης με UGV, UAV όπου το Drone εξερευνά τα όρια του OGM. Ο κάθετος άξονας αναπαριστά το ποσοστό εξερευνημένης/καλυμμένης περιοχής, ενώ ο οριζόντιος το χρόνο.



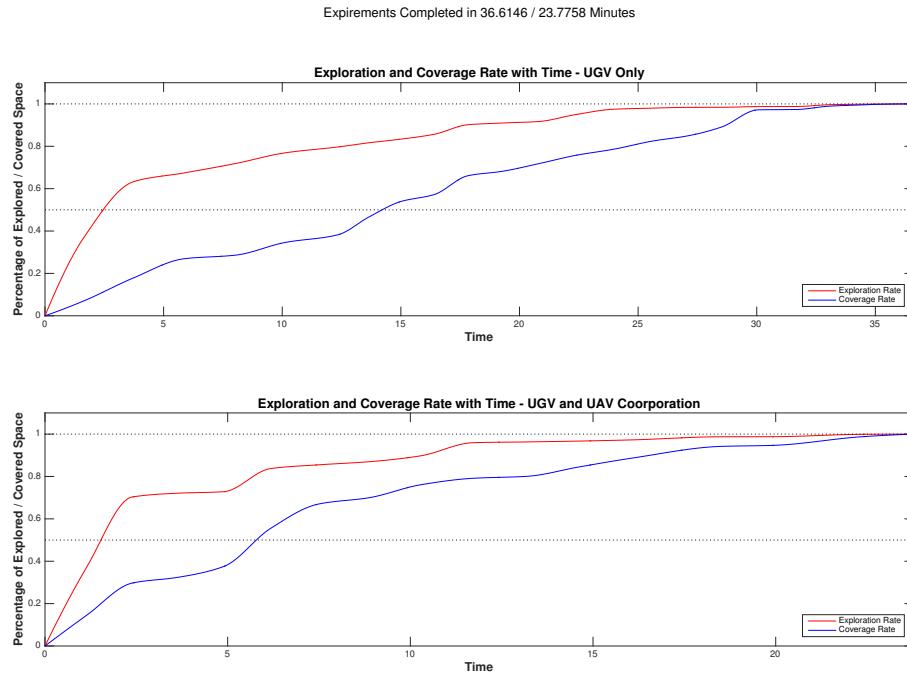
Σχήμα 44: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)



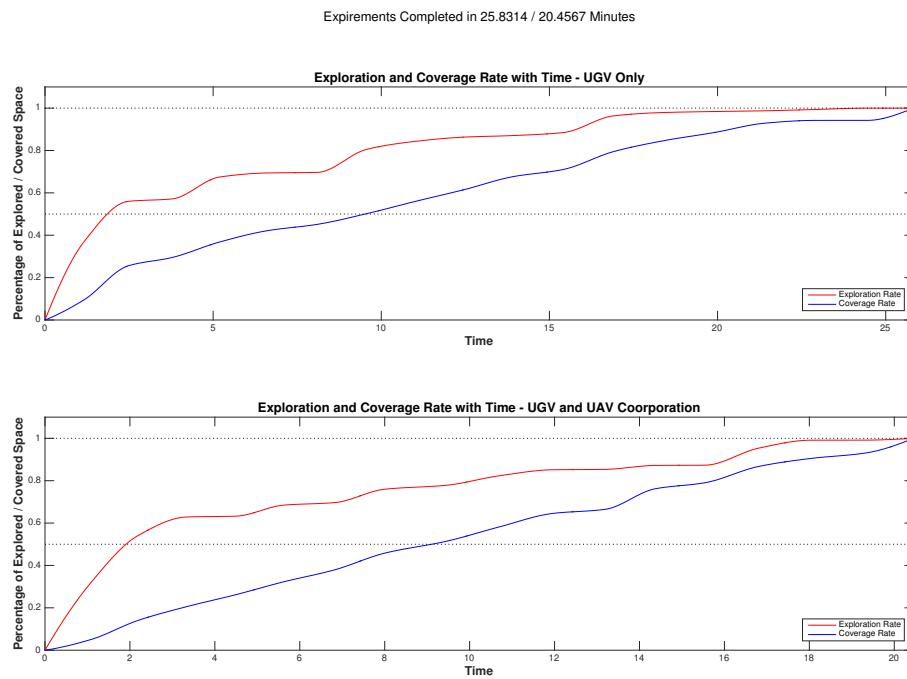
Σχήμα 45: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη (Full Coverage)



