EEE3099S Design Project 2023: Line Maze Follower Treasure Hunt

Introduction:

This course aims to assign students to a design problem relevant to the Mechatronics discipline within which they will need to design a prototype and test a sub-system. This will provide insight into understanding the intricacies of real-life complex sub-system design. Students will be expected to solve a problem methodically using the skills they have gathered over the previous semesters of the curriculum, especially from the Design Principles course. They will also get a glimpse of acquiring external knowledge which they haven't come across in their daily classes.

Project Description:

In groups of three, students must develop an autonomous mobile robot to participate in a "treasure hunt", the treasure being object placed in our maze. A robot will be placed at the beginning of the maze and with the flick of a switch or a push button, the robot will then be tasked to autonomously navigate the maze by tracking a line and identifying object placed along the maze. Each object is placed at a specific distance from a "detect line" (a line in which the robot must stop and measure distance to the object). Upon discovering all the object present in the maze, the robot will then be tasked to navigate to the object that is within the closest distance to its corresponding detect line.

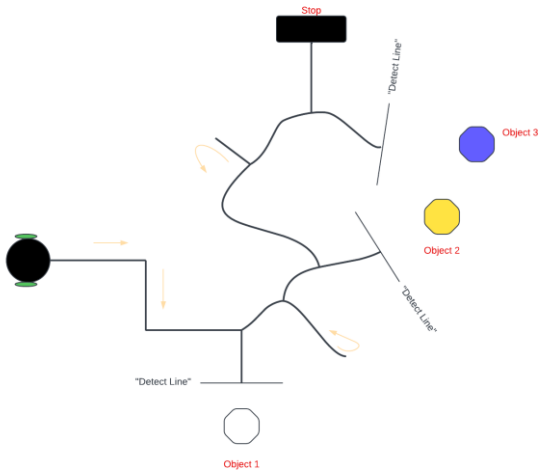

Figure 1:Shows an example treasure map with the treasure being objects. The robot should navigate to all three objects on the map and measure each objects distance from the "detect line". Upon completion of all three measurements, the robot should then be able to navigate to the closest object to its corresponding detect line upon request. NOTE: Map does not represent the actual number of objects to be placed on the maze, this will change.

Constraints

A self-powered differential drive robot will be provided with the following:

- Ultrasonic Sensor. https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi
- Arduino Nano 33 IoT. https://docs.arduino.cc/hardware/nano-33-iot

- 2 x 18650 3.7V Batteries
- 2 x Logic Level Converter Bi-Directional - Micro Robotics Logic Level Converter Bihttps://www.robotics.org.za/LEVEL-4PDirectional - Micro Robotics
- 1 x MOTOR DRIVER DUAL H-BRIDGE MODULE L298N https://www.diyelectronics.co.za/store/brushed-motor-drivers/34-motor-driverdual-h-bridge-module-l298n.html
- 1 x Turtle: 2WD Mobile Robot Platform.
    - https://wiki.dfrobot.com/2WD_Mobile_Platform_for_Arduino__SKU_ROB0005_
- Axle length: 13.6 cm
- Wheel Diameter: 6.2 cm
- 4 x Gravity: Digital Line Tracking(Following) Sensor.
    - https://wiki.dfrobot.com/Line_Tracking_Sensor_for_Arduino_V4_SKU_SEN0017
- 2 x Gravity: TT Motor Encoders Kit. https://www.dfrobot.com/wiki/index.php/Wheel_Encoders_for_DFRobot_3PA_and_4WD_Rovers_(SKU:SEN0038)

MATLAB and Simulink should be used for both simulation and real-world implementation:

- Arduino Support from Simulink (https://www.mathworks.com/hardware-support/arduinohttps://www.mathworks.com/hardware-support/arduino-simulink.htmlsimulink.html), should be used to test and run your models on the Arduino. **You may not use the Arduino IDE**
- To simulate the robot, the following tool kit should be used https://www.mathworks.com/matlabcentral/fileexchange/62966-student-competitionhttps://www.mathworks.com/matlabcentral/fileexchange/62966-student-competition-mobile-robotics-trainingmobile-robotics-training

A simulation template will be provided later.

Marks Distribution & Key Dates

Assessment Task	%	Deadline
Milestone 1 (Onramp Courses , Paper design)	25	14th August
Milestone 2 (Robot Motion Control, Line and object detection in Simulation. Complete physical circuit.)	25	14th September
Milestone 3 (The Hunt)	25	13th October
Final Report	25	20th October
Total	**100**	

MATLAB AND SIMULINK SETUP

You may install MATLAB 2023b on your personal computer (see http://www.icts.uct.ac.za/matlab). The university has a campus-wide license that makes this possible. You may also use MATLAB online (see: https://www.mathworks.com/products/matlab-online.html).
Toolboxes you need on your local machine:

- MATLAB,
- Simulink
- Control System Toolbox
- Image Processing Toolbox
- Simulink Control Design
- System Identification Toolbox
- Image Processing Toolbox
- Robotics System Toolbox
- Mobile Robotics Training Toolbox
- MATLAB Support Package for Arduino Software

PENALTIES

Each day of late submission translates to a penalty of 5% (max five days).
10% will be deducted if a report includes any hand-drawn diagram or diagram that is not visible if printed.

