Electronics and Robotics Club, BITS Goa

Induction Assignment

Submission: 11:59PM, 13th May 2022

Instructions

This assignment is open for all irrespective of whether you have enrolled in the mini-projects or the CTE on Intro to Robotics conducted by ERC.

For people who have enrolled in any of the mini projects, the assignment contains one task relevant to each mini-project that the ERC has been running this year.
Solve at least one of the given tasks
Submit a link to your solution in the application form:
https://bit.ly/erc_assignment

Resources:
Many resources and tips have been shared during the weekly meets as well as on the Discord channels for each project.
If you have not signed up for the projects till now, reach out to the contact for the specific project to get added to the discord channel or post on the #projects-assignment channel.
Discord invite: https://discord.gg/gkDECXQf

Resources are also given in the project handouts available at erc-bpgc.github.io/mini_projects

Tasks

Complete at least one task of your choice and submit your solution in the format specified. Feel free to ask your doubts on discord or message any one of the mentors. Hope you have fun while solving the assignment!

IMPORTANT: Submit whatever you have done even if it is not complete. We are looking for dedicated members who are willing to put in efforts. Please do not shy away from submitting an incomplete task.

[A] Path Planning

Contact

- Yash Yelmame: +91 7044299226
- Aditya Parandekar: +91 9922924118
- Laukik B Nakhwa: +91 9869193993

Description

You will be given a set of obstacles and one source point and a destination point. All coordinates will be given. You need to find a path from the source to the destination using RRT. The path should not intersect any obstacles and should be a perfectly valid path. You will also be given a bounding box in which you are allowed to sample the points for making the tree.

Evaluation

You will be evaluated on whether you take in consideration all the types of obstacles or not and the amount of time it takes your algorithm to find the path. We will also evaluate how readable your code is and how efficiently you used functions and objects in your code.

The Assignment

The bounding box is the square with corners at (0, 0), (0, 100), (100, 100), (100, 0).
The obstacle list, start point and goal point are as follows.

You can copy paste the following code in your program, and make functions for RRT and visualise

```
obstacle_list = [
    [(40, 0), (40, 40), (50, 50), (60, 40), (50, 40)],
    [(10, 10), (20, 20), (10, 30), (0, 20)],
    [(50, 60), (70, 80), (60, 100), (40, 80), (45, 100)],
    [(70, 20), (90, 20), (80, 40)]
]


start = (1, 1)
```

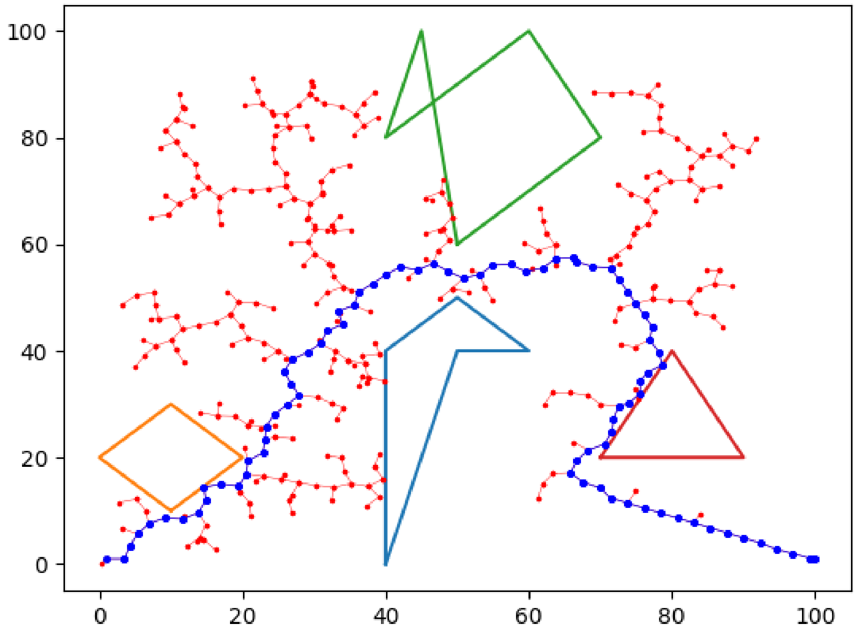
```
goal = (100, 1)

# Calculate Path using RRT
path = RRT(start,goal, obstacle_list)

# Visualize the calculated path in Matplotlib
visualize(path, obstacle_list)
```

You can choose to follow this code format or do as you wish. But, the obstacles, start point and goal point should be the same.