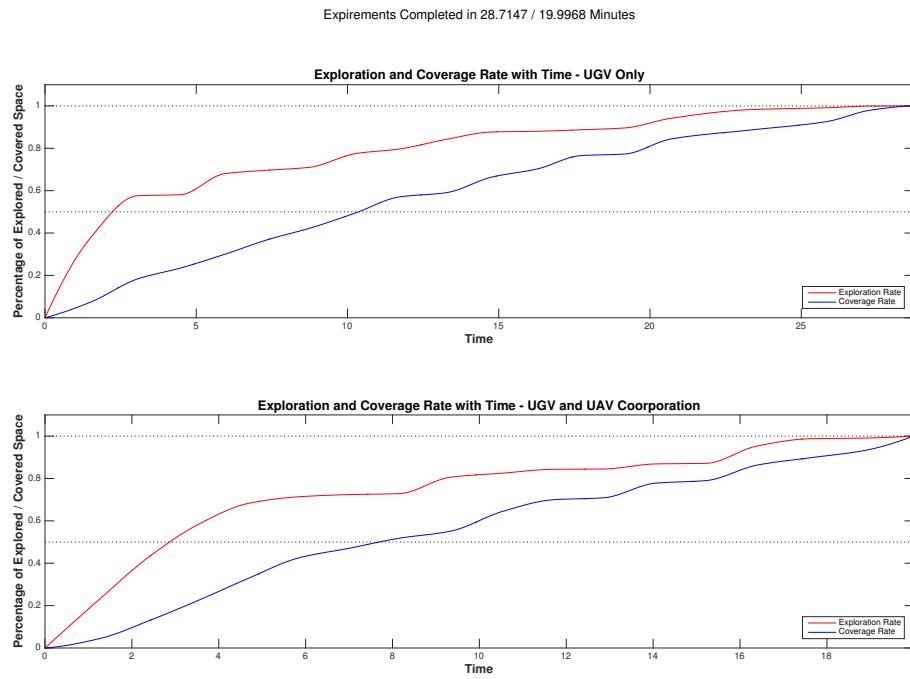
Σχήμα 46: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constraint)



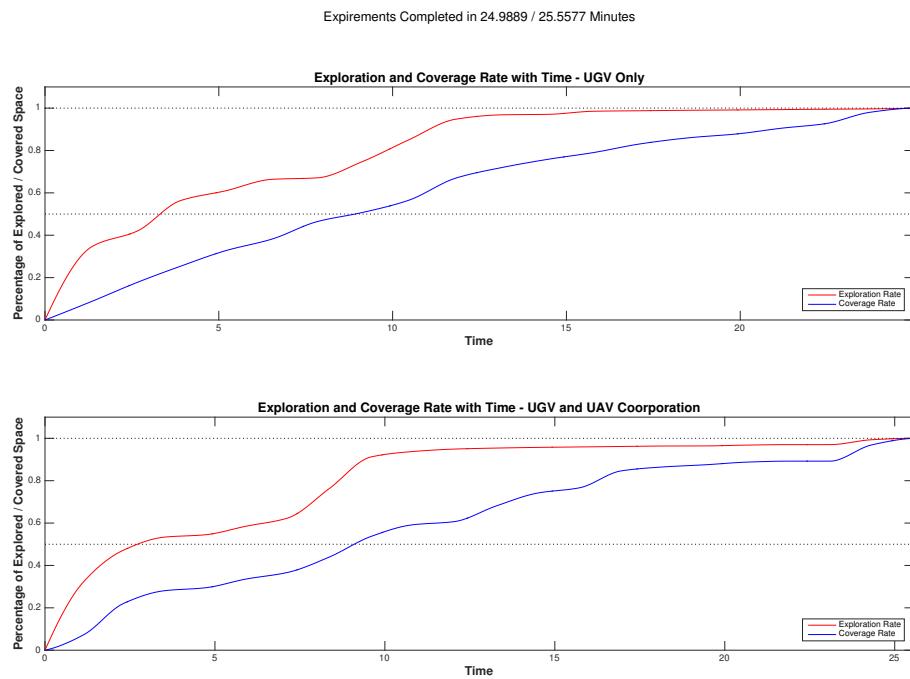
Σχήμα 47: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)



