Documentation

Description of the task:

Develop an appropriate system including ROS/Python that will prevent operation of a linear actuator if enough color "blue" is detected on an RGB video camera. The delivery should include CAD for the complete system (actuator, camera and anything else relevant). Please carefully review the job description and make sure you are showing your work at your best. Think about the use cases, how we'll read your code, your architecture, and how your solution might break. A single code file is not likely to be sufficient. You can of course make assumptions/simplifications, but state them clearly.

Assumptions:

1. enough color "blue" is detected on an RGB video camera

Assumption :

1. Segmentation is more than enough to detect an object
2. Enough color blue detected means the object is too close.

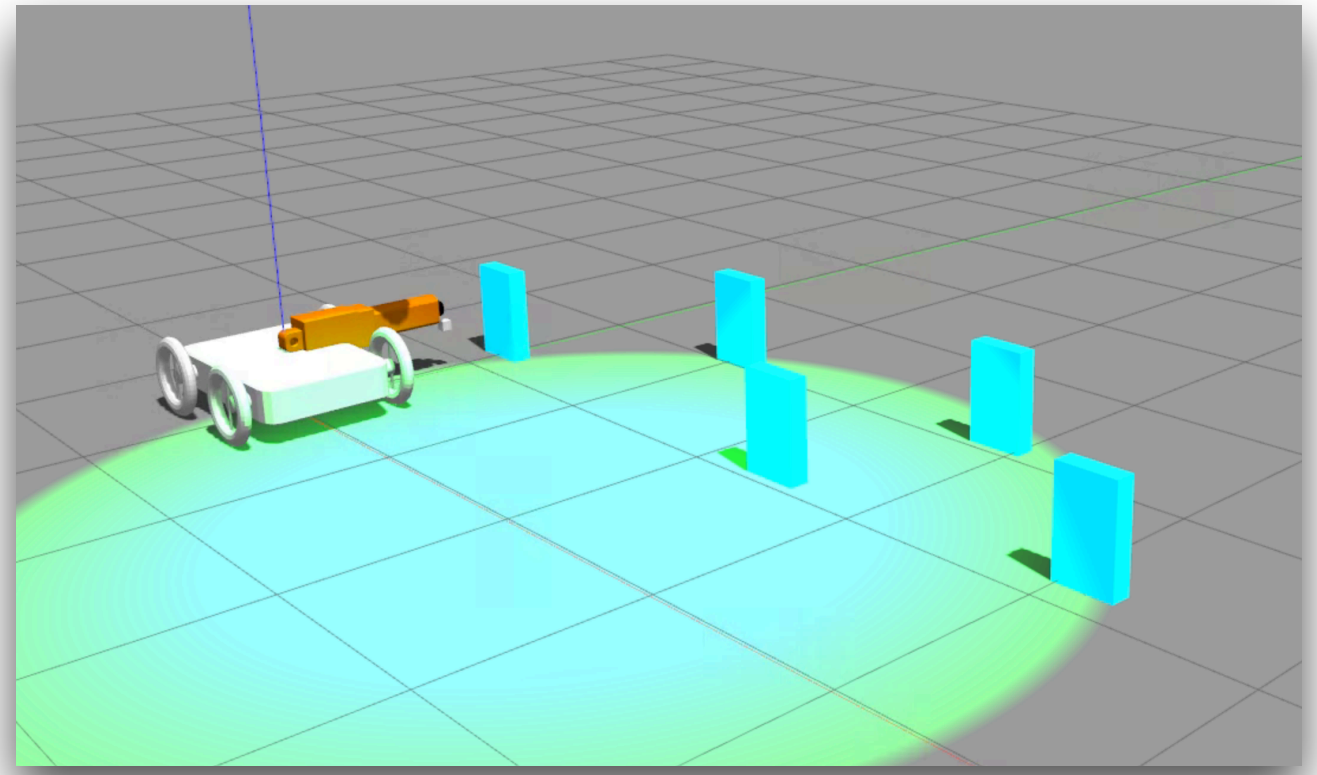
2. prevent operation of a linear actuator

Assumption:

1. Actuator is not in the middle of performing a task
2. Actuator will not be Deployed if too close (enough blue detected)