[IMPORTANT]

Please note that you need to sample the goal point every once in a while. Otherwise your code will take way longer to find a path, because of how the obstacles are designed.

Expected Result



Submission

Submit the link to a github repository containing the following:

1. Your python code (preferably have everything in one single python file)
2. A screenshot of your implementation

[B] Neural Networks

Contact

- Manan Arora: +91 8368794567
- Soham Chitnis: +91 9819765828
- Siddh Gosar: +91 8879336028

Description

Implement a neural network and its associated training procedure in NumPy to classify numerical digits(0-9) using the MNIST dataset. Create and submit the link to a GitHub repo or a Google Colab notebook containing all the code results and a short (500 words) write-up talking about your implementation and any difficulties/curiosities you came across.

[C] Computer Vision

Contact

- Siddh Gosar: +91 8879336028
- Soham Chitnis: +91 9819765828

Description

Using OpenCV in python, write a script to detect the hand gesture in video capture through the webcam. You can use any method you like to achieve the desired result. It can be a mask, a contour, or a bounding box around the hand. Create a GitHub repository containing all your python code and submit the link to the repo.
Try to add a good readme/documentation to your repo, explaining your approach.

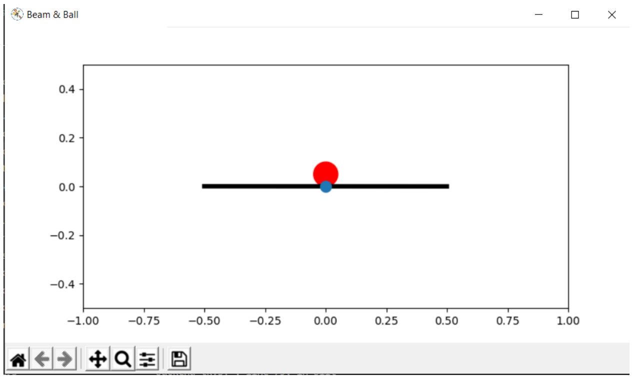
[D] Controls

Contact

- Atharva Ghotavadekar: +91 9767280239
- Dhruv Potdar: +91 70450 30300
- Yash Yelmame: +91 7044299226
- Aditya Parandekar: +91 9922924118

Description

Write a PID controller to balance a ball on a beam using Open AI gym's [ballbeam-gym environment](). Create a Github repo containing all the code along with necessary comments explaining your code and screenshots of the graphs obtained.
To install the API use `pip install ballbeam-gym`. You should use the `BallBeamSetpoint-v0` environment for this task.



Use the following parameters for the environment :

- timestep = 0.05
- Beam_length = 1.0
- Max_angle = 0.5
- Init_velocity = 0.5
- max_timesteps: 100

Apart from the inbuilt visual plot of the ball beam you also need to plot the following -

- The graph of ball position on the beam vs time step
- The graph of beam angle vs time step

Some important points:

