EE4306: DISTRIBUTED AUTONOMOUS ROBOTIC SYSTEMS 2013/2014 SEMESTER 2 PROJECT

INTRODUCTION

In this project, the Microsoft Robotics Developer Studio 2008 Release 3 (MRDS 2008 R3) [or newer versions] will be used to introduce the concept of programming a mobile robot in a virtual world. A mobile robot is one that is able to navigate within a physical environment, unlike industrial robots which are fixed at a particular spot. The mobile robot will be required to solve some tasks in a virtual world. In the process, an understanding of the difficulties involved in making an autonomous mobile robot will be acquired.

TEAM FORMATION AND COMPUTER REQUIREMENT DETAILS

TEAM FORMATION

This project must be done as a team consisting of 2 or 3 persons, and you are allowed to form your own team. Please sign up for the team using IVLE (Tools -> Groups -> Sign Up -> EE4306 Module). The deadline for registration will be given in a separate IVLE announcement. Any individuals who have not registered their teams by that deadline will be randomly assigned to form a team.

COMPUTER REQUIREMENT

MRDS R3 has varying computer requirement depending on the complexity of the simulation being carried out. Simple programming instructions and simple environments can be carried out with mid-end notebooks purchased after 2008. Running complex simulations on underperforming computing systems might result in strange mobile robot behaviours and lagging systems, even if the algorithm used for the mobile robots are functionally correct.

This project starts with low complexity simulations and moves towards higher complexity near the end. You may use your own computer if it can cope with your simulations. Alternatively, five computers in the Control and Simulation Lab (E4A Level 3) have been reserved for EE4306, with suitable hardware requirement, as indicated below:

Processor:	Intel Core 2 Duo 3.0 GHz
Memory:	3 GB
Graphics:	Nvidia GT240 1 GB

COMPUTER AVAILABILITY

The five EE4306 reserved computers in the Control and Simulation Lab (E4A Level 3) may be used during normal office hours. You may use the computers as long as they are free, but note that there will be a few days when the five reserved computers will not be available for use by EE4306 students. You will be given further information on the computer availability dates through IVLE.

PROJECT REQUIREMENT

PHYSICAL DESIGN OF ROBOT

The focus of this project will be on the algorithm involved in making mobile robots (also called entities in MRDS) perform their intended tasks. It is **not required** to design the physical characteristics of the robots. Existing physical design of mobile robots is already available in MRDS R3 for immediate use, with some examples listed below. You may alter the characteristics of the entities, such as the weight or colours, if you require it.

Existing Entity	Number of Wheels
Pioneer 3DX	Two
iRobot Create	Two
Lego NXT Tribot	Three
Sampled 4x4 Vehicle	Four

There will be no penalty involved if you choose to physically design your own robot or download them from the internet. However, it will be your own complete responsibility to ensure that the robots can work in MRDS. Any type of robots, whether wheeled robots or bipedal robots, can be used as long as the tasks involved can be accomplished.

PROGRAMMING LANGUAGE

The Simple Programming Language (SPL) is recommended for this project, and complex algorithms can be implemented by using SPL. You are free to learn how to use C# or any other programming languages to program more advanced services and manifests if you need them and are at ease with them.

VISUAL SIMULATION ENVIRONMENT

The Visual Simulation Environment (VSE) is a virtual world with which the robot can interact with. Several VSE are already available in MRDS R3 and others can be downloaded from the internet. C# is generally used to design additional items in the environment, such as chairs, boxes, television etc... You are **not required** to learn C# if you are not at ease with it. A virtual world that suits your needs can easily be created by using the existing VSE, copying the required objects, and pasting them to your virtual world. Most basic objects will be available in the existing virtual worlds, such as chairs, tables, trees, houses, walls, balls, lamp posts, sign boards etc... Alternatively, there are several programs available online that can create simple environments, such as the Maze Simulator (The *Maze Simulator* service creates a simple grid world in the Robotics Studio simulation environment. It takes a black and white image as input where every black pixel becomes a cube in the simulator)

PROJECT TASK

TASK OVERVIEW

In our real world, people use vehicles with drivers to get from point A to point B. Some experimental vehicles that can drive from point A to point B without drivers and without collision have been demonstrated, but they are far from replacing the vehicle drivers. You are required to implement a simplified part of this task in a virtual world using MRDS.

