



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

‘Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές’



Φοιτήτρια: Σέχου Φιόνα

ΑΜ: 04326

Επιβλέπουσα καθηγήτρια

Ζαχαρία Παρασκευή

Επ. Καθηγήτρια

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΣΕΠΤΕΜΒΡΙΟΣ 2021

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING

Diploma Thesis

“Using robotic simulator platforms for robotic applications”



Student: Shehu Fjona

Registration number: 04326

Supervisor

Zacharia Paraskevi

Assistant Professor

ATHENS-EGALEO, SEPTEMBER 2021

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

(Όνοματεπώνυμο), (βαθμίδα)	(Όνοματεπώνυμο), (βαθμίδα)	(Όνοματεπώνυμο), (βαθμίδα)
Παρασκευή Ζαχαρία Επίκουρη Καθηγήτρια	Κωνσταντίνος Στεργγίου Καθηγητής	Γεώργιος Χαμηλοθώρης Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΔΑ, τμήμα μηχανολόγων μηχανικών, Διπλωματική Εργασία, Σέχου Φιόνα

3

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Σέχου Φιόνα
Σεπτέμβριος, 2021

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

η κάτωθι υπογεγραμμένη Σέχου Φιόνα του Νίκο, με αριθμό μητρώου 51204326 φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ **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 μένα 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 διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι και έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση τής επιβλέπουσας καθηγήτριας.»

Η Δηλούσα
Σέχου Φιόνα



Περίληψη

Η παρούσα εργασία εστιάζει στις πλατφόρμες ρομποτικής προσομοίωσης και τις εφαρμογές που προέκυψαν βάση αυτών. Αρχικά, στις πρώτες τρεις ενότητες περιγράφονται εν συντομία σημαντικές έννοιες που αφορούν τόσο τον κλάδο της ρομποτικής όσο και τους ρομποτικούς προσομοιωτές. Ακολουθεί στην τέταρτη ενότητα η μελέτη και η ανάλυση των ρομποτικών προσομοιωτών Gazebo, RoboDK, Webots, Open-Rave και V-Rep, όπου και καταγράφονται στοιχεία για τις λειτουργίες, τις ιδιότητες, το γραφικό περιβάλλον, την αρχιτεκτονική δομή, τα ρομποτικά μοντέλα, τους αισθητήρες, τους ενεργοποιητές, αλλά και για τα μειονεκτήματα και πλεονεκτήματα που αυτά παρουσιάζουν. Ενώ, στη πέμπτη ενότητα, επιλέγονται δύο από τους προσομοιωτές που μελετήθηκαν και υλοποιούνται δυο εφαρμογές με βάση αυτούς. Ειδικότερα, η πρώτη εφαρμογή αναφέρεται σε ένα αυτοματοποιημένο όχημα, το οποίο έχει μοντελοποιηθεί στο προσομοιωμένο περιβάλλον V-Rep, και το οποίο βάση κάποιων αισθητήρων που φέρει, θα πρέπει να είναι σε θέση να επιτελεί δύο λειτουργίες. Πρώτον να ανιχνεύει τα τυχόν εμπόδια που συναντά κατά την διάρκεια εφαρμογής του και δεύτερον να αποφεύγει πιθανές συγκρούσεις αλλάζοντας κατεύθυνση. Όσον αφορά την δεύτερη εφαρμογή, αυτή επικεντρώνεται σε δύο ακίνητα ρομπότ τα οποία πρέπει να επιτύχουν ταυτόχρονα δύο εργασίες, μία τύπου pick and place, καθώς και μία τύπου paint. Τέλος παρουσιάζονται τα αποτελέσματα και τυχόν παρατηρήσεις.

Λέξεις Κλειδιά: Ρομποτική, ρομπότ, προσομοίωση, πλατφόρμες ρομποτικής προσομοίωσης

Abstract

This paper focuses on robotic simulation platforms and the applications that emerged from them. Initially, the first three sections briefly describe important concepts concerning both the field of robotics and robotic simulators. The fourth section follows the study and analysis of the Gazebo, RoboDK, Webots, Open-Rave and V-Rep robotic simulators, where data on the functions, properties, graphical environment, architectural structure, robotic models, sensors, actuators, but also on the disadvantages and advantages they present, are recorded. While, in the fifth section, two of the simulators studied are selected and two applications are implemented based on them. In particular, the first application refers to an automated vehicle, which has been modeled in the simulated V-rep environment, and which based on some sensors it carries, should be able to perform two functions. Firstly, to detect any obstacles encountered during its implementation and secondly to avoid possible collisions by changing direction. As for the second application, it focuses on two real estate robots that must simultaneously achieve two pick and place type jobs, as well as paint. Finally, the results and any observations are presented.

Key words: Robotics, robots, simulation, robotic simulation platforms

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

[_Toc82992867](#)

ΕΝΟΤΗΤΑ 1: Η ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΟΙ ΔΙΑΦΟΡΟΙ ΕΠΙΣΤΗΜΟΝΙΚΟΙ ΚΛΑΔΟΙ	12
1.1 Εισαγωγή	12
1.2 Ρομποτική Τεχνολογία.....	13
1.3 Μηχατρονική	14
1.4 Τεχνητή Νοημοσύνη	15
1.5 Βιομηχανική Ρομποτική	15
ΕΝΟΤΗΤΑ 2: ΤΑ ΒΙΟΜΗΧΑΝΙΚΑ ΡΟΜΠΟΤ	18
2.1 Εισαγωγή	18
2.2 Τι είναι το ρομποτ.....	18
2.3 Δομικά χαρακτηριστικά μιας ρομποτικής κατασκευής.....	19
2.3.1 Οι αρθρώσεις.....	21
2.3.2 Ο χώρος εργασίας του βραχίονα και του σημείου δράσης.....	22
2.3.3 Οι ενεργοποιητές	24
2.3.4 Ενεργό άκρο.....	24
2.4 Μέθοδοι ελέγχου κίνησης	25
2.5 Προγραμματισμός ρομπότ	27
ΕΝΟΤΗΤΑ 3: ΡΟΜΠΟΤΙΚΟΙ ΠΡΟΣΟΜΟΙΩΤΕΣ	29
3.1 Εισαγωγή	29
3.2 Ηλεκτρονικός υπολογιστής.....	29
3.3 Τι είναι ένας ρομποτικός προσομοιωτής	30
3.4 Τι χρειάζεται ένας προσομοιωτής ρομπότ για να λειτουργήσει.....	32
3.5 Πλεονεκτήματα- μειονεκτήματα ρομποτικών προσομοιωτών	32
ΕΝΟΤΗΤΑ 4: ΠΡΟΣΟΜΟΙΩΤΕΣ ΠΡΟΣ ΜΕΛΕΤΗ	36
4.1 Εισαγωγή	36
4.2 Προσομοιωτής Gazebo	36
4.2.1 Χαρακτηριστικά του προσομοιωτή:.....	38
4.2.2 Το Γραφικό Περιβάλλον:	43
4.2.3 Ο επεξεργαστής μοντέλων- Model Editor:	48
4.2.4 Αρχεία SVG:.....	51
4.2.5 Digital Elevation Model:	52
4.2.6 Μέρη του προσομοιωτή:	55

4.2.7 Gazebo – ROS:	57
4.2.8 Τα ρομποτικά συστήματα που υποστηρίζει το λογισμικό:.....	59
4.3 Προσομοιωτής RoboDK:.....	60
4.3.1 Χαρακτηριστικά του RoboDK:	61
4.3.2 Το Γραφικό περιβάλλον:	66
4.3.3 Το πλαίσιο του ρομπότ (Robot Panel):	70
4.3.4 Πρόσθετα-Plugins (add-ins):.....	72
4.3.5 Ρομποτικά συστήματα:	73
4.4 Webots προσομοιωτής:.....	83
4.4.1 Χαρακτηριστικά προσομοιωτή:	83
4.4.2 Προσομοίωση μέσω Webots:.....	86
4.4.3 Το γραφικό περιβάλλον του προσομοιωτή- GUI:.....	89
4.4.4 Δομή ενός τυπικού προγράμματος στο webots	92
4.4.5 Οι ενεργοποιητές του webots	92
4.4.6 Οι αισθητήρες του webots.....	93
4.4.7 Αντικείμενα προσομοίωσης	94
4.4.8 Ρομποτικά συστήματα:	95
4.5 OpenRAVE προσομοιωτής:	101
4.5.1 Ρομποτικά μοντέλα	101
4.5.2 Χαρακτηριστικά προσομοιωτή:	107
4.5.3 Η αρχιτεκτονική του λογισμικού:.....	107
4.6 V-REP προσομοιωτής:	110
4.6.1 Χαρακτηριστικά προσομοιωτή:	111
4.6.2 Το περιβάλλον προσομοίωσης.....	113
4.6.3 Ρομποτικά μοντέλα προσομοιωτή:	117
4.6.4 Αντικείμενα προσομοιωτή:	119
4.6.5 Πλεονεκτήματα-μειονεκτήματα	121
ΕΝΟΤΗΤΑ 5: Εφαρμογές σε πλατφόρμα ρομποτικής προσομοίωσης	123
5.1 Εισαγωγή	123
5.2 1 ^η Εφαρμογή (V-Rep).....	123
5.2.1 Το ρομπότ	123
5.2.2 Το περιβάλλον προσομοίωσης	125
5.2.3 Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή αυτή είναι	128

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

5.3 2^η Εφαρμογή (RoboDK)	132
5.3.1 Τα ρομπότ	133
5.3.2 Η μέθοδος επίλυσης του προβλήματος	133
5.3.3 Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή αυτή είναι	133
Βιβλιογραφία – Αναφορές – Διαδικτυακές πηγές	143

Εικονες

Εικόνα 1: Gazebo Logo	36
Εικόνα 2: Κάτω γραμμή εντολών (Gazebo).....	46
Εικόνα 3: Menu bar (Gazebo)	46
Εικόνα 4: Το γραφικό περιβάλλον (Gazebo).....	47
Εικόνα 5: Building Editor (Gazebo).....	48
Εικόνα 6: Joint (left) and link inspector(right) of Gazebo	50
Εικόνα 7: Model Tab (Gazebo)	51
Εικόνα 8: Αρχεία SVG (Gazebo).....	51
Εικόνα 9: Digital Elevation Model- δεδομένα raster (Gazebo).....	53
Εικόνα 10: Digital Elevation Model- QGIS (Gazebo).....	53
Εικόνα 11: Real Time Kinematic GPS (Gazebo)	54
Εικόνα 12: RoboDK Logo.....	60
Εικόνα 13: RoboDK Library	68
Εικόνα 14: Γραφικό Περιβάλλον RoboDK.....	70
Εικόνα 15: Robot Panel (RoboDK)	71
Εικόνα 16: Webots Logo	83
Εικόνα 17: Environments (Webots).....	88
Εικόνα 18: Nodes (Webots)	91
Εικόνα 19: Το Γραφικό Περιβάλλον (Webots)	91
Εικόνα 20: Open-Rave Logo	101
Εικόνα 21: barrett-wam Robot	102
Εικόνα 22: barrett-wam4 Robot (joint limits).....	102
Εικόνα 23: darpa-arm Robot.....	102
Εικόνα 24: exactdynamics-manusarmleft Robot	103
Εικόνα 25: kawada-hironx Robot	103
Εικόνα 26: kuka-kr30I16 Robot (joint limits)	103
Εικόνα 27: kuka-kr5-r650 Robot (joint limits).....	104
Εικόνα 28: kuka-kr5-r850 Robot (joint limits).....	104
Εικόνα 29: mitsubishi-pa10 Robot	105
Εικόνα 30: pr2-beta-static Robot	105
Εικόνα 31: schunk-lwa3 Robot.....	106
Εικόνα 32: unimation-pumaarm Robot.....	106
Εικόνα 33: Η αρχιτεκτονική του λογισμικού (Open-Rave)	108
Εικόνα 34: V-rep Logo.....	110
Εικόνα 35: Η αρχιτεκτονική δομή του V-rep.....	111
Εικόνα 36: Χαρακτηριστικά προσομοιωτή V-rep.....	113
Εικόνα 37: Το Γραφικό Περιβάλλον (V-rep).....	114
Εικόνα 38: application bar (V-rep).....	114
Εικόνα 39: menu bar (V-rep).....	114
Εικόνα 40: toolbars (V-rep).....	115
Εικόνα 41: model browser (V-rep)	116

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

Εικόνα 42: information text (V-rep)	116
Εικόνα 43: status bar(V-rep)	116
Εικόνα 44: Ρομποτικά μοντέλα προσομοιωτή- Κινούμενα(V-rep)	117
Εικόνα 45: Ρομποτικά μοντέλα προσομοιωτή- Ακίνητα(V-rep)	118
Εικόνα 46: Ρομποτικά Αντικείμενα προσομοιωτή(V-rep)	119
Εικόνα 47: Octrees	120
Εικόνα 48: Point Cloud.....	120
Εικόνα 49: 1 ^η Εφαρμογή (V-rep) - το ρομπότ	123
Εικόνα 50: 1 ^η Εφαρμογή (V-rep) - scene hierarchy του ρομπότ	124
Εικόνα 51: 1η Εφαρμογή (V-rep)-Το περιβάλλον προσομοίωσης	126
Εικόνα 52: 1η Εφαρμογή (V-rep)- scene hierarchy του περιβάλλοντος προσομοίωσης	127
Εικόνα 53: 1η Εφαρμογή (V-rep)-γραφική παράσταση	128
Εικόνα 54: 1η Εφαρμογή (V-rep)-κώδικας (μέρος α).....	129
Εικόνα 55: 1η Εφαρμογή (V-rep)-κώδικας (μέρος β)	130
Εικόνα 56: 1η Εφαρμογή (V-rep)-Proximity left code	131
Εικόνα 57: 1η Εφαρμογή (V-rep)-Proximity right code	132
Εικόνα 58: 2η Εφαρμογή (RoboDK).....	133

ΕΝΟΤΗΤΑ 1: Η ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΟΙ ΔΙΑΦΟΡΟΙ ΕΠΙΣΤΗΜΟΝΙΚΟΙ ΚΛΑΔΟΙ

1.1 Εισαγωγή

Οι Τεχνολογίες του σήμερα, καταλαμβάνουν σημαντικό ποσοστό της καθημερινότητας τα μας, καθώς όπως και παρατηρείται εμπλέκονται και δραστηριοποιούνται σε πολλούς τομείς της δουλειάς, της εκπαίδευσης, αλλά και της διασκέδαση μας. Μάλιστα, εξαιτίας αυτών, μπορούμε να διαμορφώσουμε και να αναπτύξουμε συστήματα για την παραγωγή έργου ή ισχύος ή και γενικότερα για την παραγωγή ενέργειας, δηλαδή, με αλλά λόγια για κάποιο πρακτικό όφελος. Συγκεκριμένα, πολλοί είναι αυτοί που χρησιμοποιούν τα τηλέφωνα, τους ηλεκτρονικούς υπολογιστές, τα μέσα μαζικής μεταφοράς και τα αυτοκίνητα τους, τα οποία συντρέχουν βασικές έως και περίπλοκες λειτουργίες, ενώ παράλληλα επιλύουν θέματα όπως είναι η επικοινωνία, η πληροφόρηση και η μεταφορά. Αρκετοί, επιπλέον, τις χρησιμοποιούν για να παρέχουν υπηρεσίες που θα διευκολύνουν είτε έως έναν βαθμό τους συνανθρώπους τους, είτε και ,τους ίδιους, τους εαυτούς τους. Τέτοιες αποτελεσματικές υπηρεσίες μπορεί να συναντήσουμε για παράδειγμα στην Ιατρική Επιστήμη, όπου οι γιατροί λόγο των εξελιγμένων τεχνολογικών μηχανημάτων μπορούν και να διεξάγουν πολυπληθέστερες έρευνες, να αναζητήσουν παραπλήσιες πληροφορίες και να μελετήσουν εκτενέστερα και σε μεγαλύτερο βάθος ένα θέμα, έτσι ώστε να μπορέσουν να σώσουν όσο το δυνατόν περισσότερες ζωές.

Αυτά, όμως, είναι αποτελέσματα μερικών από τα πολλά παραδείγματα που υποδηλώνουν την ανάπτυξη της Τεχνολογίας. Καθώς, η ανοδική πορεία, όπως και εμφάνιση της Τεχνολογία από μόνη της, βασίζεται σε άλλους, απλούστερους θα λέγαμε επιστημονικούς κλάδους. Σε αυτούς, λοιπόν, εντάσσονται τα Μαθηματικά, η Φυσική, η Χημεία, η Βιολογία, και γενικά όλες οι θετικές Επιστήμες. Χωρίς αυτό, όμως να σημαίνει, ότι στην διαμόρφωση του συμμετέχουν αποκλειστικά και μόνο αυτοί. Αντιθέτως, οι πιο εξειδικευμένοι τομείς της Επιστήμης, όπως είναι και αυτοί :

- Της Ηλεκτρολογίας
- Της Ηλεκτρονικής
- Της Πληροφορικής

- Της Μηχανολογίας
- Της Ρομποτικής
- Της Μηχατρονικής
- Της Βιοχημείας
- Της Βιοϊατρικής Επιστήμης
- Της Οικονομικής Επιστήμης, καθώς και πολλών άλλων.

Διαδραματίζουν και τον πιο καθοριστικό παράγοντα για την εφαρμογή της Τεχνολογίας, καθώς βάση αυτών μπορούμε να ενισχύσουμε τις έρευνες, να επεκτείνουμε τις γνώσεις μας, να δώσουμε ίσως μια καλύτερη απάντηση στις απορίες μας, και άρα να συμπληρώσουμε τις μελέτες γύρω από ένα αντικείμενο. Για τους λόγους αυτούς, σε αυτήν την ενότητα κρίνεται σκόπιμο, προτού προβούμε στην ανάλυση του κυρίως θέματος, να επιμείνουμε λίγο παραπάνω σε κάποιος από τους προαναφερθέντες τομείς της επιστήμης, έτσι ώστε να γίνουν κατανοητά από όλους βασικές έννοιες.

1.2 Ρομποτική Τεχνολογία

Ο κλάδος της Ρομποτικής είναι ένας πολλά υποσχόμενα κλάδος της Επιστήμης, ο οποίος εστιάζει κυρίως στην μελέτη, την μοντελοποίηση, τον προγραμματισμό και την ανάπτυξη των μηχανικών εγκαταστάσεων όπως είναι και τα ρομπότ. Συνδυάζει την επιστήμη της πληροφορικής, των Μαθηματικών, της Μηχανολογίας καθώς και της Ηλεκτρολογίας, με σκοπό την δημιουργία και βελτίωση ρομποτικών συστημάτων που θα μπορούν να εργάζονται ως υποκατάστατα του ανθρώπου όποτε και όπου αυτό κρίνεται απαραίτητο. Ακόμη, θα μπορούν να εναρμονίζουν την φυσική δραστηριότητα με την διαδικασία λήψης αποφάσεων, ενώ ταυτόχρονα θα επιλύουν σύνθετα προβλήματα με τον καλύτερο δυνατό τρόπο (όπως για παράδειγμα η αύξηση της παραγωγικότητας σε μια εργασία και βελτίωση της ποιότητας των παραγόμενων προϊόντων).

1.3 Μηχατρονική

Με τον όρο Μηχατρονική ή Ηλεκτρομηχανολογικοί Αυτοματισμοί, αναφερόμαστε στον κλάδο της Επιστήμης, ο οποίος θεωρείται αποτέλεσμα της συνεργασίας ευρύτερων Επιστημονικών κλάδων, όπως είναι αυτός της Ηλεκτρολογίας, της Ηλεκτρονικής, της Μηχανολογίας και της Πληροφορικής.

Στόχος της Επιστήμης αυτής, είναι η διερεύνηση, διαμόρφωση, αναπαραγωγή και εφαρμογή ευφυών μηχανικών συστημάτων, τα οποία θα συντελέσουν για τον άνθρωπο βοηθητικές εργαλειομηχανές βάση των οποίων θα μπορεί να επιλύσει πολλά προβλήματα τα οποία συναντά σε διάφορους τομείς της καθημερινότητας του. Για να υποστηρίξει, όμως, ένα μηχανικό σύστημα, και να προσφέρει την βοήθεια του στον άνθρωπο, με σκοπό την ολοκλήρωση των διεργασιών που τον απασχολούν, θα πρέπει να είναι εφοδιασμένο με συγκεκριμένα εξαρτήματα. Τέτοια εξαρτήματα, λοιπόν, συγκροτούν οι αισθητήρες, οι ενεργοποιητές και οι επεξεργαστές, τα οποία, προσδίδουν στην συνολική κατασκευή κίνηση, ψηφιακή επεξεργασία σημάτων, και έλεγχο κλειστών βρόχων. Επιπρόσθετα εκτός αυτού, παρέχουν και στον χειριστή την δυνατότητα πραγματοποίησης ενεργού, ήμι-ενεργού και παθητικού ελέγχου, καθώς και την μελέτη-ανάλυση του μηχανήματος ως προς την δυναμική του συμπεριφορά.

Αυτά τα χαρακτηριστικά τους, επομένως, τα καθιστούν κατάλληλα, για να χρησιμοποιηθούν και να εφαρμοστούν, σε άλλους πολυάριθμους τομείς της Επιστήμης. Οι κυριότερες των οποίων είναι η Ρομποτική, οι Αυτοματισμοί, Σερβο-υδραυλική μηχανική και η Επιστήμη των Ηλεκτρονικών Υπολογιστών. Ειδικότερα, θεωρούνται αρκετά χρήσιμα στην βιομηχανία (ιδιαίτως στην αυτοκινητοβιομηχανία), στα συστήματα μεταφορών ή αυτοματοποιημένα συστήματα, στους προσομοιωτές ρομπότ και στις μηχανές CNC. Για αυτόν τον λόγο, η περαιτέρω διερεύνηση και ενασχόληση, ενός ατόμου, με τον κλάδο αυτό, θεωρείται ότι θα συμβάλλει εξίσου εποικοδομητικά στη δημιουργία μιας καλύτερης κοινωνίας.

1.4 Τεχνητή Νοημοσύνη

Η Τεχνητή Νοημοσύνη ή AI (Artificial Intelligence), είναι τομέας της Επιστήμης και συγκεκριμένα της Πληροφορικής, ο οποίος πρωτοχρησιμοποιήθηκε το 1956 και επικεντρώνεται στην σύνθεση και την δημιουργία “έξυπνων” και αξιόπιστων υπολογιστικών συστημάτων, τα οποία δεν στηρίζονται και δεν απαιτούν στο μεγαλύτερο μέρος της λειτουργίας τους, την ανθρώπινη παρέμβαση. Τα συστήματα αυτά, εμπλέκουν μεθόδους, επεξεργασίας φυσικής γλώσσας, αναπαράστασης γνώσης, αυτοματοποιημένης συλλογιστικής, μηχανικής όρασης, ρομποτικής και μηχανικής μάθησης (machine learning). Χαρακτηρίζονται από πολύπλοκους μαθηματικούς αλγόριθμους, οι οποίοι με την σειρά τους, προσδίδουν στα συστήματα προσαρμοστικότητα στο περιβάλλον και την ικανότητα να προσεγγίζουν με σαφήνεια αυτό που τους ζητείται.

Ως αποτέλεσμα αυτών ήταν αρχικά να εμφανίζονται νέες ειδικότητες, οπότε συνεπαγόμενα αυτό οδήγησε στην εξάλειψη της ανεργίας, ενώ συγχρόνως όλο και περισσότεροι ενδιαφερόμενοι προσελκύνονταν και στρέφονταν ως προς αυτές. Έτσι, βήμα πρὸς βήμα ο τομέας αυτός αναπτύχθηκε περαιτέρω και πλέον σε αυτόν εντάσσονται άλλες επιμέρους κατηγορίες στις οποίες αυτός βρίσκει εφαρμογή. Μερικές από αυτές είναι και τα εμπειρικά συστήματα, η επίλυση των προβλημάτων, ο σχεδιασμός των νευρωνικών δικτύων, τα αυτοματοποιημένα οχήματα, τα συστήματα διαχείρισης βάσεων δεδομένων, τα ηλεκτρονικά παιχνίδια, τα ρομπότ, ακόμη και οι προσομοιωτές.

1.5 Βιομηχανική Ρομποτική

Η Βιομηχανική Ρομποτική συντελεί έναν πιο εξειδικευμένο κλάδο της Ρομποτικής Επιστήμης, ο οποίος φροντίζει να αφοσιώνεται, κατά κύριο λόγο, στην μοντελοποίηση και εφαρμογή των ρομπότ σε βιομηχανικό περιβάλλον.

Το βιομηχανικό ρομπότ έλαβε την ονομασία του αυτή απευθυνόμενο σε μια κατασκευή η οποία είναι προσαρμοσμένη να δέχεται, να αλληλοεπιδρά με ακρίβεια και ταυτόχρονα να επαναλαμβάνει τις εντολές που της καταχωρήθηκαν, για να πραγματοποιεί διεργασίες σε αντίξοες συνθήκες, όπου η ανθρώπινη παρέμβαση δεν συνιστάται ή και απαγορεύεται. Ωστόσο, σύμφωνα με τον Διεθνή Οργανισμό Τυποποίησης - ISO (International Organization for Standardization):

«Ένα Βιομηχανικό ρομπότ είναι ένα αυτόματος ελεγχόμενο από σερβο-μηχανισμό, ελεύθερος προγραμματιζόμενος πολυλειτουργικός χειριστής, με αρκετά μέρη, για τον χειρισμό κομματιών, εργασίας, εργαλείων ή ειδικών συσκευών. Προγράμματα μεταβλητής λειτουργίας καθιστούν εφικτή την εκτέλεση πολλαπλών εργασιών»

Στην αγορά πρωτοεμφανίστηκε κυρίως το 1971 από μια Ιαπωνική εταιρία, η οποία αρχικά μελετούσε την δραστηριοποίηση των ρομπότ μόνο σε βιομηχανικούς χώρους, όμως με το καιρό συνέθεσε έναν οργανισμό, αποκαλούμενο Ιαπωνική Ένωση Ρομπότ ή JARA (Japan Robot Association), αποτελούμενο από πολλές επιμέρους εταιρίες, και επεκτάθηκε έτσι και σε άλλες εφαρμογές των ρομπότ.

Σχετικά με την χρησιμότητα τους, παρατηρείται ένα ευρύ φάσμα εργασιών που αυτές οι μηχανές μπορούν να διατρέξουν στην καθημερινότητα. Προσκομίζοντας, πιθανότατα, στον άνθρωπο την επίλυση προβλημάτων, που εμπλέκουν ακόμα και τον χώρο εργασίας του. Ειδικότερα, μια αυτοκινητοβιομηχανία ή μια μεταλλουργική βιομηχανία, όπου επικεντρώνεται στο κέρδος, την άφθονη και ποιοτική παραγωγή, συνεπάγεται κατά βάση, την επιλογή συστημάτων, όπως είναι τα βιομηχανικά ρομπότ. Τα οποία ρομπότ μπορούν να εργάζονται ασταμάτητα και με ταχύτατους ρυθμούς, χωρίς όμως αυτό να επηρεάζει την απόδοσή τους, μπορούν να ανταπεξέλθουν στις ανθυγιεινές συνθήκες του περιβάλλοντα χώρου τους, στις υψηλές θερμοκρασίες που αναπτύσσονται και επίσης μπορούν με σχετική ευκολία να ανασηκώνουν και να μεταφέρουν ογκώδη ή μεγάλης μάζας αντικείμενα, όποτε αυτό χρειάζεται. Συνεπώς, ή παρουσία του όχι μόνο αντικαθιστά τον άνθρωπο όπου εγκρίνεται αναγκαίο αλλά συμβάλλει τρομερά και στην εξοικονόμηση χρόνου και ενέργειας.

Για να μπορεί, όμως, ένα τέτοιο σύστημα να καταταχθεί στην κατηγορία των βιομηχανικών ρομπότ και να κάνει όλες τις παραπάνω εργασίες, θα πρέπει:

- Να διαθέτει ένα σύστημα μετάδοσης της κίνησης, αισθητήρες και ένα σύστημα ελέγχου, με βάση τα οποία θα παρέχεται από τον χειριστή η πληροφορία ώστε να είναι σε θέση να την διατρέξει.
- Να μπορεί να αναπρογραμματιστεί σε διάφορες διεργασίες, που ο χρήστης της επιβάλλει.
- Να επιτυγχάνει το έργο της με μεγαλύτερη ταχύτητα και ακρίβεια από τον άνθρωπο.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Να αποδίδει την δυνατότητα ψηφιακής επικοινωνίας και με άλλα συστήματα που συμμετέχουν στην παραγωγική διαδικασία.
- Και να μπορεί να χρησιμοποιηθεί, σε δύσκολες για τον άνθρωπο καταστάσεις.

ΕΝΟΤΗΤΑ 2: ΤΑ ΒΙΟΜΗΧΑΝΙΚΑ ΡΟΜΠΟΤ

2.1 Εισαγωγή

Σε αυτήν ενότητα, θα εμβαθύνουμε περισσότερο σε έναν από τους τομείς της Επιστήμης, και αυτός είναι της Βιομηχανικής Ρομποτικής. Αρχικά λοιπόν, θα περιγράψουμε και θα αναλύσουμε την έννοια, την κατασκευή και την λειτουργία του ρομπότ ως αποτελούμενο μηχανικό σύστημα. Και έπειτα θα επεκταθούμε στα βιομηχανικά ρομπότ. Δηλαδή, στον τρόπο με τον οποίο κατασκευάζονται, τις κατηγορίες στις οποίες κατατάσσονται αναλόγως τις αρθρώσεις και τον έλεγχο μετάδοσης της κίνησής τους, ακόμη και τον προγραμματισμό τους.

2.2 Τι είναι το ρομπότ

Ο όρος αυτός αναφέρετε σε μια μηχανική συσκευή η οποία μπορεί και επενεργεί είτε αυτοματοποιημένα ή με την παρέμβαση του ανθρώπινου δυναμικού, με στόχο την παροχή υπηρεσιών και παραγωγή έργων, που από μόνος του ο άνθρωπος θα περιοριζόταν ή και θα αδυνατούσε να επιτελέσει. Στην ουσία, η συσκευή αυτή είναι διαμορφωμένη έτσι ώστε, να δέχεται σε κάθε περίπτωση κατευθυντήριες εντολές από τον άνθρωπο. Επιπλέον, ως μια “έξυπνη” μηχανή, όπως και χαρακτηρίζετε, διαθέτει έναν σκελετό, αποτελούμενο από εξαρτήματα που του υποβοηθούν στην κίνηση, αισθητήρες και ένα σύστημα ελέγχου, χάρις το οποίο είναι σε θέση να συλλαμβάνει την πληροφορία που της καταχωρείτε, να την επεξεργάζεται και στην συνέχεια να αντιδρά κατάλληλα με βάση αυτήν καθώς και τον χώρο στο οποίο βρίσκεται. Αξιοσημείωτη θεωρείται και η ικανότητα επαναπρογραμματισμού της μνήμης τους, γεγονός που τα διαφοροποιεί από όλα τα άλλα αυτοματοποιημένα συστήματα και τα καθιστά μοναδικά.

Το εύχρηστο αυτό εργαλείο χρησιμοποιείται τόσο στην βιομηχανία, την εκπαίδευση, τα διαστημικά προγράμματα, την διασκέδαση, την ιατρική όσο και σε απλές εφαρμογές της καθημερινής ζωής. Ένα τέτοιο παράδειγμα της καθημερινότητας είναι η διάσωση και η παροχή υπηρεσιών σε καταστάσεις εκτάκτου ανάγκης, όπως είναι μια πυρκαγιά.

Εντωμεταξύ, όλο και περισσότεροι επιστήμονες, τελευταία, επιλέγουν να ειδικευτούν στο τομέα της ρομποτικής και να αποκτήσουν σφαιρικές γνώσεις σχετικά με το αντικείμενο,

και αυτό είτε για να σχηματίσουν τα δικά τους υπαρκτά μοντέλα είτε για να εφαρμόσουν τις γνώσεις στους στα είδη υπάρχοντα. Ως αποτέλεσμα αυτού, τα ρομπότ να εξελίσσονται ραγδαία κάνοντας την εφαρμογή τους πιο ευκολοχέριστη και αποτελεσματική.

2.3 Δομικά χαρακτηριστικά μιας ρομποτικής κατασκευής

Ένα ρομπότ όπως και περιγράψαμε προηγουμένως είναι μια ευφυής, επαναπρογραμματιζόμενη μηχανή, η οποία εργάζεται αυτόματα. Βασιζόμενη, επομένως στην τεχνητή νοημοσύνη, κατέχει την ικανότητα πολλές φορές να συμπεριφέρεται όπως και ο άνθρωπος. Ειδικά, όμως τα βιομηχανικά ρομπότ, εμφανίζουν πολλά τέτοια ανθρωπομορφικά χαρακτηριστικά. Το συνηθέστερο, βεβαίως από αυτά είναι ο μηχανικός βραχίονας, ο οποίος στην πραγματικότητα προσομοιάζει τις λειτουργίες ενός ανθρώπινου βραχίονα. Το ερώτημα, ωστόσο που τίθεται εδώ, είναι με ποιο τρόπο κατασκευάζονται αυτά τα σπουδαία μηχανήματα, καθώς πολλά από αυτά διαφέρουν τόσο ως προς την εμφάνιση, όσο και στην διαχείριση. Βέβαια η διαφορά τους αυτή, οφείλεται κυρίως στα πρωτόκολλα κλειστής αρχιτεκτονικής ελέγχου που χρησιμοποιούν οι κατασκευάστριες εταιρίες.

Στην ουσία, όμως όλα τα βιομηχανικά ρομπότ, διαθέτουν την ίδια βασική δομή, όσον αφορά τον τρόπο συναρμολόγησης τους. Δηλαδή, αποτελούνται από μια βάση, πάνω στην οποία αναπτύσσεται η υπόλοιπη κατασκευή και η οποία παρέχει στηρικτικό ρόλο κυρίως. Ενώ, το άλλο σώμα φέρει ένα σύνολο από συνδέσμους, οι οποίοι ενώνονται μεταξύ τους με την βοήθεια των αρθρώσεων, και οι οποίοι σε συνδυασμό με την βάση και το σημείο δράσης, μας παρέχουν το σύστημα ελέγχου ή βραχίονα ή χειριστή (manipulator), όπως και αποκαλείται. Ο χειριστής είναι, δηλαδή, με άλλα λόγια ο βασικός σκελετός ενός βιομηχανικού ρομπότ, συνεπώς είναι και αυτός που καθορίζει τις ενέργειες ενός τέτοιου συστήματος.

Από την άλλη το σημείο δράσης ή ενεργό άκρο, τοποθετείται συνήθως στο ελεύθερο άκρο του τελευταίου συνδέσμου που απαρτίζει την κατασκευή. Επίσης, όπως αναφέραμε αποτελεί μέρος του χειριστή και χρησιμεύει ως εργαλείο για να μπορεί το ρομπότ να

επιτυγχάνει το σκοπό του. Έτσι, αυτό το εργαλείο μπορεί να αντιστοιχεί, για παράδειγμα, σε μια αρπάγη ή μια συσκευή ηλεκτροσυγκόλλησης, η οποία μπορεί να επενεργεί βασιζόμενη στις εντολές που ο χρήστης της παραχωρεί, με στόχο την μεταφορά ενός αντικειμένου ή ηλεκτροσυγκόλληση, όπως και για την επίτευξη άλλων διεργασιών. Και, μάλιστα, λόγω του μεγάλου εύρους εφαρμογής της, ποικίλει και αυτή στην μορφή της, όπως και οι αρθρώσεις.

Από τα παραπάνω, λοιπόν, λαμβάνουμε μια γενική εικόνα για τον σχεδιασμό και τον τρόπο διαμόρφωσης του κυρίου σώματος της ρομποτικής κατασκευής, ωστόσο ένα αυτοματοποιημένο σύστημα εκτός από το σύστημα μετάδοσης της κίνησης, για να μπορέσει να λειτουργήσει, χρειάζεται και ένα σύστημα ελέγχου. Συνεπώς, για να καλύψουμε όλες τις απαιτήσεις μοντελοποίησης ενός τέτοιου συστήματος, θα πρέπει να αναφερθούμε και στο σύστημα ελέγχου της ρομποτικής κατασκευής, το οποίο είναι αρμόδιο για τον προγραμματισμό και την διαχείριση του ρομπότ.

Έτσι, το σύστημα αυτό θα συμπεριλαμβάνει αρχικά έναν ηλεκτρονικό υπολογιστή διαμέσου του οποίου θα επιτυγχάνεται και ο έλεγχος, ένα σύστημα επικοινωνίας, χάρις το οποίο ο ρομποτικός βραχίονας μπορεί να δέχεται την πληροφορία από τον χρήστη, κάποιους ενεργοποιητές(actuators), οι οποίοι ενεργούν με στόχο να προσδώσουν κίνηση στο ρομπότ και τέλος μερικούς αισθητήρες βάση τους οποίους θα αντιλαμβάνεται το περιβάλλον του.

Κάποια από τα εξαρτήματα, όμως, που συμμετέχουν ενεργά στην κατασκευή και συναρμολόγηση ενός ρομποτικού συστήματος, όπως και υπόθηκε προηγουμένως, ομαδοποιούνται και σε άλλες επιμέρους κατηγορίες αναλόγως με τον ρόλο και την ιδιότητα που αυτές παρέχουν στην συγκεκριμένη συσκευή.

2.3.1 Οι αρθρώσεις

Οι αρθρώσεις, επειδή αποτελούν τον συνδετικό κρίκο μεταξύ δύο συνδέσμων, είναι και τα στοιχεία που καθορίζουν την ευελιξία των κινήσεων ενός χειριστή, άρα και αυτά που συντονίζουν τις ενέργειες της ρομποτικής κατασκευής.

Μπορούμε να πούμε, βεβαίως, ότι λειτουργούν και ως άξονες της κίνησης. Για είναι όμως εύκολος ο υπολογισμός τους, έχει οριστεί ένα επιπλέον στοιχείο που ονομάζεται βαθμός ελευθερίας. Μάλιστα, αυτός χρησιμοποιείται για να μας περιγράψει το σύνολο και τον τύπο των κινήσεων που μας παρέχει μια άρθρωση ή και το σημείο δράσης. Συχνά, όμως ένας βαθμός ελευθερίας αντιστοιχεί σε ένα τύπο άρθρωσης. Παρόλα αυτά, υπάρχει και ο βαθμός κινητικότητας, ο οποίος πολλές φορές συμπίπτει με τον βαθμό ελευθερίας, αλλά στην πραγματικότητα φέρει διαφορετική ερμηνεία. Σε σχέση λοιπόν με τον βαθμό ελευθερίας, ο βαθμός κινητικότητας, εξαρτάται μόνο από το σύνολο των αρθρώσεων που απαρτίζουν την συγκεκριμένη κατασκευή, οπότε εάν έχουμε 5 αρθρώσεις σε έναν ρομποτικό βραχίονα, αυτό θα μεταφράζεται αυτόματα και ως 5 βαθμοί κινητικότητας.

Όσον αφορά τώρα πάλι την άρθρωση, αυτή διαχωρίζεται σε κάποιους τύπους αναλόγως και την κίνηση που επιφέρει στην συνολική κατασκευή. Για να προσδιοριστεί με ακρίβεια, επομένως, ο τύπος κίνησης και ο κατασκευαστής να είναι σε θέση να αναγνωρίζει εύκολα και γρήγορα ποια επιθυμεί να χρησιμοποιήσει, αυτή έχει ταξινομηθεί σε κάποιες βασικές κατηγορίες, οι οποίες καταγράφονται παρακάτω:

- **Πρισματική ή Τηλεσκοπική άρθρωση:** επιτρέπει την γραμμική μετατόπιση σε ένα και μόνο από τους τρεις άξονες (X, Y, Z) του χώρου, συνεπώς χαρακτηρίζεται από έναν βαθμό ελευθερίας.
- **Περιστροφική άρθρωση:** επιτρέπει την περιστροφική κίνηση σε ένα και μόνο από τους τρεις άξονες (X, Y, Z) του χώρου, οπότε χαρακτηρίζεται και αυτή από έναν βαθμό ελευθερίας.
- **Άρθρωση κύλισης:** αυτή επιτρέπει δύο είδη κινήσεων, μια περιστροφική και μια μεταφορική, ωστόσο η περιστροφική κίνηση δρά σε άξονα κάθετο με την μεταφορική. Χαρακτηρίζεται από δύο βαθμούς ελευθερίας.

- **Κυλινδρική άρθρωση:** επιτρέπει την περιστροφική και μεταφορική κίνηση, όπως και στην άρθρωση κύλισης, με την διαφορά εδώ, ότι η περιστροφική δρά στον ίδιο άξονα με την μεταφορική κίνηση, και χαρακτηρίζεται από δύο βαθμούς ελευθερίας.
- **Ελεύθερη άρθρωση:** αυτή πάλι, χαρακτηρίζεται από δύο βαθμούς ελευθερίας, διότι, επιτρέπει μονό την περιστροφική κίνηση σε δύο από τους άξονες (X, Y, Z) του χώρου.
- **Σφαιρική άρθρωση:** η οποία επιτρέπει την περιστροφική κίνηση και στους τρεις άξονες (X, Y, Z) του χώρου, οπότε και χαρακτηρίζεται από τρεις βαθμούς ελευθερίας.

Βέβαια, από τις έξι διαφορετικές αυτές μορφές αρθρώσεις, ο κατασκευαστής ενός ρομποτικού βραχίονα, χρησιμοποιεί πολλές φορές αυτές που έχουν ένα βαθμό ελευθερίας ή εκείνες που προκύπτουν από συνδυασμούς αυτών.

2.3.2 Ο χώρος εργασίας του βραχίονα και του σημείου δράσης

Ένας κατασκευαστής, ο οποίος έχει επίγνωση του χώρου εργασίας ενός ρομποτικού συστήματος, γνωρίζει και τις δυνατότητες που έχει αυτή η μηχανή στο συγκεκριμένο χώρο. Αναλυτικότερα, είναι σε θέση να οριοθετήσει τον χώρο στον οποίο μπορεί να αποδώσει το ρομπότ, να επιλέξει την συσκευή για τον κατάλληλο τύπο εργασίας και να προσδώσει με μεγαλύτερη άνεση και ακρίβεια τις οδηγίες για τον έλεγχο του ρομποτικού συστήματος.

Στην προκειμένη περίπτωση του βραχίονα-χειριστή, ο χώρος εργασίας συμβαδίζει με το σύστημα συντεταγμένων αυτού, το οποίο με την σειρά του είναι απόλυτα εξαρτώμενο από σύστημα των αρθρώσεων που φέρει η προς μελέτη ρομποτική κατασκευή. Έτσι ο χώρος εργασίας – σύστημα συντεταγμένων του βραχίονα ομαδοποιείται με τον εξής τρόπο:

- **Σφαιρικό ή Πολικό σύστημα:** Ο περιβάλλοντας χώρος που καλύπτει αυτό το σύστημα, όπως αναφέρει και το όνομά της, είναι μια σφαίρα, η οποία αποτελείται από δύο στροφικούς και ένα γραμμικό άξονα.
- **Κυλινδρικό σύστημα:** Ο χώρος εργασίας αυτού συστήματος είναι ένας κύλινδρος αποτελούμενος από έναν στροφικό και έναν γραμμικό άξονα.

- **Ορθογώνιο ή Καρτεσιανό σύστημα:** Σε αυτή την περίπτωση ο χώρος εργασίας είναι το γνωστό σε όλους μας σύστημα συντεταγμένων (X,Y,Z) , αποτελούμενο από τρεις γραμμικούς άξονες.
- **Αρθρωτό σύστημα:** Εδώ, ο περιβάλλοντας χώρος του βραχίονα, συνιστάται από τρεις στροφικούς άξονες.
- **SCARA (Selective Compliance Assembly Robot Arm):** Το συγκεκριμένο σύστημα χαρακτηρίζεται κατάλληλο για τα ρομπότ ανάκτησης και τοποθέτησης, καθώς ο τύπος βραχίονα που φέρει ενώ παρομοιάζει με εκείνον του αρθρωτού, διαφέρει ως προς αυτόν στις αρθρώσεις του ώμου και αγκώνα, όπου στην εξεταζόμενη περίπτωση είναι κάθετοι.

Εκτός, όμως από το σύστημα συντεταγμένων του βραχίονα, ένα σύνθετο βιομηχανικό ρομπότ εμφανίζει και ένα σύστημα συντεταγμένων που περιγράφει τις κινήσεις του σημείου δράσης. Αυτό, λοιπόν το σύστημα συνθέτει και το χώρο εργασίας ή με άλλα λόγια τον χώρο πάνω στον οποίο μπορεί να δρά το σημείο δράσης. Επειδή, ωστόσο η κίνηση που μπορεί να επιτύχει, το εργαλείο αυτό ως προς τους τρεις άξονες (X,Y,Z) , είναι η περιστροφική, κατηγοριοποιείται και ανάλογα σε τρεις βασικές δραστηριότητες:

- **Roll:** Ο ρόλος αυτής της δραστηριότητας είναι να περιστρέφει το εργαλείο εγκάρσια, δηλαδή γύρω από τον άξονα X.
- **Pitch:** Ο ρόλος αυτής της δραστηριότητας είναι να περιστρέφει το εργαλείο διαμήκους του βραχίονα, δηλαδή γύρω από τον άξονα Y.
- **Yaw:** Ο ρόλος αυτής της δραστηριότητας είναι να περιστρέφει το εργαλείο γύρω από τον άξονα Z.

Συχνά, όταν αναφερόμαστε στις δραστηριότητες roll, pitch και yaw, αναφερόμαστε ταυτόχρονα και στις γωνίες Euler ή γωνίες RPY.

Ευκολά, λοιπόν, μπορούμε να συμπεράνουμε ότι το σημείο δράσης, μπορεί να διαθέτει και αυτό εώς τρεις βαθμούς ελευθερίας, άρα συμβάλλει εξίσου σημαντικά στην ελευθερία των κινήσεων και την ποικιλία των καθηκόντων που μπορεί να επιτελέσει η συνιστώσα συσκευή.

2.3.3 Οι ενεργοποιητές

Σε ένα ρομποτικό βραχίονα, παρόλο που η άρθρωση θεωρείται υπεύθυνη για το είδος και την διεύθυνση της κίνησης, από μόνη της δεν μπορεί να δράσει. Χρειάζεται, επομένως έναν μηχανισμό ικανό να την θέσει σε λειτουργία. Έτσι σε αυτόν τον ρόλο ήρθε και έδωσε ο ενεργοποιητής. Ειδικότερα, ο ενεργοποιητής αποτελεί τον μηχανισμό εκκίνησης των αρθρώσεων. Δρα εκλαμβάνοντας ως είσοδο ένα ηλεκτρικό σήμα από τον υπολογιστή και με βάση αυτό αποδίδει ως έξοδο την απαραίτητη ροπή για την ενεργοποίηση των αρθρώσεων. Επειδή, όμως οι διεργασίες διαφέρουν μεταξύ τους, θα πρέπει αναλόγως να διαφοροποιούνται και οι μηχανισμοί που ορίζονται κατάλληλοι για την πραγματοποίηση αυτών. Με την ίδια λογική, λοιπόν και οι ενεργοποιητές διαχωρίζονται σε τρεις διαφορετικούς μηχανισμούς κίνησης:

- a) Τους ηλεκτρικούς μηχανισμούς
- b) Τους πνευματικούς μηχανισμούς
- c) Και τους υδραυλικούς μηχανισμούς.