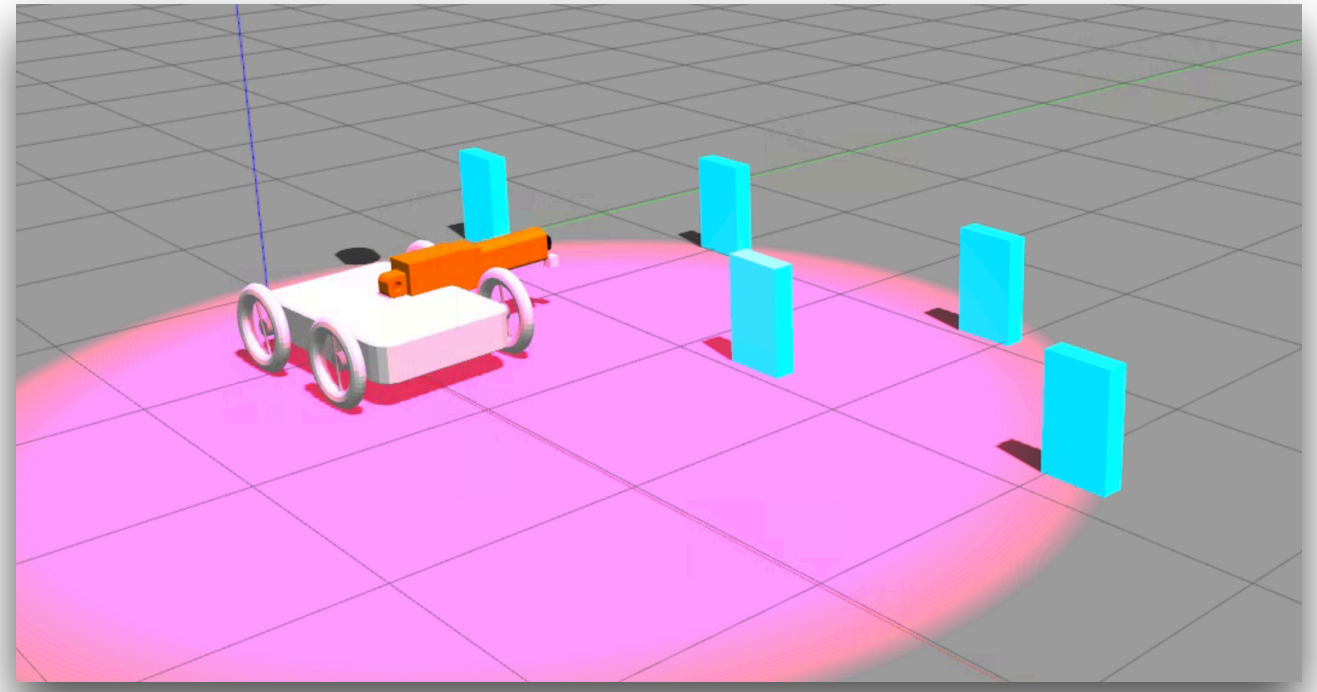
Design:

A mobile robot called “poker” is trying to poke blue blocks.

