The University of Western Australia
Dept. of Electrical, Electronic & Computer Engineering
Associate Lecturer Kieran Quirke-Brown

Mobile Robots

AUTO4508

Group Project	Autonomous Navigation	weeks 2-12

GROUPS: Form groups of 4 students

EQUIPMENT: Pioneer 3-AT Outdoor Mobile Robot Platform
Industrial Linux PC with touch screen display
IMU: Phidget Spatial 3/3/3
Camera: Stereo Camera OAK-D V2
Lidar: SICK 2D-Lidar TIM781-2174101 or richbeam lakis1
Software: ROS2

Groups will share the pioneer robots located in the robotics lab.

Part 1 – due in week 6

Tasks to complete

1. Drive the robot along a path, specified through a number of given GPS way-points. The robot has to visit each waypoint before returning to its starting position.
2. Each waypoint is marked by an orange traffic cone. Whenever a waypoint has been reached (within 1-2 meters), the robot must take a photo of the marker object and then head towards the next waypoint.
   Always leave this marker to the robot's **right** side.

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python

3. At each waypoint, an additional object (a colored bucket) will be in the vicinity, but at an unspecified distance. Identify the object, record a photo of it, and calculate its distance from the waypoint marker.
4. A summary of the journey should be presented on the completion of driving.
5. Use the Lidar sensor to avoid collisions with markers, objects and any other stationary or moving obstacles, such as walls, vehicles, people, bikes, etc.
6. For safety reasons, implement a Bluetooth link between the robot's on-board PC and a gamepad controller:
    a. Button 'X' enable **a**utomated mode.
       In automated mode, use the back pedals as a dead-man switch.
       If released, the robot has to stop.
    b. Button 'O' enable manual mode (disable automated mode).
       In manual mode, the steering controls can be used to manually drive the robot forward/backward and left/right.

Part 2 – due in week 12

Tasks to complete

Your robot has landed on an unknown surface and it must explore the area and log any details for the arrival of future modules. In order to safely deploy the modules a 15x15 meter area around the initial landing place needs to be searched. Your robot must be able to identify certain colour markers noting the location and colour. There are several areas of interest that will be marked as waypoints and after the initial exploration the robot will need to inspect them a second time as quickly as possible. Your team will need to build a simulation as well to demonstrate the robots capabilities before real world testing can begin.

1. The robot should maintain the capabilities from part 1 while extending the functionality for below.
2. From a set home position you need to explore an unknown area, mapping it as you drive.
3. Within the unknown environment are a number of hand drawn numbers, as seen in Figure 1. These numbers will be located randomly around the environment on different surfaces at approximately knee height. You will need to use image recognition to determine the number and note its location.

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python

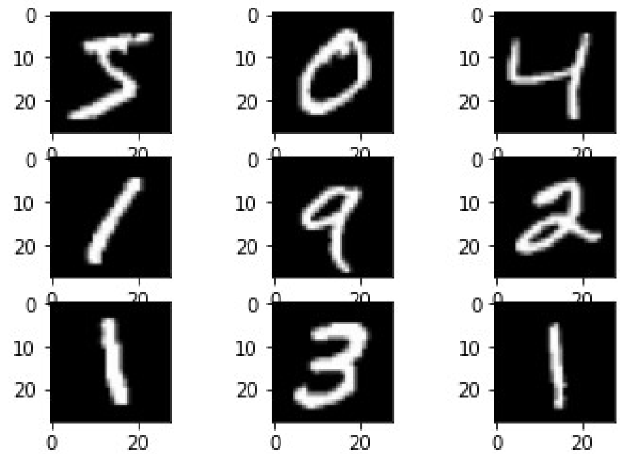

Figure 1: Hand drawn numbers[1].