Ο συνηθέστερος από αυτούς, βέβαια είναι ο ηλεκτρικός μηχανισμός διότι εξασφαλίζει μεγαλύτερη ακρίβεια, συγκριτικά με τους άλλους δύο, ενώ παράλληλα βρίσκει εφαρμογή και στις πιο πολύπλοκες διεργασίες. Τώρα, όσον αφορά τους άλλους δύο, περιορισμένη χρήση εμφανίζει ο πνευματικός μηχανισμός, καθώς προορίζεται, κυρίως για τα μικρότερα ρομπότ, ενώ αντίθετα ο υδραυλικός λόγω των μεγάλων ροπών που αγγίζει εν ώρα λειτουργίας και της μεγάλης του ταχύτητας, αυξάνει παρόμοια σχεδόν με τον ηλεκτρικό τα επίπεδα εφαρμογής του.

Εκτός, όμως αυτών οι ενεργοποιητές, καθορίζουν την ταχύτητα ανταπόκρισης και την σταθερότητα του ρομπότ. Άρα είναι σε θέση να ελέγχουν τις ταλαντώσεις που δημιουργούνται από την συσκευή προκειμένου να επιδιώξουν την επιθυμητή ισορροπία και να αυξο-μειώνουν την ταχύτητα με την οποία το όργανο δράσης μπορεί να επενεργεί από ένα σημείο σε ένα άλλο.

2.3.4 Ενεργό άκρο

Το ενεργό άκρο, σχετίζεται με την εργασία που πρόκειται να διεξάγει ο ρομποτικός βραχίονας. Στην πλειοψηφία των περιπτώσεων, εφοδιάζει, το ρομπότ, με κατάλληλα

όργανα, τα οποία μπορούν να ανταπεξέλθουν σε διεργασίες που απαιτούν την συγκράτηση, μετακίνηση-μεταφορά, κατασκευή, σύνθεση, συγκόλληση ή και βαφή αντικειμένων. Τέτοιου είδους όργανα αποτελούν συνήθως οι αρπάγες, οι οποίες διακρίνονται, όπως παρουσιάζετε και μετέπειτα, σε επιμέρους κατηγορίες, αναλόγως τον τύπο εφαρμογής τους:

- Απλές μηχανικές αρπάγες: Στην κατηγορία αυτή εμπεριέχονται απλά εργαλεία προσχεδιασμένα ώστε να αντιμετωπίσουν με σχετική ευκολία βασικές λειτουργίες. Χαρακτηριστικό παράδειγμα αυτών είναι ο γάντζος.
- Μηχανική αρπάγη: Σε αυτή την κατηγορία, εντάσσονται μηχανικά όργανα που φέρουν δυο ή και περισσότερους δακτυλίους, χάρις τους οποίους το ρομπότ μπορεί να τραβάει, να απομονώνει, να συγκρατεί, να ανασηκώνει, να μετατοπίζει και να τοποθετεί αντικείμενα.
- Αρπάγη κενού: Η λειτουργία αυτής περιορίζεται, καθώς ο μηχανισμός τους φέρει βεντούζες, οι οποίες προορίζονται μόνο για κατασκευές που διαθέτουν επίπεδη επιφάνεια.
- Αρπάγη με συγκολλητική ουσία: Η αρπάγη αυτή χρησιμοποιεί μια συγκολλητική ουσία προκειμένου να μπορεί να συγκρατεί με άνεση άλλα ευκολοδιαχειρίσιμα υλικά.
- Μαγνητική αρπάγη: Τέλος, σε αυτήν την κατηγορία, υπάγονται οι αρπάγες που φέρουν μαγνητικές επιφάνειες, πάνω στις οποίες μπορούν να προσκολληθούν άλλες μεταλλικές επιφάνειες.

2.4 Μέθοδοι ελέγχου κίνησης

Γενικότερα, ένα ρομπότ προγραμματίζεται συνήθως, έτσι ώστε να ολοκληρώνει με ακρίβεια μια εργασία που ο διαχειριστής του αναθέτει. Ωστόσο, πρώτου προχωρήσει ο χρήστης στον προγραμματισμό αυτού, θα πρέπει να αντιληφθεί ποια τροχιά θέλει να ακολουθήσει το ρομπότ στην ακόλουθη εφαρμογή και εν συνέχεια να επιλέξει με ποιόν τρόπο επιθυμεί η μηχανή αυτή να πραγματοποιήσει την συγκεκριμένη τροχιά. Εδώ, λοιπόν, πρέπει να σημειωθεί, ότι ένας ρομποτικός βραχίονας διαθέτει δύο τρόπους κίνησης στον χώρο και αυτοί είναι η:

- a) Σημειακή κίνηση

b) Κίνηση συνεχόμενης διαδρομής

Ουσιαστικά, στην σημειακή μετατόπιση, ο χρήστης καθορίζει ένα ή και περισσότερα σημεία και έπειτα αφού δοθεί εντολή στο ρομπότ να δράσει, αυτό θα μεταβιβαστεί στο κάθε ένα από αυτά, ακολουθώντας όμως την σειρά που ο χρήστης του επέβαλλε. Επομένως, αυτό που κάνει το ρομπότ στο στάδιο αυτό είναι να μετακινείται από σημείο σε σημείο, χωρίς να κατέχει κάποια πληροφορία για τα ενδιάμεσα σημεία και χωρίς βέβαια να τον ενδιαφέρουν αυτά. Όσον αφορά την διαδρομή του, αυτή πολλές φορές μπορεί να μην συμβιβάζεται με τις δικές μας απαιτήσεις. Καθώς, στην περίπτωση που στο ρομπότ δεν έχει παραχωρηθεί κάποια οδηγία για τα υπόλοιπα σημεία του χώρου και άρα δεν έχει εκφραστεί κάποιος περιορισμός για αυτά, αυτό έχει την δυνατότητα να επενεργεί ‘‘υπό δική του βούληση’’, δηλαδή διανύει ελεύθερη τροχιά. Συχνά, όταν η διαδρομή αυτή αφορά δύο σημεία το ρομπότ είναι προγραμματιζόμενο να ακολουθήσει την βέλτιστη διαδρομή, η οποία δεν είναι κάτι άλλο παρά μια ευθεία διερχόμενη από τα δύο αυτά σημεία. Αλλά τις περισσότερες φορές, την στιγμή που το ρομπότ κατευθύνεται προς ένα ζητούμενο σημείο, δεν το προσπερνάει απλά, αντιθέτως σταματάει σε αυτό είτε για να διεξάγει κάποια ενέργεια, είτε ακόμη και για να προσεγγίσει το σημείο, οπότε και να επαληθεύσει τον κώδικα που του καταχωρήθηκε εξ αρχής. Έτσι, αυτό έχει ως αποτέλεσμα η τροχιά που θα σχηματίσει να είναι δυσνόητη και απροσδιόριστη.

Βέβαια, το πλεονέκτημα του σημείο σε σημείο είναι ότι το μηχανικό σύστημα του ρομπότ, ως ένα σύστημα που έχει διαμορφωθεί ακολουθώντας τα πρότυπα των νεών τεχνολογιών και στηριζόμενο στην Τεχνητή Νοημοσύνη, διαθέτει όλα τα προσόντα αυτών, που το προσδίδουν την ικανότητα για παράδειγμα, να αποθηκεύει αρχικά στην μνήμη του τα σημεία και σε επόμενη φάση, εφόσον χρειαστεί να επιστρέψει πάλι σε αυτά, δεν θα αναζητήσει από το χρήστη να του παραχωρήσει πάλι όλες τις συντεταγμένες των σημείων, άλλα θα πάει να ανατρέξει στην μνήμη του.

Τώρα για την δεύτερη κατηγορία, που αφορά την κίνηση συνεχόμενης διαδρομής, η μετατόπιση του ρομπότ επιδιώκεται συνεχόμενα, χωρίς δηλαδή να εμφανίζονται μικρά διαλείμματα ή παύσεις του μηχανήματος στα ενδιάμεσα σημεία αυτού. Ειδικότερα, σε αυτή την μορφή εργασίας, ο ρομποτικός βραχίονας προσεγγίζει μια σειρά από σημεία τα

οποία εμφανίζουν “στενή επαφή” μεταξύ τους, δηλαδή βρίσκονται το ένα πολύ κοντά με το άλλο. Σχηματίζοντας, με τον τρόπο αυτό, ίσως μια ακανόνιστη τροχιά, η οποία μπορεί και αυτή όπως στην μετάβαση από σημείο σε σημείο να απομνημονευτεί στο ρομπότ, με την διαφορά βέβαια ότι εδώ η αποθήκευση περιλαμβάνει όλη την διαδρομή που διένυσε το ρομπότ και όχι απλώς μεμονομένα σημεία αυτής. Όμως στην εξεταζόμενη περίπτωση, το θετικό είναι ότι μπορεί να προσδιοριστεί με σχετική ευκολία η διαδρομή που θα επιτελέσει το ρομπότ. Και το συγκεκριμένο επιτυγχάνεται εν μέρη βάσει της βοήθειας των ελεγκτών, οι οποίοι για να υπολογίσουν ομαλές τροχιές, χρησιμοποιούν τους διάσημους μεθόδους παρεμβολής τους, δεδομένου βέβαια ότι το ρομπότ έχει επίγνωση για το πρώτο και τελευταίο σημείο αυτών. Ενώ, στις πιο απρόβλεπτες κινήσεις, η τροχιά προσεγγίζεται σε περίπτωση που επιβάλλουμε στο ρομπότ να διαπεράσει μια σειρά από γειτονικά σημεία.

2.5 Προγραμματισμός ρομπότ

Οι οποιεσδήποτε κινήσεις ενός του ρομπότ, συνήθως ελέγχονται μέσω του ψηφιακού συστήματος ελέγχου που διαθέτει το ίδιο το ρομπότ. Βέβαια, εμφανίζονται και εξαιρέσεις, στις οποίες οι κινήσεις μιας τέτοιας διάταξης, πραγματοποιούνται με την βοήθεια αυτοματοποιημένου συστήματος, που φέρει δηλαδή διακόπτες και εμπόδια, ικανά να υποβοηθήσουν το ρομπότ να επιτελέσει την επιθυμητή λειτουργία.

Παρόλα αυτά και οι δύο διαδικασίες αποσκοπούν τον προγραμματισμό του ρομπότ, δηλαδή στην κυριότερη εκ των όλων διαδικασία, δυνατή να αντιστρέψει την κατάσταση ακινησίας ενός ρομποτικού συστήματος. Για παράδειγμα στον βιομηχανικό κλάδο, σχεδόν κάθε εργασία που επιτυγχάνεται πολλές φορές, ακολουθεί κάποιους αλληπάλληλους κύκλους, συνεπώς και ένα ρομπότ για να μην διακόπτει την λειτουργία του ή αδρανοποιείται κάθε φορά που πραγματοποιεί μια επιμέρους ενέργεια, έχει προγραμματιστεί να λειτουργεί επαναλαμβανόμενα μέχρι να ολοκληρώσει όλους τους κύκλους εργασίας του. Η τακτική αυτή δεν βοηθάει απλώς τον διαχειριστή του, αλλά επηρεάζει συνειδητά και την παραγωγική διαδικασία. Σαφέστερα, αυτό που προκαλεί στην ρομποτική συσκευή είναι να δουλεύει ασταμάτητα, μέχρι ο χρήστης αποφασίσει να της

επιβάλλει κάτι διαφορετικό. Αποτέλεσμα αυτής, να επιταχύνεται η παραγωγική διαδικασία και να εξοικονομούνται χρήματα και χρόνος.

Ωστόσο, όπως τέθηκε εν συντομία προηγουμένως, μια ρομποτική συσκευή συνήθως προγραμματίζεται διαμέσου του ψηφιακού συστήματος που φέρει η ίδια η διάταξη. Αναλυτικότερα, η παραπάνω ενέργεια τίθεται σε ισχύ, με δύο τρόπους:

▪ **Προγραμματισμός διαμέσου υπολογιστή:**

Αφού αρχικά ο διαχειριστής διαμορφώσει τον απαραίτητο κώδικα για την κίνηση που επιθυμεί να κάνει η ρομποτική συσκευή, στην συνέχεια εφόσον επιλέξει το κατάλληλο ρομπότ και τον τρόπο με τον οποίο θέλει να εισχωρήσει στην κατασκευή το κώδικα, (δηλαδή είτε απευθείας διαμέσου του ψηφιακού συστήματος που διαθέτει, είτε εν μέσω του ηλεκτρονικού υπολογιστή), καταλήγει στην εφαρμογή του για να επιβεβαιώσει την ορθότητα του. Βέβαια, η κωδικοποίηση των κινήσεων διαμέσου ενός υπολογιστικού προγράμματος, καταστεί από μόνη της μια διαδικασία περίπλοκη και προϋποθέτει ο χρήστης να γνωρίζει μία ή και περισσότερες γλώσσες προγραμματισμού, έτσι ώστε να μπορεί να “επικοινωνήσει” και να διατυπώσει με ευχέρεια τις εντολές που πρόκειται, μετέπειτα το ρομπότ να διαβάσει.

▪ **Προγραμματισμός διαμέσου καθοδήγησης:**

Ο ρομποτικός βραχίονας, στην περίπτωση αυτή, μετατοπίζεται είτε χειροκίνητα από τον διαχειριστή, ο οποίος βήμα προς βήμα καθοδηγεί και συντελεί μαζί με το ρομπότ τις κινήσεις, είτε με την βοήθεια ενός τηλεχειριστήριου που ελέγχει και κατευθύνει συγχρόνως το ρομπότ από σημείο σε σημείο. Παράλληλα, με την διαδικασία αυτή, επιδιώκεται ο υπολογισμός της επιθυμητής ταχύτητας του ρομπότ από χρονομετρικές συσκευές, έτσι ώστε να επιτευχθεί η ζητούμενη τροχιά. Και ταυτοχρόνως, το ρομπότ επενεργεί στην καταγραφή και απομνημόνευση, τόσο των σημείων όσο και της διαδρομής που επιτέλεσε, με αποτέλεσμα να ανατρέξει σε αυτές, όταν ο χρήστης του το επιβάλλει και εν τέλει να τις εφαρμόσει με την ίδια ακριβώς μέθοδο και λογική.

ΕΝΟΤΗΤΑ 3: ΡΟΜΠΟΤΙΚΟΙ ΠΡΟΣΟΜΟΙΩΤΕΣ

3.1 Εισαγωγή

Οι ρομποτικοί προσομοιωτές συνιστούν το κυρίως θέμα της συγκεκριμένης εργασίας συνεπώς καλύπτουν και το μεγαλύτερο κομμάτι αυτής. Οπότε, σε αυτήν την ενότητα, εκ πρώτης όψεως θα προσπαθήσουμε, να καλύψουμε βασικές έννοιες, δηλαδή, να περιγράψουμε τι είναι ένας ρομποτικός προσομοιωτής, να ταξινομήσουμε τα πλεονεκτήματα και μειονεκτήματά του και τέλος να καταγράψουμε τα όργανα που απαιτούνται για την σωστή λειτουργία του.

3.2 Ηλεκτρονικός υπολογιστής

Επειδή, ο Ηλεκτρονικός Υπολογιστής συνθέτει ένα από τα βασικότερα εργαλεία για την μετέπειτα ανάλυση και ανάπτυξη του θέματος που περιλαμβάνει και η παρούσα εργασία, ενδείκνυται αναγκαίο να εστιάσουμε, αρχικά σε αυτό. Ειδικότερα, όπως επισημάνθηκε νωρίτερα, αποτελεί και αυτός προϊόν εφαρμογής της Τεχνολογίας. Και συγκροτεί την συνηθέστερη διάταξη με βάση την οποία, το άτομο:

- Διαθέτει άμεση πρόσβαση σε πληθώρας πληροφορίες, τις οποίες μπορεί να εκλαμβάνει και να χρησιμοποιεί όποτε και όπου χρειαστεί.
- Καταχωρεί και να επεξεργάζεται δεδομένα.
- Προσφέρει υπηρεσίες.
- Ανταλλάσσει απόψεις και επικοινωνεί με άλλα άτομα.
- Δημιουργεί, σχεδιάζει, οργανώνει και μοντελοποιεί πολλές εφαρμογές, αλλά και διάφορα συστήματα.
- Αναπαράγει εικόνες, βίντεο και ήχους.
- Και επενεργεί στον έλεγχο άλλων μηχανημάτων και συστημάτων.

Τα υπολογιστικά συστήμα, τα συστήματα διαχείρισης βάσεων δεδομένων, η διαχείριση μιας ιστοσελίδας, ο έλεγχος μιας ηλεκτρομηχανικής κατασκευής όπως είναι και τα ρομπότ, ακόμη και οι ρομποτικοί προσομοιωτές είναι αποτέλεσμα χρήσης της

ηλεκτρονικής αυτής συσκευής. Και, πολλές εργασίες εξαρτώνται από αυτή. Άρα, αυτό συνεπάγεται όσο καλύτερα σχεδιασμένη είναι αυτή η διαταξή τόσο ακριβέστερα και ταχύτατα θα δέχεται τις εντολές του χρήστη, θα τις αξιοποιεί και θα επιστρέφει τα αποτελέσματα αυτών.

Οπότε η χρήση του, όπως αντιλαμβανόμαστε, κρίνεται ιδιαίτερα σημαντική και αισθητή στην εργασίες που επιτελεί το κάθε άτομο ξεχωριστά, τόσο σε επαγγελματικό επίπεδο όσο και σε προσωπικό επίπεδο. Αναλυτικότερα, ένας Ηλεκτρονικός Υπολογιστής, συνθέτει μια υπολογιστική διάταξη, αποτελούμενη από κάποια εξαρτήματα, τα οποία χωρίζονται σε δύο βασικές κατηγορίες και οι οποίες καταγράφονται παρακάτω:

I. Υλισμικό Μέρος (hardware):

- Η κεντρική μονάδα επεξεργασίας ή CPU (Central Processing Unit).
- Η μνήμη RAM ή ROM-BIOS.
- Οι εσωτερικές ή εξωτερικές μονάδες ανάγνωσης και αποθήκευσης δεδομένων όπως, DVD, SSD (Solid State Drive) και σκληρός δίσκος.
- Οι μονάδες εισόδου-εξόδου (πληκτρολόγιο, ποντίκι, οθόνη, κάμερα).
- Οι ενσωματωμένες συσκευές ,όπως για παράδειγμα, ο εκτυπωτής και το μόντεμ.

II. Λογισμικό Μέρος (software):

- Το λειτουργικό σύστημα, το οποίο αποτελεί την βάση για να μπορέσει το άτομο να επικοινωνήσει με κάποιο τρόπο με το σύστημα αυτο.
- Και τα πακέτα λογισμικών, τα οποία μπορεί να ενσωματώνουν προγράμματα, γλώσσες προγραμματισμού, καθώς και άλλα παρόμοια βοηθητικά εργαλεία.

3.3 Τι είναι ένας ρομποτικός προσομοιωτής

Οι προσομοιωτές ή simulators, όπως και αλλιώς αποκαλούνται, στην σημερινή εποχή αποτελούν αναπόσπαστα εργαλεία των ερευνητών για την αποτελεσματικότερη και αποδοτικότερη μελέτη των ρομπότ. Συγκεκριμένα, ο ρομποτικός προσομοιωτής είναι ένα ρομπότ εικονικής πραγματικότητας, μια υπολογιστική εφαρμογή δηλαδή, η οποία

κατασκευάστηκε προκειμένου να μιμηθεί όσο των δυνατών καλύτερα το περιβάλλον, την συμπεριφορά, και γενικά τα λειτουργικά χαρακτηριστικά που απαρτίζουν ένα πραγματικό μοντέλο. Η μέθοδος αυτή προσομοίωσης μιας φυσικής κατασκευής, δίνει την δυνατότητα στους ερευνητές να εξετάσουν τις ιδέες-θεωρίες τους για την αποτελεσματικότητα τους και έπειτα να τις θέσουν σε εφαρμογή, αποφεύγοντας και περιορίζοντας με αυτόν τον τρόπο τα σφάλματα που ενδεχομένως να είχαν προκύψει εάν δεν είχε γίνει πρωτίστως αυτός ο έλεγχος. Η χρήση ενός πραγματικού ρομπότ για τον έλεγχο του κώδικα ενός αλγορίθμου μπορεί να καταστεί μια κουραστική και χρονοβόρα διαδικασία, τι γίνεται όμως σε περίπτωση που επιθυμούμε να μελετήσουμε παραπάνω από έναν αλγόριθμο, τότε αυτή η διαδικασία ίσως χρειαστεί πολύ περισσότερο χρόνο από όσον έχουμε στην διάθεσή μας. Σε αντίθεση ένας προσομοιωτής έχει την ικανότητα να διεξάγει πολλαπλούς ελέγχους σε ελάχιστο χρόνο, και συνεισφέρει σημαντικά στην εξοικονόμηση χώρου και χρηματικών πόρων, δυο παράμετροι που διευκολύνουν σε μεγάλο βαθμό το έργο ενός ερευνητή. Καθώς, ένα πραγματικό ρομπότ για να μπορέσει να τεθεί σε λειτουργία πρέπει να εκπληρεί κάποιες προϋποθέσεις, όπως είναι η τροφοδοσία του από κάποια πηγή ενέργειας, η εξασφάλιση των απαραίτητων εξαρτημάτων, η σωστή συναρμολόγηση τους, και ένα κατάλληλο περιβάλλον στο οποίο θα γίνεται η τοποθέτηση ή ο έλεγχος του. Όμως, όλα αυτά με την σειρά τους κοστίζουν και αυτό το θέμα δεν μπορεί να αντιμετωπιστεί από όλους. Ενώ, σε έναν προσομοιωτή δεν τίθεται τέτοιο ζήτημα, τα έξοδα είναι πολύ λιγότερα, επειδή η προσομοίωση επιτυγχάνεται διαμέσου ενός υπολογιστή, ο οποίος χρειάζεται μόνο ενέργεια για την λειτουργία του. Όσον αφορά τον χρήστη, είναι σε θέση να τροποποιήσει το περιβάλλον προσομοίωσης και να το διαμορφώσει με βάση τις δικές του προτιμήσεις και απαιτήσεις. Τέλος, οι μικροδιορθώσεις που μπορεί να εξασφαλίσει μέσω του εργαλείου αυτού όχι μόνο τον οδηγούν στην επιτυχή ολοκλήρωση της μελέτης του, αλλά συμβάλλουν και στην διατήρηση της ασφάλεια του.

Σχετικά με την πρόοδο των ρομποτικών προσομοιωτών, όπως και αναφέρθηκε εξ αρχής, παρατηρούνται τεράστια άλματα σε πολλούς τομείς της σημερινής κοινωνίας, έτσι η ορολογία αυτή χρησιμοποιείται για να περιγράψει μια μεγάλη γκάμα εφαρμογών αυτών, όπως είναι ένας ρομποτικός βραχίονας ή και ένα αυτοκινούμενο όχημα.

Πολλοί από αυτούς, μάλιστα εμφανίζονται στο εμπόριο με ενσωματωμένα υπολογιστικά μηχανήματα φυσικής, τα οποία μπορούν και αναπαριστούν φυσικά φαινόμενα όπως και το

φαινόμενο της βαρύτητας. Επίπλέον, μπορεί να παρουσιάζουν τρισδιάστατα γεωμετρικά περιβάλλοντα, που επιτρέπουν στον χρήστη την καλύτερη προσομοίωση του ενδιαφερόμενου αντικειμένου, όπως και τεράστιες βιβλιοθήκες, που βοηθούν και αυτές με την σειρά τους στο να επιταχυνθεί η διαδικασία της προσομοίωσης και μπορεί να εμπεριέχονται και αυτές εξαρχής στον προσομοιωτή ή να είναι αποτέλεσμα συνεισφοράς δεδομένων μεταξύ άλλων προγραμμάτων. Κάποια, συνεργάζονται και με άλλα λειτουργικά συστήματα τα οποία τους παρέχουν επιπρόσθετες αρμοδιότητες, ενώ και στον προγραμματισμό τους συμμετέχουν αρκετά ήδη γνωστές, για τον χειριστή, γλώσσες προγραμματισμού.

Από όλα τα παραπάνω λοιπόν, μπορεί να συμπεράνει κανείς πόσο σπουδαία είναι η χρήση αυτού εργαλείου για την ιδεατή κατανόηση και την υλοποίηση εφαρμογών σε ένα ρομποτικό μηχάνημα.

3.4 Τι χρειάζεται ένας προσομοιωτής ρομπότ για να λειτουργήσει

Για να μπορέσουμε να πραγματοποιήσουμε πειράματα ή και απλά να θέσουμε σε λειτουργία έναν ρομποτικό προσομοιωτή απαιτούνται τα εξής:

- Αρχικά ένας ηλεκτρονικός υπολογιστής
- Μια πηγή ενέργειας, η οποία θα τροφοδοτεί τον υπολογιστή, όποτε χρειαστεί
- Και την εγκατάσταση του λογισμικού που θα κάνει την προσομοίωση στην εκάστοτε υπολογιστική μονάδα.

Επίσης, κρίνεται σκόπιμο αλλά όχι αναγκαίο, σε περίπτωση που γίνεται μια μελέτη ή κάποιο πείραμα που αφορά ένα ρομπότ το οποίο ενδεχομένως να βρίσκεται ήδη κατασκευασμένο να χρησιμοποιείται και αυτό στην διαδικασία της προσομοίωσης, με σκοπό τα αποτελέσματα να είναι ικανοποιητικά.

3.5 Πλεονεκτήματα- μειονεκτήματα ρομποτικών προσομοιωτών

Ένας ρομποτικός προσομοιωτής όπως υπόθηκε προηγουμένως καλύπτει τόσο θέματα ασφάλειας, όπως και πολλαπλών ελέγχων, εξοικονόμησης χρόνου, κόστους και χώρου. Εκτός, όμως από αυτά πλεονεκτεί και σε πολλά άλλα στάδια, όπως το περιβάλλον του, το οποίο δεν μπορεί να καταστραφεί εάν κάποιο λάθος ξεφύγει από τον χρήστη. Ενώ, από την άλλη μεριά μια φυσική κατασκευή μπορεί να προκαλέσει σοβαρές ζημιές στον

περιβάλλοντα χώρο του και ταυτόχρονα να θέσει σε κίνδυνο και τον ίδιο τον χειριστή. Η χρήση τους είναι επίσης εύκολη και σε πολλές περιπτώσεις ο προσομοιωτής διατίθεται ακόμη και στην δωρεάν του έκδοση, συνεπώς μπορεί να χρησιμοποιηθεί από μεγάλο αριθμό του πληθυσμού καλύπτοντας εξίσου μεγάλο εύρος ηλικιών. Με αυτόν τον τρόπο, δίνεται η δυνατότητα στους χρήστες, να εξασκηθούν με τα ρομπότ τους και να αποκτήσουν περισσότερη εμπειρία για το πώς αυτά αντιλαμβάνονται και υλοποιούν τις εντολές τους, να εφαρμόσουν τις σκέψεις τους αρχικά σε ένα τεχνητό κόσμο για να φτιάξουν ένα επίσης τεχνητό ρομπότ, που όμως θα κατέχει ίδια μορφή και θα επιτυγχάνει τις ίδιες λειτουργίες με ένα φυσικό. Και έπειτα να προχωρήσουν στην ανάπτυξη του πραγματικού τους μοντέλου. Ακόμη, μπορούν να φέρουν εις πέρας πολύπλοκες φυσικές καταστάσεις, όπως και να εκτελέσουν παράλληλες προσομοιώσεις, εκχωρώντας ταυτόχρονα πολλαπλούς παραμέτρους για το ίδιο πείραμα μέσω διαφορετικών υπολογιστικών μονάδων, διαδικασία η οποία αδυνατεί να συμβεί σε ένα αληθινό ρομπότ. Διότι, για να μπορέσει να την επιτελέσει, θα πρέπει τουλάχιστον να συνοδεύεται από ένα μεγάλο αριθμό παρόμοιων ή και ίδιων ρομποτικών κατασκευών, αλλά δυστυχώς το κόστος μια και μόνο ρομποτικής κατασκευής μπορεί να είναι ήδη υψηλό, ποσό μάλλον πολλών τέτοιων κατασκευών.

Επιπρόσθετα, η κοινή χρήση του ευνοεί στην εκπαίδευση, και κυρίως στον τομέα τις ανώτατης εκπαίδευσης, όπου διευκολύνεται τόσο το εκπαιδευτικό προσωπικό στην διδασκαλία του, αλλά και ο ίδιος ο φοιτητής στις σπουδές του. Ειδικότερα, ο φοιτητής μπορεί να δουλέψει είτε αυτόνομα είτε συλλογικά με άλλα άτομα, ενθαρρύνοντας τον έτσι να επικοινωνεί, να ανταλλάσσει απόψεις, να συνεργάζεται με την ομάδα του, να διενεργεί στην λήψη των αποφάσεων, να σχηματίζει τα δικά του πρότυπα ρομπότ, να φτιάχνει τις δικές του καταστάσεις και συνθήκες, να διευρύνει το γνωστικό του υπόβαθρο, και να ανταποκρίνεται με επιτυχία στις υποχρεώσεις του ως φοιτητής. Έπειτα, ο σπουδαστής μετά την ολοκλήρωση του πρώτου κύκλου των σπουδών του και ως απόφοιτος πλέον, μπορεί εφόσον το επιθυμεί και έχοντας αποκτήσει μια εικόνα για τα ρομπότ και τις ιδιότητές τους, να συνεχίσει τις σπουδές του, να εξειδικευτεί περαιτέρω ή και να επεκταθεί μέσω της έρευνας από τις είδη υπάρχουσες γνώσεις σε αυτόν τον χώρο, οπότε και να τις εμπλουτίσει αυτές ακόμη πιο πολύ. Ένα επίσης σημαντικό πλεονέκτημα του προσομοιωτή είναι το ότι μπορεί να συνεργαστεί με ένα πραγματικό ρομπότ και να μεταφέρει αυτούσια την πληροφορία ή και με ελάχιστες τροποποιήσεις, και αυτό το καθιστά ιδιαίτερα

ικανοποιητικό στην εφαρμογή του, όπως διαπιστώνεται και από πολλούς διερευνητές, με βάση τα αποτελέσματα που προκύπτουν από την διεξαγωγή πολλαπλών πειραμάτων.

Ωστόσο, όμως σε μια έρευνα εάν τα συμπεράσματα που προκύπτουν από μια διεργασία προσομοίωσης μέσω ενός προσομοιωτή ρομπότ βασίζονται σε αυτήν και μόνο, τότε η έρευνα ίσως να μην είναι και τόσο αποδοτική. Και αυτό διότι ο προσομοιωτής παρουσιάζει όχι μόνο πλεονεκτήματα αλλά και μειονεκτήματα. Αυτά, συνδέονται κυρίως με τον τρόπο διαχείρισης του λογισμικού, δηλαδή εάν ο χρήστης δεν έχει αντιληφθεί καλά αυτό το οποίο πάει να προσομοιώσει ή έχει κάνει κάποιο λάθος στο σχεδιασμό της προσομοίωσης τότε και η προσομοίωση θα είναι λανθασμένη, αφού θα αφορά ένα εντελώς διαφορετικό πρόβλημα. Επίσης, ένα πραγματικό ρομπότ πιθανότατα να μην διενεργήσει κατάλληλα ή και στην χειρότερη περίπτωση να μην μπορέσει λειτουργήσει καθόλου, όταν του δοθεί ο αλγόριθμος που έχει τρέξει και έχει δουλέψει σωστά σε έναν προσομοιωτή υπό τις ίδιες συνθήκες. Το πρόβλημα αυτό κυρίως παρουσιάζεται επειδή ένα τεχνητό ρομπότ είναι φτιαγμένο να δουλεύει ιδανικά, όποτε στην προσομοίωση δυο όμοια εξαρτήματα θα είναι ακριβώς ίδια χωρίς καμία διαφορά το ένα από το άλλο. Ακόμη, και το ίδιο το περιβάλλον δουλεύει σε ιδανικές συνθήκες, στην πράξη όμως αυτό δεν συμβαίνει, θα υπάρχει έστω και ελάχιστη διαφορά στα εξαρτήματα, όπως και στις καταστάσεις που επικρατούν στον περίγυρο του ρομπότ, λόγω της πολυπλοκότητας τους, δεν θα μπορούν να αποσαφηνιστούν όταν επιτελείται η λειτουργία του.

Εξίσου σημαντικός, όμως, θεωρείται και ο θόρυβος που συναντάτε σε έναν πραγματικό αισθητήρα καθώς στην πραγματικότητα μπορεί να υποδηλώνει κάποιο χαρακτηριστικό σφάλμα του ρομπότ ή και την ορθή του λειτουργία, αναλόγως βέβαια και τον αισθητήρα, αλλά σε έναν προσομοιωτή αυτό δεν εμφανίζεται. Με αποτέλεσμα να υπάρχει πιθανότητα το σφάλμα να διατυπωθεί από το πρόγραμμα με λάθος τρόπο, ετσι ώστε να μην γίνει δηλαδή κατανοητή η εμφάνιση του από τον χρήστη ή και στην χειρότερη το πρόγραμμα να μην την αντιληφθεί, όποτε και να μην παρουσιάσει καθόλου.

Καταλήγοντας, θα πρέπει να επισημάνουμε και μια επιπλέον διαφορά, η οποία συντελεί καθοριστικό ρόλο στην ανάπτυξη, όπως και καλύτερη αναπαράσταση μιας μηχανικής διάταξης. Και αυτή η διαφορά εμπλέκει την υπολογιστική ισχύ των μηχανημάτων που συμμετέχουν στην προσομοίωση και στην πραγματικότητα. Μια υπολογιστική μονάδα που αποτελείται από έναν ‘‘ισχυρό’’ επεξεργαστή, επιτυγχάνει την διαδικασία της

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

προσομοίωσης και επεξεργασίας εντολών με τάχιστατους ρυθμούς μέσω ενός προσομοιωτή, σε σχέση με ένα πραγματικό ρομπότ, το οποίο ίσως περιλαμβάνει επεξεργαστή χαμηλότερης ισχύος και δεν κατέχει αυτήν την ικανότητα.

ΕΝΟΤΗΤΑ 4: ΠΡΟΣΟΜΟΙΩΤΕΣ ΠΡΟΣ ΜΕΛΕΤΗ

4.1 Εισαγωγή

Οι προσομοιωτές ρομπότ, με την μελέτη και την ανάλυση των οποίων θα ασχοληθούμε στην παρούσα εργασία, διατίθενται στην αγορά με την εξής ονοματολογία:

- ✓ Gazebo simulator
- ✓ RoboDK simulator
- ✓ Webots simulator
- ✓ OpenRAVE simulator
- ✓ V-REP simulator.

4.2 Προσομοιωτής Gazebo



Εικόνα 1: Gazebo Logo

Ο Gazebo είναι ένα από τα πιο επίκαιρα και ευρέως διαδεδομένα μοντέλα, ικανό να επιτελέσει με επιτυχία προσομοιώσεις υψηλού επιπέδου σε τρισδιάστατα περιβάλλοντα. Εντάσσεται στα ελεύθερα λογισμικά – ανοιχτού κώδικα (αδειοδοτημένο από το Apache 2.0), και ετούτο διότι ο πηγαίος κώδικας που φέρει το πρόγραμμα διανέμεται δωρεάν παρέχοντας έτσι την δυνατότητα στους ενδιαφερόμενους χρήστες να επιτύχουν με ευκολία οποιαδήποτε τροποποίηση. Σχετικά με το λογισμικό, αυτό θεωρείτε κατασκευαστικά

έτοιμο να διεξάγει με μεγάλη ακρίβεια και πιστότητα προσομοιώσεις πληθωρικών ρομποτικών συστημάτων και αισθητήρων, ενώ ταυτόχρονα προσδίδει την δυνατότητα αναπαράστασης αυτών ακόμη και σε δυσκολοπρόσιτα περιβάλλοντα τόσο εσωτερικών όσο και σε εξωτερικών χώρων. Στην πραγματικότητα ως λογισμικό ομαδοποιείτε σε δύο επιμέρους κατηγορίες υπεύθυνες για την λειτουργία του. Η μία από αυτές αφορά το server και επιτελείται μέσω της εντολής gzserver, ενώ η άλλη απευθύνεται στο χρήστη-client, οπότε και επιτελείται μέσω της εντολής gzclient.

Την εμφάνιση του εκ πρώτης όψεως την έκανε το 2002 στο Πανεπιστήμιο της Νότιας Καλιφόρνιας υπό μορφή μιας έρευνας, ωστόσο στην ανάπτυξη και μοντελοποίηση του αρχικού προτύπου συμμετείχαν οι Andrew Howard και Nate Koenig. Αυτοί, στόχευαν εξαρχής στην ίδρυση ενός μοντέλου που θα εκπροσωπεί με επάρκεια τις λειτουργίες ενός αληθινού ρομπότ και συγχρόνως θα επιλύει τις ανάγκες για προσομοίωση σε εξωτερικά περιβάλλοντα. Παρόλα αυτά το 2004 έως το 2011 οι ιδιότητες αυτού του προϊόντος αξιοποιήθηκαν πρωτίστως από ένα άλλο δικτυακό server, το Player, στο οποίο ο Gazebo αποτελούσε μια υποβοηθητική εργαλειομηχανή. Ενώ, παράλληλα το 2009 ο John Hsu κατάφερε να ενσωματώσει στο πρόγραμμα και άλλα, αξιολογήματα λογισμικά όπως το ROS (Robot Operating System) και το PR2 (Personal Robot 2), προσδίδοντας την δυνατότητα στο Gazebo να συνεργαστεί με αυτά, βελτιώνοντας έτσι τις ιδιότητες του και επιτυγχάνοντας ταυτόχρονα βαθμιαία εξέλιξη. Απο εκεί και έπειτα στην περαιτέρω ανέλιξη του, διευθύνοντα ρόλο διαδραμάτισε ο Willow Garage, ο οποίος υποστήριζοντας την μέχρι τότε δουλειά των πρωτοπόρων, ανέλαβε να συνδράμει οικονομικά στο έργο, ενδυναμώνοντας κατά αυτό τον τρόπο τις προσπάθειες για μελλοντικές αναβαθμίσεις του προγράμματος. Αλλά εν τέλει μετά απο κάποιο χρονικό διάστημα, το πρόγραμμα κατάφερε να διαχωριστεί από τα υπόλοιπα λογισμικά και κατέκτησε την αυτονομία του. Μάλιστα, από το 2014 έως και σήμερα το OSRF(Open Source Robotics Foundation), συνιστάμενη από μια πολλά υποσχόμενη ομάδα ερευνητών, η οποία αποσπάστηκε από την ROS το 2012 προκειμένου να συνεχίσει την ενασχόληση της και επιδιώκοντας να προσφέρει απλόχερα την υποστήριξη στο πρόγραμμα, κατάφερε να ακμάσει τα λειτουργικά χαρακτηριστικά του προγράμματος και επιπρόσθετα ενίσχυσε τις πιθανότητες για περαιτέρω ανάπτυξη.

4.2.1 Χαρακτηριστικά του προσομοιωτή:

Η προσομοίωση ενός πραγματικού ρομποτικού συστήματος αποδίδεται με τον καλύτερο δυνατό τρόπο χάρις το Gazebo, και αυτό διότι το πρόγραμμα είναι εμπλουτισμένο με εργαλεία, εντολές, πακέτα, και γενικότερα φέρει χαρακτηριστικά ικανά να καλύψουν ένα ευρύ φάσμα λειτουργιών και υπηρεσιών. Μερικά από τα χαρακτηριστικά αυτά είναι:

✓ Οι επιλεγμένες μηχανές φυσικές :

- Η ODE (Open Dynamics Engine)
- Η Bullet
- Η Simbody
- Και η DART (Dynamic Animation and Robotics Toolkit)

Η προσβασιμότητα σε αυτές τις μηχανές, του παρέχει την δυνατότητα να προσομοιώνει με επάρκεια τον “δυναμικό κόσμο” και τους φυσικούς νόμους που διέπουν την κινηματική συμπεριφορά ενός πραγματικού ρομποτικού εξοπλισμού.

✓ Συμβατότητα με άλλα εξωτερικά πακέτα λογισμικών (ROS):

Το πρόγραμμα παρόλο που αποχωρίστηκε από άλλα λογισμικά, προκειμένου να χτίσει την αυτονομία του, δεν έπαυσε να συνεργάζεται με κάποια από αυτά, αναβαθμίζοντας με αυτόν τον τρόπο τις λειτουργίες του. Ένα τέτοιο λογισμικό ενδιάμεσου επιπέδου-middleware συνιστά και το ROS (Robotic Operation System).

✓ Η τρισδιάστατη γραφική απεικόνιση του περιβάλλοντα χώρου :

Αποτελεί μια διαδικασία η οποία επιτυγχάνεται και υποστηρίζεται, από ένα αξιόλογο πλαίσιο, όπως είναι το OGRE (Object Oriented Graphics Rendering Engine). Ως αποτέλεσμα της χρήσης του οποίου το πρόγραμμα είναι σε θέση να αναπαριστά χρώματα, υφές, σκιές και φωτισμούς υψηλής απόδοσης πλησιάζοντας έτσι όλο και πιο πολύ την πραγματικότητα.

✓ Χρήση δημοφιλών γλωσσών προγραμματισμού:

Οι περισσότερες από τις γλώσσες προγραμματισμού, τις οποίες μπορεί να χρησιμοποιήσει ο καθένας σε μια τυπική εφαρμογή του παρόντος προγράμματος είναι οι:

- C
- C++
- Python
- Και Java

✓ Ποικιλία αισθητήρων:

Αυτοί έχουν διαμορφωθεί με σκοπό να προσαρμόζονται αναλόγως με τις απαιτήσεις των χρηστών, παραχωρώντας τους την δυνατότητα να συμπεριλάβουν στην προσομοίωση αισθητήρες θορύβου, οδομετρίας, στυλ Kinetic, επαφής, ροπής δύναμης, επιταχυνσίμετρα, LIDAR, συστήματα έυρεση laser, 2D/3D κάμερες όπως και πολλά άλλα.

✓ Τα πρόσθετα (plugins):

Συνιστούν με την σειρά του μαζικές βιβλιοθήκες που μπορούν να ενσωματωθούν σε ένα σύστημα και με βάση τις οποίες οι χρήστες έχουν πλήρη επίγνωση και έλεγχο των λειτουργιών του προσομοιωτή. Γενικά τα plugins που φέρει το λογισμικό του Gazebo κατηγοριοποιούνται σε έξι τύπους, και εκτιμάται ότι αποδίδουν μεγαλύτερη λειτουργικότητα στα URDF συστήματα.

✓ Regression testing:

Το πρόγραμμα είναι ικανό να επιτελέσει με αξιοπιστία και αποδοτικότητα regression tests.

✓ Πλήθος προηγμένων ρομποτικών συστημάτων:

Ο χρήστης μέσω του προγράμματος αυτού, κατορθώνει την ακριβή προσομοίωση πολυάριθμων δημοφιλών ρομποτικών συστημάτων, ενώ παράλληλα μπορεί να δοκιμάσει και να μοντελοποιήσει μέχρι και δικά του μοντέλα, είτε μέσω προγραμματισμού είτε με την βοήθεια των SDF αρχείων (Simulation Description Format). Ο όρος μοντέλα στον προσομοιωτή, χρησιμοποιείται ουσιαστικά για την περιγραφή μιας φυσικής οντότητας, με δυναμικές, κινηματικές και οπτικές ιδιότητες. Οπότε, βάση αυτού ένα μοντέλο, μπορεί να αντιπροσωπεύει για παράδειγμα, ένα απλό γεωμετρικό σχήμα ή ακόμη και μια πολύπλοκη κατασκευή

ρομπότ. Όσο για την αποθήκευση και συντήρηση των οποιοδήποτε μοντέλων που δημιουργούνται από τον καθένα, αυτά στηρίζονται σε μια βάση δεδομένων:

<https://github.com>

Κάθε μοντέλο το οποίο αρχειοθετείται σε μια πρότυπη βάση δεδομένων, ακολουθεί μια δομή. Εμφανέστερα, μια τέτοια δομή μπορεί να αποτυπωθεί με τον ακόλουθο τρόπο:

Βάση δεδομένων

- a) database.config: Αυτό το αρχείο περιέχει πληροφορίες άδειας για τα μοντέλα, ένα όνομα για τη βάση δεδομένων και μια λίστα με όλα τα έγκυρα μοντέλα. Εντούτοις, αυτό πλέον έχει εκσυγχρωτιστεί και συμπληρώνεται αυτόματα από το CMakeLists.txt.
- b) model_1: Ένας κατάλογος για το μοντέλο_1
- c) model.config: Μετα-δεδομένα σχετικά με το μοντέλο_1
- d) model.sdf: Περιγραφή των SDF του μοντέλου
- e) model.sdf.erb: Περιγραφή μοντέλου SDF με ενσωματωμένο Ruby
- f) meshes- πλέγματα: Ένας κατάλογος για όλα τα αρχεία COLLADA και STL
- g) υλικά: Ένας κατάλογος που πρέπει να περιέχει μόνο πρωτότυπα για τους υποκαταλόγους σχετιζόμενους με τις υφές.
- h) υφές: Ένας κατάλογος για αρχεία εικόνας (jpg, png, κ.λπ.).
- i) scripts: Ένας κατάλογος για χρήση και ιδιότητες των σεναρίων OGRE
- j) plugins: Ένας κατάλογος για τις πληροφορίες των πρόσθετων.

Βέβαια, στην πλειοψηφία των περιπτώσεων το πρόγραμμα έχει επιλεγθεί να εφαρμόζεται για την προσομοίωση mobile robots και manipulators.

✓ Animation - ‘actors’:

ένα δυνατό πλεονέκτημα του συγκεκριμένου προσομοιωτή είναι και η λεγόμενη χρήση των ‘κινούμενων σεναρίων’. Τα οποία και ενδείκνυνται χρήσιμα, σε περιπτώσεις που επιθυμούμε να έχουμε στην προσομοίωση οντότητες που ακολουθούν καθορισμένες τροχιές, χωρίς όμως να μεταβάλλονται και να ενεργούν με βάση τις μηχανές φυσικής. Άρα, αυτομάτως αυτό υπονοεί ότι τα σώματα αυτά

δεν θα αντιδρούν για παράδειγμα στις δυνάμεις λόγω βαρύτητας και ούτε θα συγκρούονται με άλλα μοντέλα.

Σχετικά, βέβαια με την μορφή τους αυτή θα ακολουθεί τα πρότυπα ενός 3D σώματος με 3D πλέγματα (meshes), τα οποία θα ανιχνεύονται από RGB κάμερες και GPU αισθητήρες αντίστοιχα.

Χαρακτηριστικά παραδείγματα τέτοιων σωμάτων που πραγματοποιούν συγχρόνως κίνηση, αποτελούν και οι “actors”. Μια ορολογία που χρησιμοποιείται για να δείξει ότι τα μοντέλα σε αυτήν την περίπτωση προσομοιάζουν ανθρώπους.