The task is broken down into sections and you are required to implement as much as possible within each sections. Each section has to be presented by the given deadline, as will be announced through IVLE. The sections are:

Section	Brief Title	Comments
Section 1A	Moving Obstacle Avoidance	Milestone 1
Section 1B	Path Planning Using waypoints	Milestone 2
Section 2	Simple Vehicle Overtaking on a Closed Circuit	Milestone 3: This is a combination of Section 1A and Section 1B
Section 3	Advanced Vehicle Overtaking on a Closed Circuit	Milestone 3: The bonus section consisting of more complex tasks

For all the sections, you may use any mobile robots suitable for your virtual world, and the mobile robots do not need to be of the same type. However, their relative sizes should be appropriate for the tasks involved.

Sensors suitable for the different sections include the laser range finder (LRF), compass, proximity sensor etc...

Section 1A: Moving Obstacle Avoidance

1. Create or use a virtual world that will be able to demonstrate the task of having one autonomous robot overtaking another robot. As a start, you can imagine a straight road with only two lanes. It is recommended that you do not make any visually appealing environment in this section as that part will be covered in section 2.

2. Add one entity, which we call “Robot 1” for ease of explanation. You may choose any other name for the robot when using MRDS. Make it move in a straight line from point A to point B. There is no need to make the robot stop at point A or point B in this section, that is, the separation between point A and point B can be infinitely apart. Avoid making this mobile robot from moving too fast.

3. Add a second entity, which we call “Robot 2” for ease of explanation. You may choose any other name for the robot when using MRDS. Place “Robot 2” a few metres behind “Robot 1” and ensure that “Robot 2” is moving faster than “Robot 1”. This will allow “Robot 2” to catch up with “Robot 1”

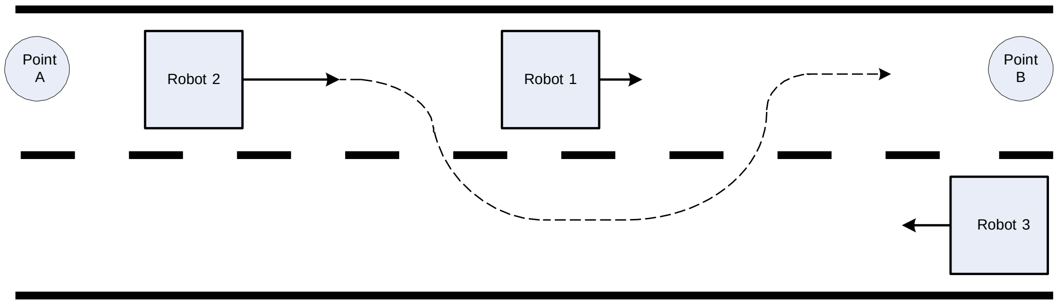
4. When "Robot 2" is slightly behind "Robot 1", "Robot 2" with appropriate sensors must be programmed to overtake "Robot 1" and move in front. "Robot 2" must always stay in its own lane unless it is overtaking "Robot 1"

5. Add a third entity, which we call "Robot 3" for ease of explanation. You may choose any other name for the robot when using MRDS. "Robot 3" is travelling in the other lane and in the opposite direction to "Robot 1" and "Robot 2".

6. Program "Robot 2" to avoid overtaking whenever there is a possibility of collision with "Robot 3". When collision is likely to happen. "Robot 2" will reduce its speed to that of "Robot 1" and only overtake when the way is clear

This section attempts to simulate what a vehicle would do in real life when there is a need to overtake. In the virtual world, "Robot 1" and "Robot 3" would be considered as a moving obstacle that would need to be avoided, so that "Robot 2" can continue on its path. There is no need for "Robot 1" and "Robot 3" to have any sensors for this section.

For ease of understanding, Figure 1 shows a possible scenario for this section, with the dotted lines showing a smooth path taken by "Robot 2" to overtake "Robot 1". Your program does not need to replicate this scenario exactly.



Brief Description for Section 1A: Robot 2 must show its ability to overtake Robot 1. There must not be any collisions involved

Robot 1:	**Speed: Constant, Slower than Robot 2** **Direction: A to B, Straight Lane 1**
Robot 2:	**Speed: Varying, Faster than Robot 1** **Direction: A to B, Straight Lane 1**
Robot 3:	**Speed: Constant, Arbitrary** **Direction: B to A, Straight Lane 2**

Figure 1: Possible Scenario for Section 1A

Section 1B: Path Planning Using Waypoints

1. Create or use a virtual world that will be able to demonstrate the task of making a robot move around a closed circuit by passing through waypoints. As a start, you can imagine a closed circuit with only two lanes. It is recommended that you do not make any visually appealing environment in this section as that part will be covered in section 2.

