



Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Βιομηχανικής Σχεδίασης και Παραγωγής

Διπλωματική εργασία

# Εκτίμηση πρόθεσης χειριστή με χρήση μηχανικής μάθησης

**Βαγγέλης Τσαγκουρλής**  
71445543

Επιβλέπων:

**Γρηγόρης Νικολάου**  
Λέκτορας Εφαρμογών

Αιγάλεω - Αθήνα, Οκτώβριος, 2022



Εγκρίθηκε από την εξεταστική επιτροπή τον Οκτώβριο 2022.

Νικολάου Γρηγόρης  
Λέκτορας Εφαρμογών

Βασιλειάδου Σουλτάνα  
Επίκουρη Καθηγήτρια

Δρόσος Χρήστος  
Ε.ΔΙ.Π

Τσαγκουρνήs Βαγγέλης  
Τμήμα Βιομηχανικής Σχεδίασης και Παραγωγής  
Πανεπιστήμιο Δυτικής Αττικής

Copyright © Τσαγκουρνήs Ευάγγελος, 2022  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Ο κάτωθι υπογεγραμμένος Τσαγκουρνήs Ευάγγελος του Εμμανουήλ, με αριθμό μητρώου 71445543 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι: «Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από εμένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Τσαγκουρνήs Ευάγγελος



*στους γονείς μου.*



# Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κ. Νικολάου Γρηγόρη για την επίβλεψη αυτής της διπλωματικής εργασίας, για τον καθοδηγητικό του ρόλο και για την εξαιρετική συνεργασία που είχαμε. Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου για την στήριξή τους στην προσπάθειά μου αυτά τα χρόνια και για την ηθική υποστήριξη που μου προσέφεραν.

01/09/2022

Βαγγέλης Τσαγκουρλής





# Περίληψη

Σκοπός της παρούσας διατριβής είναι η εκτίμηση της πρόθεσης του χειριστή ενός τηλεκατευθυνόμενου ρομπότ με τη βοήθεια μηχανικής μάθησης. Πιο συγκεκριμένα, αναπτύχθηκε ένα σύστημα δυναμικού εντοπισμού σημείων ενδιαφέροντος με χρήση μηχανικής όρασης και ενός αισθητήρα βάθους που επιτρέπει την εύρεση των τοποθεσιών τους στο χώρο. Παράλληλα, αναπτύχθηκε και εκπαιδεύτηκε το μοντέλο μηχανικής μάθησης MLOII (Machine Learning Operator Intent Inference) για την εκτίμηση της πρόθεσης του χειριστή. Το MLOII έχει την δυνατότητα να συλλέγει δεδομένα και να κάνει εκτιμήσεις για την πρόθεση του χειριστή σε πραγματικό χρόνο και αξιολογήθηκε πειραματικά σε εξομοίωση ενός τόπου καταστροφής. Τα αποτελέσματα δείχνουν πως το MLOII παρουσιάζει υψηλή αυτοπεποίθηση (confidence) σε σύγκριση με άλλες αντίστοιχες μεθόδους.

**Λέξεις Κλειδιά:** Ρομποτική, Μηχανική μάθηση, Μηχανική Όραση, Εμπιστευτικές Σημάνσεις, Τηλεκατευθυνόμενα Όχηματα, Εκτίμηση Πρόθεσης Χειριστή



# Abstract

The purpose of this thesis is to infer the intent of the operator of a remote-controlled robot using machine learning.

More specifically, a system for dynamic localization of points of interest using machine vision and a depth sensor was developed to find their locations in the environment. In addition, the machine learning model MLOII (Machine Learning Operator Intent Inference) was implemented and trained to estimate the operator's intent. MLOII has the ability to collect data and make estimates of operator intent in real time and was experimentally evaluated in a simulation of a disaster site. The results show that MLOII exhibits high confidence compared to other similar methods.

**Keywords:** Robotics, Machine Learning, Computer Vision, Fiducial Markers, Remote-controlled Vehicles, Operator Intent Inference



# Πίνακας περιεχομένων

Ευχαριστίες	iii
Περίληψη	v
Abstract	vii
Πίνακας περιεχομένων	x
Πίνακας σχημάτων	xii
Πίνακας πινάκων	xiii
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Πλαίσιο, σκοπός και στόχοι της διπλωματικής εργασίας . . . . .	3
1.2 Μεθοδολογία . . . . .	3
1.3 Περιορισμοί . . . . .	4
1.4 Ορισμοί . . . . .	4
1.5 Οργάνωση, κεφαλαίωση, διάρθρωση της εργασίας . . . . .	5
<b>2 Θεωρητικό μέρος – Βιβλιογραφική έρευνα – Σχετικές προσπάθειες</b>	<b>7</b>
2.1 Βιβλιογραφική έρευνα . . . . .	7
2.1.1 Μηχανική Μάθηση . . . . .	7
2.2 Σχετικές προσπάθειες – έρευνες . . . . .	15
2.2.1 Μηχανική Μάθηση . . . . .	15
2.2.2 Νευρωνικά Δίκτυα . . . . .	17
2.2.3 Πιθανοτικές μέθοδοι . . . . .	19
<b>3 Μεθοδολογία ή/και Σχέδιο εργασιών – Υλοποίηση – Εφαρμογή</b>	<b>23</b>
3.1 Robot Operating System - ROS . . . . .	23
3.1.1 Nodes . . . . .	24
3.1.2 Topics . . . . .	25
3.1.3 Messages . . . . .	25
3.1.4 Rosbag . . . . .	26
3.1.5 Packages . . . . .	27
3.1.6 Catkin . . . . .	27
3.1.7 Gazebo . . . . .	27
3.1.8 Husky . . . . .	28
3.2 Kinect Sensor . . . . .	28
3.2.1 Aruco Markers . . . . .	29
3.3 Περιγραφή των λειτουργιών που αναπτύχθηκαν ως nodes . . . . .	30
3.3.1 Το μοντέλο μηχανικής μάθησης MLOII . . . . .	37

<b>4</b>	<b>Αποτελέσματα – Ευρήματα / Επιτεύγματα</b>	<b>43</b>
4.1	Αναλυτική παρουσίαση αποτελεσμάτων . . . . .	43
4.1.1	Αποτελέσματα της εκπαίδευσης - Αξιολόγηση του MLOΠ . . . . .	43
4.2	Κυριότερα ευρήματα / αποτελέσματα . . . . .	50
<b>5</b>	<b>Συζήτηση – Συμπεράσματα – Μελλοντικές επεκτάσεις</b>	<b>51</b>
5.1	Ανακεφαλαίωση . . . . .	51
5.2	Συζήτηση / Συμπεράσματα . . . . .	51
5.3	Αξιοποίηση / Πρακτικές προεκτάσεις της έρευνας . . . . .	51
5.4	Μελλοντικές επεκτάσεις / Πρακτικές Προεκτάσεις της Έρευνας . . . . .	52
<b>6</b>	<b>Παράρτημα</b>	<b>53</b>
6.1	Πηγαίος κώδικας . . . . .	53
6.1.1	Points of Interest Detection Node . . . . .	53
6.1.2	Points of Interest Data Generation Node . . . . .	55
6.1.3	Data Persistence Node . . . . .	61
6.1.4	Operator Intent Inference Node . . . . .	65
	<b>Βιβλιογραφικές Αναφορές</b>	<b>69</b>
	<b>Πρόσθετη Βιβλιογραφία</b>	<b>71</b>

# Πίνακας σχημάτων

Εικόνα 1.	Εικόνες από αποστολή μέτρησης ραδιενέργειας και δειγματοληψία του αέρα από τον αντιδραστήρα της 2ης μονάδας του πυρηνικού σταθμού παραγωγής ενέργειας της Φουκουσίμα, 2011 [1] . . . . .	2
Εικόνα 2.	Παράδειγμα υπερπροσαρμογής. Η μαύρη γραμμή δείχνει πως το μοντέλο έχει μια καλή γενίκευση, ενώ η πράσινη υποδηλώνει υπερπροσαρμογή του μοντέλου στα δεδομένα. . . . .	8
Εικόνα 3.	Ένας πίνακας σύγκρισης για την ταξινόμηση αντικειμένου σε σκύλο ή μη σκύλο. . . . .	11
Εικόνα 4.	Απεικόνιση του Τεχνητού Πεδίου Δυναμικού. Το κόκκινο αυτοκίνητο είναι το αμάξι του ενδιαφέροντος. Περίπτωση (a): το μπλε αυτοκίνητο κινείται με μεγαλύτερη ταχύτητα από το κόκκινο. Περίπτωση (b): το μπλε αυτοκίνητο έχει μικρότερη ταχύτητα από το κόκκινο. [13] . . . . .	16
Εικόνα 5.	Εκτίμησης πρόθεσης ανθρώπου σε σενάριο συνεργασίας ανθρώπου-ρομπότ. [14] . . . . .	17
Εικόνα 6.	Η εξαγωγή των δεδομένων [15] . . . . .	17
Εικόνα 7.	Η σημαντικότητα των χαρακτηριστικών των αισθητήρων [15] . . . . .	18
Εικόνα 8.	Η σημαντικότητα των χαρακτηριστικών των αισθητήρων μαζί με της επικοινωνίας μεταξύ των οχημάτων [15] . . . . .	18
Εικόνα 9.	Σενάριο εκτίμησης της ανθρώπινης πρόθεσης στο επιτραπέζιο tennis. [17]	20
Εικόνα 10.	Μικτή αυτονομία (Shared Autonomy). [8] . . . . .	21
Εικόνα 11.	Οι σχέσεις μεταξύ των ROS Nodes, Topics και Messages . . . . .	25
Εικόνα 12.	Ενεργά topics του ROS και ο τύπος των messages μέσω του εργαλείου rqt που παρέχει το ROS . . . . .	26
Εικόνα 13.	Το περιβάλλον εξομοίωσης του Gazebo . . . . .	28
Εικόνα 14.	Η ρομποτική πλατφόρμα Husky [22] . . . . .	28
Εικόνα 15.	Ο αισθητήρας Kinect V2 της Microsoft . . . . .	29
Εικόνα 16.	Παραδείγματα Aruco markers από διάφορα dictionaries . . . . .	29
Εικόνα 17.	Η δομή των υπολειτουργιών για τον εντοπισμό των σημείων ενδιαφέροντος και της εκτίμησης της πρόθεσης του χειριστή . . . . .	31
Εικόνα 18.	Οπτικοποίηση του εντοπισμού του Aruco Marker με ID = 0 από το POIDN μέσω της μεθόδου drawDetectedMarkers . . . . .	32
Εικόνα 19.	Μια εικόνα βάθους της ίδιας σκηνής με την Εικόνα 5 . . . . .	33
Εικόνα 20.	Η κάτοψη του πρώτου περιβάλλοντος συλλογής δεδομένων. Το ρομπότ είναι στο κέντρο των τριών σημείων ενδιαφέροντος, όπου το κάθε ένα φέρει ένα marker. Τα σημεία ενδιαφέροντος είναι ο άνθρωπος, το ασθενοφόρο και το πυροσβεστικό όχημα. . . . .	37
Εικόνα 21.	Πλοήγηση του ρομπότ στο χώρο με τη χρήση τηλεχειριστηρίου. . . . .	38

Εικόνα 22.	(a) Αναγνώριση του marker με id: 15 στον άνθρωπο (b) Αναγνώριση του marker με id: 0 στο ασθενοφόρο (c) Αναγνώριση του marker με id: 8 στο πυροσβεστικό όχημα . . . . .	38
Εικόνα 23.	Η κάτοψη του δεύτερου περιβάλλοντος συλλογής δεδομένων. Το ρομπότ είναι τοποθετημένο απέναντι στο ασθενοφόρο το οποίο φέρει τα 3 markers. . . . .	39
Εικόνα 24.	Το δεύτερο περιβάλλον από την οπτική γωνία της κάμερας του ρομπότ. Λόγω του ότι και τα 3 markers είναι μέσα στην ίδια εικόνα, δεν χρειάζεται η περιήγηση του ρομπότ στο χώρο. . . . .	39
Εικόνα 25.	Η μορφή των δεδομένων μετά την επεξεργασία. . . . .	40
Εικόνα 26.	Ο πίνακας σύγκρισης του MLOII. . . . .	43
Εικόνα 27.	Ο κανονικοποιημένος πίνακας σύγκρισης του MLOII. . . . .	44
Εικόνα 28.	Η σημασία των δεδομένων εισόδου για το MLOII. . . . .	46
Εικόνα 29.	Πρώτο σενάριο. . . . .	47
Εικόνα 30.	Δεύτερο σενάριο. . . . .	47
Εικόνα 31.	Τρίτο σενάριο. . . . .	48
Εικόνα 32.	Ορθότητα του BOIR και του MLOII από τα πειράματα. . . . .	49
Εικόνα 33.	Cross-Entropy Loss του BOIR και του MLOII από τα πειράματα. . . . .	49



# Πίνακας πινάκων

Πίνακας 1. Τα είδη των πεδίων που είναι διαθέσιμα για τα messages . . . . .	26
---	----



# Κεφάλαιο 1

## Εισαγωγή

Τις τελευταίες δεκαετίες, η ιλιγγιώδης ταχύτητα με την οποία εξελίσσεται η τεχνολογία έχει οδηγήσει σε αυτόνομα συστήματα τα οποία είναι ικανά να εκτελέσουν πολύπλοκες λειτουργίες που θεωρούνταν δυνατές μόνο από τον άνθρωπο. Για παράδειγμα, η μηχανική όραση σε συνδυασμό με τη μηχανική μάθηση έχουν την δυνατότητα να αντικαταστήσουν έναν χειριστή κάμερας ασφαλείας, ένας ρόλος που παραδοσιακά απαιτούσε έναν άνθρωπο λόγω της πολυπλοκότητας του. Με την πάροδο του χρόνου, όλο και περισσότερες δουλειές μπορούν να αντικατασταθούν από συστήματα τα οποία έχουν τη δυνατότητα να συμπεριφερθούν και να ομοιάσουν στον άνθρωπο.

Στα πλαίσια των αποστολών έρευνας και διάσωσης (Search and Rescue, SAR), η ελαχιστοποίηση του κινδύνου για τα συνεργεία διάσωσης είναι ένα πολύ σημαντικό θέμα, καθώς αυτοί οι άνθρωποι συχνά χρειάζεται να εισέλθουν σε επικίνδυνο περιβάλλον και υπό δύσκολες συνθήκες (υψηλές θερμοκρασίες, σκοτάδι, ραδιενέργεια, κίνδυνος ανάφλεξης ή έκρηξης) [1] [2]. Λόγω των παραπάνω, χρησιμοποιούνται πολλές φορές μη-επανδρωμένα οχήματα, εναέρια (Unmanned Air Vehicles, UAVs) ή και επίγεια (Unmanned Ground Vehicles, UGVs) για την εξερεύνηση εσωτερικών αλλά και εξωτερικών χώρων μετά από καταστροφές που ευθύνονται είτε σε ανθρώπινο παράγοντα είτε σε φυσικά αίτια.

Τα μη-επανδρωμένα οχήματα χρειάζονται ανθρώπινους χειριστές οι οποίοι είναι υπεύθυνοι για τον έλεγχό τους και για την διεκπεραίωση της αποστολής. Στις περισσότερες περιπτώσεις αποστολών έρευνας και διάσωσης, ένα ρομπότ αναθέτεται σε δύο ανθρώπους [3]. Ο χειριστής του ρομπότ έχει ως ευθύνη την πλοήγηση του οχήματος στον χώρο καθώς και την αποφυγή εμποδίων. Οι συγκεκριμένες λειτουργίες είναι ήδη αρκετά απαιτητικές για τον χειριστή, και καταλαμβάνουν ένα πολύ μεγάλο μέρος της προσοχής του. Ως αποτέλεσμα, ο δεύτερος άνθρωπος έχει την αρμοδιότητα να βοηθήσει τον χειριστή του ρομπότ με το να φέρει σε πέρας την αποστολή που μπορεί να είναι η οπτική επιθεώρηση του χώρου ή ο εντοπισμός θυμάτων.

Φαινόμενα που επηρεάζουν αρνητικά την επίδοση των χειριστών είναι πολύ συχνά η έλλειψη ύπνου και η πίεση που δέχονται λόγω της μεγάλης διάρκειας μιας αποστολής. Τέτοιου είδους φαινόμενα επηρεάζουν σε πολύ μεγάλο βαθμό την αποτελεσματικότητά τους. Εάν δεν υπάρχουν αρκετοί χειριστές ώστε να αναλαμβάνουν με βάρδιες τη χρήση των οχημάτων ώστε να μπορούν οι υπόλοιποι να ξεκουράζονται, η αποτελεσματικότητα φθίνει σημαντικά με την πάροδο του χρόνου.

Η επίδοση των χειριστών επηρεάζεται σε μεγάλο βαθμό και από τις συνθήκες που επικρατούν στο περιβάλλον στο οποίο εκτελείται μια αποστολή έρευνας και διάσωσης. Τα ρομπότ συχνά χρειάζεται να λειτουργήσουν σε χαοτικά περιβάλλοντα [2], ενώ οι χειριστές συχνά δεν έχουν επαρκή επίγνωση της κατάστασης (Situational Awareness) λόγω του ότι καλούνται να αναγνωρίσουν το περιβάλλον τους μέσα από μια κάμερα η οποία είναι τοποθετημένη επάνω στα ρομπότ.

Ακόμα, οι συνθήκες του περιβάλλοντος πολλές φορές δημιουργούν προβλήματα στην επικοινωνία μεταξύ του οχήματος και του χειριστή. Εσωτερικοί χώροι οι οποίοι είναι γεμάτοι με συντρίμια ή περιβάλλονται από χοντρούς τοίχους μπορούν να παρεμποδίσουν ή να διακόψουν εντελώς τον χειρισμό του οχήματος από απόσταση. Σε τέτοιες περιπτώσεις, ένα ρομπότ χάνει το σήμα



Εικόνα 1. Εικόνες από αποστολή μέτρησης ραδιενέργειας και δειγματοληψία του αέρα από τον αντιδραστήρα της 2ης μονάδας του πυρηνικού σταθμού παραγωγής ενέργειας της Φουκουσίμα, 2011 [1]

καταλήγει να είναι εντελώς αβοήθητο, όπου και εγκαταλείπεται [2]. Για αυτούς τους λόγους, οι αυτόνομες λειτουργίες είναι υψηλής σημασίας για αυτού του είδους τα ρομπότ.

Η πλήρης αυτονομία για τα ρομπότ αυτά δεν είναι ακόμα έτοιμη σε πολλές από τις περιπτώσεις [2], ιδιαίτερα στα μη-επανδρωμένα οχήματα εδάφους. Ένας λόγος αυτού είναι η έλλειψη επίγνωσης της κατάστασης που παρατηρείται στα ρομπότ [4]. Ακόμα και εάν έχει αναπτυχθεί ένα πλήρες αυτόνομο σύστημα, όταν δεν εμπλέκεται μέσα σε αυτό με κάποιο τρόπο ο άνθρωπος, οι ενδιαφερόμενοι για αυτό τείνουν να μην το εμπιστεύονται [3] [5]. Αυτό οφείλεται στο ότι προτιμούνται οι λειτουργικότητες οι οποίες έχουν περάσει από μια πληθώρα δοκιμών, καθώς και συστήματα τα οποία δεν είναι υπερευαίσθητα ή περίπλοκα, ιδιαίτερα όταν υπάρχει κίνδυνος αποτυχίας που πιθανόν θα στοιχίσει σε ανθρώπινες ζωές.

Ωστόσο σε συγκεκριμένες περιπτώσεις, η μερική αυτονομία μπορεί να αξιοποιηθεί με διαμοιρασμένο έλεγχο (Shared Control), επιτρέποντας στον χειριστή να επικεντρωθεί σε πολλές λειτουργίες που απαιτούν την προσοχή του με το ίδιο το ρομπότ να βοηθά ενεργά σε πολλές από αυτές. Αυτό επιτυγχάνεται με τη χρήση ελεγκτών που έχουν αναπτυχθεί για να αποφασίζουν δυναμικά για το επίπεδο της αυτονομίας (Level of Autonomy, LOA) [6] [7]. Ένας τέτοιος ελεγκτής λαμβάνει αποφάσεις για το ποιές λειτουργίες θα εκτελούνται από τον άνθρωπο και ποιές από το ίδιο το ρομπότ, συγκρίνοντας την αποτελεσματικότητά τους. Αυτό επιτρέπει τη χρήση τόσο του ανθρώπου όσο και της Τεχνητής Νοημοσύνης (Artificial Intelligence, AI) ούτως ώστε να αντιμετωπιστούν τα παραπάνω προβλήματα με τον βέλτιστο τρόπο συνδυάζοντας τις ικανότητές τους. Επάνω σε αυτή τη λογική αναπτύσσεται ο έλεγχος Ανθρώπινης Πρωτοβουλίας (Human-Initiative, HI) όπου ο άνθρωπος λαμβάνει αποφάσεις και εκτελεί αλλαγές στο LOA, ο έλεγχος Ρομποτικής Πρωτοβουλίας (Robot-Initiative, RI) όπου το ρομπότ επιλέγει και εκτελεί αλλαγές στο LOA και ο έλεγχος Μικτής Πρωτοβουλίας (Mixed-Initiative, MI) όπου μπορούν τόσο ο άνθρωπος όσο και το ρομπότ να εκτελούν αλλαγές στο LOA. Από τη δουλειά του Chiou *et al* [7], φαίνεται πως ο έλεγ-

χος Mixed-Initiative είναι πιο αποτελεσματικός σε σύγκριση με τους υπόλοιπους, παρόλα αυτά παρατηρείται πως έχει πολύ πιο συχνές εναλλαγές στο LOA. Αυτό οφείλεται στο γεγονός ότι σε αρκετές περιπτώσεις υπάρχει διαμάχη και ανταγωνισμός μεταξύ του ανθρώπου και του ρομπότ για τον έλεγχο.

Παρά το γεγονός ότι ο διαμοιρασμένος έλεγχος φαίνεται να ξεπερνάει σε αποτελεσματικότητα τον άνθρωπο και το ρομπότ από μόνα τους [8], υπάρχουν κάποια ζητήματα που δυσχεραίνουν την εφαρμογή του [7]. Όπως αναφέρθηκε νωρίτερα, η διαμάχη στον έλεγχο MI μεταξύ του ανθρώπου και του ρομπότ για τον έλεγχο είναι ένα από τα σημαντικότερα ζητήματα, το οποίο οφείλεται στο ότι τόσο ο άνθρωπος όσο και το ρομπότ δεν έχουν επίγνωση για την πρόθεση του άλλου μέλους. Για να καταφέρει το ρομπότ να βοηθήσει τον άνθρωπο στον στόχο του, οφείλει να μπορέσει να καταλάβει/συμπεράνει την πρόθεσή του. Θεωρητικά, αυτό θα μπορούσε να επιτευχθεί με τον άνθρωπο να επικοινωνήσει ρητά την πρόθεσή του. Παράδειγμα αυτού μπορεί να είναι ο προφορικός λόγος ή η χρήση κάποιας διεπαφής (οθόνη αφής με προκαθορισμένες επιλογές) [9]. Αυτό όμως προσθέτει στον άνθρωπο παραπάνω φόρτο εργασίας, καθώς τώρα οφείλει μαζί με όλες τις άλλες αρμοδιότητες που του έχουν ανατεθεί, να πρέπει να κρατάει το ρομπότ ενήμερο για κάθε αλλαγή της πρόθεσής του, κάτι το οποίο θα κουράσει πιο γρήγορα τον ίδιο.

Οι άνθρωποι είναι πολύ καλοί στο να συμπεραίνουν την πρόθεση άλλων ανθρώπων με τη χρήση μη λεκτικών ενδείξεων [10], απλά και μόνο παρακολουθώντας την κίνηση του σώματός του ή την κατεύθυνση την οποία κοιτάνε. Αυτού του είδους το συμπέρασμα έχει τη δυνατότητα να δώσει στο ρομπότ την πολύτιμη πληροφορία για το σκοπό του ανθρώπινου χειριστή χωρίς να επιβαρύνει τον ίδιο με παραπάνω αρμοδιότητες.

## 1.1 Πλαίσιο, σκοπός και στόχοι της διπλωματικής εργασίας

Σε αυτή την εργασία παρουσιάζεται μια μέθοδος δυναμικής εύρεσης σημείων ενδιαφέροντος από μια ρομποτική πλατφόρμα την οποία ελέγχει ένας χειριστής μέσω ενός χειριστηρίου. Η εύρεση των σημείων ενδιαφέροντος γίνεται μέσω κάμερας με ενσωματωμένο αισθητήρα βάθους και μέσω αυτών εξάγονται χρήσιμες πληροφορίες όπως η τοποθεσία των σημείων ενδιαφέροντος στον χώρο. Με την τοποθεσία των σημείων ενδιαφέροντος και πληροφορίες από το ρομπότ όπως η ταχύτητά του και η τοποθεσία του, εξάγονται ως πληροφορίες η ταχύτητα, η γωνία και η απόσταση του σε σχέση με τον κάθε σημείο ενδιαφέροντος που έχει ανιχνευτεί. Στη συνέχεια οι πληροφορίες αυτές χρησιμοποιούνται ως δεδομένα εισόδου στο MLOII για την εκτίμηση του σημείου ενδιαφέροντος που προσπαθεί να προσεγγίσει ο χειριστής του ρομπότ. Ο σκοπός του συστήματος αυτού είναι η εκτίμηση της πρόθεσης του χειριστή της ρομποτικής πλατφόρμας με χρήση πληροφοριών που μπορούν να εξαχθούν από το ίδιο το ρομπότ, χωρίς κάποιο σύστημα παρακολούθησης του χειριστή και της βιολογικής δραστηριότητάς του όπως π.χ την παρακολούθηση του βλέμματος (Gaze Tracking).

Η σωστή αναγνώριση της πρόθεσης του χειριστή μπορεί να χρησιμοποιηθεί από άλλες λειτουργίες του ρομπότ ώστε το σύστημα άνθρωπος-ρομπότ να μπορεί να λειτουργήσει πιο αποτελεσματικά, ελαχιστοποιώντας τη διαμάχη για τον έλεγχο στις περιπτώσεις συστημάτων μεταβλητής αυτονομίας Mixed-Initiative.

## 1.2 Μεθοδολογία

Σε αυτή την εργασία έγινε εκτενής βιβλιογραφική έρευνα ως προς τις μεθόδους εκτίμησης της πρόθεσης των ανθρώπων από διάφορες εφαρμογές. Στη συνέχεια αναπτύχθηκε η λογική για τη δυναμική εύρεση των στόχων μέσω της χρήσης της μηχανικής όρασης, καθώς και η εξαγωγή της θέσης τους με τη χρήση του αισθητήρα βάθους. Δημιουργήθηκε και εκπαιδεύτηκε το MLOII με δεδομένα που εξάχθηκαν από τους αισθητήρες του ρομπότ, συγκεκριμένα την ταχύτητα, τη

γωνία και την απόσταση του ρομπότ σε σχέση με το κάθε σημείο ενδιαφέροντος που έχει ανιχνευτεί. Το MLOP συγκρίθηκε με μια σύγχρονη μέθοδο από τη βιβλιογραφία και παρουσιάζονται τα αποτελέσματα από τα πειράματα.

### 1.3 Περιορισμοί

Η εργασία αυτή περιορίζεται στο περιβάλλον εξομοίωσης "Gazebo", με τη χρήση πιστής εξομοίωσης του ρομποτικού οχήματος "Husky" το οποίο έχει εξοπλιστεί με έναν αισθητήρα "Kinect". Παρότι το "Gazebo" εξομοιώνει τους αισθητήρες και τη συμπεριφορά του ρομποτικού οχήματος με ακρίβεια, παραμένει ένα περιβάλλον εξομοίωσης και το πείραμα δεν έχει δοκιμαστεί σε πραγματικό ρομπότ.

Η αναγνώριση των στόχων γίνεται με τη χρήση σημάτων που λειτουργούν ως σημεία αναφοράς (fiducial markers), γνωστά ως "Aruco Tags", τα οποία πληρούν το ρόλο των σημείων ενδιαφέροντος. Προκειμένου ο αισθητήρας βάθους να μπορέσει να λάβει τη μέτρηση απόστασης για τα σημεία ενδιαφέροντος, θα πρέπει αυτά να εισέλθουν σε απόσταση μικρότερη ή ίση με τα πέντε (5) μέτρα από αυτόν. Ταυτόχρονα, πρέπει να βρεθούν εντός του οπτικού πεδίου της κάμερας τουλάχιστον μια φορά, ούτως ώστε να γίνει η οπτική αναγνώριση. Έπειτα, υπολογίζεται και εξάγεται η τοποθεσία των σημείων αυτών στο δυσδιάστατο χώρο με βάση την τοποθεσία που καταγράφει το ρομπότ για το ίδιο. Αυτό προϋποθέτει ότι το ρομπότ έχει επίγνωση της ακριβούς του θέσης στο χώρο.

### 1.4 Ορισμοί

Εδώ αναφέρονται οι ορισμοί των σημαντικότερων όρων της εργασίας.

- **Αλληλεπίδραση Ανθρώπου-Ρομπότ (Human-Robot Interaction):** Η Αλληλεπίδραση Ανθρώπου-Ρομπότ είναι η μελέτη των αλληλεπιδράσεων μεταξύ ανθρώπων και ρομπότ. Συχνά αναφέρεται ως HRI από τους ερευνητές.
- **Μη Επανδρωμένα Οχήματα (Unmanned Vehicles):** Πρόκειται για οχήματα τα οποία μπορούν να λειτουργήσουν χωρίς ανθρώπινη παρουσία επί του ίδιου.
- **Μικτή Πρωτοβουλία (Mixed-Initiative):** Μια αλληλεπίδραση υπολογιστή-ανθρώπου στην οποία είτε ο υπολογιστής είτε ο άνθρωπος μπορεί να αναλάβει πρωτοβουλία και να αποφασίσει τι θα κάνει στη συνέχεια.
- **Κοινόχρηστος Έλεγχος (Shared Control):** Κοινόχρηστος έλεγχος ή διαμοιρασμένος έλεγχος είναι ένα επίπεδο αυτονομίας όπου τόσο ο άνθρωπος όσο και το ρομπότ συμβάλλουν στην πλοήγησή του ρομπότ.
- **Τεχνητή Νοημοσύνη (Artificial Intelligence):** Η τεχνητή νοημοσύνη είναι η νοημοσύνη που επιδεικνύουν οι μηχανές, σε αντίθεση με τη φυσική νοημοσύνη που επιδεικνύουν τα ζώα και οι άνθρωποι. Ο όρος "τεχνητή νοημοσύνη" είχε χρησιμοποιηθεί προηγουμένως για να περιγράψει μηχανές που μιμούνται και επιδεικνύουν "ανθρώπινες" γνωστικές δεξιότητες που συνδέονται με το ανθρώπινο μυαλό, όπως η "μάθηση" και η "επίλυση προβλημάτων". Ο ορισμός αυτός έχει έκτοτε απορριφθεί από σημαντικούς ερευνητές της τεχνητής νοημοσύνης, οι οποίοι περιγράφουν πλέον την τεχνητή νοημοσύνη με όρους ορθολογισμού και ορθολογικής δράσης, γεγονός που δεν περιορίζει τον τρόπο με τον οποίο μπορεί να διατυπωθεί.