✓ Ενσωματώνει την χρήση λογισμικών όπως το MeshLab και το Solidworks:

Ο προσομοιωτής εντυπωσιάζει για ακόμα μια φορά, καθώς επιτρέπει την χρήση λογισμικών όπως το MeshLab και το Solidworks, προκειμένου να υπολογίσει με ακρίβεια και να συλλέξει δεδομένα σχετικά με τους παραμέτρους αδράνειας που συμμετέχουν σε μια προσομοίωση. Τέτοια δεδομένα, ενδεχομένως να αφορούν την μάζα των μοντέλων, το κέντρο της θέσης μάζας, αλλά και την ροπή αδράνειας (moment of inertia) αυτών. Μάλιστα, ο προσδιορισμός της ροπής αδράνειας (ή γωνιακής μάζας ή και περιστροφικής αδράνειας όπως και αποκαλείται) ενός άκαμπτου σώματος, είναι ιδιαίτερα χρήσιμος, διότι παρέχει πληροφορίες για την χωρική κατανομή της μάζας αυτού, και επιπρόσθετα καθορίζουν την ροπή που απαιτείται για μια επιθυμητή γωνιακή επιτάχυνση γύρω από έναν άξονα περιστροφής. Για αυτό και εξαρτάται από την μάζα του σώματος, το μέγεθος, σχήμα και τον επιλεγμένο άξονα στον οποίο και επιτυγχάνεται η περιστροφή. Η μάζα από την άλλη μεριά υπολογίζεται μέσω του λογισμικού, πολλαπλασιάζοντας το γεωμετρικό όγκο που καταλαμβάνει το σώμα με την αντίστοιχη πυκνότητα αυτού

✓ Μεταφορά από TCP/IP:

Αυτή η ενέργεια επιτρέπει στο πρόγραμμα να διεξάγει οποιαδήποτε προσομοίωση σε μια απομακρυσμένη συσκευή που φέρει κατάλληλες διεπαφές, εν μέσω μιας υπηρεσίας μετάδοσης μηνυμάτων χρησιμοποιώντας το Google Protobufs.

✓ Οι ελεγκτές του μπορούν να μεταφερθούν απευθείας σε πραγματικά ρομποτικά συστήματα

✓ Προσθήκη δυνάμεων και ροπών:

Το Gazebo έχει αναβαθμίσει πολλές από τις λειτουργίες, ενώ συγχρόνως έχει προσθέσει αρκετές νέες ιδιότητες, έτσι ώστε να συμβαδίζει με τις απαιτήσεις μιας ρεαλιστικής κατασκευής. Μια τέτοια, επομένως ιδιότητα αυτού είναι η προσθήκη δυνάμεων και ροπών πάνω στο σώμα της, ως προς μελέτη, προσομοίωσης. Εκτενέστερα, η προσθήκη μιας δύναμης ή αντιστοίχως μιας ροπής, συνήθως εφαρμόζεται σε έναν από τους συνδέσμους της κατασκευής, ενώ η διεύθυνση αυτής μπορεί να ‘ακολουθεί’, (δηλαδή μπορεί να είναι παράλληλη ή να εφάπτεται), έναν από τους άξονες X,Y,Z του συστήματος. Επιπρόσθετα, τα πεδία που σχηματίζονται με τους άξονες διαδραματίζουν και αυτά σημαντικό ρόλο, διότι καθένα από αυτά καθορίζει και την ποσότητα της δύναμης που θα ασκηθεί σε αυτή την κατεύθυνση.

✓ Προσομοίωση εν μέσω cloud:

Εφαρμόζοντας το CloudSim ο χειριστής κατέχει την ικανότητα να επιτελέσει προσομοιώσεις στο Amazon AWS και στο GzWeb.

✓ Πρόσθετα στην γραμμή εντολών:

Τα πρόσθετα εργαλεία που προβάλλονται στην γραμμή εντολών παρέχουν την δυνατότητα στους χρήστες να διευθύνουν, να ελέγχουν και να καταγράφουν τις προσομοιώσεις.

✓ Επιτρέπει την αποθήκευση και λήψη στιγμιότυπων οθόνης

✓ Ταχύτατη δοκιμή αλγορίθμων:

Μέσω του προγράμματος ο χρήστης συνθέτει τους δικούς του αλγορίθμους και επιτυγχάνει γρήγορες δοκιμές.

✓ Εξοικείωση με τα συστήματα AI:

Η αξιοποίηση και υλοποίηση των AI επιφέρει την δημιουργία αποδοτικότερων ρομποτικών συστημάτων, με εκτεταμένες λειτουργίες.

✓ Εγκατάσταση σε λειτουργικά συστήματα Linux και Windows:

Αρχικά η εγκατάσταση του προγράμματος περιοριζόταν όσον αφορούσε τα λειτουργικά συστήματα. Συγκεκριμένα, όλες οι παλαιότερες εκδόσεις του ήταν διαθέσιμες μόνο σε ένα λειτουργικό σύστημα, το Ubuntu, το οποίο με την σειρά του βασιζόταν στο Linux, ενώ θεωρείται ελεύθερο ως προς την χρήση του. Όμως με τα χρόνια η ραγδαία ανάπτυξη της τεχνολογίας και συνεπακόλουθα οι αναβαθμίσεις που αυτό υπέστη, επέτρεψαν στους χρήστες να μπορούν να χρησιμοποιήσουν το

πρόγραμμα σε λειτουργικά συστήματα του Windows, όπως και σε άλλα παρεμφερή του Linux:

- Debian
- Fedora
- Arch
- Και Gentoo.

Βέβαια, ως πρόγραμμα εξακολουθεί να είναι ιδιαίτερα πιο αποδοτικό στο Ubuntu, ενώ

η εγκατάσταση σε αυτό προϋποθέτει την χρήση ενός ηλεκτρονικού συστήματος με μια καλή κάρτα γραφικών, έναν ισχυρό επεξεργαστή (I5 Intel), και τουλάχιστον 500MB ελεύθερου χώρου.

✓ Ελεγκτής Razer Hydra:

Το Gazebo υποστηρίζει τον ελεγκτή Razer Hydra, για να μπορεί ο χρήστης να ανιχνεύει με ακρίβεια την θέση και τον προσανατολισμό των σωμάτων που επιθυμεί.

✓ Υστερεί σε τεχνικές rendering

✓ Συμβατό με βιβλιοθήκες μηχανικής οράσεως:

Το πρόγραμμα είναι συμβατό με αρκετές βιβλιοθήκες που στηρίζονται στην μηχανική όραση και που σκοπό έχουν την παροχή πληροφοριών, δεδομένων και επεξεργασία εικόνας. Σε αυτές τις βιβλιοθήκες εμπεριέχονται:

- Το OpenCV
- Το OpenNI
- Και το PCL

✓ Ελεύθερη πρόσβαση:

Τέλος το λογισμικό είναι κοινόχρηστο και διανέμεται χωρίς κάποια χρέωση, συνεπώς μπορεί να χρησιμοποιηθεί από το καθένα.

4.2.2 Το Γραφικό Περιβάλλον:

Μόλις το πρόγραμμα εγκατασταθεί στον υπολογιστή, και εφόσον τεθεί υπό εκκίνηση, ως πρώτη όψη αυτό που ο χρήστης θα εκλάβει εν μέσω της οθόνης του, είναι το γραφικό

περιβάλλον – GUI(Grafical User Interface). Ετούτο, ουσιαστικά αποτελεί και το “ χώρο εργασίας ” του ρομπότ, οπότε για να είναι πιο εύκολη η επικοινωνία και συνεργασία μεταξύ χρήστη - ρομπότ, συνιστάται από επιμέρους πολυάριθμα εργαλεία-εντολές. Συνεπώς, το γραφικό περιβάλλον όπως παρουσιάζεται και από την εικόνα 1, συντίθεται από πέντε ξεχωριστές εργαλειοθήκες. Και αυτές είναι:

▪ Η Σκηνή:

Αποτελεί κατα βάση το περιβάλλον στο οποίο διαδραματίζεται η προσομοίωση του ρομποτικού εξοπλισμού.

▪ Το Αριστερό Πάνελ:

Εμπεριέχει τρεις επιμέρους εντολές:

- Το World: Με την επιλογή της εντολής αυτής εμφανίζεται στην οθόνη του υπολογιστή μια καρτέλα που αντιπροσωπεύει τα μοντέλα που φέρει την στιγμή εκείνη η Σκηνή. Και δεδομένου αυτής, ο χειριστής μπορεί να επέμβει στην τροποποίηση της στάσης του μοντέλου, την στάση της κάμερας αλλά και άλλων παραμέτρων που αφορούν και πάλι το μοντέλο.
- Το Insert: Με αυτήν την εντολή, ο χρήστης κατορθώνει να προσθέσει στην διαδικασία την προσομοίωσης νέα μοντέλα.
- Το Layers: Με την επιλογή της παρούσας εντολής ο χρήστης αναλαμβάνει να ρυθμίζει τα διάφορα επίπεδα οπτικοποίησης στα οποία ενσωματώνονται και τα μοντέλα. Έτσι μπορεί να προβεί είτε στην εμφάνιση για παράδειγμα ενός τέτοιου επιπέδου είτε στη απόκρυψη αυτού και συνεπαγόμενα του μοντέλου ή των μοντέλων που βρίσκονται σε αυτό.

▪ Το Δεξί Πάνελ:

Η συγκεκριμένη καρτέλα παρέχει κυρίως την δυνατότητα στο χρήστη να επεξεργάζεται και να αλληλεπιδρά με τις πληροφορίες που αφορούν τα κινητά μέρη του ρομπότ, όπως και είναι οι αρθρώσεις του. Επομένως, εάν στην Σκηνή δεν εντοπιστεί κάποιο ρομποτικό μοντέλο προς προσομοίωση, αυτό σηματοδοτεί αυτόματα την αδρανοποίηση και εκκαθάριση των πληροφοριών που φέρει αυτή η καρτέλα.

▪ Η Άνω Γραμμή Εντολών:

Αυτή η γραμμή εντολών φέρει μια σειρά από αλληπάλληλες εντολές και βρίσκεται τοποθετημένη στο πάνω μέρος της Σκηνής. Ειδικότερα οι εντολές αυτές είναι:

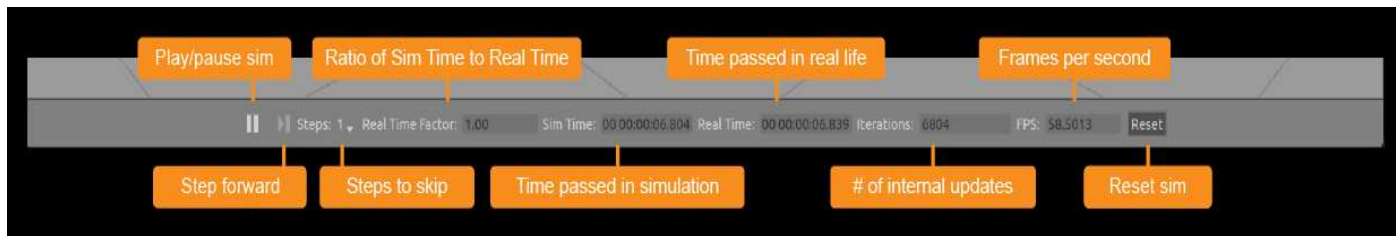
- **Select mode:** μια εντολή η οποία επιτρέπει τον προσανατολισμό του μοντέλου ως προς την Σκηνή.
 - **Translate mode:** αυτή ενδείκνυται για την μεταφορά κάποιου από τα είδη υπάρχοντα μοντέλα.
 - **Rotate mode:** βάση αυτής επιδιώκουμε την περιστροφή ενός από τα παρευρισκόμενα μοντέλα.
 - **Scale mode:** επιτέπει την μεγέθυνση ή σμίκρυνση του επιθυμητού μοντέλου.
 - **Undo/Redo:** εντολές που συμβάλλουν στην επαναφορά ή και επανάληψη μιας ενέργειας/κατάστασης.
 - **Simple shapes:** η εντολή αυτή επιτυγχάνει την αυτόματη επιλογή και εμφάνιση απλών σχημάτων στην Σκηνή. Χαρακτηριστικό παράδειγμα αυτών αποτελεί μια σφαίρα ή ένας κύλινδρος.
 - **Lights:** μια ενέργεια που συμβάλει στην ρύθμιση και τροποποίηση του φωτισμού.
 - **Copy/paste:** εντολές κατάλληλες για αντιγραφή και επικόλληση αντίστοιχα.
 - **Align:** βάση αυτής της εντολής ο χρήστης καταφέρνει να αντιστοιχίσει τα μοντέλα του με την σειρά που εκείνος επιθυμεί.
 - **Snap:** ενέργεια που προκαλεί την προώθηση ενός μοντέλου δίπλα σε ένα άλλο.
 - **Change view:** η εντολή αυτή επιφέρει την αλλαγή της όψης του περιβάλλοντα χώρου.
- **Η Κάτω Γραμμή Εντολών:**

Αυτή η γραμμή εντολών εντοπίζεται στο κάτω μέρος της σκηνής και εμπεριέχει εργαλεία- παραμέτρους, εν μέσου των οποίων ο χειριστής δέχεται πληροφορίες για το “χρόνο προσομοίωσης”, το “πραγματικό χρόνο” και τον δείκτη RTF (Real Time Factor), όπως αντικατοπτρίζεται και στην εικόνα.

Ο “χρόνος της προσομοίωσης” εκφράζει την χρονικό διάστημα που απαιτεί το λογισμικό να φέρει εις πέρας μια συγκεκριμένη προσομοίωση. Ενώ, ο

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

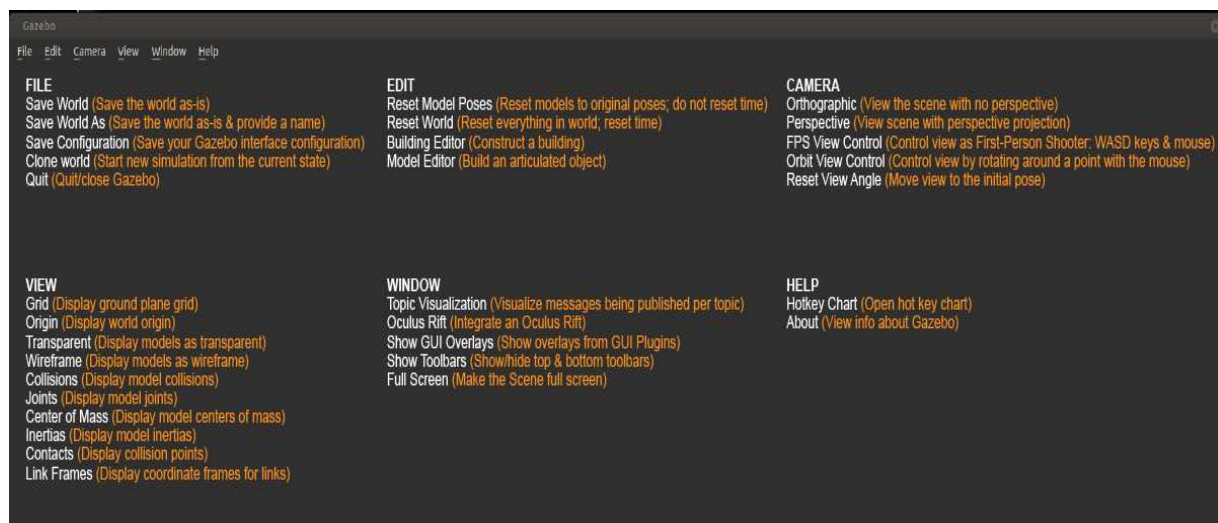
“πραγματικός χρόνος” αναφέρεται απαιτούμενο χρονικό διάστημα που ένα πραγματικό ρομποτικό μοντέλο, υπό πραγματικές συνθήκες θα επαναλάβει επακριβώς την ίδια διαδικασία με αυτή του προσομοιωτή. Και τέλος ο δείκτης RTF, μεταφράζεται ουσιαστικά ως μια μεταβλητή σύγκρισης ανάμεσα στο “χρόνο προσομοίωσης” και τον “πραγματικό χρόνο”, για αυτό και υπολογίζεται από τον λόγο αυτών. Έτσι, την στιγμή που αυτός ο δείκτης συγκλίνει ως προς την τιμή της μονάδας, αυτό αυτομάτως θα υποδηλώνει ότι ο χρόνος με τον οποίο πραγματοποιείται η προσομοίωση είναι ταυτόσημος με τον πραγματικό.



Εικόνα 2: Κάτω γραμμή εντολών (Gazebo)

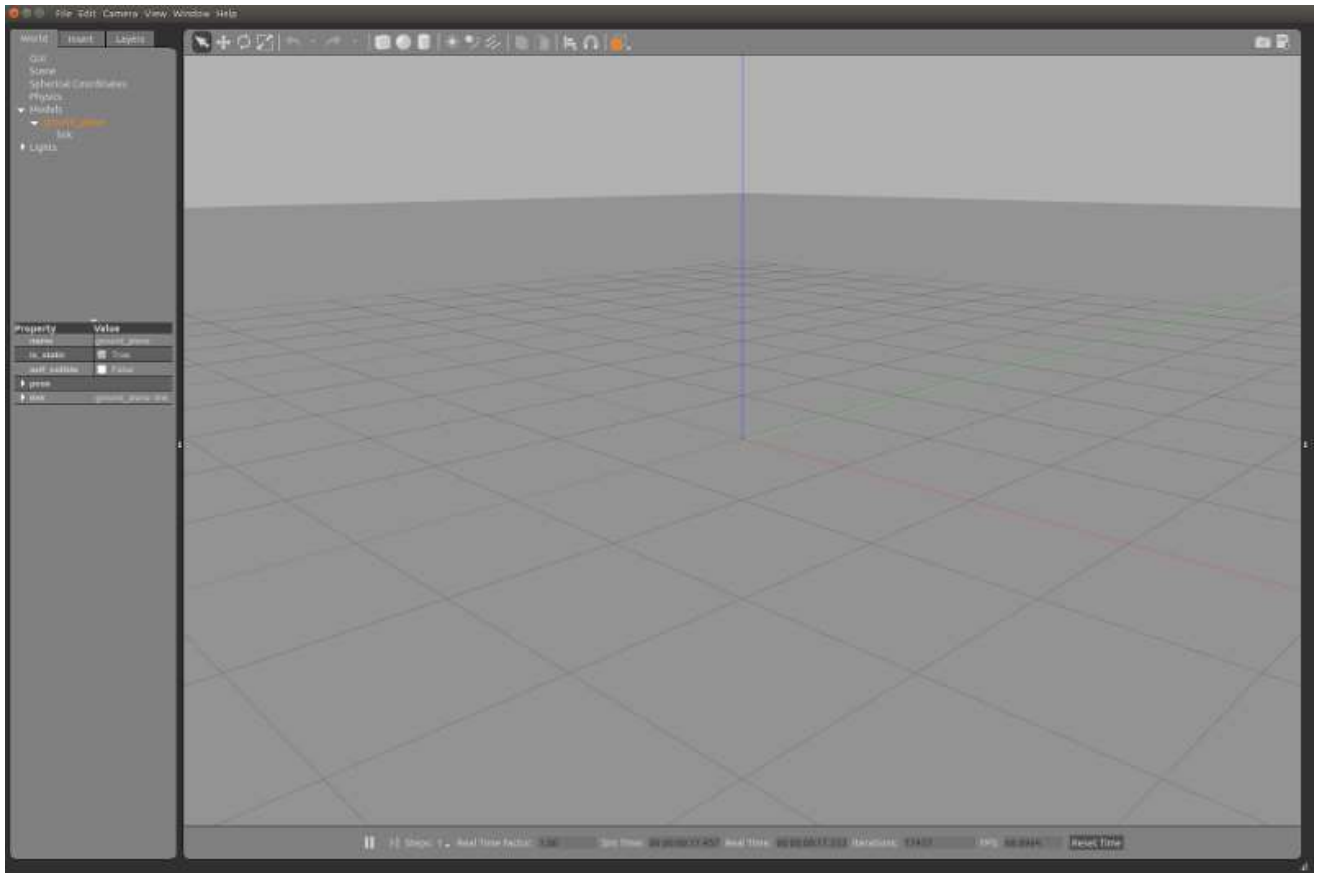
▪ Και το Μενού:

Αυτό συνδυάζει με την σειρά του πολλά από τα στοιχεία που εμπεριέχουν και οι άλλες ομάδες, ενώ φέρει ταυτόχρονα και νέες εξειδικευμένες ιδιότητες, όπως απεικονίζονται και στην ακόλουθη εικόνα.



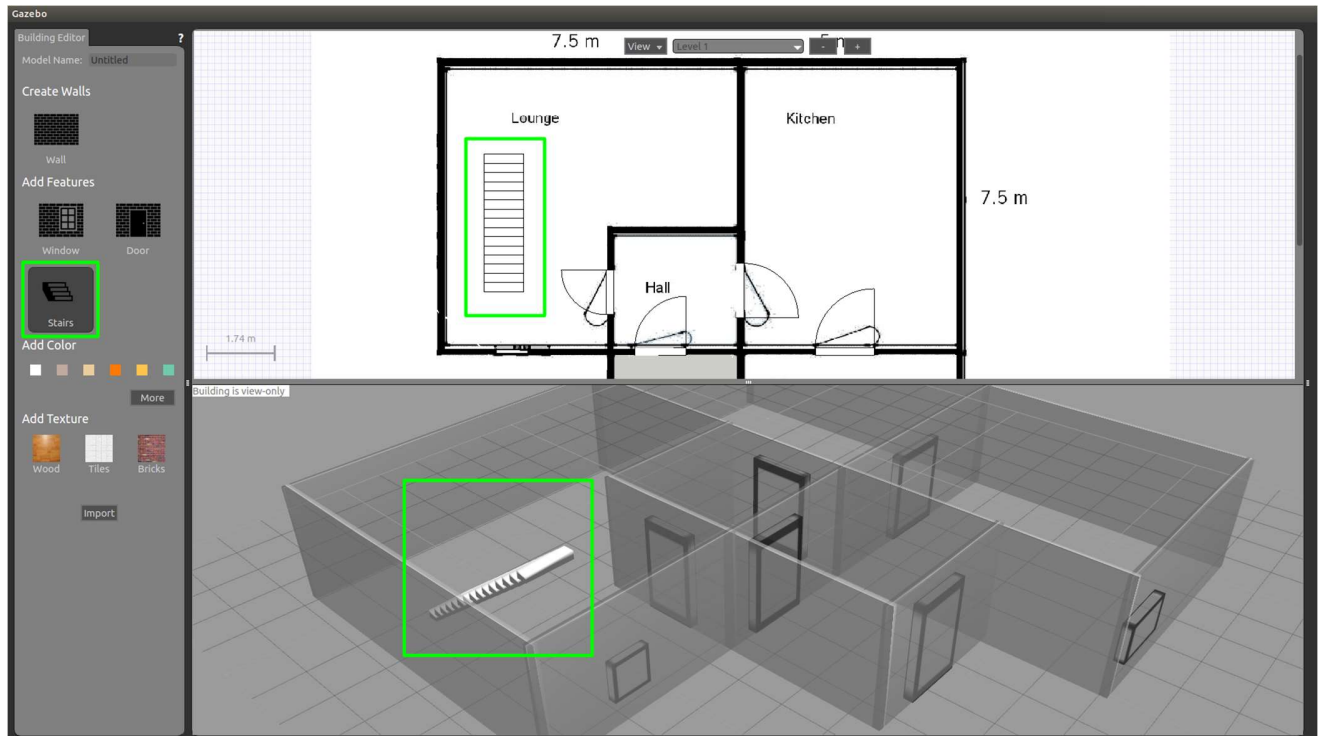
Εικόνα 3: Menu bar (Gazebo)

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 4: Το γραφικό περιβάλλον (Gazebo)

Εκτός, βεβαίως αυτών το γραφικό περιβάλλον παραθέτει στον χειριστή εν μέσω του Building Editor, μια μεγάλη ποικιλία υλικών, υφών, χρωμάτων και άλλων παρόμοιων χαρακτηριστικών, έτσι ώστε να μπορεί να σχεδιάσει με σαφήνεια και σε πραγματικό χρόνο επιθυμητά στατικά αντικείμενα, όπως είναι για παράδειγμα τα κτίρια. Ειδικότερα, χάρις αυτό το πολύτιμο στοιχείο του προσομοιωτή η εικόνα που λαμβάνει ο χρήστης μέσω της οθόνης, του επιτρέπει να ελέγχει το μοντέλο του ταυτόχρονα και από τις τρεις διαστάσεις, καθώς και να το προσχεδιάζει συγχρόνως, εικόνα . Οπότε, καταφέρνει να αναγνωρίζει ευκολότερα τυχόν σφάλματα που προκύπτουν, και συνεπώς επεμβαίνει γρηγορότερα στην εξάλειψη τους.



Εικόνα 5: Building Editor (Gazebo)

4.2.3 Ο επεξεργαστής μοντέλων- Model Editor:

Για την σύνθεση και διαμόρφωση απλών ρομποτικών συστημάτων επι του παρόντος προγράμματος, χρειάζεται να απευθυνθούμε στο Model Editor, δηλαδή τον επεξεργαστή μοντέλων. Αυτός μας επιτρέπει να μοντελοποιήσουμε τα ευκολοδούλευτα συστήματα μας απευθείας στο γραφικό περιβάλλον (GUI), ενώ την αρμοδιότητα, για την κατασκευή απαιτητικών και πολύπλοκων μοντέλων, την λαμβάνουν τα αρχεία SDF.

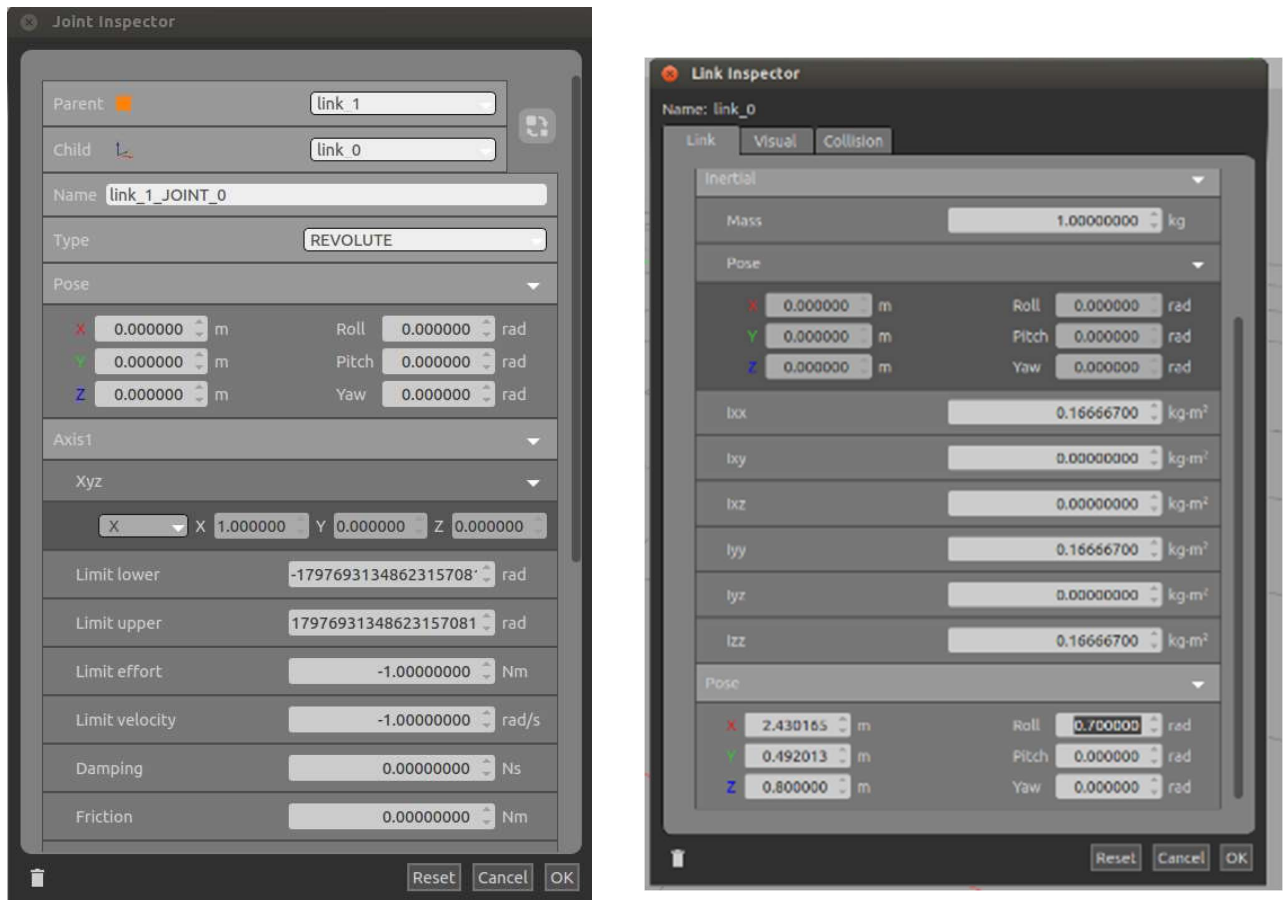
Ο χρήστης εφόσον επιλέξει το Model Editor, επιτυγχάνει ταυτόχρονα την παύση της προσομοίωσης, ενώ στην θέση του γραφικού περιβάλλοντος αυτόματα εμφανίζεται το περιβάλλον του Model Editor το οποίο αν και παρομοιάζει με αυτό του γραφικού διαφέρει σε πολλά τμήματα αυτού. Ειδικότερα, το περιβάλλον του Model Editor φέρει τα εξής εργαλεία:

- Toolbar-Γραμμή εντολών: εμπεριέχει εργαλεία για την επεξεργασία των μοντέλων εντός της Σκηνής, μερικά από τα οποία είναι η περιστροφή (rotate), αναίρεση & επανάληψη (undo/redo), αντιγραφή & επικόλληση (copy/paste) και ευθυγράμμιση (align). Ωστόσο, οι ικανότητες του περιορίζονται και δεν υποστηρίζουν λειτουργίες

του CAD, επεξεργασίες κάποιων ένθετων μοντέλων, ορισμένων γεωμετρικών τύπων (όπως τα Plane-Polyline) και χάρτες ύψους (heightmaps).

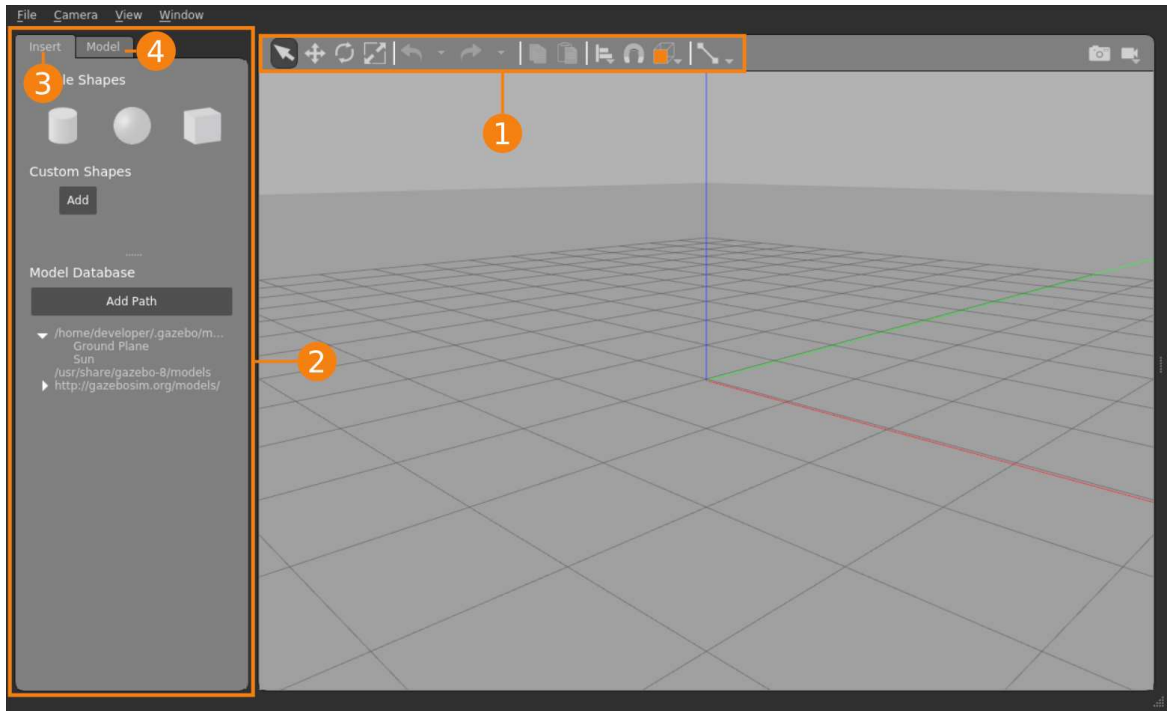
- Palette : αυτή ουσιαστικά αποτελείται από την Καρτέλα Εισαγωγής και Καρτέλα Εξαγωγής.
- Insert tab – Καρτέλα Εισαγωγής: εν μέσω αυτής πραγματοποιείτε η εισαγωγή συνδέσμων και μοντέλων στο Model Editor. Συμπεριλαμβάνει:
 - Simple Shapes: αναφέρεται σε απλά σχήματα, τα οποία μπορούμε να επιλέξουμε για την δημιουργία ενός μοντέλου.
 - Custom Shapes: αφορά την προσθήκη προσαρμοσμένων σχημάτων. Υποστηρίζει αρχεία COLLADA (.dae), 3D Systems (.stl), Wavefront (.obj) και W3C SVG (.svg).
 - Model Database: εμφανίζει μια λίστα μοντέλων τα οποία από την στιγμή που θα εισαχθούν στον περιβάλλον του Model Editor μετονομάζονται σε nested models.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 6: Joint (left) and link inspector(right) of Gazebo

- Model tab – Καρτέλα Μοντέλων: η καρτέλα αυτή εμφανίζει μια σειρά από παραμέτρους που αφορούν το μοντέλο μας. Οι παράμετροι που εμπλέκονται σε αυτή την περίπτωση απευθύνονται στους συνδέσμους, τις αρθρώσεις, τα nested models και τα πρόσθετα (plugins).



Εικόνα 7: Model Tab (Gazebo)

4.2.4 Αρχεία SVG:

Τα SVG αρχεία, στην πραγματικότητα είναι 2D εικόνες, που αποσκοπούν στην δημιουργία 3D πλεγμάτων για τα διερευνόμενα μοντέλα του προσομοιωτή. Βέβαια, πολλές φορές η μοντελοποίηση των επιμέρους κομματιών, από τα οποία και απαρτίζεται ένα τέτοιο μοντέλο, εκτιμάται ευκολότερη με την χρήση ενός ελεύθερου προγράμματος όπως το Inkscape ή το Illustrator.



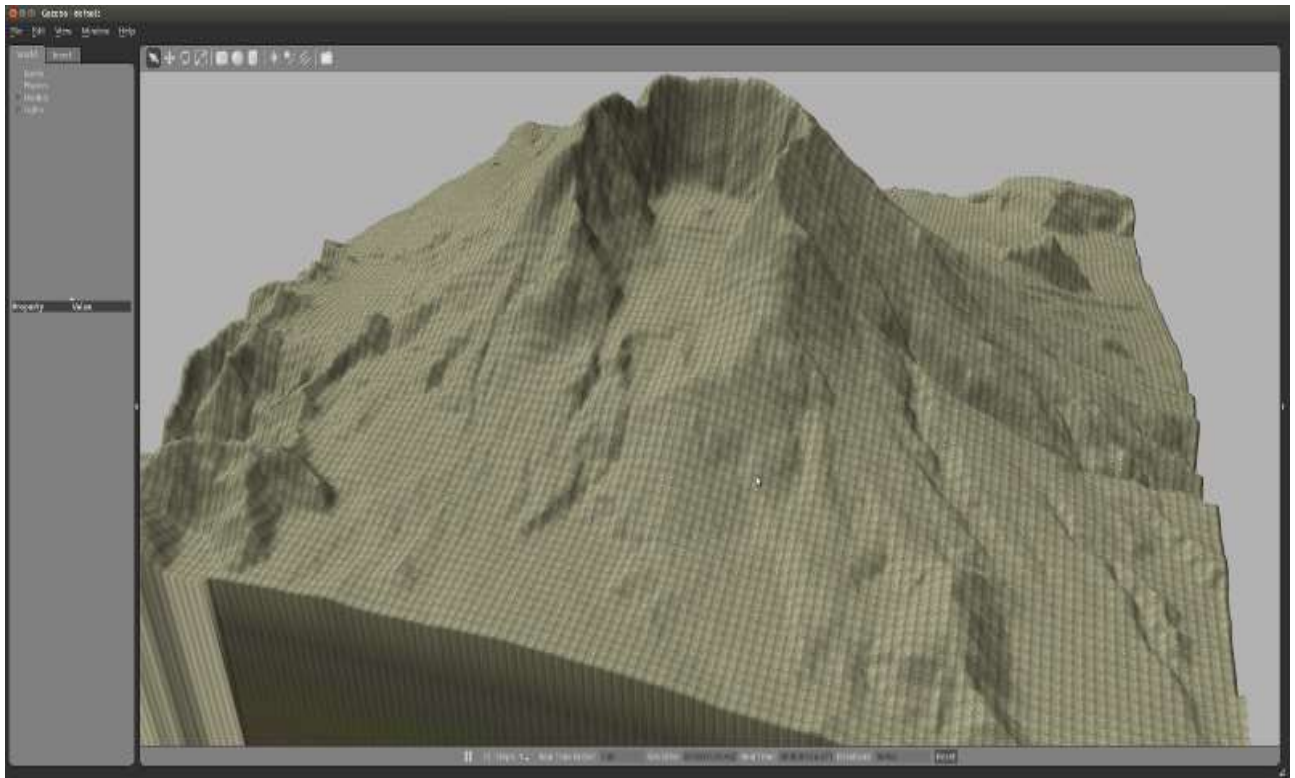
Εικόνα 8: Αρχεία SVG (Gazebo)

4.2.5 Digital Elevation Model:

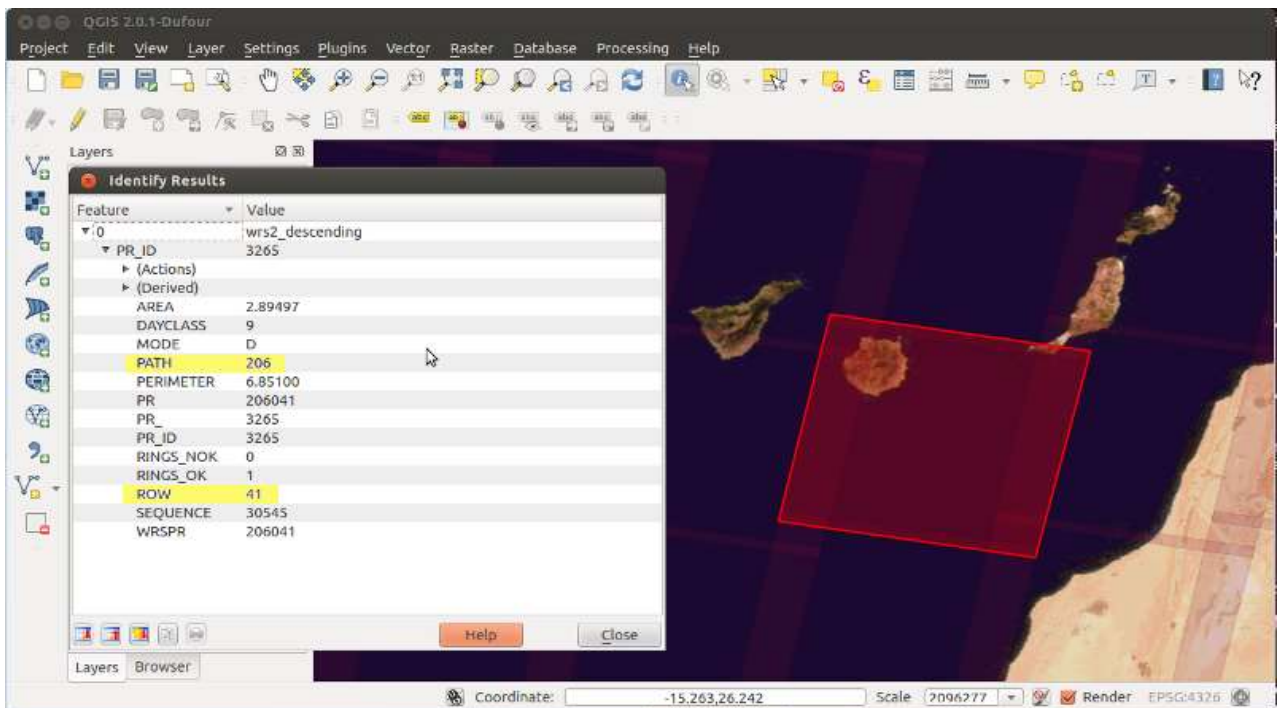
Το Digital Elevation Model (DEM), αποτελεί ένα σπουδαίο εργαλείο στον προσομοιωτή, καθώς δεδομένου αυτού, είναι σε θέση να αναπαριστά με ρεαλιστικό τρόπο, την μορφή μιας επιφάνειας ή ενός εδάφους, μέσα στο οποίο δεν καταγράφονται κτίρια ή καποίο είδος βλάστησης. Έτσι, οι χρήστες μπορούν εν μέσω αυτού, να δοκιμάσουν και να μελετήσουν τον τρόπο με τον οποίο αλληλεπιδρούν τα ρομπότ τους με τα προσομοιωμένα αυτά περιβάλλοντα, πραγματοποιώντας ουσιαστικά εφαρμογές, όπως η διάσωση ή και η γεωργία.

Ωστόσο, για την σύνθεση τέτοιων χώρων, θα πρέπει να ληφθούν υπόψη και να χρησιμοποιηθούν πολυάριθμοι αισθητήρες, τύπου LIDAR, ραντάρ, Real Time Kinematic GPS, τοπογραφικούς χάρτες και κάμερες. Η αναπαράσταση τους, επιπλέον, μπορεί να έχει την μορφή ράστερ- raster (πλέγμα τετραγώνων, επίσης γνωστό ως χάρτης ύψους όταν αντιπροσωπεύει υψόμετρο) ή την όψη τριγωνικού ακανόνιστου δικτύου (TIN). Ως εκ τούτου, μέχρι σήμερα, το Gazebo υποστηρίζει μόνο δεδομένα raster στις υποστηριζόμενες μορφές που διατίθενται στο GDAL. Τέλος, αξιολογήστε τη δράση του και με άλλα εργαλεία, τα οποία χαρίζουν την δυνατότητα στους ενδιαφερόμενους να συνθέσουν αρχεία DEMs, επιλέγοντας εκείνοι την συγκεκριμένη περιοχή. Τέτοια λοιπόν εργαλεία συνιστούν το Global Land Cover Facility, (το οποίο συγκεντρώνει μια ψηφιακή τοπογραφική βάση δεδομένων υψηλής ανάλυσης του πλανήτη), και το QGIS (το οποίο ορίζει ουσιαστικά ένα πρόγραμμα γεωγραφικών πληροφοριών ανοιχτού κώδικα πολλαπλών πλατφορμών που παρέχει δυνατότητες προβολής, επεξεργασίας και ανάλυσης δεδομένων).

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 9: Digital Elevation Model- δεδομένα raster (Gazebo)

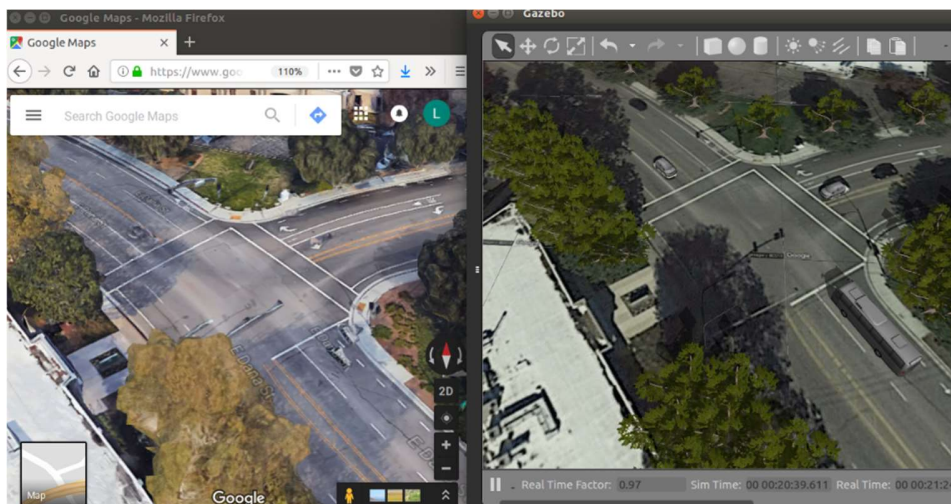


Εικόνα 10: Digital Elevation Model- QGIS (Gazebo)

Πέρα όμως, από την προσομοίωση μιας περιοχής μέσω ενός DEMs, αξιοπρόσεκτη κρίνεται και η προσπάθεια του προσομοιωτή να χρησιμοποιεί δορυφορικές εικόνες, πάνω στις οποίες καταγράφονται επίπεδα μέρη, με σκοπό να επιτυγχάνει εφαρμογές οπτικοποίησης και υπολογιστικής όρασης, σε μεγάλα εξωτερικά περιβάλλοντα, Εικόνα. Αναλυτικότερα, αυτή του η ιδιότητα, τίθεται σε ισχύ με την βοήθεια των πρόσθετων Static Map world plugin και των διεπαφών Google Static Map API. Επιπλέον, τα plugins που συμβάλλουν στην επίτευξη της παρούσας διαδικασίας, εξαρτώνται από κάποιους παραμέτρους, οι κυριότεροι των οποίων είναι:

- `<center>`: Γεωγραφικό πλάτος και μήκος του κέντρου του χάρτη. Οι δύο τιμές πρέπει να είναι οριοθετημένες στο χώρο.
- `<world_size>`: ορισμός του επιθυμητού περιβάλλοντος και κάλυψη μεγέθους του κόσμου (σε μέτρα).
- `<api_key>`: Κλειδί διεπαφής προγραμματισμού εφαρμογών της (API) Google.

Παρόλα αυτά όμως η χρήση του Static Map world plugin είναι περιορισμένη στον Gazebo, καθώς δεν υποστηρίζει ορθογώνιες περιοχές αλλά μόνο τετράγωνα μοντέλα χάρτη με βάση το `<world_size>`.



