IVDC

Intelligent Vehicle Design and Control Club

Automathon'24

AUTONOMY TEAM CHALLENGE

Guidelines:

1. You are given 4 Problem Statements for Inducting into Autonomy Division- IVDC Club
2. COMPUTER VISIONAND ML, PATH PLANNING ALGORITHMS, ESTIMATION AND LOCALISATION and ROS based PS each having a background info and different tasks with individual points associated.
3. Each Task in a PS has separate points as mentioned beside the tasks, to a. Study the existing methods b. Solve the tasks as required
4. **You can attempt one or more PS , partial points will be awarded as well. (More points implies more potential to get into club)**
5. Quick Start resources are provided for each task.
6. You are free to use any website, dataset, publication, frameworks/ modules/ libraries, algorithms, Github repos, etc.
7. But in all the above cases, you should clearly mention references to your sources in the submission document or a separate references.txt file.
8. There will be a penalty for plagiarism (i.e., not giving clear references)
9. You are not allowed to plagiarize from your fellow candidates. In all such cases, both entries will be outright rejected.
10. For Doubts you can use IVDC INTEREST group

Submission Details:

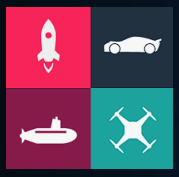
Deadline : 14th April 2024

Submit your code as a
- shared drive folder
- provide access to Anyone with the link by sharing in Google Drive.

OR

(preferred)
- Share link of your public Github repository
- Remember to work in private github repo before submission

"The world is changing fast, and autonomous vehicles are going to change it even faster."
- Jeff Holden, Co-founder and CEO of LV



I V D C

Intelligent Vehicle Design and Control Club

UBUNTU DUAL BOOT GUIDELINES:

1. Download ubuntu 20.04 iso image from the official website: click here
2. For usb flashing use : balena etcher or rufus (preferred)
3. Refer VIDEO For further steps on dual boot. You can also ask doubts related to it in ivdc interest group

ROS INSTALLATION GUIDELINES:

RUN THE FOLLOWING COMMANDS

```
$ sudo apt install ros-noetic-desktop-full
$ source /opt/ros/noetic/setup.bash
```

For further details: refer ros wiki

For ROS tutorials refer ros_tutorials. Complete the tutorials for a basic understanding before approaching the ps.

NOTE: UBUNTU AND ROS GUILDLINES ARE FOR ROS BASED PS ONLY, FOR OTHER PS YOU MAY USE ANY OS (WINDOWS / MAC/ LINUX)

For computer vision and ml PS - NIKETH (79814 03918)
SOHAM (98201 44271)

For ROS, path planning and ekf PS – BHAWNA (98915 00451)
ARJUN (96058 83021)
AMPADY (94462 48549)

"The world is changing fast, and autonomous vehicles are going to change it even faster."
- Jeff Holden, Co-founder and CEO of LV

1) COMPUTER VISION AND ML BASED PS

Tasks:

1. How do different types of filters (e.g., Gaussian, Sobel, Laplacian) impact image processing in computer vision, and what are their specific applications? [3]

2. How do adaptive filters adapt to local variations in images, and what advantages do they offer over fixed filters in tasks like image denoising and edge detection? [4]

3. Explain the concept of filter kernels in convolutional neural networks (CNNs), and how altering kernel size and weights influence feature extraction and model performance. [2]

4. Discuss the importance of filter design in real-time computer vision applications, considering factors like computational efficiency, accuracy, and adaptability to varying environmental conditions. [3]

5. Go through all the filters which are available and get a thorough understanding why and how these filters are used, if possible, make a note of them. Then make a try to design a filter that can achieve the task below? (reducing intensity of horizontal-power lines) [5]

Figure 1: before processing

Recognizing traffic signals is the most important task while dealing with the autonomous vehicles, for which a very good accuracy is required

6. Divide the dataset into train, validation and test data splits and apply data preprocessing if required, try different types of models and explore all of them, you need a tradeoff between accuracy and inference speed [8]

7. Now your task is to train a model for recognizing traffic signs and maintain a higher accuracy. Build a model to your best [10]

For dataset : click here

RESOURCES:

- StanfordVL/cs131_notes: Class notes for CS 131. (github.com)
- SzeliskiBookDraft_20210828.pdf (dropbox.com)
- training vs inference