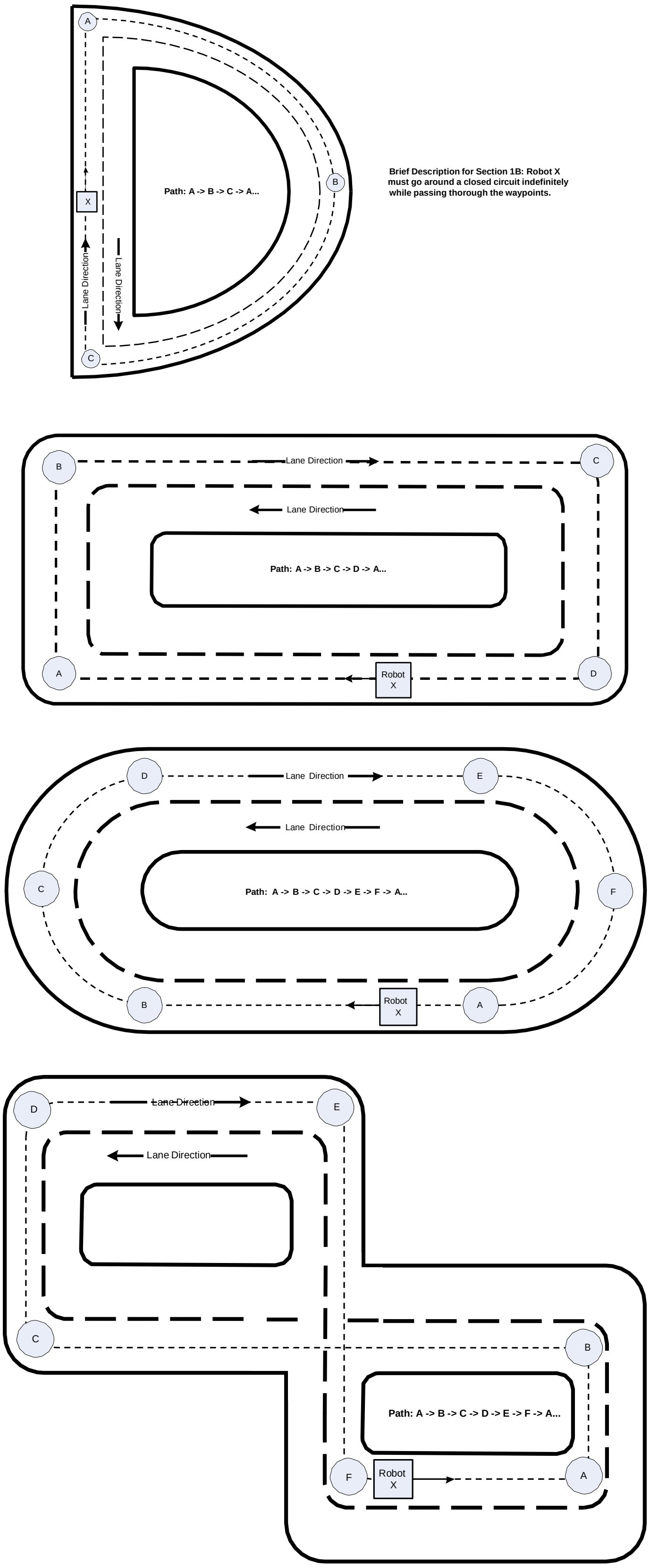
2. Add one entity in any of the two lanes, which we call "Robot X" for ease of explanation. You may choose any other name for the robot when using MRDS. Program "Robot X' to move through several waypoints on your circuit

3. Make use of appropriate sensors to ensure that the robot has reached a particular waypoint and is in the correct orientation to reach the next waypoint. Making use of only trigonometric functions without sensors will make your algorithm less robust and more prone to errors

4. "Robot X" should keep repeating the circuit, that is, after one complete cycle "Robot X" will do another complete cycle. This process repeats itself non-stop.

Additional info: *You can make use of landmarks that indicate the position of the robot. For example, there can be some markings on the road and the robot can use visual recognition sensors to recognize them. A map of the environment can be determined with the laser range finder if there are walls around the circuit, and therefore sharp corners can also be determined.*

For ease of understanding, a few possible scenarios for this section are shown in Figure 2, with the dotted lines showing the path taken by "Robot X" to move through all the waypoints. Your program does not need to replicate these scenarios exactly and you are encouraged to use closed circuits matching your own needs and capability.



Brief Description for Section 1B: Robot X must go around a closed circuit indefinitely while passing thorough the waypoints.