Εικόνα 11: Real Time Kinematic GPS (Gazebo)

4.2.6 Μέρη του προσομοιωτή:

- Αρχεία World: Τα αρχεία αυτά διαμορφώνονται με την βοήθεια των SDF - Simulation Description Format και αποθηκεύονται υπό την μορφή (**.world**). Στην ουσία ο ρόλος τους είναι να συλλέγουν και να συσσωρεύουν δεδομένα σχετικά με τα υλικά σώματα που θα συμμετάσχουν στην προσομοίωση. Ακριβέστερα συγκεντρώνουν στοιχεία για το ρομπότ, τα φώτα, του αισθητήρες και στατικά αντικείμενα που πρόκειται να συμπεριληφθούν σε μια τέτοια διαδικασία. Έπειτα, τα αρχεία αυτά διαβάζονται από το server του Gazebo (gzserver) και συγχρόνως δημιουργούνται με βάση αυτό.
- Model αρχεία: Είναι αρχεία τα οποία μορφοποιούνται με αντίστοιχο τρόπο όπως τα αρχεία World, δηλαδή με την χρήση των SDF. Ωστόσο σκοπός τους είναι να συμπεριλάβουν ένα και μόνο στοιχείο του μοντέλου το οποίο μπορεί να ενσωματωθεί μετέπειτα στα αρχεία World και να επαναχρησιμοποιηθεί όποτε ζητηθεί. Έτσι, με την παραπάνω τακτική διευκολύνουν και απλοποιούν το έργο των αρχείων World.
- Μεταβλητές χώρου (Environment Variables): αυτές χρησιμεύουν για τον εντοπισμό αρχείων και τον έλεγχο επικοινωνίας μεταξύ server και client και συνήθως λαμβάνουν προεπιλεγμένες τιμές. Οι μεταβλητές αυτές εμφανίζονται στον προσομοιωτή με την εξής ονομασία:
 - GAZEBO_MODEL_PATH
 - GAZEBO_RESOURCE_PATH
 - GAZEBO_MASTER_URI
 - GAZEBO_PLUGIN_PATH
 - GAZEBO_MODEL_DATABASE_URI
- Gazebo Server: το σπουδαίο αυτό εργαλείο επιδιώκει την μοντελοποίηση των υλικών αντικειμένων που συμμετέχουν στην προσομοίωση. Και την ενέργεια αυτή την υλοποιεί βασιζόμενο στα αρχεία World, τα οποία περιλαμβάνουν τα κατάλληλα δεδομένα για αυτά τα αντικείμενα. Εντωμεταξύ, δεδομένου ότι ο gzserver δεν διαθέτει κάποια διεπαφή γραφικών, αδυνατεί να απεικονίσει με κάποιο τρόπο την προσομοίωση στην περίπτωση που λάβει εντολή να δράση μόνος του.
- Graphical Client: αυτό το στοιχείο χρησιμεύει για την τροποποίηση της τρέχουσας προσομοίωσης, ενώ ταυτόχρονα συνδέεται με το gzserver για την απεικόνιση των

αντικειμένων που παραπέμπουν σε αυτή την διαδικασία. Στην προκείμενη λοιπόν περίπτωση την αρμοδιότητα αυτή του gzclient, την έχει αναλάβει το γραφικό περιβάλλον.

- Πρόσθετα (Plugins): Τα plugins συνιστούν κομμάτια κώδικα, τα οποία διαθέτουν την ικανότητα πρωτίστως να μεταγλωττίζονται σε κοινόχρηστες βιβλιοθήκες και εν συνέχεια να εντάσσονται στην προσομοίωση ως κλάσεις της C++. Παραθέτοντας με τον εξής τρόπο, την δυνατότητα στους χρήστες να έχουν άμεση πρόσβαση στις λειτουργίες του προσομοιωτή και συνεπαγόμενα αυτού να μπορούν να ελέγχουν κάθε πτυχή του. Βέβαια, την ενέργεια αυτή εξαρχής, είχαν αναλάβει οι ελεγκτές, όμως με την πάροδο του χρόνου και την εμφάνιση των plugins, διαπιστώθηκε ότι αυτοί υστερούσαν σε σχέση με τα πρόσθετα (plugins). Ακριβέστερα, οι ελεγκτές αφενός λειτουργούσαν όπως και τα πρόσθετα, αφετέρου όμως η διαδικασία μεταγλώττισης τους, ακολουθούσε στατική πορεία. Αντίθετα, τα πρόσθετα θεωρήθηκαν πιο ευπροσάρμοστα, ενώ ταυτόχρονα ο χρήστης επιλέγει μέσω αυτών, ακόμη και την λειτουργία που θέλει να συμπεριλάβει στην τρέχουσα προσομοίωση. Μάλιστα, ως αυτοματοποιημένες υπορουτίνες, μπορούν με σχετική ευκολία είτε να αποσυρθούν είτε να εισαχθούν σε ένα οποιοδήποτε σύστημα, ενώ παράλληλα εκτιμάται ότι καταφέρνουν εξίσου εύκολα να διαχωρίζονται εντός αυτού. Ειδικότερα, η εφαρμογή τους προτρέπεται από τους ειδικούς, σε περιπτώσεις όπου, ο χρήστης:
 - a) στοχεύει την διαμόρφωση της παρούσας προσομοίωσης (για παράδειγμα στην μετακίνηση κάποιων από τα είδη υπάρχοντα μοντέλα).
 - b) Επιθυμεί ταχύτερη διασύνδεση με το Gazebo, χωρίς για παράδειγμα την απαίτηση κάποιας σειριοποίησης.
 - c) Σκοπεύει να μοιραστεί τον κώδικά του με άλλα άτομα, με αποτέλεσμα να τους επωφεληθεί.

Τέλος τα ενεργά αυτά εργαλεία, κατηγοριοποιούνται σε έξι (6) υποομάδες, τις οποίες και μπορούμε να συναντήσουμε στο commandline του προγράμματος ή μέσω SDF αρχείων.

Αυτές οι υποομάδες καταγράφονται στον προσομοιωτή ως έχουν:

1. World: το οποίο έχει οριστεί από την κατασκευή του, να ελέγχει τον κόσμο της προσομοίωσης. Επομένως είναι υπεύθυνο για τα χαρακτηριστικά που διέπουν αυτόν τον κόσμο, όπως και είναι οι μηχανές φυσικής ή ο φωτισμός του περιβάλλοντος, καθώς και άλλα παρόμοια παραδείγματα.
2. Model: ετούτο είναι συνδεδεμένο με τα μοντέλα που συμμετέχουν στην προσομοίωση, και επομένως κάθε ένα από αυτά αποσκοπεί στον έλεγχο ενός τέτοιου τύπου.
3. Sensor: η προσθήκη αυτή έχει οριστεί να εμπλέκει την συγκέντρωση πληροφοριών και την ρύθμιση των ιδιοτήτων που αφορούν τους αισθητήρες.
4. System: αυτή πάλι εμφανίζεται στη γραμμή εντολών, ενώ φορτώνεται αρχικά με την εκκίνηση του Gazebo. Συνεπώς, εν μέσω αυτής ο χρήστης ελέγχει την διαδικασία εκκίνησης του προσομοιωτή.
5. Visual: παραπέμπει στην απεικόνιση της διαδικασίας της προσομοίωσης, τον τρόπο που αποτυπώνονται μέσω της οθόνης τα μοντέλα και αντίστοιχα επιδιώκει την επεξεργασία τους από τον χειριστή.
6. GUI: η προσθήκη αυτή επικοινωνεί με το γραφικό περιβάλλον της προσομοίωσης και επιτρέπει την τροποποίηση αυτού από τον χρήστη.

Έτσι, διαπιστώνουμε ότι κάθε προσθήκη(plug-ins), απευθύνεται και διευθύνει, ένα διαφορετικό κομμάτι του προσομοιωτή.

4.2.7 Gazebo – ROS:

Το Λειτουργικό Σύστημα Ρομπότ- Robotic Operation System (ROS), είναι στην ουσία ένα ελεύθερο λογισμικό ενδιάμεσου επιπέδου- middleware, (αδειοδοτημένο από το Berkeley Software Distribution (BSD)), το οποίο χρησιμεύει στην μοντελοποίηση, τον προγραμματισμό, την ανάπτυξη και τον έλεγχο περίπλοκων ρομποτικών συστημάτων. Η ανάπτυξή του αρχικά ξεκίνησε το 2007, από το εργαστήριο Τεχνητής Νοημοσύνης stanford, ενώ από το 2008 ο Willow Garage προσπάθησε να ενισχύσει την πορεία αυτή ως προς την ανέλιξη του. Προς το παρόν, ωστόσο, το δύσκολο έργο της επίβλεψης και συγχρόνως συντήρησης του εργαλείου αυτού έχει αναλάβει το Open Source Robotics Foundation (OSRF).

Ο απώτερος βεβαίως στόχος, που οδήγησε τους κατασκευαστές, στην δημιουργία ενός τέτοιου λογισμικού, ήταν η ανάγκη των μελετητών-δημιουργών να ανταπεξέλθουν στις δυσκολίες που εμφανίζονταν κατά τα στάδια ανάπτυξης και μοντελοποίησης των πολλαπλών στοιχείων ενός ρομποτικού συστήματος. Έπειτα, όμως με την εφαρμογή και την αναβάθμιση του πολύτιμου αυτού εργαλείου οι χρήστες κατάφεραν να πραγματοποιήσουν και άλλες ενέργειες διευκολύνοντας σε μεγαλύτερο βαθμό το έργο τους. Έτσι πλέον μπορούν, εκτός από να κατασκευάζουν τα μοντέλα τους, να υλοποιούν αλλαγές σε οποιοδήποτε σημείο του κώδικα τους, να τα μοιράζονται με άλλα άτομα, να δοκιμάζονται σε άλλα ρομπότ με ελάχιστες έως και καθόλου αλλαγές και να ‘επικοινωνούν’ με άλλα λογισμικά, όπως και το Gazebo.

Όσο για τον τρόπο λειτουργία του, αυτός στηρίζεται αρχιτεκτονική γραφημάτων με κεντρική τοπολογία, όπου η επεξεργασία πραγματοποιείται και ρυθμίζεται από τους λεγόμενους κόμβους (nodes). Αναφορικά, κάθε κόμβος έχει και τον δικό του σκοπό, δηλαδή το ένα από αυτά μπορεί να συλλέγει δεδομένα από έναν αισθητήρα, ένα άλλο να ελέγχει κάποιον ενεργοποιητή και ου το καθεξής. Ωστόσο, όλα αυτά μπορούν να συλαμβάνουν και να μεταφέρουν πληροφορίες μεταξύ τους μέσω μηνυμάτων.

Το Gazebo, όπως και προηγουμένως έχει ειπωθεί, εντάσσεται στην κατηγορία των αυτόνομων λογισμικών. Παρόλα αυτά μπορεί με ευχέρια να συνεργαστεί και με άλλα λογισμικά πακέτα, όπως το ROS. Σαφέστερα, για να προβεί σε αυτή την ενέργεια ο χρήστης, θα πρέπει πρωτίστως να κοιτάζει κατά την εγκατάσταση του προσομοιωτή, εάν το λογισμικό εμπεριέχει το εξής πακέτο `:gazebo_ros_pkgs`. Οπότε, σε αντίθεση που δεν το βρει θα πρέπει να αναζητήσει την σχετική έκδοση που το περιλαμβάνει και να επαναλάβει την διαδικασία της εγκατάστασης για εκείνη επομένως την μορφή.

Γενικότερα, μερικές από τις δυνατότητες που προσφέρει το ROS στο Gazebo είναι:

- Η πλήρης υποστηριξή των λειτουργιών του αυτόνομου συστήματος Gazebo, χωρίς να έχει δεσμεύσεις με την ROS.
- Δημιουργίες μέσω catkin.
- Επιτυγχάνει και αντιμετωπίζει το ίδιο αποδοτικά την μοντελοποίηση ρομποτικών συστημάτων σε URDF και SDF.
- Βελτιώνει την υποστήριξη εκτός ελέγχου για τους ενεργοποιητές που χρησιμοποιούν το `ros_control`.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Βελτιώνει την αποδοτικότητα των ελεγκτών σε πραγματικό χρόνο, ενσωματώνοντας το DARPA Robotics Challenge.

4.2.8 Τα ρομποτικά συστήματα που υποστηρίζει το λογισμικό:

Ο Gazebo είναι σχεδιασμένο έτσι ώστε να πραγματοποιεί προσομοιώσεις αμέτρητων ρομποτικών συστημάτων. Ειδικότερα, εκτιμάται ιδανική η εφαρμογή του σε μικρούς στόλους οχημάτων, διότι εκεί παρέχει καλύτερα και αξιόπιστα αποτελέσματα. Ωστόσο, υπολείπεται στον τομέα των ιπτάμενων ρομπότ, καθώς αδυνατεί να προσομοιώσει τον αέρα. Παρόλα αυτά μερικά από τα γνωστότερα μοντέλα που εμπεριέχει το πρόγραμμα είναι τα εξής:

- PR2 (Personal Robot 2)
- iRobot Create
- Pioneer 2 DX
- TurtleBot
- Videre Design ERRATIC mobile robot platform
- White Box Robotics 914 PC-BOT
- CoroWare Inc. Corobot and Explorer
- MobileRobots (formerly ActivMedia) PSOS/P2OS/AROS-based robots.
- iRobot Roomba Vacuuming Robot
- Acroname's Garcia
- K-Teams Robotics Extension Board attached to Kameleon 376 BC
- Segway's Robotic Mobility Platform
- UPenn GRASP's Clodbuster
- K-Team's Khepera
- Evolution Robotics ER1, ERSDK robots
- Botrics's Obot d100
- Nomadics NOMAD200 mobile robots
- RWI/iRobot's RFLEX-based robots.

4.3 Προσομοιωτής RoboDK:



Εικόνα 12: RoboDK Logo

Το RoboDK συνιστά ένα δημοφιλές λογισμικό-πρόγραμμα, το οποίο επικεντρώνεται στην προσομοίωση και τον προγραμματισμό των ρομπότ. Μάλιστα, το εύχρηστο αυτό εργαλείο επιτυγχάνει την διαδικασία προγραμματισμού υπό συνθήκες offline. Αναλυτικότερα, αυτό που καταφέρνει ο χειριστής, εν μέσω αυτής της ιδιότητας είναι να προσχεδιάζει τα ρομποτικά του μοντέλα, να τα αναπαριστά μέσω του λογισμικού και γενικότερα να ακολουθεί τα βήματα που οδηγούν στην τελική προσομοίωση αυτού, χωρίς βεβαίως να χρειάζεται για αυτό να είναι εντός του περιβάλλοντος παραγωγής.

Σχετικά, με τα ρομποτικά συστήματα που το πρόγραμμα ενσωματώνει και υποστηρίζει, εκτιμάται, ότι το εύρος τους αγγίζει τα 500 (αλληλόμορφα) μοντέλα. Συνεπώς, αυτό έχει ως αποτέλεσμα, η χρήση του να αυξάνεται και να επεκτείνεται σε μεγάλο βαθμό, σε πολλούς εργασιακούς τομείς, καλύπτοντας, κατά αυτόν τον τρόπο, εξίσου μεγάλο ποσοστό των αναγκών τους. Ωστόσο, πρὸς το παρόν θεωρείται ιδιαίτερα αποδοτικό και αποτελεσματικό σε εργασίες κατασκευής που περιλαμβάνουν βιομηχανικά ρομπότ. Για αυτό και η χρήση του προτείνεται περισσότερο σε αυτές τις περιπτώσεις.

Τα πρώτα βήματα που σηματοδότησαν βεβαίως και την ανάπτυξη του, καταγράφονται το 2015 από τον Albert Nubiola και Lauren Ierullo σε ένα από τα εργαστήρια του Πανεπιστημίου ETS, συγκεκριμένα στο CoRo, το οποίο και βρίσκεται τοποθετημένο στο Μόντρεαλ του Καναδά. Ειδικότερα, σύμφωνα με όσα ισχυρίζεται ο ιδρυτής του προγράμματος Nubiola, το RoboDK εμφανίστηκε την περίοδο που εκείνος πραγματοποιούσε το διδακτορικό του, ενώ αποτέλεσε εκτεταμένη έκδοση του RoKiSim. Επιπλέον, ο ίδιος υποστηρίζει ότι 'το 2015 ένα παραδοσιακό λογισμικό προσομοίωσης

κοστίζει σχεδόν όσο ένα βιομηχανικό ρομπότ. Ωστόσο, η ίδρυση του RoboDK, πρόσφερε ένα λογισμικό προηγμένης τεχνολογίας για την προσομοίωση και τον προγραμματισμό ρομπότ, σε προσιτές τιμές για όλους”.

Σχετικά πάλι με το CoRo, αυτό χαρακτηρίζεται ως ένα από πιο τα “φημισμένα” και ευρέως αναγνωρισμένα εργαστήρια ρομποτικής του Καναδά, καθώς είναι εμπλουτισμένο με πληθωρικά βιομηχανικά-ρομποτικά συστήματα, ενώ παράλληλα ο κύριος στόχος δραστηριοποίησής του είναι αυτός της εφαρμοσμένης έρευνας.

Επίσης, το 2015 το ισχυρό αυτό λογισμικό κατόρθωσε να συνεργαστεί με το πρώτο της πελάτη, ο οποίος αξιοποίησε τις λειτουργίες αυτού, με σκοπό την βαθμονόμηση ενός ρομπότ ABB και την έπειτα υλοποίησή του για την άλεση.

4.3.1 Χαρακτηριστικά του RoboDK:

Ο προσομοιωτής αυτός, όπως και κάθε άλλος προσομοιωτής που προωθείτε διαμέσου της αγοράς, αναπτύσσει μερικά χαρακτηριστικά γνώρισμα. Κάποια από αυτά εμφανίζουν άλλοτε ομοιότητες με τους υπόλοιπους προσομοιωτές και άλλοτε διαφοροποιήσεις. Εντούτοις, όλα τους θεωρούνται σπουδαίες πηγές πληροφόρησης, πάνω στις οποίες βασίζεται η πλειοψηφία των χρηστών, με σκοπό την καλύτερη μάθηση και εξοικείωση του με το πρόγραμμα.

Σαφέστερα, τα χαρακτηριστικά τα οποία εντοπίζονται στο RoboDK, και συνεπαγόμενα προσδιορίζουν και τις υπόλοιπες λειτουργίες αυτού είναι τα ακόλουθα:

- ✓ Το πρόγραμμα υποστηρίζεται σε αρκετά από τα γνωστότερα λειτουργικά συστήματα:
Ειδικότερα το λογισμικό δουλεύει με επάρκεια και το ίδιο αποδοτικά σε λειτουργικά συστήματα όπως τα Windows, Linux, σε Mac ακόμη και σε android.
- ✓ Διαθέτει έξυπνη γραφική διεπαφή
Το χαρακτηριστικό ετούτο επιτρέπει στον χρήστη, να προγραμματίζει εύκολα οποιοδήποτε ρομπότ για οποιαδήποτε κατασκευαστική εφαρμογή, χωρίς να απαιτούνται για αυτό δεξιότητες προγραμματισμού.
- ✓ η απόκτηση του λογισμικού προϋποθέτει πληρωμή:

το προϊόν αυτό μπορεί να μην διανέμεται δωρεάν, όπως πολλά άλλα. Ωστόσο, η τιμή που αναγράφεται για την αγορά αυτού, κυμαίνεται ανάμεσα στα ‘‘κανονικά πλαίσια’’ και χαρακτηρίζεται παρόμοια με την αγορά ενός οποιοδήποτε, άλλου βασικού προγράμματος.

✓ 3D αναπαράσταση εικόνας:

Τα ρομπότ προσομοιώνονται σε ένα τρισδιάστατο περιβάλλον, αποκτώντας με τον ίδιο τρόπο 3D όψη. Χαρακτηριστικό που τα κάνει να προσεγγίζουν με τεράστια πιθανότητα, τον αντίστοιχο πραγματικό κόσμο.

✓ Υποστηρίζει αρχεία της μορφής:

- STL
- STEP (ή STP)
- IGES (ή IGS)
- WRML
- 3DS
- OBJ

Τα αρχεία αυτά επιδιώκουν την απεικόνιση τρισδιάστατων στοιχείων που συμμετέχουν στην προσομοίωση.

✓ Μέσω του προγράμματος ο χρήστης μπορεί να σχεδιάζει, να μοντελοποιήσει, να προσθέσει, καθώς και να βαθμονόμηση τα δικά του εργαλεία: Χαρακτηριστικό παράδειγμα αυτών αποτελεί η βαθμονόμηση του Tool Center Point(TCP). Μια ενέργεια, η οποία χρησιμεύει κυρίως για την μετατόπιση του ρομποτικού μοντέλου σε μια προκαθορισμένη καρτεσιανή θέση .

✓ Απευθείας χρήση και μεταφορά ενός υπολογιστικού προγράμματος σε κάποιο από τα προς μελέτη ρομποτικά μοντέλα:

Το λογισμικό καταφέρει εν μέσω των πρωτόκολλων FTP, να μεταφέρει επιλεγμένα προγράμματα απευθείας στα ρομπότ.

✓ Η χρήση του επεκτείνεται σε πολλές εφαρμογές:

Το λογισμικό κατορθώνει να ανταπεξέλθει τόσο στις απαιτήσεις των διαχειριστών τους όσο και στα εμπόδια που προκύπτουν κατά την διάρκεια δραστηριοποίησής τους,

καλύπτοντας με αυτόν τον τρόπο τεράστια γκάμα εφαρμογών. Μερικές, λοιπόν από τις εφαρμογές, στις οποίες και έχει σημειωθεί αποδοτική συμπεριφορά είναι:

- Τα κατασκευαστικά έργα,
- Το φρεζάρισμα,
- Η συγκόλληση,
- Η επιλογή και μεταφορά (pick and place) ,
- Η παλετοποίηση,
- Η ζωγραφική,
- Η βαθμονόμηση ρομπότ και άλλα.

✓ Υποστηρίζει την Εικονική Πραγματικότητα:

Η εικονική πραγματικότητα- VR(Virtual Reality) που ενσωματώνει το πρόγραμμα, προσφέρει στον χρήστη μια υπέροχη εμπειρία, απόλυτα συνδεδεμένη με την εξοικείωση του τρόπου λειτουργίας του προσομοιωτή. Σαφέστερα, το VR υποστηρίζεται στο RoboDK από τα εξής εργαλεία:

- SteamVR
- HTC VIVE
- Oculus Rift S
- Και HTC Cosmos.

✓ Η βιβλιοθήκη που φέρει το λογισμικό εμπεριέχει πάνω από 500 διαφορετικά ρομποτικά συστήματα:

Αυτά εντάσσονται τα περισσότερα στην κατηγορία των βιομηχανικών ειδών. Οπότε, και αποτελούν κατά βάση βραχίονες ή manipulators.

Επίσης, ο κάθε ενδιαφερόμενος μπορεί με ευκολία να δημιουργήσει τα δικά του μοντέλα. Ακριβέστερα, το λογισμικό υποστηρίζει τους εξής μηχανισμούς διαμόρφωσης:

- Έναν γραμμικό άξονα
- Έναν περιστροφικό άξονα
- Έναν γραμμικό και περιστροφικό άξονα
- Δύο γραμμικούς άξονες
- Δύο περιστροφικούς άξονες

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Τρεις γραμμικούς άξονες
- Αρπάγες με δύο δάχτυλα
- Ρομπότ τύπου SCARA τεσσάρων αξόνων
- 6-αξονικός ρομποτικός βραχίονας
- 7-αξονικός ρομποτικός βραχίονας

Ακόμη, δίνεται η δυνατότητα στον χρήστη, να χρησιμοποιήσει μέσω του λογισμικού τον βραχίονα του είτε ως μια μηχανή CNC είτε και ως 3D printer.

✓ Προσομοιώνει και μετατρέπει ταυτόχρονα NC προγράμματα σε ρομποτικά προγράμματα:

- APT,
- G-code ACL,
- TAP ή NCI

✓ το λογισμικό RoboDK επιτυγχάνει αυτόματα την βελτιστοποίηση της διαδρομής του ρομπότ, ενώ συγχρόνως εστιάζει την προσοχή του, στο να παραμείνει απομακρυσμένος από τα όρια του άξονα καθώς και από τις συγκρούσεις.

✓ Ο προσομοιωτής επιτρέπει την εμφάνιση του χώρου εργασίας ενός ρομποτικού μοντέλου: αποδίδοντας ,με αυτόν τον τρόπο ,στον χειριστή μια πληρέστερη και ξεκάθαρη εικόνα για τα σημεία του χώρου που το ρομπότ μπορεί να προσεγγίσει.

✓ Ικανό να ανιχνεύει πιθανές συγκρούσεις:

Εξαιτίας ενός εργαλείου που το πρόγραμμα φέρει, μπορεί ,εφόσον βεβαίως ο χρήστης το ενεργοποιήσει, να ελέγξει αρχικά το πείραμα για τυχόν συγκρούσεις, και έπειτα σε επόμενο στάδιο να προβεί στην εμφάνιση με κόκκινο χρώμα, των στοιχείων που επηρεάζονται από την σύγκρουση. Έτσι, ώστε να γίνει αντιληπτή όλη η κατάσταση από τον χρήστη και να προσπαθήσει να την αποφύγει.

Παρόλα αυτά, ακόμα και εάν δυσκολεύεται ο χρήστης να προβεί στην αποφυγή της σύγκρουσης, μπορεί και πάλι να απευθυνθεί στο λογισμικό. Και ακολούθως αυτό θα θέσει σε λειτουργία έναν αλγόριθμο, ικανό να δημιουργήσει στο χώρο εργασίας έναν χάρτη που θα οριοθετεί το περιβάλλον στο οποίο το ρομπότ μπορεί να εργάζεται ελεύθερα χωρίς ο χρήστης να ανησυχεί για πιθανές συγκρούσεις. Αυτός λοιπόν ο αλγόριθμος ονομάζεται “πιθανολογικός χάρτης πορείας” (probabilistic roadmaps-PRM) και συγκροτείται από δύο επιμέρους χαρακτηριστικές ομάδες λειτουργίας:

- Την πιθανότητα
- Τον χάρτη τροχιάς

✓ Προγραμματισμός με δημοφιλήs γλώσσες:

Κάτα την διάρκεια εγκατάστασης του RoboDK, παρατηρείται επίσης η αυτόματη εγκατάσταση του Python καθώς και ενός συνόλου δειγματικών μακροεντολών, τα οποία σε συνδυασμό με το RoboDK API παρέχουν την ικανότητα στους χρήστες να μπορούν να διεξάγουν υψηλότερου επιπέδου προσομοιώσεις, ενώ ταυτόχρονα να μπορούν να επέμβουν σε αυτές και να τις προσαρμόζουν ανάλογα με τις δικές τους ανάγκες.

Άρα, ο χρήστηs μπορεί από την ιδιότητα αυτή που τα παραχωρείται, να συμπεριλάβει στην προσομοίωση του τον προγραμματισμό ρομποτικών μοντέλων χρησιμοποιώντας την γλώσσα Python, την προσομοίωση 2D κάμερας, τη μετατροπή αρχείων SVG σε προγράμματα ρομπότ, την αυτόματη ρύθμιση ενός TCP που έχει σταματήσει, καθώς και αρκετά άλλα.

Ωστόσο, του δίνεται η δυνατότητα να επιλέξει και άλλες γλώσσες προγραμματισμού εκτός από την Python και αυτές είναι:

- C #,
- και C ++

✓ offline programming: η διαδικασία αυτή βοηθά τον χρήστη, να ασχολείται με την ταυτόχρονη μελέτη πολλαπλών σεναρίων, επιτυγχάνοντας με τον ακόλουθο τρόπο την τη πιθανότητα έυρεσης σφαλμάτων και την εξάλειψη τους.

✓ Πολλά μακροσκελή προγράμματα δεν υποστηρίζονται:

Αυτό συμβαίνει, διότι ο εκλεκτής διαθέτει συγκεκριμένους περιορισμούς, οι οποίοι σχετίζονται συνήθως με το μέγεθος του αρχείου ή των αριθμό των γραμμών ενός κώδικα.

Παρόλα αυτά, ο προσομοιωτής έχει την δυνατότητα αρχικά να σπάσει αυτά τα μακροσκελή προγράμματα σε μικρότερα υποπρογράμματα, στην συνέχεια να τα τρέξει με την ίδια σειρά και λογική που είχαν αποτυπωθεί, και επομένως να φέρει ένα ικανοποιητικό αποτέλεσμα .

✓ Συνεργασία με λογισμικά όπως το Matlab :

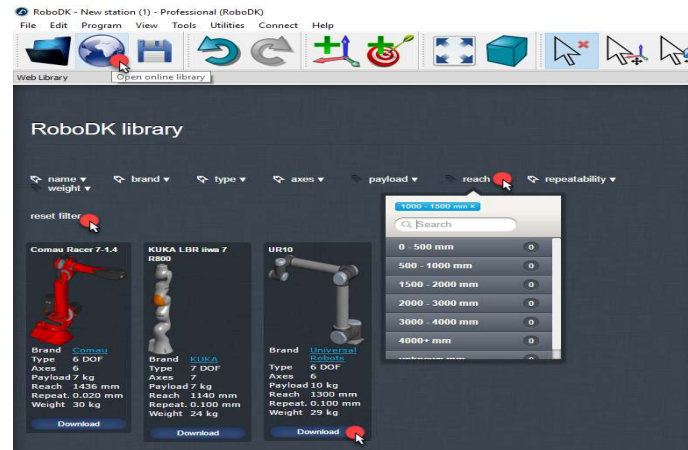
- Με στόχο την σχεδίαση συναρτήσεων, δεδομένων, όπως και εφαρμογή αλγορίθμων.
- ✓ Η ταχύτητα με τη οποία μπορεί να υλοποιηθεί μία προσομοίωση στο RoboDK είναι πέντε φορές μεγαλύτερη από αυτή σε πραγματικό χρόνο:
Πρακτικά, η ταχύτητα προσομοίωσης ή λόγος προσομοίωσης, όπως και αποκαλείται, αναφέρετε στο πόσο αργά ή γρήγορα επιτυγχάνεται μία διαδικασία προσομοίωσης στο RoboDK, σε σχέση με τον πραγματικό κόσμο. Συνεπώς, την στιγμή που ο λόγος αυτός λάβει την τιμή ένα (1), αυτομάτως θα μεταφράζει και την ισότητα μεταξύ του χρόνου της προσομοίωσης και του πραγματικού.
 - ✓ Επίσης το πρόγραμμα μπορεί και προσδιορίζει με επάρκεια το χρόνο ενός κύκλου διεργασιών (cycle time):
Έχοντας, λοιπόν αρχικά διευκρινήσει στο πρόγραμμα τις τιμές των προαπαιτούμενων παραμέτρων, (όπως είναι οι ταχυτητες και επιταχύνσεις κοντά στα όρια του ρομπότ, το ωφέλιμο φορτίο αυτού, η θέση του, καθώς και άλλα) αυτό επείτα μπορεί να προχωρήσει στον υπολογισμό του χρονικού διαστήματος που χρειάζεται για την ολοκλήρωση του κύκλου μιας τέτοιας διεργασίας-ενέργειας. Επίσης, σημειώνεται ότι κατά την διάρκεια του υπολογισμού, το λογισμικό λαμβάνει υπόψη του ότι το ρομπότ κάνει καθορισμένες κινήσεις από σημείο σε σημείο, χωρίς να εξομαλύνει τις γωνίες.
 - ✓ Post processor:
Ο Post Processor συνθέτει στην ουσία είναι ένα εργαλείο- πρωτόκολλο, το οποίο και ακολουθεί το λογισμικό προκειμένου να διαχωρίσει, και να κατανοήσει με ποιόν τρόπο πρέπει να διαμορφωθούν τα προγράμματα που αποσκοπούν σε ένα συγκεκριμένο ελεγκτή ρομπότ.

4.3.2 Το Γραφικό περιβάλλον:

Η πρώτη επαφή του ενδιαφερόμενου με το πρόγραμμα αναπτύσσεται κατά το στάδιο εκκίνησης του. Ειδικότερα, ο χειριστής την στιγμή εκείνη, αυτό που λαμβάνει ως πρώτη όψη είναι το γραφικό περιβάλλον. Ο ιδανικός χώρος δηλαδή, για την επίτευξη την διαδικασίας προσομοίωσης. Πέρα, από το περιβάλλον όμως στο οποίο διαδραματίζεται η προσομοίωση, το γραφικό περιβάλλον φέρει, επιπρόσθετα μια σειρά από εργαλεία και εντολές που υποβοηθούν εξίσου στην διαδικασία αυτή. Για την ακρίβεια, επομένως το γραφικό περιβάλλον, αποτελείται από (Εικόνα):

- Το Μενού (Main Menu): το οποίο βρίσκεται τοποθετημένο στην κορυφή του γραφικού περιβάλλοντος και περιλαμβάνει εντολές όπως:
 - I. File
 - II. Edit
 - III. Program
 - IV. View
 - V. Tools
 - VI. Utilities
 - VII. Connect
 - VIII. Help
- Την γραμμή εργαλείων (Toolbar): αποτελείται από βασικά εργαλεία-εντολές τα οποία συμμετέχουν με σκοπό να διευκολύνουν την προσομοίωση. Τα εργαλεία-εντολές που καταγράφονται στην συγκεκριμένη γραμμή είναι:
 1. **Open:** η εντολή αυτή επιτρέπει την φόρτωση ενός νέου αρχείου από τον φάκελο (όπως, RoboDK RDK Station) ή και την προτίμηση ενός υποστηριζόμενου τύπου (όπως, ένα ρομπότ, κάποιο εργαλείο, ένα STEP, IGES, STL και πολλών άλλων)
 2. **Open online library:** η εντολή αυτή επιτρέπει στον χρήστη να έχει απευθείας πρόσβαση στην λίστα των των ρομποτικών συστημάτων που διαθέτει η βιβλιοθήκη του προγραμματός και να επιλέξει το επιθυμητό για εκείνον (εικόνα).

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 13: RoboDK Library

- 3. Save Station:** αυτή πραγματοποιεί την αποθήκευση του δημιουργούμενου RDK φακέλου.
- 4. Undo/ Redo:** δύο εντολές που χρησιμεύουν για την επαναφορά/επανάληψη μιας κατάστασης.
- 5. Add a reference frame:** τα πλαίσια ή συστήματα αναφοράς, συνιστούν ουσιαστικά τους άξονες συντεταγμένων βάσει των οποίων προσανατολίζονται τα συνιστώμενα εξαρτήματα του ρομπότ στο τρισδιάστατο χώρο που επιτυγχάνεται η προσομοίωση.
- 6. Add a new target:** αυτή η εντολή επιτρέπει στον χρήστη να αποθηκεύει την θέση που βρίσκεται το ρομπότ σε σχέση με το πλαίσιο αναφοράς ή της συντεταγμένες της άρθρωσης.
- 7. Fit All:** η επιλογή της εντολής αυτής προκαλεί την εμφάνιση όλων των στοιχείων σε 3D όψη .
- 8. Isometric View:** μόνο προεπιλεγμένα αντικείμενα απεικονίζονται σε 3D
- 9. Move reference Frames:** επιφέρει την μετατόπιση επιλεγμένων πλαισίων αναφοράς
- 10. Check collisions:** η συγκεκριμένη εντολή επιδιώκει τον έλεγχο τυχόν συγκρούσεων κατά την διάρκεια προσομοίωσης.
- 11. Move TCP (robot tool):** επιτρέπει την μετακίνηση ενός TCP ρομπότ
- 12. Fast simulation/ Pause simulation:** δύο εντολές που αποσκοπούν στην επιτάχυνση/πάυση της προσομοίωσης.
- 13. Add Program:** προσθέτει ένα νέο πρόγραμμα

14. Add Python Program: εισάγει ένα πρόγραμμα Python

15. Move Joint Instruction: επιτρέπει την προσθήκη μια νέας καθοδηγούμενης άρθρωσης

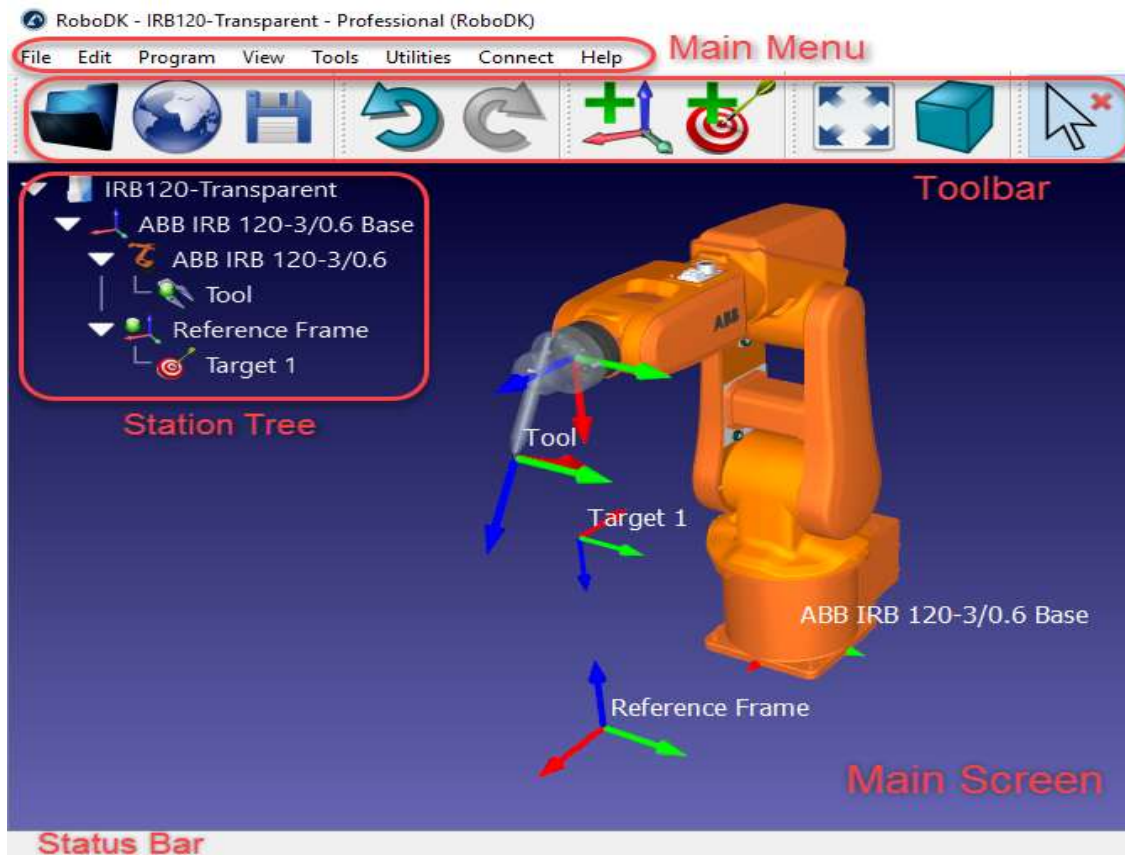
16. Move Linear Instruction: προσθέτει μια νέα μορφή γραμμικής κίνησης.

17. Export Simulation: επιδιώκει την μετατροπή ενός πειράματος προσομοίωσης σε αρχείο 3D PDF ή 3D HTML.

- Μια γραμμή κατάστασης (Status Bar): εμφανίζεται στο κάτω μέρος της οθόνης και λειτουργεί με στόχο την καταγραφή χρήσιμων συμβουλών για ορισμένες λειτουργίες
- Και την κύρια οθόνη (Main Screen): η κύρια οθόνη ουσιαστικά αποτυπώνει το χώρο εργασίας του ρομπότ. Εκτός, όμως αυτού, εμπεριέχει και το Station Tree, το οποίο με τη σειρά του αποτελεί ένα διαγραμματά καταγραφής όλων των στοιχείων που απεικονίζονται εκείνη την στιγμή στο χώρο εργασίας. Επιπλέον, ο όρος στοιχεία μπορεί να απευθύνεται σε ένα από τα ρομπότ, τις αρπάγες αυτών, το πλαίσιο αναφοράς, τους στόχους ή και γενικότερα σε ρυθμίσεις που αφορούν άλλοτε το κατασκευαστικό μέρος των εξοπλισμών και άλλοτε την βαθμονόμηση τους.

Το Station Tree, παρέχει σπουδαίες πληροφορίες, διότι εκτός από το είδος και την ποσότητα των αντικειμένων που συμμετέχουν στην προσομοίωση, περιγράφει και τον “δεσμό” ανάμεσα τους, δηλαδή τον τρόπο με τον οποίο το κάθενα σχετίζεται, και επηρεάζεται από τα υπόλοιπα.

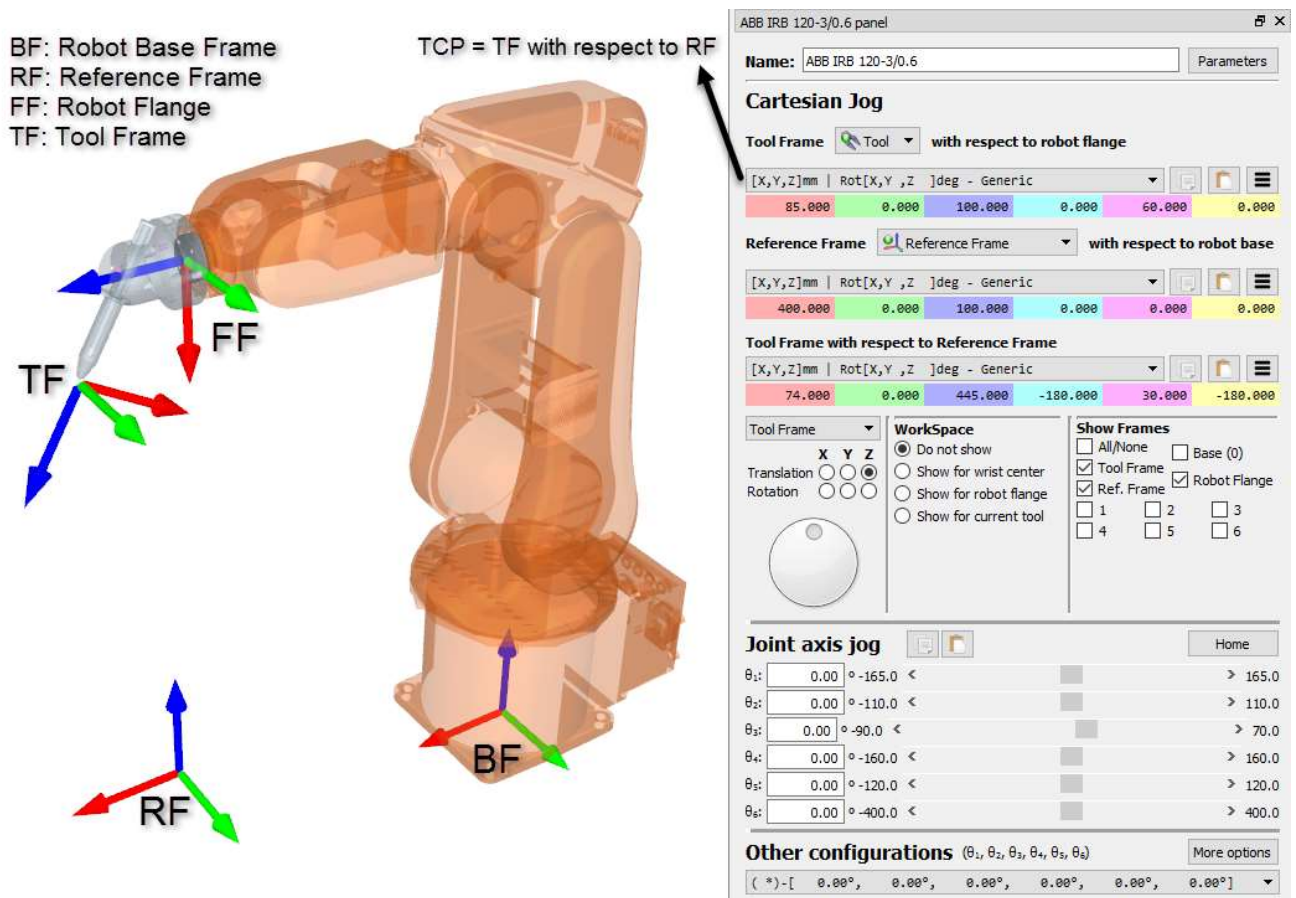
Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 14: Γραφικό Περιβάλλον RoboDK

4.3.3 Το πλαίσιο του ρομπότ (Robot Panel):

Όταν ο ενδιαφερόμενος χρήστης επιθυμεί να συλλέξει δεδομένα σχετικά με την κινηματική συμπεριφορά του ως προς μελέτη ρομπότ, εμφανίζει το Robot Panel (Εικόνα), πατώντας απλώς ένα διπλό κλικ στο εικονιζόμενο μοντέλο.



Εικόνα 15: Robot Panel (RoboDK)

Ετούτο, εκτός από πληροφορίες, στην πράξη, παρέχει την δυνατότητα στον χρήστη να διαχειρίζεται το μοντέλο, επιφέροντας αλλαγές σε αυτό και το περιβάλλον του. Χαρακτηριστικά παραδείγματα, τέτοιων αλλαγών, αποτελούν κατά βάση η μεταβολή των συντεταγμένων της αρπάγης και ολοκληρω του εξοπλισμού, η οριοθέτηση της γωνίας των αρθρώσεων, ακόμη και η βαθμονόμηση τους.

Εκτενέστερα, το πλαίσιο αυτό διαχωρίζεται με τον εξής τρόπο, όπως φαίνεται και από την Εικόνα:

- **Στο Tool Frame (TF) with respect to the Robot Flange (FF):** είναι κομμάτι του πλαισίου, στο οποίο καταγράφονται οι συντεταγμένες και η περιστροφή του εργαλείου τελικής δράσης (ως προς του άξονες X,Y,Z) που φέρει το ρομπότ, αναφορικά με τον άξονα συντεταγμένων του Flange.
- **Στο Reference Frame (RF) with respect to the Robot Base (BF):** είναι κομμάτι του πλαισίου, στο οποίο καταγράφονται οι συντεταγμένες και η περιστροφή (ως προς του

άξονες X,Y,Z) του Reference Frame (αρχικό συστημα αξόνων) , αναφορικά με τον άξονα συντεταγμένων της βάσης του ρομπότ.

- **Στο Tool Frame (TF) with respect to the Reference Frame (RF):** είναι κομμάτι του πλαισίου, στο οποίο καταγράφονται οι συντεταγμένες και η περιστροφή του εργαλείου τελικής δράσης (ώς προς του άξονες X,Y,Z) που φέρει το ρομπότ, αναφορικά με τον άξονα συντεταγμένων Reference Frame(αρχικό συστημα αξόνων).
- **Στο WorkSpace**
- **Στο Show Frames**
- **Στο Joint axis jog:** σε αυτό το κομμάτι του πλαισίου ο χρήστης μπορεί να προβεί είτε σε κάποια τροποποίηση της γωνίας περιστροφής που σχετίζεται με την άρθρωση είτε στην οριοθέτηση αυτής.

4.3.4 Πρόσθετα-Plugins (add-ins):

- **Solidworks:** τα συγκεκριμένα add-in, χρησιμεύουν στην μοντελοποίηση των τρισδιάστατων ρομποτικών συστημάτων, με τα οποία ο χρήστης επιθυμεί να εργαστεί. Επομένως, μπορεί να συμπεριλάβει με αβίαστο τρόπο, στην κατασκευή του, μια σειρά μονοαξονικών, διαξονικών, μέχρι και 7- αξονικών, (που διαθέτουν δηλαδή 7 βαθμούς ελευθερίας), μοντέλων. Τα οποία, ενδεχομένως να εφαρμοστούν σε εργασίες , όπως η συγκόλληση, η κοπή, η διάτρηση, το 3D printing, καθώς και πολλά άλλα.
- **Siemens Solid Edge:** αυτά τα πρόσθετα, προσδίδουν την δυνατότητα στον διαχειριστή, να συνδυάζει κατά την σύνθεση των ρομποτικών συστημάτων, τις λειτουργίες 3D του Solid Edge με εκείνες του RoboDK.
- **Mastercam:** η επιλογή αυτού του πρόσθετου επιτρέπει στον χρήστη, να επιτυγχάνει, τόσο την παράλληλη διαμόρφωση και επεξεργασία μηχανικών σεναρίων για τα ρομποτικά τους συστήματα, όσο και την αποφυγή σφαλμάτων που ενδεχομένως να προέκυπταν κατά το στάδιο αυτό.
- **Rhino:** το λογισμικό υποστηρίζει add-ins, τα οποία λειτουργούν κυρίως για τις εκδόσεις 5 και 6 του Rhino. Ενώ, επιδιώκει μέσω αυτού, επίσης τον σχεδιασμό μηχανικών προγραμμάτων, όπως το Mastercam.