- **Μηχανική Μάθηση (Machine Learning):** Η Μηχανική Μάθηση είναι μια συλλογή από υπολογιστικές μεθόδους μέσω των οποίων ένα μοντέλο εκπαιδεύεται σε ένα σύνολο δεδομένων, γνωστά ως δεδομένα εκπαίδευσης, προκειμένου να κάνουν προβλέψεις ή να λαμβάνουν αποφάσεις χωρίς να είναι ρητά προγραμματισμένο από τον άνθρωπο. Η μηχανική μάθηση είναι μια προσέγγιση της Τεχνητής Νοημοσύνης.
- **Έρευνα και Διάσωση (Search and Rescue, SAR):** Πρόκειται για αποστολές που διεξάγονται σε περιπτώσεις φυσικών ή μη καταστροφών, προκειμένου να εντοπιστούν θύματα και στη συνέχεια να διασωθούν.
- **Σημεία Ενδιαφέροντος (Points of Interest):** Ως σημεία ενδιαφέροντος εννοούμε σημεία του χώρου τα οποία έχουν αξία για τον χειριστή του ρομπότ και τον ενδιαφέρουν. Ο χειριστής του ρομπότ στην παρούσα εργασία επιχειρεί να τα προσεγγίσει.
- **Εμπιστευτικές Σημάνσεις (Fiducial Markers):** Οι εμπιστευτικές σημάνσεις είναι αντικείμενα που τοποθετούνται στο οπτικό πεδίο ενός συστήματος απεικόνισης και εμφανίζονται στην παραγόμενη εικόνα, για χρήση ως σημεία αναφοράς ή μέτρησης. Στην εργασία γίνεται χρήση των Aruco Tags και χρησιμοποιούνται ως σημεία ενδιαφέροντος.
- **Σχεδιαστής Διαδρομής (Path Planner):** Οι σχεδιαστές διαδρομών είναι αλγόριθμοι που βρίσκουν τη συντομότερη διαδρομή μεταξύ δύο σημείων του χώρου. Είναι μια πιο πρακτική παραλλαγή της επίλυσης λαβυρίνθων.

## 1.5 Οργάνωση, κεφαλαίωση, διάρθρωση της εργασίας

Στο κεφάλαιο 2 παρουσιάζεται η σχετική βιβλιογραφία καθώς και σχετικές έρευνες που ασχολήθηκαν με την εκτίμηση της πρόθεσης του ανθρώπου σε διάφορες εφαρμογές. Στο κεφάλαιο 3 περιγράφεται το σχέδιο της υλοποίησης του συστήματος και αναλύονται όλα τα μέρη που το αποτελούν. Στο κεφάλαιο 4 αναλύονται τα αποτελέσματα της εκμάθησης του μοντέλου της μηχανικής μάθησης καθώς και τα αποτελέσματα από τη σύγκριση του μοντέλου μηχανικής μάθησης με μια μέθοδο από τη βιβλιογραφία, ενώ στο κεφάλαιο 5 δίνεται μια σύνοψη της εργασίας και δίνονται πιθανές αξιοποιήσεις της εργασίας ως έχει καθώς και δίνονται μελλοντικές επεκτάσεις επάνω σε αυτή.





## Κεφάλαιο 2

# Θεωρητικό μέρος – Βιβλιογραφική έρευνα – Σχετικές προσπάθειες

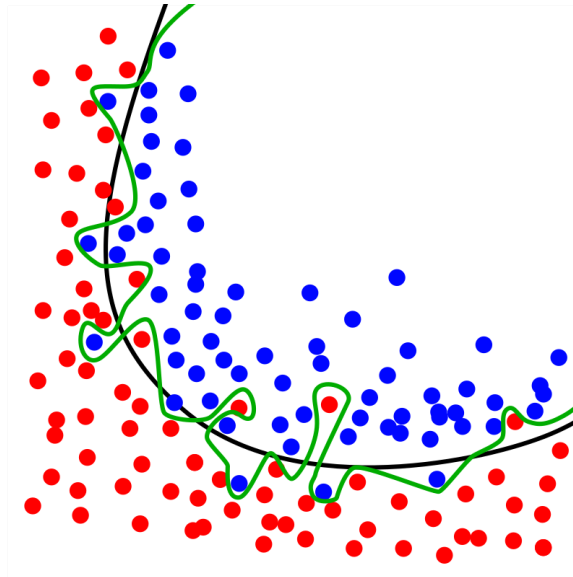
## 2.1 Βιβλιογραφική έρευνα

### 2.1.1 Μηχανική Μάθηση

Η μηχανική μάθηση [11] είναι μια συλλογή από υπολογιστικές μεθόδους μέσω των οποίων πραγματοποιούνται προβλέψεις αξιοποιώντας την "εμπειρία" τους. Ο όρος "εμπειρία" αναφέρεται στην ιδιότητα των μεθόδων της μηχανικής μάθησης να μαθαίνουν από προηγούμενα δεδομένα τα οποία έχουν συλλεχτεί και έχουν αναλυθεί, καθώς και να προσαρμόζονται ανάλογα σε αυτά. Προκειμένου οι προβλέψεις να είναι ακριβείς και σωστές, τα δεδομένα οφείλουν να είναι υψηλής ποιότητας και αρκετά σε πλήθος ούτως ώστε ο αλγόριθμος να καταφέρει να εκπαιδευτεί επαρκώς. Με την έννοια της υψηλής ποιότητας των δεδομένων εννοείται πως τα δεδομένα εκπαίδευσης πρέπει να είναι ορθά. Λόγω του ότι η επιτυχία του αλγόριθμου εξαρτάται από τα δεδομένα που χρησιμοποιήθηκαν, η μηχανική μάθηση σχετίζεται με την ανάλυση δεδομένων και τη στατιστική.

Η μηχανική μάθηση έχει να κάνει με τη σχεδίαση αποτελεσματικών και ακριβείς αλγόριθμους πρόβλεψης, κάτι που υποδηλώνει τη φύση των προβλημάτων και τη σχέση τους με την επιστήμη των υπολογιστών. Εκτός από τη χρονική και τη χωρική πολυπλοκότητά τους, στη μηχανική μάθηση έχει σημασία η έννοια της πολυπλοκότητας του δείγματος για να αξιολογηθεί το μέγεθος του δείγματος που απαιτείται για να μάθει ο αλγόριθμος μια οικογένεια εννοιών. Γενικότερα, οι θεωρητικές εγγυήσεις μάθησης για έναν αλγόριθμο εξαρτώνται από την πολυπλοκότητα των εξεταζόμενων κλάσεων εννοιών και το μέγεθος του δείγματος εκπαίδευσης.

Ο σκοπός ενός αλγόριθμου της μηχανικής μάθησης είναι να μπορεί να κάνει σωστή γενίκευση, δηλαδή να μάθει τα γενικά χαρακτηριστικά των δεδομένων και τη σχέση τους με το αποτέλεσμα της εξόδου χωρίς να υπερπροσαρμόζεται σε αυτά. Μια οπτικοποίηση της υπερπροσαρμογής (overfitting) φαίνεται στην εικόνα 2, όπου τα δεδομένα είναι χωρισμένα σε δυο κατηγορίες: κόκκινα και μπλε. Δεδομένου ότι τα δεδομένα εκπαίδευσης (training data) του αλγόριθμου μηχανικής μάθησης είναι αυτά που βλέπουμε στην εικόνα, το μοντέλο που χαράσσει τη μαύρη διαχωριστική γραμμή έχει καλύτερη γενίκευση από το μοντέλο που χαράσσει την πράσινη διαχωριστική γραμμή, παρά το γεγονός ότι δυο μεριές έχουν ανάμεικτα σημεία (και κόκκινα αλλά και μπλε). Αυτό φαίνεται καλύτερα όταν τα δυο αυτά μοντέλα δεν προβλέπουν τις κατηγορίες από δεδομένα που ήδη έχουν δει και έχουν εκπαιδευτεί επάνω τους, αλλά σε άγνωστα για αυτά δεδομένα. Η διαφορά έγκειται στο γεγονός ότι το μοντέλο της μαύρης γραμμής θα έχει μεγαλύτερη ακρίβεια σε άγνωστα δεδομένα σε σύγκριση με το πράσινο, καθώς το πράσινο έχει μάθει μόνο τα δεδομένα της εκπαίδευσης, όχι το χαρακτηριστικό που διαχωρίζει τις δυο κατηγορίες.



Εικόνα 2. Παράδειγμα υπερπροσαρμογής. Η μαύρη γραμμή δείχνει πως το μοντέλο έχει μια καλή γενίκευση, ενώ η πράσινη υποδηλώνει υπερπροσαρμογή του μοντέλου στα δεδομένα.

### Είδη προβλημάτων στα οποία μπορεί να εφαρμοστεί η μηχανική μάθηση

Η μηχανική μάθηση είναι ικανή να εφαρμοστεί στα παρακάτω προβλήματα:

- **Ταξινόμηση - Classification**

Το πρόβλημα αυτό αφορά την ταξινόμηση κάθε στοιχείο. Στο προηγούμενο παράδειγμα, αυτό είναι η ταξινόμηση κάθε σημείου σε κόκκινο ή μπλε. Συχνά ο αριθμός των κατηγοριών είναι λίγες, συχνά λιγότερες από μερικές εκατοντάδες, παρόλα αυτά σε ειδικές περιπτώσεις όπως αναγνώριση και ταξινόμηση κειμένου ή αναγνώριση της ομιλίας μπορεί να ξεπεράσει αυτό το όριο.

- **Παλινδρόμηση - Regression**

Η πρόβλεψη μιας πραγματικής αξίας για κάθε στοιχείο. Αυτό θα μπορούσε να είναι η αξία ενός διαμερίσματος ή η αξία μιας μετοχής. Στην παλινδρόμηση, η ποινή για μια "λανθασμένη" πρόβλεψη εξαρτάται από την εγγύτητα της τιμής της πρόβλεψης και της πραγματικής τιμής. Στο πρόβλημα της ταξινόμησης η έννοια της εγγύτητας δεν υπάρχει.

- **Κατάταξη - Ranking**

Το πρόβλημα της κατάταξης είναι η εκμάθηση του αλγόριθμου ούτως ώστε να μπορεί να ταξινομεί διάφορα αντικείμενα με βάση κάποιο κριτήριο. Για παράδειγμα, η σειρά με την οποία μια μηχανή αναζήτησης στο internet θα γυρίσει τις σελίδες αποτελεί ένα τέτοιο πρόβλημα, όπου το κριτήριο είναι το πόσο σχετικές είναι με το ερώτημα της αναζήτησης. Αυτού του είδους τα προβλήματα προκύπτουν συχνά στην επεξεργασία της φυσικής γλώσσας (natural language processing systems), καθώς και σε συστήματα εξαγωγής πληροφοριών.

- **Ομαδοποίηση (Συσταδοποίηση) - Clustering**

Αυτού του είδους το πρόβλημα είναι ο διαχωρισμός των δεδομένων σε ομοιογενή υποσύνολα και χρησιμοποιείται όταν χρειάζεται να αναλυθεί ένα πολύ μεγάλο σύνολο δεδομένων. Για παράδειγμα, σε συστήματα που συστήνουν διαφημίσεις (κυρίως σε διαδικτυακές εφαρμογές), ομαδοποιούν τους ανθρώπους με βάση τα ενδιαφέροντά τους, προηγούμενες αναζητήσεις κ.α προκειμένου να δώσουν διαφημίσεις σε αυτούς που θα τους ενδιαφέρουν.

- **Μείωση διαστάσεων ή μάθηση πολλαπλών μορφών - Dimensionality reduction or manifold learning**

Το πρόβλημα εδώ είναι η αναπαράσταση των στοιχείων σε πολλαπλές διαστάσεις, όπου για την επεξεργασία τους απαιτείται η μετατροπή των στοιχείων σε μια αναπαράσταση λιγότερων διαστάσεων, διατηρώντας παράλληλα κάποιες από τις ιδιότητες από την αρχική αναπαράσταση. Το συγκεκριμένο πρόβλημα αφορά την προεπεξεργασία ψηφιακών εικόνων στην μηχανική όραση / εργασίες όρασης υπολογιστών (Computer Vision, CV).

### Αξιολόγηση του μοντέλου της μηχανικής μάθησης

Για την αξιολόγηση ενός μοντέλου μηχανικής μάθησης, χρησιμοποιούνται διάφορες μετρικές (metrics). Κάθε εργασία μηχανικής μάθησης μπορεί να αναλυθεί είτε σε παλινδρόμηση είτε σε ταξινόμηση, η κάθε μία με τις δικές της μετρικές. Παρακάτω θα αναλυθούν κάποιες από τις προαναφερθείσες μετρικές.

- **Μετρικές για προβλήματα παλινδρόμησης**

Τα μοντέλα παλινδρόμησης έχουν συνεχή έξοδο. Επομένως, χρειαζόμαστε μια μετρική που να βασίζεται στον υπολογισμό κάποιου είδους απόστασης μεταξύ της προβλεπόμενης και της πραγματικής τιμής.

- Μέσο Απόλυτο Σφάλμα (Mean Absolute Error, MAE)

Το μέσο απόλυτο σφάλμα είναι ο μέσος όρος της διαφοράς μεταξύ των πραγματικών τιμών και των προβλεπόμενων τιμών. Μαθηματικά, αναπαρίσταται ως εξής:

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j|$$

Όπου:

$y_j$ : Η πραγματική τιμή

$\hat{y}_j$ : Η τιμή της πρόβλεψης

N: Ο αριθμός των δεδομένων

Μερικά βασικά σημεία για το MAE:

- \* Είναι πιο ανθεκτικό έναντι των ακραίων τιμών από τον MAE, καθώς δεν υπερτονίζει τα σφάλματα.
- \* Μας δίνει ένα μέτρο του πόσο απέχουν οι προβλέψεις από την πραγματική τιμή. Ωστόσο, δεδομένου ότι το MAE χρησιμοποιεί την απόλυτη τιμή του υπολοίπου, δεν μας δίνει μια ιδέα για την κατεύθυνση του σφάλματος, δηλαδή αν υπο-προβλέπουμε ή υπερ-προβλέπουμε τα δεδομένα.
- \* Το MAE δεν είναι διαφοροποιήσιμο, σε αντίθεση με το MSE.

- Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error, MSE)

Το μέσο τετραγωνικό σφάλμα είναι ίσως η πιο δημοφιλής μετρική που χρησιμοποιείται για προβλήματα παλινδρόμησης. Ουσιαστικά βρίσκει το μέσο όρο της τετραγωνικής διαφοράς μεταξύ της αληθινής τιμής και της τιμής που προβλέπει το μοντέλο παλινδρόμησης.

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2$$

Όπου:

$y_j$ : Η πραγματική τιμή

$\hat{y}_j$ : Η τιμή της πρόβλεψης

N: Ο αριθμός των δεδομένων

Μερικά βασικά σημεία για το MSE:

- \* Είναι διαφοροποιήσιμο, οπότε μπορεί να βελτιστοποιηθεί καλύτερα.
- \* Επιβαρύνει ακόμη και τα μικρά σφάλματα με τον τετραγωνισμό τους, γεγονός που ουσιαστικά οδηγεί σε υπερεκτίμηση του πόσο κακό είναι το μοντέλο.
- \* Η ερμηνεία των σφαλμάτων πρέπει να γίνεται με γνώμονα τον παράγοντα τετραγωνισμού (κλίμακα).
- \* Λόγω του τετραγωνισμού της διαφοράς, είναι θεμελιωδώς πιο επιρρεπής σε ακραίες τιμές σε σχέση με άλλες μετρικές.

– Ρίζα Μέσου Τετραγωνικού Σφάλματος (Root Mean Squared Error, RMSE)

Η ρίζα του μέσου τετραγωνικού σφάλματος αντιστοιχεί στην τετραγωνική ρίζα του μέσου όρου της τετραγωνικής διαφοράς μεταξύ της τιμής στόχου και της τιμής που προβλέπει το μοντέλο παλινδρόμησης. Στην ουσία, είναι η τετραγωνική ρίζα του MSE. Παριστάνεται μαθηματικά ως εξής:

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2}$$

Όπου:

$y_j$ : Η πραγματική τιμή

$\hat{y}_j$ : Η τιμή της πρόβλεψης

N: Ο αριθμός των δεδομένων

Μερικά βασικά σημεία για το RMSE:

- \* Διατηρεί τη διαφοροποιήσιμη ιδιότητα του MSE.
- \* Αντιμετωπίζει την τιμωρία των μικρότερων σφαλμάτων που γίνεται από το MSE με τετραγωνική ρίζα.
- \* Η ερμηνεία των σφαλμάτων μπορεί να γίνει ομαλά, δεδομένου ότι η κλίμακα είναι τώρα η ίδια με την τυχαία μεταβλητή.
- \* Δεδομένου ότι οι συντελεστές κλίμακας είναι ουσιαστικά κανονικοποιημένοι, είναι λιγότερο επιρρεπής στις περιπτώσεις ακραίων τιμών.

• **Μετρικές για προβλήματα ταξινόμησης**

Τα προβλήματα ταξινόμησης είναι ένας από τους πιο ευρέως ερευνημένους τομείς παγκοσμίως. Περιπτώσεις χρήσης υπάρχουν σε όλα σχεδόν τα παραγωγικά και βιομηχανικά περιβάλλοντα. Μερικά παραδείγματα είναι η αναγνώριση ομιλίας, η αναγνώριση προσώπου και η ταξινόμηση κειμένου.

Τα μοντέλα ταξινόμησης έχουν διακριτή έξοδο, επομένως χρειάζεται μια μετρική που να συγκρίνει διακριτές κλάσεις με κάποια μορφή. Οι μετρικές ταξινόμησης αξιολογούν την απόδοση ενός μοντέλου και δείχνουν την ποιότητα της ταξινόμησης, αλλά κάθε μία από αυτές αξιολογεί το μοντέλο με διαφορετικό τρόπο.

– Ορθότητα (Accuracy)

Η ορθότητα της ταξινόμησης είναι η πιο απλή μετρική που μπορεί να χρησιμοποιηθεί. Είναι ο αριθμός των σωστών προβλέψεων προς το συνολικό αριθμό των προβλέψεων, πολλαπλασιασμένος με το 100.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Number of all Predictions}} * 100\%$$

– Πίνακας Σύγχυσης (Confusion Matrix)

Ο πίνακας σύγχυσης δεν είναι πραγματική μετρική, παρόλα αυτά η κατανόησή του είναι θεμελιώδης για την κατανόηση των υπόλοιπων μετρικών. Αποτελεί μια απεικόνιση σε μορφή πίνακα των ετικετών της πραγματικής τιμής και της τιμής από την πρόβλεψη του μοντέλου. Η κάθε γραμμή του αντιπροσωπεύει τις περιπτώσεις σε μία προβλεπόμενη κλάση και κάθε στήλη αντιπροσωπεύει τις περιπτώσεις στην πραγματική κλάση ή και το αντίστροφο. Για να κατανοηθεί ο πίνακας σύγχυσης, το παράδειγμα της ταξινόμησης ενός αντικειμένου σε σκύλο ή μη θα χρησιμοποιηθεί (βλ. Εικ. 3). Λαμβάνοντας τη μηδενική υπόθεση  $H^0$  "Το αντικείμενο είναι σκύλος". Κάθε κελί στον πίνακα σύγχυσης αντιπροσωπεύει έναν παράγοντα αξιολόγησης.

		Predicted	
		Dog	Not Dog
Ground Truth	Dog	TP	FP
	Not Dog	FN	TN

Εικόνα 3. Ένας πίνακας σύγχυσης για την ταξινόμηση αντικειμένου σε σκύλο ή μη σκύλο.

- \* Το True Positive (TP) δηλώνει πόσα θετικά δείγματα κλάσεων προέβλεψε σωστά το μοντέλο.
- \* Το True Negative (TN) δηλώνει πόσα αρνητικά δείγματα κλάσεων προέβλεψε σωστά το μοντέλο.
- \* Το False Positive (FP) δηλώνει πόσα αρνητικά δείγματα κλάσεων προέβλεψε λανθασμένα το μοντέλο. Αυτός ο παράγοντας αντιπροσωπεύει το σφάλμα τύπου A στη στατιστική ονοματολογία. Αυτή η τοποθέτηση σφάλματος στον πίνακα σύγχυσης εξαρτάται από την επιλογή της μηδενικής υπόθεσης.

\* Ο συντελεστής False Negative(FN) δηλώνει πόσα θετικά δείγματα κλάσεων προέβλεψε λανθασμένα το μοντέλο. Αυτός ο παράγοντας αντιπροσωπεύει σφάλμα τύπου B στη στατιστική ονοματολογία. Αυτή η τοποθέτηση σφάλματος στον πίνακα σύγκρισης εξαρτάται επίσης από την επιλογή της μηδενικής υπόθεσης.

#### – Ακρίβεια (Precision)

Η ακρίβεια είναι ο λόγος των πραγματικών θετικών αποτελεσμάτων προς το σύνολο των θετικών αποτελεσμάτων που προβλέπονται:

$$P = \frac{TP}{TP + FP}$$

Για το παραπάνω παράδειγμα της ταξινόμησης σε σκύλους και μη:

$$P = \frac{\text{σωστά ταξινομημένοι σκύλοι}}{\text{σωστά ταξινομημένοι σκύλοι} + \text{σκύλοι ταξινομημένοι λανθασμένα ως μη σκύλοι}}$$

Το εύρος των τιμών του P:  $0 \leq P \leq 1$

Η μετρική ακρίβειας επικεντρώνεται στα σφάλματα τύπου A (FP). Ένα σφάλμα τύπου A συμβαίνει όταν απορρίπτεται μια αληθής μηδενική υπόθεση( $H^0$ ). Έτσι, σε αυτή την περίπτωση, το σφάλμα τύπου A είναι η εσφαλμένη επισήμανση σκύλων ως μη.

Μια βαθμολογία ακρίβειας 1 δείχνει ότι το μοντέλο δεν έχασε κανένα αληθώς θετικό αποτέλεσμα και είναι σε θέση να ταξινομήσει καλά μεταξύ σωστής και λανθασμένης επισήμανσης των σκύλων. Αυτό που δεν μπορεί να μετρήσει είναι η ύπαρξη σφάλματος τύπου B, το οποίο είναι τα ψευδώς αρνητικά - περιπτώσεις όπου ένας μη σκύλος αναγνωρίζεται ως σκύλος.

Ένα χαμηλό σκορ ακρίβειας ( $<0,5$ ) σημαίνει ότι ο ταξινομητής έχει μεγάλο αριθμό ψευδώς θετικών αποτελεσμάτων, τα οποία μπορεί να είναι αποτέλεσμα ανισόρροπης κλάσης ή μη συντονισμένων υπερπαραμέτρων του μοντέλου.

#### – Ανάκληση (Recall)

Η ανάκληση είναι ουσιαστικά ο λόγος των πραγματικών θετικών αποτελεσμάτων προς όλα τα πραγματικά θετικά αποτελέσματα.

$$R = \frac{TP}{TP + FN}$$

Για το παράδειγμα της ταξινόμησης σε σκύλους και μη:

$$R = \frac{\text{σωστά ταξινομημένοι σκύλοι}}{\text{σωστά ταξινομημένοι σκύλοι} + \text{μη σκύλοι ταξινομημένοι λανθασμένα ως σκύλοι}}$$

Το εύρος των τιμών του R:  $0 \leq R \leq 1$

Η μετρική ανάκλησης επικεντρώνεται στα σφάλματα τύπου Β (FN). Ένα σφάλμα τύπου Β συμβαίνει όταν αποδεχόμαστε μια ψευδή μηδενική υπόθεση ( $H^0$ ). Έτσι, σε αυτή την περίπτωση, το σφάλμα τύπου Β είναι η εσφαλμένη επισήμανση μη σκύλων ως σκύλων.

Όταν η ανάκληση είναι 1 σημαίνει ότι το μοντέλο δεν έχασε κανένα αληθώς θετικό αποτέλεσμα και είναι σε θέση να ταξινομήσει καλά μεταξύ της σωστής και λανθασμένης επισήμανσης των σκύλων.

Αυτό που δεν μπορεί να μετρήσει είναι η ύπαρξη σφάλματος τύπου Α, το οποίο είναι τα ψευδώς θετικά αποτελέσματα, δηλαδή οι περιπτώσεις όπου ένας σκύλος αναγνωρίζεται ως μη σκύλος.

Ένα χαμηλό σκορ ανάκλησης ( $<0,5$ ) σημαίνει ότι ο ταξινομητής έχει μεγάλο αριθμό ψευδώς αρνητικών αποτελεσμάτων, τα οποία μπορεί να είναι αποτέλεσμα ανισόρροπης κλάσης ή μη συντονισμένων υπερπαραμέτρων του μοντέλου.

#### – F1-score

Το F1-score είναι μια μετρική που χρησιμοποιεί την ακρίβεια και την ανάκληση.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

Όπου:

P: Ακρίβεια

R: Ανάκληση

Ένα υψηλό F1-score συμβολίζει υψηλή ακρίβεια καθώς και υψηλή ανάκληση. Παρουσιάζει μια καλή ισορροπία μεταξύ ακρίβειας και ανάκλησης και δίνει καλά αποτελέσματα σε ανισόροπα προβλήματα ταξινόμησης. Υψηλό F1-score σημαίνει ότι πιθανότατα το μοντέλο έχει υψηλή ακρίβεια και ανάκληση σε ένα μεγάλο μέρος της απόφασης (το οποίο είναι πληροφοριακό).

Ένα χαμηλό F1-score δεν υποδυναμίζει σχεδόν τίποτα. Για παράδειγμα, η χαμηλή ανάκληση υποδηλώνει ότι το μοντέλο δεν τα πήγε καλά σε όλες τις ταξινομήσεις των δεδομένων προς δοκιμή, ενώ η χαμηλή ακρίβεια δείχνει πως για όλες τις περιπτώσεις που το μοντέλο ταξινόμησε ως θετικές, δεν πέτυχε πολλές από αυτές σωστά. Όμως το F1-score δεν δείχνει ποιο είναι το πρόβλημα: χαμηλή ακρίβεια ή χαμηλή ανάκληση, και αν το μοντέλο είναι επιρρεπές σε σφάλματα τύπου Α ή τύπου Β.

Παρακάτω θα αναφερθούν κάποιες μέθοδοι με τις οποίες ένας αλγόριθμος μηχανικής μάθησης μπορεί να εκπαιδευτεί.

#### • **Επιβλεπόμενη μάθηση - Supervised learning**

Ο αλγόριθμος λαμβάνει ένα σύνολο από κατηγοριοποιημένα παραδείγματα ως δεδομένα εκπαίδευσης, εκπαιδευτεί επάνω σε αυτά και μετά χρησιμοποιείται για να κάνει προβλέψεις σε δεδομένα που δεν έχει εκπαιδευτεί. Το σενάριο αυτό είναι ένα από τα πιο συνηθισμένα και σχετίζεται με την ταξινόμηση (classification) που αναφέραμε νωρίτερα. Ένα τέτοιο παράδειγμα μπορεί να είναι εικόνες με γάτες και σκύλους, όπου ο αλγόριθμος προσπαθεί να μάθει από τα δεδομένα εκπαίδευσης και ύστερα να εφαρμόσει την "εμπειρία" του σε νέες φωτογραφίες, προβλέποντας τι από τα δύο απεικονίζεται.

- **Μη επιβλεπόμενη μάθηση - Unsupervised learning**

Ο αλγόριθμος μηχανικής μάθησης λαμβάνει αποκλειστικά μη κατηγοριοποιημένα δεδομένα εκπαίδευσης και καλείται αφού μάθει από αυτά να κάνει προβλέψεις για καινούρια δεδομένα που δεν έχει δει. Λόγω του ότι δεν έχουμε κατηγοριοποιημένα δεδομένα, είναι δύσκολο ένας τέτοιος αλγόριθμος να μπορέσει να αξιολογηθεί, καθώς δεν είναι εύκολη η κατανόησή του. Παρόλα αυτά, η μη επιβλεπόμενη μάθηση μπορεί να χρησιμοποιηθεί στην περίπτωση που η ομάδα των ανθρώπων έχουν δεδομένα στα χέρια τους και δεν ξέρουν τι ακριβώς ψάχνουν μέσα από αυτά. Ο αλγόριθμος είναι ικανός να βρεί άγνωστες ομοιότητες ή και διαφορές μέσα από αυτά, καθώς και να ορίσει ομάδες (συστάδες) μόνος του μέσα από αυτές. Ένα παράδειγμα για τη μη επιβλεπόμενη μάθηση είναι σε συστήματα συστάσεων σε πλατφόρμες ηλεκτρονικού εμπορίου (e-commerce). Πολλές πλατφόρμες έχουν ήδη ενσωματώσει ένα τέτοιο σύστημα το οποίο λαμβάνει δεδομένα από κάθε αγοραστή και αναγνωρίζει ότι συγκεκριμένα προϊόντα τείνουν να αγοραστούν μαζί. Έτσι, μελλοντικά κάποιος ο οποίος πάει να αγοράσει ένα προϊόν, το σύστημα θα του προτείνει προϊόντα τα οποία έμαθε από προηγούμενους αγοραστές ότι πιθανότατα τον ενδιαφέρουν.

- **Ημιεπιβλεπόμενη μάθηση - Semi-supervised learning**

Η ημιεπιβλεπόμενη μάθηση είναι ένας συνδυασμός της επιβλεπόμενης και της μη επιβλεπόμενης μάθησης, με την έννοια ότι τα δεδομένα εκπαίδευσης έχουν τόσο κατηγοριοποιημένα δεδομένα όσο και μη, ενώ ο σκοπός του αλγόριθμου είναι η πρόβλεψη όλων των άγνωστων δεδομένων. Η ημιεπιβλεπόμενη μάθηση εφαρμόζεται συχνά σε περιπτώσεις όπου τα μη κατηγοριοποιημένα δεδομένα είναι εύκολο να βρεθούν, αλλά η κατηγοριοποίησή τους είναι δαπανηρή (π.χ χρονικά). Διάφοροι τύποι προβλημάτων που προκύπτουν σε εφαρμογές, συμπεριλαμβανομένων των εργασιών ταξινόμησης, παλινδρόμησης ή κατάταξης, μπορούν να διαμορφωθούν ως περιπτώσεις ημιεπιβλεπόμενης μάθησης. Ο λόγος ύπαρξης της ημιεπιβλεπόμενης μάθησης είναι η ελπίδα ότι η κατανομή των μη επισημειωμένων δεδομένων που είναι διαθέσιμα στον αλγόριθμο μπορεί να το βοηθήσει στο να έχει καλύτερη αποτελεσματικότητα και επίδοση σε σύγκριση με ένα περιβάλλον επιβλεπόμενης μάθησης. Η ανάλυση και η εύρεση των περιστάσεων κάτω από τις οποίες είναι εφικτή η χρήση της ημιεπιβλεπόμενης μάθησης απασχολεί μεγάλο κομμάτι της σύγχρονης θεωρίας και εφαρμοσμένης έρευνας του πεδίου της μηχανικής μάθησης.