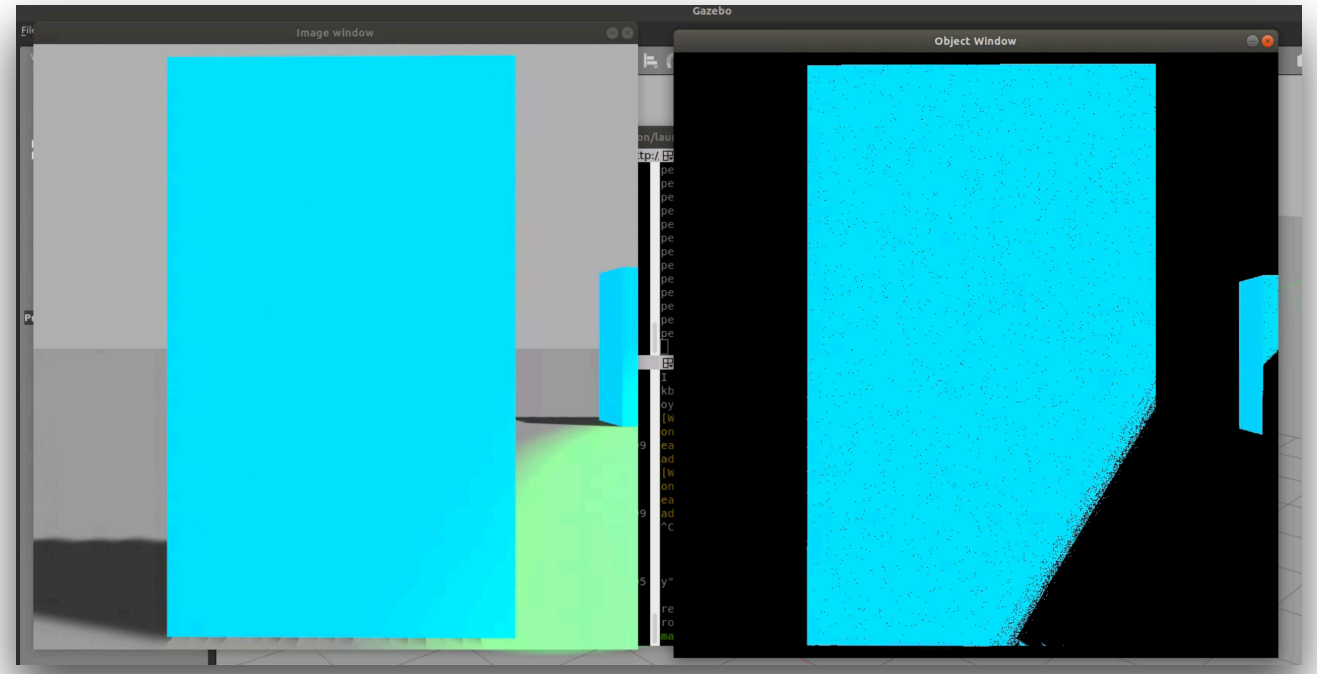


The camera is mounted underneath the linear actuator. When the object is close enough (enough color blue is detected), a linear actuator can be deployed to poke the object. The green light shining indicates that the object is close enough.

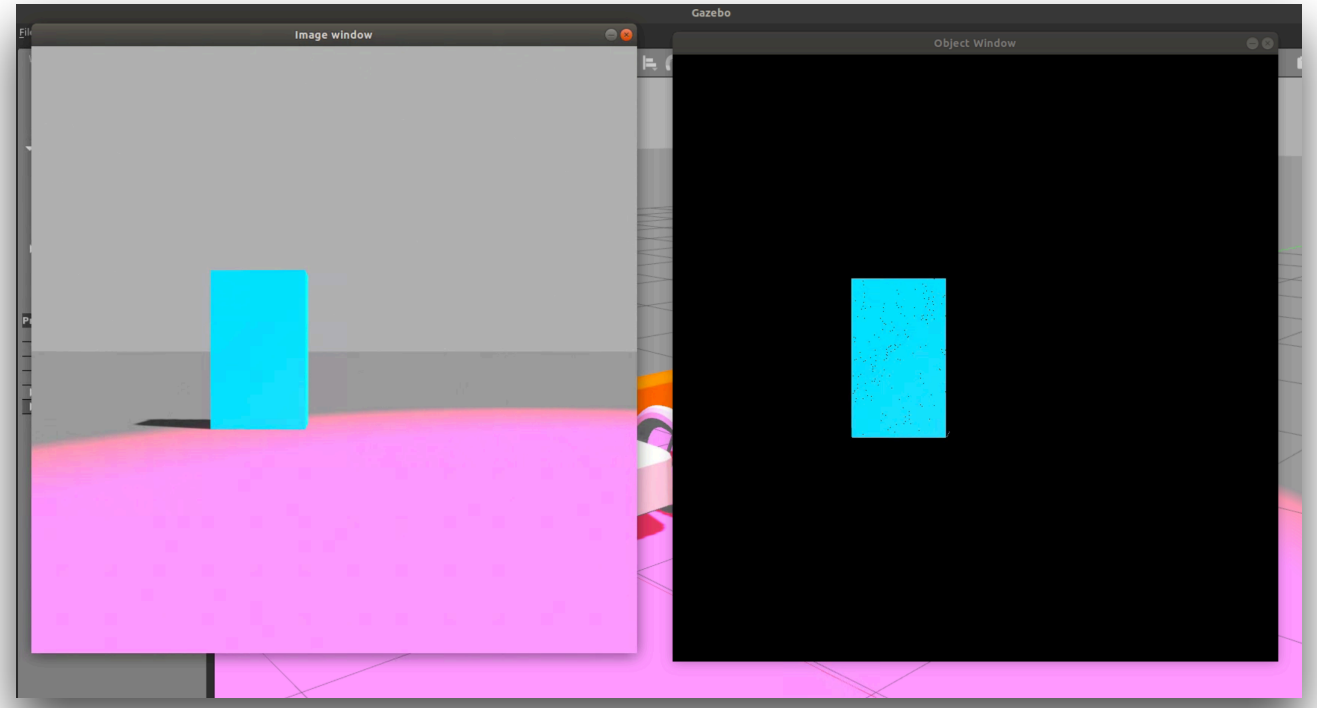
When the object is far (not enough blue detected) the linear actuator can not be deployed. We can also see the light has turned red, indicating that the linear actuator can not be deployed.



viewer.py located in “Xihelm_ws/src/detector/scripts” performs the computer vision task. Figure bellow shows when the object is close enough.