- Mecsoft: αυτό πάλι συνδυάζει τις ιδιότητες τόσο του RhinoCAM , όσο και του Visual CAD/CAM, προκειμένου να επιφέρει αποδοτικότερα αποτελέσματα στην προσομοίωση των ρομπότ που το RoboDK επιτελεί.
- Inventor: ο RoboDK καταφέρνει μέσω των συγκεκριμένων add-in, να “δανειστεί” ουσιαστικά εν μέρη λειτουργίες του Autodesk Inventor, με στόχο να υλοποιήσει με μεγαλύτερη ακρίβεια και αξιοπιστία την τρισδιάστατη συναρμολόγηση, οπτικοποίηση και προσομοίωση των απαιτούμενων ρομποτικών συσκευών.
- FeatureCAM: όσο για αυτά τα add-ins, εύκολα μπορούμε να πούμε ότι προσηλώνονται και επιλέγουν, ιδιότητες του Autodesk FeatureCAM, που σχετίζονται κυρίως με τον προγραμματισμό των ρομπότ.
- Fusion 360: Τέλος, ετούτα τα add-ins, επιτρέπουν την πλήρη αξιοποίηση των λειτουργιών που το Fusion 360 προσφέρει, ώστε ο χειριστής να μπορεί να σχεδιάζει με ασφάλεια και σαφήνεια τα ρομπότ του, ενώ ταυτόχρονα μπορεί να προγραμματίζει και να χρησιμοποιεί τέτοιου είδους μοντέλα, (που πλησιάζουν την κλίμακα των 400), απευθείας από Autodesk Fusion 360.

4.3.5 Ρομποτικά συστήματα:

Ο RoboDK χαρακτηρίζεται ως ένας προσομοιωτής, ο οποίος αφοσιώνεται κυρίως στην εφαρμογή των ρομποτικών συστημάτων του σε βιομηχανικό επίπεδο.

Μερικά από τα πιο πολυαναγνωρισμένα μοντέλα που διατίθενται στην βιβλιοθήκη του προσομοιωτή RoboDK, έως σήμερα είναι τα:

- 1) ABB robots: η βιβλιοθήκη του προσομοιωτή ενσωματώνει 97 τέτοια μοντέλα
 - Τύπος: 6DOF, Delta, palletizing, Ext. axis
 - Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.5kg και 2000 kg
 - Άξονες: αυτοί μπορεί να λαμβάνουν τιμές 1, 2, 4 και 6
 - Βάρος: το ελαφρύτερο αυτού του είδους είναι 25 kg, ενώ το βαρύτερο αγγίζει την τιμή 4525 kg.
- 2) AUBO: σε αυτήν την υποκατηγορία εντάσσονται τρία διαφορετικά είδη (Aubo-i3,i5,και i10)
 - Τύπος: 6DOF

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 10 kg
- Άξονες: 6
- Βάρος: 15 kg, 24kg, 37kg.

3) Adept:

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 2kg και 20 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 20 kg, και το βαρύτερο 280 kg

4) Annin Robotics:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 10 kg
- Άξονες: 6
- Βάρος: 10 kg.

5) Codian Robotics:

- Τύπος: Delta
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 2kg και 60 kg
- Άξονες: αυτοί μπορεί να λαμβάνουν τιμές 3, 4 και 6
- Βάρος: το βάρος αυτών αγγίζει τιμές από 75 kg έως 100 kg

6) Comau:

- Τύπος: 6DOF, palletizing
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 650 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 11kg, ενώ το βαρύτερο αγγίζει την τιμή 2703 kg

7) Denso robots:

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 2kg και 20 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 15 kg, ενώ το βαρύτερο αγγίζει την τα 56 kg

8) Dobot:

- Τύπος: 6DOF, palletizing, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.3kg και 5 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 2.40 kg, ενώ το βαρύτερο αγγίζει την τιμή 23kg

9) Doosan Robotics:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 5kg και 20 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 27 kg, ενώ το βαρύτερο είναι 33 kg

10) Efort:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 20 kg
- Άξονες: 6
- Βάρος: 220kg

11) Elibot:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 12 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 29kg, ενώ το βαρύτερο αγγίζει την τιμή 60 kg

12) Epson:

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.5kg και 2000 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 8 kg, ενώ το βαρύτερο αγγίζει την τιμή 64 kg

13) Estun:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 16 kg
- Άξονες: 6
- Βάρος: 170kg

14) Exechon:

- Αντοχή σε φορτίο: 50kg
- Άξονες: 5
- Βάρος: 300kg

15) Fanuc robots:

- Τύπος: 5DOF, 6DOF, Delta, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.5kg και 1700kg
- Άξονες: αυτοί μπορεί να λαμβάνουν τιμές 4, 5 και 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 14 kg, ενώ το βαρύτερο αγγίζει την τιμή 12500 kg

16) GSK:

- Τύπος: 6DOF, palletizing
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 165kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 180 kg, ενώ το βαρύτερο 1650 kg.

17) HIWIN:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 5kg και 12 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 40 kg, ενώ το βαρύτερο αγγίζει την τιμή 152kg.

18) Han's:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 10 kg
- Άξονες: 6

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Βάρος: το ελαφρύτερο αυτού του είδους είναι 15 kg, ενώ το βαρύτερο 45kg.

19) Hanwha:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 12 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 13 kg, ενώ το βαρύτερο 51kg.

20) Hyundai robotics:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 6kg και 50 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 145 kg, ενώ το βαρύτερο 645kg.

21) Jaka:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 18 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 14 kg, ενώ το βαρύτερο αγγίζει την τιμή 41 kg.

22) KUKA robots: το λογισμικό εμπεριέχει 152 διαφοροποιημένα μοντέλα τέτοιου είδους

- Τύπος: 6DOF, 7DOF, palletizing, Ext. axis, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 6500 kg
- Άξονες: αυτοί μπορεί να λαμβάνουν τιμές 1, 2, 4, 6 και 7
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 20 kg, ενώ το βαρύτερο αγγίζει την τιμή 4740 kg.

23) Kawasaki:

- Τύπος: 6DOF, Delta, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 2kg και 700 kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 20 kg, ενώ το βαρύτερο αγγίζει την τιμή 2800kg.

24) Kinova:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Τύπος: 6DOF,7DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 14 kg
- Άξονες: 6 και 7
- Βάρος: το ελαφρύτερο είναι 7.20 kg, και το βαρύτερο 30kg

25) Leantec:

- Τύπος: 6DOF, palletizing
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 60kg
- Άξονες: 4 και 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 32 kg, ενώ το βαρύτερο 780kg.

26) MIP Robotics:

- Τύπος: scara
- Αντοχή σε φορτίο: 5kg
- Άξονες: 4
- Βάρος: 14kg.

27) Mecademics:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 0.5kg
- Άξονες: 6
- Βάρος: 4.50kg.

28) Mitsubishi:

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 2kg και 13kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 17 kg, ενώ το βαρύτερο αγγίζει την τιμή 130 kg.

29) Motoman robots:

- Τύπος: 6DOF, 7DOF,Delta, palletizing, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 1kg και 800 kg
- Άξονες: αυτοί μπορεί να λαμβάνουν τιμές 4, 6 και 7
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 7 kg, ενώ το βαρύτερο 3050kg.

30) Nachi:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3.5kg και 500 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 25 kg, ενώ το βαρύτερο αγγίζει την τιμή 3000 kg.

31) Niryo:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 1kg
- Άξονες: 6
- Βάρος: 10kg.

32) OTC Daihen:

- Τύπος: 6DOF, 7DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 166 kg
- Άξονες: 6 ή 7
- Βάρος: το ελαφρύτερο είναι 140 kg, ενώ το βαρύτερο 1010 kg.

33) Puma:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: 2.5kg
- Άξονες: 6
- Βάρος: 55 kg.

34) Panasonic:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 6kg και 8kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 145 kg, ενώ το βαρύτερο 203 kg.

35) Physic Instrument:

- Αντοχή σε φορτίο: 10kg
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 10 kg, ενώ το βαρύτερο 12kg.

36) Precise:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 1kg και 3kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 14 kg, ενώ το βαρύτερο 30kg.

37) Rainbow Robotics:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 5kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 21 kg, ενώ το βαρύτερο 40 kg.

38) Robostar:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 12kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 27 kg, ενώ το βαρύτερο 38 kg.

39) RobotiQ:

- Τύπος: αρπάγη
- Αντοχή σε φορτίο: 5kg
- Άξονες: 1

40) Rockae:

- Τύπος: 6DOF, Scara
- Αντοχή σε φορτίο: 7kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 50 kg, ενώ το βαρύτερο 76 kg.

41) Servotronix:

- Τύπος: Delta
- Αντοχή σε φορτίο: 0.3
- Άξονες: 3
- Βάρος: 10kg.

42) Siasun:

- Τύπος: 6DOF, 7DOF

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 80kg
- Άξονες: 6 ή 7
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 22 kg, ενώ το βαρύτερο 500kg.

43) Staubli:

- Τύπος: 5DOF, 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.5kg και 220kg
- Άξονες: 5 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 14 kg, ενώ το βαρύτερο 250 kg.

44) Techman Robot:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 14kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 25 kg, ενώ το βαρύτερο 33kg.

45) Toshiba:

- Τύπος: Scara
- Αντοχή σε φορτίο: 10kg
- Άξονες: 4
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 23 kg, ενώ το βαρύτερο 33kg.

46) uFactor:

- Τύπος: 6DOF, Palletizing
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 0.5kg και 5kg
- Άξονες: 4 ή 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 2.20 kg, ενώ το βαρύτερο 10 kg.

47) Yamaha:

- Τύπος: Scara
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 20kg
- Άξονες: 4
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 17 kg, ενώ το βαρύτερο 52 kg.

48) Universal Robots:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 3kg και 16kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 11 kg, ενώ το βαρύτερο 23 kg.

49) Omron Robots:

- Τύπος: 6DOF
- Αντοχή σε φορτίο: η τιμή του κυμαίνεται μεταξύ 4kg και 14 kg
- Άξονες: 6
- Βάρος: το ελαφρύτερο αυτού του είδους είναι 25 kg, ενώ το βαρύτερο 33kg.

4.4 Webots προσομοιωτής:



Εικόνα 16: Webots Logo

Το πρόγραμμα Webots συνιστά μια τρισδιάστατη ρομποτική πλατφόρμα προσομοίωσης, ανοικτού- κώδικα (αδειοδοτημένη από το Apache 2.0), η οποία διαθέτει κυρίως εκπαιδευτικό σκοπό και ερευνητικό χαρακτήρα, ενώ ταυτόχρονα η εφαρμογή της εντοπίζεται και στον βιομηχανικό τομέα. Ανακλύφθηκε για πρώτη φορά, από τον Olivier Michel το 1996, σε ένα από τα Πανεπιστήμια της Ελβετίας, και συγκεκριμένα στο Swiss Federal Institute of Technology (EPFL)(το οποίο βρίσκεται τοποθετημένο στο Lausanne). Ωστόσο, στην μετέπειτα ανάπτυξη και ολοκλήρωση του προγράμματος, κινητοποιήθηκε και συμμετείχε ενεργά από το 1998 έως σήμερα, η εταιρία Cyberbotics Ltd.

4.4.1 Χαρακτηριστικά προσομοιωτή:

Όπως και οι προηγούμενοι προσομοιωτές που αναφέραμε, έτσι και ο Webots εμφανίζει μεμονωμένα χαρακτηριστικά και ιδιότητες, χάρις των οποίων αφένος και εξασφαλίζεται η σωστή λειτουργία του, και αφετέρου επιδιώκεται η ιδιαιτερότητα του στην αγορά. Αυτό έχει ως συνέπεια, να βρίσκεται σε πιο ευεργετική θέση, όσον αφορά την επιλογή του από το αγοραστικό κοινό, σε σχέση με άλλους προσομοιωτές που κυκλοφορούν.

Ετούτα, τα χαρακτηριστικά επομένως είναι:

- ✓ Υποστηρίζεται και εφαρμόζεται άψογα από πολλά λειτουργικά συστήματα, ωστόσο προυποθέτει κάποιες προδιογραφές:

Και τα λειτουργικά συστήματα είναι:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Windows: υποστηρίζεται όμως μόνο για Windows 10 και 8.1
- Mac: κυρίως macOS 10.15 "Catalina" και 10.14 "Mojave".
- Linux: υποστηρίζεται μόνο από εκδόσεις του Ubuntu 16.04 έως και 20.04 και από RedHat, Mandrake, Debian, Gentoo, Arch, SuSE, and Slackware.

Αλλά για να μπορέσει να εγκατασταθεί προϋποθέτει έναν αρκετά καλό υπολογιστή, εξοπλισμένο με διπύρηνο (2 GHz dual core CPU clock speed) ή τετραπύρηνο (quad-core) επεξεργαστή, μνήμη 2 GB RAM, και καρτα γραφικών NVIDIA και AMD OpenGL με τουλάχιστον 512 MB μνήμη RAM.

✓ Το πρόγραμμα διατίθεται ήδη μεταφρασμένο σε πολλές γλώσσες:

Έτσι άτομα που δεν είναι καλοί γνώστες Αγγλικών, ή που διαθέτουν ως μητρική τους γλώσσα μια από τις επακόλουθες, διευκολύνονται και αποκτούν περισσότερη άνεση στην χρήση και την λειτουργία του προγράμματος. Ειδικότερα οι γλώσσες στις οποίες αναφερόμαστε, είναι: Γαλλικά, Γερμανικά, Ισπανικά, Κινέζικα και Γιαπωνέζικα.

✓ Δυνατότητα μοντελοποίησης και προσομοίωσης κινούμενων μοντέλων:

Το κάθε άτομο επιχειρεί μέσω του προγράμματος να σχηματίσει τα δικά του υπαρκτά κινούμενα μοντέλα και στην συνέχεια τα εξετάζει μέσω της προσομοίωσης για την αποτελεσματικότητά τους. Ειδικότερα, ετούτα τα μοντέλα μπορεί να εκφράζουν παθητικά ή ενεργά αντικείμενα και επομένως να αποτυπώνονται στην οθόνη με τροχούς, πόδια ή ιπτάμενα ρομποτικά εξαρτήματα.

✓ Περιλαμβάνει ικανοποιητικό αριθμό αισθητήρων, ενεργοποιητών και ρομπότ.

Εκ των οποίων, μερικοί είναι αισθητήρες απόστασης, κάμερες, αισθητήρες αφής, πομποί ή δέκτες.

✓ Κάθε μοντέλο έχει την ικανότητα να προγραμματιστεί ξεχωριστά από τα άλλα και με τον δικό της τρόπο.

✓ Επιτρέπει τον προγραμματισμό των ρομπότ μέσω C, C++, Python, BotStudio,Java και MATLAB.

Οπότε, απαραίτητο για κάποιον που επιδιώκει να δημιουργήσει μια προσομοίωση είναι η βασική γνώση ενός τουλάχιστον από τις παραπάνω γλώσσες προγραμματισμού. Το **BotStudio** είναι και το πιο εύκολο από τα παραπάνω, αφού είναι γραφικού χαρακτήρα για αυτό και προτείνεται κυρίως για εκείνους που δεν γνωρίζουν κανένα από τα υπόλοιπα. Παρόλα αυτά,

εάν κάποιος επιλέξει να αναπτύξει τα δικά του πρότυπα, απαιτείται πρωτίστως να αναπτυχθεί μια εξοικείωση με τα τρισδιάστατα υπολογιστικά γραφικά περιβάλλοντα και την γλώσσα περιγραφής **VRML97(Virtual Reality Modeling Language)**.

- ✓ Η αξιοποίηση του προγράμματος στους διάφορους επαγγελματικούς κλάδους, έχει συμβάλει συνειδητά στο έργο πολλών επαγγελματιών:

Παρόλο που η λειτουργία του επικεντρώνεται κυρίως στα ερευνητικά και εκπαιδευτικά προγράμματα των κινούμενων ρομπότ. Το πρόγραμμα βρίσκει ευρεία εφαρμογή στην ακαδημαϊκή έρευνα, αυτοκινητοβιομηχανία, αεροναυτική, βιομηχανία ηλεκτρικών σκουπών, βιομηχανία παιχνιδιών, κινηματική των ρομπότ, νευρωνικά δίκτυα, AI-Τεχνητή Νοημοσύνη, διδασκαλία ρομποτικής, διαγωνισμούς ρομπότ καθώς και αρκετών άλλων.

- ✓ Προσφέρει την δυνατότητα προβολής της προσομοίωσης μέσω:

- Joystick API
- Virtual Reality Headset
- HTC Vive
- Oculus Rift

- ✓ Το πρόγραμμα παρέχει τυπικές υπηρεσίες λειτουργικού συστήματος τα οποία εμπεριέχονται στο πρόγραμμα ROS (Robot Operating System):

Βέβαια, η χρήση των πακέτων που περιέχει το ROS γίνεται με δύο τρόπους:

1. Το standard ROS controller, το οποίο είναι κατάλληλο μόνο για Linux.
2. Το custom ROS controller, το οποίο λειτουργεί μόνο για Windows και MacOS.

Παρόλα αυτά, το custom ROS controller απαιτεί ο χειριστής να αναπτύξει δικούς του ελεγκτές.

- ✓ Τα πρωτοκόλλα TCP / IP, παρέχουν κάποιες διεπαφές οι οποίες μπορούν να εφαρμοστούν προκειμένου: να χρησιμοποιηθούν για παράδειγμα άλλες γλώσσες προγραμματισμού ή να βάλουμε πολλά προσομοιωμένα ρομπότ στον ίδιο κόσμο χρησιμοποιώντας αρκετές παρουσίες του ίδιου ελεγκτή, να ελέγξουμε ένα πραγματικό ρομπότ από το λογισμικό τρίτου ή να διαδώσουμε τα προγράμματα του ελεγκτή σε ένα δίκτυο υπολογιστών.

4.4.2 Προσομοίωση μέσω Webots:

Η προσομοίωση από μόνη της είναι εξαρτώμενη από συγκεκριμένες “μεταβλητές” οι οποίες προφανώς πρέπει να λάβουν μέρος προκειμένου να πραγματοποιηθεί. Και αυτές είναι οι ακόλουθες:

- I. Το **world file**: Αυτός ο φάκελος περιλαμβάνει στοιχεία σχετικά με τις ιδιότητες του ρομπότ, τα αντικείμενα από τα οποία αυτό απαρτίζεται, την θέση, τον προσανατολισμό, την γεωμετρία, την εμφάνιση, το χρώμα, τις φυσικές ιδιότητες αυτών των αντικειμένων, και γενικά το περιβάλλον το οποίο έχει αναπτυχθεί. Επίσης μπορεί να εμφανίζει και το όνομα για παράδειγμα του ελεγκτή που χρειάζεται για κάθε ρομπότ, όμως δεν πρόκειται να εμφανίσει κώδικες αυτών. Αξιοσημείωτο είναι το γεγονός ότι όλες αυτές οι πληροφορίες οι οποίες αποθηκεύονται στο φάκελο, βρίσκονται κατοχυρωμένες με προκαθορισμένο τρόπο. Για παράδειγμα ένα αντικείμενο μπορεί να περιλαμβάνει ένα ή και περισσότερα αντικείμενα (όπως ένα ρομπότ το οποίο αποτελείται από μια αρπάγη), οπότε στην ουσία στο φάκελο αναλύεται και η συνδεσμολογία τους. Η αποθήκευση τους στην μνήμη του προγράμματος γίνεται εφόσον πρώτα έχουν αποθηκευτεί ως αρχεία (.wbt).

Από μόνο του το πρόγραμμα κατά την εγκατάσταση, παρέχει ένα πακέτο με αρχεία τύπου world, τα οποία χρησιμεύουν ως παραδείγματα και μπορούν να δοκιμαστούν οποιαδήποτε χρονική στιγμή. Ετούτα είναι:

1. Demos: εδώ βρίσκονται αποθηκευμένα συνολικά έξι παραδείγματα:
 - anaglyph.wbt: για το πώς δύο κάμερες μπορούν να αναπαριστούν το ανάγλυφο μιας περιοχής.
 - gantry.wbt: ένα ρομπότ τύπου gantry ακολουθεί ένα παιχνίδι ("Towers of Hanoi").
 - hexarod.wbt: ένα ρομπότ το οποίο έχει σχήμα εντόμου και κινείται.
 - moon.wbt: δύο ρομπότ τύπου koala έχουν τοποθετηθεί στο φεγγάρι και κάνουν κύκλους πάνω σε αυτό.
 - soccer.wbt: δύο ομάδες ρομπότ παίζουν ποδόσφαιρο έχοντας ως διαιτητή έναν επόπτη ελεγκτή.
 - stewart_platform.wbt: μια πλατφόρμα τύπου Stewart η οποία είναι ένα είδος παράλληλου χειριστή που χρησιμοποιεί ένα οκταεδρικό συγκρότημα γραμμικών ενεργοποιητών.

2. Devices: αυτός ο υποκατάλογος παραθέτει παραδείγματα μεμονομένων συσκευών και των API τους. Μερικά από αυτά καταγράφονται παρακάτω.

- accelerometer.wbt
- battery.wbt
- brake.wbt
- bumper.wbt
- compass.wbt
- camera.wbt
- camera_auto_focus.wbt
- camera_motion_blur.wbt
- camera_noise_mask.wbt
- camera_recognition.wbt
- connector.wbt
- display.wbt
- distance_sensor.wbt
- emitter_receiver.wbt
- encoders.wbt
- force_sensor.wbt
- force3d_sensor.wbt

3. How To: εδώ βρίσκονται αρχειοθετημένα μερικά παραδείγματα τα οποία περιγράφουν την φυσική συμπεριφορά των υπο μελέτη συσκευών και τον τρόπο λειτουργίας τους.

- asymmetric_friction1.wbt
- asymmetric_friction2.wbt
- binocular.wbt
- biped.wbt
- center_of_mass.wbt
- custom_html_robot_window.wbt
- cylinder_stack.wbt
- force_control.wbt
- four_wheels.wbt
- inverted_pendulum.wbt
- mouse_events.wbt
- omni_wheels.wbt
- passive_dynamic_walker.wbt
- pedal_racer.wbt
- physics.wbt
- rope.wbt
- sick_terrain_scanning.wbt
- spinning_top.wbt
- supervisor_draw_trail.wbt
- texture_change.wbt
- vision.wbt
- ziegler_nichols.wbt

4. Geometries: Αυτή η ενότητα δείχνει τα γεωμετρικά χαρακτηριστικά και ιδιότητες των συσκευών τα οποία υποστηρίζει το πρόγραμμα.

- extended_solids.wbt
- extrusion.wbt
- floating_geometries.wbt
- geometric_primitives.wbt
- high_resolution_indexedfaceset.wbt
- muscle.wbt
- physics_primitives.wbt
- polygons.wbt
- textured_boxes.wbt
- textured_proto_shapes.wbt
- textured_shapes.wbt
- webots_box.wbt

5. Environments: Αυτή η ενότητα παρέχει μερικά από τα πιθανά περιβάλλοντα και αντικείμενα που είναι διαθέσιμα στο Webots. Αυτοί οι χώροι μπορούν να μορφοποιηθούν βάση κάποιων παραμέτρων.

- apartment.wbt
- break_room.wbt
- complete_apartment.wbt
- factory.wbt
- hall.wbt
- kitchen.wbt



Εικόνα 17: Environments (Webots)

6. **Rendering:** Αυτή η ενότητα αποτελείται από τρία παραδείγματα τα οποία εστιάζουν στην αποδοτικότητα των προγραμμάτων. Και αυτά είναι:
 - `animated_skin.wbt`
 - `physically_based_rendering.wbt`
 - `sponza.wbt`
 7. **Community Projects:** αυτό απευθύνεται για τα άτομα τα οποία ενδεχομένως να εργάζονται με κάποια ρομπότ τα οποία όμως το webots δεν υποστηρίζει. Έτσι από αυτή την ενότητα μπορούν να αντλήσουν πληροφορίες και να καθοδηγηθούν αντίστοιχα ώστε να προσθέσουν οι ίδιοι τα ρομπότ τους ή και κάποιον ελεγκτή τους.
- I. Ένα ή και περισσότερους **Ελεγκτές-Controllers:** Ένας Ελεγκτής αποτελεί ένα υπολογιστικό πρόγραμμα το οποίο εξετάζει την λειτουργία ενός ρομπότ που περιλαμβάνει το world file. Επιπλέον, κάθε ρομπότ έχει προκαθορισμένο Ελεγκτή, όμως αρκετά μπορούν να χρησιμοποιούν και άλλους. Η γραφή τους μπορεί να επιτευχθεί από τις υποστηριζόμενες γλώσσες προγραμματισμού του προγράμματος (παραδειγμα matlab, Java).
 - II. Και τα **πρόσθετα-Plugins:** τα πρόσθετα λειτουργούν, όπως και για κάθε άλλο προσομοιωτή για να υποβοηθήσουν το έργο της προσομοίωσης. Μάλιστα, πολλές φορές χρησιμεύουν για παράδειγμα, για τη συλλογή πληροφοριών σχετικά με τα προσομοιωμένα αντικείμενα, όπως θέση, προσανατολισμός, γραμμική ή γωνιακή ταχύτητα. Επιπλέον, μπορούν και να χρησιμοποιηθούν για την προσθήκη ροπών, αρθρώσεων, ή και μοντελοποίηση αεροδυναμικών και υδροδυναμικών μοντέλων για ένα ιπτάμενο ή πλωτό ρομπότ.

4.4.3 Το γραφικό περιβάλλον του προσομοιωτή- GUI:

Το γραφικό περιβάλλον είναι το παράθυρο το οποίο εμφανίζεται με το που ανοίξουμε το πρόγραμμα. Προς το επάνω μέρος αυτού βρίσκεται η **γραμμή εντολών** η οποία φέρει 9 βασικές εντολές:

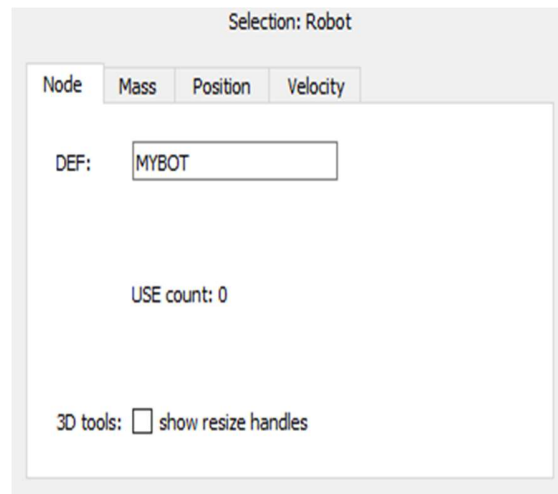
1. **File Menu:** είναι ένας κατάλογος ο οποίος εμπεριέχει αλλές υποεντολές, που λειτουργούν είτε για την αποθήκευση ενός αρχείου τύπου word, είτε την εμφάνιση ενός νέου.

2. **Edit Menu:** ετούτο το μενού παρέχει εργαλεία τα οποία χρησιμευούν κυρίως για την τροποποίηση, την αντιγραφή, την επικόλληση στο Text editor.
3. **View Menu:** και αυτό φέρει έναν μεγάλο αριθμό υποεντολών τα οποία βοηθούν να γίνουν τροποποιήσεις στην απεικόνιση της προσομοίωσης.
4. **Simulation Menu:** περιέχει υποεντολές οι οποίες προκαλούν την πάυση, ενεργοποίηση και επιτάχυνση της προσομοίωσης.
5. **Build Menu:** αυτό το μενού ασχολείται με ιδιότητες του μεταγλωτιστή.
6. **Overlays Menu:** παρέχει εργαλεία τα οποία ελεγχούν την λειτουργία κάμερας, οθόνης και Αποστασιομέτρων-Rangefinder.
7. **Tools Menu:** κατάλογος με άλλες υποεντολές, όπως το 3D View, το Documentation και το Edit Physics Plugin.
8. **Wizards Menu:** χρησιμό εάν θέλουμε να φτιάξουμε νέα προγράμματα, ή νέους ελεγκτές και νέα πρόσθετα-plugins.
9. **Help Menu:** παρέχει γενικές πληροφορίες για τον προσομοιωτή και τον τρόπο λειτουργίας.

Εκτός όμως από την γραμμή εντολών, το γραφικό περιβάλλον παρέχει και το **3D Window**, το οποίο βρίσκεται στο επίκεντρο του παραθύρου και το οποίο ουσιαστικά παρουσιάζει μέσω την οθόνης όλα τα αντικείμενα και το περιβάλλον χώρο της προσομοίωσης. Επιπρόσθετα με την χρήση του ποντικιού παρέχεται η ικανότητα επιλογής αντικείμενων που βρίσκονται στο περιβάλλον, μεταφοράς τους, περιστροφής και απεικόνισής τους από άλλη οπτική γωνία.

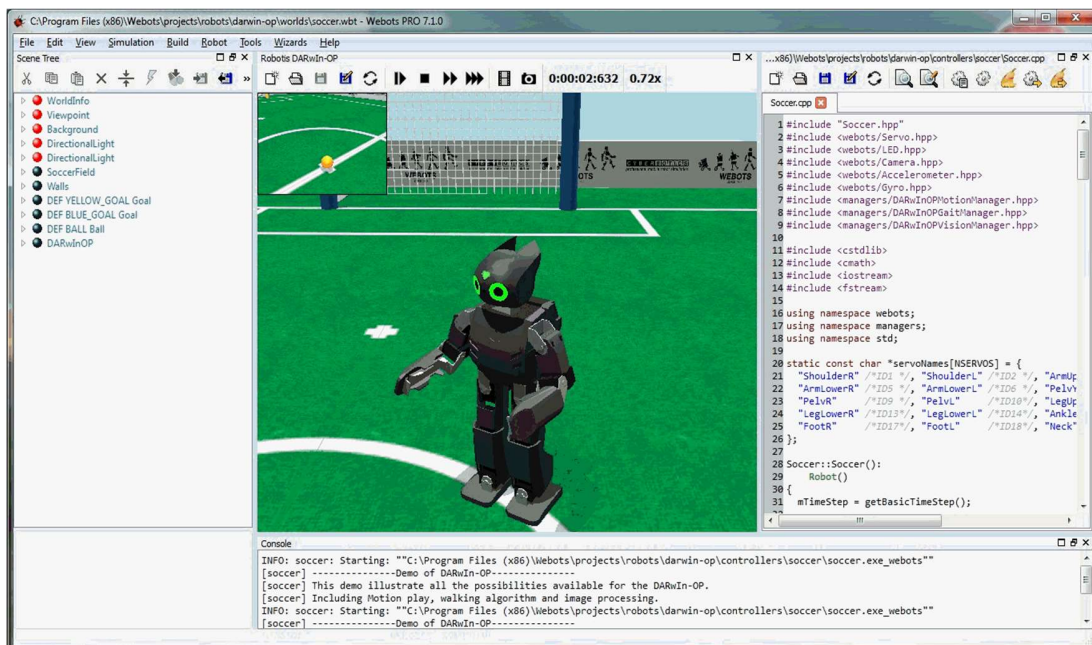
Το γραφικό περιβάλλον περιλαμβάνει και την καρτέλα **Scene tree** όπου βρίσκεται στο αριστερό μέρος αυτής. Ετούτη, καταγράφει όλα τα στοιχεία που συμμετέχουν στην προσομοίωση, ενώ παράλληλα τα αποθηκεύει σε αρχεία τύπου world. Ειδικότερα, είναι δομημένη από κόμβους(nodes), εκ των οποίων ο καθένας περιέχει κάποια πεδία (fields). Τα πεδία, από την άλλη μπορεί να αποτελούνται συμβολοσειρές κειμένου, αριθμητικές τιμές ή να περιέχουν άλλους κόμβους. Βέβαια, τα πεδία αυτά μπορούν να επεξεργαστούν, αφού κάνουμε διπλό κλικ στο αντίστοιχο κόμβο.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 18: Nodes (Webots)

Τέλος, προς τα δεξιά το γραφικό περιβάλλον περιλαμβάνει την **Text editor**, ενώ κάτω από αυτή βρίσκεται η **Console**. Αυτές, προβάλλουν αντίστοιχα τον πηγαίο κώδικα του προγράμματος, καθώς και τα αποτελέσματα που προκύπτουν από το πρόγραμμα όταν υλοποιηθεί ο πηγαίος κώδικας.



Εικόνα 19: Το Γραφικό Περιβάλλον (Webots)

4.4.4 Δομή ενός τυπικού προγράμματος στο webots

Ένα τυπικό πρόγραμμα στο Webots, το οποίο αναπτύχθηκε πρόσφατα, υπάγεται σε κάποιους κανόνες προκειμένου να αποθηκευτεί. Ακριβέστερα, ο φάκελος που θα περιλαμβάνει το δεδομένα αυτού του προγράμματος, θα πρέπει να ακολουθεί μια βασική δομή. Αυτή η δομή προϋποθέτει ο φάκελος να περιλαμβάνει τις εξής κατηγορίες:

- I. The Root Directory of a Project: ετούτη η κατηγορία περιλαμβάνει τουλάχιστον τα αρχεία world, (που όπως έχει αναφερθεί παρέχουν πληροφορίες για τα αντικείμενα που βρίσκονται στο περιβάλλον προσομοίωσης). Εκτός, όμως από αυτά περιλαμβάνει και αρχεία τύπου:
 - Controllers: καταγράφουν ποιοι ελεγκτές συμμετέχουν στην προσομοίωση.
 - Protos: καταγράφουν όλα τα αρχεία τύπου PROTO που χρησιμεύουν στην προσομοίωση. Τα αρχεία PROTO παρέχουν την ικανότητα στους χρήστες να επεκτείνουν το σύνολο των κόμβων προσθέτοντας τους δικούς τους κόμβους. Επομένως, οι χρήστες μπορούν να δημιουργήσουν και να επαναχρησιμοποιήσουν πολύπλοκα αντικείμενα.
 - Plugins: καταγράφουν τα πρόσθετα που είναι διαθέσιμα στην προκειμένη προσομοίωση.
- II. The Project Files: η κατηγορία αυτή περιέχει πληροφορίες πάλι για το περιβάλλον της προσομοίωσης μόνο που αυτές οι πληροφορίες γράφονται από το webots όταν το αρχείο World κλείσει σωστά.
- III. The "controllers" Directory: εδώ βρίσκονται όλοι οι ελεγκτές.

4.4.5 Οι ενεργοποιητές του webots

Το πρόγραμμα υποστηρίζει έναν σημαντικό αριθμό ενεργοποιητών, βάση των οποίων τα ρομπότ είναι σε θέση να δραστηριοποιούνται. Αυτοί, είναι οι παρακάτω:

- Brake (φρένα): κατάλληλο για προσομοιώσεις μηχανικών συστημάτων πέδησης.
- Connector: Προσομοιώνει αντικείμενα που έχουν συνδεθεί με κάποιο τρόπο μεταξύ τους.
- Display(οθόνη): προσομοιώνει την οθόνη ενός ηλεκτρονικού υπολογιστή.
- Emitter(πομποί): Προσομοιώνει ραδιοφωνικούς, σειριακούς ή υπέρυθρους πομπούς που στέλνουν δεδομένα σε άλλα ρομπότ.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- LinearMotor: προσομοιώνει ένα είδος γραμμικού κινητήρα.
- LED
- Muscle: αναφέρεται ένα τεχνητό μυϊκό εξοπλισμό, ο οποίος αναπαριστά την λειτουργία και την μορφή ενός πραγματικού μυς.
- Pen: αναπαριστά ένα μολύβι.
- Propeller: προσομοιώνει μια έλικα πλοίου ή αεροσκάφους.
- RotationalMotor: αναπαριστά έναν περιστροφικό κινητήρα.
- Speaker: αναπαριστά ένα ομιλητή.
- Track
- 3F Gripper: προσομοιώνει μια αρπάγη που αποτελείται από τρία δάχτυλα.

4.4.6 Οι αισθητήρες του webots

Όσον αφορά τους αισθητήρες που παρέχονται από το πακέτο του webots, αυτοί αποτυπώνονται στο αντίστοιχο κατάλογο ως εξής:

- Accelerometer(Επιταχυνσιόμετρο): μελετά την συμπεριφορά ενός αντικειμένου που κινείται με κάποια επιτάχυνση και επηρεάζεται από δυνάμεις βαρύτητας.
- Κάμερες
- Πυξίδα
- Αισθητήρες απόστασης: υπολογίζει την απόσταση μεταξύ δύο αντικειμένων με την βοήθεια υπέρυθρου φωτός, ηχούς σόναρ ή ακτίνας λέιζερ.
- GPS
- Γυροσκόπια: υπολογίζει την γωνιακή ταχύτητα ενός αντικειμένου σε ακτίνια ανά δευτερόλεπτο (roll, pitch και yaw).
- InertialUnit: προσομοιώνει μια μονάδα αδρανειακής μέτρησης. Ακριβέστερα, υπολογίζει και επιστρέφει τις γωνίες του roll, pitch και yaw σε σχέση με το σύστημα συντεταγμένων του περιβάλλοντος χώρου.
- Αισθητήρες τύπου Lidar (laser-scanner): Προσομοιώνει μια κάμερα για την εύρεση βάθους
- Αισθητήρας θέσης: κατάλληλο για τον έλεγχο θέσης κάποιας άρθρωσης, (παράδειγμα ενός ρομπότ).

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Radar
- RangeFinder: Προσομοιώνει μια κάμερα για την εύρεση βάθους, όπως η Lidar .
- Receiver (Δέκτης)
- TouchSensor: χρησιμοποιείται για τη μοντελοποίηση ενός προφυλακτήρα ή ενός αισθητήρα δύναμης και μελετά τύχον συγκρούσεις ή δυνάμεις που ασκούνται αντίστοιχα.

4.4.7 Αντικείμενα προσομοίωσης

Εκτός, όμως από αυτά το πρόγραμμα προσφέρει και ένα ευρύ φάσμα αντικειμένων που μπορούν να λάβουν μέρος στην προσομοίωση. Και αυτά καταγράφονται με συντομία παρακάτω:

I. Αντικείμενα

- Δομή διαμερίσματος
- Μπάλες
- Τουαλέτα
- Υπνοδωμάτιο
- Κτίρια
- Διαφημιστική ταμπέλα
- Καρέκλες
- Υπολογιστές
- Τοίχοι
- Backgrounds
- Αναψυκτικά
- Εργοστάσιο
- Containers
- Μεταφορική ταινία (Conveyors)
- Σωλήνες
- Εργαλεία
- Βαλβίδες
- Δάπεδα
- Φορτία
- Φρούτα
- Κήποι
- Γεωμετρίες
- Κουζίνα
- Ψυγείο
- Φούρνος
- Σκεύη
- Lego
- Φώτα
- Έπιπλα καθιστικού
- Καθρέφτης
- Εμπόδια
- Πίνακες ζωγραφικής
- Πάνελ
- Φυτά
- Δρόμος
- Πετρώματα
- Σχολικά έπιπλα
- Σχήματα

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- Στερεά
- Σκάλες
- Έπιπλα δρόμου
- Πίνακες
- Τηλέφωνο
- Τηλεόραση
- Παιχνίδια
- Δέντρα

4.4.8 Ρομποτικά συστήματα:

Τα σπουδαιότερα ρομποτικά μοντέλα, που υποστηρίζει το λογισμικό και με άνεση μπορεί να προσομοιώσει, είναι τα ακόλουθα:

- 1) AIBO: είναι ένα ρομποτικό σύστημα, σχεδιασμένο από την εταιρία SONY, όπου η μορφή του βασίζεται σε αυτή ενός σκύλου. Ακριβέστερα, το λογισμικό υποστηρίζει το μοντέλο ERS7, το οποίο περιέχει συνολικά 29 αισθητήρες, εκ των οποίων οι 26 είναι LED, ενώ οι 3 ανήκουν στην κατηγορία των υπέρυθρων. Επιπλέον, ενσωματώνει άλλα 20 μηχανικά μέρη, αρμόδια για τον έλεγχο των κινήσεων του ρομπότ.
- 2) ALTINO: το συγκεκριμένο μοντέλο, δημιουργήθηκε από την εταιρία Sacon, και αφορά στην ουσία ένα αυτοκινούμενο ρομποτικό όχημα, διαμορφωμένο από τέσσερις τροχούς, επιταχυνσιόμετρα, και πολλούς αισθητήρες.
- 3) Atlas: ετούτο το ρομπότ, εντάσσεται στα ανθρωπομορφικά μοντέλα, τα οποία προορίζονται για διάφορες διεργασίες, μια από τις οποίες είναι και η διάσωση. Ως μηχανική κατασκευή, εκτιμάται ότι σχεδιάστηκε από το Boston Dynamics. Ωστόσο, εξίσου σημαντικό ρόλο κατείχε και η DAPRA, η οποία ανέλαβε την οικονομική υποστήριξη αυτού.
- 4) BB-8: αυτό απευθύνεται, σε μια αρθρωτή ρομποτική διάταξη, αποτελούμενη από ένα ημισφαιρικό κεφάλι, και ενός σφαιρικού μεγέθους σώμα το οποίο φέρει στο εσωτερικό του ένα τροχοφόρο μηχανικό ρομπότ. Τα στοιχεία αυτά, ενοποιούνται μεταξύ τους

μέσω δύο περιστροφικών αρθρώσεων (pitch-yawn), διαμορφώνοντας κατα αυτόν τον τρόπο την συνολική κατασκευή. Ως κατασκευάστρια εταιρία, αυτού του είδους, εμφανίζεται η Sphero.

- 5) Bioloid: η Bioloid συνιστά προϊόν μοντελοποίησης της εταιρίας Robotics, και χρησιμεύει για την δημιουργία αρθρωτών μοντέλων, όπως αυτό ενός ρομποτικού σκύλου.
- 6) Blimp: το εξής μηχάνημα απευθύνεται σε ένα εναέριο ρομπότ, το οποίο κατασκευάστηκε από το εργαστήριο LIS (Laboratory of Intelligent Systems) στο EPFL.
- 7) Boe-Bot: αυτό το μοντέλο δημιουργήθηκε από την Parallax Inc., και στην ουσία κατατάσσεται στην κατηγορία των αυτοκινούμενων οχημάτων. Διαθέτει τρεις τροχούς, εκ των οποίων οι δύο είναι μηχανοκίνητοι, ενώ ο υπολειπόμενος παθητικός.
- 8) Create: ετούτο πάλι το μηχάνημα αποτελεί προϊόν συναρμολόγησης του iRobot, και χρησιμεύει για τον αυτόματο καθαρισμό ενός χώρου.
- 9) DARwIn-OP: αποτελεί ένα ρομπότ με ανθρωπομορφικά χαρακτηριστικά, το οποίο αναπτύχθηκε και κατασκευάστηκε από την ROBOTIS. Στο σύνολο της κατασκευής του, συνιστάται από 20 μηχανικά υποσυστήματα (άρτια συνδεδεμένα μεταξύ τους), 20 αισθητήρες θέσεως (οι οποίοι τοποθετούνται σε καθένα υποσύστημα), 5 αισθητήρες τύπου LED (από τους οποίους οι δύο είναι της μορφής RGB), 3 επιταχυνσιόμετρα και γυροσκόπια, μια κάμερα και ένα σύστημα μετάδοσης ήχου (speaker). Για τους λόγους αυτούς διαθέτει και έναν αριθμό βαθμών ελευθερίας που ανέρχεται τα 20 (εκ των οποίων δύο τα συναντάμε στο κεφάλι, τρία σε κάθε χέρι και έξι έχουν διαμοιραστεί σε κάθε πόδι)
- 10) Elisa 3: αναφέρεται σε ένα μοντέλο, με δύο τροχοφόρα μαγνητικά τμήματα, ικανά να μεταδώσουν στην συνολική διάταξη την κίνηση σε οριζόντια και κάθετη διεύθυνση. Ειδικότερα, το μοντέλο διαμορφώθηκε από την GCTronic και φέρει ένα σύνολο από αισθητήρες εγγύτητας, αισθητήρες γείωσης, αισθητήρες RGB, επιταχυνσιόμετρα, κινητήρες, πράσινα led, και ραδιοεπικοινωνία.
- 11) e-puck: είναι επίσης μια διάταξη μοντελοποιημένη από την GCTronic και την EPFL. Αλλά σε αντίθεση με την Elisa 3, ενσωματώνει αισθητήρες IR, κάμερα, μικρόφωνα, γυροσκόπια αλλά και Bluetooth για επικοινωνία.

- 12) FireBird 6: αυτό το δημιούργημα του NEX Robotics, αναφέρεται σε ένα δίτροχο αυτοκινούμενο μηχανήμα, το οποίο περιλαμβάνει 8 ultrasonic αισθητήρες, επιταχυνσιόμετρα, πυξίδες, 2 αιθητήρες θέσεως και πολλά άλλα.
- 13) GhostDog: όσο για αυτό το μοντέλο, στηριζόμενοι στις ιδιότητες και την εικονική του όψη, οι ειδικοί το εντάσσουν, στα ρομπότ τύπου σκύλου. Για αυτό και η αναπτυσσόμενη συσκευή φέρει κάμερες, αισθητήρες τύπου touch και τέσσερα πόδια, τα οποία είναι κατάλληλα διαμορφωμένα από αρθρώσεις ενεργού και παθητικού τύπου, επιτυγχάνοντας με άνεση τις απαιτούμενες κινήσεις. Η σύνθεσή του πραγματοποιήθηκε από την EPFL BioRob laboratory.
- 14) Hemisson: σχεδιάστηκε από το K-TEAM, και εμπεριέχει, δύο τροχούς, πληθώρα αισθητήρων, τύπου θέσεως, LED, distance, ανίχνευσης φωτός και αρκετά ακόμη.
- 15) HOAP-2: είναι ένα είδος ανθρωπομορφικού μοντέλου, που ενσωματώνει πολλαπλούς αισθητήρες τύπου touch και θέσεως, αρκετά μηχανικά υποσυστήματα (τα οποία συγκροτούν τα χέρια, τα πόδια, καθώς και το υπόλοιπο σώμα της συσκευής), GPS και Emitter. Ακόμη, εκτιμάται πως παρουσιάζει ‘‘βαθμό ελευθερίας’’ της τάξεως των 25.
- 16) iCub: είναι αποτέλεσμα της εξάισιας δουλειάς που πραγματοποίησε η ΙΙΤ, και κατατάσσεται στα ανθρωπόμορφα είδη. Ειδικότερα έχει ύψος 104cm, βάρος 30 κιλά , διαθέτει 53 βαθμούς ελευθερίας, δύο κάμερες, δυο μικρόφωνα, και αρκετούς αισθητήρες(τύπου TACTILE, FORCE/ torque και άρθρωσης)
- 17) IPR: το webots υποστηρίζει τέσσερα είδη αυτού του τύπου (HD6M180, HD6Ms180, HD6M90,HD6Ms90). Τα οποία και συνιστούν ρομποτικούς βραχίονες, με 6 βαθμούς ελευθερίας , διαμορφωμένους από την Neuronics.
- 18) IRB 4600/40: το ρομπότ ετούτο μοντελοποιήθηκε από την εταιρία ABB, και συνιστά έναν ρομποτικό βραχίονα τύπου 6DOF.
- 19) Khepera I, II, III, IV: τα μοντέλα αυτά είναι δίτροχα οχήματα και αποτελούν προϊόντα παρασκευής της K-Team
- 20) KHR-2HV, KHR-3HV: ανθρωποειδή ρομποτικά συστήματα, με ελευθερία κινήσεων της τάξεως των 17 βαθμών. Ώς κατασκευές, δημιουργήθηκαν από την εταιρία Kondo.
- 21) Koala: αποτελεί ένα αυτοκινούμενο όχημα, με έξι τροχούς και συνολικά 18 αισθητήρες. Ενώ είναι άλλο ένα εμπορικό μηχανήμα της K-team.
- 22) Mantis: είναι ένα εξάποδο ρομπότ, με 36 αισθητήρες.