- **Μεταγωγική συμπερασματολογία - Transductive inference**

Όμοια με την ημιεπιβλεπόμενη μάθηση, τα δεδομένα διαθέσιμα στον αλγόριθμο αποτελούνται από κατηγοριοποιημένα δεδομένα και μη. Η διαφορά με την ημιεπιβλεπόμενη μάθηση έγκειται στο στόχο της μεταγωγικής συμπερασματολογίας, ο οποίος είναι να προβλέψει ετικέτες μόνο για συγκεκριμένα σημεία δοκιμής. Η μεταγωγική συμπερασματολογία φαίνεται να είναι ευκολότερη και ταιριάζει με το σενάριο που συναντάται σε διάφορες σύγχρονες εφαρμογές. Ωστόσο, όπως και στην περίπτωση της ημιεπιβλεπόμενης διάταξης, οι υποθέσεις υπό τις οποίες μπορεί να επιτευχθεί καλύτερη απόδοση σε αυτό το πλαίσιο αποτελούν αντικείμενο ερευνητικού ενδιαφέροντος που δεν έχουν επιλυθεί πλήρως.

- **Ζωντανή μάθηση - On-line learning** Σε αντίθεση με όλα τα προηγούμενα σενάρια, στη ζωντανή μάθηση εμπλέκονται πολλά στάδια εκπαίδευσης και δοκιμής (training and testing), τα οποία αναμειγνύονται όσον αφορά τη χρονική σειρά με την οποία λαμβάνουν χώρα. Σε κάθε δοκιμή, ο αλγόριθμος λαμβάνει μη-κατηγοριοποιημένα δεδομένα εισόδου, κάνει πρόβλεψη, εν συνεχεία λαμβάνει την αληθή κατηγορία και ανάλογα λαμβάνει μια ποινή (loss), σε αντίθεση με τις προηγούμενες μεθόδους όπου ο αλγόριθμος εκπαιδευόταν σε όλα τα δεδομένα εκπαίδευσης με τη μια. Η ζωντανή μάθηση χρησιμοποιείται συχνά σε προβλήματα όπου είναι υπολογιστικά αδύνατη η εκπαίδευση επάνω σε όλα τα δεδομένα. Παραδείγματα



χρήσης της ζωντανής μάθησης είναι η εκτίμηση της αξίας των μετοχών, όπου τα δεδομένα είναι μια συνάρτηση του χρόνου και απαιτείται από τον αλγόριθμο να μαθαίνει δυναμικά μαθαίνοντας καινούρια μοτίβα στα δεδομένα. Αξίζει να σημειωθεί πως οι αλγόριθμοι που κάνουν χρήση της ζωντανής μάθησης είναι επιρρεπή στο φαινόμενο της "καταστροφικής παρεμβολής" (catastrophic interference), όπου ο αλγόριθμος μπορεί να ξεχάσει απότομα και πλήρως τα προηγούμενα δεδομένα που έχει μάθει τη στιγμή που μάθει κάτι καινούριο. Αυτό είναι η αρνητική πλευρά της ιδιότητας του αλγόριθμου να μαθαίνει καινούρια μοτίβα διαρκώς.

- **Ενισχυτική μάθηση - Reinforcement learning**

Στην ενισχυτική μάθηση, όπως και στη ζωντανή μάθηση, οι φάσεις εκπαίδευσης και δοκιμής αναμειγνύονται. Για να λάβει πληροφορίες για την εκπαίδευσή του, ο αλγόριθμος αλληλεπιδρά ενεργά με το περιβάλλον, ενώ σε ορισμένες περιπτώσεις ακόμα το επηρεάζει, όπου σε κάθε αλληλεπίδρασή του λαμβάνει μια "επιβράβευση" (reward). Η επιβράβευση συσχετίζεται άμεσα με την πράξη του αλγόριθμου στο περιβάλλον, κάτι το οποίο θα οδηγήσει τον αλγόριθμο σταδιακά στη μεγιστοποίηση της επιβράβευσής του, μαθαίνοντας και επιδρώντας με τον τρόπο τον οποίο του προσφέρει τη μεγαλύτερη επιβράβευση. Ωστόσο, η επιβράβευση δίνεται με βάση την κάθε πράξη ξεχωριστά και δεν έχει τρόπο για την επιβράβευση της ολικής αλληλεπίδρασής του με το περιβάλλον, κάτι το οποίο θέτει ένα δίλημμα στον αλγόριθμο μηχανικής μάθησης: να προσπαθήσει να βρεί έναν καινούριο τρόπο αλληλεπίδρασης με το περιβάλλον με τον οποίο μπορεί να πετύχει μεγαλύτερη επιβράβευση (exploration) ή να συνεχίσει να πράττει με τον ίδιο τρόπο, αξιοποιώντας ότι έχει μάθει έως τότε (exploitation). Έτσι, ο αλγόριθμος κατά περίπτωση θα πρέπει να έχει μια ισορροπία ως προς την εξερεύνηση νέων δεδομένων και την εκμετάλλευση όσων ήδη γνωρίζει.

- **Ενεργής μάθηση - Active learning**

Η ενεργής μάθηση είναι μια ιδιαίτερη περίπτωση της μηχανικής μάθησης, όπου ο αλγόριθμος μπορεί διαδραστικά να ζητήσει πληροφορίες από έναν χρήστη (ή κάποια άλλη πηγή πληροφοριών) για να κατηγοριοποιήσει νέα δεδομένα με τον επιθυμητό τρόπο. Η ενεργής μάθηση βρίσκει μεγάλη χρήση σε σενάρια όπου υπάρχει πληθώρα μη κατηγοριοποιημένων δεδομένων, όπου η ταξινόμηση από τον άνθρωπο είναι δαπανηρή. Σε αυτού του είδους τις περιπτώσεις, η ενεργής μάθηση μπορεί να αξιοποιηθεί, όπου ζητάει από τον "μάντη" ή "δάσκαλο" (teacher/oracle στην αγγλική βιβλιογραφία) ή αλλιώς την πηγή της πληροφορίας για τις κατηγορίες των δεδομένων. Η ενεργής μάθηση είναι μια μέθοδος επαναληπτικής επιβλεπόμενης μάθησης. Επειδή ο αλγόριθμος είναι σε θέση ο ίδιος να διαλέξει τα δεδομένα επάνω στα οποία θέλει να μάθει την κατηγορία από τον δάσκαλο, μπορεί να μειώσει σημαντικά το πλήθος των δεδομένων που χρειάζεται για να μάθει μια έννοια σε σύγκριση με την απλή επιβλεπόμενη μάθηση. Ωστόσο, ο αλγόριθμος διατρέχει τον κίνδυνο να κατακλυστεί από μη χρήσιμα παραδείγματα.

## 2.2 Σχετικές προσπάθειες – έρευνες

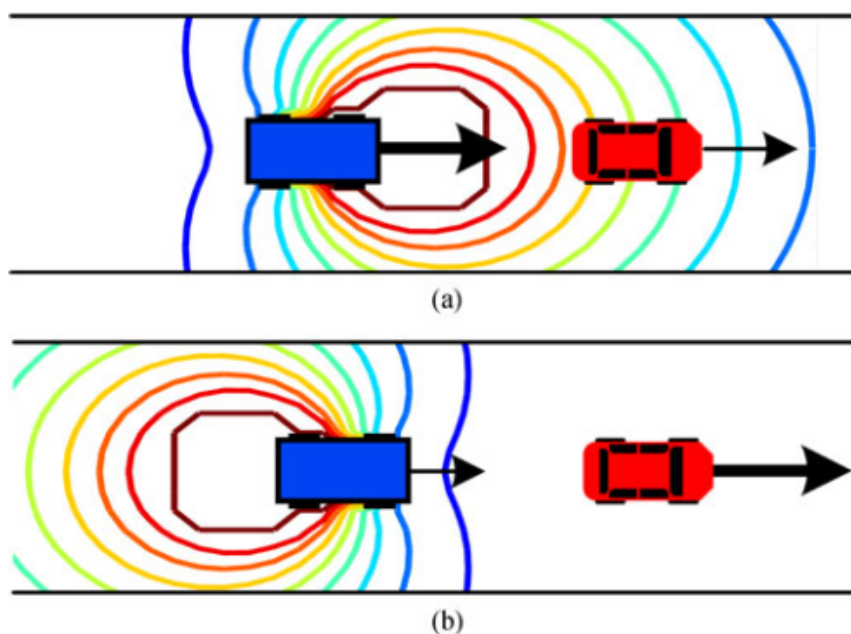
Μεγάλη προσπάθεια έχει καταβληθεί από την επιστημονική κοινότητα για την εκτίμηση της ανθρώπινης πρόθεσης σε περιβάλλοντα και εφαρμογές αρκετά διαφορετικές μεταξύ τους.

### 2.2.1 Μηχανική Μάθηση

Οι Woo *et al* [12] [13] εφηύραν μια νέα μέθοδο για την εκτίμηση της πρόθεσης του οδηγού ενός αυτοκινήτου να αλλάξει λωρίδα, καθώς και την εκτίμηση της τροχιάς του. Σε αντίθεση με την προηγούμενη προσέγγιση, η παρούσα προσπάθεια επιχειρεί να εντοπίσει τυχών συγκρούσεις

με άλλα οχήματα που βρίσκονται γύρω από το αυτοκίνητο του ενδιαφέροντος, το οποίο συμπεριλαμβάνει στην τελική εκτίμηση της πρόθεσης του οδηγού.

Με τη χρήση ενός αισθητήρα θέσης RT3003 και 6 σαρωτές λέιζερ Ibeo LUX (LiDAR), το ίδιο το όχημα είχε τη δυνατότητα να βρει τη δική του θέση στο χώρο καθώς και τις θέσεις και τις ταχύτητες των υπόλοιπων αυτοκινήτων γύρω από το ίδιο. Με την επεξεργασία αυτών των δεδομένων προέκυψαν τα εξής χαρακτηριστικά: η απόσταση από την κεντρική γραμμή, την πλευρική ταχύτητα (η ταχύτητα της οποίας ο άξονας είναι κάθετος στην κατεύθυνση του αυτοκινήτου), καθώς και ένα Τεχνητό Πεδίο Δυναμικού (Artificial Potential Field). Το Τεχνητό Πεδίο Δυναμικού προκύπτει από μια σειρά μεθόδων που αναπτύχθηκαν για χρήση σε κινητά ρομπότ και σε ρομποτικούς βραχιώνες με σκοπό την αποφυγή των εμποδίων.



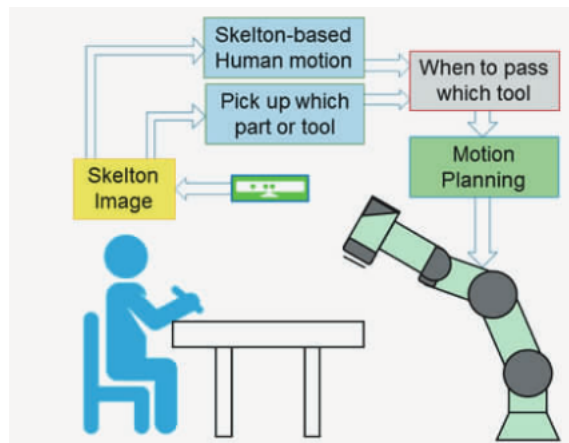
Εικόνα 4. Απεικόνιση του Τεχνητού Πεδίου Δυναμικού. Το κόκκινο αυτοκίνητο είναι το αμάξι του ενδιαφέροντος. Περίπτωση (a): το μπλε αυτοκίνητο κινείται με μεγαλύτερη ταχύτητα από το κόκκινο. Περίπτωση (b): το μπλε αυτοκίνητο έχει μικρότερη ταχύτητα από το κόκκινο. [13]

Τα χαρακτηριστικά αυτά αποτελούν την είσοδο μιας Μηχανής Διανυσμάτων Υποστήριξης (Support Vector Machine, SVM), το οποίο σαν εξόδους έχει την πρόθεση του οδηγού: αν έχει πρόθεση ο οδηγός να αλλάξει λωρίδα ή εάν θα παραμείνει στην ίδια. Η προτεινόμενη μέθοδος πάει ένα βήμα παραπέρα, υπολογίζοντας την τροχιά του αυτοκινήτου και βρίσκοντας μέσω αυτής την μελλοντική θέση του αυτοκινήτου, ενώ στη συνέχεια κάνει έναν έλεγχο για σύγκρουση με διπλανά αυτοκίνητα. Στην περίπτωση που βρεθεί σύγκρουση σε κάποια μελλοντική σύγκρουση, η τροχιά υπολογίζεται εκ νέου και γίνεται νέα εκτίμηση της πρόθεσης του οδηγού.

Η μέθοδος αυτή επιτυγχάνει να μειώσει τις ψευδώς θετικές εκτιμήσεις για την αλλαγή λωρίδας κατά 53% σε σύγκριση με την εκτίμηση της πρόθεσης του οδηγού χωρίς τον έλεγχο σύγκρουσης και επαναυπολογισμού της τροχιάς. Η μέθοδος έχει ποσοστό ακρίβειας (precision) 96.3%, ανάκλησης (recall) 100%, ενώ το σκορ F1 (F1-score) ανέρχεται στο 98.1% και ο χρόνος ανίχνευσης της αλλαγής λωρίδας από τη χρονική στιγμή της πρόβλεψης μέχρι τη χρονική στιγμή που το αμάξι περνάει επάνω από το διαχωριστικό είναι 1.74 δευτερόλεπτα. Σε σύγκριση με άλλες μεθόδους, η συγκεκριμένη έχει καλύτερη απόδοση σε όλα τα παραπάνω κριτήρια.

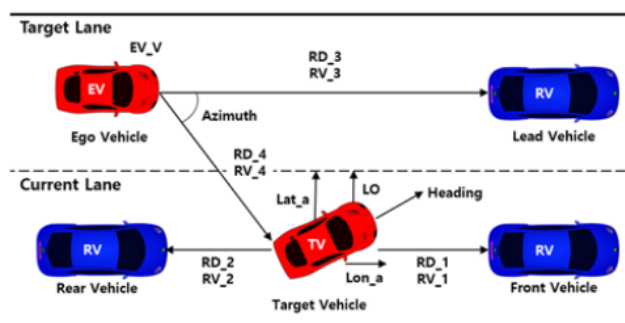
### 2.2.2 Νευρωνικά Δίκτυα

Οι Yan *et al* [14] ανέπτυξαν ένα Νευρωνικό Δίκτυο Μακροχρόνιας Βραχυπρόθεσμης Μνήμης (Long Short-Term Memory, LSTM) για την αναγνώριση της ανθρώπινης πρόθεσης. Η δουλειά αυτή έχει βασιστεί στην εκτίμηση της πρόθεσης του ανθρώπου χρησιμοποιώντας έναν αισθητήρα Kinect V2, μέσω του οποίου καταγράφεται η κίνηση του ανθρώπου. Μέσω αυτής της διεργασίας, η κίνηση του σκελετού του ανθρώπου καταγράφεται, όπου μετά αναλύεται από το λογισμικό και τροφοδοτείται σε ένα σύστημα προγραμματισμού των εργασιών του ρομπότ. Η εικόνα 5 δείχνει ένα τέτοιο σενάριο, όπου το ρομπότ παρακολουθεί τον άνθρωπο και την χρήση των εργαλείων από αυτόν, εξάγει χρήσιμη πληροφορία και την επεξεργάζεται για να εκτιμήσει την πρόθεσή του και στη συνέχεια μέσω του προγραμματισμού της κίνησης (motion planning) του να τον βοηθήσει στη δουλειά του.



Εικόνα 5. Εκτίμησης πρόθεσης ανθρώπου σε σενάριο συνεργασίας ανθρώπου-ρομπότ. [14]

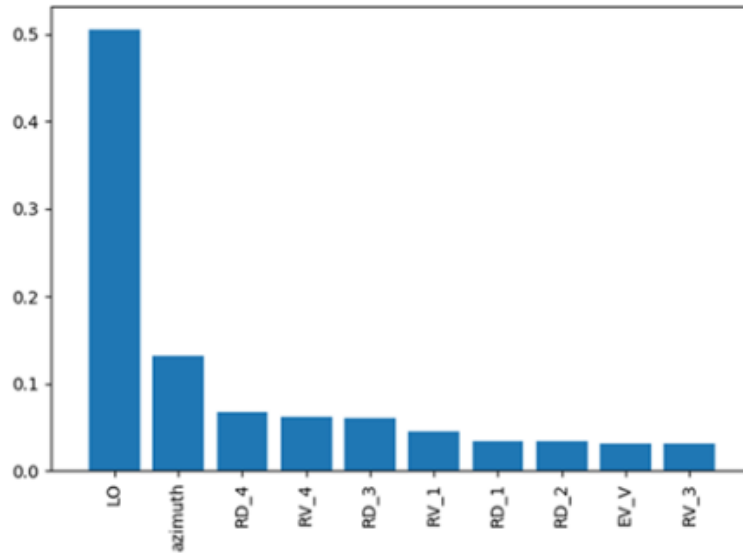
Οι Choi *et al* [15] επιχείρησαν να εκτιμήσουν την πρόθεση των οδηγών αυτοκινήτων για αλλαγή λωρίδας με χρήση μεθόδων Μηχανικής Μάθησης και Νευρωνικών Δικτύων. Η εκτίμηση της πρόθεσης του οδηγού αφορά τον ίδιο τον οδηγό του αυτοκινήτου που εξετάζεται, και όχι των γύρω οδηγών άλλων αυτοκινήτων. Με την χρήση του NGSIM Dataset, ένα dataset που δημιουργήθηκε από τη συλλογή πληροφοριών από κάμερες που παρακολουθούσαν κομμάτια αυτοκινητοδρόμων, εξάχθηκαν πληροφορίες όπως η σχετική απόσταση μεταξύ των αυτοκινήτων, το αζιμουθίο, η κατεύθυνση των οχημάτων, η απόστασή των οχημάτων από τις λωρίδες κ.α.



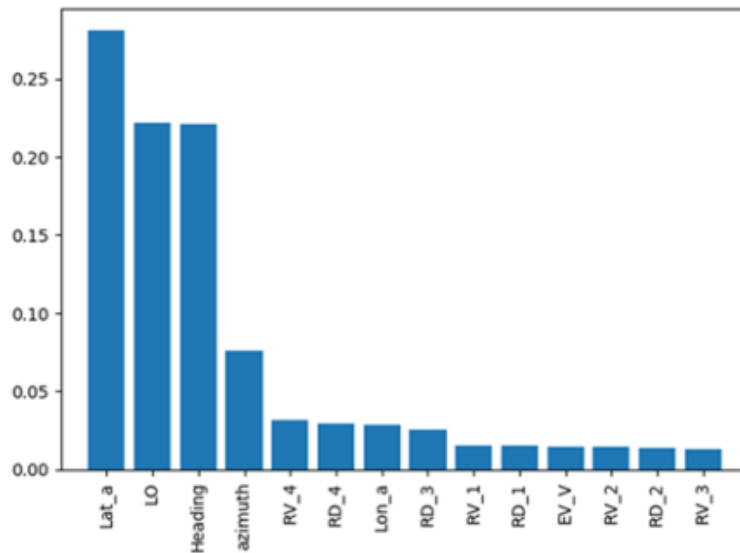
Εικόνα 6. Η εξαγωγή των δεδομένων [15]

Από την επεξεργασία αυτών των δεδομένων, τα δεδομένα χωρίστηκαν σε δύο κατηγορίες: σε αυτά που μπορούν να εξαχθούν από αισθητήρες επάνω στο ίδιο το αυτοκίνητο και σε αυτά που

μπορούν να εξαχθούν από την επικοινωνία μεταξύ των αυτοκινήτων. Έτσι, δημιουργήθηκαν δύο συλλογές δεδομένων, η πρώτη με δεδομένα εισόδου μόνο από τους αισθητήρες του ίδιου του αυτοκινήτου και η δεύτερη με δεδομένα εισόδου τόσο από τους αισθητήρες του αυτοκινήτου όσο και από τα δεδομένα από την επικοινωνία μεταξύ των αυτοκινήτων. Με τη χρήση του κριτηρίου Gini Impurity επιλέχθηκαν τα πιο σημαντικά δεδομένα σε κάθε κατηγορία ούτως ώστε να αποφευχθεί η υπερπροσαρμογή (overfitting) των μοντέλων.



Εικόνα 7. Η σημαντικότητα των χαρακτηριστικών των αισθητήρων [15]



Εικόνα 8. Η σημαντικότητα των χαρακτηριστικών των αισθητήρων μαζί με της επικοινωνίας μεταξύ των οχημάτων [15]

Τα μοντέλα τεχνητής νοημοσύνης που χρησιμοποιήθηκαν ήταν δύο κατηγοριών: Μηχανικής Μάθησης και Επαναλαμβανόμενων Νευρωνικών Δικτύων (Recurrent Neural Network, RNN). Στην κατηγορία της Μηχανικής Μάθησης έγινε χρήση των μοντέλων Τυχαίου Δάσους (Random Forest, RF) και SVM, ενώ στην κατηγορία των Επαναλαμβανόμενων Νευρωνικών Δικτύων χρη-

σιμοποιήθηκαν τα μοντέλα LSTM και Επαναλαμβανόμενων Μονάδων με Πύλη (Gated Recurrent Unit, GRU). Από τα παραπάνω μοντέλα, το Random Forest με τα δεδομένα από την επικοινωνία μεταξύ των αυτοκινήτων και τα δεδομένα από τους αισθητήρες του αυτοκινήτου πέτυχε το καλύτερο αποτέλεσμα με ποσοστό επιτυχίας 82%.

Οι Izquierdo *et al* [16] επιδίωξαν την εκτίμηση της τροχιάς του αυτοκινήτου καθώς και την εκτίμηση της αλλαγής της λωρίδας με χρήση Νευρωνικών Δικτύων και μοντέλων μηχανικής μάθησης αντίστοιχα.

Δύο νευρωνικά δίκτυα εξετάστηκαν, ένα Μη Γραμμικό Αυτοπαλίνδρομο Νευρωνικό Δίκτυο (Nonlinear Autoregressive Neural Network, NARNN) καθώς και ένα Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης (Feed-Forward Neural Network, FFNN). Το NARNN χρησιμοποιεί ως είσοδο την ίδια μεταβλητή με αυτή που προσπαθεί να μαντέψει, την πλευρική θέση και η δυνατότητα της εκτίμησης μπορεί να είναι μόνο ένα βήμα μπροστά από την είσοδο. Αυτό σημαίνει πως για την εκτίμηση πολλών βημάτων μπροστά από την τωρινή είσοδο, θα πρέπει το Νευρωνικό Δίκτυο να χρησιμοποιήσει ως είσοδο την ίδια του την έξοδο. Αν π.χ το δίκτυο προσπαθεί να μαντέψει  $n$  φορές μπροστά, θα πρέπει η ίδια του η έξοδος να χρησιμοποιηθεί ως είσοδος στο ίδιο το δίκτυο  $n - 1$  φορές. Το FFNN μπορεί να πάρει εισόδους που δεν είναι απαραίτητα ίδιου τύπου με τις εξόδους του. Στη συγκεκριμένη υλοποίηση ως είσοδοι είναι η πλευρική θέση, η κατεύθυνση και η πλευρική ταχύτητα του οχήματος, ενώ ως εξόδους δίνεται η πλευρική θέση, η κατεύθυνση και η πλευρική ταχύτητα του οχήματος για 1 έως και 4 χρονικές στιγμές μετά την χρονική στιγμή των τιμών της εισόδου.

Η εκτίμηση της αλλαγής λωρίδας γίνεται με τη χρήση της τροχιάς του αυτοκινήτου, δηλαδή την έξοδο από τα Νευρωνικά Δίκτυα, ως είσοδο σε ένα SVM, του οποίου ο πυρήνας (kernel) είναι μια Συνάρτηση Ακτινικής Βάσης (Radial Basis Function, RBF). Αναλυτικότερα, η είσοδος αποτελείται από την πραγματική πλευρική θέση, την κατεύθυνση του οχήματος και την πλευρική ταχύτητα του οχήματος για τα προηγούμενα 2 δευτερόλεπτα, καθώς και τις εκτιμήσεις των μελλοντικών τιμών για αυτά τα δεδομένα που δίνουν ως έξοδο τα Νευρωνικά Δίκτυα.

Σε σύγκριση με τη βασική γραμμή αναφοράς, η εκτίμηση της μελλοντικής θέσης του οχήματος με τη χρήση του NARNN δεν φάνηκε να έχει κάποιο πλεονέκτημα, αντιθέτως η εκτίμηση με τη χρήση του FFNN κατάφερε να μειώσει το μέσο απόλυτο σφάλμα κατά ένα σημαντικό βαθμό, ειδικότερα κατά 23% στο σύνολο δεδομένων PKU και κατά 30% στο UAH αντίστοιχα μέχρι και 4 δευτερόλεπτα στο μέλλον. Αντίστοιχα, το SVM κατάφερε να εκτιμήσει με μεγάλο βαθμό αξιοπιστίας την αλλαγή της λωρίδας 3 δευτερόλεπτα πριν συμβεί.

### 2.2.3 Πιθανοτικές μέθοδοι

Οι Wang *et al* [17] ανέπτυξαν το "Μοντέλο Δυναμικής Καθοδηγούμενης Πρόθεσης" (Intention-Driven Dynamics Model, IDDM), ένα μοντέλο το οποίο αξιοποιεί πιθανοτικές μεθόδους για την εκτίμηση της πρόθεσης του ανθρώπου από την παρακολούθηση της ανθρώπινης κίνησης και της ανάλυσής της. Συγκεκριμένα, χρησιμοποιεί Γκαουσιανές Διεργασίες (Gaussian Process, GP) [18], οι οποίες έχουν χρησιμοποιηθεί ξανά με επιτυχία στη μοντελοποίηση της ανθρώπινης κίνησης. Στη συγκεκριμένη περίπτωση, μια ομάδα από GPs μοντελοποιούν τη συνάρτηση μετάβασης στον λανθάνων χώρο με βάση την πρόθεση. Μια δεύτερη ομάδα από GPs μοντελοποιούν την αντιστοίχιση των εκτιμώμενων καταστάσεων με τις πραγματικές καταστάσεις που λαμβάνονται από την παρατήρηση και καταγραφή.

Το μοντέλο αξιολογήθηκε παίζοντας επιτραπέζιο τένις εναντίον ανθρώπων με τη χρήση ενός ρομποτικού βραχίονα (Barrett WAM robot arm), με βασική γραμμή αναφοράς το Gaussian Process Regression (GPR) [18]. Συλλέχτηκαν δεδομένα καταγράφοντας την κίνηση από πολλούς ανθρώπους που χτυπούσαν το μπαλάκι με τη ρακέτα, καθώς και την κίνηση της μπάλας στο χώρο ούτως ώστε να καταγραφεί η πραγματική τροχιά της μπάλας και να εξαχθεί ο πραγματικός στόχος του ανθρώπου. Το IDDM κατάφερε να πετύχει μικρότερο μέσο απόλυτο σφάλμα από το GPR, με βελ-

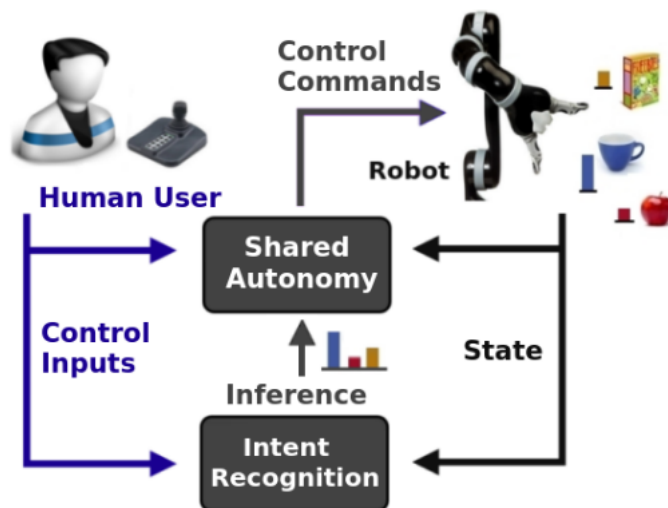
τίωση της τάξης του 11.3% με χρήση του batch αλγόριθμου, ενώ ο online αλγόριθμος πετυχαίνει βελτίωση της ακρίβειας κατά 8.5% σε σύγκριση με το GPR. Παρά τη βελτίωση του σφάλματος και στις δυο περιπτώσεις, αξίζει να σημειωθεί πως το συγκεκριμένο μοντέλο είναι υπολογιστικά πιο απαιτητική σε σύγκριση με το GPR. Αυτό όμως δεν το αποκλείει από τη χρήση του σε πραγματικές εφαρμογές, καθώς καταφέρνει να κάνει εκτιμήσεις μέσα σε χρονικά περιθώρια τα οποία παραμένουν αρκετά μικρά για τα πλαίσια αυτά.



Εικόνα 9. Σενάριο εκτίμησης της ανθρώπινης πρόθεσης στο επιτραπέζιο tennis. [17]

Οι Jiang *et al* [19] χρησιμοποίησαν μια κάμερα παρακολούθησης του βλέμματος του οδηγού και μια ακόμα κάμερα για την λήψη φωτογραφιών του περιβάλλοντος μπροστά από το αυτοκίνητο, λαμβάνοντας πληροφορία τόσο από το σημείο το οποίο κοιτάει ο οδηγός όσο και για τα σημεία ενδιαφέροντος (Points of Interest) από την εμπρόσθια κάμερα. Αξιοποιώντας ένα δυναμικό Markov Random Field, κατάφεραν να συσχετίσουν το σημείο που κοιτάει ο οδηγός με κάποιο από τα σημεία ενδιαφέροντος από την εμπρόσθια κάμερα με μεγαλύτερη επιτυχία από άλλες μεθόδους βασισμένες σε φίλτρα (Filter Based Approach) και αναζήτησης πλησιέστερου γείτονα (Nearest Neighbor Search).

Οι Jain *et al* [8] επιχείρησαν την εκτίμηση της πρόθεσης του χειριστή από το ρομπότ στα πλαίσια του διαμοιρασμένου ελέγχου/αυτονομίας (Shared Autonomy) με τη χρήση Αναδρομικού Μπεϋζιανού φιλτραρίσματος (Recursive Bayesian filtering), την οποία οι συγγραφείς ονόμασαν Αναδρομική Μπεϋζιανή Συμπερασματολογία Προθέσεων (Recursive Bayesian Intent Inference, RBII). Η συγκεκριμένη εφαρμογή οφείλει να γνωρίζει εξ αρχής όλους τους πιθανούς προσβάσιμους στόχους, όπου από αυτούς θα επιλέξει το στόχο με τη μεγαλύτερη πιθανότητα την οποία θεωρεί ότι ο άνθρωπος προσπαθεί να προσεγγίσει. Καθιστάται σαφές πως η εφαρμογή έχει μεριμνήσει ούτως ώστε η εκτίμηση του στόχου να μπορεί να αλλάξει με την πάροδο του χρόνου, καθώς υπάρχει η πιθανότητα να αλλάξει ο άνθρωπος στόχο κατά τη διάρκεια του πειράματος.