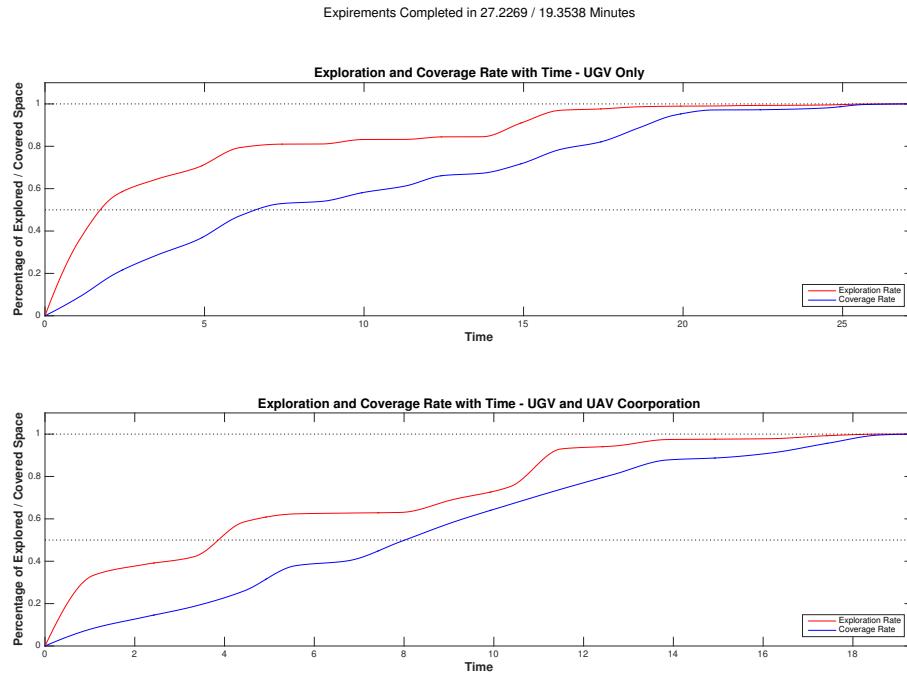
Σχήμα 48: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη (Full Coverage)



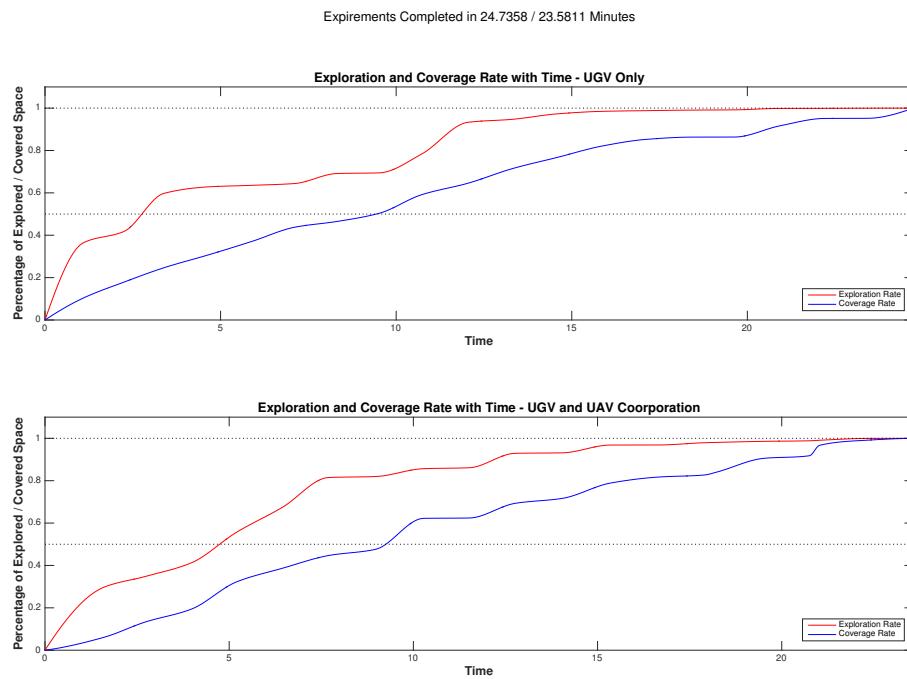
Σχήμα 49: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)