- 23) Mavic 2 PRO: ένα drone, 5 αισθητήρες τύπου θέσεως- LED, GPS, πυξίδα και πολλά άλλα. Κατασκευαζόμενο από το DJI.
- 24) Microbot: προσομοιάζει ένα αυτοκινούμενο όχημα που φέρει δύο τροχούς. Σχεδιάστηκε από την PICAXE.
- 25) Mindstorms: η κατασκευή του στηρίχθηκε στην ανάπτυξη των Lego's Mindstorms, για αυτό και περιλαμβάνει 5 Lego's Mindstorms, ενώ έχει δύο βαθμούς ελευθερίας.
- 26) Moose: αυτοκινούμενο όχημα, που φέρει συνολικά 8 τροχούς και προορίζεται για εξωτερικής χρήσεως εργασίες.
- 27) Nao: είναι ένα ανθρωπομορφικού τύπου μοντέλο, με 7 αισθητήρες τύπου Touch, sonars, 4 μικρόφωνα και ηχεία, δύο 2d κάμερες και στο σύνολο 25 είδη βαθμών ελευθερίας. Εκτιμάται ως το πρώτο ρομποτικό μηχάνημα, που κατασκεύασε η SoftBank Robotics.
- 28) P-Rob 3: είναι ένας ρομποτικός βραχίονας με 8 βαθμούς ελευθερίας και συνολικά 25 αισθητήρες.
- 29) Pioneer 2, Pioneer 3-AT, Pioneer 3-DX: ετούτα τα μοντέλα, συνιστούν αντικείμενα μελέτης της Adept και προσομοιάζουν οχήματα που φέρουν 3 έως και 4 τροχούς, 16 σοναρ και 16 ultrasonic αισθητήρες.
- 30) Portal: είναι μια συρόμενη πύλη, η οποία μοντελοποιήθηκε από την A4 με σκοπό να διερευνήσει την συμπεριφορά των αυτοκινούμενων οχημάτων σε τέτοιες περιπτώσεις.
- 31) PR2: αυτή η ρομποτική διάταξη συντίθεται από τέσσερις τροχούς (μηχανισμοί που διευκολύνουν την κίνηση), ένα αρθρωτό σώμα και κεφάλι, δύο αρθρωτούς βραχίονες (καθένας εκ των οποίων φέρει και από μία αρπάγη), πολλαπλούς αισθητήρες (τύπου LED, Lidar, position), κάμερες και επιταχυνσιόμετρα. Ως μηχανικό σύστημα, βρίσκεται υπό την διαρκείς επίβλεψη της Clearpath Robotics.
- 32) PUMA: συνιστά έναν ρομποτικό βραχίονα, μοντελοποιημένο από την Unimation, και με βαθμό ελευθερίας που αγγίζει το μέγεθος των 9.
- 33) QRIO: αποτελεί ένα ανθρωποειδές ρομποτικό μηχάνημα, με συνολικά 21 μεγέθοι βαθμού ελευθερίας, ύψος 0.6 μέτρα και βάρος γύρω στα 7.5 κιλά. Κατασκευάστρια εταιρία αυτού, είναι η Sony.
- 34) Robotino 3: δημιουργήθηκε από την εταιρία FESTO και παραπέμπει ουσιαστικά σε ένα τρίτροχο μοντέλο, με αισθητήρες (τύπου bumper, distance, κ.α.) γυροσκόπια,

- RGDB κάμερες και πολλά άλλα. Χρησιμοποιείται κυρίως για εκπαιδευτικούς και ερευνητικούς σκοπούς.
- 35) Robotis OP2, Robotis OP3: συνιστούν και τα δύο ανθρωποειδή ρομποτικά συστήματα, μικρού μεγέθους, σχεδιασμένα από την ROBOTIS. Ακριβέστερα, το Robotis OP2, διαθέτει 20 βαθμούς ελευθερίας, 20 αισθητήρες τύπου θέσεως, 5 LED, (2 RGB), 3 επιταχυνσιόμετρα, και 3 γυροσκόπια, μια κάμερα και ένα speaker. Ενώ σε αντίθεση το Robotis OP3, εμπεριέχει δύο μόνο LED, εκ των οποίων το ένα είναι μπλέ και το άλλο τύπου RGB.
- 36) Salamander: η ανάπτυξη του εκτιμάται ότι έλαβε χώρο στο εργαστήριο EPFL BioRob. Ως ρομποτικό μοντέλο, οι κινήσεις, οι ιδιότητες του και η μορφή του παρομοιάζουν με αυτές μιας Σαύρας. Έχει τέσσερα πόδια, 10 βαθμούς ελευθερίας, δυο αισθητήρες τύπου distance, καθώς και GPS.
- 37) Shrimp: αυτό παραπέμπει σε ένα αυτοκινούμενο ρομπότ το οποίο και κατέχει έξι πόδια, για να επιτυγχάνει με ευκολία την μετακίνηση του. Είναι κατάλληλο για ανώμαλα εδάφη, για να ανεβοκατεβαίνει σκάλες, για να προσπερνά εμπόδια και γενικότερα για οποιαδήποτε είδους επιφάνειας. Το μοντέλο αυτό παράχθηκε από την Bluebotics.
- 38) Sojourner: αποτελεί προϊόν παραγωγής από την NASA, και στην ουσία ο μηχανισμός του, προσομοιάζει ένα κινητό ρομπότ με έξι τροχούς.
- 39) Spot: αυτό το μοντέλο παρομοιάζει με έναν τετράποδο σκύλο, που έχει ύψος 83 cm και μέγιστη ταχύτητα 1.6 m/s. Ενσωματώνει πληθώρα είδη από κάμερες, αισθητήρες και επιταχυνσιόμετρα. Και είναι κατάλληλο για να ανταπεξέλθει σε απαιτητικές εργασίες, όπως η μεταφορά αντικείμενων, η ανέγωση μιας σκάλας, η μετακίνηση σε σκληρές και τραχιές επιφάνειες και πολλά άλλα.
- 40) Surveyor SRV-1: ετούτο το μοντέλο συνιστά ένα μηχανοκίνητο σύστημα, με δύο ατέρμονες μάντες, έναν επεξεργαστή Blackfin, μια ψηφιακή βιντεοκάμερα, αισθητήρες, emitter και δίκτυο WiFi
- 41) TIAGo Base, TIAGo Iron, TIAGo Steel, TIAGo Titanium, TIAGo++ : ανήκουν στην κατηγορία, των δίτροχων μοντέλων και εκτός από το TIAGo Base, όλα τα υπόλοιπα εντάσσονται και στα ανθρωπόμορφα ρομπότ. Φέρουν επιταχυνσιόμετρα,

γυροσκόπια, και αισθητήρες (Touch, Distance, Position, Lidar κ.α), με αποτέλεσμα να είναι αρκετά προσαρμόσιμα με τα περιβάλλοντά τους.

- 42) Thymio II : είναι ένα προϊόν παραγόμενο από το Mobsya, και στην ουσία προσομοιάζει ένα μηχανοκίνητο ρομποτικό εξάρτημα μικρού μεγέθους, με αρκετούς συμπεριλαμβανομένους αισθητήρες Touch, Distance, LED και Position, καθώς και πολλά επιταχυνσιόμετρα. Επιπρόσθετα, το λογισμικό του webots, ενσωματώνει το Aseba studio και το VPL έτσι ώστε ο χρήστης να μπορεί να επέμβει και στον προγραμματισμό του μοντέλου.
- 43) Tinkerbots: ετούτο αποτελεί ένα σετ συστημάτων ικανό να χρησιμοποιηθεί για την διαμόρφωση των επιθυμητών ρομποτικών εξοπλισμών. Έτσι, ο χρήστης μπορεί να αναπαραγεί, να συναρμολογεί και να προγραμματίζει ουσιαστικά πολυάριθμα ρομποτικά είδη, απλά μόνο με αυτό το κιτ.
- 44) Turtlebot3 Burger : είναι μια κινητή ρομποτική μονάδα μικρού μεγέθους με ιδιότητες στηριζόμενες στο ROS. Παρέχει αρκετούς αισθητήρες (τύπου Lidar και Position), μια πυξίδα, ένα επιταχυνσιόμετρο και ένα γυροσκόπιο. Ενώ, η χρήση του προορίζεται κυρίως για κάποιο ερευνητικό ή εκπαιδευτικό έργο. Κατασκευάστηκε από την Robotis.
- 45) UR3e, UR5e and UR10e: συνιστούν ρομποτικούς βραχίονες μικρού μεγέθους, που ενσωματώνουν αισθητήρες τύπου Position, ενώ διαθέτουν βαθμό ελευθερίας που αγγίζει την κλίμακα των 6. Συγκεκριμένα, ο UR3e χρησιμεύει για ελαφριές εργασίες, καθώς φέρει ωφέλιμο φορτίο έως 3 κιλά ,αλλά έχει βάρος 11 κιλά. Σε αντίθεση, το ωφέλιμο φορτίο του UR5e και του UR10e, το οποίο αποτελεί ίσως και την βασική διαφορά μεταξύ αυτών, είναι 5 και 10 κιλά αντίστοιχα.
- 46) Yamor: αυτό είναι ένα μικρό αρθρωτό ρομποτικό εξάρτημα μοντελοποιημένο από το EPFL BioRob laboratory, και συνδέεται με άλλα τέτοιου είδους τμήματα προκειμένου να συνθέσει μια μεγαλύτερη ρομποτική μονάδα. Όσο για την μορφή που ο χρήστης μπορεί να προσδώσει σε αυτά τα μοντέλα, αυτή ποικιλεί.
- 47) youBot: είναι ένας μηχανοκίνητος ρομποτικός βραχίονας, σχεδιασμένος από την KUKA. Ειδικότερα το σώμα του βραχίονα στηρίζεται σε μια αυτοκινούμενη βάση, η οποία διαθέτει τέσσερις τροχούς και μπορεί να κινείται σε οποιαδήποτε κατεύθυνση. Ακόμη φέρει μια αρπάγη τύπου linear, ενώ ο συνολικός βαθμός ελευθερίας εκτεινάζεται ίσως με 5.

4.5 OpenRAVE προσομοιωτής:



Εικόνα 20: Open-Rave Logo

Το πρόγραμμα OpenRAVE (Open Robotics Automation Virtual Environment), αποτελεί ένα λογισμικό προσομοίωσης, το οποίο μοντελοποιήθηκε από Rosen Diankov . Άλλα αποτέλεσε πηγή έμπνευσης του προσομοιωτή RAVE, ενός άλλου λογισμικού που είχε αναπτυχθεί νωρίτερα, γυρω το 1995 από James Kuffner. Αναλυτικότερα, η πρώτη έκδοση το 2006, είχε κυρίως υποστηρικτικό ρόλο για το RAVE και λειτουργούσε για να το συμπληρώσει. Όμως, αργότερα έγινε αυτόνομο και άρχισε να εστιάζει περισσότερο στην βελτίωση και στην ανάπτυξη. Βέβαια, πολλοί εξειδικευμένοι επιστήμονες της ρομποτικής παρακολουθούν το πρόγραμμα και το οδηγούν προς την εξέλιξη. Η προσομοίωση στο OpenRAVE, εφαρμόζεται κυρίως για τη ανάλυση και επίλυση προβλημάτων που σχετίζονται με την κινηματική συμπεριφορά και την γεωμετρικά χαρακτηριστικά που παρουσιάζουν τα μελετούμενα αντικείμενα. Παρόλα αυτά, όμως παρέχει την δυνατότητα στους χρήστες να ελεγχουν, να μοντελοποιούν, και να εξετάζουν πρωτίστως τα μοντέλα τους, όπως και τους αλγορίθμους με ασφάλεια και ύστερα να μεταβιβάσουν την πληροφορία σε πραγματικά ρομπότ. Το πρόγραμμα, υπάγεται στην κατηγορία των λογισμικών “ανοιχτού-κώδικα”, όπου ο πηγαίος κώδικας του προγράμματος χρησιμοποιείται ελευθέρως. Επιπλέον θεωρείται αδειοδοτημένο από την GNU Lesser General, Public License, Creative Commons Attribution 3.0. και Apache License Version 2.0 .

4.5.1 Ρομποτικά μοντέλα

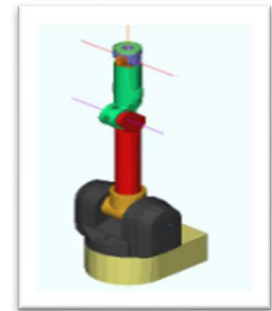
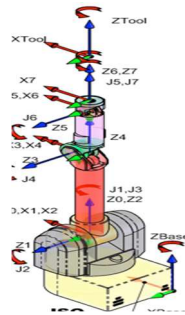
Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

1. barrett-wam Robot: είναι ένας ρομποτικός βραχίονας ο οποίος μοντελοποιήθηκε το 2004 από το Barrett Technology και MIT. Χρησιμοποιείται κυρίως για ερευνητικούς σκοπούς. Και έχει 7 βαθμούς ελευθερίας.

Ύψος από την βάση: 91cm

Μήκος: 72 cm

Βάρος: 27 kg



Εικόνα 21: barrett-wam Robot

2. barrett-wam4 Robot: ετούτο το ρομποτικό μοντέλο παρομοιάζει με το barrett-wam Robot, αλλά στην πραγματικότητα δεν είναι έτσι, καθώς αυτό αποτελείται από τέσσερις βαθμούς ελευθερίας.

Ύψος από την βάση: 90 cm

Ο παρακάτω πίνακας, καταγράφει και τα όρια κάθε άρθρωσης.

Joint	Positive Joint Limit Rad (deg)	Negative Joint Limit Rad (deg)
1	2.6 (150)	-2.6 (-150)
2	2.0 (113)	-2.0 (-113)
3	2.8 (157)	-2.8 (-157)
4	3.1 (180)	-0.9 (-50)
5	1.24 (71)	-4.76 (-273)
6	1.6 (90)	-1.6 (-90)
7	3.0 (172)	-3.0 (-172)

Εικόνα 22: barrett-wam4 Robot (joint limits)

3. darpa-arm Robot:

Ένας ρομποτικός βραχίονας με έξι βαθμούς ελευθερίας, όπως φαίνεται παρακάτω.



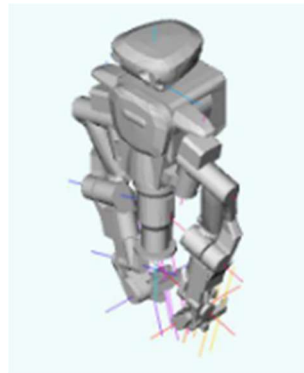
Εικόνα 23: darpa-arm Robot

4. exactdynamics-manusarmleft Robot:



Εικόνα 24: exactdynamics-manusarmleft Robot

5. kawada-hironx Robot:



Εικόνα 25: kawada-hironx Robot

6. kuka-kr30l16 Robot:

Το μοντέλο αυτό, μπορεί να προσεγγίζει αντικείμενα ή σημεία που βρίσκονται μέχρι 3102 mm μακριά. Και μπορεί να αντιμετωπίσει σχετικά ευκολα εργασίες με μέγιστο φορτίο 16 κιλών. Ο βαθμός ελευθερίας αυτή την περίπτωση είναι έξι.

J1	$\pm 130^\circ$
J2	$+35^\circ - 135^\circ$
J3	$+158^\circ - 120^\circ$
J4	$\pm 350^\circ$
J5	$\pm 130^\circ$
J6	$\pm 350^\circ$



Εικόνα 26: kuka-kr30l16 Robot (joint limits)

7. kuka-kr5-r650 Robot:

Ακόμη, ένα άλλο είδος βραχίονα, εξαιρετικά αποδοτικό και λειτουργικό. Κατάλληλο για ελαφριά αντικείμενα βάρους έως και 5 κιλών. Και είναι ικανό να προσεγγίζει αντικείμενα έως και 650 mm. Έχει έξι βαθμούς ελευθερίας, ενώ το βάρος του φτάνει 127.00kg.

J1	375 °/s (6.54 rad/s)	J1	±170°
J2	300 °/s (5.24 rad/s)	J2	+45° - 190°
J3	375 °/s (6.54 rad/s)	J3	+165° - 119°
J4	410 °/s (7.16 rad/s)	J4	±190°
J5	410 °/s (7.16 rad/s)	J5	±120°
J6	660 °/s (11.52 rad/s)	J6	±358°

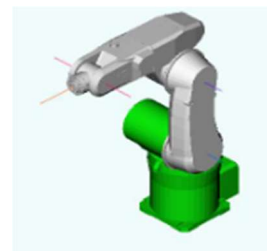


Εικόνα 27: kuka-kr5-r650 Robot (joint limits)

8. kuka-kr5-r850 Robot:

Ρομποτικός βραχίονας είδους kuka, με ίδια χαρακτηριστικά όπως και το kuka-kr5-r650 Robot, μόνο που έχει την ικανότητα να προσεγγίζει αντικείμενα μέχρι και 850 mm. Επιπλέον η γωνιακή ταχύτητα των αρθρώσεων είναι η παρακάτω:

R850
250°/s
250°/s
250°/s
410°/s
410°/s
660°/s

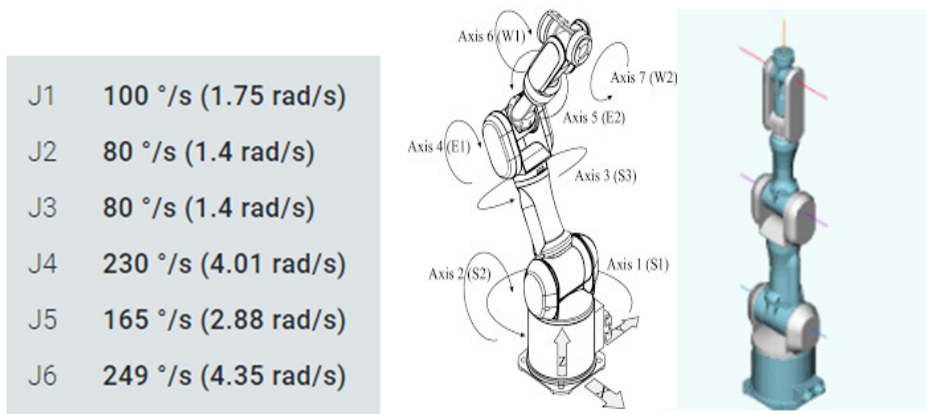


Εικόνα 28: kuka-kr5-r850 Robot (joint limits)

9. mitsubishi-pa10 Robot:

Το Mitsubishi PA10-6C είναι ένας βραχίονας ρομπότ έξι βαθμών ελευθερίας που χρησιμεύει για έρευνα στον τομέα της βιομηχανικής ρομποτικής, εστιάζοντας κυρίως σε σενάρια logistics και παραγωγής. Το ρομπότ είναι ελαφρύ. Το σωματικό βάρος είναι περίπου 40 κιλά και το φορτίο που μπορεί να αντέξει αγγίζει 10 κιλά.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

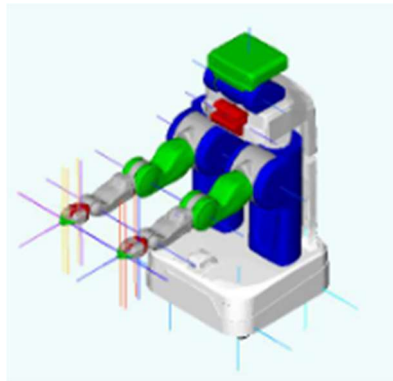


Εικόνα 29: mitsubishi-pa10 Robot

10. pr2-beta-static Robot:

το μοντέλο αυτό είναι ένα από τα ανθρωποειδή ρομπότ που αναπτύχθηκε από τον Willow Garage το 2010 και χρησιμοποιείται για ερευνητικούς σκοπούς κυρίως, αλλά και για να φέρει εις πέρας απλές καθημερινές εργασίες, όπως το καθάρισμα ενός τραπεζιού ή το δίπλωμα των πετσετών και αλλά πολλά.

Ύψος: 165cm
Μήκος: 66.8 cm
Βάρος: 226.8 kg

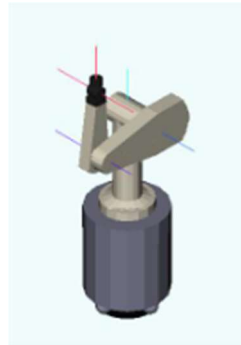


Εικόνα 30: pr2-beta-static Robot

11. schunk-lwa3 Robot:

Η ανοιχτή αρχιτεκτονική- λογισμικού επιτρέπει τη σύνδεση και τη λειτουργία οποιουδήποτε τύπου ενεργού άκρου στον βραχίονα, όπως μια αρπάγη με δύο άκρα ή ένα SCHUNK Hand και άλλα. Έχει έξι βαθμούς ελευθερίας και το φορτίο που μπορεί να αντέξει αγγίζει μέχρι και 5 κιλά.

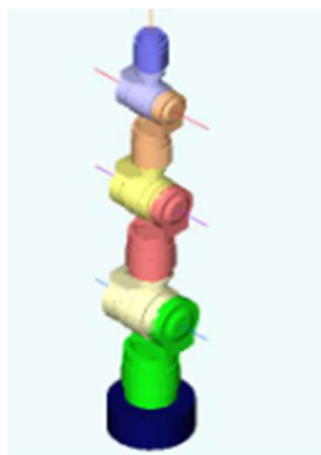
Joint maximums	Degrees
Waist	315
Shoulder	320
Elbow	300
Wrist bend	236
Wrist roll	575
Tool flange	525



Εικόνα 31: schunk-lwa3 Robot

12. unimation-pumaarm Robot:

Το ρομπότ ετούτο είναι ένας βιομηχανικός ρομποτικός βραχίονας που αναπτύχθηκε από τον Victor Scheinman στην εταιρία Unimation. Έχει έξι βαθμούς ελευθερίας και μπορεί να προσεγγίζει εντικείμενα εώς και 400 mm. Επιπλέον, το σωματικό βάρος είναι περίπου 13.2 κιλά και το φορτίο που μπορεί να αντέξει αγγίζει 2.2 κιλά.



Εικόνα 32: unimation-pumaarm Robot

4.5.2 Χαρακτηριστικά προσομοιωτή:

Το λογισμικό του Open RAVE, αποτελείται από ένα περιβάλλον βάση του οποίου είναι σε θέση να αναπαράγει, εφαρμόζει και να δοκιμάζει αλγορίθμους σχεδίασης κίνησης. Αυτό το περιβάλλον, αναπτύχθηκε με καθορισμένο τρόπο, οπότε παρουσιάζει κάποιες χαρακτηριστικές ιδιότητες προκειμένου να μπορεί να ανταπεξέλθει στις προ-αναφερόμενες λειτουργίες, και αυτές είναι οι εξείς:

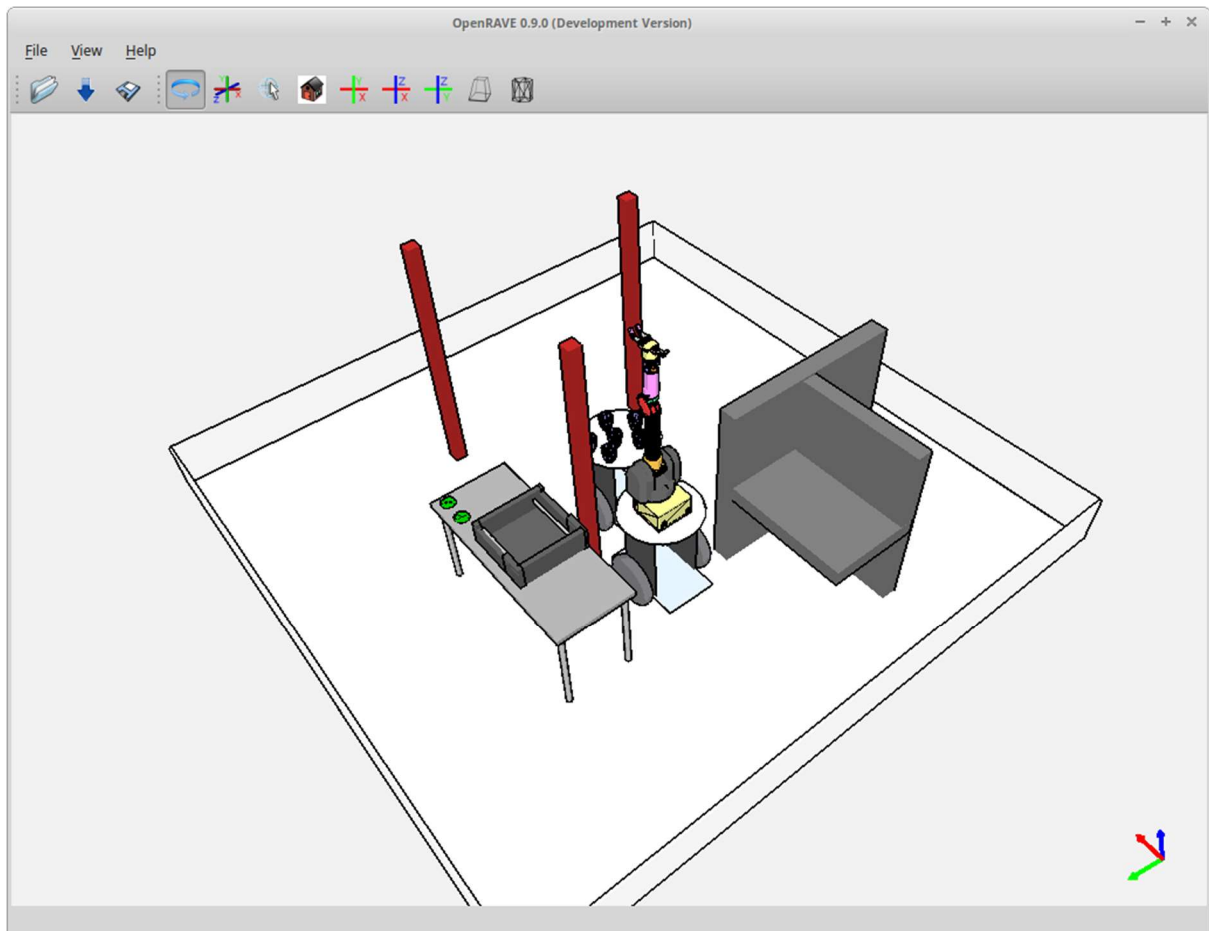
- ✓ Παρέχει ένα αρκετά μικρό εκτελέσιμο πυρήνα, και ετούτο διευκολύνει την χρήση του μέσα σε μικρό-ελεγκτές αλλά και μεγαλύτερα frameworks.
- ✓ Προσφέρει πολυάριθμα εργαλεία εκτέλεσης μέσω της γραμμής εντολών.
- ✓ Εμπεριέχει μια εργαλειομηχανή το IKFAST, ικανή να εργάζεται όπως τα περισσότερα εργαλεία λύσης κινηματικών προβλημάτων:

Περαιτέρω, το εργαλείο αυτό μπορεί και επιλύει αρχικά με αναλυτικό τρόπο πολύπλοκες κινηματικές εξισώσεις και αργότερα παράγει συγκεκριμένα αρχεία προγραμματιστικών γλωσσών. Η επίλυση των παραπάνω προβλημάτων εκτελείται μέχρι και σε 5 μικρό-δευτερόλεπτα σε σύγχρονους επεξεργαστές.

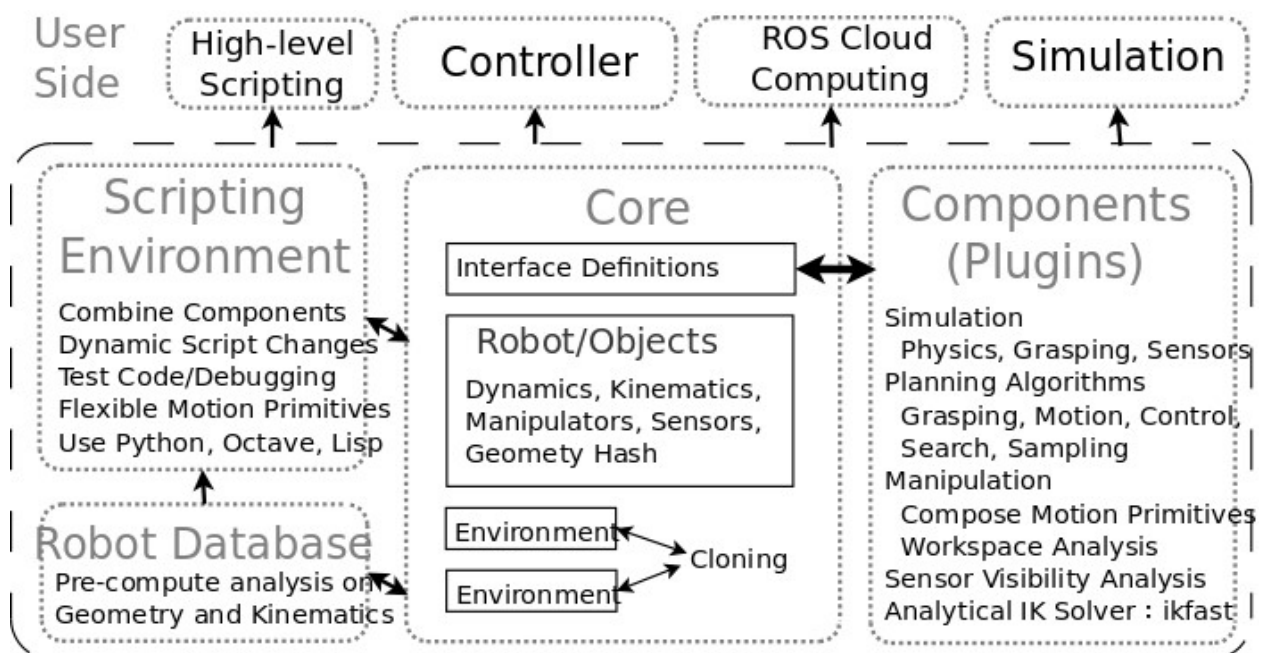
- ✓ Υποστηρίζεται το είδος αρχείων COLLADA (COLLABorative Design Activity), όπου και χρησιμοποιούνται για την αποθήκευση των πληροφοριών που περιγράφουν τα ρομπότ.
- ✓ Ένα από τα πλεονεκτήματα του λογισμικού σε σύγκριση με άλλα πακέτα προγραμματισμού είναι ότι μπορεί να εφαρμόσει αλγόριθμους σε πραγματικά ρομπότ, με πολύ μικρή τροποποίηση.
- ✓ Το κύριο API κωδικοποιείται σε C ++ χρησιμοποιώντας τις βιβλιοθήκες Boost C ++.
- ✓ Η εγκατάσταση του λογισμικού υποστηρίζει από λειτουργικά τύπου Linux, Windows, Mac OSX
- ✓ Ακόμη ένα πλεονέκτημα του προγράμματος, είναι η ικανότητα να παρέχει πολυάριθμα περιβάλλοντα προσομοίωσης για το ίδιο έργο. Με άλλα λόγια, ο χρήστης μπορεί να εργάζεται και να εξετάζει τα ρομπότ παράλληλα σε πολλούς περιβαλλοντικούς χωρούς.

4.5.3 Η αρχιτεκτονική του λογισμικού:

Γραφικό περιβάλλον



Ο τρόπος με το οποίο έχει αναπτυχθεί το λογισμικό και με τον οποίο υλοποιούνται οι λειτουργίες του φαίνεται και πιο κάτω.



Εικόνα 33: Η αρχιτεκτονική του λογισμικού (Open-Rave)

Οι κατηγορίες από τις οποίες αποτελείται το λογισμικό είναι οι εξής:

- **Core Layer:** Ο πυρήνας εξαρτάται από ένα σύνολο βασικών κλάσεων διασύνδεσης που καθορίζουν τον τρόπο με τον οποίο τα πρόσθετα μοιράζονται πληροφορίες. Για αυτό και στον πυρήνα, εμπεριέχονται βιβλιοθήκες - δεδομένα και πληροφορίες σχετικά με τα ρομποτικά μοντέλα, τα αντικείμενα και τα περιβάλλοντα που μπορούν να χρησιμοποιηθούν στην προσομοίωση.
- **Plugins Layer:** τα πρόσθετα, είναι χρήσιμα εργαλεία καθώς μπορούν να συνδεθούν με άλλες βιβλιοθήκες ρομποτικής και με αυτό τον τρόπο να επεκτείνουν τη λειτουργικότητα του προγράμματος ή μπορούν να προσδώσουν υπηρεσίες OpenRAVE σε άλλο σύστημα ρομποτικής. Για παράδειγμα μπορούν να ενδυναμώσουν το πρόγραμμα και την προσομοίωση με μηχανές φυσικής, αισθητήρες και άλλα.
- **Scripting Layer:** η γραφή προγραμμάτων παρέχεται μέσω του Scripting Layer. Αναλυτικότερα, το λογισμικό αναγνωρίζει τις παρακάτω γλώσσες προγραμματισμού:
 1. Octave / Matlab
 2. Και Python

Η Python συνδέεται απευθείας με τον πυρήνα, ενώ το Octave/Matlab στέλνει εντολές μέσω TCP / IP.

- **Robot Database Layer:** η βάση δεδομένων καταγράφει κινηματικές, σχεδόν στατικές, δυναμικές και γεωμετρικές αναλύσεις του ρομπότ και της εργασίας για το οποίο έχει προγραμματιστεί.

4.6 V-REP προσομοιωτής:

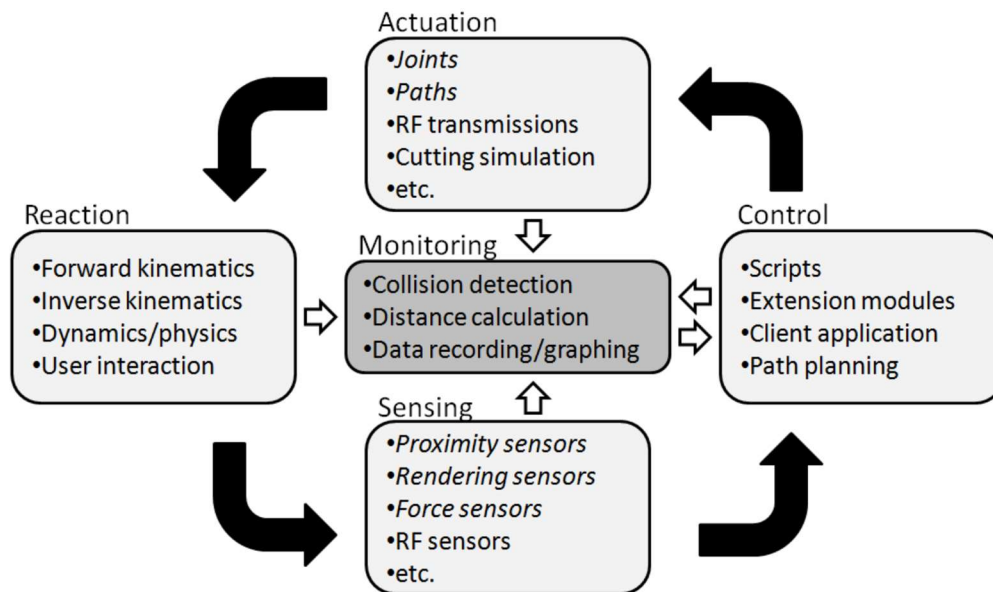


Εικόνα 34: V-rep Logo

Το λογισμικό V-rep (Virtual Robot Experimentation Platform), ουσιαστικά είναι ένα πρόγραμμα ικανό να προσομοιώνει με ακρίβεια, αποδοτικότητα και σε τρισδιάστατα περιβάλλοντα πολυαριθμά ρομποτικά μοντέλα. Αρχικά εμφανίστηκε τον Μάρτιο του 2010 από την Toshiba R&D, ωστόσο οι αλληπάλληλες αναβαθμίσεις γίνονταν κάθε χρόνο από ειδικούς μιας εταιρίας (Coppelia Robotics AG) με έδρα στην Ελβετία, και με την πιο πρόσφατη έκδοση του να περιλαμβάνει το V-REP V3.6.2 το οποίο κυκλοφόρησε το 2019.

Το πρόγραμμα ενσωματώνει αρκετά πακέτα, όπως μηχανές φυσικής, ρομποτικά μηχανήματα, αισθητήρες και πρόσθετα που του παρέχουν επιπλέον λειτουργίες και ιδιότητες, με αποτέλεσμα να υλοποιεί με σχετική ευκολία μεγάλο αριθμό εφαρμογών και ταυτόχρονα επιλύει τυχόν προβλήματα που οι χρήστες αντιμετωπίζουν.

Αναλυτικότερα, το λογισμικό κατά την διάρκεια προσομοίωσης ακολουθεί έναν επαναλαμβανόμενο κύκλο εντολών, όπως και φαίνεται παρακάτω. Επομένως, αρχικά επεξεργάζεται τις εντολές που του έχουν καταχωρηθεί από την εφαρμογή, τις μεταφέρει στην συνέχεια στα κινούμενα μέρη (ενεργοποιητες) και εκείνα αποδίδουν την κίνηση. Έπειτα με την βοήθεια μηχανών φυσικής, κινηματικών εξισώσεων, αισθητήρων, αλλά και άλλων εργαλείων ανιχνεύει τυχόν προβλήματα και λάθη, και τα επιλύει ή τα μεταφέρει με μορφή μηνυμάτων ή γραφικών παραστάσεων, προκειμένου να γίνουν αντιληπτά από το άτομο που χρησιμοποιεί την εφαρμογή.



Εικόνα 35: Η αρχιτεκτονική δομή του V-rep

4.6.1 Χαρακτηριστικά προσομοιωτή:

Αναλυτικότερα μερικές από τις ιδιότητες, λειτουργίες και εργαλειομηχανές που χαρακτηρίζουν το λογισμικό, αναλύονται με μεγαλύτερη ακρίβεια παρακάτω.

- ✓ Υποστηρίζεται από λειτουργικά τύπου Windows, MacOSX και Linux.
- ✓ Υποστηρίζει έξι γλώσσες προγραμματισμού C/C++, Python, Java, Lua, Matlab και Octave.
- ✓ Παρέχει πάνω από 400 λειτουργίες μέσω API, εκ των οποίων μερικές είναι χρήσιμες:
 - Για την εύρεση συγκρούσεων.
 - Για τον υπολογισμό και την εύρεση ελάχιστης απόστασης.
 - Για την επίλυση κινηματικών εξισώσεων.
 - Για την μελέτη δυναμικής συμπεριφοράς αντικειμένων, γραφικών, αιθητήρων, αρθρώσεων και άλλων.
- ✓ Εμπεριέχει τέσσερις (4) εργαλειο-μηχανές φυσικής, ικανές να ανιχνεύσουν λάθη κινηματικής και δυναμικής συμπεριφοράς:
 - I. ODE,
 - II. Bullet,
 - III. Vortex,
 - IV. Newton.

- ✓ Υπολογίζει και επιλύει με ακρίβεια πολύπλοκες κινηματικές εξισώσεις, χάρις των IK και FK που παρέχει.
- ✓ Μπορεί να αναζητήσει τυχόν συγκρούσεις μεταξύ δύο αντικειμένων, χρησιμοποιώντας μια λειτουργία API (`sim.checkCollision`).
- ✓ Παρέχει μια λειτουργία προγραμματισμού κίνησης μέσω ενός plugin που χρησιμοποιεί δεδομένα από την βιβλιοθήκη OMPL:
Οπότε, στην ουσία το πρόγραμμα παρέχει μέσω ενός πρόσθετου κατευθυντήριες εντολές για την δημιουργία ενός κώδικα κίνησης.
- ✓ Προσφέρει μια ευρεία κλίμακα αισθητήρων, όπως αισθητήρες όρασης (Vision sensors) ή αισθητήρες εγγύτητας (proximity sensors).
- ✓ Παρέχει εκτεταμένες λειτουργίες και ιδιότητες από βιβλιοθήκες όπως Reflexxes Motion Library type 2 και από (RRS-1) -Realistic Robot Simulation :
RRS-Interfaces επιτρέπει την ενσωμάτωση οποιουδήποτε λογισμικού ρομπότ, για παράδειγμα ενός ελεγκτή, σε οποιονδήποτε προσομοιωτή.
- ✓ Καταγράφει τα αποτελέσματα και τις πληροφορίες που προκύπτουν σε γραφικές παραστάσεις προκειμένου να παρέχει μια καλύτερη εικόνα στον αναγνώστη.
- ✓ Προσφέρει την δυνατότητα μοντελοποίησης αντικειμένων, κυρίως απλών σχημάτων (ένα τετράγωνο, ένα κύβο ή μια σφαίρα και άλλα), ενώ υποστηρίζει και προγράμματα CAD όπως το AutoCAD, 3D Studio Max και πολλά άλλα.
- ✓ Παρέχει έξι δυνατότητες προγραμματισμού και κωδικοποίησης ενός προγράμματος:
 1. embedded script: ένα πρόγραμμα το οποίο προγραμματίζεται με αυτό τον τρόπο, αποτελείται από έναν κώδικα, το οποίο χρησιμεύει για ένα και μόνο περιβάλλον ή μοντέλο.
 2. add-on / the sandbox script: αντίθετα, με το embedded script το add-on / sandbox script δεν χρησιμεύει για ένα και μόνο περιβάλλον, αλλά υλοποιείται σε όλα τα περιβάλλοντα-μοντέλα με τον ίδιο τρόπο.
 3. plugin: τα πρόσθετα στην ουσία εμπλουτίζουν τις ιδιότητες και λειτουργίες στο λογισμικό.
 4. remote API: αυτή είναι μια διεπαφή που μπορεί να ενσωματώσει στην προσομοίωση μια εξωτερική εφαρμογή, όπως ένα ρομπότ.
 5. ROS node: η ROS ενσωματώσει στην προσομοίωση μια εξωτερική εφαρμογή, όπως ένα ρομπότ.
 6. TCP/IP, ZeroMQ:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

	Embedded script	Add-on / sandbox script	Plugin	Remote API client	ROS / ROS2 node	ZeroMQ node
Control entity is external (i.e. can be located on a robot, different machine, etc.)	No	No	No	Yes	Yes	Yes
Difficulty to implement	Easiest	Easiest	Relatively easy	Easy	Relatively easy	Easy
Supported programming language	Lua	Lua	C/C++	C/C++, Python, Java, Matlab, Octave, Lua	Any ¹	Any
Code execution speed	Relativ. slow ²	Relativ. slow ²	Fast	Depends on programming language	Depends on programming language	Depends on programming language
Communication lag	None	None	None	Yes, reduced	Yes, reduced	Yes, reduced
Control entity is fully contained in a scene or model, and is highly portable	Yes	No	No	No	No	No
Control entity relies on	CoppeliaSim	CoppeliaSim	CoppeliaSim	BlueZero framework, or sockets	ROS / ROS2 framework	Various, e.g. ZeroMQ, TCP/IP
Synchronous operation ³	Yes, inherent. No delays	Yes, inherent. No delays	Yes, inherent. No delays	Yes. Slower due to comm. Lag	Yes. Slower due to comm. Lag	Yes. Slower due to comm. Lag
Asynchronous operation ³	Yes, via threads	Yes, via threads	No (threads available, but API access forbidden)	Yes, default operation mode	Yes, default operation mode	Yes

¹ Depends on ROS / ROS2 bindings

² The execution of API functions is however very fast

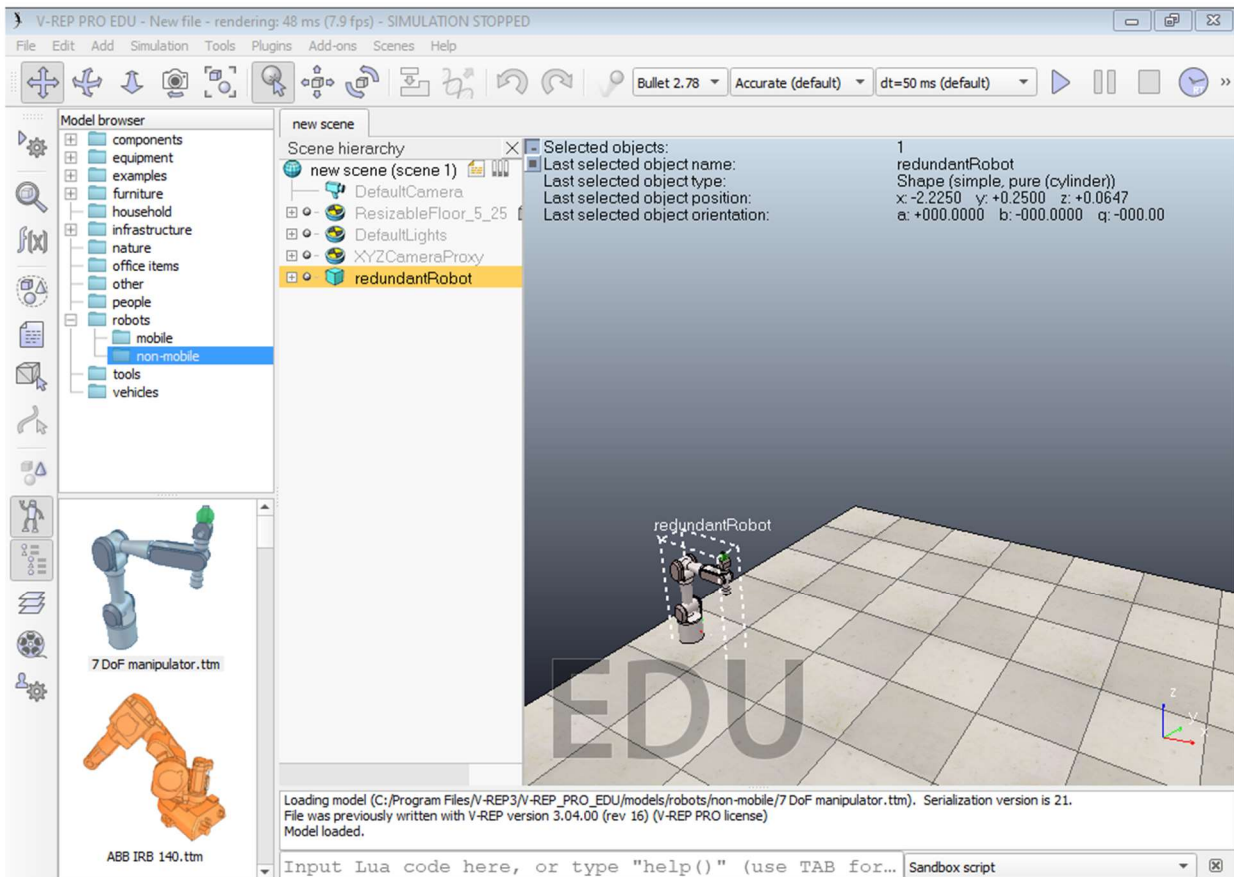
³ *Synchronous* in the sense that each simulation pass runs synchronously with the control entity, i.e. simulation step by step

Εικόνα 36: Χαρακτηριστικά προσομοιωτή V-rep

4.6.2 Το περιβάλλον προσομοίωσης

Το πρόγραμμα, μόλις ενεργοποιηθεί εμφανίζει ένα κεντρικό παράθυρο (User interface), το οποίο στην ουσία αποτελείται από μερικές μικρότερες καρτέλες, η καθεμιά από τις οποίες παρέχει εξειδικευμένες πληροφορίες για την προσομοίωση. Ειδικότερα, αυτές οι καρτέλες ομαδοποιούνται κατά αυτό τον τρόπο:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές



Εικόνα 37: Το Γραφικό Περιβάλλον (V-rep)

- application bar: αυτό το τμήμα, περιγράφει τα στοιχεία του προγράμματος, για παράδειγμα ποια έκδοση χρησιμοποιείται (V-REP PRO EDU), πως ονομάζεται ο φάκελος και ποιος ο χρόνος της προσομοίωσης.