Εικόνα 10. Μικτή αυτονομία (Shared Autonomy). [8]

Η εφαρμογή είναι ικανή να λάβει ως είσοδο παρατηρήσεις (δεδομένα) από πολλές διαφορετικές πηγές, οι οποίες μεταξύ άλλων είναι εντολές ελέγχου όπως αυτές από ένα τηλεχειριστήριο ή ενδείξεις από το βλέμμα του ανθρώπου. Με τη χρήση αυτών των παρατηρήσεων εκτιμάται η κατανομή των πιθανοτήτων με βάση τον κανόνα του Bayes (Bayes' rule) για κάθε ένα από τους πιθανούς στόχους. Η πρόθεση του χειριστή θεωρείται ως ο στόχος που συγκεντρώνει τη μεγαλύτερη πιθανότητα. Μαζί με την εκτίμηση του στόχου εξάγεται και η αβεβαιότητα του συμπεράσματος, την οποία οι συγγραφείς αναφέρουν ως "αυτοπεποίθηση" (confidence), κάτι το οποίο μπορεί να χρησιμοποιηθεί ανάλογα με την περίπτωση για την εφαρμογή συγκεκριμένων συμπεριφορών από το ρομπότ.

Όσον αφορά την υλοποίηση του αλγόριθμου τους, εξέτασαν δυο συστήματα παρατηρήσεων. Το πρώτο (RBII-1) λαμβάνει ως μοναδικό χαρακτηριστικό την Ευκλείδεια απόσταση μεταξύ της θέσης του ρομπότ και της θέσης του στόχου, ενώ το δεύτερο (RBII-2) εκτός από την Ευκλείδεια απόσταση λαμβάνει ως δεύτερο χαρακτηριστικό και τις ενέργειες του ανθρώπινου παράγοντα. Εκτός από τα δυο συστήματα παρατηρήσεων, ακολούθησαν και δυο διαφορετικές προσεγγίσεις εκτίμησης: Τη Συμπερασματολογία της Αμνησίας (Amnesic Inference) και τη Συμπερασματολογία βάσει μνήμης (Memory-based Inference). Στην πρώτη, θεωρείται ως πιθανότερος στόχος αυτός ο οποίος είναι πλησιέστερος στο ρομπότ, ενώ στη δεύτερη προσέγγιση λαμβάνεται υπόψιν του το ιστορικό μιας τροχιάς για την εκτίμηση του πιθανότερου στόχου.

Για την αξιολόγηση της εφαρμογής, έγιναν πειράματα όπου ελέγχθηκε με τις τέσσερις προαναφερθείσες μεθόδους (Amnesic Inference, Memory-based Inference) σε σύγκριση με την ανθρώπινη ικανότητα εκτίμησης της πρόθεσης αξιοποιώντας ένα ρομποτικό βραχίονα και διάφορα αντικείμενα τα οποία ήταν οι πιθανοί στόχοι. Για την ανθρώπινη ικανότητα, το ρομπότ πραγματοποίησε καταγεγραμμένες τροχιές προς τα αντικείμενα, με σκοπό οι άνθρωποι που έλαβαν μέρος στο τεστ να εκτιμήσουν το αντικείμενο που προσπαθεί να φτάσει το ρομπότ με βάση την τροχιά του. Στη συνέχεια, πραγματοποιήθηκαν πειράματα όπου το ρομπότ χειριζόντουσαν οι άνθρωποι που έλαβαν μέρος στο τεστ. Έγιναν οι απαραίτητες δοκιμές τόσο για το Amnesic Inference όσο και για το Memory-based Inference χωρίς βοήθεια από το ρομπότ, ενώ στη συνέχεια επαναλήφθηκαν με βοήθεια από το ρομπότ κάνοντας χρήση πρώτα του RBII-1 και μετά του RBII-2. Για την κάθε προσπάθεια, ζητήθηκε από τους ανθρώπους να αλλάξουν τον στόχο τους κατά την εκτέλεση του πειράματος. Από τα πειράματα χωρίς τη βοήθεια του ρομπότ αποδείχτηκε ότι η Memory-based Inference μέθοδος μπόρεσε να συμπεράνει τον επιδιωκόμενο στόχο νωρίτερα από τις άλλες μεθόδους στην περίπτωση που δεν άλλαζε κατά την εκτέλεση του πειράματος, ενώ όταν άλλαζε το RBII-2 ήταν καλύτερο στο να προσδιορίζει με μεγαλύτερη σιγουριά το σωστό στόχο. Στα πειρά-

ματα με βοήθεια από το ρομπότ τα RBII-1 και RBII-2 ήταν σημαντικά πιο γρήγορα στην ολοκλήρωση του έργου και με λιγότερες εναλλαγές στη λειτουργία ελέγχου σε σύγκριση με τον απλό τηλεχειρισμό, ενώ το RBII-2 ήταν και αρκετά πιο γρήγορο από το RBII-1. Επίσης, στον απλό τηλεχειρισμό χωρίς βοήθεια υπήρχαν περισσότερες εναλλαγές σε σύγκριση με το RBII-1 και το RBII-2, ενώ το RBII-2 είχε λιγότερες εναλλαγές από το RBII-1, αλλά η διαφορά δεν ήταν σημαντική.

Οι Panagopoulos *et al* [9] ανέπτυξαν το μοντέλο εκτίμησης της πρόθεσης του χειριστή "Αναγνώριση Πρόθεσης Χειριστή κατά Μπέυζ (Bayesian Operator Intent Recognition, BOIR). Το μοντέλο κάνει χρήση του Συμπεράσματος του Μπέυζ το οποίο αξιοποιεί με τη σειρά του τρία επιπλέον μοντέλα: ένα μοντέλο "παρατήρησης" (observation model), ένα μοντέλο "μετάβασης" (transition model) και ένα μοντέλο "δράσης" (action model, AIRM). Το μοντέλο "παρατήρησης" λαμβάνει δυο πηγές πληροφοριών για έναν από όλους τους προκαθορισμένους στόχους (goal) οι οποίοι είναι γνωστοί εξ αρχής: την γωνία μεταξύ του προσανατολισμού του ρομπότ και του στόχου, καθώς και την απόσταση που δίνεται από ένα "σχεδιαστή διαδρομής" (path planner). Το μοντέλο "μετάβασης" επιτρέπει την παροχή πληροφοριών σχετικά με την προβλεπόμενη/τρέχουσα κατάσταση δεδομένης της αμέσως προηγούμενης. Το μοντέλο "δράσης" επιτρέπει στο BOIR να λάβει ασύγχρονα και κατά απαίτηση ρητή πληροφορία για την πρόθεση του χειριστή όταν και εφόσον είναι διαθέσιμη αυτή. Οι συγγραφείς σημειώνουν πως αυτό γίνεται μέσω μιας γραφικής διεπαφής χρήστη, το οποίο αντιστοιχεί σε σημάδι προς τον επιδιωκόμενο στόχο.

Το BOIR-AIRM (BOIR με το μοντέλο "δράσης") συγκρίθηκε με το απλό BOIR (χωρίς τη χρήση του μοντέλου "δράσης"), καθώς και με το μοντέλο RBII-1 από τη δουλειά των Jain *et al* [8] και με τη μέθοδο Εκθετικής Συνάρτησης Βεβαιότητας (Exponential Confidence Function, ECF) από τους Carlson *et al* [20]. Εξετάστηκαν τέσσερα σενάρια συνολικά, όπου στο πρώτο σενάριο οι χειριστές έπρεπε να κινήσουν το ρομπότ από την αρχική του θέση στο στόχο "b", ενώ στη διάρκεια της κίνησης να αλλάξουν το στόχο τους προς το "a". Στο πρώτο σενάριο δεν υπήρχαν εμπόδια μεταξύ του ρομπότ και των στόχων. Το δεύτερο σενάριο ήταν παρόμοιο με το πρώτο, με τη διαφορά ότι για να προσεγγίσει τους στόχους το ρομπότ έπρεπε να παρακάμψει κάποια εμπόδια που βρίσκονταν στο δρόμο του, ενώ ο στόχος ήταν η προσέγγιση του στόχου "c". Στο τρίτο σενάριο το ρομπότ έπρεπε να πλοηγηθεί διαδοχικά από την αρχική θέση στην "a", στη συνέχεια στη "b" και τέλος να καταλήξει στη "c". Κάθε φορά που το ρομπότ έφτανε σε έναν στόχο, μέτραγε ως αλλαγή στην πρόθεση του χειριστή. Στο τέταρτο και τελευταίο σενάριο ο χειριστής του ρομπότ έπρεπε να πλοηγηθεί διαδοχικά σε δυο τυχαία διαλεγμένους στόχους από ένα πλήθος στόχων που ήταν ήταν σημαντικά μεγαλύτερος από τα προηγούμενα σενάρια. Στο τέταρτο σενάριο το περιβάλλον δεν είχε εμπόδια μεταξύ του ρομπότ και των στόχων, όπως και στο πρώτο σενάριο.

Το BOIR κατάφερε να βρει τον σωστό στόχο με μεγαλύτερη ορθότητα σε σύγκριση με το μοντέλο RBII-1 και τη μέθοδο ECF σε όλες τις δοκιμές εκτός μιας, όπου μεταξύ του ρομπότ και των στόχων δεν υπήρχαν εμπόδια για να χρειαστεί το ρομπότ να κάνει κάποια παράκαμψη προκειμένου να τους προσεγγίσει. Αξίζει να σημειωθεί ότι το BOIR-AIRM, ξεπέρασε σε αποτελεσματικότητα όλα τα άλλα μοντέλα και μεθόδους, μιας και ο χειριστής μπορούσε να επικοινωνήσει ρητά την πρόθεσή του στο μοντέλο.



## Κεφάλαιο 3

# Μεθοδολογία ή/και Σχέδιο εργασιών – Υλοποίηση – Εφαρμογή

Στο κεφάλαιο αυτό θα εξηγηθεί η δομή και η υλοποίηση των λειτουργιών που συνθέτουν το σύστημα εντοπισμού των σημείων ενδιαφέροντος, την τοποθέτησή τους στο χώρο καθώς και την εκτίμηση της πρόθεσης του χειριστή από το μοντέλο μηχανικής μάθησης. Ονομάζουμε και θα αναφερόμαστε στο συγκεκριμένο μοντέλο μηχανικής μάθησης ως MLOII (Machine Learning Operator Intent Inference).

### 3.1 Robot Operating System - ROS

Το Robot Operating System (ROS) [21] είναι ένα πακέτο ανάπτυξης λογισμικού ανοιχτού κώδικα για την κατασκευή ρομποτικών εφαρμογών, που προορίζεται για χρήση σε όλα τα επίπεδα, από την έρευνα και την εκπαίδευση έως τα εμπορικά και βιομηχανικά περιβάλλοντα. Το ROS έχει σχεδιαστεί με γνώμονα την εύκολη πρόσβαση πόρων υλισμικού (hardware) μέσω διεπαφών καθώς και τη δυνατότητα ελέγχου συσκευών χαμηλού επιπέδου με γλώσσες προγραμματισμού υψηλού επιπέδου, παρέχοντας υπηρεσίες για τις διάφορες διαδικασίες ανταλλαγής δεδομένων μεταξύ τους. Παρέχει πολλά εργαλεία και βιβλιοθήκες για την ανάπτυξη, τη δοκιμή και την οπτικοποίηση (visualisation) εφαρμογών, καθώς και τη διαχείριση πακέτων λογισμικού για την επίλυση των εξαρτήσεων.

Παρά το ότι μπορεί να υποδηλώνει το όνομα, το ROS δεν είναι πραγματικό λειτουργικό σύστημα καθώς δεν παρέχει τις συνήθεις υπηρεσίες που παρέχει ένα λειτουργικό σύστημα. Αυτό ουσιαστικά σημαίνει ότι το ROS βασίζεται σε ένα υποκείμενο λειτουργικό σύστημα για αυτή τη λειτουργικότητα. Συνήθως αξιοποιούνται τα Linux με πειραματικές εκδόσεις macOS και Windows υπό ανάπτυξη τη στιγμή της συγγραφής της διατριβής.

Μερικά από τα πλεονεκτήματα της χρήσης του ROS:

- Επίσημη υποστήριξη για τις γλώσσες προγραμματισμού C++ και Python, με βιβλιοθήκες που επιτρέπουν τη χρήση και άλλων γλωσσών. Εάν δεν υπάρχει βιβλιοθήκη για μια συγκεκριμένη γλώσσα, υπάρχει η δυνατότητα εγκατάστασης του `roslaunch` και να το χρησιμοποιεί, αρκεί να μπορεί η γλώσσα να μπορεί να καταλάβει JSON.
- Ισχυρή υποστήριξη από την κοινότητα του ROS, με πολλά πακέτα διαθέσιμα για τις περισσότερες λειτουργίες των εφαρμογών ρομποτικής, όπως αλγόριθμοι ταυτόχρονου εντοπισμού και χαρτογράφησης (SLAM), τηλεχειρισμού, μηχανική όταση και χειρισμός.
- Περιβάλλον προσομοίωσης για δοκιμή της εφαρμογής πριν από την εγκατάσταση σε πραγματικό ρομπότ. Παρέχονται εργαλεία για τη δημιουργία ενός μοντέλου με προσαρμοσμένα

χαρακτηριστικά, ή χρήση έτοιμων μοντέλων από κατασκευαστές ρομπότ, καθώς οι περισσότεροι από αυτούς παρέχουν πακέτα για την προσομοίωση των ρομπότ τους μέσα στο περιβάλλον προσομοίωσης του ROS.

- Το ROS είναι ανοιχτού κώδικα, με χρήστες από όλο τον κόσμο να συνεισφέρουν σε αυτό, διασφαλίζοντας τη συνεχή ανάπτυξη και βελτιώσεις.

Το ROS κυκλοφορεί σε διανομές, όπως και οι εκδόσεις του δημοφιλούς λειτουργικού συστήματος Ubuntu, το οποίο βασίζεται στον πυρήνα (kernel) του Linux. Οι διανομές ROS είναι μια συλλογή πακέτων λογισμικού συγκεκριμένων εκδόσεων (versions). Σκοπός αυτού είναι οι προγραμματιστές να έχουν πρόσβαση σε μια μεγάλη και σταθερή βάση κώδικα μέχρι να είναι έτοιμοι να αναβαθμίσουν σε μια νεότερη διανομή. Αυτές οι διανομές συνοδεύονται από ημερομηνία λήξης της υποστήριξής τους (EOL) και αυτή η ημερομηνία εξαρτάται από τη διανομή. Οι εκδόσεις μακροπρόθεσμης υποστήριξης (LTS) μπορούν να έχουν ενημερώσεις έως και πέντε χρόνια μετά από την έκδοσή του.

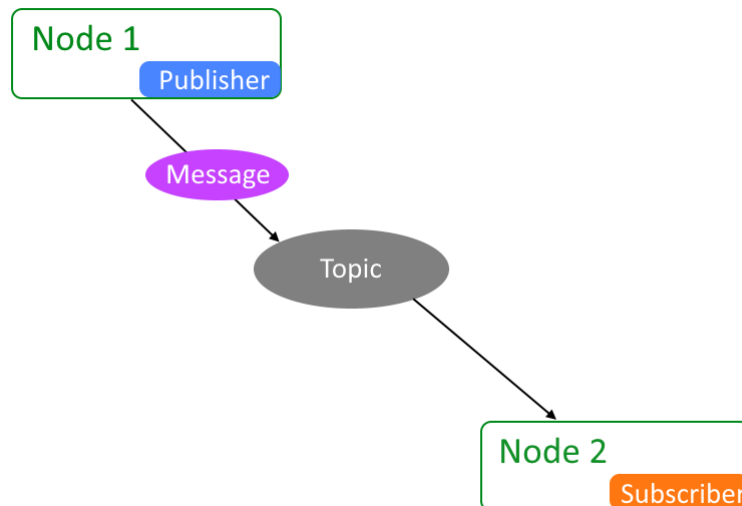
Στη συνέχεια θα συζητηθούν ορισμένες θεμελιώδεις έννοιες του ROS:

### 3.1.1 Nodes

Τα nodes στο ROS μπορούν να θεωρηθούν ως απλές διεργασίες που εκτελούν υπολογισμούς. Κάθε node είναι υπεύθυνο για μία μόνο πτυχή της λειτουργικότητας του ρομπότ. Για παράδειγμα, ένα node μπορεί να είναι υπεύθυνο για την κίνηση της βάσης του ρομπότ μέσα στο περιβάλλον, ενώ ένα άλλο node μπορεί να είναι υπεύθυνο για την αντίληψη του περιβάλλοντος του ρομπότ καθώς αυτό πλοηγείται μέσα σε αυτό. Αυτές οι διεργασίες μπορούν να εκτελούνται παράλληλα μεταξύ τους και δεν υπάρχει περιορισμός ως προς το πόσες διεργασίες μπορούν να είναι ενεργές σε μια δεδομένη στιγμή.

Ένα ειδικού τύπου node είναι το master node, το οποίο παρέχει υπηρεσίες εγγραφής (registration services) καθώς και ονοματοδοσίας σε κάθε άλλο node στο ROS. Το master node παρακολουθεί τους publishers και τους subscribers των topics καθώς και τις υπηρεσίες. Για παράδειγμα, ένα node που δημοσιεύει εικόνες σε ένα topic α θα ενημερώσει το master node για την ύπαρξή του, καθώς και για το όνομα του topic. Ένα άλλο node κάποια στιγμή αργότερα μπορεί να ειδοποιήσει το master node ότι θέλει να κάνει subscribe στο ίδιο topic, για να αποκτήσει πρόσβαση στις εικόνες. Σε αυτό το σημείο, το publisher node και το subscriber node δεν γνωρίζουν το ένα την ύπαρξη του άλλου. Μόλις το master node δει ότι υπάρχει και publisher και subscriber στο ίδιο topic, ειδοποιεί και τα δύο για την ύπαρξη του άλλου node και αρχίζει η ανταλλαγή εικόνων μεταξύ των publisher και subscriber nodes.

Γενικά, θεωρείται καλή πρακτική ο κώδικας να χωρίζεται σε διαφορετικά nodes ανάλογα με τη λειτουργικότητα. Εάν ένα node κάνει crash, τα υπόλοιπα nodes θα συνεχίσουν να είναι ενεργά, διασφαλίζοντας ότι τουλάχιστον τμήματα της λειτουργικότητας θα παραμείνουν ενεργά. Ένα άλλο πλεονέκτημα της χρήσης πολλών μικρών nodes είναι η ευκολία αποσφαλμάτωσης, οι ταχύτεροι χρόνοι ανάπτυξης και η δυνατότητα επαναχρησιμοποίησης του κώδικα. Φυσικά, αυτά τα nodes πρέπει συχνά να ανταλλάσσουν πληροφορίες, και αυτό γίνεται μέσω topics και messages.



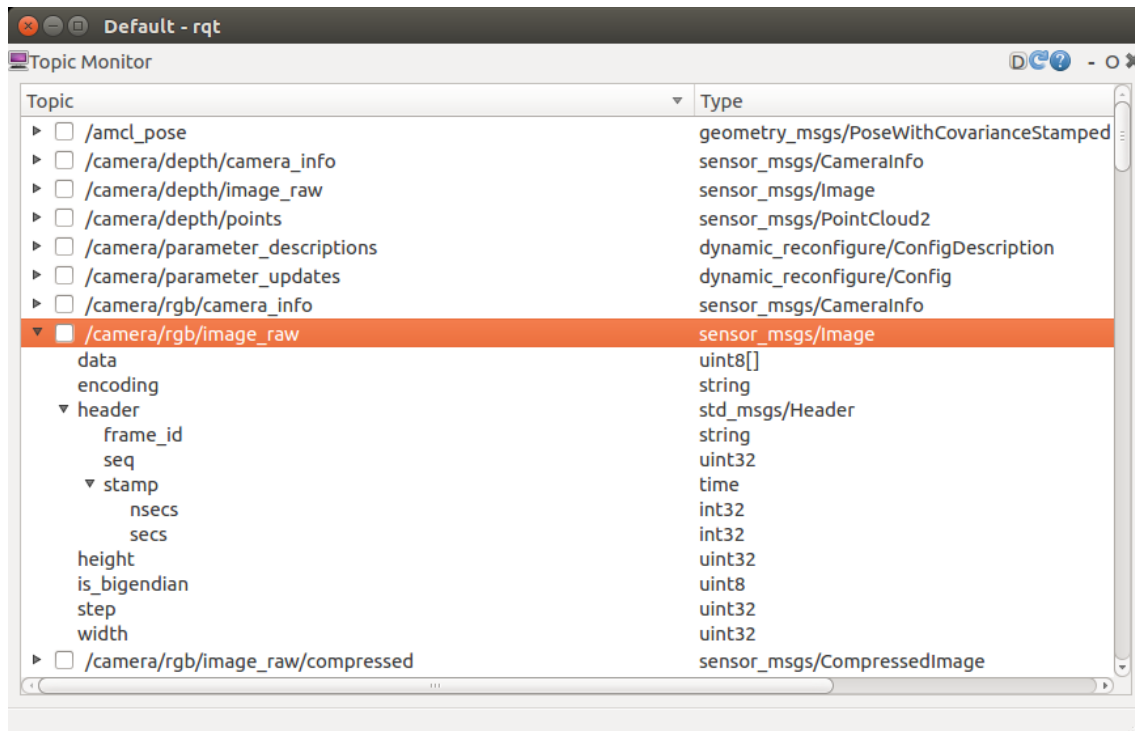
Εικόνα 11. Οι σχέσεις μεταξύ των ROS Nodes, Topics και Messages

### 3.1.2 Topics

Η ασύγχρονη επικοινωνία μεταξύ των nodes επιτυγχάνεται με τη χρήση των topics και των messages. Τα topics είναι δίαυλοι με ονομασία πάνω στους οποίους τα nodes ανταλλάσσουν messages. Τα nodes χρησιμοποιούν συνήθως το μοντέλο publish - subscribe, έτσι ξεχωρίζεται η παραγωγή πληροφοριών από την κατανάλωσή τους. Σε αυτό το μοντέλο, τα διάφορα nodes δεν γνωρίζουν με ποιον επικοινωνούν. Ένα node που ενδιαφέρεται για δεδομένα κάνει subscribe στο σχετικό topic, ενώ ένα node που παράγει δεδομένα κάνει publish στο σχετικό topic. Όταν το node κάνει publish ένα message στο συγκεκριμένο topic, όλα τα άλλα nodes που είναι εγγεγραμμένα στο συγκεκριμένο topic λαμβάνουν το message που δημοσιεύθηκε. Ένα node δεν έχει περιορισμούς ως προς το σε πόσα topics μπορεί να κάνει subscribe ή publish. Ταυτόχρονα, ένα topic δεν έχει κανέναν περιορισμό ως προς το πόσοι publishers/subscribers κάνουν publish/subscribe σε αυτό. Ένα χρήσιμο εργαλείο για να δει κάποιος όλα τα ενεργά topics στο ROS καθώς και τους τύπους των messages είναι το rqt που παρέχεται από το ROS.

### 3.1.3 Messages

Τα messages είναι δομές δεδομένων που περιέχουν πεδία συγκεκριμένου τύπου. Ένα πεδίο μπορεί να είναι ενός πρωτόγονου τύπου όπως ακέραιος (integer), κινητής υποδιαστολής (floating-point), δυαδικής τιμής (boolean) καθώς και πίνακες πρωτόγονων τύπων. Άλλοι τύποι που μπορούν να είναι τα πεδία είναι συμβολοσειράς (string) καθώς και άλλα .msg τα οποία αποτελούνται με τη σειρά τους από δικά τους πεδία. Το ROS παρέχει έναν τρόπο για τον ορισμό προσαρμοσμένων messages μέσω της χρήσης αρχείων .msg



Εικόνα 12. Ενεργά topics του ROS και ο τύπος των messages μέσω του εργαλείου rqt που παρέχει το ROS

Πίνακας 1. Τα είδη των πεδίων που είναι διαθέσιμα για τα messages

Message Field Types				
Primitive Type	Serialization	C++	Python2	Python3
bool	unsigned 8-bit int	uint8_t	bool	
int8	signed 8-bit int	int8_t	int	
uint8	unsigned 8-bit int	uint8_t	int	
int16	signed 16-bit int	int16_t	int	
uint16	unsigned 16-bit int	uint16_t	int	
int32	signed 32-bit int	int32_t	int	
uint32	unsigned 32-bit int	uint32_t	int	
int64	signed 64-bit int	int64_t	long	int
uint64	unsigned 64-bit int	uint64_t	long	int
float32	32-bit IEEE float	float	float	
float64	64-bit IEEE float	double	float	
string	ascii string	std::string	str	bytes
time	secs/nsecs unsigned 32-bit ints	ros::Time	rospy.Time	
duration	secs/nsecs signed 32-bit ints	ros::Duration	rospy.Duration	

### 3.1.4 Rosbag

Το Rosbag είναι ένα πακέτο με μια σειρά από εργαλεία που παρέχονται από το ROS για την εγγραφή και αναπαραγωγή topics του ROS. Με τη λειτουργία της καταγραφής (record), το Rosbag μπορεί να καταγράψει τα messages που γίνονται publish σε κάποια επιλεγμένα ή σε όλα τα topics που είναι ενεργά εκείνη τη στιγμή, παράγοντας ένα αρχείο .bag. Το αρχείο αυτό με τη λειτουργία

της αναπαραγωγής (play), κάνει publish όλα τα messages στα αντίστοιχα topics με τη σειρά που καταγράφηκαν.

### 3.1.5 Packages

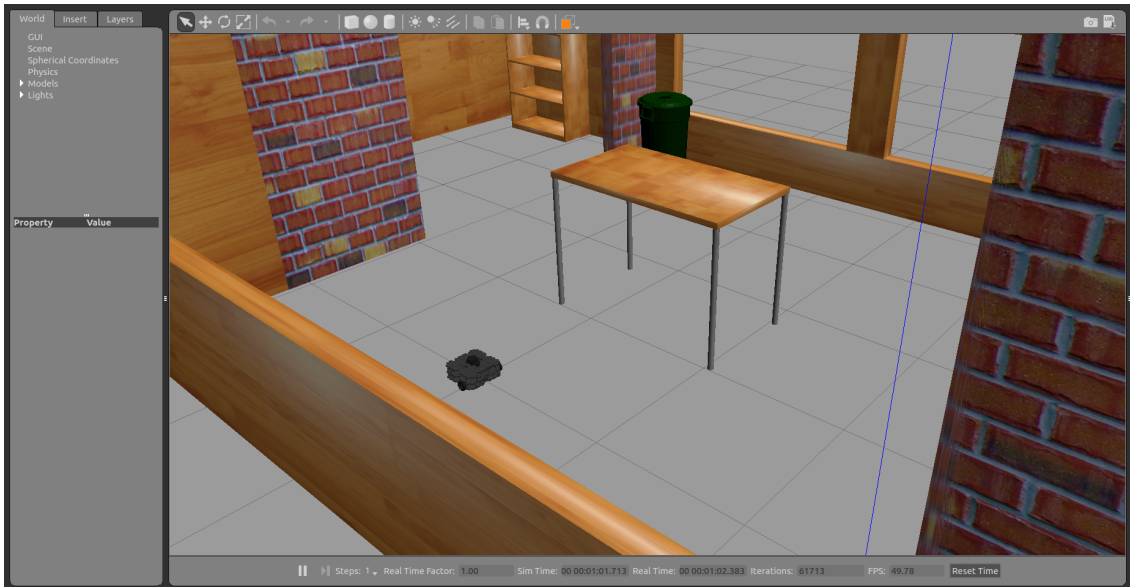
Στο ROS, το λογισμικό είναι οργανωμένο σε πακέτα. Ένα πακέτο μπορεί να περιέχει κόμβους ROS, βιβλιοθήκες, σύνολα δεδομένων, αρχεία ρυθμίσεων, λογισμικό τρίτων, άλλων ειδών αρχεία και οτιδήποτε μπορεί να χρειάζεται για τη λειτουργικότητά του. Τα πακέτα ROS μπορούν να δημιουργηθούν είτε με το χέρι είτε με τη χρήση του εργαλείου `catkin_create_pkg` που παρέχεται από το `catkin`.

### 3.1.6 Catkin

Το Catkin είναι το επίσημο build system για το ROS και ο διάδοχος του `roscpp`, του πρώτου build system για το ROS. Ο κύριος σκοπός του είναι η μεταγλώττιση (compilation) και η δημιουργία (build) των εκτελέσιμων προγραμμάτων από τον πηγαίο κώδικα. Ουσιαστικά, μπορεί να χτίσει τις βιβλιοθήκες που του έχουμε ορίσει ότι θέλουμε να χτίσει καθώς και τα εκτελέσιμα αρχεία, scripts κ.λπ. Ο τρόπος με τον οποίο το Catkin δημιουργεί τα εκτελέσιμα αρχεία παρομοιάζει το CMake, ένα άλλο δημοφιλές build system, αλλά με κάποια πρόσθετη λειτουργικότητα για την επίλυση εξαρτήσεων των πακέτων. Για παράδειγμα, ένα πακέτο μπορεί να εξαρτάται από ένα άλλο για τη δημιουργία του και αυτή η εξάρτηση απαιτεί το build του δεύτερου πακέτου πριν από το πρώτο. Το Catkin μπορεί να αποφασίσει την ιεραρχία της δομής του έργου (project) και να το χτίσει με τη σωστή σειρά.

### 3.1.7 Gazebo

Το Gazebo είναι μια προσομοίωση κόσμου (world simulation) που είναι διαθέσιμη για χρήση στο ROS και αποτελεί ένα από τα πιο βασικά εργαλεία για τη δοκιμή και την αποσφαλμάτωση λογισμικού σε ένα ασφαλές περιβάλλον. Διαθέτει μια κομψή και εύχρηστη διεπαφή χρήστη που επιτρέπει τη δημιουργία περιβαλλόντων (κόσμων) και ρομποτικών συστημάτων. Το Gazebo βρίσκεται στο `gazebo_ros_pkgs` το οποίο δεν αποτελεί βασικό πακέτο του ROS και πρέπει να εγκατασταθεί ξεχωριστά. Το πακέτο αυτό εριέχει πολλές διεπαφές για την προσομοίωση ενός ρομπότ, καθώς και πολλούς προσομοιωμένους αισθητήρες που είναι άμεσα διαθέσιμοι και έτοιμοι για χρήση. Στην παρούσα διατριβή, το Gazebo χρησιμοποιήθηκε για την ανάπτυξη της λειτουργικότητας του ρομπότ καθώς και για δοκιμές.



Εικόνα 13. Το περιβάλλον εξομοίωσης του Gazebo

### 3.1.8 Husky

Το Husky (βλ. Εικ. 14) πρόκειται για μια πλατφόρμα ανάπτυξης ρομπότ μεσαίου μεγέθους. Χαρακτηρίζεται από τη στιβαρή του κατασκευή και ένα σύστημα μετάδοσης κίνησης υψηλής ροπής που του επιτρέπει να πλοηγηθεί ακόμη και σε δύσβατες περιοχές [22]. Ακόμη, μπορεί να προσαρμοστεί με πολλών ειδών αισθητήρες όπως στερεοκάμερες, LiDAR, GPS, IMU αλλά και με άλλων ειδών εξαρτήματα όπως ρομποτικοί βραχίονες. Αυτό το καταφέρνει χάρη τη μεγάλη του χωρητικότητα ωφέλιμου φορτίου και τα συστήματα ισχύος. Ταυτόχρονα, η Clearpath Robotics υποστηρίζει την εξομοίωση του Husky, προσφέροντας επίσης κώδικα και παραδείγματα από την κοινότητά του ROS. Στα πλαίσια της διπλωματικής εργασίας, χρησιμοποιείται το εξομοιωμένο μοντέλο του Husky στο περιβάλλον Gazebo του ROS.