Σχήμα 50: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Εξερεύνηση (Full Exploration)

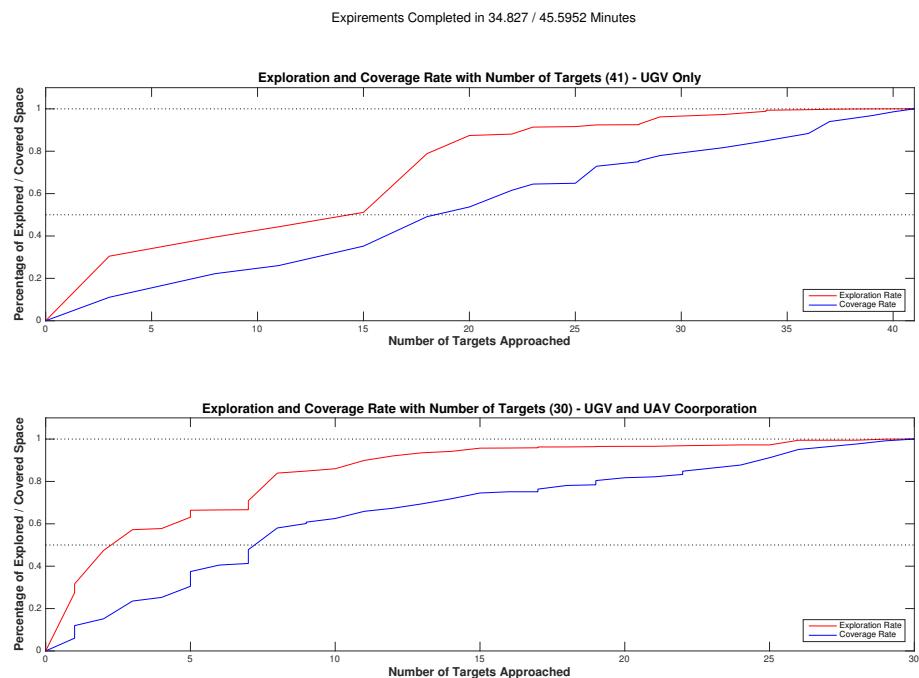


Σχήμα 51: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη (Full Coverage)



