

ΠΛΗ 412 - Αυτόνομοι Πράκτορες - 2025

3η Εργαστηριακή Άσκηση

25 Νοεμβρίου 2025

Καράλης Αστερινός AM: 2020030107

1. To Setup

Η υλοποίηση του project πραγματοποιήθηκε σε περιβάλλον Ubuntu 24.04 LTS (Virtual Machine με host OS Windows 11). Εγκαταστάθηκε ο προσωμοιωτής Gazebo Jetty, μέσω του οποίου αναπτύχθηκαν τα επιμέρους στάδια της εργασίας. Κάθε φάση βασίστηκε στο αντίστοιχο tutorial, με τις τροποποιήσεις στον κώδικα να αναλύονται στη συνέχεια.

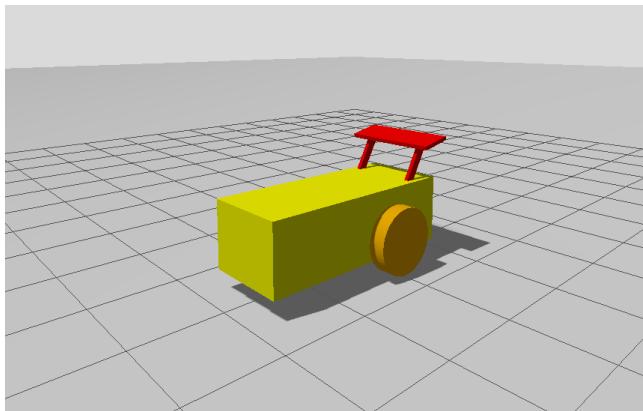
2. Υλοποίηση

2.1. Build your own robot

Στην ενότητα αυτή, τροποποιήθηκε το βασικό μοντέλο του ρομπότ από το tutorial με στόχο τη δημιουργία ενός custom robot. Οι κυριότερες αλλαγές συνοψίζονται ως εξής:

- **Διαστάσεις και μάζα:** Τροποποιήθηκαν οι διαστάσεις στο σασί και τις ρόδες, ενώ αυξήθηκε η μάζα για την επίτευξη μεγαλύτερης σταθερότητας.
- **Εμφάνιση:** Προστέθηκαν visual elements για τη δημιουργία αεροτομής στο πίσω μέρος του ρομπότ, ενώ εφαρμόστηκαν διαφορετικοί χρωματισμοί στα επιμέρους τμήματά του.
- **Sensors:** Η θέση του sensor προσαρμόστηκε στο πάνω μέρος του ρομπότ, ώστε η λειτουργία του να μην επηρεάζεται από τις νέες διαστάσεις.

Παρακάτω απεικονίζεται το τελικό μοντέλο μετά την ολοκλήρωση των αλλαγών.



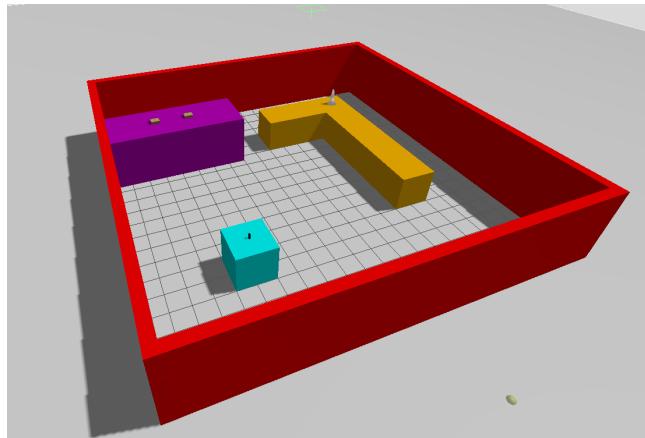
Σχήμα 1: Το τελικό όχημα-robot

2.2. Moving the robot

Αναφορικά με την κίνηση του ρομπότ, αρχικά ακολουθήθηκε η μεθοδολογία του tutorial, χρησιμοποιώντας το script που επιτρέπει χειροκίνητο έλεγχο μέσω των arrow keys. Στη συνέχεια, υλοποιήθηκε αυτόνομη κίνηση με χρήση δεδομένων από τους sensors, προσδιδόντας στο ρομπότ σχετική αυτόνομη κίνηση.