RULES

- **Report**:
  - With each hand-in, you need to submit a 1-page report detailing which members were responsible for each aspect of the hand-in and what each member did. o All reports should be submitted on Amathuba in PDF format only unless otherwise stated.
  - No handwritten reports and hand-drawn diagrams will be accepted in the report submission.
- **Demonstration**:
  - For demonstrations, at least one member must be present for the time slot they signed up for. Members that are not present need to have a valid excuse; otherwise, marks will be deducted from their individual demo marks.
  - Demonstrations can only occur during the time slot sign up for by the group. (No swapping or longer demonstrations).
- **Teamwork**:
  - The lecturer may use the team's summary of contribution by each member to distribute marks unequally amongst the team members in the case of a dispute.
  - Group complaints will not be dealt with one week prior to or any time after any hand-in is due.
  - Weekly progress reports will be required upon completion of each lab session.
- **The tutors, TA and lecturer may refuse**:
  - to assist the students if the students have put no effort into the solving of the problem.
  - to assist the students if the students have conducted no or insufficient research in the solving of the problem.

- to assist the students with debugging if the students have no schematics, flow charts, datasheets etc. for the tutors, TA and lecturer.

- **General liability and responsibility**:
  - Students should undertake to:
    - Treat staff and fellow students with dignity and respect, especially in lectures, tutorials, laboratories, and libraries.
    - Prepare for and attend all classes, tutorials, practical and other activities scheduled for the courses in which they are registered.
    - Complete all submissions and any other course requirements to the best of their ability, handing in work on time.
    - Take responsibility for their own learning, while also interacting constructively with their fellow students, lecturers, and tutors.
    - Address personal issues that might reduce the chances of success in good time so that these do not limit learning opportunities.
    - Not cheat, and not submit work of others as their own.
    - Complete course and lecturer evaluations for each course they are registered for.

  - Students are fully liable for:
    - The consequence of not submitting the required component table in the specified format. The tutors, TA and lecturer will not run after the component table from each group.
    - The consequence of breaking the component(s). The tutors, TA and lecturer will not provide the students with alternatives if any component is damaged.
    - The consequence of breaking/losing the robot. The tutors, TA and lecturer will not provide the students with extra components and materials if the robot is damaged or lost.

  - It is students' responsibility to:
    - submit the requested hand-ins in time in a specified format.
    - sign up for a timeslot for the demonstrations.
    - appear on time for the demonstrations.
    - ask questions.
    - inform the TA and lecturer of any valid reasons for late hand-ins and conflict in the group at least a week prior to any hand-ins.

Extra Reading Material

Below are links to external resources that might be of some help when planning out your project:

Line Tracking in Simulink:
https://www.mathworks.com/help/supportpkg/legomindstormsev3/examples/line-tracking.html

Line Follower Application for Arduino® Robot:
https://www.mathworks.com/help/supportpkg/arduino/examples/arduino-robot-line-followerapplication.html

Mobile Robotics Training: https://www.mathworks.com/videos/series/student-competition-mobilerobotics-training.html

MATLAB and Simulink Code Generation Resources:

- Arduino Support from Simulink - Hardware Support - MATLAB & Simulink (mathworks.com) - Additional Capabilities with Simulink Coder and Embedded Coder - MATLAB & Simulink (mathworks.com)
- Code Execution Profiling for Arduino Hardware in External Mode - MATLAB & Simulink Example (mathworks.com)

Code Execution Profiling for Arduino Hardware in External Mode - MATLAB & Simulink Example (mathworks.com)

Differential drive: http://planning.cs.uiuc.edu/node659.html

Mobile Robotics Training: https://www.mathworks.com/videos/series/student-competitionmobile-robotics-training.html

Encoder Ticks: http://faculty.salina.k-state.edu/tim/robot_prog/MobileBot/Pose/encoders.html

RomeoV2: https://wiki.dfrobot.com/Romeo_V2-

All_in_one_Controller__R3___SKU_DFR0225_

Dead reckoning lab:
http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/s07/labs/NXTLabs/Lab%2
03.html

Encoders:
https://www.youtube.com/watch?v=oLBYHbLO8W0&ab_channel=SparkFunElectronics

Maze Solving Algorithms:

- https://en.wikipedia.org/wiki/Maze-solving_algorithm

- https://medium.com/@TowardInfinity/coding-a-line-follower-robot-using-lsrb-andfinding-the-shortest-path-d906ffec71d