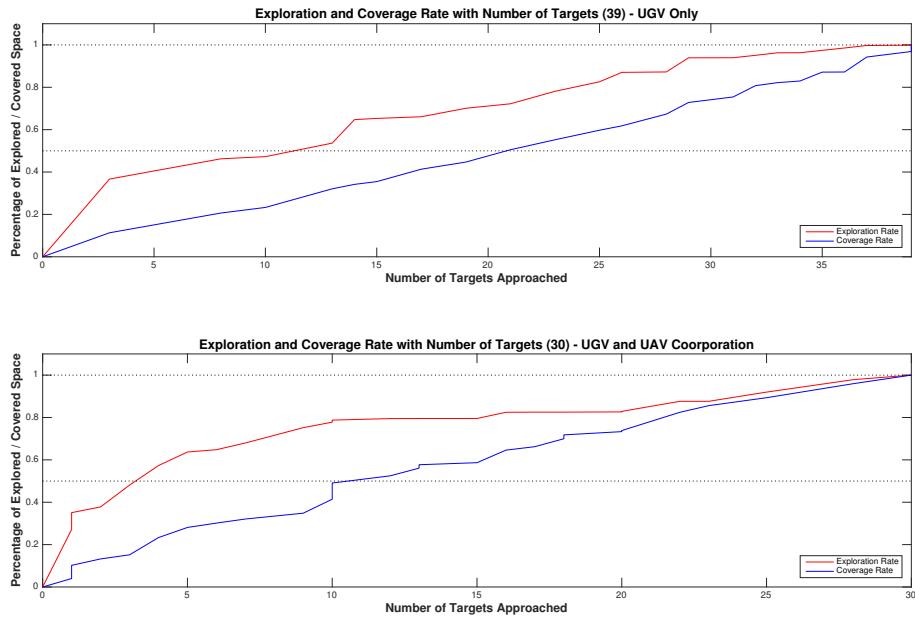
Σχήμα 52: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)

Τέλος παρουσιάζονται τα συγχρητικά διαγράμματα της εξερεύνησης μόνο από το UGV και της εξερεύνησης με UGV, UAV όπου το Drone εξερευνά τα όρια του OGM και αξιολογεί τους υποψήφιους στόχους. Ο κάθετος άξονας αναπαριστά το ποσοστό εξερευνημένης/καλυμμένης περιοχής, ενώ ο οριζόντιος το χρόνο.



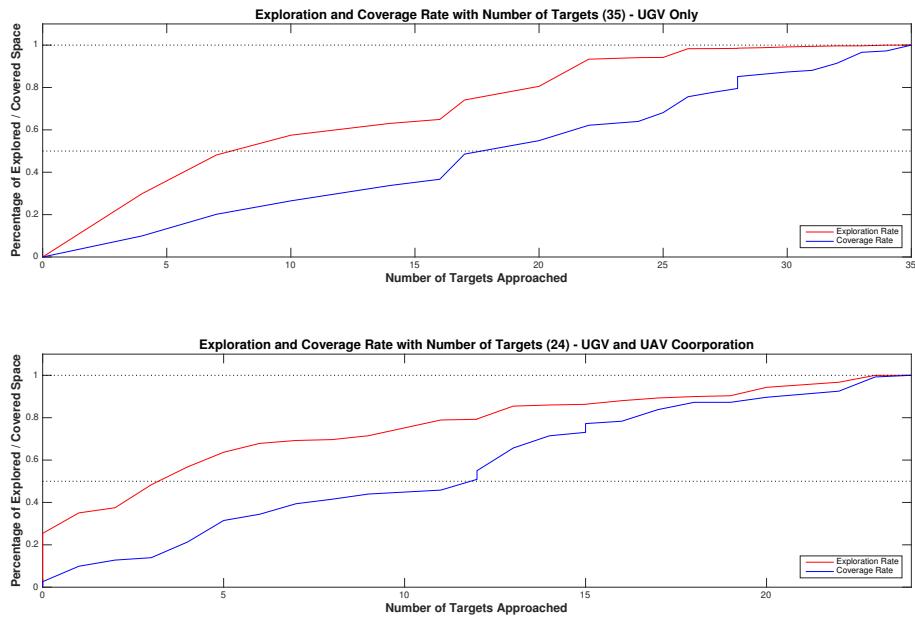
Σχήμα 53: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την εξερεύνηση (Full Exploration)