2.3. SDF World

Σχεδιάστηκε ένας custom κόσμος, ο οποίος οριοθετείται από κόκκινους τοίχους στις άκρες του βασικού plain. Το περιβάλλον εμπλουτίστηκε με μεγάλα εμπόδια διαφόρων χρωμάτων, καθώς και μοντέλα από τη βιβλιοθήκη Fuel, όπως κουτιά και κώνους.



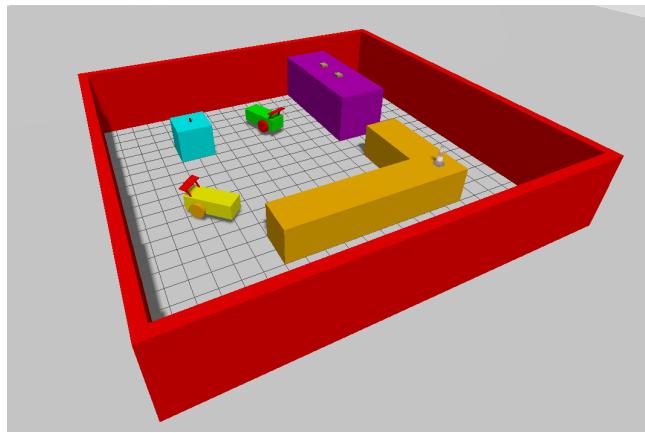
Σχήμα 2: Το τελικό περιβάλλον.

2.4. Sensors και αλλαγές

Στο στάδιο αυτό ενσωματώθηκαν οι sensors για την επίτευξη της επιθυμητής αυτονομίας. Παρόλο που ακολουθήθηκε η βάση του tutorial, ο κώδικας (`lidar_node.cc`) τροποποιήθηκε ώστε να υποστηρίζει την ταυτόχρονη ύπαρξη και έλεγχο δύο ρομπότ στον χώρο (με το δεύτερο να φέρει διαφορετικό χρωματισμό για λόγους διάκρισης).

Συγκεκριμένα, το βασικό node κάνει `subscribe` σε δύο διαφορετικά Lidar topics (`/lidar` και `/lidar_green`) και δημοσιεύει (public) εντολές ταχύτητας σε δύο ξεχωριστά topics.

Επιπλέον, διαφοροποιήθηκε η λογική αποφυγής εμποδίων ώστε τα ρομπότ να επιδεικνύουν διακριτή συμπεριφορά: το κίτρινο ρομπότ στρίβει αριστερά όταν ανιχνεύσει εμπόδιο, ενώ το πράσινο στρίβει δεξιά.



Σχήμα 3: Το τελικό simulation με τα δύο ρομπότ.

3. Video

Στο βίντεο παρουσιάζεται η εκτέλεση του simulation όπου τα δύο ρομπότ πλοηγούνται στον χώρο. Κατά τον εντοπισμό εμποδίου μέσω του sensor, πραγματοποιούν την αντίστοιχη στροφή και συνεχίζουν την πορεία τους.

4. Δομή εργασίας

Τα αρχεία που απαρτίζουν την εργασία και είναι προσβάσιμα εντός του simulator είναι τα εξής:

- **gz sim robot_only.sdf**: Αρχείο για την προβολή του μοντέλου του οχήματος.
- **gz sim env_only.sdf**: Αρχείο για τη φόρτωση αποκλειστικά του περιβάλλοντος (world).
- **gz sim sensor_tutorial.sdf**: Αρχείο που περιλαμβάνει τον τελικό χόσμο με τα ρομπότ.
- **gz launch sensor_launch.gzlaunch**: (Το κυρίως αρχείο εκτέλεσης) Χρησιμοποιείται για την εκκίνηση του simulation. Φορτώνει ταυτόχρονα το περιβάλλον, τα δύο ρομπότ και τα control nodes για την αυτόνομη κίνησή τους.
- **lidar_node.cc**: Περιέχει τη λογική πλοήγησης των ρομπότ. Το αρχείο είναι ήδη built, οπότε αρκεί η εκτέλεση του simulation (μέσω της παραπάνω εντολής). Διαφορετικά πρέπει να γίνει build με cmake και έπειτα make.