

Research in Robotics

By Pranav A. Bhounsule, pranav@uic.edu

Outcomes of Research

At the end of this research, students will be able to:

1. Learn basic usage of the robotics simulator, CoppeliaSim
<https://www.coppeliarobotics.com/>
2. Create models of robotic systems
3. Write Lua code to control robots
4. Evaluate how to use sensors and actuators
5. Program a robot(s) to achieve a desired task.

Equipment list and some resources

1. Please download and install the education version of CoppeliaSim (it is free):
<https://www.coppeliarobotics.com/downloads>.
2. We will use the programming Language Lua to program the robots. Here is one resource to learn Lua. <https://manual.coppeliarobotics.com/en/luaCrashCourse.htm>. Please feel free to search the internet for other resources.
3. Please see (and bookmark) the regular API. This provides the list of functions that you can use to program robots. <https://manual.coppeliarobotics.com/en/apiFunctions.htm>.
4. There is an active CoppeliaSim forum where you can ask and get answers.
<https://forum.coppeliarobotics.com/>.
5. There are many examples and videos on CoppeliaSim projects on YouTube and Github. Also, I have video tutorials with code available using different versions of CoppeliaSim. See here <https://tiny.cc/educate>.

1 Part 1: Introduction to CoppeliaSim

1.1 Basics of CoppeliaSim (1 week)

1. Create, modify, and move shapes <https://youtu.be/qrchIEqt2ig>
2. Change the viewpoint (pan/rotate/zoom/fit-to-view) <https://youtu.be/98VnjXg1wBc>
3. Create composite shapes https://youtu.be/v_NuNIdAo_U
4. Create a simulation https://youtu.be/H_Li3QeVM-M

Exercise: Your goal is to create a cascading effect like that in Rube Goldberg machine using geometries available in CoppeliaSim. You can also arrange by different orientation/position/size/shape to create more interesting behaviors. Use at least 4 objects. Here is one example from YouTube.

<https://youtu.be/OHwDf8njVfo?t=85>

Send your .ttt file to pranav@uic.edu

1.2 Modeling and controlling a differential drive car (1 week)

1. Modeling a differential drive car <https://youtu.be/uoL4J9QDZK0>

2. Controlling a differential drive car https://youtu.be/IW1-4hy_yEg

Exercise: Get the car to move in a rectangular path by controlling the speed of the wheels.

Help: If you find it difficult to model the car, then you can download the model from here and proceed to control https://github.com/pab47/CoppeliaSim/blob/main/430/differential_drive.ttt

Send your .ttt file to pranav@uic.edu

1.3 Line Following Robot (1 week)

Download the Line Following Robot

https://github.com/pab47/CoppeliaSim/blob/main/430/line_follower_1_speed.ttt

1. Understanding how to access the vision sensors <https://youtu.be/pNJh83cp1lY>

2. Algorithm for line following <https://youtu.be/jduuJwK8uME>

Exercise: Download this scene and copy paste the code you developed.

https://github.com/pab47/CoppeliaSim/blob/main/430/line_follower_exercise.ttt

Tune the controller to complete the loop in the fastest time.

Send your .ttt file to pranav@uic.edu

1.4 Wall following robot (1 week)

Download the wall following robot here

https://github.com/pab47/CoppeliaSim/blob/main/430/wall_follower.ttt

1. Understanding how to access the proximity sensors <https://youtu.be/iD2Dc6r7PeQ>

2. Algorithm for wall following (feel free to develop your own too). <https://youtu.be/oxQeojd7RTM>

Exercise: Download this scene and copy paste the code you developed for wall following robot.

https://github.com/pab47/CoppeliaSim/blob/main/430/maze_solver_exercise.ttt.

Tune the controller to move from start to goal in the fastest time without colliding with the walls.

Here is an example of the finished maze navigation: <https://youtu.be/NZeG0vLbzJ0>.

Send your .ttt file to pranav@uic.edu

2 Part 2: Project in CoppeliaSim (2 weeks)

Your goal is to use the knowledge you have gained to make an informed choice for a robotic project. Then, you will implement the project in CoppeliaSim and create a short presentation explaining your work.

Here is a list of possible robot projects based on mobile cars. But feel free to choose something else. Search YouTube and the internet for more ideas.

1. A robot that navigates through a maze
2. A robot that avoids obstacles and drives to the goal.
3. A robot that pushes objects out of a marked area. The robot detects objects using proximity sensors. (HINT: search for sumo bot)
4. Create a video game using a mobile robot. This video shows how to program keyboard press <https://youtu.be/dd0Rw3HiAes?si=R6tMWsPtbtBVLhni>.
5. A robot that parallel parks using vision sensors.
6. A robot that drives on a road using vision sensors.

Send your .ttt file and presentation to pranav@uic.edu