Figure 2: Possible Scenarios for Section 1B

Section 2: Simple Vehicle Overtaking on a Closed Circuit

1. You are now required to add “Robot 1” and “Robot 3” from Section 1A on the closed circuit designed in Section 1B. These robots should go around the closed circuit continuously, that is, they should keep repeating their path around the closed circuit. They must pass through your designated waypoints like in Section 1B but on their own lanes like in Section 1A

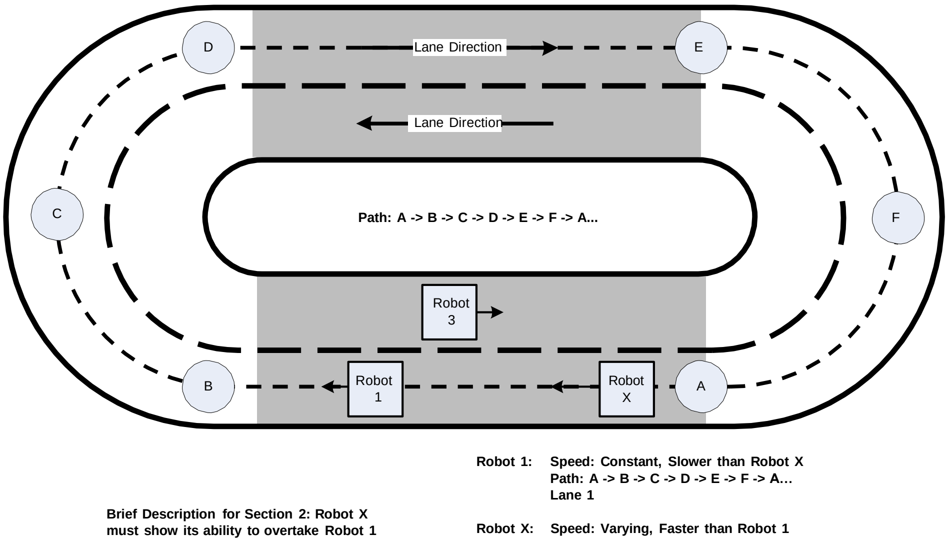
2. “Robot X” from Section 1B now takes the role of “Robot 2” from section 1A. This “Robot X” should also go through the waypoints like “Robot 1”

3. Program “Robot X” to overtake “Robot 1” when they are in a particular region of the closed circuit. That particular region must be a straight road. After “Robot X” overtakes “Robot 1” and returns in its own lane, “Robot X” will continue on its path at maximum speed until it is once again behind “Robot 1”. The cycle of overtaking then repeats itself. Do not forget that “Robot X” must reduce its speed to that of “Robot 1” if there is a risk of collision with “Robot 3”, and only overtakes when the way is clear at the designated region of overtaking

4. You may now improve the visual aspect of the virtual environment for your project presentation and demonstration.

This section is a combination of section 1A and section 1B, with some improvement in the visual aspect of the virtual environment. While not exactly the same, you can imagine a Formula One (F1) race, where cars have to go through several laps and overtaking the slower cars. The only difference is that there are no cars going in the opposite direction in F1 races.

Figure 3 shows a possible scenario for this section, with the shaded region indicating the region where “Robot X” is allowed to overtake “Robot 1”. You are reminded to use a closed circuit and shaded region that matches your own needs and capability



Robot 1: Speed: Constant, Slower than Robot X
Path: A -> B -> C -> D -> E -> F -> A...
Lane 1

Brief Description for Section 2: Robot X must show its ability to overtake Robot 1 when it is in the shaded region. The shaded region consists of straight roads. There must not be any collisions involved

Robot X: Speed: Varying, Faster than Robot 1
Path: A -> B -> C -> D -> E -> F -> A...
Lane 1

Robot 3: Speed: Constant, Arbitrary
Path: Reverse of Path Taken by Robot 1 and Robot X
Lane 2

Figure 3: Possible Scenarios for Section 2

Section 3: Advanced Vehicle Overtaking on a Closed Circuit

This section is an extension of Section 2 and only needs to be done if you are confident with Section 2. Section 3 consists of more complex tasks and marks given here will form the bonus component of the judging criteria. You are required to indicate the main contributor for each task of this section during presentation. Please note that some parts in this section might require a good computer, like those five reserved computers in the Control and Simulations Lab (E4A Level 3). You might also need to modify your virtual world environment for these tasks.

Note that marks will only be awarded in this section if the starting position of the intelligent robots can arbitrary within their lanes, yet still able to complete the tasks. This is to prevent students from hard coding the robot to do particular actions after a set amount of time.