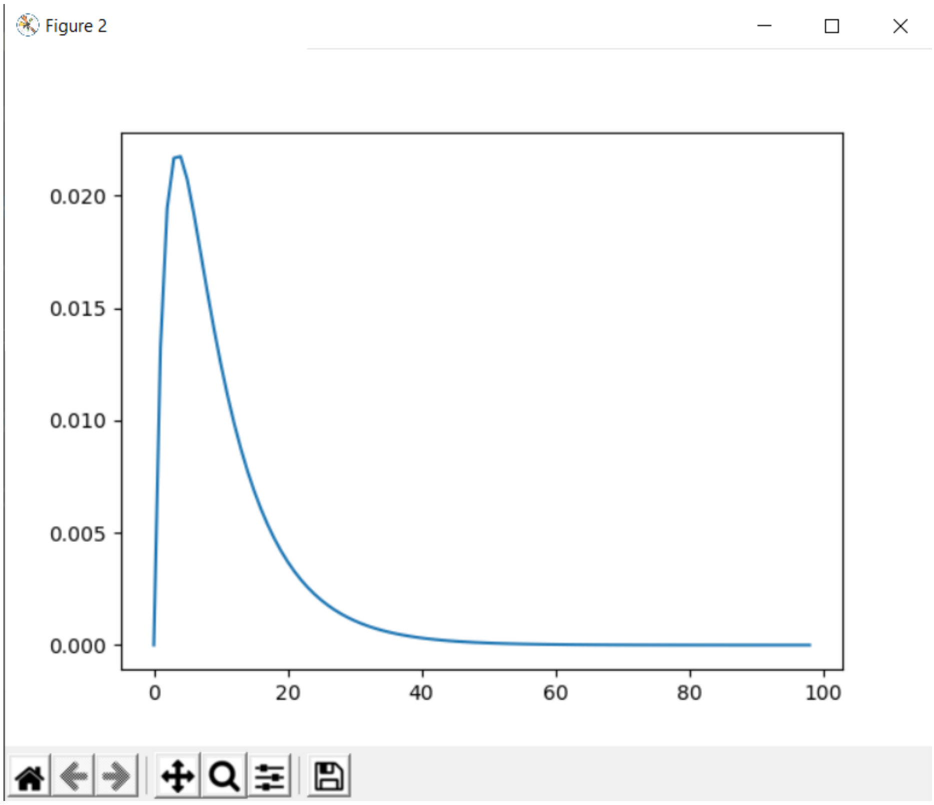
- Position of the ball must be taken from the observation space matrix of the environment
- Error should be taken as the distance of the ball position from the centre of the beam
- You can choose any appropriate Kp, Ki and Kd values for the controller that settles the ball position to 0 by the end of 100 timesteps(it is appreciated to find optimum values such as your system has least overshoot and settling time)
- You should run at least three episodes of the simulation to see if there are any changes in the graphs (you need plot graphs for all three episodes separately)

NOTE: Please go through the [balbeam-gym]() repo and [Open AI gym docs]() thoroughly before attempting this question

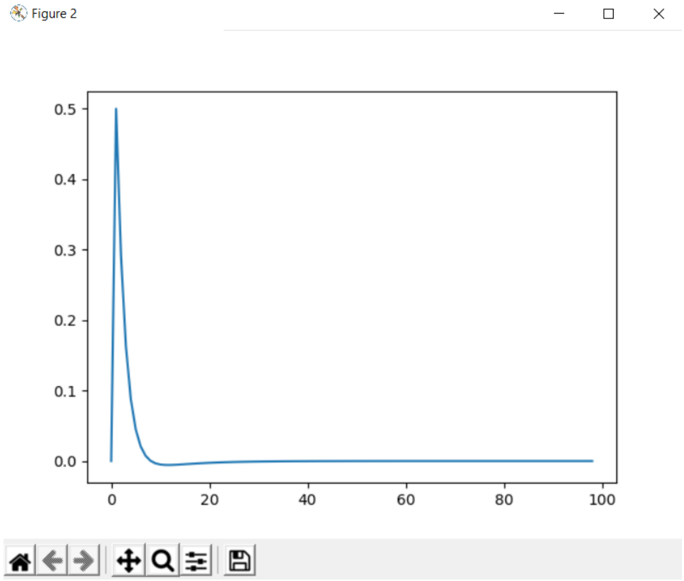
Example plots:

Note: You don't have to replicate the given graphs, they are just examples for your reference

