Chapter 7: Implementation of a Manual Mode to Control the Ego Vehicle with a Keyboard

This chapter deals with an implementation of a manual mode so that an ego vehicle can be driven via a keyboard in RViz environment. This guide will provide the readers a detailed approach on how to control the vehicle from a keyboard. However, it should be noted that some initial commands that were run on the terminal are very similar to the ones explained in the previous chapter (Chapter 6), and they will not be repeated here to avoid repetition, but a reference to the figure number and step number will certainly be provided.

Step 1: We will launch the Autoware runtime manager from the terminal. Please refer to ***Step1*** in chapter 6 for the detailed information on how to launch it.

Step 2: After the Autoware Runtime manager window pops up on the screen, under the “Setup” tab, Vehicle Model will be selected. The path to the file for vehicle model can also be seen in the figure 30. Vehicle Model button can be clicked to select it.

Step 3: Next step is to setup the “Map” tab. The file paths used for Vector Map and TF can be seen on figure 31.

Step 4: Now we will setup the “computing” tab. Firstly, the *vel_pose_connect* app will be selected. Please, make sure that the simulation mode is selected as below.

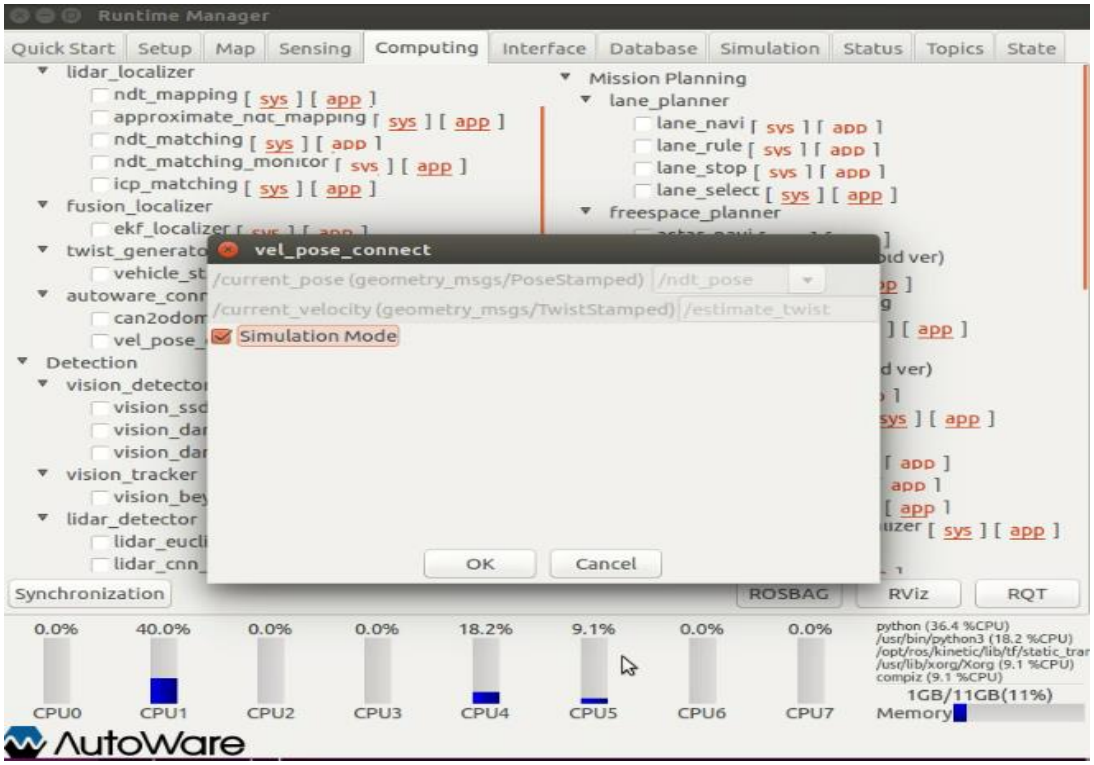

Figure 1: A screenshot of vel_pose_connect app

Step 5: Please turn on the wf_simulator and then click on RViz to open it.

Step 6: After launching RViz, set the 2D Pose Estimate, and the car will appear. After doing this, change the camera view to *TopDownOrtho* and target frame to *sim_base_link* located at the top right of the window.