Experiments Completed in 31.4642 / 32.5186 Minutes



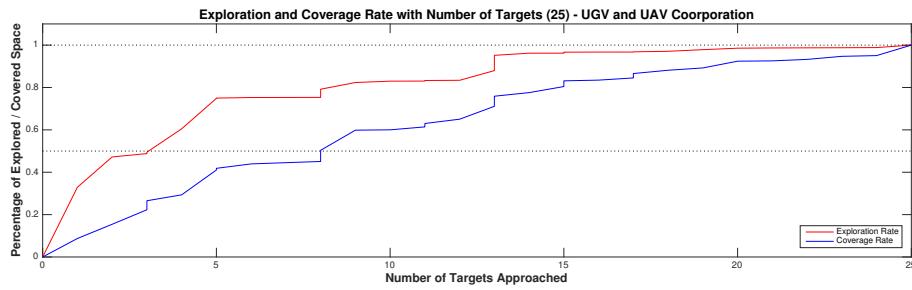
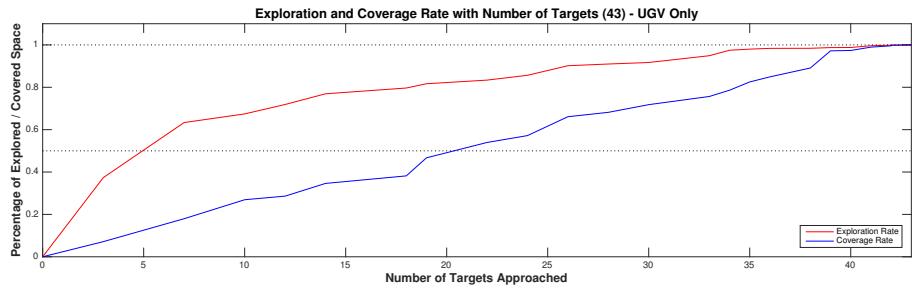
Σχήμα 54: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη(Full Coverage)

Experiments Completed in 27.8719 / 30.4809 Minutes



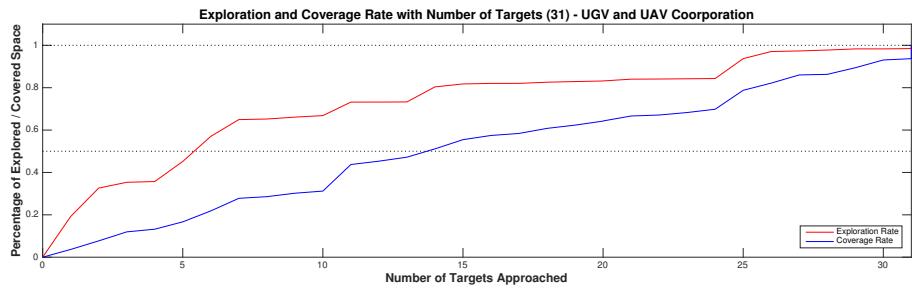
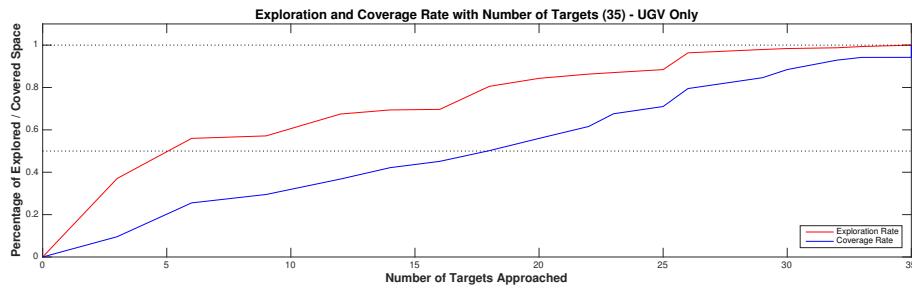
Σχήμα 55: Διαγράμματα Πειραμάτων στον Χώρο 1 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)

