CE3890/CE5999, CS3891/CS5891 AV&T

Homework 3: From Simulink to Ros

Due: 09/24/2024

Instructors: J. Sprinkle & D. Work *Fall 2024*

Release Notes:

Version 0: Initial release

0 Assignment overview

In this homework, you will transition your solution to Homework 2 to run in Ros. You will provide a video demonstrating your system's behavior.

Answers to the following questions should be typed in a suitable editor (e.g., Word or LaTex) with appropriate references (if you used them), figures, and text, as needed to explain your results. You are free to use your favorite data analysis tools (python, Matlab, C++, an abacus, etc.; just use what you are comfortable with and show your work).

Your submission should include a zip of the following:

1. Your MATLAB/Simulink file for the Ros version of your controller must be included with your submission
2. The generated .tgz file from Simulink.
3. The output .bag file that is saved when running your SWIL in Docker.
4. A (potentially edited) video of no more than 5 minutes, showing your system running from generation of code, through execution on Docker, and quick view of the analysis.
5. A brief report of no more than 2 pages.

1 GOTO Ros

Using the provided Docker and Simulink tutorials and materials, demonstrate how your Homework 2 can be transitioned from Simulink to Ros.

Question 1.1 Create Homework 2 in a Simulink-based Ros model Create a new Simulink model with an appropriate name (e.g, homework3ros.slx). Paste the controller that you used for the ego car in Homework 2 into this model, and replace the data lines from Simulink to be Subscriber and Publisher nodes of the appropriate types. Refer to course materials for data types and topic names, as well as example video tutorials. A screenshot of your model should be included in your brief report, which clearly shows the topic names and types as evidenced in the models.

Question 1.2 Generate the code Modify your Simulink model to generate code for the Robot Operating System (not ROS2). Include your output .tgz file in your submission.

Question 1.3 Run the code in Ros through Software In The Loop (SWIL), using Docker Using or modifying the Dockerfile provided in the tutorial, run the simulation in Ros to play the provided bagfile, and capture a new bagfile with the system outputs.

Question 1.4 Video execution Record and narrate a screen recording your Dockerfile executing your output code and producing the bagfile. Edit the video to be no more than 5 minutes with a narration that describes your process, and any key points. This is key for your future memory of how you can carry out these tasks repeatedly.

Question 1.5 Brief report Compile a report of no more than 2 pages that shows plots with descriptions of key results from analysis of your bagfile. Some discussion should include a comparison of the results to what you saw from your Homework 2 submission of the same controller.

Resources available for your submission:

1. `subtractor` Ros node github repository: `https://github.com/jmscslgroup/subtractor`
2. `carsimplesimulink` Ros node github repository: `https://github.com/jmscslgroup/carsimplesimulink`
3. `profproject` Ros node github repository: `https://github.com/jmscslgroup/profproject`
4. `cs3891project2023` launch files to use within Docker: `https://github.com/jmscslgroup/cs3891proj2023`
5. In `cs3891project2023`, look at the different launchfiles as examples for how to run your Ros network, especially `profprojecttest.launch`, which will (upon launching the Ros network) play a bagfile, and record a *new* bagfile with the prefix `profprojecttest`. *Optional:* mildly modify these parameters and make your own launchfile, to reflect your name/project information as the filename.

An example Dockerfile to bootstrap your homework is given on the next page. Using this Dockerfile, you should clone the ros repositories into the folder where you plan to run it, and also include the mybag.bag file there.

```
# to run, use
# docker build -t cs3891:test .
# docker run --mount type=bind,source=.,target=/ros/catkin_ws -it cs3891:test

# syntax=docker/dockerfile:1
FROM ros:noetic-robot

# change the default shell command
SHELL ["/bin/bash", "-c"]

# this gets run in this image
RUN source ros_entrypoint.sh

# now add this to all future calls
RUN echo "source /ros_entrypoint.sh" >> /etc/bash.bashrc

# clone the packages we need
# get the git package
RUN apt-get -y update
RUN apt-get -y install git

# change to our working directory
WORKDIR /ros/catkin_ws/

# RUN source /ros_entrypoint.sh && catkin_make

# you only get one command
# CMD roscore
```