4. You should take photos and note the location of any yellow or red colour obstacles in the area as these are of special interest to the team back home.
5. Avoid collision with moving obstacles and record an incident if an emergency stop is needed (an emergency stop should occur if a moving object comes within 1m of the robot). Note you don't need to press the estop, this is a software requirement.
6. If a estop event occurs, save the last 5 seconds of recorded data so that it can be reviewed by the team.
7. Upon completion of mapping the unknown environment, print the map, all marker photos and locations to the screen.
8. Implement a user interface with graphics and text on the robot's display that always displays the robot's internal state and its intended actions.
9. Once your first run is completed you will be given 3 waypoints noted by the hand drawn numbers to drive to. Your team must plan the fastest path to each of these locations and the path back to home. Your robot must then drive through the unknown environment a second time to each of these waypoints and return home in the fastest time possible.
10. Display the robots planned path graphically to a screen.
11. Your system should record the drives so it can be review again offline.

Getting started

You can program the robot using the ROS2 software stack. Make use of the software libraries for OAK-D (stereo camera), sick_scan_xd (Lidar) as well as OpenCV (image processing) which stops you from having to reinvent the wheel. As a suggested first node, try and get the PlayStation controller working using the joy library. The Phidget IMU does have a ROS2 library but making your own node tends to achieve better results.

Note that the robots are designed to **only drive outside** on grass or sand.
For **indoor testing**, you need to wrap all tires with tape to allow the rigid wheels to slip – otherwise the robot's motors will burn out (this already happened in the past)!

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python

Resources

Pioneer:

- ARIA Library: https://github.com/cinvesrob/Aria

Phidget IMU:

- User Guide: https://www.phidgets.com/?tier=3&catid=10&pcid=8&prodid=1204
- Code Samples https://www.phidgets.com/?tier=3&catid=10&pcid=8&prodid=1204

OAK-D Camera:

- DepthAI API: https://docs.luxonis.com/projects/sdk/en/latest/
- Code Samples: https://docs.luxonis.com/projects/api/en/latest/tutorials/code_samples/

SICK Lidar:

- Info/ROS-Driver: https://www.sick.com/fr/en/tim781-2174101/p/p594148

Lakibeam:

- ROS2 driver + info: https://github.com/RichbeamTechnology/Lakibeam_ROS2_Driver

---

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python

VIDEO

Create a video of about 1-2 min. in length that shows your robot's capabilities, you should outline your team's journey, what went well and what didn't go so well, as well as design decisions and changes in decisions. If anything goes wrong during demonstration day, at least you have the video to show your work so make sure your features are captured accurately. It is probably worth filming during the entire length of the project to capture working parts.
Submission is in week 11, one week before the project demonstration.

DEMONSTRATION

On the scheduled presentation day at the end of the semester, all groups will give a practical demonstration of their projects and answer the project supervisors' questions re. their implementation. In week 6 you will need to demo part 1 of the project. In week 10 we will have a pre demonstration for part 2 of the project.

SUBMISSION

Submit a hardcopy with official coversheet incl. declarations of all team members:

1. Project design report (*pdf*), which includes
    - Report on which team member did what
    - Software design description
    - Diagrams, photos, screenshots, plots, etc.
    - Include page numbers
    - You need to reference any sources that you used.
    - Max 10 pages plus 1 Title page and bibliography page
    - This should be a document you could hand to another design team, and they could develop the same/similar solution based on your design decisions

      Do NOT include:
    - Program code (snippets are ok for explanation purposes)
    - Table of contents, etc.
2. User Manual (*pdf*)
    - Max 5 pages, **no** Title page
    - This document should be a guide for a general (non-technical) user on how to use all the features of your system.
    - As if it was sold to a customer
        - Think about what you see in other user manuals both bad and good
        - A good suggestion would be to show a family member or someone not familiar with robotics and make sure they understand how to use your system.
3. Source code *(email to project supervisor only, no hardcopy)*, clearly marking any imported code with referencing the source.

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python

MARKING

70%	Functional Performance, Design, Complexity, Innovation

- 15% Simulation
- 55% Real world
  - 15% Part 1
  - 40% Part 2

15%	Project Design Report

5%	Video

10%	User Manual

1 https://www.digitalocean.com/community/tutorials/mnist-dataset-in-python