Experiments Completed in 36.6146 / 38.1715 Minutes



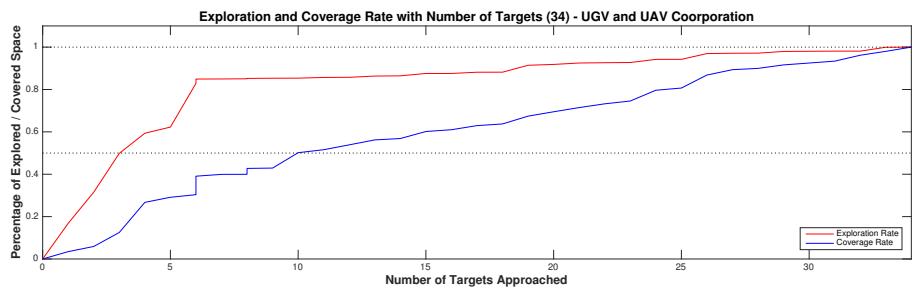
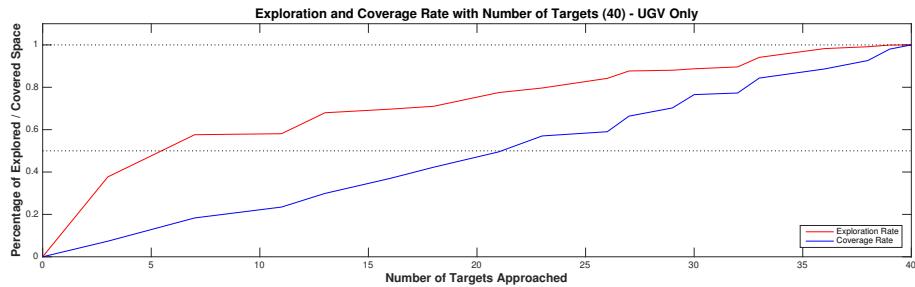
Σχήμα 56: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την εξερεύνηση (Full Exploration)

Experiments Completed in 25.8314 / 39.8642 Minutes



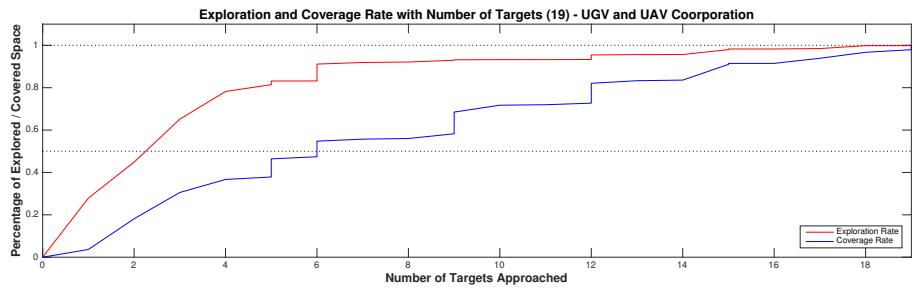
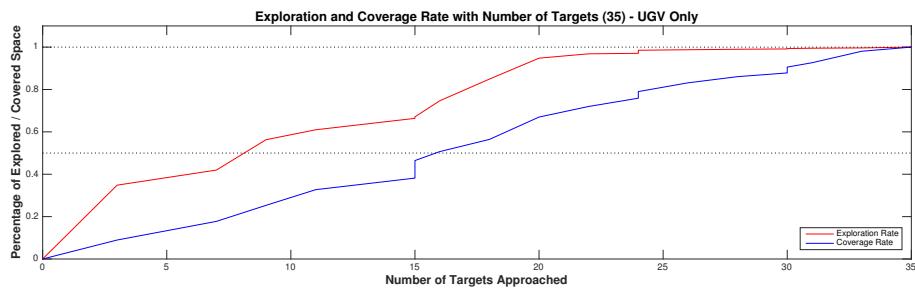
Σχήμα 57: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη (Full Coverage)