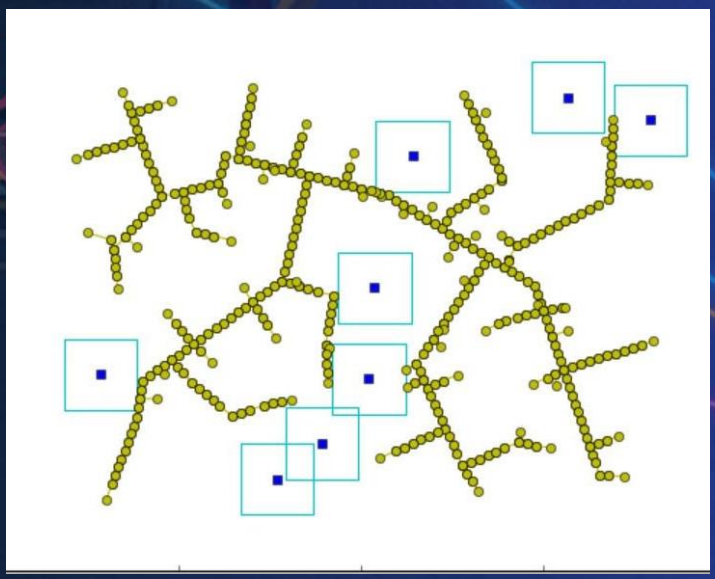
2) PATH PLANNING ALGORITHMS

Tasks:

1. Explore different path planning algorithms and classify each algorithm. Aslo compare them. Should not exceed 5 to 6 lines. [5]

2. Which algorithm do you think could be used by an UGV to navigate a track consisting of obstacles such as trees, shrubs, light posts, street signs, barrels, etc. It should be fast and computationally efficient. [2]

NOTE : We do not expect you to write elaborative answers, rather understand them. We would grade you based on your understanding.



3. Understand the RRT* algorithm and complete the given code in path planning folder [8]

References :

- rrt*
- Rrt
- rrt wiki
- rrt notes
- path planning
- path planning algorithms

3) ROS based PS

Tasks:

Getting started :

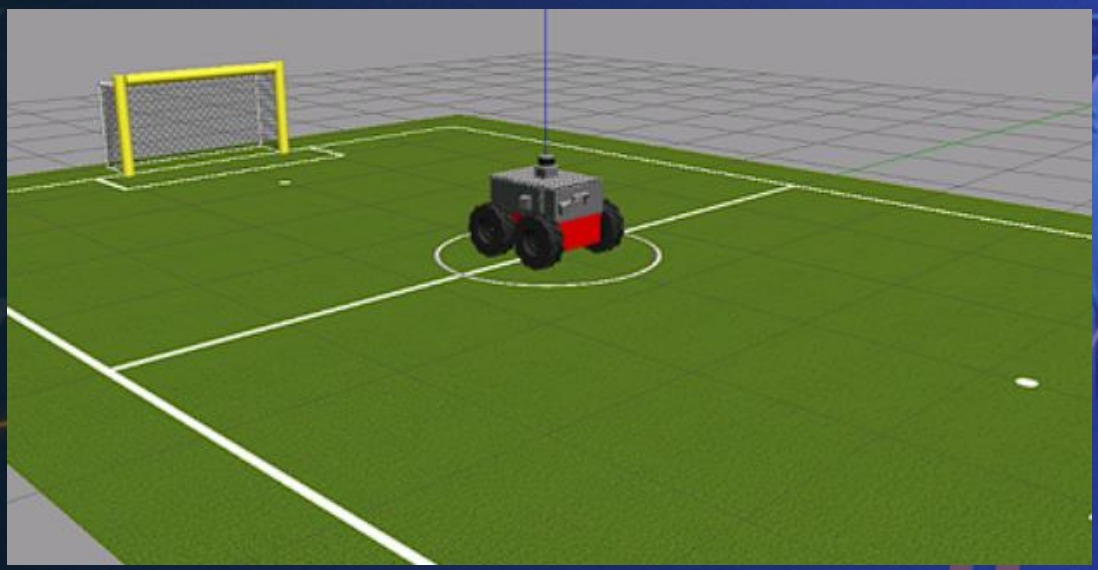
What is ROS ? Explain in brief why we need ROS ? Which are the main ROS distros being used today ? [Need only brief, not more than 4-5 lines. We would grade you based on your understanding.] [5]

1. Mention the file name in workspace which provides information about robot type and sensors used ? Also name the sensors and robots used. [2]

2. What is service, actions, messages, nodes, topics and rqt_graph ? (Tell in brief). [2]

3. Run appropriate commands to show active topic list, messages and message details (eg, type, subscribed topic, publishing to, etc…). [Take a screenshot of the terminal window] [4]

4. Run command to display the rqt_graph. [Take a screenshot of the terminal window] [2]



Now…

5. Create a subscriber node to the topic which is used to control bot's velocity. [10]

6. Create a subscriber node to the /scan topic. [10]

7. Create a node to move the bot 10 m straight and stop, then scan the environment in front of the bot using laser scan. [20]

8. Show the laser scan in RViz and add RViz to your launch file. Make sure to set odomas the Fixed Frame (under Global Options). take screen shot of RViz screen Store this scan data. [Refer to ros_wiki to find how to store sensor data]. (Save this data file inside smb_gazebo. This stored data will be evaluated.)