Εικόνα 14. Η ρομποτική πλατφόρμα Husky [22]

## 3.2 Kinect Sensor

Το Kinect (βλ. Εικ. 15) είναι μια σειρά συσκευών εισόδου με αισθητήρα κίνησης που παράγονται από τη Microsoft και κυκλοφόρησαν για πρώτη φορά το 2010. Οι συσκευές περιέχουν γενικά κάμερες RGB μαζί με υπέρυθρους προβολείς και ανιχνευτές που χαρτογραφούν το βάθος είτε μέσω δομημένου φωτός είτε μέσω του χρόνου πτήσης (time of flight, ToF). Οι πληροφορίες αυτές μπορούν με τη σειρά τους να χρησιμοποιηθούν για την αναγνώριση χειρονομιών σε πραγματικό χρόνο και την ανίχνευση του σκελετού σώματος, μεταξύ άλλων δυνατοτήτων. Περιέχουν επίσης μικρόφωνα που μπορούν να χρησιμοποιηθούν για αναγνώριση ομιλίας και φωνητικό έλεγχο.

Η σειρά των συσκευών Kinect δημιουργήθηκαν αρχικά ως χειριστήρια κίνησης για τις κονσόλες βιντεοπαιχνιδιών Xbox. Στη συνέχεια, ο αισθητήρας άρχισε να χρησιμοποιείται και σε εφαρμογές εκτός των παιχνιδιών, σε ακαδημαϊκά και εμπορικά περιβάλλοντα, καθώς ήταν μια φθηνότερη και πιο εύρωστη επιλογή σε σύγκριση με άλλες τεχνολογίες ανίχνευσης βάθους εκείνη την εποχή.

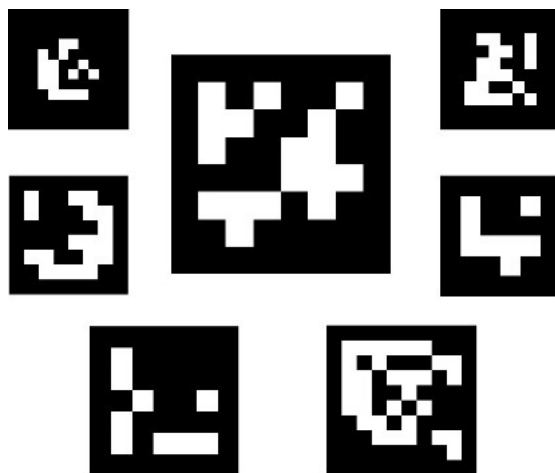
Στα πλαίσια της διπλωματικής εργασίας, το Kinect V2 και οι λειτουργίες του εξομοιώθηκαν με τη χρήση ενός πακέτου που διατίθεται για χρήση στο ROS και το Gazebo.



Εικόνα 15. Ο αισθητήρας Kinect V2 της Microsoft

### 3.2.1 Aruco Markers

Τα Aruco markers (βλ. Εικ. 16)[23] [24] είναι δείκτες αναφοράς που χρησιμοποιούνται σε εικόνες για χρήση ως σημείο αναφοράς ή μέτρο κατά την εκτίμηση της θέσης της κάμερας σε εφαρμογές επαυξημένης/εικονικής πραγματικότητας (augmented/virtual reality) καθώς και σε εφαρμογές ρομποτικής. Τα markers προτιμώνται έναντι φυσικών χαρακτηριστικών, όπως σημεία-κλειδιά ή υφές, λόγω του ότι είναι πολύ εύκολο να εντοπιστούν και παρέχουν υψηλό επίπεδο ακρίβειας.



Εικόνα 16. Παραδείγματα Aruco markers από διάφορα dictionaries

Ένα Aruco marker μπορεί να θεωρηθεί ως ένα πλέγμα δυαδικών ψηφίων, με μαύρο περίγραμμα περιμετρικά. Τα λεξικά (dictionaries) είναι το σύνολο των Aruco markers που εξετάζονται σε μια συγκεκριμένη εφαρμογή. Ουσιαστικά είναι το σύνολο των δυαδικών κωδικοποιήσεων κάθε ενός από τους δείκτες του. Οι ιδιότητες ενός λεξικού είναι το μέγεθος του λεξικού και το μέγεθος του marker. Το μέγεθος του λεξικού είναι το πλήθος των markers που συνθέτουν το λεξικό και το μέγεθος των markers είναι ο αριθμός των bits που κωδικοποιούν χρήσιμη πληροφορία. Η βιβλιοθήκη υποστηρίζει τον ορισμό προσαρμοσμένων λεξικών που δεν έχουν τετράγωνο μέγεθος,

αλλά για το πλαίσιο της παρούσας διπλωματικής εργασίας εξετάζονται μόνο οι προκαθορισμένες βιβλιοθήκες.

Το ωφέλιμο φορτίο είναι η πληροφορία που κωδικοποιείται στο εσωτερικό του Aruco marker και έχει μέγεθος  $n^2$ , εάν  $n$  είναι το πλάτος του πλέγματος του ωφέλιμου φορτίου και το marker είναι τετράγωνο. Το συνολικό μέγεθος του Aruco marker είναι  $(n + 2)^2$ , λαμβάνοντας υπόψη το μαύρο πλαίσιο που περιβάλλει το payload. Το μαύρο περίγραμμα που περιβάλλει το ωφέλιμο φορτίο δεν κωδικοποιεί καμία πληροφορία, ωστόσο εξακολουθεί να είναι απαραίτητο για τη λειτουργία του αλγόριθμου ανίχνευσης που παρέχει η βιβλιοθήκη Aruco. Εάν οποιοδήποτε από τα bits που αποτελούν το περίγραμμα δεν είναι 0 (μαύρο), ο αλγόριθμος παραλείπει την ανάγνωση του ωφέλιμου φορτίου και απορρίπτει το marker.

Στην παρούσα διατριβή, τα Aruco markers χρησιμοποιήθηκαν ως σημεία ενδιαφέροντος που ανιχνεύονται στο χώρο, των οποίων οι θέσεις χρησιμοποιούνται για την άντληση χρήσιμων πληροφοριών, όπως η απόστασή τους από την κάμερα του ρομπότ και τη γωνία τους από την οπτική γωνία του ρομπότ.

### 3.3 Περιγραφή των λειτουργιών που αναπτύχθηκαν ως nodes

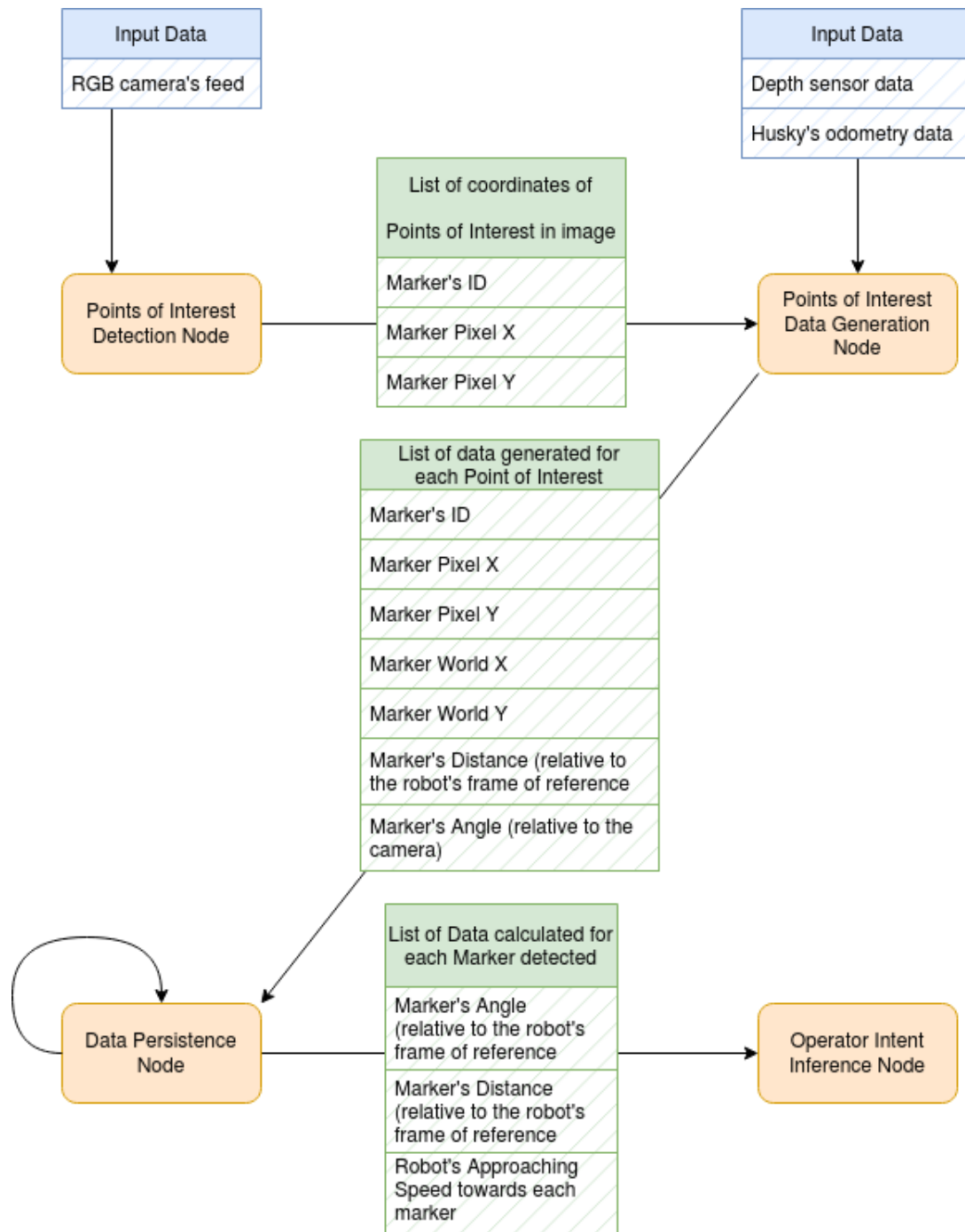
Παρακάτω θα παρουσιαστεί αναλυτικά η λειτουργία του κάθε node που αναπτύχθηκε με σκοπό την αναγνώριση των εμπιστευτικών σημάτων "Aruco Markers", τον υπολογισμό των θέσεων τους στο χώρο καθώς και πληροφορίες σχετικά με την κίνηση του ρομπότ και την αξιοποίησή τους για την εκτίμηση της πρόθεσης του χειριστή <sup>1</sup>.

#### Points of Interest Detection Node

Το Points of Interest Detection Node (POIDN) (βλ. Πηγ. Κωδ. 6.1.1) κάνει subscribe στο topic της κάμερας που έχει τοποθετηθεί επάνω στο ρομπότ "/camera/rgb/image\_raw". Από αυτό λαμβάνει κάθε καρέ της κάμερας και σε κάθε καινούριο καρέ που λαμβάνει, τρέχει την ρουτίνα imageCallback. Η ρουτίνα αυτή καλεί τη μέθοδο detectMarkers από τη βιβλιοθήκη των Aruco markers, προκειμένου να εντοπίσει τα markers στην εικόνα. Από αυτή τη μέθοδο παίρνουμε πίσω μια λίστα από τα ID's των markers που εντοπίζονται στο καρέ, καθώς και τις τέσσερις γωνίες τους μέσα στην εικόνα. Τις πληροφορίες αυτές τις γεμίζουμε στο message τύπου "marker\_collection" και τις κάνουμε publish στο topic "aruco/markers\_loc".

<sup>1</sup>Ολόκληρο το πακέτο λογισμικού που αναπτύχθηκε για τη διπλωματική εργασία μπορεί να βρεθεί στο σύνδεσμο: [https://github.com/EvangelosTsagkournis/operator\\_intent](https://github.com/EvangelosTsagkournis/operator_intent)





Εικόνα 17. Η δομή των υπολειτουργιών για τον εντοπισμό των σημείων ενδιαφέροντος και της εκτίμησης της πρόθεσης του χειριστή



Εικόνα 18. Οπτικοποίηση του εντοπισμού του Aruco Marker με ID = 0 από το POIDN μέσω της μεθόδου drawDetectedMarkers

### Points of Interest Data Generation Node

Το Points of Interest Data Generation Node (POIDGN) (βλ. Πηγ. Κωδ. 6.1.2) κάνει subscribe στο topic "aruco/markers\_loc" ούτως ώστε να λάβει τις τοποθεσίες των markers που κάνει publish το POIDN. Ταυτόχρονα κάνει subscribe και στα topics "camera/depth/image\_raw" και "husky\_base\_ground\_truth" για την "εικόνα βάθους" (depth image) και την οδομετρία (odometry) του ρομπότ. Η εικόνα βάθους (βλ. Εικ. 19) είναι ένας τρόπος οπτικοποίησης της απόστασης των αντικειμένων μπροστά από την κάμερα. Η μέθοδος callback συγχρονίζει τα δεδομένα από τα τρία topics που αναφέραμε προηγουμένως, και μόλις υπάρχουν δεδομένα διαθέσιμα και από τα τρία, βρίσκει την κατεύθυνση / γωνία του ρομπότ με σύστημα αναφοράς τον κόσμο του Gazebo. Μετά, για κάθε marker που έχει εντοπιστεί, βρίσκει το κέντρο του από το σημείο που τέμνονται οι τέσσερις γωνίες του. Για αυτό το σημείο ανατρέχει στην εικόνα βάθους και λαμβάνει μια μέτρηση για την απόσταση του. Ταυτόχρονα, υπολογίζει και τη γωνία υπό την οποία η κάμερα βλέπει το σημείο ενδιαφέροντος. Αυτή η γωνία ορίζεται από τον άξονα που είναι κάθετος στον φακό του αισθητήρα και της ευθείας που ενώνει το κέντρο του marker με το κέντρο του επίπεδο που ορίζει ο φακός της κάμερας.

Ο υπολογισμός της γωνίας αυτής γίνεται χρησιμοποιώντας το οριζόντιο οπτικό πεδίο (field of view, FOV) της κάμερας, που από το μοντέλο εξομοίωσής της είναι 70°. Ταυτόχρονα, οι εικόνες που κάνει publish η κάμερα είναι ανάλυσης 640x480 pixels, όπου από εκεί γνωρίζουμε πλέον ότι το πλάτος ενός καρέ είναι 640 pixels. Αυτό σημαίνει πως το κάθε pixel κατά το πλάτος αντιστοιχεί σε:

$$\frac{\text{camera field of view}}{\text{camera pixel width}} = \frac{70 \text{ degrees}}{640 \text{ pixels}} = 0.109375 \frac{\text{degrees}}{\text{pixel}}$$

Για παράδειγμα, εάν ένα marker έχει το κέντρο του στις συντεταγμένες  $(x, y) = (125, 240)$  του καρέ, όπου  $x$  είναι το πλάτος και  $y$  το ύψος, μπορούμε να πούμε πως η γωνία υπό την οποία η κάμερα βλέπει το marker είναι:

$$\text{angle from camera} = \text{pixel width} * \frac{\text{degrees}}{\text{pixel}} - \text{offset}$$

Όπου το offset είναι:

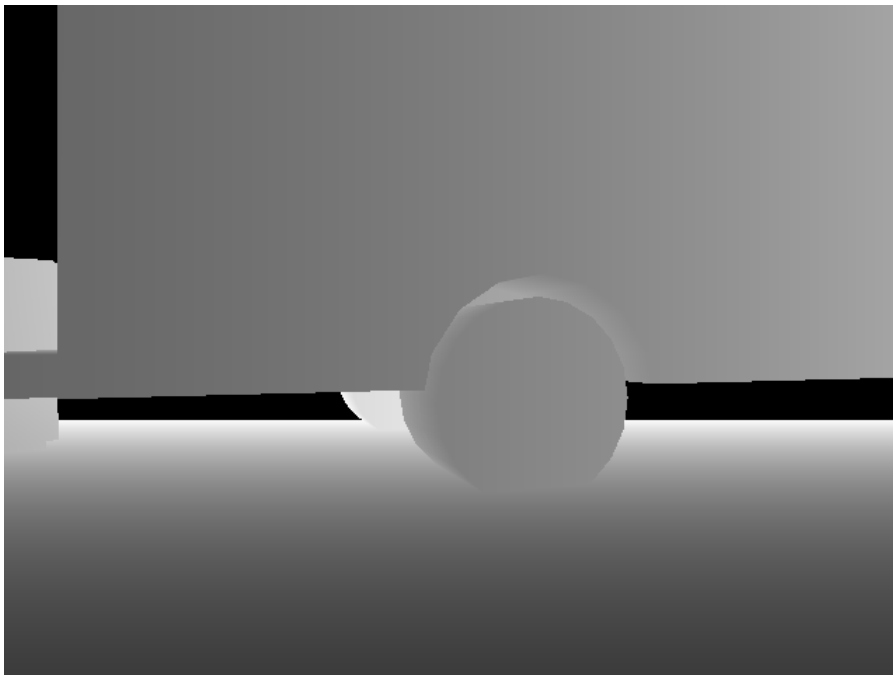
$$\text{offset} = \frac{\text{field of view}}{2}$$

Αντικαθιστώντας:

$$\text{angle from camera} = 125 * 0.109375 - \frac{70}{2} = 21.328125 \text{ degrees}$$

Το εύρος τιμών της γωνίας είναι: [-35, 35] μοίρες ή [-0.610865, 0.610865] σε ακτίνια. Στη συγκεκριμένη περίπτωση η γωνία βγαίνει αρνητική καθώς το σημείο ενδιαφέροντος είναι στα αριστερά από το κέντρο της κάμερας. Εάν το σημείο ήταν στα δεξιά από το κέντρο της κάμερας, η γωνία θα ήταν θετική.

Για την απόσταση, πειραματικά βρέθηκε πως από την εικόνα βάθους που γίνεται publish στο topic "/camera/depth/image\_raw" μπορούμε να πάρουμε μετρήσεις μέχρι τα 5 μέτρα, παρόλο που ο αισθητήρας του kinect αναφέρει πως η μέγιστη απόσταση λήψης πληροφορίας βάθους είναι τα 8 μέτρα. Αυτό συμβαίνει διότι μέχρι τα πέντε μέτρα το kinect μπορεί να "ακολουθήσει" τις κινήσεις του ανθρώπινου σώματος. Οι τιμές για οποιοδήποτε εικονοστοιχείο (pixel) των καρέ που λαμβάνουμε είναι [0, 255] στην κλίμακα του γκρι. Σε αυτού του είδους την εικόνα, τα πλησιέστερα προς την κάμερα σημεία έχουν μικρότερη τιμή από τα πιο απομακρυσμένα. Τα σημεία τα οποία είναι πέρα των 5 μέτρων στο καρέ είναι εντελώς μαύρα, δηλαδή έχουν την τιμή 0.



Εικόνα 19. Μια εικόνα βάθους της ίδιας σκηνής με την Εικόνα 5

Η απόσταση υπολογίζεται ως εξής:

$$\text{distance from camera} = \frac{\text{pixel value}}{\text{pixel max value}} * \text{max sensor range}$$

Για παράδειγμα, για ένα εικονοστοιχείο με την τιμή 139 μπορούμε να πούμε ότι η απόσταση αυτού του σημείου από τον αισθητήρα είναι:

$$\text{distance from camera} = \frac{139}{255} * 5000 \text{ mm} = 2725.49 \text{ mm}$$

Γνωρίζοντας την απόσταση και τη γωνία του σημείου ενδιαφέροντος αλλά και με τα δεδομένα της οδομετρίας που δείχνουν που είναι το ρομπότ στον χώρο και τον προσανατολισμό του, το node υπολογίζει τις συντεταγμένες του marker στο δισδιάστατο χώρο (x και y), αγνοώντας την τρίτη διάσταση z. Τις συντεταγμένες του marker στον χώρο, τις συντεταγμένες του κέντρου του ως προς την εικόνα από την κάμερα, τη γωνία και την απόσταση από την κάμερα τις γεμίζουμε στο message τύπου "marker\_coordinates\_with\_distance". Αυτό το κάνουμε για κάθε marker που έχουμε εντοπίσει, και ύστερα όλα τα messages του τύπου "marker\_coordinates\_with\_distance" τα βάζουμε στο message με το όνομα "marker\_coordinates\_with\_distance\_collection", συμπληρώνοντας το timestamp της καταγραφής και την πληροφορία για την ανάλυση της κάμεράς μας. Οι πληροφορίες αυτές γίνονται publish στο topic "/aruco/marker\_coordinates\_with\_distance\_collection".

### Data Persistence Node

Το Data Persistence Node (DPN)(βλ. Πηγ. Κωδ. 6.1.3) έχει την αρμοδιότητα να κρατάει στη μνήμη όλα τα markers που έχει καταγράψει το POIDGN όσο τρέχει, ταυτόχρονα εμπλουτίζοντας με περισσότερα δεδομένα τη ροή των πληροφοριών.

Κάνοντας εγγραφή στο topic "/aruco/marker\_coordinates\_with\_distance\_collection", το DPN λαμβάνει όλες τις σχετικές πληροφορίες για τα markers τα οποία εντοπίζονται από το POIDGN, αντιστοιχίζοντάς τα με όσα έχει ήδη κρατήσει στη μνήμη του. Για όσα όσα βρίσκει αντιστοιχίες, ανανεώνει τα δεδομένα του με τα καινούρια, εξασφαλίζοντας την έγκυρη πληροφόριση του ρομπότ για τα markers. Ταυτόχρονα, εάν βρεθεί κάποια καινούρια εγγραφή η οποία δεν υπήρχε πριν στη μνήμη, προστίθεται σε αυτή μαζί με τα σχετικά της δεδομένα. Τέλος, εάν βρεθεί κάποια εγγραφή στη μνήμη η οποία στο καινούριο message δεν είναι παρούσα (που σημαίνει ότι είτε το marker δεν βρίσκεται στο οπτικό πεδίο της κάμερας ή ότι δεν αναγνωρίστηκε από αυτήν, λαμβάνει ως δεδομένο ότι η θέση του marker στον χώρο έχει παραμείνει η ίδια. Έτσι, έχοντας χτίσει την παρούσα κατάσταση όλων των markers, προσθέτει για το κάθε ένα τρεις παρακάτω πληροφορίες.

Τα δεδομένα που προσθέτει το node για κάθε marker στην παρούσα κατάσταση είναι τα εξής:

- Την ταχύτητα προσέγγισης του marker από το ρομπότ.
- Την απόσταση του marker από το ρομπότ.
- Τη γωνία του marker με σημείο αναφοράς το ρομπότ και τον προσανατολισμό του.

Για την ταχύτητα προσέγγισης του marker από το ρομπότ, χρησιμοποιείται η μέθοδος calculateApproachingSpeed. Η μέθοδος αυτή υπολογίζει ένα διάνυσμα από τις τοποθεσίες του κάθε marker και του ρομπότ στον δυσδιάστατο χώρο και στη συνέχεια το κανονικοποιεί. Από την οδομετρία του ρομπότ μπορούμε να λάβουμε τις ταχύτητες του ρομπότ ως προς τον x και y άξονα. Το εσωτερικό γινόμενο μεταξύ του κανονικοποιημένου διανύσματος κατεύθυνσης και του διανύσματος της ταχύτητας του ρομπότ θα μας δώσει την ταχύτητα προσέγγισης του ρομπότ προς το marker.

Για παράδειγμα, ας πούμε πως ένα marker στο δυσδιάστατο χώρο βρίσκεται στη θέση  $(x_m, y_m) = (4, 7)$ , και πως το ρομπότ βρίσκεται στη θέση  $(x_r, y_r) = (1, 3)$ , ενώ το διάνυσμα της γραμμικής ταχύτητας του ρομπότ ως προς τις x και y διαστάσεις είναι:

$$\vec{v} = (v_x, v_y) = \left(\frac{5}{3}, 0\right)$$

Το διάνυσμα της κατεύθυνσης από την θέση του ρομπότ στη θέση του marker θα είναι το

$$v_{rmd} = (x_m - x_r, y_m - y_r) = (3, 4)$$

και κανονικοποιημένο:

$$\hat{v}_{rmd} = \frac{v_{rmd}}{\|v_{rmd}\|} = \left(\frac{3}{5}, \frac{4}{5}\right)$$

Το εσωτερικό γινόμενο των  $v_{rmd}$  και  $\vec{v}$  μας δίνει την ταχύτητα προσέγγισης:

$$v_{rmd} \cdot \vec{v} = (3/5 \quad 4/5) \begin{pmatrix} 5/3 \\ 0 \end{pmatrix} = 1$$

Το node είναι υπεύθυνο να ανανεώνει και τις αποστάσεις των marker από το ρομπότ που έχουν καταγραφεί στο παρελθόν, παρόλο που τα markers μπορεί να μην είναι στο οπτικό πεδίο του ρομπότ την παρούσα στιγμή. Αυτό το επιτυγχάνει με τη μέθοδο calculateDistanceOfTwoPoints. Έχοντας καταγράψει την τοποθεσία των markers στον δυδιάστατο χώρο, δηλαδή  $(x_m, y_m)$  και από την οδομετρία γνωρίζοντας την τοποθεσία και του ρομπότ στον χώρο  $(x_r, y_r)$ , η απόσταση βγαίνει από την εξίσωση:  $d = \sqrt{(x_m - x_r)^2 + (y_m - y_r)^2}$

Για το προηγούμενο παράδειγμα όπου  $(x_m, y_m) = (4, 7)$  και  $(x_r, y_r) = (1, 3)$ :

$$d = \sqrt{(4 - 1)^2 + (7 - 3)^2} = 5m$$

Για τη γωνία με σύστημα αναφοράς τη βάση του ρομπότ, η μέθοδος calculateAngleOfMarkerFromRobotFrameOfReference υπολογίζει τη διεύθυνση στον χώρο στην οποία βρίσκεται το marker, χρησιμοποιώντας την θέση του marker και του ρομπότ στον δυδιάστατο χώρο. Η φορά είναι από τη θέση του ρομπότ προς την θέση του marker. Τα δυο αυτά μαζί δίνουν το διάνυσμα της κατεύθυνσης από το ρομπότ προς το marker. Το διάνυσμα κανονικοποιείται ούτως ώστε να έχει μέτρο 1. Ταυτόχρονα, από την οδομετρία του ρομπότ λαμβάνεται ο προσανατολισμός του ρομπότ, ο οποίος γίνεται publish σε μορφή τετραδονίου, το οποίο η μέθοδος μετατρέπει σε γωνίες Euler για πιο εύκολη διαχείριση. Από τις τρεις γωνίες (περιστροφή/roll, κλίση/pitch, εκτροπή/yaw), μόνο η εκτροπή ενδιαφέρει την εφαρμογή καθώς το ρομπότ έχει ελευθερία μόνο ως προς αυτή. Από την εκτροπή υπολογίζεται ένα μοναδιαίο διάνυσμα το οποίο δίνει τον προσανατολισμό του ρομπότ. Έχοντας τόσο το διάνυσμα της κατεύθυνσης του ρομπότ όσο και το διάνυσμα της κατεύθυνσης από το ρομπότ προς το marker, υπολογίζεται η γωνία απόκλισης ως:

$$angle = atan2(det(v_{rd}, v_{rmd}), v_{rd} \cdot v_{rmd})$$

όπου η atan2 είναι μια συνάρτηση με συνθήκες:

$$atan2(y, x) = \begin{cases} 2arctan\left(\frac{y}{\sqrt{x^2+y^2}+x}\right), & \text{if } x > 0, \\ -\pi - 2arctan\left(\frac{\sqrt{x^2+y^2}+x}{y}\right), & \text{if } x \leq 0 \text{ and } y \neq 0, \\ \pi, & \text{if } x < 0 \text{ and } y = 0, \\ \text{undefined}, & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

Η συνάρτηση atan2 δίνει ένα εύρος τιμών  $(-\pi, \pi]$ , όπου το 0 δείχνει πως τα διανύσματα έχουν μηδενική γωνία μεταξύ τους, άρα και το marker βρίσκεται ευθεία μπροστά από το ρομπότ. Για θετικές τιμές, το marker βρίσκεται προς τα αριστερά του ρομπότ και για αρνητικές τιμές το marker βρίσκεται προς τα δεξιά του ρομπότ.

Για παράδειγμα, έχοντας για ακόμα μια φορά τις προηγούμενες θέσεις του marker και του ρομπότ, δηλαδή:  $(x_m, y_m) = (4, 7)$  και  $(x_r, y_r) = (1, 3)$ , ενώ η εκτροπή του ρομπότ (σε μορφή

γωνίας Euler) να είναι:  $\gamma_{aw_r} = \pi$ : Υπολογίζεται το διάνυσμα της κατεύθυνσης από το ρομπότ προς το marker:

$$v_{rmd} = (x_m - x_r, y_m - y_r) = (3, 4)$$

Κανονικοποιώντας το διάνυσμα:

$$\hat{v}_{rmd} = \frac{v_{rmd}}{\|v_{rmd}\|} = \left(\frac{3}{5}, \frac{4}{5}\right)$$

Το μοναδιαίο διάνυσμα του προσανατολισμού του ρομπότ υπολογίζεται ως εξής:

$$v_{rd} = (\cos \pi, \sin \pi) = (-1, 0)$$

Με τα δυο αυτά διανύσματα, υπολογίζεται στη συνέχεια η διακρίνουσα:

$$\det(v_{rd}, \hat{v}_{rmd}) = \det\left(\begin{array}{cc} -1 & 0 \\ \frac{3}{5} & \frac{4}{5} \end{array}\right) = -\frac{4}{5}$$

Και το εσωτερικό γινόμενο  $v_{rd} \cdot \hat{v}_{rmd}$ :

$$v_{rd} \cdot \hat{v}_{rmd} = -1 * \frac{3}{5} + 0 * \frac{4}{5} = -\frac{3}{5}$$

Καθώς το  $v_{rd} \cdot \hat{v}_{rmd} < 0$  και το  $\det(v_{rd}, \hat{v}_{rmd}) \neq 0$ , η συνάρτηση για την γωνία είναι:

$$angle = -\pi - 2\arctan\left(\frac{\sqrt{x^2 + y^2} + x}{y}\right)$$

Αντικαθιστώντας αντίστοιχα τα x και y με:

$$y = \det(v_{rd}, \hat{v}_{rmd}) = -\frac{4}{5}$$

$$x = v_{rd} \cdot \hat{v}_{rmd} = -\frac{3}{5}$$

$$angle = -\pi - 2\arctan\left(\frac{\sqrt{\left(-\frac{3}{5}\right)^2 + \left(-\frac{4}{5}\right)^2} - \frac{3}{5}}{-\frac{4}{5}}\right) \approx -2.21430 \text{ radians}$$

Αφού ολοκληρωθεί η ανανέωση της παρούσας κατάστασης, τα δεδομένα αυτά γίνονται publish στο topic `"/aruco/persistent_marker_coordinates_with_distance_collection"`.

### Operator Intent Inference Node

Το Operator Intent Inference Node (OIIN) (βλ. Πηγ. Κωδ. 6.1.4) αξιοποιεί τα δεδομένα που δίνει το DPN για να κάνει εκτίμηση της πρόθεσης του χειριστή. Το node διαβάζει ένα αρχείο ρύθμισης παραμέτρων ούτως ώστε να γνωρίζει για ποιά marker id's ψάχνει. Αυτό γίνεται διότι πειραματικά διαπιστώθηκε πως εντοπίζονται πολλά ψευδώς θετικά aruco markers από το POIDN και χρειάζεται να φιλτραριστούν για να βρεθούν μόνο τα id's που ενδιαφέρουν την εφαρμογή. Από το topic `"/aruco/persistent_marker_coordinates_with_distance_collection"` στο οποίο το node κάνει subscribe, λαμβάνει τα δεδομένα για τα markers που έχουν βρεθεί και τα επεξεργάζεται κατάλληλα ώστε να μπορούν να αξιοποιηθούν από το MLOII. Το MLOII κάνει την εκτίμηση της πρόθεσης του χειριστή και κάνει publish την εκτίμηση αυτή μαζί με τη "σιγουριά" (confidence) για την επιλογή της στο topic `"/operator_intent_inference"`.