Experiments Completed in 28.7147 / 40.1292 Minutes



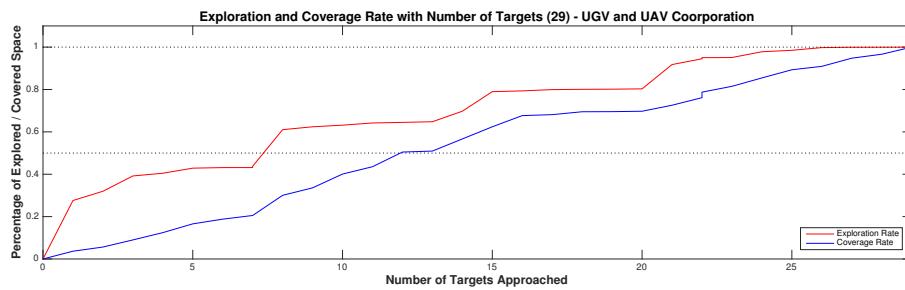
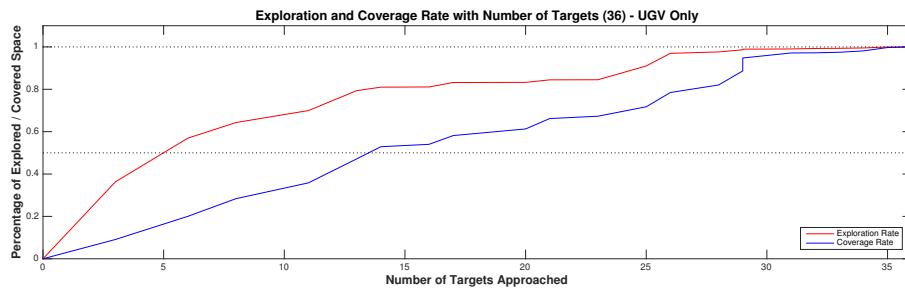
Σχήμα 58: Διαγράμματα Πειραμάτων στον Χώρο 2 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constrain)

Experiments Completed in 24.9889 / 34.0651 Minutes



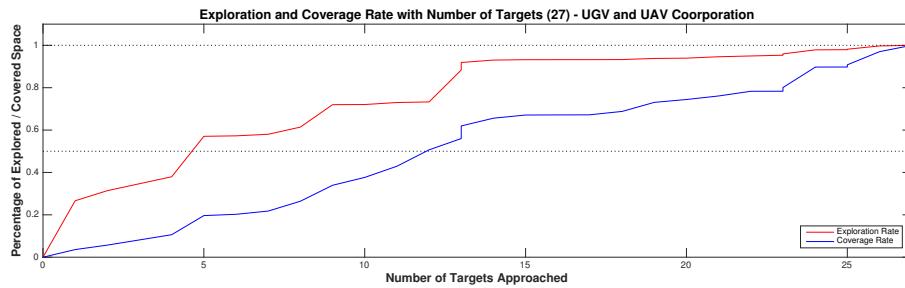
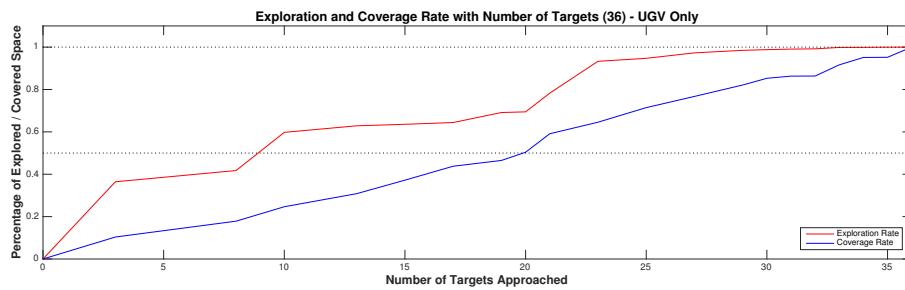
Σχήμα 59: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την εξερεύνηση (Full Exploration)

Experiments Completed in 27.2269 / 40.5795 Minutes



Σχήμα 60: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη (Full Coverage)

Experiments Completed in 24.7358 / 38.1619 Minutes



Σχήμα 61: Διαγράμματα Πειραμάτων στον Χώρο 3 με σκοπό την Πλήρη Κάλυψη σε περιορισμένο χρόνο (Full Coverage under Time Constraint)