Optional [BONUS] : [20]

9. Study about Quaternions and Euler angles. [write in brief] (Not more than 5-6 lines.).

10. Mention the coordinates of both goal posts and convert them to quaternions. [We do not expect you to understand them in depth overnight, but every effort from your side will be graded.]

You can work on construct

By

creating an ->account
-> new rosject(short for ROS project)
➔ ROS noetic distro

OR

work in your own Ubuntu 20.04 and ROS Noetic (if you installed it properly)

Clone the following repo inside src of your ws : github repo

Run the following cmds in terminal (of Constructsim Online or your Linux OS)
(there should bean folder named catkin_ws before these cmds)

$ cd ~/catkin_ws/src
$ git clone https://github.com/IVDC-Club-IIT Indore/Automathon2024_ROS_PS.git
$ cd ..
$ catkin_make

You are all set to code now!

ALL THE BEST ! (Sometimes starting is more important than worrying, we look in you, your enthusiasm and efforts and IT WILL NOT GO UNNOTICED.)

We have provided a Node(a file) inside smb_gazebo/src/ named controller.cpp
you need to change this only for writing the velocity commands.
(there are hints provided inside the file)

Submission should be a Workspace(src the one inside catkin_ws) folder as seen above containing required code.

HINTS :

1. Try referring our Workshop and then course material over here
2. For this PS you can refer to this resource
3. For all the ROS enthusiasts, join this group for ROS related queries

4) ESTIMATION AND LOCALISATION

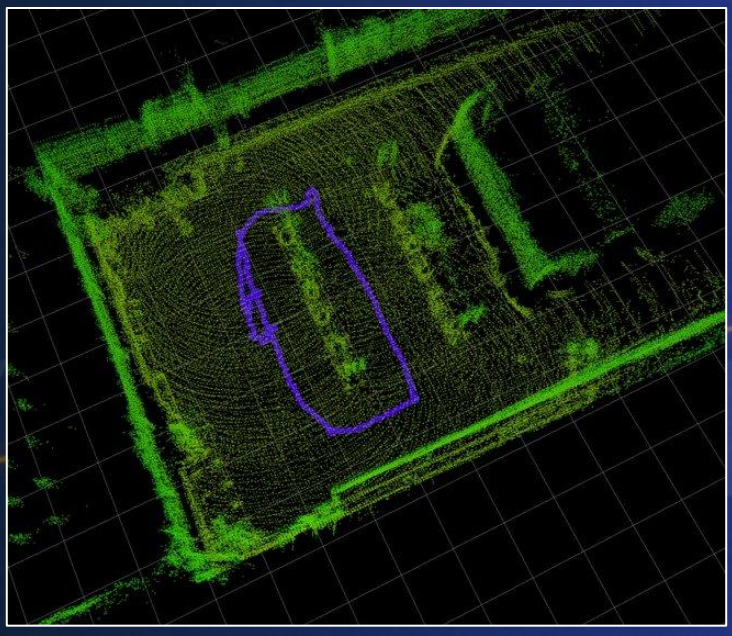
HINT: Simultaneous localization and mapping (**SLAM**) is the computational problem of constructing a map of an unknown environment while simultaneously keeping track of an agent's location within it. Lidar scans are one of the commonly used estimation method based on point cloud

Tasks:

1. Explain the basic point cloud data structure (try to explain how we would translate the raw lidar data into a form that can be processed using python) (Hint: consider an example of a stationary lidar in front of a tree having finite number of laser scan points.) [5]

2. You are a person testing lidar based algorithm on a controlled environment. Mention some of the precautionary measures you would take. [3]
Hint : think of when lidar measurement would fail

3. The rotation of the vehicle frame w.r.t the lidar frame is denoted by the rotation matrix c. given any point p in the lidar frame and considering only rotation transform it into the vehicle frame. (give the mathematical model) [4]



4. Explain Kalman filter by considering the problem of estimating the 1d problem of a vehicle. (Cosider approximate assumptions) [We expect you to understand why Kalman Filter is needed.) [5]

5. Explore the pros and cons of Kalman filter and explain the need of extended Kalman filter. Complete the ekf algorithm in the given ekf_ps folder. You can refer the folder for more hints on the mathematical algorithm. [10]

NOTE: we don't expect you to complete the code, but partial solving will be considered as bonus

References :

1. kalman filter wiki
2. ekf wiki
3. kalman filter
4. ekf resource
5. Read Chapter 6, Sections 1 and 2 of Timothy D. Barfoot, State Estimation for Robotics (2016) – for point cloud
   http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf
6. lidar