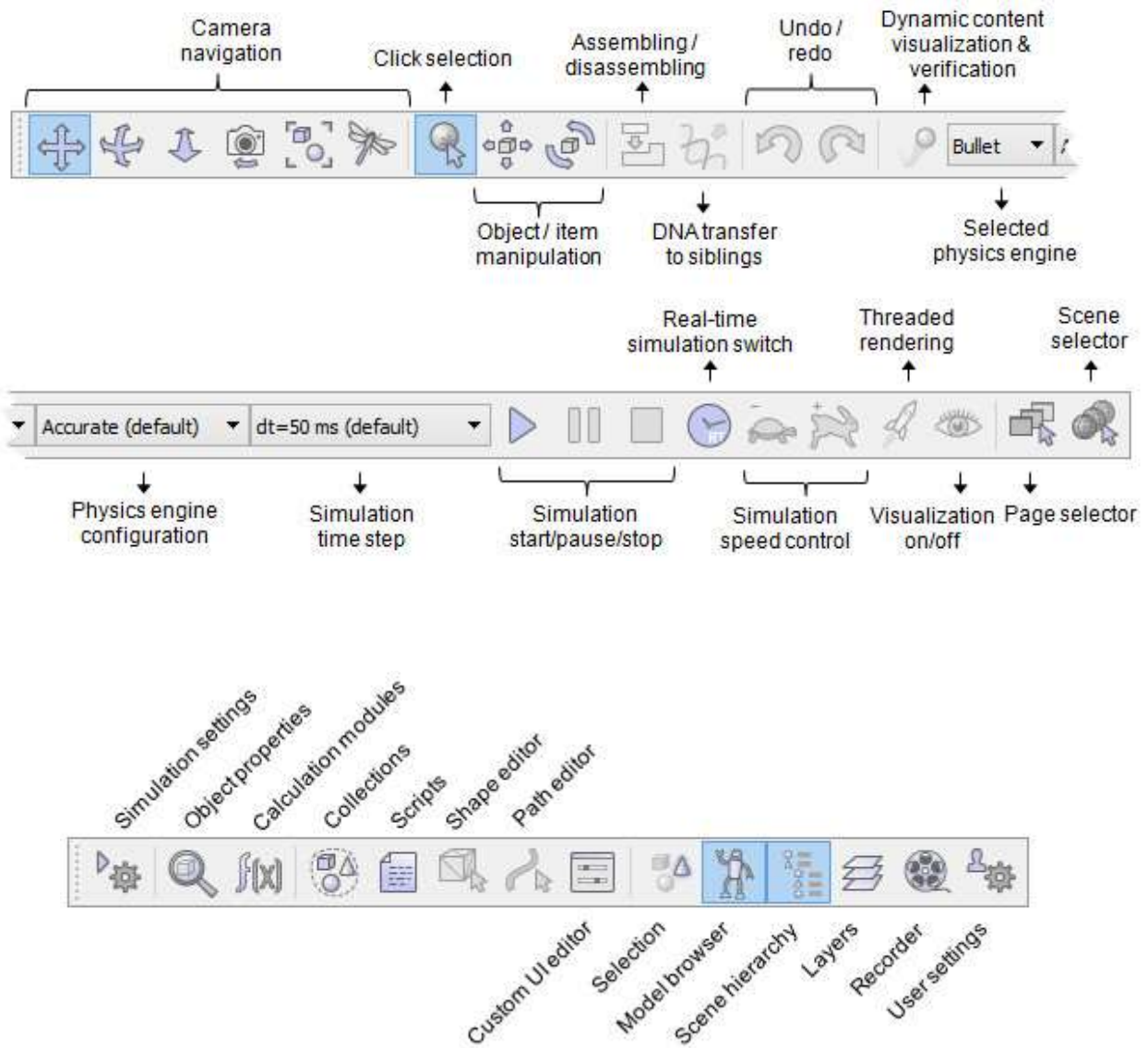
Εικόνα 38: application bar (V-rep)

- menu bar: η γραμμή εντολών, αποτελείται από υπολοεντολές χρήσιμες στην προσομοίωση.



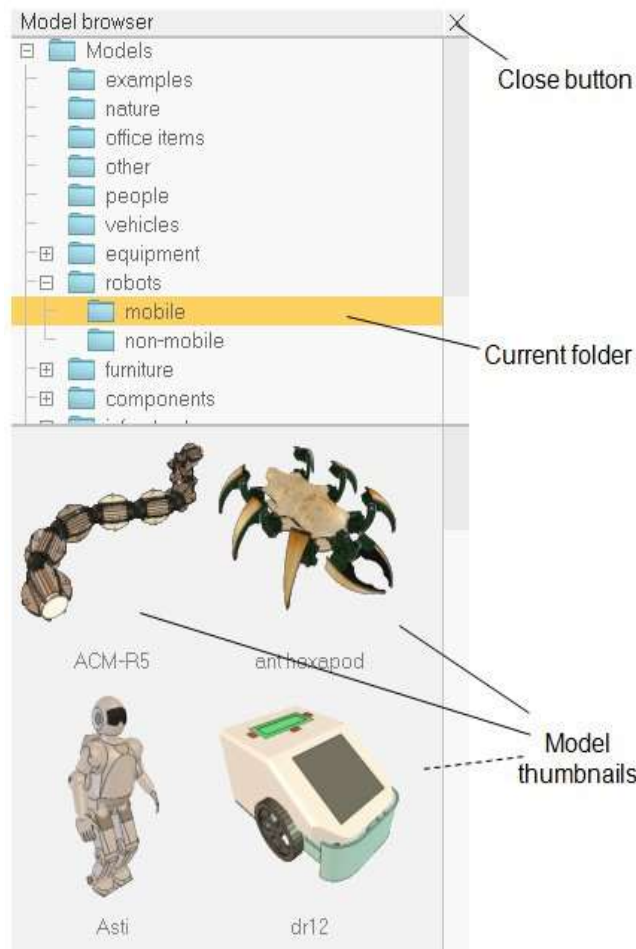
Εικόνα 39: menu bar (V-rep)

- toolbars: αυτή η γραμμή βρίσκεται στο κάτω μέρος την γραμμής εντολών, και εμπεριέχει μικροεργαλεία της γραμμής εντολών τα οποία είναι παρουσιαζόμενα με αυτό τρόπο προκειμένου να διευκολύνουν τους χρήστες.



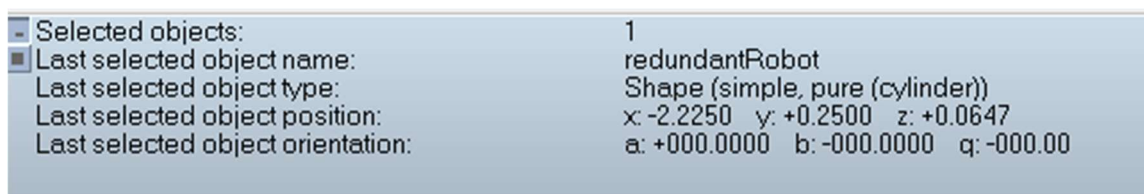
Εικόνα 40: toolbars (V-rep)

- model browser: η καρτέλα αυτή εμπεριέχει τα μοντέλα (ρομπότ, αντικείμενα, επιφάνειες, εργαλεία) που μπορούν να προσομοιωθούν.



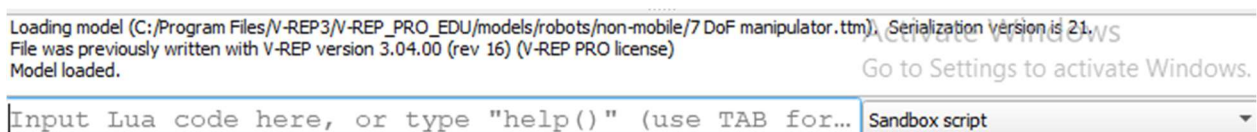
Εικόνα 41: model browser (V-REP)

- information text: βρίσκεται στο πάνω μέρος του περιβάλλοντος προσομοίωσης και παρέχει πληροφορίες σχετικά με τα αντικείμενα που επιλέγονται να συμμετάσχουν στην προσομοίωση, (για παράδειγμα το είδος του αντικειμένου, το όνομα του μοντέλου και την τοποθεσία του στην σκηνή).



Εικόνα 42: information text (V-REP)

- status bar: η γραμμή αυτή βρίσκεται στο κάτω μέρος του περιβάλλοντος προσομοίωσης και λειτουργεί για την καταχώρηση κωδικοποιημένων εντολών.



Εικόνα 43: status bar (V-REP)

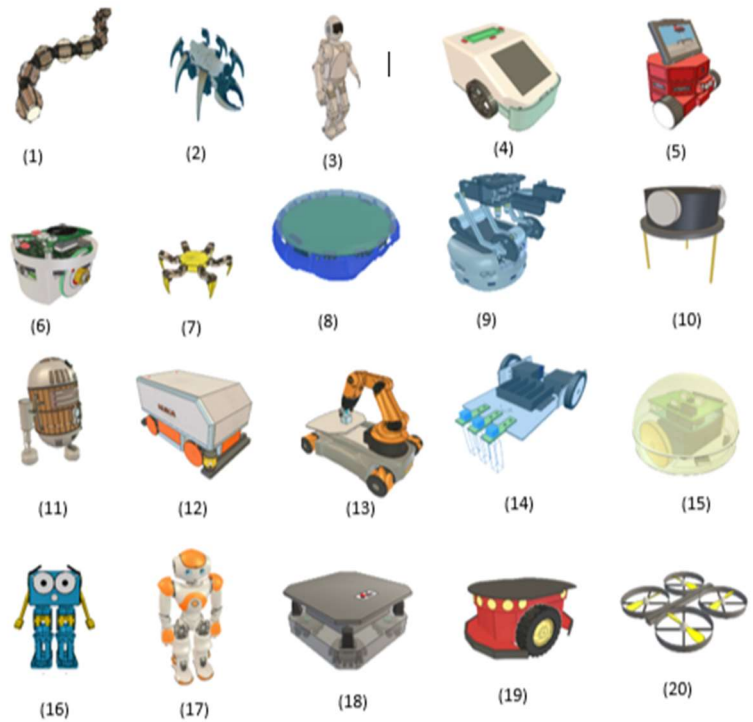
- page: αυτό το κομμάτι αποτελεί το περιβάλλον προσομοίωσης, στο οποίο και αποτυπώνονται σε τρισδιάστατη μορφή όλα τα αντικείμενα που λαμβάνουν χώρα στην προσομοίωση.

4.6.3 Ρομποτικά μοντέλα προσομοιωτή:

Τα ρομποτικά πακέτα που ενσωματώνει το λογισμικό, κατηγοριοποιούνται με αυτό τον τρόπο:

I. Κινούμενα ρομπότ:

- ACM-R5 (1)
- ant hexapod (2)
- Asti
- ball robot
- dr12, dr20
- e-puck ROS
- e-puck
- hexapod
- K-Junior
- khepera3
- Kilobot
- Kilobot_Controller
- kinematic steam punk r2d2
- KUKA Omnirob
- KUKA YouBot
- line follower
- lumibot
- Marty the Robot
- NAO
- Omnidirectional Platform
- pioneer p3dx
- Quadricopter
- Robotnik_Summit_XL140701
- rollerWalker

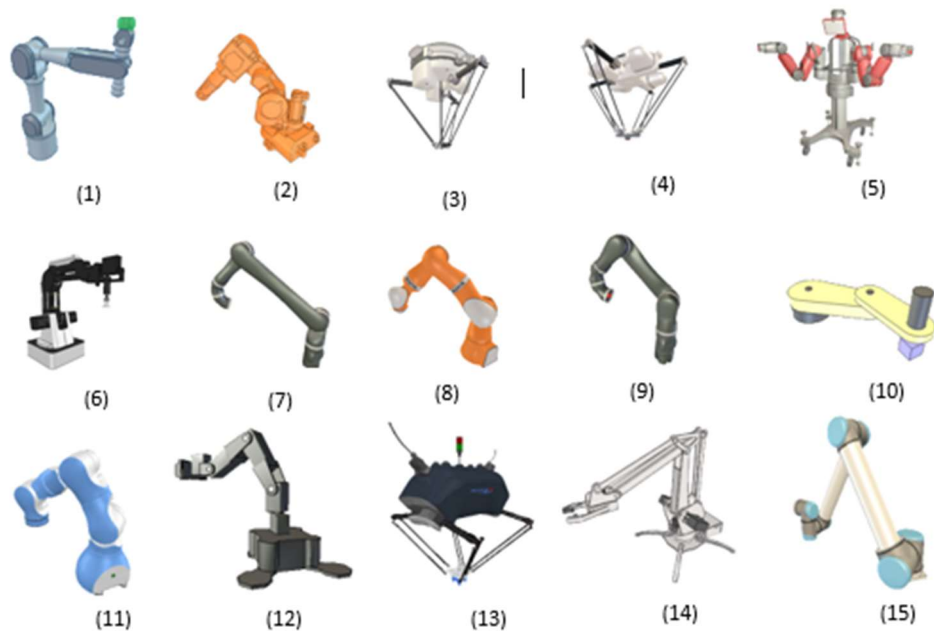


Εικόνα 44: Ρομποτικά μοντέλα προσομοιωτή- Κινούμενα(V-rep)

II. Ακίνητα ρομπότ:

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

- 7 DoF manipulator
- ABB IRB 140, ABB IRB 360, ABB IRB 4600-40-255
- Adept Quattro 650HS
- Baxter
- Dobot Magician
- Jaco arm
- KUKA LBR iiwa 14 R820, KUKA LBR iiwa 7 R800, KUKA LBR4+
- Mico arm
- MTB robot
- P-Arm
- PhantomX Pincher
- ragnar
- Sawyer
- uarm with gripper, uarm
- UR10,UR3,UR5



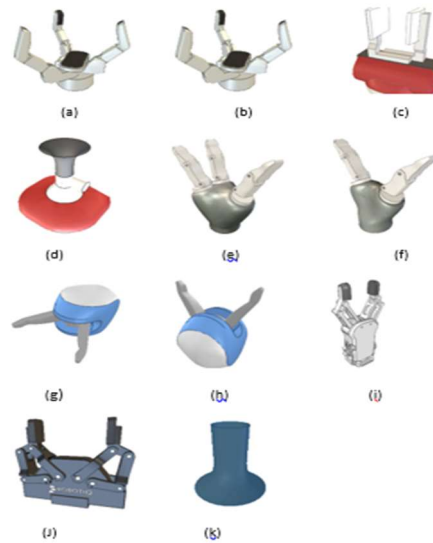
Εικόνα 45: Ρομποτικά μοντέλα προσομοιωτή- Ακίνητα(V-rep)

4.6.4 Αντικείμενα προσομοιωτή:

Το λογισμικό υποστηρίζει ένα τεράστιο σύνολο από αντικείμενα και εργαλεία, τα οποία παρέχουν επιπρόσθετες λειτουργίες και ιδιότητες στην προσομοίωση. Ειδικότερα αυτές είναι:

1. Joints (αρθρώσεις): Υποστηρίζονται τέσσερις τύποι (περιστροφικές αρθρώσεις, πρισματικές αρθρώσεις, βίδες και σφαιρικές αρθρώσεις).
2. Shapes: τα σχήματα αποτελούνται από τριγωνικές όψεις.
3. Gripper: τα ενεργά άκρα- αρπάγες που το λογισμικό παρέχει είναι:

- Barrett Hand (simplified)
- Barrett Hand
- Baxter gripper
- Baxter vacuum cup
- Jaco hand
- Mico hand
- P-Grip-right-angle
- P-Grip-straight
- RG2
- ROBOTIQ 85
- suction pad



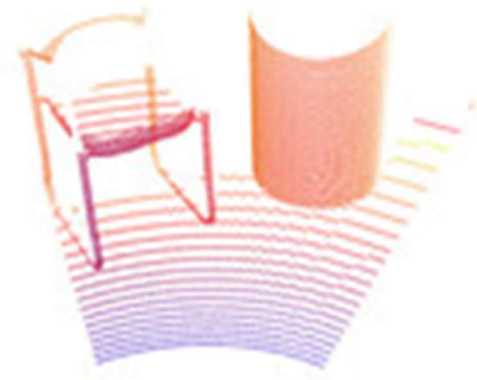
Εικόνα 46: Ρομποτικά Αντικείμενα προσομοιωτή(V-Rep)

4. Graphs: οι γραφικές παραστάσεις καταγράφουν δεδομένα και πληροφορίες που προκύπτουν κατά την διάρκεια της προσομοίωσης προκειμένου να μελετηθεί η συμπεριφορά των αντικειμένων.
5. Dummies: είναι στατικά σημεία που αποθηκεύονται στην μνήμη του λογισμικού, ώστε να μπορούμε εύκολα να επανέλθουμε σε αυτά.
6. Proximity sensors: οι αισθητήρες αυτοί χρησιμεύουν κυρίως για τον εντοπισμό αντικειμένων με γεωμετρικά ακριβή τρόπο. Το λογισμικό εμπεριέχει τέσσερις κατηγορίες αυτού του είδους (πυραμίδας, κυλίνδρου, δίσκου, κώνου και ακτίνων).
7. Vision sensors: ένας αισθητήρας όρασης είναι ένας αισθητήρας τύπου κάμερας.
8. Force sensors: ένας αισθητήρας, ικανός να προσμετρά δυνάμεις και ροπές που εμφανίζονται σε αυτό.
9. Cameras: μια κάμερα αποτυπώνει στην οθόνη του υπολογιστή την εικόνα που λαμβάνει κατά την χρήση της.
10. Lights (φώτα)

11. Paths: αυτή η κατηγορία αντικειμένων, χρησιμεύει προκειμένου να εντοπισθεί μια προκαθορισμένη διαδρομή.
12. Octrees: τα αντικείμενα αυτά προκύπτουν από την σύνθεση μικρότερων δομών (voxels), τα οποία αποτελούνται από οκταδικά δέντρα.
13. Point clouds: ένα Octree το οποίο περιέχει σημεία.



Εικόνα 47: Octrees



Εικόνα 48: Point Cloud

4.6.5 Πλεονεκτήματα-μειονεκτήματα

4.6.5.1 RoboDK

Πλεονεκτήματα

- Είναι ρομποτικό-αγνωστικιστικό, έτσι μπορεί να χρησιμοποιηθεί για τον προγραμματισμό οποιασδήποτε μάρκας ή μοντέλου ρομπότ με το ίδιο interface.
- Είναι απλό στην εκμάθηση και στη χρήση και δεν απαιτεί επανεκπαίδευση κατά τη μετάβαση σε μια νέα μάρκα ρομπότ.
- Εξαιρετικά ευέλικτες δυνατότητες, καθώς επιπλέον λειτουργικότητες μπορούν να επεκταθούν μέσω πρόσθετων, ακόμη και αν δεν αποτελούν μέρος του βασικού προγράμματος.

Μειονεκτήματα

- Απαιτεί ένα επιπλέον κομμάτι λογισμικού σε σύγκριση με άλλα προγράμματα
- Περισσότερα βήματα προγραμματισμού από ό, τι με χειροκίνητη καθοδήγηση
- Απαιτεί έναν υπολογιστή για την εκτέλεση του λογισμικού προγραμματισμού εκτός σύνδεσης.

4.6.5.2 OpenRave

Πλεονεκτήματα

- Ειδικεύεται στη ρομποτική κίνηση
- ομαλότερος και καλύτερος σχεδιασμός κίνησης λόγω της ισχυρής δυνατότητας μοντελοποίησης κινηματικής
- σχετικά ελαφρύ
- Ακριβής και αξιόπιστη απόδοση σχεδιασμού

Μειονεκτήματα

- Πιο περίπλοκες διαδικασίες εγκατάστασης
- Λιγότερο φιλικό προς ROS
- σχετικά λιγότερους διαδικτυακούς πόρους και μικρότερη κοινοτική υποστήριξη

4.6.5.3 V-Rep

Πλεονεκτήματα

- Μια ολοκληρωμένη βιβλιοθήκη μοντέλων
- Η ικανότητα ενός χρήστη να αλληλεπιδρά με τον κόσμο κατά τη διάρκεια της προσομοίωσης
- Και, το πιο σημαντικό, χειρισμός και βελτιστοποίηση κατά τη διάρκεια της προσομοίωσης

Μειονεκτήματα

- Ο πιο πολύπλοκος και ο πιο ‘διψασμένος’ για πόρους προσομοιωτής

4.6.5.4 Gazebo

Πλεονεκτήματα

- Τα προεπιλεγμένα μοντέλα ρομπότ είναι πολύ πιο απλά

Μειονεκτήματα

- Η χρηστικότητα του Gazebo είναι σχετικά φτωχή
- Ενώ μπορεί να εισάγει τρισδιάστατα πλέγματα, δεν υπάρχουν επιλογές επεξεργασίας, καθιστώντας δύσκολη την αλλαγή και βελτιστοποίηση μοντέλων
- Ένα άλλο πρόβλημα είναι το interface που έχει πολλά ζητήματα και δεν ακολουθεί καθιερωμένες συμβάσεις
- δυσκολίες κατά την εγκατάσταση εξαρτήσεων για το Gazebo και για πολλά από τα μοντέλα τρίτων παροχών.

4.6.5.3 Webots

Πλεονεκτήματα

- Οι ελεγκτές μπορούν να γραφτούν σε c, c++, python και MatLab.
- Υπάρχουν πολλοί αισθητήρες.
- Δυνατότητα προσθήκης προσαρμοσμένης φυσικής για προσομοιώσεις όπως ο άνεμος.
- Δυνατότητα προσθήκης OpenStreetMap και εκτέλεσης του προσομοιωτή σε περιβάλλοντα πολύ παρόμοια με την πραγματικότητα.

ΕΝΟΤΗΤΑ 5: Εφαρμογές σε πλατφόρμα ρομποτικής προσομοίωσης

5.1 Εισαγωγή

Η ενότητα αυτή περιέχει τις εφαρμογές που επιτεύχθηκαν σε δύο από τις προεπιλεγμένες πλατφόρμες ρομποτικής προσομοίωσης που είχαν αναλυθεί στο προηγούμενο κεφάλαιο. Ειδικότερα επιλέχθηκαν οι παρακάτω πλατφόρμες:

- V-rep(coppeliastim)
- Και RoboDK.

Καθώς σε αντίθεση με τις υπόλοιπες δεν απαιτούσαν εξειδικευμένα εργαλεία για την χρήση τους, άρα ήταν εύκολη η εγκατάστασή τους στον υπολογιστή, όπως και η μετέπειτα χρήση.

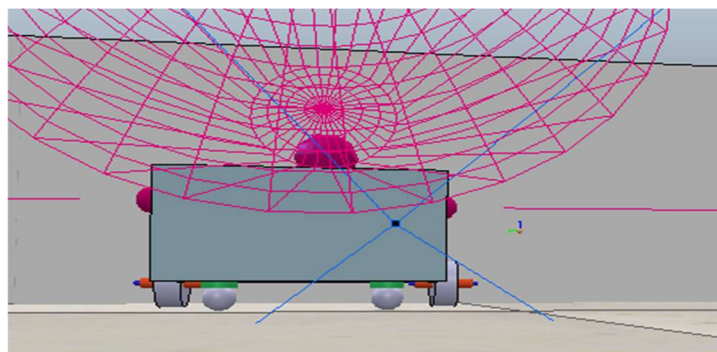
Αυτές περιγράφονται παρακάτω με λεπτομερή τρόπο, όπως από τον τρόπο δημιουργίας, τον κώδικα χρήσης έως και τα αποτελέσματά τους.

5.2 1^η Εφαρμογή (V-rep)

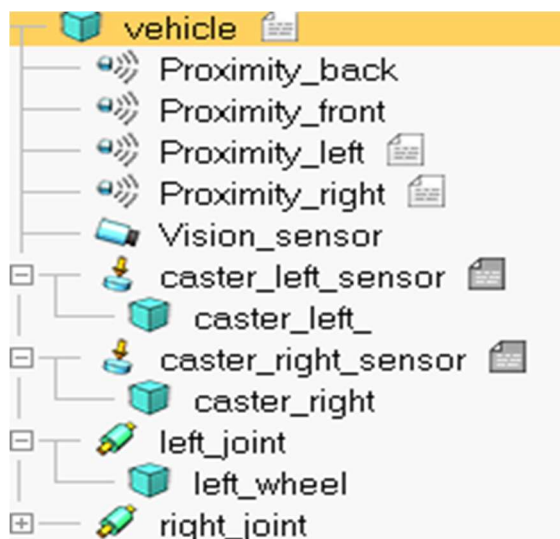
Η εφαρμογή αυτή αφορά ένα αυτοκινούμενο ρομπότ το οποίο μέσω αλληπάλληλων αισθητήρων έχει στόχο να ανιχνεύει τα πολλαπλά εμπόδια που έχουν τοποθετηθεί στο κατάλληλα μοντελοποιημένο περιβάλλον εργασίας του, και στην συνέχεια αφού τα εντοπίζει, θα πρέπει να μπορεί να τα αποφεύγει αλλάζοντας πορεία. Να επισημανθεί ότι για την συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η τελευταία έκδοση του V-rep (Version 4.2.0).

5.2.1 Το ρομπότ

Το ρομποτικό μοντέλο στην προκειμένη περίπτωση κατασκευάστηκε με τον εξής τρόπο:



Εικόνα 49: 1^η Εφαρμογή (V-rep) - το ρομπότ



Εικόνα 50: 1^η Εφαρμογή (V-rep) - scene hierarchy του ρομπότ

Όπως φαίνεται και από το scene hierarchy, το ρομπότ αποτελείται από μια κύρια κυβική επιφάνεια (vehicle), η οποία απαρτίζει τον ρόλο του (parent) και στην οποία είναι συνδεδεμένα τα υπόλοιπα εργαλεία τα οποία έχουν το ρόλο του (child). Συνολικά υπάρχουν δύο ρόλοι (parent και child). Κάθε ένας από τους ρόλους προσδίδει και εξειδικευμένες ιδιότητες στο κάθε αντικείμενο. Ειδικότερα ο ρόλος (parent) δίνει την δυνατότητα στο αντικείμενο να ακολουθείτε από το (child).

Η κυβική αυτή επιφάνεια, έχει μάζα ίση με 5 κιλά και διαστάσεις ($x=0.4, y=0.4, z=0,2$).

Επιπλέον το μοντέλο φέρει και ένα σύνολο από αισθητήρες οι οποίες έχουν τοποθετηθεί και ως προς τις τέσσερις μεριές της κύριας κυβικής επιφάνειας. Οι δύο από αυτές (front και back), είναι κατάλληλα προγραμματισμένες, αρχικά για να ανιχνεύουν τα εμπόδια σε απόσταση μέχρι 1 και 0.5 μέτρα αντίστοιχα, και έπειτα για να αλλάζουν πορεία. Ενώ οι υπόλοιπες δύο, εντοπίζουν τα εμπόδια και καταγράφουν την απόσταση αυτών σε μια κοινή γραφική παράσταση.

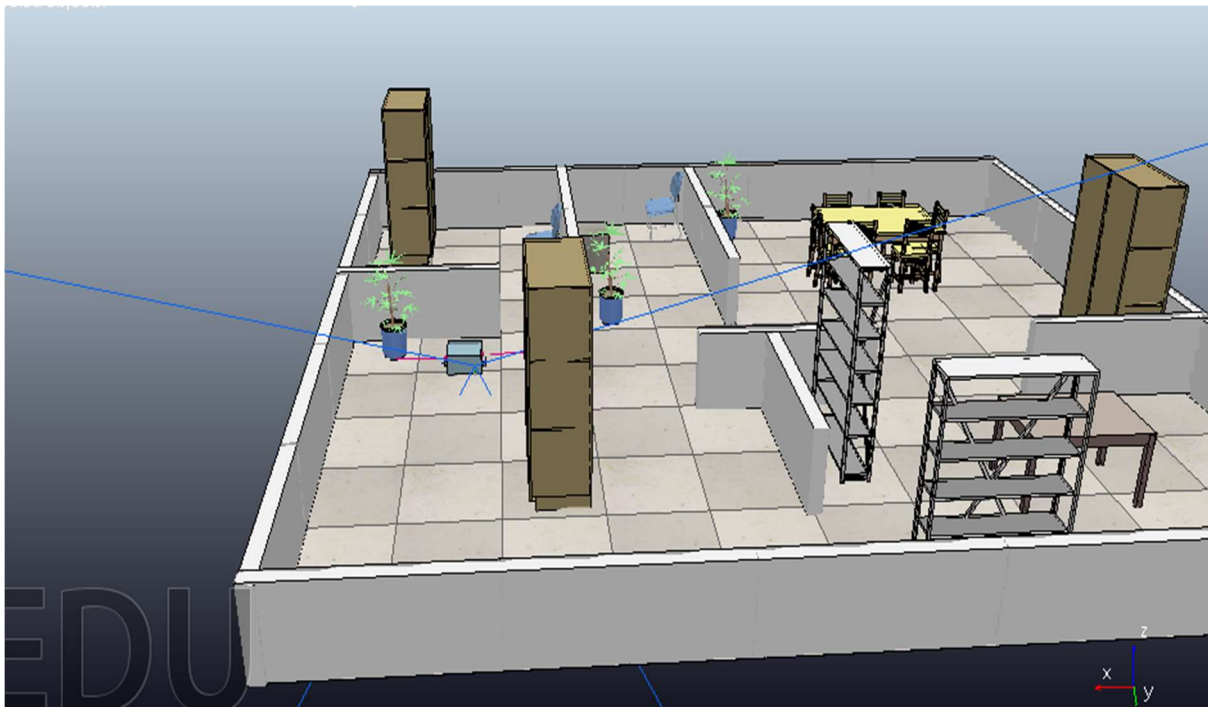
Ακόμη, το ρομπότ φέρει και μια κάμερα η οποία είναι σε θέση να αποδίδει την εικόνα του προσομοιωτή κατά την προσομοίωση. Ενώ ως μηχανισμό μετάδοσης κίνησης, παρέχει δύο τροχούς σε μορφή δίσκων (left and right wheel), οι οποίοι αποδίδουν μηχανική κίνηση μέσω δύο συνδέσμων (left, right joint) τύπου (revolute joint). Τέλος, φέρει και δύο αισθητήρες τύπου δύναμης (caster) οι οποίοι παρέχουν κυρίως στηρικτικό ρόλο στο ρομπότ, αφού υποβοηθούν

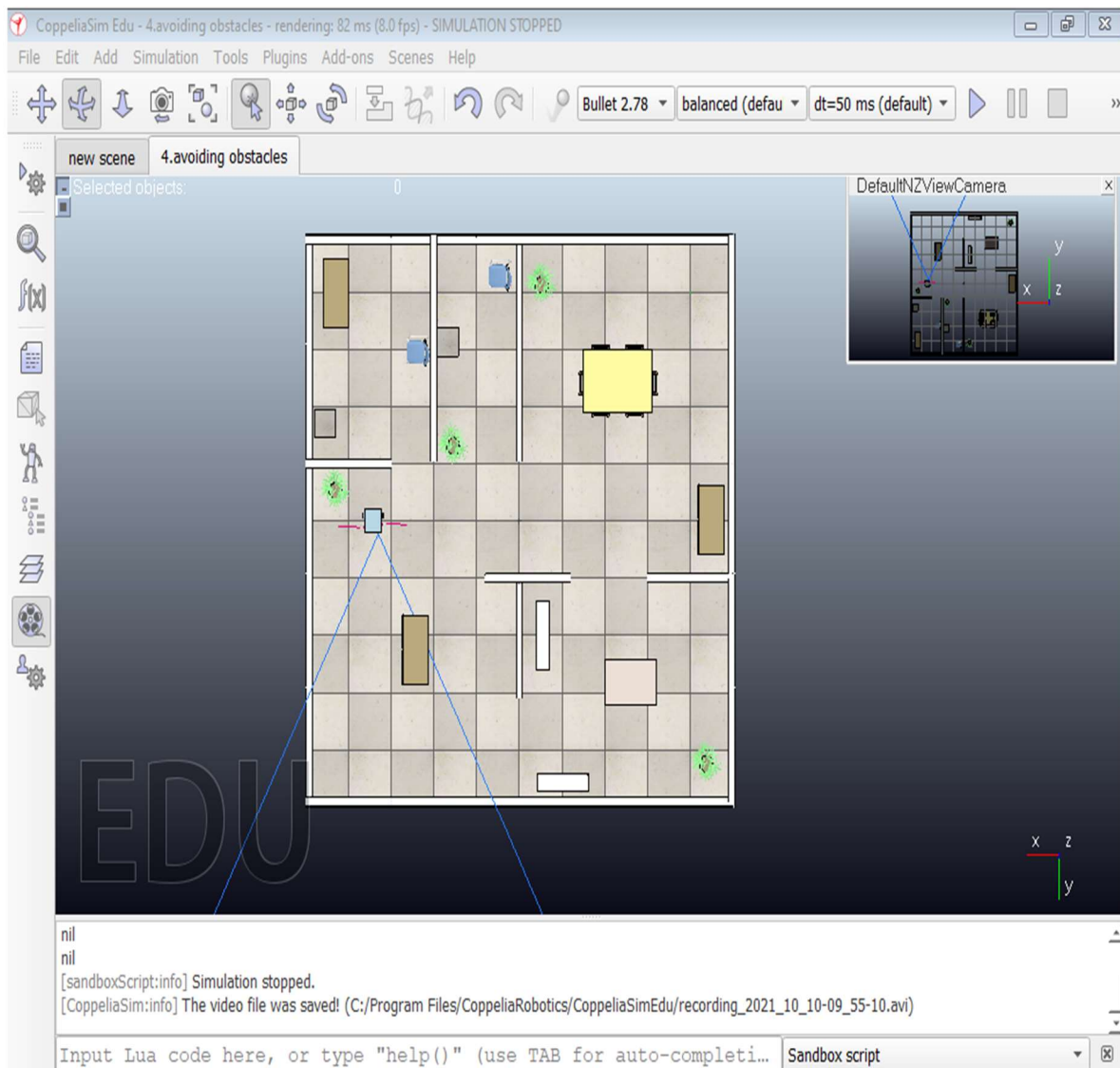
την κατασκευή να μπορέσει να κρατήσει μια ισορροπία κατά την διάρκεια της κίνησης και άρα να μπορέσει να κινηθεί ομαλά.

Τέλος, το μοντέλο συνολικά θεωρείται ως dynamic, detectable και respondable, ενώ χρησιμοποιεί για την επιλυση του προβλήματος την μέθοδο Κίνηση συνεχόμενης διαδρομής, καθώς και τις εξείς μηχανές φυσικής: Bullet physic engine (V2.78).

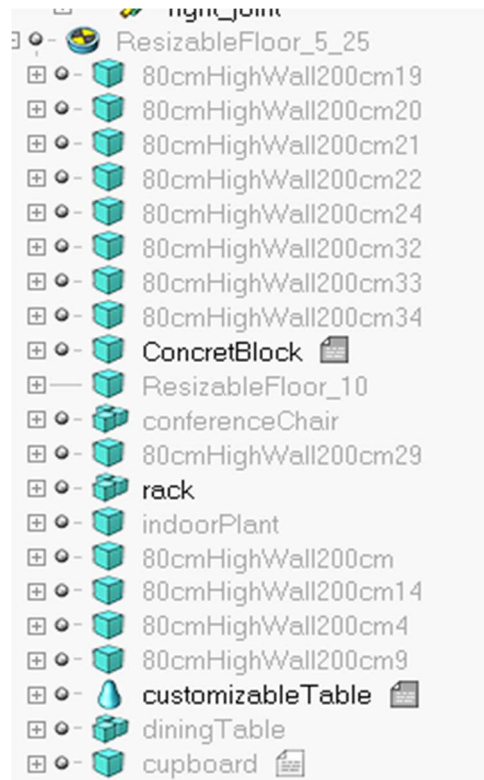
5.2.2 Το περιβάλλον προσομοίωσης

Το περιβάλλον προσομοίωσης αποτελεί ένα διαμέρισμα, το οποίο είναι περιτριγυρισμένος από τοίχους, ενώ στο εσωτερικό του καταγράφονται αρκετά αντικείμενα, όπως καρεκλες, φυτά, ντουλάπες και άλλα τα οποία λειτουργούν ως εμπόδια στο συγκεκριμένο πρόβλημα. Όσο για την ρομποτική κατασκευή αυτή διαθέτει τις εξείς συντεταγμένες και προσανατολισμό αντίστοιχα : $(x= +3.4254, y= -0.000012, z= +0.1775)$ και $(a=0, b=0, q= -90)$.



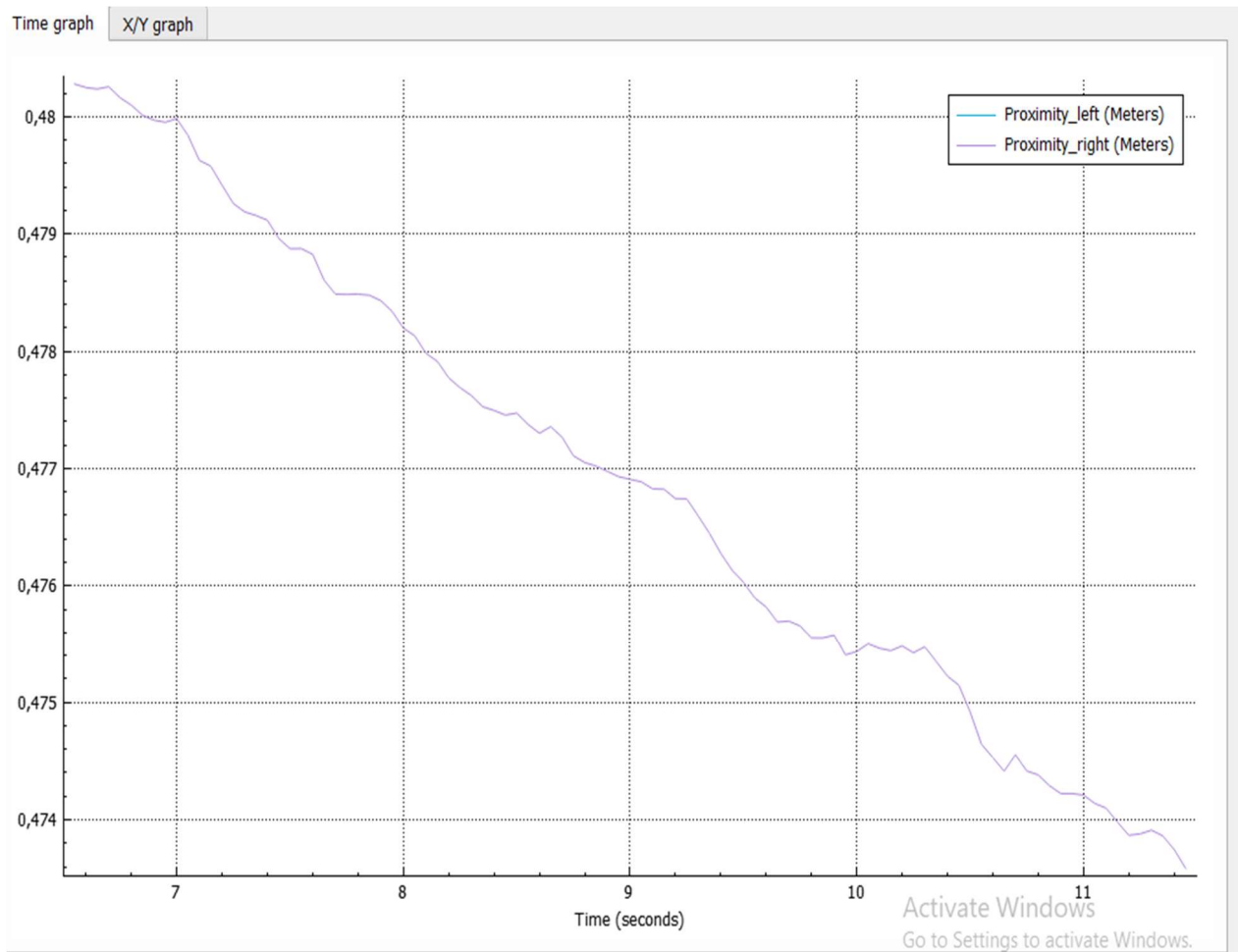


Εικόνα 51: 1η Εφαρμογή (V-rep)-Το περιβάλλον προσομοίωσης



Εικόνα 52: 1η Εφαρμογή (V-rep)- scene hierarchy του περιβάλλοντος προσομοίωσης

Εκτός όμως από αυτά η προσομοίωση εμπεριέχει και την γραφική παράσταση η οποία έχει την εξής μορφή (εικόνα 53) , ενώ καταγράφει δεδομένα για την απόσταση μεταξύ εμποδίων και ρομπότ από τα αριστερά και δεξιά. Συγκεκριμένα, αυτό που κάνει είναι να αναπαριστά την απόσταση που προσμετράνε οι αισθητήρες από τα αριστερα και δεξιά αντίστοιχα, για κάθε λεπτό του χρόνου προσομοίωσης. Έτσι, λοιπόν όταν ένας απο αυτούς του αισθητήρες, πλησιάζει με ένα από τα αντικείμενα του περιβάλλοντος προσομοίωσης σε απόσταση μικρότερη του 0.5 μέτρα, τότε αμέσως λαμβάνουμε στην γραφική παράσταση μια καμπύλη με τις τιμές της απόστασης που ο αισθητήρας υπολογίζει για κάθε λεπτό της προσομοίωσης που περνά.