Ball Position Vs Time Step



Beam Angle Vs Time step



`

[E] Mechanical

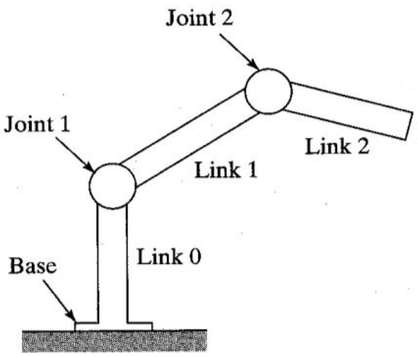
Contact

- Ishan Tandon: +91 9818860396
- Yash Chavan: +91 9049005126
- Sai Akshit: +91 7337093444

Description

Design a 2 DOF (Degree of Freedom) robotic arm on Fusion 360 or Solidworks.

A diagram of the 2 DOF arm looks like this:



NOTE : You are free to choose your own design, and use references from the internet for design inspiration. The files you submit should be original.

Bonus tasks (*optional*) :

1) Design a gripper to be attached at the end of the arm. You are free to choose a mechanism for the gripper.
2) Write down forward kinematics equations for your two links. Refer to this video to learn what forward kinematics is: https://youtu.be/3WAk8ABj-kg

Submission

Your submission should include the Fusion 360 or Solidworks files + a PDF containing the bonus task calculations. Upload all of these in a github repository and submit the link to the repository in the submission form

NOTE : Submit whatever you are able to do, your effort and method matters.
We will discuss your approach during the interviews.

[F] Arduino Clock

Contact

- Ajay Krishna Gurubaran: +91 8451012540
- Tanay Patni: +91 8793131372

Description

Using a 16x2 LCD Display and 4x4 Numeric Keypad, build a circuit integrating clock, timer, and stopwatch. The user must set the timer and clock using a numeric keypad. Use a piezo buzzer that buzzes when the timer counts down to 0. For the stopwatch, the timer starts from 0 seconds. Use push buttons or keypad (choice is left to you) to implement stop, start, reset, and pause.

Submission: You have to simulate the whole circuit and the code using Tinkercad. Submit the link of your Tinkercad project in the assignment form given above.

[IMPORTANT] : Don't forget to make your project public, you can share the link only after that.

[G] Arduino Car

Contact

- Sahil Shingote +91 7038110010
- Rushil Parihar +91 9341787445

Description

Using the distance read by HCSR04 and you have to implement an obstacle avoiding car using the differential drive model. HCSR04 will be mounted on a servo so you can see obstacles in left and right and hence make decisions on where to rotate or to stop.
So now consider 4 HCSR04s are connected to make a square and each gives the value of distance to obstacles in range, check the value of the distances continuously. Your car should make a decision on the basis of these readings and rotate to face the direction where no obstacle is detected.

Submission: You have to simulate the whole circuit and the code using Tinkercad. Submit the link of your Tinkercad project in the assignment form given above.

[IMPORTANT] : Don't forget to make your project public, you can share the link only after that.

Using Github

Submissions other than the Arduino mini projects require you submit the link to a github repository containing your solution. Here are a few resources to get familiar with github:

1) Creating a github account:
   Sign up here: https://github.com/

2) Creating a new repository:
   https://docs.github.com/en/get-started/quickstart/create-a-repo

3) Uploading a file to a repository:
   https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository

NOTE (**Please ignore this if you have not used github before**):
In case you have used github before and try to use SSH, SSH will not work over open networks like BITS internet.

Feel free to reach out to any of the mentors in case of any doubts or difficulties with Github as well. Hope to see you in ERC soon!

For any other queries contact
Ashutosh Gupta - +91 9909230000

Social Links

- Official Website: https://erc-bpgc.github.io//
- ERC Blog: https://erc-bpgc.github.io/blog/
- Github: https://github.com/ERC-BPGC
- Instagram: https://www.instagram.com/erc_bitsgoa/
- Facebook: https://www.facebook.com/ElectronicsAndRoboticsClub
- LinkedIn: https://www.linkedin.com/company/electronics-robotics-club-bits-goa
- Twitter: https://twitter.com/erc_bpgc?s=08