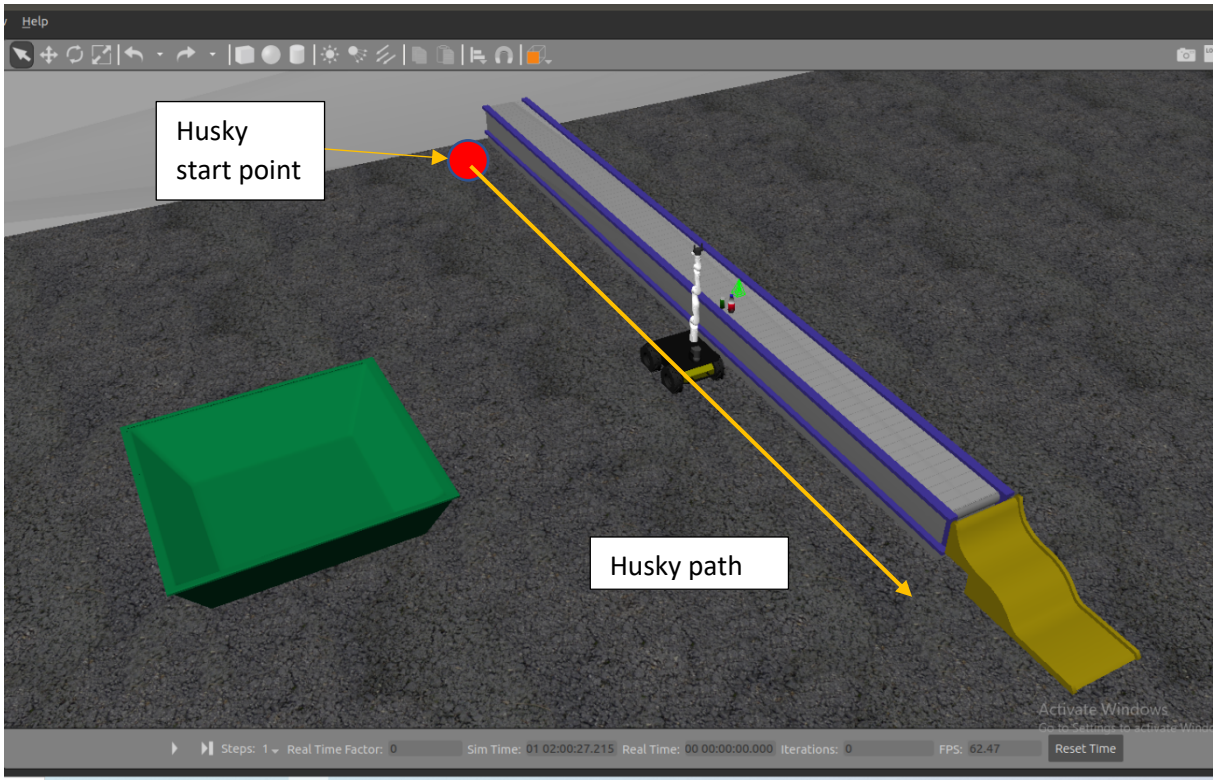
EE4002 ROBOTIC SYSTEM DESIGN CAPSULE PROJECT

Assigned: May 1.

Deadline: June 15, 23:59.

ROS and Gazebo Installation with MATLAB Simulink

- Ensure that you have
  - 64-bit Windows 10
  - MATLAB version at least R2021b (check if previous MATLAB versions suffice)
- Start with downloading a virtual machine including Ubuntu 18.04 Bionic and ROS2 network following the instructions in the link:
  https://www.mathworks.com/help/ros/ug/get-started-with-gazebo-and-a-simulated-turtlebot.html
  https://www.mathworks.com/support/product/robotics/ros2-vm-installation-instructions-v5.html
- After the installation, open the "Mobile Manipulator World.sh" file in the VM desktop and bring up the Husky with Kinova arm in the warehouse environment.
- Setup the environment as seen in the following figure and remove the other objects.
- Set properly the IP addresses of the host machine and the VM.



Your aim is to:

- [Phase 1] Start the Husky robot from the indicated location above, drive the Husky robot alongside the conveyor belt through the yellow line, and find the objects on the conveyor belt with sensors mounted on the manipulator while moving the Husky;
- [Phase 2] grasp an object with the manipulator;
- [Phase 3] transport the object toward the green bin;

- [Phase 4] place the object into the bin.

Instructions:

- You must use the 2D LIDAR sensor to control the Husky motion in Phase 1.
- You may put any object on the belt. Initially, some bottles and cans are located in the simulation. Instead, you can put a small cubical box with certain color.
- In Phase 1, you can use a proximity sensor or camera to detect the box while Husky is in motion. Once the object is detected, Husky should position itself to a configuration where it can manipulate the object easily near the belt.
- There is a camera attached on the gripper of the manipulator. In Phase 2, you can use computer vision techniques to align your manipulator gripper on top of the object. You may use a color detection algorithm or the embedded OpenCV functions to compute precisely the object location. You are expected to implement the forward/inverse kinematics solutions in this phase.
- In Phase 3, assume that the green bin location and dimensions are known exactly. *Bonus: If you use a localization algorithm for the Husky when moving the Husky toward the green bin, you will be awarded extra points. In that case, you need to use the estimated Husky position in the Husky controller.*
- You can use pre-defined ROS packages to move your manipulator. You are recommended to plan the trajectory of the manipulator.
- You may use the MATLAB-ROS connection and write your code in MATLAB, or completely use Ubuntu python scripts.

You are expected to explain your state machine, workflow, and code clearly and coherently so that your system can be reproducible by the instructors. In a live demo session, the object position on the conveyor belt will be randomly selected by your instructor, and you will be asked to implement your code.

You may analyze the code in the following page and extract some useful code/idea for your implementation:

https://www.mathworks.com/help/robotics/ug/design-and-simulate-a-warehouse-pick-and-place-application-using-a-mobile-manipulator.html

Please submit your **code package** folder and explanatory **report** (including all analytical derivations, figures, etc.) together with a short **video** of your simulation showing the robot's operation. Your code should include the necessary explanations as comments. Please take special care when writing comments in your code as the readability of your code will be graded.