CAIC TROPHY

TECH GC 2025

INTERACTIVE ROBOTICS

Robotics and Dev

Introduction & Motivation

In this PS, you will dive into the world of robotics and simulation, with the goal of developing a fully functional control system for robotic arm in CoppeliaSim. The challenge involves not only simulating these robots but also controlling their movements and interactions with the environment through external tools. You will also create a user interface that allows others to interact with and control the robot at different levels of abstraction.

Problem Statement

1. **Robot Simulation**

   In this section, you are required to choose a robotic arm from the prebuilt robots available in CoppeliaSim. Alternatively, you may design your own custom robotic arm from scratch. The following functionalities are required:

   a. **Joint Control:**

      I. You must be able to control the robot's joint angles with a precision of at least 10% of its total range of motion.

      II. For joints with discrete states (e.g [open/close], [high/low]), there are no precision requirements.

   b. **End-Effector Control:**

      I. You should be able to control the position of the end effector with a precision of 1cm. Manipulating the end effector to move in predefined paths will fetch additional points.

c. Interaction with the Environment:

Your robot should interact meaningfully with the environment:

I. Manipulate the orientation and position of regular objects such as cuboids, spheres, cylinders, and irregularly shaped objects (e.g., cups, saucers, pots, and bowls). You can create and import your own objects in simulation. There is no restriction on the dimensions of the objects, but it should be somewhat to scale

II. Stacking of both regular and irregular objects. Stacking irregular objects will fetch higher points.

2. Developing the User Interface

In this section, you will create a user interface at multiple levels of abstraction, each of which should be demonstrated separately. You are free to choose your preferred programming language to implement these interfaces.

a. Terminal-based Control:

I. The robot's control commands should be sent via an external script, allowing the user to interact with the robot through a terminal interface.

II. Your system should handle illegal or out-of-bound commands gracefully, providing informative error messages to guide the user.

b. Graphical User Interface (GUI):

i. Build a graphical user interface to control the robot. This could be a native desktop application or a web-based interface.
ii. If you choose a web interface, we recommend using React for building the UI.
iii. The GUI should support all robot control features, including movement and interaction with the environment. Additionally, ensure that the system can handle illegal inputs appropriately and alert the user to any mistakes.
iv. Also note that the GUI should run on a different laptop/computer.

c. Voice-Controlled Interface:

i. Implement a voice-controlled interface, allowing the user to control the robot using spoken commands.
ii. The system should not only accept voice commands but also provide voice feedback to the user, confirming actions or reporting errors.
iii. You are encouraged to use open source LLMs for this section.
iv. Ensure that illegal or out-of-bound voice commands are handled correctly and provide clear auditory feedback to guide the user.
v. The voice interface should run on a different laptop/computer.

Solution Deliverables

1. We expect the following in a zipped folder with your Team ID.
    1. A report as per the guidelines provided by CAIC.
    2. Readme file containing the steps to start the simulation on a fresh system.
    3. In a folder named *'src'* : All the code/script/config files required to start the Simulation/Controller.
    4. In a folder named *'Visuals'* : Images and Videos demonstration of each section implemented.
2. A presentation will be organized to showcase your solutions.

Evaluation Parameters

1. Robot Simulation

Section	Max Points
a. Joint control 1. U => joints/degrees of Freedom that was controlled	3U
b. End-Effector Control 1. V => number of predefined paths the end effector can move.	9 + 11*V
c. Interaction with the Environment 1. W => number of regular objects manipulated. 2. X => number of irregular objects manipulated. 3. Y => number of regular objects stacked. 4. Z => number of irregular objects stacked.	13*W + 17*X + 21*Y + 23*Z

V, W, X, Y, Z ∈ {0,1,2,3,4}, and U ∈ {0,1,2,3,4,5,6,7,8,9} all Integers

Total Score for Robotics: **R** = 3*U + (9+11*V) + (13*W + 17*X + 21*Y + 23*Z)

2. **Developing the User Interface**

Section	Max Points
a. Terminal-based Control:	5
b. Graphical User Interface (GUI):	20
c. Voice-Controlled Interface: 1. A => multiplier in case LLM is integrated	30 + 50A

A ∈ [0,1], all Real

Total Score for Dev: **D** = 5 + 20 + (30+50A)

3. **Total Score for Problem Statement: R*D**

Team Size

Each team can have a maximum of 6 members

Resources

1. Coppeliasim: https://www.coppeliarobotics.com/
2. React: https://react.dev/