### 3.3.1 Το μοντέλο μηχανικής μάθησης MLOII

#### Συλλογή δεδομένων

Για το MLOII, αρχικά χρειάστηκε να συλλεχτούν δεδομένα. Για το σκοπό αυτό, έγινε χρήση του εργαλείου `rosvbag`, το οποίο επιτρέπει την εγγραφή των μηνυμάτων από τα διάφορα topics, τα οποία και δίνεται η δυνατότητα να επιλέξουμε. Τα δεδομένα τα οποία θα χρησιμοποιηθούν είναι:

- Η ταχύτητα προσέγγισης του marker από το ρομπότ.
- Η απόσταση του marker από το ρομπότ.
- Η γωνία του marker με σημείο αναφοράς το ρομπότ και τον προσανατολισμό του.

Τα δεδομένα αυτά τα κάνει publish το DPN στο topic `"/aruco/persistent_marker_coordinates_with_distance_collection"`. Για το σκοπό της συγκέντρωσης των δεδομένων έπρεπε να δημιουργηθούν κάποια σενάρια στο περιβάλλον εξομοίωσης. Δημιουργήθηκαν δυο από αυτά, όπου στο πρώτο το ρομπότ βρίσκεται στη μέση τριών αντικειμένων που το περιβάλλουν, τα οποία φέρουν επάνω τους από ένα marker (βλ. Εικ. 20).



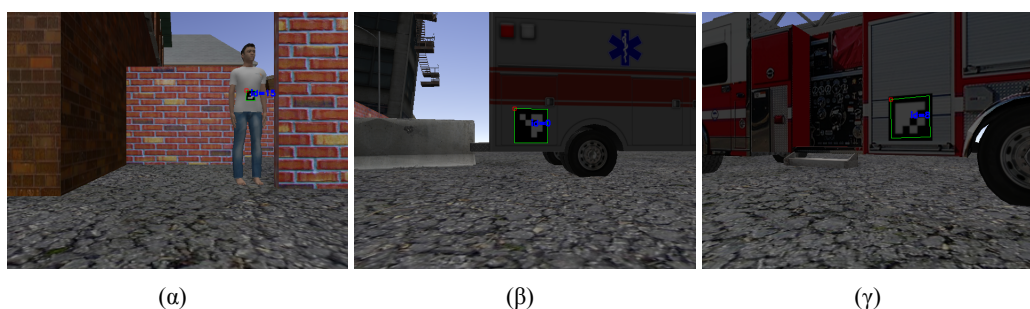
Εικόνα 20. Η κάτοψη του πρώτου περιβάλλοντος συλλογής δεδομένων. Το ρομπότ είναι στο κέντρο των τριών σημείων ενδιαφέροντος, όπου το κάθε ένα φέρει ένα marker. Τα σημεία ενδιαφέροντος είναι ο άνθρωπος, το ασθενοφόρο και το πυροσβεστικό όχημα.

Το πείραμα ξεκινάει με το ρομπότ στη μέση των τριών στόχων. Το ρομπότ πλοηγείται με τη χρήση τηλεχειριστηρίου (βλ. Εικ. 21) στον χώρο με σκοπό την αναγνώριση και καταγραφή όλων των markers στον χώρο (βλ. Εικ. 22), ούτως ώστε να βρεί τη θέση τους στο χώρο προκειμένου να δημιουργία των δεδομένων που μας ενδιαφέρουν. Αφού το ρομπότ δει όλα τα markers, επιστρέφεται στην αρχική του θέση και με τη χρήση του εργαλείου `rosvbag` ξεκινάει η καταγραφή του topic `"/aruco/persistent_marker_coordinates_with_distance_collection"`:

1 `S rosvbag record /aruco/persistent_marker_coordinates_with_distance_collection`



Εικόνα 21. Πλοήγηση του ρομπότ στο χώρο με τη χρήση τηλεχειριστηρίου.



Εικόνα 22. (α) Αναγνώριση του marker με id: 15 στον άνθρωπο (β) Αναγνώριση του marker με id: 0 στο ασθενοφόρο (γ) Αναγνώριση του marker με id: 8 στο πυροσβεστικό όχημα

Με την εκκίνηση της εγγραφής του topic, το ρομπότ πλοηγείται προς το επιθυμητό σημείο ενδιαφέροντος / marker, το οποίο ο χειριστής έχει θέσει εξαρχής. Με την άφιξη του ρομπότ στο στόχο, σταματάει η εγγραφή, και το αρχείο με τις εγγραφές αποθηκεύεται για περαιτέρω επεξεργασία, ταυτόχρονα κατηγοριοποιώντας το με το στόχο του χειριστή του ρομπότ. Για παράδειγμα, ξεκινάει μια εγγραφή και ο χειριστής του ρομπότ ανακοινώνει πως ο στόχος του είναι ο άνθρωπος ο οποίος φέρει επάνω του το marker με id: 15 (βλ. Εικ. 22). Πλοηγεί το ρομπότ προς τον άνθρωπο και αφού φτάσει τον επιθυμητό του στόχο η εγγραφή σταματάει. Αυτό γίνεται για κάθε στόχο ίσες φορές, ούτως ώστε το σύνολο των δεδομένων να είναι όσο το δυνατόν πιο αντιπροσωπευτικό για όλους τους στόχους.

Στο δεύτερο πείραμα γίνεται η ίδια ακριβώς διαδικασία όπως παραπάνω, αλλά σε ένα νέο περιβάλλον (βλ. Εικ. 23). Η αιτία της δημιουργίας του δεύτερου πειράματος ήταν η υπόθεση και κατόπιν ελέγχου διαπίστωση ότι το MLOII δεν είχε καλή γενίκευση εάν τα σημεία στο χώρο δεν είχαν παρόμοια διάταξη με αυτά του πρώτου πειράματος. Στο νέο περιβάλλον και τα τρία markers βρίσκονται επάνω στο ίδιο αντικείμενο, το οποίο είναι το ασθενοφόρο. Αυτή είναι μια ιδιαίτερη



περίπτωση όπου και οι τρεις στόχοι είναι επάνω στο ίδιο επίπεδο του χώρου (βλ. Εικ. 24).



Εικόνα 23. Η κάτοψη του δεύτερου περιβάλλοντος συλλογής δεδομένων. Το ρομπότ είναι τοποθετημένο απέναντι στο ασθενοφόρο το οποίο φέρει τα 3 markers.



Εικόνα 24. Το δεύτερο περιβάλλον από την οπτική γωνία της κάμερας του ρομπότ. Λόγω του ότι και τα 3 markers είναι μέσα στην ίδια εικόνα, δεν χρειάζεται η περιήγηση του ρομπότ στο χώρο.

### Επεξεργασία δεδομένων

Έχοντας συλλέξει όλα τα δεδομένα, γίνεται η επεξεργασία τους ούτως ώστε να έρθουν σε μορφή κατάλληλη για να αξιοποιηθούν από τις βιβλιοθήκες της μηχανικής μάθησης. Το κάθε αρχείο καταγραφής των δεδομένων διαβάζεται και ταυτόχρονα ελέγχεται για μη έγκυρα δεδομένα που τυχόν έχουν συλλεχτεί. Λόγω του ότι χρειαζόμαστε αντιπροσωπευτικό δείγμα για τις προσεγγίσεις κάθε στόχου, διαλέγονται 100 έγκυρες εγγραφές από το κάθε αρχείο, τυχαία. Εν συνεχεία, επισυνάπτεται σε κάθε εγγραφή η κατηγορία (Label) που αφορά το στόχο προσέγγισης. Μαζεύοντας όλες τις εγγραφές σε ένα αρχείο, ετοιμάζεται το σύνολο των δεδομένων, πλέον κατηγοριοποιημένα.

marker_0 distance	marker_0_angle radians	marker_0_approach speed	marker_8 distance	marker_8_angle radians	marker_8_approach speed	marker_15 distance	marker_15_angle radians	marker_15_approach speed	Goal
3000.9374	-0.1334337765	0.5167654916	3325.555	-2.7393845739	-0.3999840641	4704.65622	1.5102105123	-0.1243806491	marker_0
2811.2241	-0.1898325395	0.4555174055	3445.212	3.0020824688	-0.4528588523	5564.62402	1.2287575266	0.0281547161	marker_0
3793.0413	0.0555179712	0.4353671036	3388.09	-0.4912893968	0.3990715972	4905.00945	-0.9301675184	0.2840364481	marker_0
5167.2076	-1.5777354931	-0.0127135539	3025.386	2.99525867	-0.4800675493	4046.56355	-0.1469277808	0.4800198458	marker_15
1462.4021	0.3572825014	-0.4401295955	5535.024	-1.6975701344	0.0535103982	3528.25497	2.8304984678	0.4513689176	marker_15
5171.3999	0.9368758958	0.3123335188	6222.652	-0.2220292296	0.5042215056	1075.09678	-3.0061172853	-0.5145276314	marker_8
1629.8106	-0.0047198857	0.5022987344	1876.316	-1.1924865342	0.1841990626	4099.36737	-1.4660332575	0.0511044235	marker_0
5210.0027	-1.6969675906	-0.0565590593	3658.706	3.0241590521	-0.4990307991	3422.3706	-0.1708764017	0.4955298934	marker_15
4252.7548	2.5052311291	-0.4185508182	1878.265	-0.1738197821	0.5010803699	5462.42901	-2.6959374065	-0.4468454944	marker_8

Εικόνα 25. Η μορφή των δεδομένων μετά την επεξεργασία.

### Εκπαίδευση του MLOH

Επιλέχθηκε ως αλγόριθμος μηχανικής μάθησης το "Τυχαίο Δάσος" (Random Forest, RF), ένας αλγόριθμος που μπορεί να χρησιμοποιηθεί τόσο πρόβλημα της παλινδρόμησης όσο και στο πρόβλημα της ταξινόμησης. Ο συγκεκριμένος αλγόριθμος [25] εκπαιδεύει με τη σειρά του απλούστερα "Δέντρα Αποφάσεων" (Decision Trees) και χρησιμοποιεί το σύνολο των δέντρων στις εκτιμήσεις του. Στο πρόβλημα της παλινδρόμησης ως έξοδος είναι ο μέσος όρος των αποτελεσμάτων των δέντρων αποφάσεων, ενώ στην ταξινόμηση λαμβάνεται ως αποτέλεσμα η πρόβλεψη στην οποία έχουν καταλήξει τα περισσότερα δέντρα αποφάσεων. Το RF είναι λιγότερο επιρρεπές στην υπερπροσαρμογή στα δεδομένα εκμάθησης από ένα απλό δέντρο αποφάσεων, εξαλείφοντας τις τυχόν μεροληψίες του κάθε μεμονωμένου δέντρου απόφασης.

Το πρόβλημα ανήκει στην κατηγορία της ταξινόμησης επειδή προβλέπεται ο στόχος που θέλει να προσεγγίσει ο χρήστης και όχι κάποια αριθμητική τιμή, επομένως έγινε χρήση του Ταξινομητή Τυχαίου Δάσους (Random Forest Classifier). Για τον Ταξινομητή Τυχαίου Δάσους επιλέχτηκε μετά από πειραματισμό ο αριθμός των τυχαίων δέντρων να είναι πενήντα. Για τα δεδομένα και το πρόβλημα, οι πενήντα εκτιμητές (estimators) προσφέρουν καλή απόδοση, ενώ ταυτόχρονα το τελικό μοντέλο είναι σχετικά ελαφρύ από υπολογιστική άποψη.

```
1 rfc = RandomForestClassifier(n_estimators=50)
```

Τα δεδομένα συνολικά ανέρχονται στις δυο χιλιάδες εκατό εγγραφές, χωρίζονται σε δεδομένα εκπαίδευσης (training) και σε δεδομένα δοκιμών (testing), όπου τα δεδομένα εκπαίδευσης στο πλήθος είναι χίλια τετρακόσια εβδομήντα και τα δεδομένα δοκιμών είναι εξακόσια τριάντα.

```
1 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state=133)
```

Το MLOH εκπαιδεύεται με τα δεδομένα εκπαίδευσης.

```
1 rfc.fit(X_train, Y_train.values.ravel())
```

Στη συνέχεια, με το MLOπ πλέον εκπαιδευμένο, χρησιμοποιούνται τα δεδομένα δοκιμών για να γίνουν οι προβλέψεις για αυτά.

```
1 Y_pred = rfc.predict(X_test)
```



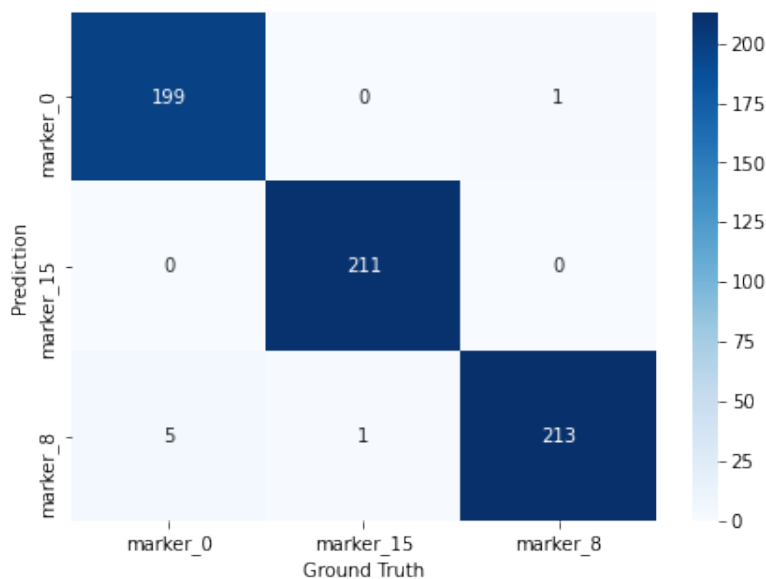
## Κεφάλαιο 4

# Αποτελέσματα – Ευρήματα / Επιτεύγματα

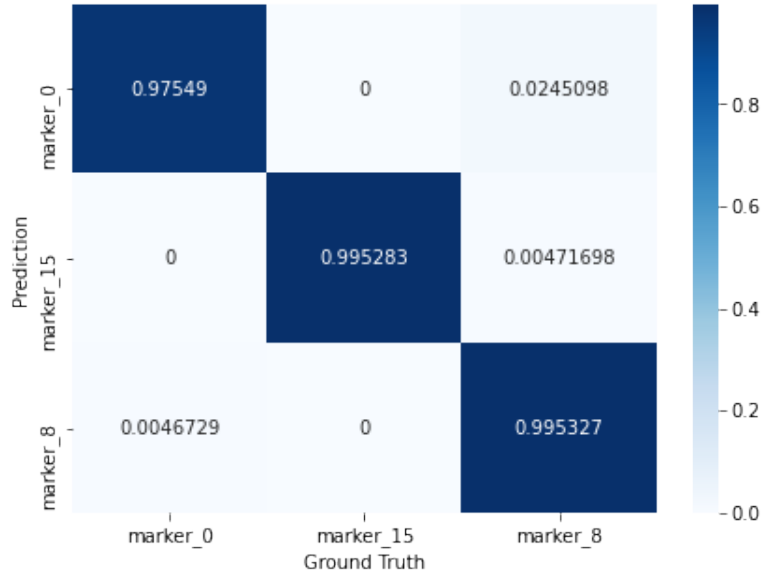
### 4.1 Αναλυτική παρουσίαση αποτελεσμάτων

#### 4.1.1 Αποτελέσματα της εκπαίδευσης - Αξιολόγηση του MLOII

Έχοντας λάβει τις προβλέψεις από το MLOII, συγκρίνονται με τα πραγματικά δεδομένα. Ένας τρόπος για να ελέγξουμε τα δεδομένα είναι με το λεγόμενο "Πίνακα Σύγχυσης" (Confusion Matrix). Ο πίνακας σύγχυσης επιτρέπει την οπτικοποίηση της απόδοσης ενός αλγορίθμου, συνήθως με επιβλεπόμενη μάθηση. Ο πίνακας σύγχυσης για την απόδοση του MLOII (βλ. Εικ. 26) δείχνει πως το MLOII είναι αρκετά καλό στην απόδοσή του. Αυτό συμπεραίνεται από τη διαγώνιο, η οποία έχει στο σύνολό της εξακόσιες εικόσι τρεις (623) σωστές προβλέψεις (συμπεραίνεται καθώς η αληθινή τιμή ταυτίζεται με την προβλεπόμενη τιμή), ενώ εκτός της διαγωνίου βρίσκονται επτά (7) λανθασμένες προβλέψεις.



Εικόνα 26. Ο πίνακας σύγχυσης του MLOII.



Εικόνα 27. Ο κανονικοποιημένος πίνακας σύγκυσης του MLOII.

Δεδομένου ότι το σύνολο των τιμών προς δοκιμή είναι εξακόσιες τριάντα, η ορθότητα (accuracy) του MLOII είναι:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Number of all Predictions}} * 100\% = \frac{623}{630} * 100\% \approx 98.89\%$$

### Ακρίβεια (Precision)

$$Precision = \frac{TP}{TP + FP} =$$

Ο υπολογισμός αυτός πρέπει να γίνει για κάθε μια από τις κατηγορίες, λόγω του ότι η ταξινόμηση είναι πολλαπλών κλάσεων και όχι δυαδικής μορφής.

- Για το marker 0:

$$Precision_0 = \frac{TP_0}{TP_0 + FP_0} = \frac{199}{199 + 1} = 0.995$$

- Για το marker 15:

$$Precision_{15} = \frac{TP_{15}}{TP_{15} + FP_{15}} = \frac{211}{211 + 0} = 1$$

- Για το marker 8:

$$Precision_8 = \frac{TP_8}{TP_8 + FP_8} = \frac{213}{213 + 6} \approx 0.97260274$$

Για την εύρεση της μέσης ακρίβειας:

$$Precision_{avg} = \frac{Precision_0 + Precision_8 + Precision_{15}}{3} \approx 0.9892$$

**Ανάκληση (Recall)**

$$R = \frac{TP}{TP + FN}$$

Ο υπολογισμός αυτός πρέπει να γίνει για κάθε μια από τις κατηγορίες, λόγω του ότι η ταξινόμηση είναι πολλαπλών κλάσεων και όχι δυαδικής μορφής.

- Για το marker 0:

$$Recall_0 = \frac{TP_0}{TP_0 + FN_0} = \frac{199}{199 + 5} \approx 0.9754902$$

- Για το marker 15:

$$Recall_{15} = \frac{TP_{15}}{TP_{15} + FN_{15}} = \frac{211}{211 + 1} \approx 0.99528302$$

- Για το marker 8:

$$Recall_8 = \frac{TP_8}{TP_8 + FN_8} = \frac{213}{213 + 1} \approx 0.9953271$$

Για την εύρεση της μέσης ανάκλησης:

$$Recall_{avg} = \frac{Recall_0 + Recall_8 + Recall_{15}}{3} \approx 0.9887$$

**F1-score**

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

Ο υπολογισμός αυτός πρέπει να γίνει για κάθε μια από τις κατηγορίες, λόγω του ότι η ταξινόμηση είναι πολλαπλών κλάσεων και όχι δυαδικής μορφής.

- Για το marker 0:

$$F1_0 = \frac{2}{\frac{1}{Precision_0} + \frac{1}{Recall_0}} = \frac{2}{\frac{1}{0.995} + \frac{1}{0.9754902}} \approx 0.98514851$$

- Για το marker 15:

$$F1_{15} = \frac{2}{\frac{1}{Precision_{15}} + \frac{1}{Recall_{15}}} = \frac{2}{\frac{1}{1} + \frac{1}{0.99528302}} \approx 0.99763593$$

- Για το marker 8:

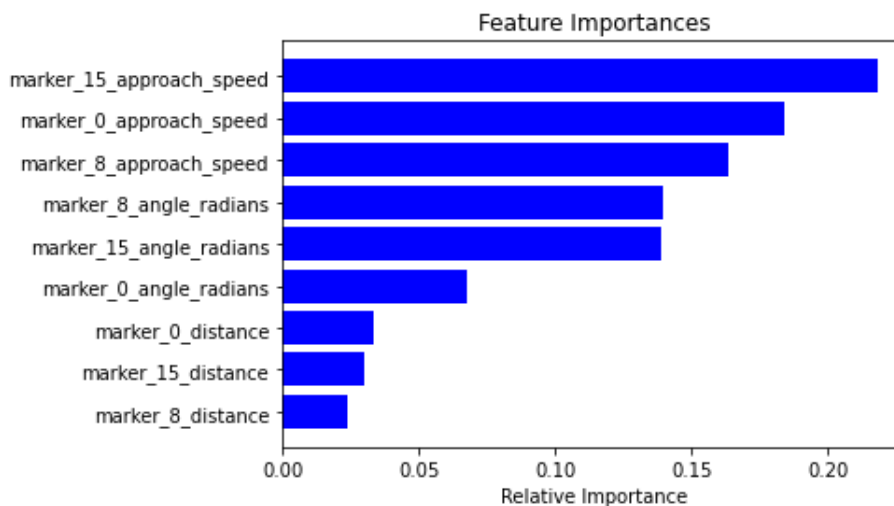
$$F1_8 = \frac{2}{\frac{1}{Precision_8} + \frac{1}{Recall_8}} = \frac{2}{\frac{1}{0.97260274} + \frac{1}{0.9953271}} \approx 0.98383372$$

Για την εύρεση της μέσης ανάκλησης:

$$F1_{\text{avg}} = \frac{F1_0 + F1_8 + F1_{15}}{3} \approx 0.9889$$

Η ορθότητα του MLOII ανέρχεται σε υψηλό ποσοστό, υποδηλώνοντας την ικανότητα του MLOII να κάνει σωστά ταξινόμηση σε ένα μεγάλο αριθμό των δεδομένων. Τόσο η ακρίβεια όσο και η ανάκληση του MLOII είναι αρκετά υψηλή, χωρίς η μια μετρική να υπερέχει σημαντικά έναντι της άλλης, μιας και δεν υπάρχει σημαντικός αριθμός ψευδώς θετικών (FP) ή ψευδώς αρνητικών (FN). Αυτό εξηγεί και το υψηλό F1-score.

Αξία έχει επίσης η σχετική σημασία του κάθε δεδομένου εισόδου για το αποτέλεσμα της εκτίμησης του MLOII. Στην εικόνα 28 φαίνεται πως οι ταχύτητες προσέγγισης των στόχων έχουν τη μεγαλύτερη σημασία για την εκτίμηση της πρόθεσης του χειριστή από το MLOII. Δεύτερες σε σημασία είναι οι γωνίες που διαγράφονται από τη θέση των markers σε σύγκριση με τη θέση του ρομπότ και του προσανατολισμού του, ενώ τελευταίες σε σημασία φαίνονται πως είναι οι αποστάσεις μεταξύ του ρομπότ και όλων των markers.



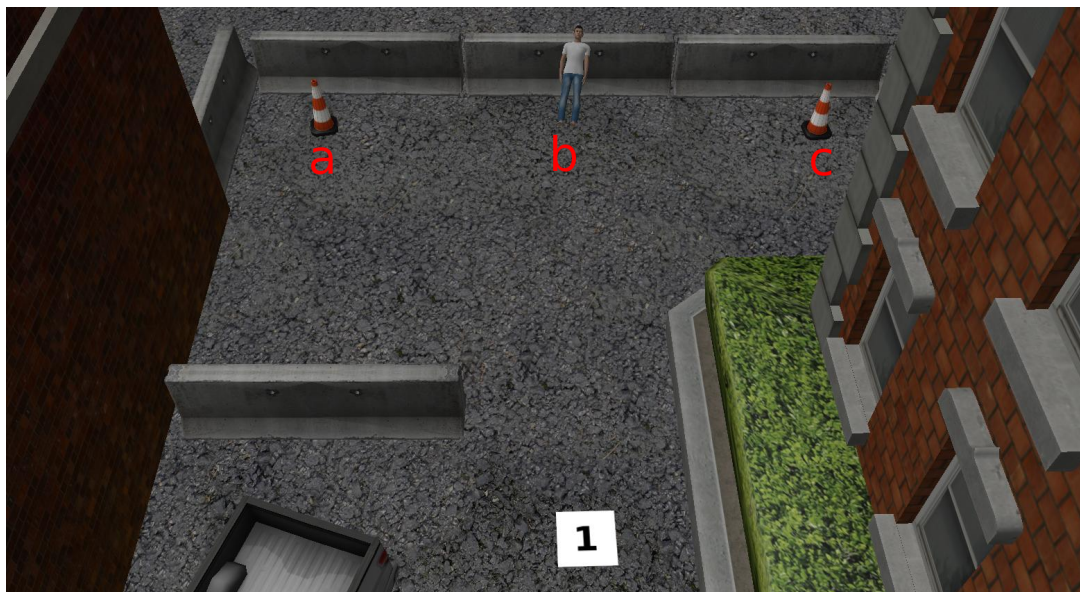
Εικόνα 28. Η σημασία των δεδομένων εισόδου για το MLOII.

### Πειραματικό πρωτόκολλο

Για να δοκιμαστεί το MLOII στα πλαίσια μιας εφαρμογής κινητής ρομποτικής (mobile robotics), διεξήχθησαν τρία πειράματα στο περιβάλλον εξομίωσης Gazebo. Τα τρία αυτά πειράματα αποτελούνται από τρία διαφορετικά σενάρια σε ένα τόπο καταστροφής (disaster site):

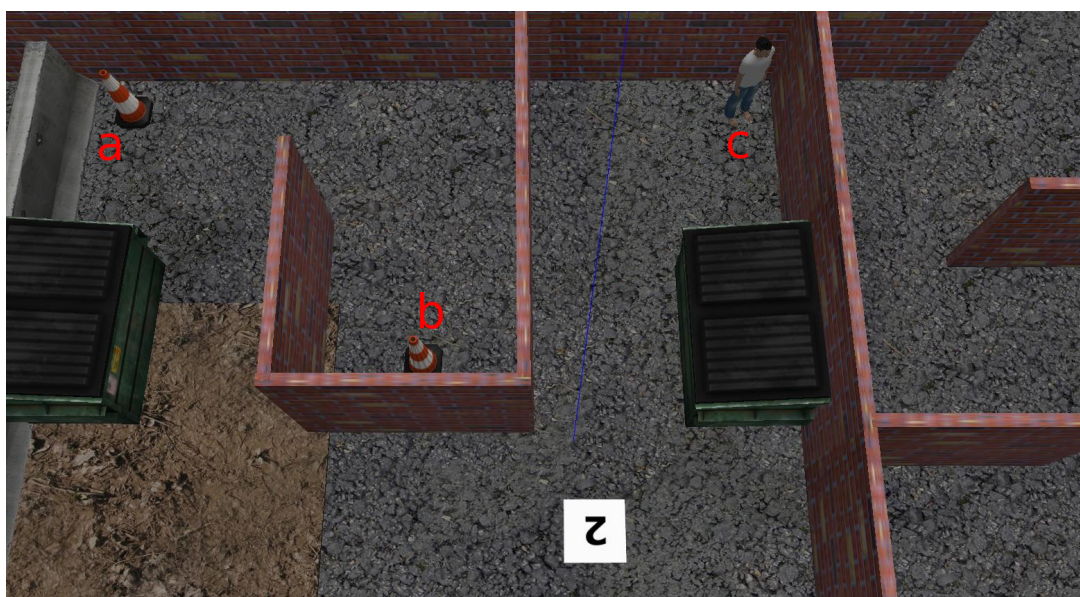
- Για το πρώτο σενάριο (βλ. Εικ. 29), ο χειριστής του ρομπότ ξεκινάει να πλοηγεί το ρομπότ προς το στόχο b. Στα μισά του πειράματος η πρόθεση του χειριστή αλλάζει και επιχειρεί να πλοηγηθεί προς το στόχο a.





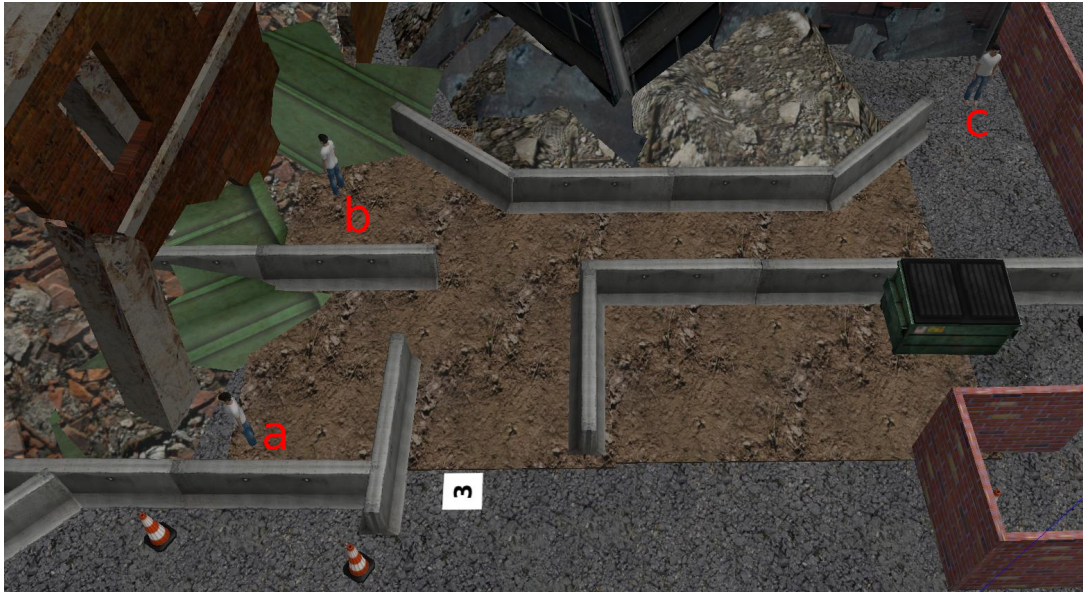
Εικόνα 29. Πρώτο σενάριο.

- Για το δεύτερο σενάριο (βλ. Εικ. 30), ο χειριστής του ρομπότ έχει ως στόχο και πλοηγεί το ρομπότ προς το c. Η πρόθεση του χειριστή δεν αλλάζει για ολόκληρη τη διάρκεια της δοκιμής.



Εικόνα 30. Δεύτερο σενάριο.

- Για το τρίτο σενάριο (βλ. Εικ. 31), ο χειριστής του ρομπότ επιχειρεί να προσεγγίσει διαδοχικά τους στόχους με τη σειρά a, b και τέλος c. Ο στόχος του χειριστή αλλάζει μόλις ο χειριστής προσεγγίσει αρκετά τον κάθε στόχο, όπου θεωρείται ότι αλλάζει και η πρόθεσή του.



Εικόνα 31. Τρίτο σενάριο.