The tasks could include:

- ❖ Overtaking is smooth, without the need for the robot to stop or make sharp turn
- ❖ The robots that are not capable of overtaking vary their speed significantly as they go around the closed circuits. “Robot X” must be able to overtake these robots that have varying speed
- ❖ In the event “Robot 1” changes its speed to or near maximum, “Robot X” must not try to overtake “Robot 1”
- ❖ In the event “Robot 1” changes its speed to or near maximum while “Robot X” is in the middle of the overtaking process, “Robot X” must stop the overtaking process and allow “Robot 1” to stay in front. “Robot X” will then return to its lane and keep following “Robot 1” until the next overtaking opportunity arises
- ❖ In each lane, there is one robot not able to overtake. Also, in each lane, there is one robot able to overtake. That is, there are four robots in total on the closed circuit, and one robot able to overtake in each lane. You are required to show that the two intelligent robots that are able to overtake, are able to anticipate each other’s overtaking action and therefore avoid collision
- ❖ Instead of overtaking just one robot, there might be several robots in a line that need to be overtaken. This is similar to having a robot that is longer in length as compared to the other robots. In the real world, it could be similar to the act of overtaking several closely packed cars in one go
- ❖ Suppose there is another entity crossing the road, that is, it is moving perpendicularly to the lanes on the circuit. Any robots must stop before colliding with the crossing entity and cannot overtake.Once the road is clear again, the robots can continue normally on their track and overtake when required
- ❖ Any improvement related to car navigation in urban environments that can demonstrate your ability to creatively or intelligently program autonomous robotic systems. This can include creative use of sensors or creative situations

JUDGING CRITERIA

This project will form 75% of the marks for EE4306 Distributed Autonomous Robotic Systems. The 75% is made up of:

Component	Percentage of Module
Milestone 1	15
Milestone 2	15
Milestone 3	45
Total	**75**

Bonus	15

FINAL DEMONSTRATION AND PRESENTATION (MILESTONE 3)

You will be assessed on whether the tasks could be accomplished without significant issues. The suitability of the visual environment used and the complexity involved in solving the task are considered.

This component also looks at how easy it is to understand your work, and whether you are able to emphasise on parts of the simulations which you think are worth special mention. You may use presentation slides to briefly introduce these special parts of your simulation before running the simulation. Your ability to comment on what the mobile robot is doing while the simulation is running, and at the right time, will also be considered.

A maximum of 10 PowerPoint slides is recommended. Please focus more on live demonstrations, videos and recordings. Live demonstrations of your program is preferred if you have enough time for them. Appropriate commenting and/or relevant presentation slides during demonstration is required.

Non-productive waiting time for the robot to complete particular tasks must be avoided. Immediately after your presentation, other students from the class will be allowed to ask questions about your simulation. There is no need for all team members to present and you may choose the one who is most comfortable and fluent at presenting. You will be given a maximum of 15 minutes for the presentation. This includes the preparation time, and the questions and answers part.

REPORT (MILESTONE 3)

The report should have about 5 to 15 pages, with font size of about 10 to 12. Please include your team number, team member names and team member matriculation numbers on the cover page. There is no strict format and content for the report. You may include:

(1) Brief indication of the tasks involved
(2) Approaches and logics used to solve the tasks (If/Else, Fuzzy, GA, State machine)
(3) Robots/Entities and sensors used, and a brief description about them if needed
(4) Algorithms used to tackle the tasks
(5) Flowcharts or dataflow diagrams for the algorithms or tasks
(6) Analysis of the results and how real world scenarios might differ
(7) Literature or program references used
(8) Descriptive screenshots of the simulated visual environment if applicable

You are encouraged to add other details that demonstrate your effort and understanding of this project.

BONUS

In this project, the bonus section (Section 3) is generally hard and complex. Depending on how many of the teams will be able to complete some of the tasks within that bonus section, bonus marks will be awarded. The more unique your solution is among the whole class, the higher the bonus marks will be. This 15% of the bonus component is on top of the 75% for this project. However, the maximum mark for this project is still 75% of the EE4306 module.

PROJECT DEADLINES

MILESTONE 1: Week 7
MILESTONE 2: Week 9
MILESTONE 3: Week 12

UPLOAD:
Each team will need to upload a softcopy of materials related to their project in one zipped file. The zip file usually consists of:

(1) Report in PDF format
(2) SPL Program used for presentation
(3) Additional files / images used for the program to work
(4) Instructions on how to run your program
(5) Video recordings (.avi, .mpg) or MRDS recorded playback (.plb)
(6) Presentation slides used if available
(7) Any other related materials

The folder to upload to on EE4306 IVLE is: "(2013) MRDS TEAM SUBMISSION". Only one zipped file must be uploaded and only one member from each team needs to upload the file. The name of the zipped file should be in the format: teamXX.zip where XX is the number assigned to your team. The deadline for upload is on the day of your final demonstration.

HELPFUL LINKS:

SPL available at:
http://www.helloapps.com