Εικόνα 53: 1η Εφαρμογή (V-Rep)-γραφική παράσταση

5.2.3 Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή αυτή είναι

Για την κίνηση και ανίχνευση των αντικειμένων του ρομπότ χρησιμοποιήθηκε ένα child script (nonthreaded), και σε γλώσσα Lua


```

1
2 function sysCall_init()
3     -- do some initialization here:
4     vehicle=sim.getObjectHandle(sim.handle_self)
5     left_joint=sim.getObjectHandle("left_joint")
6     right_joint=sim.getObjectHandle("right_joint")
7     Proximity_front=sim.getObjectHandle("Proximity_front")
8     Proximity_back=sim.getObjectHandle("Proximity_back")
9
10
11     minMaxSpeed={200*math.pi/180,400*math.pi/180}
12     backwardTime=-1 -- Tells whether vehicle is in forward or backward mode
13     -- Create the custom UI:
14     xml = '<ui title="" ..sim.getObjectName(vehicle).. ' speed" closeable="false"
15           resizeable="false" activate="false">'..[[
16           <hslider minimum="0" maximum="300" on-change="speedChange_callback" id="1"/>
17           <label text="" style="* {margin-left: 300px;}/>
18           </ui>
19           ]]
20     ui=simUI.create(xml)
21     speed=(minMaxSpeed[1]+minMaxSpeed[2])*0.5
22     simUI.setSliderValue(ui,1,100*(speed-minMaxSpeed[1])/(minMaxSpeed[2]-minMaxSpeed[1]))
23
24
25 end
26

```

Εικόνα 54: 1η Εφαρμογή (V-rep)-κώδικας (μέρος α)

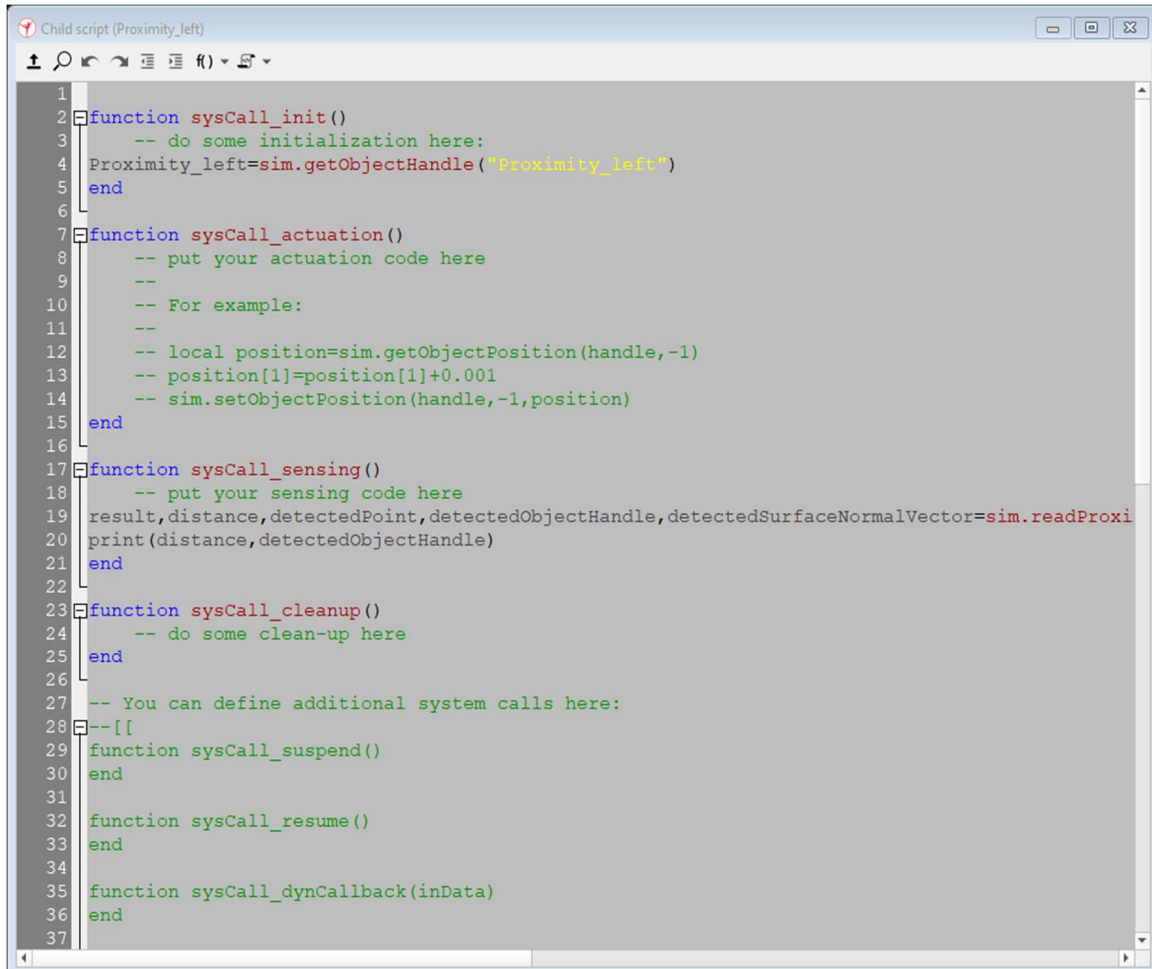
```

Child script (vehicle)
function sysCall_actuation()
  resultA=sim.readProximitySensor(Proximity_front)
  resultB=sim.readProximitySensor(Proximity_back)-- Read the proximity sensor
  -- If we detected something, we set the backward mode:
  if (resultA>0) then backwardTime=sim.getSimulationTime()+4 end
  if (backwardTime<sim.getSimulationTime()) then
    -- When in forward mode, we simply move forward at the desired speed
    sim.setJointTargetVelocity(left_joint,speed)
    sim.setJointTargetVelocity(right_joint,speed)
  else
    -- When in backward mode, we simply backup in a curve at reduced speed
    sim.setJointTargetVelocity(left_joint,-speed/2)
    sim.setJointTargetVelocity(right_joint,-speed/8)
  end
  -- If we detected something, we set the backward mode:
  if (resultB>0) then backwardTime=sim.getSimulationTime()-4 end
  if (backwardTime>sim.getSimulationTime()) then
    -- When in forward mode, we simply move forward at the desired speed
    sim.setJointTargetVelocity(left_joint,-speed/2)
    sim.setJointTargetVelocity(right_joint,-speed/4)
  else
    -- When in backward mode, we simply backup in a curve at reduced speed
    sim.setJointTargetVelocity(left_joint,speed)
    sim.setJointTargetVelocity(right_joint,speed)
  end
end
function sysCall_cleanup()
  -- do some clean-up here
  simUI.destroy(ui)
end

```

Εικόνα 55: 1η Εφαρμογή (V-rep)-κώδικας (μέρος θ)

5.2.3.1 Για τον υπολογισμό ελάχιστης απόστασης και την απεικόνιση της γραφικής παράστασης των αισθητήρων left and right



```
1
2 function sysCall_init()
3     -- do some initialization here:
4     Proximity_left=sim.getObjectHandle("Proximity_left")
5 end
6
7 function sysCall_actuation()
8     -- put your actuation code here
9     --
10    -- For example:
11    --
12    -- local position=sim.getObjectPosition(handle,-1)
13    -- position[1]=position[1]+0.001
14    -- sim.setObjectPosition(handle,-1,position)
15 end
16
17 function sysCall_sensing()
18     -- put your sensing code here
19     result,distance,detectedPoint,detectedObjectHandle,detectedSurfaceNormalVector=sim.readProxi
20     print(distance,detectedObjectHandle)
21 end
22
23 function sysCall_cleanup()
24     -- do some clean-up here
25 end
26
27 -- You can define additional system calls here:
28 --[[
29 function sysCall_suspend()
30 end
31
32 function sysCall_resume()
33 end
34
35 function sysCall_dynCallback(inData)
36 end
37
```

Εικόνα 56: 1η Εφαρμογή (V-rep)-Proximity left code

```

1
2
3 function sysCall_init()
4     -- do some initialization here:
5     Proximity_right=sim.getObjectHandle("Proximity_right")
6
7 end
8
9 function sysCall_actuation()
10
11 end
12
13 function sysCall_sensing()
14     -- put your sensing code here
15 end
16
17 function sysCall_cleanup()
18     -- do some clean-up here
19     result,distance,detectedPoint,detectedObjectHandle,detectedSurfaceNormalVector=sim.readProximi
20     print(distance,detectedObjectHandle)
21 end
22
23 -- You can define additional system calls here:
24 --[[
25 function sysCall_suspend()
26 end
27
28 function sysCall_resume()
29 end
30
31 function sysCall_dynCallback(inData)
32 end
33
34 function sysCall_jointCallback(inData)
35     return outData

```

Εικόνα 57: 1η Εφαρμογή (V-rep)-Proximity right code

5.3 2^η Εφαρμογή (RoboDK)

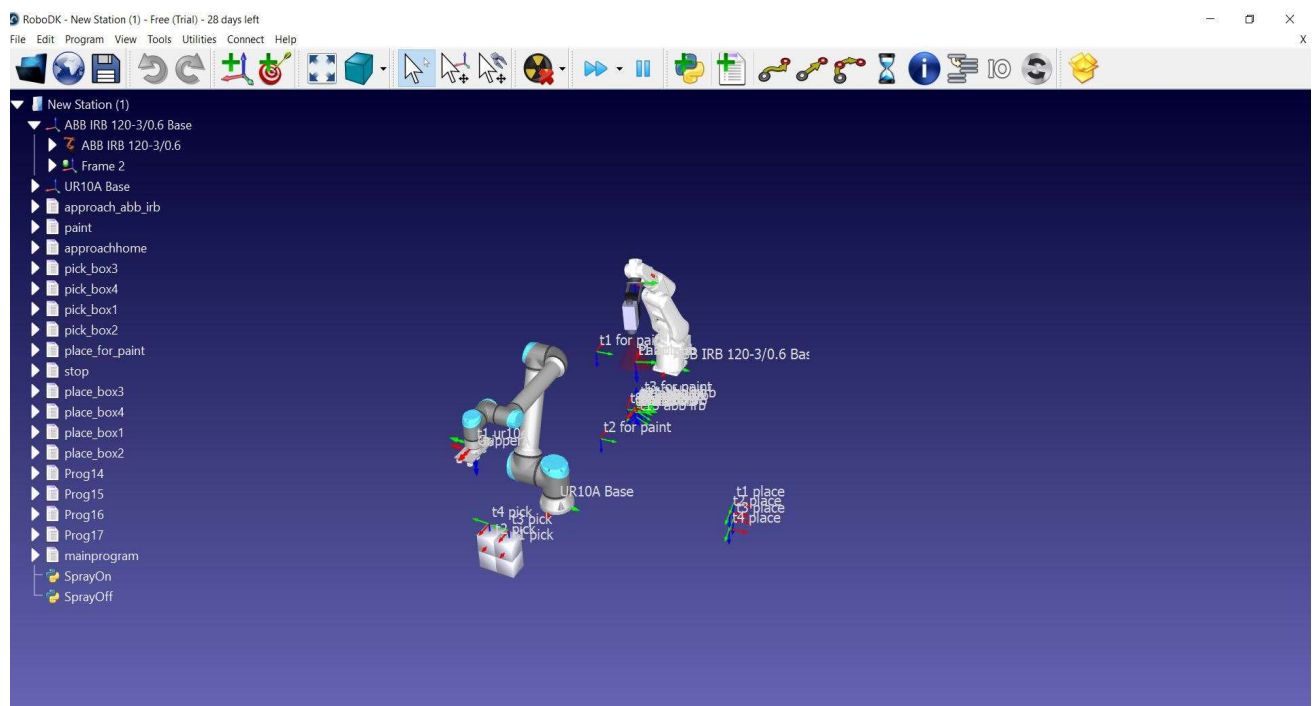
Η εφαρμογή αυτή αφορά ένα ρομπότ τύπου μηχανικού βραχίονα ABB IRB το οποίο μέσω ενός ενεργού άκρου τύπου paint έχει στόχο να βάψει τις τρεις από τις τέσσερις επιφάνειες των κύβων που βρίσκονται στο περιβάλλον προσομοίωσης. Οι κύβοι αυτοί είναι τέσσερις και τοποθετούνται κοντά στο ABB IRB μέσω ενός μηχανικού βραχίονα τύπου UR10e, ο οποίος φέρει στο ενεργό άκρο του μια μηχανική αρπάγη κατάλληλη να επιτελεί εργασίες pick and place. Να επισημανθεί ότι για την συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η τελευταία έκδοση του RoboDK (Version 4.2.0).

5.3.1 Τα ρομπότ

Τα ρομποτικά μοντέλα στην προκειμένη περίπτωση έχουν επιλεγθεί έτοιμα από την βιβλιοθήκη του προσομοιωτή, και σε αυτά έχουν τοποθετηθεί το paint gun και η αρπάγη αντίστοιχα:

5.3.2 Η μέθοδος επίλυσης του προβλήματος

Για την εξής εφαρμογή χρησιμοποιήθηκαν αρκετά Target και ακολουθήθηκε η σημειακή μέθοδος επίλυσης. Ειούτα, λειτουργούν στο λογισμικό ως συντεταγμένες οι οποίες αποθηκεύονται στην μνήμη του υπολογιστή προκειμένου έπειτα να χρησιμοποιηθούν για την δημιουργία επιλεγμένης διαδρομής, πάνω στην οποία θα μετακινηθεί το ρομπότ. Οπότε μέσω πολλαπλών προγραμμάτων που βασίζονται στα target, επιτυγχάνεται η μετακίνηση των ρομπότ στον χώρο. Τέλος χρησιμοποιούνται και δυο κώδικες τύπου python για να τεθεί σε λειτουργία ενεργοποίησης και απενεργοποίησης το ενεργό άκρο για το βάλιμο.



Εικόνα 58: 2η Εφαρμογή (RoboDK)

5.3.3 Ο κώδικας που χρησιμοποιήθηκε για την εφαρμογή αυτή είναι

5.3.3.1 Για την επιτευξη της εργασίας approach abb irb:

Ο κώδικας αυτός μας δείχνει την διαδρομή που ακολουθεί το ρομπότ προκειμένου να φτάσει στην καταλληλη θέση για να ξεκινήσει η βαφή.

```

1  %%%
2  VERSTON:1
3  LANGUAGE:ENGLISH
4  %%%
5  MODULE MOD_approach_abb_irb
6
7  ! -----
8  ! Define customized variables here
9  ! ...
10
11 ! Tool variables:
12 PERS tooldata Paintgun := [TRUE,[[50.000,0.000,450.000],[0.96592583,0.00000000,0.25881905,0.00000000]],1,[0,0,20],[1,0,0,0],[0,0,0.005]];
13
14 ! Reference variables:
15 PERS wobjdata Frame2 := [FALSE, TRUE, "", [[500.000,0.000,0.000],[1.00000000,0.00000000,0.00000000,0.00000000]],[[0,0,0],[1,0,0,0]]];
16
17 PROC Main()
18   ConfJ \On;
19   ConfL \Off;
20   ! Program generated by RoboDK v5.2.5 for ABB IRB 120-3/0.6 on 07/10/2021 13:58:58
21   ! Using nominal kinematics.
22   MoveAbsJ [[-18.860200,28.132600,-70.168000,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
23   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
24
25 ENDPROC
26
27 ENDMODULE
    
```

5.3.3.2 Για την επίτευξη της εργασίας paint:

Ο κώδικας αυτός παρουσιάζει την διαδικασία βαφής που ακολουθεί το abb irb. Άρα, σε αυτό καταγράφονται όλες οι θέσεις και οι συντεταγμένες που προσεγγίζει το ρομπότ όταν εφαρμόζει την εξεί εργασία.

```

1  %%%
2  VERSTON:1
3  LANGUAGE:ENGLISH
4  %%%
5  MODULE MOD_paint
6
7  ! -----
8  ! Define customized variables here
9  ! ...
10
11 ! Tool variables:
12 PERS tooldata Paintgun := [TRUE,[[50.000,0.000,450.000],[0.96592583,0.00000000,0.25881905,0.00000000]],1,[0,0,20],[1,0,0,0],[0,0,0.005]];
13
14 ! Reference variables:
15 PERS wobjdata Frame2 := [FALSE, TRUE, "", [[500.000,0.000,0.000],[1.00000000,0.00000000,0.00000000,0.00000000]],[[0,0,0],[1,0,0,0]]];
16
17 PROC Main()
18   ConfJ \On;
19   ConfL \Off;
20   ! Program generated by RoboDK v5.2.5 for ABB IRB 120-3/0.6 on 07/10/2021 13:59:58
21   ! Using nominal kinematics.
22   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
23   MoveAbsJ [[-19.171300,-2.742030,35.316600,35.734500,37.256400,-57.020300],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
24   MoveAbsJ [[-15.618200,3.479750,28.788800,34.081100,35.790000,-52.593000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
25   MoveAbsJ [[-17.490600,6.030810,26.252700,34.873900,36.674000,-54.815300],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
26   MoveAbsJ [[-14.983400,8.963870,22.731300,33.358100,35.968100,-51.285800],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
27   MoveAbsJ [[-4.345030,58.493300,-60.746800,7.284200,85.714900,-18.347400],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
28   MoveAbsJ [[-9.920000,61.410000,-66.860000,7.260000,88.880000,-17.910000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
29   MoveAbsJ [[-9.910000,61.410000,-63.240000,3.610000,90.230000,72.170000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
30   MoveAbsJ [[-12.307100,37.771100,-23.317700,3.819430,74.133100,68.708000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
31   MoveAbsJ [[40.760000,-25.460000,69.920000,-66.700000,45.930000,165.410000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
32   MoveAbsJ [[59.672900,-16.380300,64.956300,-60.606600,55.806200,177.534000],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
33   MoveAbsJ [[43.738400,25.424700,15.666100,-46.514800,60.306200,45.500200],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
34   MoveAbsJ [[40.907300,32.638600,7.027100,-46.897000,59.713400,42.595200],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
35   MoveAbsJ [[38.493900,40.200100,-2.888840,-46.649800,59.949200,39.114500],[9E+09,9E+09,9E+09,9E+09,9E+09],v500,z1,Paintgun \WObj:=Frame2;
36
37 ENDPROC
38
39 ENDMODULE
    
```

5.3.3.3 Για την επίτευξη της εργασίας approach home:

Εδώ πάλι το ρομπότ abb irb προσεγγίζει την αρχική του θέση. Αυτό, διότι αφού υλοποιήσει την βαφή μετά θα πρέπει να απομακρυνθεί από αυτή για να μην προκαλέσει πιθανή σύγκρουση με τον άλλο βραχίονα, ο οποίος θα πρέπει να πλησιάσει για να παραλάβει το πλέον βαμμένο κυβάκι, και να τοποθετήσει άλλο στην θέση του.

```

1  approachhome.mod x
2  %%%
3  VERSION:1
4  LANGUAGE:ENGLISH
5  %%%
6  MODULE MOD_approachhome
7
8  ! -----
9  ! Define customized variables here
10 ! ...
11
12 ! Tool variables:
13 PERS tooldata Paintgun := [TRUE,[[50.000,0.000,450.000],[0.96592583,0.00000000,0.25881905,0.00000000],[1,[0,0,20],[1,0,0,0],0,0,0.005]];
14
15 ! Reference variables:
16 PERS wobjdata Frame2 := [FALSE, TRUE, "", [[500.000,0.000,0.000],[1.00000000,0.00000000,0.00000000,0.00000000],[[0,0,0],[1,0,0,0]]];
17
18 PROC Main()
19   ConfJ \On;
20   ConfL \Off;
21   ! Program generated by RoboDK v5.2.5 for ABB IRB 120-3/0.6 on 07/10/2021 14:00:36
22   ! Using nominal kinematics.
23   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
24   MoveAbsJ [[-18.860200,28.132600,-70.168000,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
25 ENDPROC
26 ENDMODULE
27

```

5.3.3.4 Για την επίτευξη της εργασίας pick box 3:

Ο κώδικας αυτός χρησιμοποιήθηκε προκειμένου να μπορέσει ο βραχίονας που φέρει την αρπάγη να προσεγγίσει τα κυβάκια ένα - ένα (και ειδικότερα το κυβάκι 3), για να τα μετφέρει έπειτα για βαφή.

```

1  def pick_box3():
2  # Global parameters:
3  global speed_ms = 0.250
4  global speed_rads = 0.750
5  global accel_mss = 1.200
6  global accel_radss = 1.200
7  global blend_radius_m = 0.001
8  global ref_frame = p[0,0,0,0,0,0]
9
10 #-----
11 # Add any default subprograms here
12 # For example, to drive a gripper as a program call:
13 # def Gripper_Open():
14 # ...
15 # end
16 #
17 # Example to drive a spray gun:
18 def SprayOn(value):
19 # use the value as an output:
20 DO_SPRAY = 5
21 if value == 0:
22   set_standard_digital_out(DO_SPRAY, False)
23 else:
24   set_standard_digital_out(DO_SPRAY, True)
25 end
26 end
27
28 # Example to drive an extruder:
29 def Extruder(value):
30 # use the value as an output:
31 if value < 0:
32   # stop extruder
33 else:
34   # start extruder
35 end
36 end
37
38 # Example to move an external axis
39 def MoveAxis(value):
40 # use the value as an output:
41 DO_AXIS_1 = 1
42 DI_AXIS_1 = 1
43 if value <= 0:
44   set_standard_digital_out(DO_AXIS_1, False)
45
46   # Wait for digital input to change state
47   #while (get_standard_digital_in(DI_AXIS_1) != False):
48   # sync()
49   #end
50 else:
51   set_standard_digital_out(DO_AXIS_1, True)
52
53   # Wait for digital input to change state
54   #while (get_standard_digital_in(DI_AXIS_1) != True):
55   # sync()
56   #end
57 end
58 end
59 #-----
60
61
62 # Main program:
63 # Program generated by RoboDK v5.2.5 for UR10e on 07/10/2021 14:01:22
64 # Using nominal kinematics.
65 ref_frame = p[-1.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000]
66 movej([0.058022, -1.215910, -1.907139, -1.577900, 1.558342, -1.523566],accel_radss,speed_rad
67 movej([0.381693, -1.627078, -2.521652, -0.556910, 1.555296, -1.191877],accel_radss,speed_rad
68 # Attach to GripperA
69 # End of main program
70 end
71
72 pick_box3()
73

```

5.3.3.5 Για την επίτευξη της εργασίας place for paint :

Εδώ ο κώδικας περιγράφει την διαδρομή που κάνει το ρομπότ με την αρπάγη για να τα μεταφέρει στο άκρο με το paint gun. Βέβαια, όπως προηγουμένως ο κώδικας αφορά το κυβάκι νούμερο 3.

```

1 def place_for_paint():
2     # Global parameters:
3     global speed_ms = 0.250
4     global speed_rads = 0.750
5     global accel_mss = 1.200
6     global accel_radss = 1.200
7     global blend_radius_m = 0.001
8     global ref_frame = p[0,0,0,0,0,0]
9
10    #-----
11    # Add any default subprograms here
12    # For example, to drive a gripper as a program call:
13    # def Gripper_Open():
14    # ...
15    # end
16    #
17    # Example to drive a spray gun:
18    def SprayOn(value):
19        # use the value as an output:
20        DO_SPRAY = 5
21        if value == 0:
22            set_standard_digital_out(DO_SPRAY, False)
23        else:
24            set_standard_digital_out(DO_SPRAY, True)
25        end
26    end
27
28    # Example to drive an extruder:
29    def Extruder(value):
30        # use the value as an output:
31        if value < 0:
32            # stop extruder
33        else:
34            # start extruder
35        end
36    end
37
38    # Example to move an external axis
39    def MoveAxis(value):
40        # use the value as an output:
41        DO_AXIS_1 = 1
42        DI_AXIS_1 = 1
43        if value <= 0:
44            set_standard_digital_out(DO_AXIS_1, False)
45
46    # Wait for digital input to change state
47    #while (get_standard_digital_in(DI_AXIS_1) != False):
48    # sync()
49    #end
50    else:
51        set_standard_digital_out(DO_AXIS_1, True)
52
53    # Wait for digital input to change state
54    #while (get_standard_digital_in(DI_AXIS_1) != True):
55    # sync()
56    #end
57    end
58 end
59 #-----
60
61
62 # Main program:
63 # Program generated by RoboDK v5.2.5 for UR10e on 07/10/2021 14:03:34
64 # Using nominal kinematics.
65 ref_frame = p[-1.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000]
66 movej([-2.787116, -1.215796, -1.907121, -1.577778, 1.558230, -1.322785], accel_radss, speed_ra
67 movej([-2.801847, -1.950632, -2.513379, -0.236496, 1.558403, -1.337519], accel_radss, speed_ra
68 movej([-3.008599, -2.362844, -1.574205, -0.761809, 1.561141, -1.340586], accel_radss, speed_ra
69 # Detach from GripperA
70 # End of main program
71 end
72
73 place_for_paint()
74

```


5.3.3.6 Για την επίτευξη της εργασίας stop

Αφού τοποθετηθεί για βαφή το κυβάκι, ο βραχιόνας που το μετέφερε, παύει να λειτουργεί για μικρο χρονικό διάστημα και προσεγγίζει μια απομακρυσμένη θέση μέχρι να ολοκληρωθεί η βαφή του εντικειμένου.

```

1 def stop():
2   # Global parameters:
3   global speed_ms = 0.250
4   global speed_rads = 0.750
5   global accel_mss = 1.200
6   global accel_radss = 1.200
7   global blend_radius_m = 0.001
8   global ref_frame = p[0,0,0,0,0,0]
9
10  #-----
11  # Add any default subprograms here
12  # For example, to drive a gripper as a program call:
13  # def Gripper_Open():
14  # ...
15  # end
16  #
17  # Example to drive a spray gun:
18  def SprayOn(value):
19  # use the value as an output:
20  DO_SPRAY = 5
21  if value == 0:
22    set_standard_digital_out(DO_SPRAY, False)
23  else:
24    set_standard_digital_out(DO_SPRAY, True)
25  end
26  end
27
28  # Example to drive an extruder:
29  def Extruder(value):
30  # use the value as an output:
31  if value < 0:
32    # stop extruder
33  else:
34    # start extruder
35  end
36  end
37
38  # Example to move an external axis
39  def MoveAxis(value):
40  # use the value as an output:
41  DO_AXIS_1 = 1
42  DI_AXIS_1 = 1
43  if value <= 0:
44    set_standard_digital_out(DO_AXIS_1, False)
45
46    # Wait for digital input to change state
47    #while (get_standard_digital_in(DI_AXIS_1) != False):
48    # sync()
49    #end
50  else:
51    set_standard_digital_out(DO_AXIS_1, True)
52
53    # Wait for digital input to change state
54    #while (get_standard_digital_in(DI_AXIS_1) != True):
55    # sync()
56    #end
57  end
58  end
59  #-----
60
61
62  # Main program:
63  # Program generated by RoboDK v5.2.5 for UR10e on 07/10/2021 14:07:21
64  # Using nominal kinematics.
65  ref_frame = p[-1.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000]
66  movej([-3.008599, -2.362844, -1.574205, -0.761009, 1.561141, -1.340586], accel_radss, speed_ra
67  movej([-2.001847, -1.950632, -2.513379, -0.226496, 1.558403, -1.337519], accel_radss, speed_ra
68  movej([-2.787116, -1.215796, -1.907121, -1.577778, 1.558230, -1.322785], accel_radss, speed_ra
69  # End of main program
70  end
71
72  stop()
73

```

5.3.3.7 Για την επίτευξη της εργασίας place box 3:

Ο κώδικας στην προκειμένη περίπτωση περιλαμβάνει στοιχεία για την την διαδρομή που ακολουθεί ο βραχιόνας που φέρει την αρπάγη, προκειμένου να απομακρύνει τα βαμμένα κυβάρια και να τα τοποθετήσει σε ένα προκαθορισμένο σημείο.

```

1 def place_box3():
2   # Global parameters:
3   global speed_ms = 0.250
4   global speed_rads = 0.750
5   global accel_mss = 1.200
6   global accel_radss = 1.200
7   global blend_radius_m = 0.001
8   global ref_frame = p[0,0,0,0,0,0]
9
10  #-----
11  # Add any default subprograms here
12  # For example, to drive a gripper as a program call:
13  # def Gripper_Open():
14  #   ...
15  # end
16  #
17  # Example to drive a spray gun:
18  def SprayOn(value):
19  # use the value as an output:
20  DO_SPRAY = 5
21  if value == 0:
22    set_standard_digital_out(DO_SPRAY, False)
23  else:
24    set_standard_digital_out(DO_SPRAY, True)
25  end
26 end
27
28 # Example to drive an extruder:
29 def Extruder(value):
30 # use the value as an output:
31 if value < 0:
32 # stop extruder
33 else:
34 # start extruder
35 end
36 end
37
38 # Example to move an external axis
39 def MoveAxis(value):
40 # use the value as an output:
41 DO_AXIS_1 = 1
42 DI_AXIS_1 = 1
43 if value <= 0:
44   set_standard_digital_out(DO_AXIS_1, False)
45
46 # Wait for digital input to change state
47 #while (get_standard_digital_in(DI_AXIS_1) != False):
48 # sync()
49 #end
50 else:
51   set_standard_digital_out(DO_AXIS_1, True)
52
53 # Wait for digital input to change state
54 #while (get_standard_digital_in(DI_AXIS_1) != True):
55 # sync()
56 #end
57 end
58 end
59 #-----
60
61
62 # Main program:
63 # Program generated by RoboDK v5.2.5 for UR10e on 07/10/2021 14:09:35
64 # Using nominal kinematics.
65 ref_frame = p[-1.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000]
66 movej([-2.801847, -1.950632, -2.513379, -0.236496, 1.558403, -1.337519], accel_radss, speed_ra
67 movej([-3.008599, -2.362844, -1.574205, -0.761009, 1.561141, -1.340586], accel_radss, speed_ra
68 # Attach to GripperA
69 movej([-4.346219, -2.362827, -1.574112, -0.760964, 1.561022, -1.340413], accel_radss, speed_ra
70 # Detach from GripperA
71 movej([0.050022, -1.215910, -1.907139, -1.577900, 1.558342, -1.523566], accel_radss, speed_rad
72 # End of main program
73 end
74
75 place_box3()
76

```

5.3.3.8 Για την επίτευξη της εργασίας program 14:

Εδώ συλλέγονται όλες οι διαδικασίες και τα βήματα που ακολουθούν οι βραχιόνες, ώστε να ολοκληρώσουν μια εργασία βάφης για ένα από τα κυβάρια.

Πλατφόρμες ρομποτικών προσομοιωτών και εφαρμογές

```
Prog14.mod X
1  %%%
2  VERSION:1
3  LANGUAGE:ENGLISH
4  %%%
5  MODULE MOD_Prog14
6
7  ! -----
8  ! Define customized variables here
9  ! ...
10
11 ! Tool variables:
12 PERS tooldata Paintgun := [TRUE,[[50.000,0.000,450.000],[0.96592583,0.00000000,0.25881905,0.00000000]], [1,[0,0,20],[1,0,0,0],0,0,0.005]];
13
14 ! Reference variables:
15 PERS wobjdata Frame2 := [FALSE, TRUE, "", [[500.000,0.000,0.000],[1.00000000,0.00000000,0.00000000,0.00000000]], [[0,0,0],[1,0,0,0]]];
16
17 PROC approach_abb_irb()
18   ConfJ \On;
19   ConfL \Off;
20   MoveAbsJ [[-18.860200,28.132600,-70.168000,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
21   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
22 ENDPROC
23
24 PROC paint()
25   ConfJ \On;
26   ConfL \Off;
27   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
28   MoveAbsJ [[-19.171300,-2.742030,35.316600,35.734500,37.256400,-57.020300],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
29   MoveAbsJ [[-15.618200,3.479750,28.788800,34.081100,35.790000,-52.593000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
30   MoveAbsJ [[-17.490600,6.030810,26.252700,34.873900,36.674000,-54.815300],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
31   MoveAbsJ [[-14.983400,8.963870,22.731300,33.358100,35.968100,-51.285800],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
32   MoveAbsJ [[-4.345030,58.493300,-60.746800,7.284200,85.714900,-18.347400],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
33   MoveAbsJ [[-9.920000,61.410000,-66.860000,7.260000,88.880000,-17.910000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
34   MoveAbsJ [[-12.307100,37.771100,-23.317700,3.819430,74.133100,68.708800],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
35   MoveAbsJ [[-12.307100,37.771100,-23.317700,3.819430,74.133100,68.708800],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
36   MoveAbsJ [[40.760000,-25.460000,69.920000,-66.700000,45.930000,165.410000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
37   MoveAbsJ [[59.672900,-16.380300,64.956300,-60.606600,55.805600,177.534000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
38   MoveAbsJ [[43.738400,25.424700,15.666100,-46.514800,60.306200,45.500200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
39   MoveAbsJ [[40.907300,32.630600,7.027100,-46.897000,59.713400,42.595200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
40   MoveAbsJ [[38.493900,40.200100,-2.888840,-46.649800,59.949200,39.114500],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
41 ENDPROC
```

```
43 PROC approachhome()
44   ConfJ \On;
45   ConfL \Off;
46   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
47   MoveAbsJ [[-18.860200,28.132600,-70.168000,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
48 ENDPROC
49
50 PROC Main()
51   ConfJ \On;
52   ConfL \Off;
53   ! Program generated by RoboDK v5.2.5 for ABB IRB 120-3/0.6 on 07/10/2021 14:14:20
54   ! Using nominal kinematics.
55   pick_box3;
56   place_for_paint;
57   stop;
58   approach_abb_irb;
59   paint;
60   paint;
61   paint;
62   approachhome;
63   place_box3;
64 ENDPROC
65
66 ENDMODULE
67
```

5.3.3.9 main program

Και ο κωδικός που περιλαμβάνει την βαφή όλων των κύβων είναι ο εξής

```

mainprogram.mod
1  ***
2  VERSION:1
3  LANGUAGE:ENGLISH
4  ***
5  MODULE MOD_mainprogram
6
7  ! -----
8  ! Define customized variables here
9  ! ...
10
11 ! Tool variables:
12 PERS tooldata Paintgun := [TRUE,[[50.000,0.000,450.000],[0.95592583,0.00000000,0.25881905,0.00000000]],[[0,0,20],[1,0,0,0],[0,0,0,0.005]]];
13
14 ! Reference variables:
15 PERS wobjdata Frame2 := [FALSE, TRUE, "", [[500.000,0.000,0.000],[1.00000000,0.00000000,0.00000000,0.00000000]],[[0,0,0],[1,0,0,0]]];
16
17 PROC Prog14()
18   ConfJ \On;
19   ConfL \Off;
20   pick_box3;
21   place_for_paint;
22   stop;
23   approach_abb_irb;
24   paint;
25   paint;
26   paint;
27   approachhome;
28   place_box3;
29 ENDPROC
30
31 PROC approach_abb_irb()
32   ConfJ \On;
33   ConfL \Off;
34   MoveAbsJ [[-18.860200,28.132600,-70.168800,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
35   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
36 ENDPROC
37
38 PROC paint()
39   ConfJ \On;
40   ConfL \Off;
41   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
42   MoveAbsJ [[-19.171300,-2.742030,35.316600,35.734500,37.256400,-57.020300],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
43   MoveAbsJ [[-15.618200,3.479750,28.788800,34.081100,35.790000,-52.593000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
44   MoveAbsJ [[-17.490600,6.030810,26.252700,34.873900,36.674000,-54.815300],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
45   MoveAbsJ [[-14.983400,8.963870,22.731300,33.358100,35.968100,-51.285800],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
46   MoveAbsJ [[-4.345030,58.493300,-60.746800,7.284200,85.714900,-18.347400],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
47   MoveAbsJ [[-9.920000,61.410000,-66.860000,7.260000,88.880000,-17.910000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
48   MoveAbsJ [[-9.910000,61.410000,-63.240000,3.610000,90.230000,72.170000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
49   MoveAbsJ [[-12.307100,37.771100,-23.317700,3.819430,74.133100,68.708800],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
50   MoveAbsJ [[40.760000,-25.460000,69.920000,-66.700000,45.930000,165.410000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
51   MoveAbsJ [[59.672900,-16.380300,64.956300,-60.606600,55.805600,177.534000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
52   MoveAbsJ [[43.738400,25.424700,15.666100,-46.514800,60.306200,45.500200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
53   MoveAbsJ [[40.907300,32.638600,7.027100,-46.897000,59.713400,42.595200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
54   MoveAbsJ [[38.493900,40.200100,-2.888840,-46.649800,59.949200,39.114500],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
55 ENDPROC
56
57 PROC approachhome()
58   ConfJ \On;
59   ConfL \Off;
60   MoveAbsJ [[-22.203000,-7.179920,39.592300,36.618600,38.897200,-60.212200],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
61   MoveAbsJ [[-18.860200,28.132600,-70.168800,21.344900,105.137000,-21.096100],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,Paintgun \WObj:=Frame2;
62 ENDPROC
63
64 PROC Prog15()
65   ConfJ \On;
66   ConfL \Off;
67   pick_box4;
68   place_for_paint;
69   stop;
70   approach_abb_irb;
71   paint;
72   paint;
73   paint;
74   approachhome;
75   place_box4;
76 ENDPROC

```

```
mainprogram.mod x
78 PROC Prog16()
79   Conf1 \On;
80   Conf1 \Off;
81   pick_box1;
82   place_for_paint;
83   stop;
84   approach_abb_irb;
85   paint;
86   paint;
87   paint;
88   approachhome;
89   place_box1;
90   Program1;
91 ENDPROC
92
93 PROC Prog17()
94   Conf1 \On;
95   Conf1 \Off;
96   pick_box2;
97   place_for_paint;
98   stop;
99   approach_abb_irb;
100  paint;
101  paint;
102  paint;
103  approachhome;
104  place_box2;
105 ENDPROC
106
107 PROC Main()
108   Conf1 \On;
109   Conf1 \Off;
110   ! Program generated by RoboDK v5.2.5 for ABB IRB 120-3/0.6 on 07/10/2021 14:11:10
111   ! Using nominal kinematics.
112   SprayOn;
113   Prog14;
114   Prog15;
115   Prog16;
116   Prog17;
117   SprayOff;
118 ENDPROC
119
120 ENDMODULE
121
```

Αποτελέσματα

Με τις δύο αυτές εφαρμογές, μporέσαμε να χρησιμοποιήσουμε και να αξιοποιήσουμε επαρκώς τα εργαλεία που μας προσféρανε οι προσομοιωτές των V-rep και RoboDK. Αναλυτικότερα, με τις μηχανές φυσικής τους αισθητήρες και τα ρομποτικά μοντέλα μporέσαμε να απλοποιήσουμε τα πρόβλημά μας και στην συνέχεια να τα επιλύσουμε με ακρίβεια και αποτελεσματικότητα. Τέλος, αποκτήσαμε περισσότερες γνώσεις για το αντικείμενο προς μελέτη και κατανοήσαμε καλύτερα το τρόπο λειτουργίας του.

Συμπεράσματα

Στην διπλωματική αυτή εργασία, πραγματοποιήθηκε η παρουσίαση των ρομποτικών προσομοιωτών Gazebo, RoboDK, Open-Rave, Webots και V-rep. Αυτή περιελάμβανε την μελέτη και ανάλυση του κάθε λογισμικού, ως προς την αρχιτεκτονική δομή που αυτό ακολουθεί προκειμένου και να λειτουργήσει. Αναλυτικότερα, καταγράφηκαν πληροφορίες και μελετήθηκαν λεπτομερώς: το γραφικό περιβάλλον του κάθε προσομοιωτή, οι μηχανές φυσικής και κινηματικής που αυτό εμπεριέχει για να επιλύσει τα πολλαπλά προβλήματα προσομοίωσης, τα ρομποτικά μοντέλα που αυτό φέρει, οι αισθητήρες/ενεργοποιητές που ενσωματώνει, τα εργαλεία-αντικείμενα που αυτό παρέχει, αλλά και τα μειονεκτήματα-πλεονεκτήματα που σημειώνονται για αυτό ως λογισμικό.

Στην συνέχεια, με την χρήση δύο προσομοιωτών του V-rep και του RoboDK πραγματοποιήθηκαν δυο εφαρμογές, εκ των οποίων η πρώτη περιέγραφε ένα μηχανοκίνητο όχημα καταλλήλα διαμορφωμένο με αισθητήρες και μια κάμερα, το οποίο είχε τοποθετηθεί σε ένα περιβάλλον γεμάτο αντικείμενα, όπου με την σειρά τους είχαν το ρόλο των εμποδίων. Αυτό λοιπόν που δοκιμάστηκε σε αυτή την εφαρμογή ήταν εάν μέσω του προσομοιωτή V-rep, μπορούσε το όχημα εκ πρώτης όψης να “αντιληφθεί” τα εμπόδια και έπειτα να “αντιδράσει” και να τα αποφύγει.

Εντωμεταξύ, η δεύτερη εφαρμογή η οποία αναπτύχθηκε στο περιβάλλον του RoboDK, αφορούσε σε σχέση με την πρώτη δύο non-mobile ρομπότ. Και ειδικότερα δυο βιομηχανικούς βραχίονες, οι οποίοι είχαν προγραμματιστεί για να επιτελέσουν μια εργασία τύπου pick and place, όπως και μια τύπου paint. Έτσι, εφαρμόζοντας μια ακολουθία εντολών αυτό που κατάφεραν να κάνουν στην ουσία είναι να μεταφέρουν τα κυβάρια που είχαν επιλεχθεί εξαρχής, μέσω του βραχίονα που φέρει την αρπάγη, και μετά να τα τοποθετήσουν για βαφή. Αφού, είχαν τοποθετηθεί εκεί, ο δευτερός βραχίονας με την χρήση του paint gun υλοποιούσε την βαφή. Αυτή η διαδικασία βέβαια αποτελούσε ένα κύκλο εργασιών, και επαναλαμβανόταν για κάθε ένα από τα κυβάρια ξεχωριστά.

Επομένως, μέσω του ρομποτικού προσομοιωτή, το κάθε άτομο, μπορεί αντίστοιχα να δοκιμάσει και να επιλύσει αναρίθμητα προβλήματα, εφαρμόζοντας είτε τα δικά τους μοντέλα είτε αυτά που προσφέρει ο προσομοιωτής. Μπορεί, ακόμη να χρησιμοποιήσει παράλληλες προσομοιώσεις και να μοντελοποιήσει το περιβάλλον του ή και γενικότερα την προσομοίωση όσες φορές χρειαστεί. Επιπλέον, μπορεί αποφύγει αρκετά λάθη κατά την διάρκεια επιλυσης του προβλήματος, άρα και να μειώσει την πιθανότητα καταστροφής του περιβάλλοντος ή και του ρομπότ. Ωστόσο, το σημαντικότερο, μπορεί να καλύψει θέματα ασφαλείας, όπως και να εξοικονομήσει όχι μόνο χρόνο αλλά και χρηματικές μονάδες, καθώς η ρομποτική προσομοίωση είναι μια εφαρμογή που δεν απαιτεί έξοδα για την αγορά ρομποτικού εξοπλισμού ή άλλων εξαρτημάτων.

Βιβλιογραφία – Αναφορές – Διαδικτυακές πηγές

1. Pavel Hamet, Johanne Tremblay, 2017, *Artificial intelligence in medicine* [e-journal], available at website www.sciencedirect.com [accessed August 2021]
2. Lentin Joseph, June 2018, *Learning Robotics using Python* [e-book] available at website www.books.google.gr[accessed August 2021]
3. Jan Hodicky, 2015, *Modelling and Simulation for Autonomous Systems*[e-book] available at website www.books.google.gr[accessed August 2021]
4. Gonçalo Nuno dos Santos Augusto,2013, *RobotTeamSim – 3D Visualization of Cooperative Mobile Robot Missions in Gazebo Virtual environment* [e-book] available at website www.books.google.gr[accessed August 2021]
5. Wyatt Newman, 2017, *A Systematic Approach to Learning Robot Programming with ROS* [e-book],available at website www.books.google.gr[accessed August 2021]
6. Anis Koubaa, 2020, *Robot Operating System (ROS): The Complete Reference (Volume 5)* [e-book],available at website www.books.google.gr[accessed August 2021].
7. Gregory Junker,2007, *Pro OGRE 3D Programming*, [e-book],available at website www.books.google.gr[accessed August 2021].
8. Enrique Fernández, Luis Sánchez Crespo, Anil Mahtani,2015, *Learning ROS for Robotics Programming (Edition 2)*, [e-book],available at website www.books.google.gr[accessed August 2021].
9. Carol Fairchild, Dr. Thomas L. Harman, 2017, *ROS Robotics By Example*, [e-book],available at website www.books.google.gr[accessed August 2021].
10. Michael A. Sherman, Ajay Seth, Scott L. Delp, 2017, *Multibody dynamics for biomedical research* [e-journal], available at website www.sciencedirect.com [accessed August 2021]
11. Kenta Takaya, Toshinori Asai, Valeri Kroumov and Florentin Smarandache, 2016, *Simulation environment for mobile robots testing using ROS and Gazebo* [pdf], available at website www.ntua.gr [accessed April 2021]
12. Καρακωνσταντίνου Χρίστου, Τερζή Ιωάννα, 2013, *Πρότυπος ρομποτικός κειμενογράφος* [pdf]
13. Προκόπης Κιούσης, 2017, *Ρομποτικός βραχίονας 4 βαθμών ελευθερίας* [pdf]

14. Open-Source Robotics Foundation, 2012, *DRC Kickoff: Gazebo Workshop* [pdf], available at website <http://www.gazebosim.org> [accessed August 2021]
15. Γερολυμάτος Σταύρος, 2019, *Προσομοίωση Οδήγησης Αυτοκινούμενων Οχημάτων με αισθητήρες όρασης σε προκαθορισμένες τροχιές* [pdf]
16. Γιαχούδης Νικόλαος, 2014, *Ευφυής έλεγχος και πλοήγηση ρομπότ μέσω της πλατφόρμας ARIA Advanced Robot Interface for Applications* [pdf]
17. Lucas Nogueira, 2018, *Comparative Analysis Between Gazebo and V-REP Robotic Simulators* [pdf]
18. Κοντογεωργάκος Σπυρίδων, Λούτα Παναγιώτα - Βασιλική, 2010, *Ανάπτυξη λογισμικού ευφυούς αυτόνομου ρομπότ* [pdf]
19. Παναγιώτη – Νικόλαο Κοσκινόπουλο, Γεώργιο Δημήτριο Ξύραφα, 2019, *Συγκριτική μελέτη εγκατάστασης αυτόνομου ρομποτικού συστήματος ανακύκλωσης αστικών απορριμμάτων* [pdf]
20. Νικόλαο Φ. Μιχαηλίδη, 2014, *Προσομοίωση Ρομποτικών Εφαρμογών με χρήση του λογισμικού “V-Rep”* [pdf]
21. Νικολάου Γ. Πεχλιβανίδη, 2013, *Σχεδίαση και ανάπτυξη προσομοιωτή του ρομπότ της ομάδας PANDORA σε εικονικά περιβάλλοντα* [pdf]
22. Βιαννιτάκη Βασιλική, 2014, *ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ, ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ ΥΓΕΙΑΣ* [pdf]
23. Beth Crane, Stephen Sherratt, 2013, *rUNSWift 2D Simulator; Behavioural Simulation Integrated with the rUNSWift Architecture* [pdf]
24. ΓΙΑΝΤΣΙΟΣ ΓΕΩΡΓΙΟΣ, 2012, *Παιχνίδι περιπέτειας σε PROLOG* [pdf]
25. Συριανίδου Σοφία, 2017, *Προσομοίωση της δυναμικής συμπεριφοράς εξωσκελετικού μηχανισμού και ελέγχου του, από ενσωματωμένο πρόγραμμα αποτελούμενο από λογισμικό ρομποτικού συστήματος και ελέγχου σύμφωνα με το πρότυπο IEC61499.* [pdf]
26. Καρράς Άγις, *Βιομηχανική Ρομποτική* [pdf]
27. Δ. Τσούμπας, 2015, *Έλεγχος βιομηχανικού ρομπότ με βάση ανοιχτό λογισμικό ψηφιακής καθοδήγησης εργαλειομηχανών.* [pdf]
28. ΜΗΛΙΩΤΗΣ Α. ΣΤΕΦΑΝΟΣ, 2019, *Σχεδιασμός και Υλοποίηση Παραμετροποιήσιμου Προσομοιωτή Μη Επανδρωμένων Εναέριων Οχημάτων* [pdf]
29. Απόστολος Α. Πουρνάρας, 2011, *Προσομοίωση Αλγορίθμων Διάταξης με Εκπαιδευτικό Ρομπότ* [pdf]
30. Γεώργιου Εξίζογλου, 2019, *Μελέτη και Ανάλυση Ρομποτικών Προσομοιωτών* [pdf]

31. Αδάμου Ηλιάνα, 2018, *Πλατφόρμες ρομποτικής προσομοίωσης η περίπτωση Gazebo* [pdf]
32. Αρετή Τσορμπάρη, 2018, *Προγραμματισμός ρομπότ σε περιβάλλοντα μεικτής πραγματικότητας για την απόκτηση γεωγραφικών και ιστορικών γνώσεων.* [pdf]
33. Leon ˇ Zlajpah, 2008, *Simulation in robotics* [e-journal], available at website www.sciencedirect.com [accessed July 2021]
34. Raúl Crespo, René García, Samuel Quiroz, 2015, *Virtual Reality Simulator For Robotics Learning* [pdf], available at website www.ntua.gr [accessed April 2021]
35. Mirella Santos Pessoa de Melo, Jos´e Gomes da Silva Neto, Pedro Jorge Lima da Silva, Joˆao Marcelo Xavier Natario Teixeira and Veronica Teichrieb, 2019, *Analysis and Comparison of Robotics 3D Simulators* [pdf], available at website www.ntua.gr [accessed April 2021]
36. Tim Laue, Kai Spiess, and Thomas Rofer, 2006, *SimRobot – A General Physical Robot Simulator and Its Application in RoboCup* [pdf]
37. E. Abelea, F. Haehna, M. Pischana, F. Herra, 2016, *Time optimal path planning for industrial robots using STL data files* [e-journal], available at website www.sciencedirect.com [accessed November 2020]
38. Rosen Diankov, James Kuffner, 2008, *OpenRAVE: A Planning Architecture for Autonomous Robotics* [pdf]
39. E. Peralta E. Fabregas G. Farias H. Vargas, S. Dormido, 2016, *Development of a Khepera IV Library for the V-REP Simulator* [e-journal], available at website www.sciencedirect.com [accessed November 2020]
40. Eric Rohmer, Surya P. N. Singh and Marc Freese, 2013, *V-REP: a Versatile and Scalable Robot Simulation Framework* [pdf], available at website www.ntua.gr [accessed November 2020]
41. Miguel A. Olivares-Mendez, Somasundar Kannan, Holger Voos, 2015, *Vision Based Fuzzy Control Autonomous Landing with UAVs: From V-REP to Real Experiments* [pdf], available at website www.ntua.gr [accessed November 2020]
42. Eric Rohmer, Surya P. N. Singh and Marc Freese, 2018, *CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework* [pdf], available at website www.coppeliarobotics.com [Accessed April 2021]
43. Marc Freese, *V---REP and its Matlab interface* [pdf], available at website www.coppeliarobotics.com [Accessed April 2021]

44. Lukas Hohl, Ricardo Tellez, Olivier Michel, Auke Jan Ijspeert, 2006, *Aibo and Webots: Simulation, wireless remote control and controller transfer* [e-journal], available at website www.sciencedirect.com [accessed November 2020]
45. Michel, O. / Cyberbotics Ltd - Webots™: *Professional Mobile Robot Simulation*, pp. 39-42, International Journal of Advanced Robotic Systems, Volume 1 Number 1 (2004), ISSN 1729-8806.
46. Gazebo Tutorials, available at website gazebo.org [Accessed April 2021]
47. robodk.com
48. cyberbotics.com
49. <http://openrave.org/>
50. <https://www.coppeliarobotics.com/>
51. <https://www.iso.org/>
52. <https://github.com>
53. http://www.me.unm.edu/~starr/research/WAM_UsersGuide_AE-00.pdf
54. <https://support.barrett.com/wiki/WAM/KinematicsJointRangesConversionFactors>
55. <https://www.robots.com>
56. <https://robotsdoneright.com/KUKA/KRC2-robots/kuka-kr-5-r650.html>
57. <https://www.kuka.com>
58. https://www.researchgate.net/figure/Mitsubishi-PA10-7CE-robot_fig1_221908922
59. <https://schunk.com>