Στη διάρκεια όλων των δοκιμών, συλλέχθηκαν και καταγράφηκαν δεδομένα από τα topics του ROS μέσω της χρήσης του Rosbag, τα οποία μετά χρησιμοποιήθηκαν με τις δύο μεθόδους, το MLOII και το BOIR (Bayesian Operator Intent Recognition) [9] προκειμένου να βγουν τα αποτελέσματα. Για τα πειράματα, το MLOII και το BOIR γνωρίζουν εξαρχής τις τοποθεσίες των στόχων, καθώς το BOIR δεν έχει κάποιο σύστημα για το δυναμικό εντοπισμό στόχων και την εύρεση των θέσεών τους στο χώρο. Με την αναπαραγωγή των Rosbags λήφθηκαν και από τα δυο μοντέλα η εκτίμηση του στόχου καθώς και οι πιθανότητες (probabilities) για τον κάθε πιθανό στόχο να είναι η πρόθεση του χειριστή. Η κάθε πιθανότητα (probability) στη συγκεκριμένη περίπτωση μπορεί να ερμηνευτεί και ως αυτοπεποίθηση (confidence) του μοντέλου για την ταξινόμησή του.

Αξιοποιήθηκαν δύο μέτρα σύγκρισης:

- Ορθότητα (Accuracy)
- Cross-Entropy Loss

Η συνάρτηση Cross-Entropy Loss ορίζεται ως εξής:

$$L = - \sum_{i=1}^n t_i * \log(p_i)$$

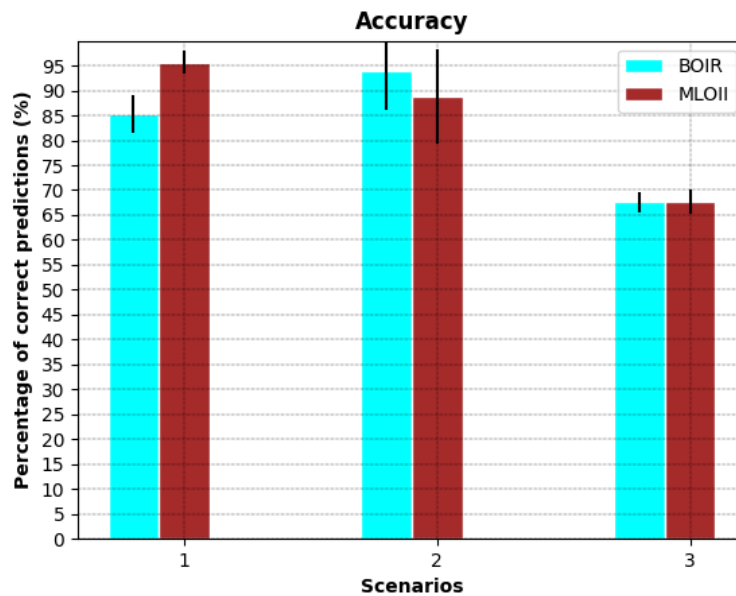
Όπου το  $t$  είναι ένα διάνυσμα που ορίζει την πραγματική τιμή, ενώ το  $p$  είναι ένα διάνυσμα κατανομής των πιθανοτήτων για την προβλεπόμενη τιμή.

Για παράδειγμα, για  $t = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$  και για  $p = \begin{bmatrix} 0.72 \\ 0.21 \\ 0.07 \end{bmatrix}$  :

$$L = - \sum_{i=1}^n t_i * \log(p_i) = -(1 * \log(0.72) + 0 * \log(0.21) + 0 * \log(.07)) \approx 0.143$$

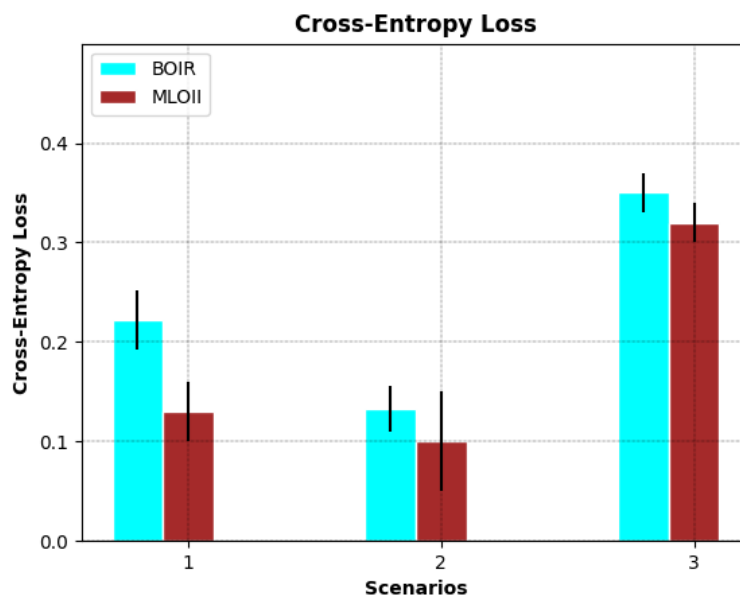
Αξίζει να σημειωθεί ότι όσο μικρότερο είναι το αποτέλεσμα της Cross-Entropy Loss τόσο καλύτερο είναι το μοντέλο. Ένα ιδανικό μοντέλο θα έχει 0 loss.

Ακολουθούν τα αποτελέσματα από τα πειράματα:



Εικόνα 32. Ορθότητα του BOIR και του MLOII από τα πειράματα.

Από το σύνολο των εκτιμήσεων της πρόθεσης του χειριστή και από τα δυο μοντέλα, το MLOII φαίνεται να έχει μεγαλύτερη ορθότητα στο πρώτο σενάριο με ποσοστό 95.58% έναντι του BOIR με 85.30%. Στο δεύτερο σενάριο, η ορθότητα του MLOII ανέρχεται στο 88.86% έναντι του BOIR στο 94%. Στο τρίτο σενάριο, η ορθότητα του MLOII και του BOIR είναι στα ίδια περίπου επίπεδα, με το MLOII να επιτυγχάνει 67.66% και το BOIR 67.65%.



Εικόνα 33. Cross-Entropy Loss του BOIR και του MLOII από τα πειράματα.

Όσον αφορά τη μετρική Cross-Entropy, τα αποτελέσματα δείχνουν ότι σε όλα τα σενάρια

το MLOII είχε μεγαλύτερη αυτοπεποίθηση (confidence) στις προβλέψεις του σε σύγκριση με το BOIR. Για το πρώτο σενάριο το MLOII έχει loss 0.13 σε σύγκριση με το BOIR με 0.222. Στο δεύτερο σενάριο το MLOII καταφέρνει να έχει 0.1 loss έναντι του BOIR, το οποίο επιτυγχάνει 0.133. Τέλος, στο τρίτο σενάριο, το MLOII έχει loss 0.32, ενώ το BOIR 0.35.

## 4.2 Κυριότερα ευρήματα / αποτελέσματα

Από τις μετρικές της ορθότητας, της ακρίβειας, της ανάκλησης και του F1-score οι οποίες ανέρχονται όλες πάνω από το 98%, συμπεραίνεται πως το MLOII είναι ιδιαίτερα εύστοχο στην εκτίμηση της πρόθεσης του χειριστή του ρομπότ.

Από τα αποτελέσματα των τριών σεναρίων, το MLOII επιτυγχάνει συγκρίσιμη ορθότητα με τη μέθοδο BOIR από τη βιβλιογραφία, το πραγματικό πλεονέκτημα του MLOII είναι στη μεγαλύτερη αυτοπεποίθηση στις προβλέψεις του σε σύγκριση με το BOIR, κάτι που παρατηρήθηκε σε όλα τα σενάρια που εξετάστηκαν.

## Κεφάλαιο 5

# Συζήτηση – Συμπεράσματα – Μελλοντικές επεκτάσεις

### 5.1 Ανακεφαλαίωση

Αυτή η εργασία παρουσίασε μια μέθοδο εύρεσης στόχων στο περιβάλλον ενός ρομπότ και τον υπολογισμό των θέσεών τους στον χώρο. Ταυτόχρονα έγινε χρήση ενός μοντέλου μηχανικής μάθησης, που υπολογίζει κατά τη διάρκεια της πλοήγησης του ρομπότ την πρόθεση του χειριστή όσον αφορά το στόχο που θα προσεγγίσει. Τα δεδομένα που θα χρησιμοποιήσει το μοντέλο μηχανικής μάθησης υπολογίζονται με χρήση της θέσης του ρομπότ και τον προσανατολισμό του, καθώς και τις υπολογισμένες θέσεις όλων των στόχων, χωρίς να αξιοποιείται κάποιο σύστημα παρακολούθησης του χειριστή και της βιολογικής του δραστηριότητάς.

### 5.2 Συζήτηση / Συμπεράσματα

Η εκτίμησης της πρόθεσης του χειριστή είναι 98.89% ορθή με βάση τα αποτελέσματα από το Κεφάλαιο 4, κάτι που δίνει στο ρομπότ μεγάλη ακρίβεια στο συμπέρασμα που καταλήγει. Αξίζει να σημειωθεί πως τα πειράματα διεξήχθησαν σε περιβάλλοντα που δεν είχαν περίπλοκη γεωμετρία, δηλαδή δεν υπήρξαν εμπόδια μεταξύ του ρομπότ και των στόχων που εντοπίζει. Εκτιμάται πως σε περιβάλλοντα περίπλοκης γεωμετρίας η εκτίμηση της πρόθεσης δεν θα είναι το ίδιο ακριβής, καθώς ο χειριστής οφείλει να κάνει παρακάμψεις γύρω από τα εμπόδια προκειμένου να φτάσει στον στόχο του. Λόγω του ότι το μοντέλο έχει δώσει μεγάλη βαρύτητα στην ταχύτητα προσέγγισης του στόχου, το μοντέλο πιθανότατα θα αστοχήσει στην εκτίμησή της πρόθεσης του χειριστή για τη διάρκεια της παρακάμψης. Φυσικά αυτό εξαρτάται κατά μεγάλο βαθμό και από το πόσο εκτός πορείας θα χρειαστεί να βγει το ρομπότ για να την εκτελέσει, όπως και από τις τοποθεσίες των υπόλοιπων στόχων στο χώρο. Οι Panagoroulos *et al* [9] στη δική τους δουλειά αξιοποίησαν την απόσταση από ένα σχεδιαστή διαδρομής, και όχι την ευκλείδια απόσταση δυο σημείων. Η ιδέα αυτή τους δίνει ένα σημαντικό πλεονέκτημα σε περιπτώσεις σαν και αυτή της περίπλοκης γεωμετρίας.

### 5.3 Αξιοποίηση / Πρακτικές προεκτάσεις της έρευνας

Το σύστημα μπορεί να χρησιμοποιηθεί σε ρομπότ που αξιοποιούν διαμοιρασμένο έλεγχο, προκειμένου το ρομπότ να έχει καλύτερη πληροφόρηση για την πρόθεση του χειριστή. Η πληροφορία αυτή λαμβάνεται χωρίς να επιβαρύνει ο χειριστής με παραπάνω φόρτο εργασίας και πιθανότατα μπορεί να οδηγήσει σε καλύτερη απόδοση το σύστημα διαμοιρασμένου ελέγχου, με λιγότερες διαμάχες για τον έλεγχο του ρομπότ από το σύστημα του.

Ακόμα, με την ίδια λογική για το μοντέλο μηχανικής μάθησης με 3 στόχους, μπορούν να εκπαιδευτούν πολλαπλά μοντέλα για πλήθος  $n$  markers, όπου  $n \in \mathbb{N}$ , ούτως ώστε το σύστημα να διαλέγει ανάλογα με το πλήθος των markers που έχουν εντοπιστεί το αντίστοιχο μοντέλο για την εκτίμηση της πρόθεσης του χειριστή.

## 5.4 Μελλοντικές επεκτάσεις / Πρακτικές Προεκτάσεις της Έρευνας

Στη θέση των εμπιστευτικών σημάνσεων Aguco οφείλουν να μπουν αλγόριθμοι ανίχνευσης αντικειμένων (object detection algorithms). Σε ένα πραγματικό σενάριο έρευνας και διάσωσης, το ρομπότ οφείλει να εντοπίζει σημεία ενδιαφέροντος όπως άνθρωποι, πηγές κινδύνου κ.α χωρίς την αξιοποίηση χαρακτηριστικών όπως αυτά των εμπιστευτικών σημάνσεων. Μερικά παραδείγματα αλγόριθμων ανίχνευσης αντικειμένων είναι τα Region-based Convolutional Neural Networks (R-CNN), Spatial Pyramid Pooling (SPP-net), YOLO (You Only Look Once) και Histogram of Oriented Gradients (HOG).

Ένας ακόμα περιορισμός έγκειται στην απόσταση που φτάνει η εικόνα βάθους του Kinect. Καθώς το Kinect έχει σχεδιαστεί για άλλη χρήση σε βιντεοπαιχνίδια όπου οι άνθρωποι στέκονται αρκετά κοντά στην τηλεόραση/οθόνη, η απόσταση των 5 μέτρων που προσφέρει ο αισθητήρας δεν είναι αρκετή για χρήση σε πραγματικά σενάρια έρευνας και διάσωσης. Ο αισθητήρας αυτός θα αντικατασταθεί από άλλους αισθητήρες που προσφέρουν μεγαλύτερη εμβέλεια, προκειμένου να ξεπεραστεί ο περιορισμός και να αναγνωρίζονται οι θέσεις των στόχων από μεγαλύτερη απόσταση.

Τέλος, η εισαγωγή της πληροφορίας που δίνει ένας σχεδιαστής διαδρομής (path planner) θα προστεθεί στην εκτίμηση της πρόθεσης του χειριστή. Ο σχεδιαστής διαδρομής θα προσφέρει στο μοντέλο πληροφορία η οποία είναι πολύ σημαντική όταν το περιβάλλον έχει περίπλοκη γεωμετρία και υπάρχουν εμπόδια στο χώρο. Αυτό θα οδηγήσει σε καλύτερη εκτίμηση της πρόθεσης του χειριστή και κατά τη διάρκεια μιας παράκαμψης όσο επιχειρεί να προσεγγίσει κάποιο στόχο.

# Κεφάλαιο 6

## Παράρτημα

### 6.1 Πηγαίος κώδικας

#### 6.1.1 Points of Interest Detection Node

Thesis/Source code/detect\_aruco.cpp

```
1 // Image conversion from ROS to CV format libraries
2 #include <ros/ros.h>
3 #include <image_transport/image_transport.h>
4 #include <cv_bridge/cv_bridge.h>
5 #include <sensor_msgs/image_encodings.h>
6 #include <opencv2/imgproc/imgproc.hpp>
7 #include <opencv2/highgui/highgui.hpp>
8
9 // Custom msgs
10 #include <operator_intent_msgs/point_2dc.h>
11 #include <operator_intent_msgs/marker.h>
12 #include <operator_intent_msgs/marker_collection.h>
13
14 // Aruco relevant libraries
15 #include <opencv2/aruco.hpp>
16 #include <opencv2/opencv.hpp>
17 #include <opencv2/highgui.hpp>
18
19 // Node template
20 #include "node_template.cpp"
21
22 class DetectAruco
23 {
24 private:
25     const std::string OPENCV_WINDOW = "Aruco Marker Detection";
26     ros::NodeHandle nh_;
27     image_transport::ImageTransport it_;
28     image_transport::Subscriber image_sub_;
29     image_transport::Publisher image_pub_;
30     ros::Publisher markers_loc_pub_;
31     std::string pub_topic_, sub_rgb_image_topic_;
32
33     // Initialize vectors for the ID's of the markers and the marker corners
34     std::vector<int> marker_ids;
35     std::vector<std::vector<cv::Point2f>> marker_corners;
36
```

```

37 // Define the dictionary to be detected
38 cv::Ptr<cv::aruco::Dictionary> dictionary = cv::aruco::getPredefinedDictionary(cv::aruco::
    DICT_4X4_50);
39 void imageCallback(const sensor_msgs::ImageConstPtr &msg);
40
41 public:
42 DetectAruco(ros::NodeHandle, ros::NodeHandle);
43 ~DetectAruco();
44 };
45
46 void DetectAruco::imageCallback(const sensor_msgs::ImageConstPtr &msg)
47 {
48     cv_bridge::CvImagePtr cv_ptr;
49     try
50     {
51         cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
52     }
53     catch (cv_bridge::Exception &e)
54     {
55         ROS_ERROR("cv_bridge exception: %s", e.what());
56         return;
57     }
58
59     // Detect tags in the image, using the provided detectMarkers method
60     cv::aruco::detectMarkers(cv_ptr->image, dictionary, marker_corners, marker_ids);
61     // Draw the bounding box around the detected tags using the provided drawDetectedMarkers method
62     // Commented out for debugging
63     cv::aruco::drawDetectedMarkers(cv_ptr->image, marker_corners, marker_ids);
64     // Checking if the marker_corners array is empty, if it is, we skip it (keep in mind the
65     // dimensions of the marker_corners array is Nx4, where N is the number of tags detected)
66
67     // Publish the marker corners to the topic "aruco/markers_loc"
68     operator_intent_msgs::marker_collection marker_collection;
69     for (unsigned long int i = 0; i < marker_ids.size(); i++)
70     {
71         operator_intent_msgs::marker marker;
72         marker.marker_id = marker_ids[i];
73         for (unsigned long int j = 0; j < 4; j++)
74         {
75             operator_intent_msgs::point_2dc point_2dc;
76             point_2dc.x = marker_corners[i][j].x;
77             point_2dc.y = marker_corners[i][j].y;
78             marker.corner_points[j] = point_2dc;
79         }
80         marker_collection.markers.push_back(marker);
81     }
82     marker_collection.header.stamp = ros::Time::now();
83     markers_loc_pub.publish(marker_collection);
84
85     // Update GUI Window
86     cv::imshow(OPENCV_WINDOW, cv_ptr->image);
87     cv::waitKey(3);
88 }
89
90 DetectAruco::DetectAruco(ros::NodeHandle nh, ros::NodeHandle pnh)
91 : it_(nh_)

```



```

92 {
93   sub_rgb_image_topic_ = "/camera/rgb/image_raw";
94   pub_topic_ = "/image_converter/output_video";
95   // Subscribe to input video feed and publish output video feed
96   image_sub_ = it_.subscribe(sub_rgb_image_topic_, 1,
97                             &DetectAruco::imageCallback, this);
98   image_pub_ = it_.advertise(pub_topic_, 1);
99   markers_loc_pub_ = nh_.advertise<operator_intent_msgs::marker_collection>("aruco/markers_loc", 1)
100   ;
101
102   cv::namedWindow(OPENCV_WINDOW);
103 }
104 DetectAruco::~DetectAruco()
105 {
106   cv::destroyWindow(OPENCV_WINDOW);
107 }
108
109 int main(int argc, char **argv)
110 {
111   NodeMain<DetectAruco>(argc, argv, "DetectArucoNode");
112 }

```

### 6.1.2 Points of Interest Data Generation Node

Thesis/Source code/calculate\_object\_distance\_and\_angle.cpp

```

1 #include <ros/ros.h>
2 #include <image_transport/image_transport.h>
3 #include <image_transport/subscriber_filter.h>
4 #include <cv_bridge/cv_bridge.h>
5 #include <sensor_msgs/image_encodings.h>
6 #include <sensor_msgs/Image.h>
7 #include <nav_msgs/Odometry.h>
8 #include <geometry_msgs/Quaternion.h>
9 #include <opencv2/imgproc/imgproc.hpp>
10 #include <opencv2/highgui/highgui.hpp>
11 #include <message_filters/subscriber.h>
12 #include <message_filters/synchronizer.h>
13 #include <message_filters/sync_policies/approximate_time.h>
14 #include <tf2/LinearMath/Quaternion.h>
15 #include <tf2/LinearMath/Matrix3x3.h>
16
17 // Custom msgs
18 #include <operator_intent_msgs/point_2dc.h>
19 #include <operator_intent_msgs/marker.h>
20 #include <operator_intent_msgs/marker_collection.h>
21 #include <operator_intent_msgs/marker_coordinates_with_distance.h>
22 #include <operator_intent_msgs/marker_coordinates_with_distance_collection.h>
23
24 #include <cmath>
25 #include <iostream>
26
27 // Node template
28 #include "node_template.cpp"
29

```

```

30 // Define Infinite (Using INT_MAX caused overflow problems)
31 #define INF 10000
32
33 /* Sources online mention that a real kinect V1 (old kinect) has a resolution of 640x480,
34 and the fov's are 62 degrees horizontally and 48.6 degrees vertically, resulting in
35 10.322580645 x 9.87654321 pixels per degree (roughly 10 x 10).
36 */
37 #define REAL_KINECT_V1_HORIZONTAL_FOV 62
38 #define REAL_KINECT_V1_VERTICAL_FOV 48.6
39
40 /* Found by searching for the kinect_camera.urdf.xacro file in husky/husky_description/urdf/accessories,
41 the horizontal FOV is 70 degrees
42 <horizontal_fov>${70.0*M_PI/180.0}</horizontal_fov>
43 */
44 #define KINECT_CAMERA_HORIZONTAL_FOV_DEG 70
45
46 /* Since the kinect_camera.urdf.xacro file has no mention of the vertical fov, we set it ourselves.
47 By changing the horizontal fov in the the kinect_camera.urdf.xacro, the vertical fov seems to change as
48 well.
49 Taking for granted that the ration of horizontal_fov/vertical_fov remains constant, as well as trusting the
50 source mentioning the
51 horizontal and vertical fov's online, the calculated vertical fov is:
52 */
53 #define KINECT_CAMERA_VERTICAL_FOV_DEG
54     KINECT_CAMERA_HORIZONTAL_FOV_DEG * REAL_KINECT_V1_VERTICAL_FOV /
55     REAL_KINECT_V1_HORIZONTAL_FOV
56
57 #define INVALID_DISTANCE -1
58
59 #define KINECT_CAMERA_WIDTH 640
60 #define KINECT_CAMERA_HEIGHT 480
61
62
63 class CalculateObjectDistanceAndAngle
64 {
65 private:
66     ros::NodeHandle nh_;
67     ros::NodeHandle pnh_;
68     image_transport::ImageTransport it_;
69     image_transport::Subscriber image_sub_;
70     image_transport::Publisher image_pub_;
71     image_transport::SubscriberFilter depth_image_sub_;
72     message_filters::Subscriber<operator_intent_msgs::marker_collection> marker_loc_sub_;
73     message_filters::Subscriber<nav_msgs::Odometry> odometry_sub_;
74     ros::Publisher marker_coordinates_with_distance_collection_pub_;
75
76     typedef union U_FloatParse
77     {
78         float float_data;
79         unsigned char byte_data[4];
80     } U_FloatConvert;
81
82     int readDepthData(cv::Point2i, sensor_msgs::ImageConstPtr depth_image);
83     double findAngleInRadiansFromCameraPointOfReference(cv::Point2i, char axis);
84     double findAngleInRadiansFromQuaternion(geometry_msgs::Quaternion &quaternion);

```

```

81 bool findPositionOfMarker(
82     double &distance,
83     double &angle_radians,
84     double &robot_angle_radians,
85     double robot_position_x,
86     double robot_position_y,
87     double &marker_position_x,
88     double &marker_position_y);
89
90 bool intersection(cv::Point2i o1, cv::Point2i p1, cv::Point2i o2, cv::Point2i p2, cv::Point2i &r);
91
92 public:
93     CalculateObjectDistanceAndAngle();
94     CalculateObjectDistanceAndAngle(ros::NodeHandle, ros::NodeHandle);
95     ~CalculateObjectDistanceAndAngle();
96
97 void callBack(
98     const operator_intent_msgs::marker_collectionConstPtr &marker_collection,
99     const sensor_msgs::ImageConstPtr &image,
100    const nav_msgs::OdometryConstPtr &odometry);
101 };
102
103 CalculateObjectDistanceAndAngle::CalculateObjectDistanceAndAngle(ros::NodeHandle nh, ros::
104     NodeHandle pnh)
105     : nh_(nh), pnh_(pnh_), it_(nh_), depth_image_sub_(it_, "camera/depth/image_raw", 1)
106     {
107     // Subscribe to input video feed and publish output video feed
108     marker_loc_sub_.subscribe(nh_, "aruco/markers_loc", 1);
109     odometry_sub_.subscribe(nh_, "husky_base_ground_truth", 1);
110     marker_coordinates_with_distance_collection_pub_ = nh_.advertise<operator_intent_msgs::
111         marker_coordinates_with_distance_collection>("aruco/
112         marker_coordinates_with_distance_collection", 1);
113
114     typedef message_filters::sync_policies::ApproximateTime<operator_intent_msgs::marker_collection,
115         sensor_msgs::Image, nav_msgs::Odometry> MySyncPolicy;
116     message_filters::Synchronizer<MySyncPolicy> sync(MySyncPolicy(10), marker_loc_sub_,
117         depth_image_sub_, odometry_sub_);
118     sync.registerCallback(boost::bind(&CalculateObjectDistanceAndAngle::callBack, this, _1, _2, _3));
119     ros::spin();
120     }
121
122 CalculateObjectDistanceAndAngle::~CalculateObjectDistanceAndAngle()
123     {
124     }
125
126 int CalculateObjectDistanceAndAngle::readDepthData(cv::Point2i intersection_point, sensor_msgs::
127     ImageConstPtr depth_image)
128     {
129     // If position is invalid
130     if ((intersection_point.y >= depth_image->height) || (intersection_point.x >= depth_image->width))
131         return INVALID_DISTANCE;
132     int index = (intersection_point.y * depth_image->step) + (intersection_point.x * (depth_image->step /
133         depth_image->width));
134     // If data is 4 byte floats (rectified depth image)
135     if ((depth_image->step / depth_image->width) == 4)
136     {

```

```

130 U_FloatConvert depth_data;
131 int i, endian_check = 1;
132 // If big endian
133 if ((depth_image->is_bigendian && (*(char *)&endian_check != 1)) || // Both big endian
134     (!depth_image->is_bigendian) && (*(char *)&endian_check == 1)) // Both lil endian
135 {
136     for (i = 0; i < 4; i++)
137         depth_data.byte_data[i] = depth_image->data[index + i];
138     // Make sure data is valid (check if NaN)
139     if (depth_data.float_data == depth_data.float_data)
140         return int(depth_data.float_data * 1000);
141     return INVALID_DISTANCE; // If depth data invalid
142 }
143 // else, one little endian, one big endian
144 for (i = 0; i < 4; i++)
145     depth_data.byte_data[i] = depth_image->data[3 + index - i];
146 // Make sure data is valid (check if NaN)
147 if (depth_data.float_data == depth_data.float_data)
148     return int(depth_data.float_data * 1000);
149     return INVALID_DISTANCE; // If depth data invalid
150 }
151 // Otherwise, data is 2 byte integers (raw depth image)
152 int temp_val;
153 // If big endian
154 if (depth_image->is_bigendian)
155     temp_val = (depth_image->data[index] << 8) + depth_image->data[index + 1];
156 // If little endian
157 else
158     temp_val = depth_image->data[index] + (depth_image->data[index + 1] << 8);
159 // Make sure data is valid (check if NaN)
160 if (temp_val == temp_val)
161     return temp_val;
162 return INVALID_DISTANCE; // If depth data invalid
163 }
164
165 double CalculateObjectDistanceAndAngle::findAngleInRadiansFromCameraPointOfReference(cv::
    Point2i intersection_point, char axis)
166 {
167     double angleInRadians = 0.0;
168     cv::Point2i middle_pixel(KINECT_CAMERA_WIDTH / 2, KINECT_CAMERA_HEIGHT / 2);
169     if (axis == 'H') {
170         double degrees_per_pixel = (double)KINECT_CAMERA_HORIZONTAL_FOV_DEG / (double)
            KINECT_CAMERA_WIDTH;
171         angleInRadians = (degrees_per_pixel * (middle_pixel.x - intersection_point.x)) * M_PI / 180;
172     } else if (axis == 'V') {
173         double degrees_per_pixel = (double)KINECT_CAMERA_VERTICAL_FOV_DEG / (double)
            KINECT_CAMERA_HEIGHT;
174         angleInRadians = (degrees_per_pixel * (middle_pixel.y - intersection_point.y)) * M_PI / 180;
175     }
176     return angleInRadians;
177 }
178
179 bool CalculateObjectDistanceAndAngle::findPositionOfMarker(
180     double &distance_mm,
181     double &angle_radians,
182     double &robot_angle_radians,

```

```

183     double robot_position_x,
184     double robot_position_y,
185     double &marker_position_x,
186     double &marker_position_y)
187     {
188     if (distance_mm != -1.0)
189     {
190     double consolidated_angle_radians = robot_angle_radians + angle_radians;
191     marker_position_x = robot_position_x + distance_mm / 1000 * cos(consolidated_angle_radians);
192     marker_position_y = robot_position_y + distance_mm / 1000 * sin(consolidated_angle_radians);
193     return true;
194     }
195     return false;
196     }
197
198 double CalculateObjectDistanceAndAngle::findAngleInRadiansFromQuaternion(geometry_msgs::
    Quaternion &quaternion)
199 {
200     tf2::Quaternion q(
201         quaternion.x,
202         quaternion.y,
203         quaternion.z,
204         quaternion.w);
205     tf2::Matrix3x3 m(q);
206     double roll, pitch, yaw;
207     m.getRPY(roll, pitch, yaw);
208     return yaw;
209 }
210
211 // Finds the intersection of two lines, or returns false.
212 // The lines are defined by (o1, p1) and (o2, p2).
213 bool CalculateObjectDistanceAndAngle::intersection(cv::Point2i o1, cv::Point2i p1, cv::Point2i o2, cv::
    Point2i p2, cv::Point2i &r)
214 {
215     cv::Point2i x = o2 - o1;
216     cv::Point2i d1 = p1 - o1;
217     cv::Point2i d2 = p2 - o2;
218
219     float cross = d1.x * d2.y - d1.y * d2.x;
220     if (abs(cross) < /*EPS*/ 1e-8)
221         return false;
222
223     double t1 = (x.x * d2.y - x.y * d2.x) / cross;
224     r = o1 + d1 * t1;
225     return true;
226 }
227
228 void CalculateObjectDistanceAndAngle::callBack(
229     const operator_intent_msgs::marker_collectionConstPtr &marker_collection,
230     const sensor_msgs::ImageConstPtr &image,
231     const nav_msgs::OdometryConstPtr &odometry)
232 {
233     cv_bridge::CvImagePtr cv_ptr;
234     try
235     {
236         cv_ptr = cv_bridge::toCvCopy(image, sensor_msgs::image_encodings::TYPE_32FC1);