In contrast, this figure shows when the object is too far.



viewer.py calculate the percentage of blue detected and publishes it to the topic "'/light_control'". This topic is used for gazebo visualisation. In addition, the same topic is used by mission_control.py to prevent deployment of the linear arm if the object is too far. mission_control.py is located at "Xihelm_ws/src/control/scripts".

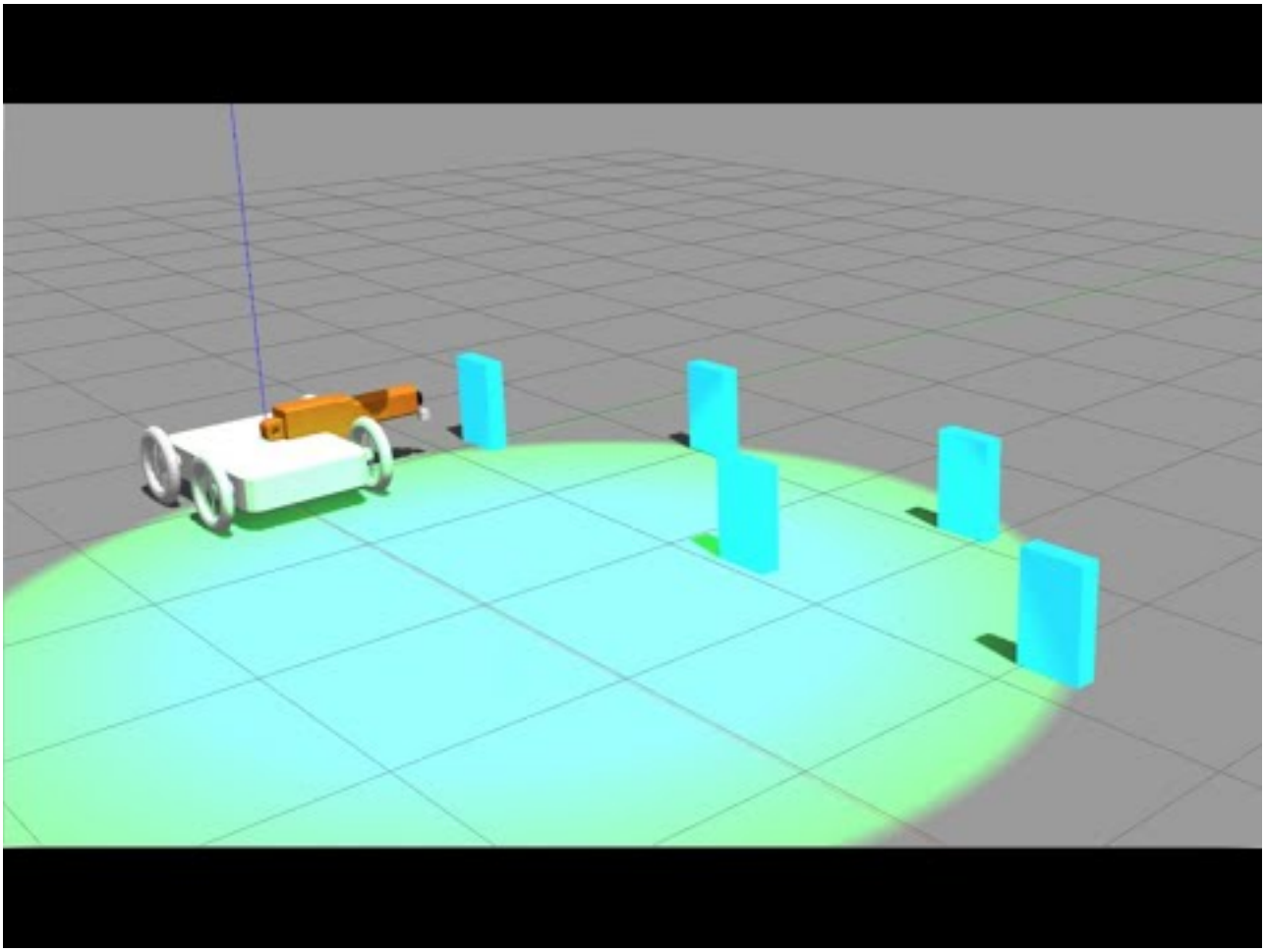
Finally, wheeler.py, located at "Xihelm_ws/src/control/scripts", handles the robot movement.

mission_control.py is an action client and wheeler.py is an action server.

The robot can be moved by increasing it's linear velocity by using up and down arrow keys. The linear actuator can be deployed using the space key. However, if the object is not close enough, it will not be deployed (red light would be shining).

Demo:

Here is a YouTube video of poker in action:



Installation:

Please install dependencies by running “requirement_install.sh”

Requirement:

You would need ROS noetic.

Feel free to contact me if you need any help or have any questions :)