```

```

237 }
238 catch (cv_bridge::Exception &e)
239 {
240     ROS_ERROR("cv_bridge exception: %s", e.what());
241     return;
242 }
243
244 // Set robot angle
245 geometry_msgs::Quaternion temp_quaternion = odometry->pose.pose.orientation;
246 double robot_angle = findAngleInRadiansFromQuaternion(temp_quaternion);
247
248 operator_intent_msgs::marker_coordinates_with_distance_collection
249     marker_coordinates_with_distance_collection;
250 // For each marker [0..n]:
251 for (unsigned long int i = 0; i < marker_collection->markers.size(); i++)
252 {
253     operator_intent_msgs::marker_coordinates_with_distance marker_coordinates_with_distance;
254     cv::Point2i marker_points[4];
255     cv::Point2i intersection_point;
256     for (int j = 0; j < (sizeof(marker_points) / sizeof(marker_points[0])); j++)
257     {
258         marker_points[j] = cv::Point2i(
259             marker_collection->markers[i].corner_points[j].x,
260             marker_collection->markers[i].corner_points[j].y);
261     }
262     // If the four corners of the marker intersect:
263     if (intersection(marker_points[0], marker_points[2], marker_points[1], marker_points[3],
264         intersection_point))
265     {
266         // Assign the marker's ID
267         marker_coordinates_with_distance.marker_id = marker_collection->markers[i].marker_id;
268         // Write the depth data
269         double distance_to_point = readDepthData(intersection_point, image);
270         if (distance_to_point == -1.0) {
271             marker_coordinates_with_distance.distance_mm = distance_to_point;
272         } else {
273             marker_coordinates_with_distance.distance_mm =
274                 distance_to_point / cos(findAngleInRadiansFromCameraPointOfReference(intersection_point, 'V')
275                 );}
276
277         // Write the angle in radians
278         marker_coordinates_with_distance.angle_radians =
279             findAngleInRadiansFromCameraPointOfReference(intersection_point, 'H');
280
281         double marker_x, marker_y;
282         // Find marker position in world
283         if (findPositionOfMarker(
284             marker_coordinates_with_distance.distance_mm,
285             marker_coordinates_with_distance.angle_radians,
286             robot_angle,
287             odometry->pose.pose.position.x,
288             odometry->pose.pose.position.y,
289             marker_x,
290             marker_y))
291         {
292             marker_coordinates_with_distance.marker_world_x = marker_x;

```

```

289     marker_coordinates_with_distance.marker_world_y = marker_y;
290 }
291 // Assign the center of the marker
292 marker_coordinates_with_distance.marker_pixel_x = intersection_point.x;
293 marker_coordinates_with_distance.marker_pixel_y = intersection_point.y;
294
295     marker_coordinates_with_distance_collection.markers.push_back(marker_coordinates_with_distance)
296     ;
297 }
298 }
299 marker_coordinates_with_distance_collection.header.stamp = ros::Time::now();
300 marker_coordinates_with_distance_collection.camera_height = image->height;
301 marker_coordinates_with_distance_collection.camera_width = image->width;
302
303 marker_coordinates_with_distance_collection_pub.publish(
304     marker_coordinates_with_distance_collection);
305 }
306
307 int main(int argc, char **argv)
308 {
309     NodeMain<CalculateObjectDistanceAndAngle>(argc, argv, "CalculateObjectDistanceAndAngleNode");
310 }

```

### 6.1.3 Data Persistence Node

Thesis/Source code/persistent\_markers.cpp

```

1 #include <ros/ros.h>
2
3 #include <operator_intent_msgs/point_2dc.h>
4 #include <operator_intent_msgs/marker.h>
5 #include <operator_intent_msgs/marker_collection.h>
6 #include <operator_intent_msgs/marker_coordinates_with_distance.h>
7 #include <operator_intent_msgs/marker_coordinates_with_distance_collection.h>
8 #include <nav_msgs/Odometry.h>
9 #include <geometry_msgs/Quaternion.h>
10 #include <geometry_msgs/Vector3.h>
11 #include <message_filters/subscriber.h>
12 #include <message_filters/synchronizer.h>
13 #include <message_filters/sync_policies/approximate_time.h>
14 #include <tf2/LinearMath/Quaternion.h>
15 #include <tf2/LinearMath/Matrix3x3.h>
16
17 // Node template
18 #include "node_template.cpp"
19
20 struct Point
21 {
22     double x;
23     double y;
24 };
25
26 class PersistentMarkers
27 {
28
29 private:

```

```

30 ros::NodeHandle nh_;
31 ros::NodeHandle pnh_;
32 operator_intent_msgs::marker_coordinates_with_distance_collection persistent_marker_collection;
33 // ros::Subscriber marker_collection_sub_;
34 message_filters::Subscriber<operator_intent_msgs::marker_coordinates_with_distance_collection>
   marker_collection_sub_;
35 message_filters::Subscriber<nav_msgs::Odometry> odometry_sub_;
36 ros::Publisher persistent_marker_collection_pub_;
37 void callBack(const operator_intent_msgs::marker_coordinates_with_distance_collectionConstPtr &
   marker_coordinates_with_distance_collection_msg,
38               const nav_msgs::OdometryConstPtr &odometry_msg);
39 double calculateDistanceOfTwoPoints(Point &p1, Point &p2);
40 void updatePersistentMarkerCollectionFromRobotPose(nav_msgs::OdometryConstPtr odometry_msg);
41 double calculateApproachingSpeed(Point robot_position, Point marker_position, geometry_msgs::
   Vector3 robot_linear_velocity);
42 double calculateAngleOfMarkerFromRobotFrameOfReference(Point robot_position, Point
   marker_position, nav_msgs::Odometry robot_odometry);
43
44 double dotProduct(Point vector_1, Point vector_2);
45 double determinant(Point vector_1, Point vector_2);
46
47 public:
48   PersistentMarkers(ros::NodeHandle nh, ros::NodeHandle pnh);
49 };
50
51 PersistentMarkers::PersistentMarkers(ros::NodeHandle nh, ros::NodeHandle pnh) : nh_(nh), pnh_(pnh)
52 {
53   persistent_marker_collection_pub_ = nh.advertise<operator_intent_msgs::
   marker_coordinates_with_distance_collection>("/aruco/
   persistent_marker_coordinates_with_distance_collection", 1);
54   marker_collection_sub_.subscribe(nh_, "/aruco/marker_coordinates_with_distance_collection", 1);
55   odometry_sub_.subscribe(nh_, "/husky_base_ground_truth", 1);
56   typedef message_filters::sync_policies::ApproximateTime<operator_intent_msgs::
   marker_coordinates_with_distance_collection, nav_msgs::Odometry> MySyncPolicy;
57   message_filters::Synchronizer<MySyncPolicy> sync(MySyncPolicy(10), marker_collection_sub_,
   odometry_sub_);
58   sync.registerCallback(boost::bind(&PersistentMarkers::callBack, this, _1, _2));
59   ros::spin();
60 }
61
62 double PersistentMarkers::calculateDistanceOfTwoPoints(Point &p1, Point &p2)
63 {
64   return sqrt((p2.x - p1.x) * (p2.x - p1.x) + (p2.y - p1.y) * (p2.y - p1.y));
65 }
66
67 double PersistentMarkers::calculateApproachingSpeed(Point robot_position, Point marker_position,
   geometry_msgs::Vector3 robot_linear_velocity)
68 {
69   // Step 1: Calculate the direction vector for the marker
70   // the Point struct was used for the vector for simplicity
71   Point direction_vector;
72   direction_vector.x = marker_position.x - robot_position.x;
73   direction_vector.y = marker_position.y - robot_position.y;
74
75   // Step 2: Normalize the vector

```



```

76     double vector_magnitude = sqrt(direction_vector.x * direction_vector.x + direction_vector.y *
77         direction_vector.y);
77     direction_vector.x = direction_vector.x / vector_magnitude;
78     direction_vector.y = direction_vector.y / vector_magnitude;
79
80     // Step 3: Find the dot product between the direction vector and the linear velocity vector
81     double approaching_speed_meters_per_sec = direction_vector.x * robot_linear_velocity.x +
82         direction_vector.y * robot_linear_velocity.y;
83
84     return approaching_speed_meters_per_sec;
85 }
86
87 double PersistentMarkers::determinant(Point vector_1, Point vector_2)
88 {
89     return (vector_1.x * vector_2.y - vector_1.y * vector_2.x);
90 }
91
92 double PersistentMarkers::dotProduct(Point vector_1, Point vector_2)
93 {
94     return (vector_1.x * vector_2.x + vector_1.y * vector_2.y);
95 }
96
97 double PersistentMarkers::calculateAngleOfMarkerFromRobotFrameOfReference(Point robot_position,
98     Point marker_position, nav_msgs::Odometry robot_odometry)
99 {
100     double marker_x_difference = marker_position.x - robot_position.x;
101     double marker_y_difference = marker_position.y - robot_position.y;
102     double marker_vector_magnitude = sqrt(marker_x_difference * marker_x_difference +
103         marker_y_difference * marker_y_difference);
104     Point marker_world_angle_from_robot_position_normalized = {marker_x_difference /
105         marker_vector_magnitude, marker_y_difference / marker_vector_magnitude};
106
107     tf2::Quaternion q(
108         robot_odometry.pose.pose.orientation.x,
109         robot_odometry.pose.pose.orientation.y,
110         robot_odometry.pose.pose.orientation.z,
111         robot_odometry.pose.pose.orientation.w);
112
113     tf2::Matrix3x3 m(q);
114     double roll, pitch, yaw;
115     m.getRPY(roll, pitch, yaw);
116
117     Point robot_world_direction_vector_normalized = {cos(yaw), sin(yaw)};
118
119     return atan2(
120         determinant(robot_world_direction_vector_normalized,
121             marker_world_angle_from_robot_position_normalized),
122         dotProduct(robot_world_direction_vector_normalized,
123             marker_world_angle_from_robot_position_normalized)
124     );
125 }
126
127 void PersistentMarkers::updatePersistentMarkerCollectionFromRobotPose(nav_msgs::OdometryConstPtr
128     odometry_msg)
129 {

```

```

124 // Set robot's position in the world
125 Point robot_position = {odometry_msg->pose.pose.position.x, odometry_msg->pose.pose.position.y};
126 for (int i = 0; i < persistent_marker_collection.markers.size(); i++)
127 {
128     // Set marker's position in the world
129     Point marker_position = {persistent_marker_collection.markers[i].marker_world_x,
130                             persistent_marker_collection.markers[i].marker_world_y};
131
132     // Update approach_speed
133     persistent_marker_collection.markers[i].approaching_speed_meters_per_sec =
134     calculateApproachingSpeed(
135         robot_position, marker_position,
136         odometry_msg->twist.twist.linear);
137
138     // Update distance
139     persistent_marker_collection.markers[i].distance_mm = calculateDistanceOfTwoPoints(
140     robot_position, marker_position) * 1000;
141
142     // Update angle
143     persistent_marker_collection.markers[i].angle_radians =
144     calculateAngleOfMarkerFromRobotFrameOfReference(robot_position, marker_position, *
145     odometry_msg);
146 }
147 return;
148 }
149
150 void PersistentMarkers::callBack(
151 const operator_intent_msgs::marker_coordinates_with_distance_collectionConstPtr &
152     marker_coordinates_with_distance_collection_msg,
153 const nav_msgs::OdometryConstPtr &odometry_msg)
154 {
155     persistent_marker_collection.header = marker_coordinates_with_distance_collection_msg->header;
156     persistent_marker_collection.camera_height = marker_coordinates_with_distance_collection_msg->
157     camera_height;
158     persistent_marker_collection.camera_width = marker_coordinates_with_distance_collection_msg->
159     camera_width;
160
161     // If the marker_coordinates_with_distance_collection_msg isn't empty aka no markers detected
162     if (marker_coordinates_with_distance_collection_msg->markers.size() != 0)
163     {
164         // For each marker in marker_coordinates_with_distance_collection_msg
165         for (int i = 0; i < marker_coordinates_with_distance_collection_msg->markers.size(); i++)
166         {
167             // Bool value to check if a match has been found
168             bool found = false;
169             // For each marker in persistent_marker_collection
170             for (int j = 0; j < persistent_marker_collection.markers.size(); j++)
171             {
172                 // If the marker_id in the marker_coordinates_with_distance_collection_msg matches the one in
173                 // persistent_marker_collection
174                 if (marker_coordinates_with_distance_collection_msg->markers[i].marker_id ==
175                     persistent_marker_collection.markers[j].marker_id)
176                 {
177                     // And if the marker isn't out of bounds for the depth sensor
178                     if (!(marker_coordinates_with_distance_collection_msg->markers[i].marker_world_x == 0.0
179                         && marker_coordinates_with_distance_collection_msg->markers[i].marker_world_y == 0.0 &&

```

```

169     marker_coordinates_with_distance_collection_msg->markers[i].distance_mm == -1.0))
170     {
171         // Update the marker in the persistent_marker_collection with the one in the
172         marker_coordinates_with_distance_collection_msg
173         persistent_marker_collection.markers[j] =
174         marker_coordinates_with_distance_collection_msg->markers[i];
175     }
176     // Set the found to true and break
177     found = true;
178     break;
179 }
180 // If a match has not been found:
181 if (!found)
182 {
183     // Add a new marker to the persistent_marker_collection
184     operator_intent_msgs::marker_coordinates_with_distance temp_marker =
185     marker_coordinates_with_distance_collection_msg->markers[i];
186     persistent_marker_collection.markers.push_back(temp_marker);
187 }
188 }
189 updatePersistentMarkerCollectionFromRobotPose(odometry_msg);
190 // Publish the current state of the persistent_marker_collection
191 if (persistent_marker_collection.markers.size() != 0)
192 {
193     persistent_marker_collection_pub.publish(persistent_marker_collection);
194 }
195 }
196 int main(int argc, char **argv)
197 {
198     NodeMain<PersistentMarkers>(argc, argv, "PersistentMarkersNode");
199 }

```

### 6.1.4 Operator Intent Inference Node

Thesis/Source code/goal\_inference.py

```

1  #!/usr/bin/env python3
2
3  import os
4  import json
5  import rospy
6  import rospkg
7  import std_msgs.msg
8  from operator_intent_msgs.msg import operator_intent_inference
9  import pandas as pd
10 import joblib
11 from collections import Counter
12 from sklearn.ensemble import RandomForestClassifier
13 from operator_intent_msgs.msg import marker_coordinates_with_distance_collection
14
15
16 class GoalInference:
17     def __init__(self, markers_set):

```

```

18 self.markers_set = markers_set
19 # Print the configuration for the markers to look for
20 print("Markers set: ", self.markers_set)
21 rospack = rospkg.RosPack()
22 self.model = RandomForestClassifier
23 # Loading the model saved in "${detect_poi}/ml_models/", it depends on the number of targets
24 # declared in the config.json file
25 self.model = joblib.load(os.path.join(rospack.get_path(
26     "detect_poi"), "ml_models/intent_inference_model_{}_targets.sav".format(len(self.markers_set)))
27 )
28
29 # Initialize ros stuff
30 rospy.init_node('operator_intent_inference_node', anonymous=True)
31 self.persistent_marker_sub = rospy.Subscriber("/aruco/
32     persistent_marker_coordinates_with_distance_collection",
33     marker_coordinates_with_distance_collection, self.callback, queue_size=1)
34 self.operator_intent_pub = rospy.Publisher("/operator_intent_inference", operator_intent_inference,
35     queue_size=1)
36 rospy.spin()
37
38 def is_subset(self, l1, l2):
39     c1, c2 = Counter(l1), Counter(l2)
40     return not c1 - c2
41
42 def model_prediction(self):
43     return self.model.predict()
44
45 def callback(self, persistent_marker_collection):
46     current_state_df = pd.DataFrame()
47
48     persistent_marker_collection_id_list = list()
49     for i in persistent_marker_collection.markers:
50         persistent_marker_collection_id_list.append(i.marker_id)
51
52     if self.is_subset(self.markers_set, persistent_marker_collection_id_list):
53         # Logic for marshalling the data in the appropriate format to pass to the model for prediction i.e
54         self.model.predict()
55         # For every marker in the persistent collection
56         for i in persistent_marker_collection.markers:
57             # If the marker_id is in the predefines set of markers we seek
58             if i.marker_id in self.markers_set:
59                 current_state_df["marker_{}_distance".format(i.marker_id)] = [
60                     i.distance_mm]
61                 current_state_df["marker_{}_angle_radians".format(i.marker_id)] = [
62                     i.angle_radians]
63                 current_state_df["marker_{}_approach_speed".format(i.marker_id)] = [
64                     i.approaching_speed_meters_per_sec]
65
66         # Creating the input data labels from the markers_set
67         input_data_labels = list()
68
69         for i in self.markers_set:
70             input_data_labels.append("marker_{}_distance".format(i))
71             input_data_labels.append("marker_{}_angle_radians".format(i))
72             input_data_labels.append("marker_{}_approach_speed".format(i))

```

```
69
70     # Move the dataframe columns to the appropriate place
71     current_state_df = current_state_df[input_data_labels]
72
73     # Make the prediction and get the probability for it
74     prediction = self.model.predict(current_state_df)
75     prediction_probability = self.model.predict_proba(current_state_df).max()
76
77     # Construct the message to be sent
78     operator_intent_inference_msg = operator_intent_inference()
79     # Fill the fields of the message
80     h = std_msgs.msg.Header()
81     h.stamp = rospy.Time.now()
82     operator_intent_inference_msg.header = h
83     operator_intent_inference_msg.prediction = str(prediction)
84     operator_intent_inference_msg.prediction_probability = prediction_probability
85
86     # Publish the message
87     self.operator_intent_pub.publish(operator_intent_inference_msg)
88
89
90
91
92 if __name__ == "__main__":
93     rospack = rospkg.RosPack()
94     with open(os.path.join(rospack.get_path("detect_poi"), "config/config.json")) as f:
95         json_content = json.loads(f.read())
96         markers_set = list()
97         for marker in json_content["markers"]:
98             markers_set.append(marker["id"])
99         markers_set.sort()
100     try:
101         gi = GoalInference(markers_set)
102     except rospy.ROSInterruptException:
103         pass
```



## Βιβλιογραφικές Αναφορές

- [1] Keiji Nagatani κ.ά. «Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots». Στο: *Journal of Field Robotics* 30 (1 Ιαν. 2013), σσ. 44–63. ISSN: 15564959. DOI: [10.1002/rob.21439](https://doi.org/10.1002/rob.21439). URL: <https://onlinelibrary.wiley.com/doi/10.1002/rob.21439>.
- [2] Jennifer Casper και Robin Roberson Murphy. «Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center». Στο: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 33.3 (2003). ISSN: 10834419. DOI: [10.1109/TSMCB.2003.811794](https://doi.org/10.1109/TSMCB.2003.811794).
- [3] Jeffrey Delmerico κ.ά. «The current state and future outlook of rescue robotics». Στο: *Journal of Field Robotics* 36.7 (2019). ISSN: 15564967. DOI: [10.1002/rob.21887](https://doi.org/10.1002/rob.21887).
- [4] Robin R. Murphy και Jennifer L. Burke. «Up from the rubble: Lessons learned about HRI from search and rescue». Στο: *Proceedings of the Human Factors and Ergonomics Society* 49 (3 Σεπτ. 2005), σσ. 437–441. ISSN: 10711813. DOI: [10.1177/154193120504900347](https://doi.org/10.1177/154193120504900347). URL: <http://journals.sagepub.com/doi/10.1177/154193120504900347>.
- [5] Geert Jan M. Kruijff κ.ά. «Rescue robots at earthquake-hit Mirandola, Italy: A field report». Στο: 2012. DOI: [10.1109/SSRR.2012.6523866](https://doi.org/10.1109/SSRR.2012.6523866).
- [6] Manolis Chiou, Rustam Stolkin, Goda Biaksaite, Nick Hawes, Kimron L. Shapiro και Timothy S. Harrison. «Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot». Στο: *IEEE International Conference on Intelligent Robots and Systems*. Τόμ. 2016–November. 2016. DOI: [10.1109/IRoS.2016.7759527](https://doi.org/10.1109/IRoS.2016.7759527).
- [7] Manolis Chiou, Nick Hawes και Rustam Stolkin. *Mixed-initiative variable autonomy for remotely operated mobile robots*. 2019.
- [8] Siddarth Jain και Brenna Argall. «Recursive Bayesian Human Intent Recognition in Shared-Control Robotics». Στο: IEEE, Οκτ. 2018, σσ. 3905–3912. ISBN: 9781538680940. DOI: [10.1109/IRoS.2018.8593766](https://doi.org/10.1109/IRoS.2018.8593766). URL: <https://ieeexplore.ieee.org/document/8593766/>.
- [9] Dimitris Panagopoulos, Giannis Petousakis, Rustam Stolkin, Grigoris Nikolaou και Manolis Chiou. «A Bayesian-Based Approach to Human Operator Intent Recognition in Remote Mobile Robot Navigation». Στο: IEEE, Οκτ. 2021, σσ. 125–131. ISBN: 9781665442077. DOI: [10.1109/SMC52423.2021.9658942](https://doi.org/10.1109/SMC52423.2021.9658942). URL: <https://ieeexplore.ieee.org/document/9658942/>.
- [10] Andrew J. Calder, Andrew D. Lawrence, Jill Keane, Sophie K. Scott, Adrian M. Owen, Ingrid Christoffels και Andrew W. Young. «Reading the mind from eye gaze». Στο: *Neuropsychologia* 40 (8 2002). ISSN: 00283932. DOI: [10.1016/S0028-3932\(02\)00008-8](https://doi.org/10.1016/S0028-3932(02)00008-8).
- [11] Neeta Nathani και Abhishek Singh. *Foundations of Machine Learning*. CRC Press, Οκτ. 2021. DOI: [10.1201/9781003104445-4](https://doi.org/10.1201/9781003104445-4). URL: <https://www.taylorfrancis.com/books/9781003104445/chapters/10.1201/9781003104445-4>.

- [12] Hanwool Woo, Yonghoon Ji, Hitoshi Kono, Yusuke Tamura, Yasuhide Kuroda, Takashi Sugano, Yasunori Yamamoto, Atsushi Yamashita και Hajime Asama. «Automatic Detection Method of Lane-Changing Intentions Based on Relationship with Adjacent Vehicles Using Artificial Potential Fields». Στο: *International Journal of Automotive Engineering* 7 (4 2016), σ. 20164094. ISSN: 2185-0984. DOI: [10.20485/jjsaeijae.7.4\\_127](https://doi.org/10.20485/jjsaeijae.7.4_127). URL: [https://www.jstage.jst.go.jp/article/jjsaeijae/7/4/7\\_20164094/\\_article](https://www.jstage.jst.go.jp/article/jjsaeijae/7/4/7_20164094/_article).
- [13] Hanwool Woo, Yonghoon Ji, Hitoshi Kono, Yusuke Tamura, Yasuhide Kuroda, Takashi Sugano, Yasunori Yamamoto, Atsushi Yamashita και Hajime Asama. «Lane-Change Detection Based on Vehicle-Trajectory Prediction». Στο: *IEEE Robotics and Automation Letters* 2 (2 2017). ISSN: 23773766. DOI: [10.1109/LRA.2017.2660543](https://doi.org/10.1109/LRA.2017.2660543).
- [14] Liang Yan, Xiaoshan Gao, Xiongjie Zhang και Suokui Chang. «Human-robot collaboration by intention recognition using deep LSTM neural network». Στο: 2019, σσ. 1390–1396. ISBN: 9781728103112. DOI: [10.1109/FPM45753.2019.9035907](https://doi.org/10.1109/FPM45753.2019.9035907).
- [15] Dongho Choi και Sangsun Lee. «Comparison of Machine Learning Algorithms for Predicting Lane Changing Intent». Στο: *International Journal of Automotive Technology* 22 (2 2021). Grigoris, σσ. 507–518. ISSN: 19763832. DOI: [10.1007/s12239-021-0047-x](https://doi.org/10.1007/s12239-021-0047-x).
- [16] R. Izquierdo, I. Parra, J. Munoz-Bulnes, D. Fernandez-Llorca και M. A. Sotelo. «Vehicle trajectory and lane change prediction using ANN and SVM classifiers». Στο: τόμ. 2018-March. 2018. DOI: [10.1109/ITSC.2017.8317838](https://doi.org/10.1109/ITSC.2017.8317838).
- [17] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf και Jan Peters. «Probabilistic movement modeling for intention inference in human-robot interaction». Στο: τόμ. 32. 2013, σσ. 841–858. DOI: [10.1177/0278364913478447](https://doi.org/10.1177/0278364913478447).
- [18] Carl Edward Rasmussen και C K I Williams. *Gaussian processes for machine learning*. 2006. Τόμ. 38. 2006.
- [19] Yu Sian Jiang, Garrett Warnell και Peter Stone. «Inferring user intention using gaze in vehicles». Στο: The application obtains the gaze point of the human driver, the object bounding boxes from the driving scene, then uses a dynamic Markov Random Field (MRF) model and energy minimization to infer user intent. ACM, Οκτ. 2018, σσ. 298–306. ISBN: 9781450356923. DOI: [10.1145/3242969.3243018](https://doi.org/10.1145/3242969.3243018). URL: <https://dl.acm.org/doi/10.1145/3242969.3243018>.
- [20] Tom Carlson και Yiannis Demiris. «Human-wheelchair collaboration through prediction of intention and adaptive assistance». Στο: 2008, σσ. 3926–3931. ISBN: 9781424416479. DOI: [10.1109/ROBOT.2008.4543814](https://doi.org/10.1109/ROBOT.2008.4543814).
- [21] *ROS/Introduction*. URL: <http://wiki.ros.org/ROS/Introduction>.
- [22] *Husky UGV - Outdoor Field Research Robot by Clearpath*. URL: <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>.
- [23] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas και M. J. Marín-Jiménez. «Automatic generation and detection of highly reliable fiducial markers under occlusion». Στο: *Pattern Recognition* 47.6 (2014). ISSN: 00313203. DOI: [10.1016/j.patcog.2014.01.005](https://doi.org/10.1016/j.patcog.2014.01.005). URL: <https://www.sciencedirect.com/science/article/pii/S0031320314000235>.
- [24] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas και Rafael Medina-Carnicer. «Speeded up detection of squared fiducial markers». Στο: *Image and Vision Computing* 76 (2018). ISSN: 02628856. DOI: [10.1016/j.imavis.2018.05.004](https://doi.org/10.1016/j.imavis.2018.05.004).
- [25] Leo Breiman. «Random forests». Στο: *Machine Learning* 45 (1 2001), σσ. 5–32. ISSN: 08856125. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).



## Πρόσθετη Βιβλιογραφία

- [26] Joachim De Greeff, Tina Mioch, Willeke Van Vught, Koen Hindriks, Mark A. Neerincx και Ivana Kruijff-Korbayová. «Persistent Robot-Assisted Disaster Response». Στο: *ACM/IEEE International Conference on Human-Robot Interaction*. 2018. DOI: [10.1145/3173386.3177049](https://doi.org/10.1145/3173386.3177049).
- [27] G. Ajay Kumar, Jin Hee Lee, Jongrak Hwang, Jaehyeong Park, Sung Hoon Youn και Soon Kwon. «LiDAR and camera fusion approach for object distance estimation in self-driving vehicles». Στο: *Symmetry* 12.2 (2020). ISSN: 20738994. DOI: [10.3390/sym12020324](https://doi.org/10.3390/sym12020324).
- [28] Qi Liu, Zirui Li, Shihua Yuan, Yuzheng Zhu και Xueyuan Li. «Review on Vehicle Detection Technology for Unmanned Ground Vehicles». Στο: *Sensors* 21.4 (Φεβ. 2021), σ. 1354. ISSN: 1424-8220. DOI: [10.3390/s21041354](https://doi.org/10.3390/s21041354). URL: <https://www.mdpi.com/1424-8220/21/4/1354>.
- [29] Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser και Victor Hugo C. de Albuquerque. «Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions». Στο: *IEEE Transactions on Intelligent Transportation Systems* 22.7 (Ιούλ. 2021), σσ. 4316–4336. ISSN: 1524-9050. DOI: [10.1109/TITS.2020.3032227](https://doi.org/10.1109/TITS.2020.3032227). URL: <https://ieeexplore.ieee.org/document/9284628/>.
- [30] Ming Gao, Jan Oberlander, Thomas Schamm και J. Marius Zöllner. «Shared autonomy for assisted mobile robot teleoperation by recognizing operator intention as contextual task». Στο: 2014, σσ. 82–87. ISBN: 9781479973965. DOI: [10.1109/ROBIO.2014.7090311](https://doi.org/10.1109/ROBIO.2014.7090311).
- [31] Siddarth Jain και Brenna Argall. «Probabilistic Human Intent Recognition for Shared Autonomy in Assistive Robotics». Στο: *ACM Transactions on Human-Robot Interaction* 9 (1 Ιαν. 2020), σσ. 1–23. ISSN: 2573-9522. DOI: [10.1145/3359614](https://doi.org/10.1145/3359614). URL: <https://dl.acm.org/doi/10.1145/3359614>.
- [32] Karan Khokar, Redwan Alqasemi, Sudeep Sarkar, Kyle Reed και Rajiv Dubey. «A novel telerobotic method for human-in-the-loop assisted grasping based on intention recognition». Στο: IEEE, Μάι. 2014, σσ. 4762–4769. ISBN: 978-1-4799-3685-4. DOI: [10.1109/ICRA.2014.6907556](https://doi.org/10.1109/ICRA.2014.6907556). URL: <http://ieeexplore.ieee.org/document/6907556/>.
- [33] Reuben M. Aronson και Henny Admoni. «Eye gaze for assistive manipulation». Στο: 2020, σσ. 552–554. ISBN: 9781450370578. DOI: [10.1145/3371382.3377434](https://doi.org/10.1145/3371382.3377434).
- [34] Tarek Taha, Jaime Valls Miró και Gamini Dissanayake. «POMDP-based long-term user intention prediction for wheelchair navigation». Στο: 2008, σσ. 3920–3925. ISBN: 9781424416479. DOI: [10.1109/ROBOT.2008.4543813](https://doi.org/10.1109/ROBOT.2008.4543813).
- [35] Graeme Best και Robert Fitch. «Bayesian intention inference for trajectory prediction with an unknown goal destination». Στο: τόμ. 2015-Decem. 2015, σσ. 5817–5823. ISBN: 9781479999941. DOI: [10.1109/IROS.2015.7354203](https://doi.org/10.1109/IROS.2015.7354203).
- [36] Deepak Gopinath, Siddarth Jain και Brenna D. Argall. «Human-in-the-loop optimization of shared autonomy in assistive robotics». Στο: *IEEE Robotics and Automation Letters* 2 (1 2017), σσ. 247–254. ISSN: 23773766. DOI: [10.1109/LRA.2016.2593928](https://doi.org/10.1109/LRA.2016.2593928).

- [37] Anca D. Dragan και Siddhartha S. Srinivasa. «A policy-blending formalism for shared control». Στο: τόμ. 32. 2013, σσ. 790–805. DOI: [10.1177/0278364913490324](https://doi.org/10.1177/0278364913490324).
- [38] Alexander Huntemann, Eric Demeester, Emmanuel Vander Poorten, Hendrik Van Brussel και Joris De Schutter. «Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model». Στο: IEEE, Μάι. 2013, σσ. 4376–4383. ISBN: 9781467356411. DOI: [10.1109/ICRA.2013.6631197](https://doi.org/10.1109/ICRA.2013.6631197). URL: <http://ieeexplore.ieee.org/document/6631197/>.
- [39] Anca Dragan και Siddhartha Srinivasa. «Formalizing Assistive Teleoperation». Στο: τόμ. 8. Robotics: Science και Systems Foundation, Ιούλ. 2012. ISBN: 9780262519687. DOI: [10.15607/RSS.2012.VIII.010](https://doi.org/10.15607/RSS.2012.VIII.010). URL: <http://www.roboticsproceedings.org/rss08/p10.pdf>.
- [40] Ming Gao, Jan Oberlander, Thomas Schamm και J. Marius Zollner. «Contextual task-aware shared autonomy for assistive mobile robot teleoperation». Στο: 2014. DOI: [10.1109/IROS.2014.6943023](https://doi.org/10.1109/IROS.2014.6943023).