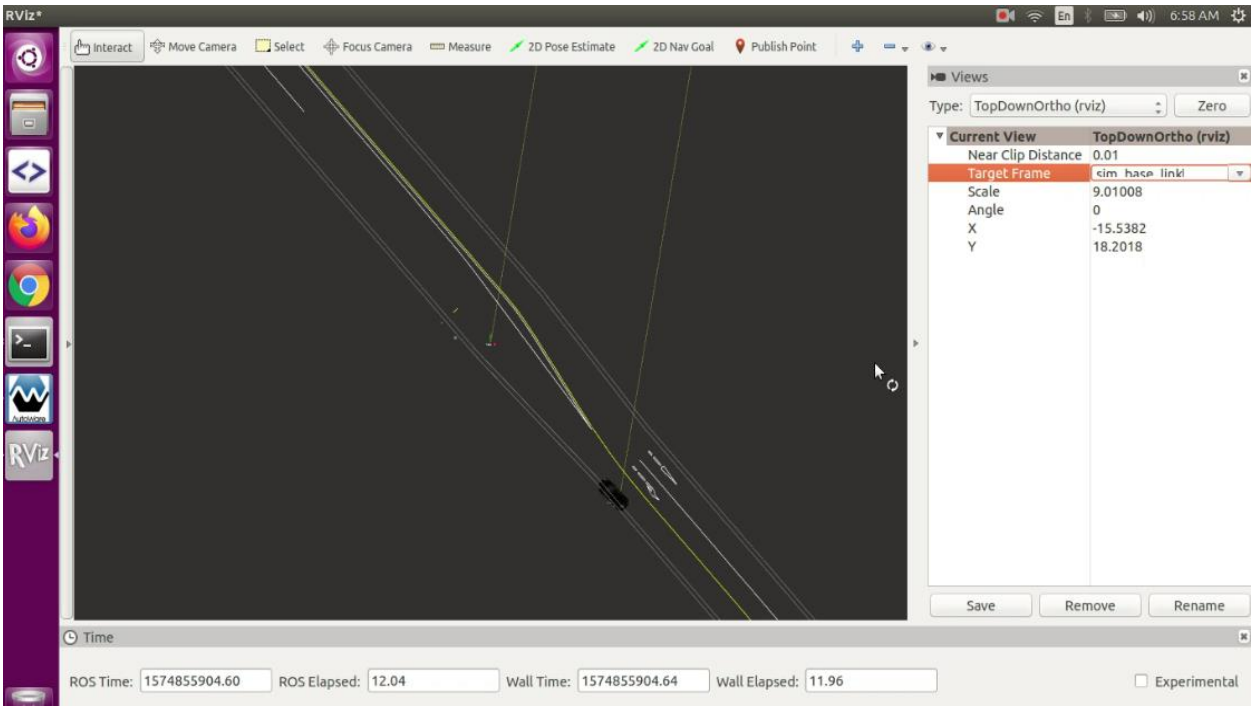

Figure 2: Initial RViz window

Step 7: Run MATLAB from the terminal and change the path as shown below to locate the folder MATLAB Adds-Ons. For instance, in our case this looks like this: /home/phuris/Desktop/MATLAB Add-Ons/Collections/Simulink Keyboard Input v2

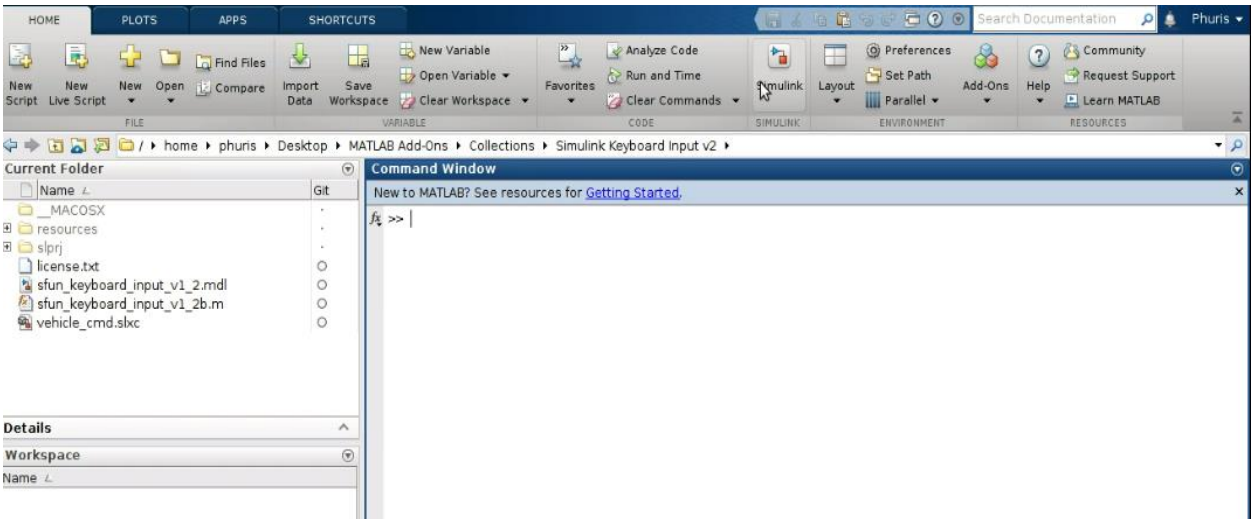

Figure 3: Locating the folder on MATLAB

Step 8: Run the following command in MATLAB to initialize the ROS node.
>> rosinit
Now, launch Simulink.
After running the model, please make sure that Keyboard Window is shown up.

Step 9: Open *vehicle_cmd.slx* and run it

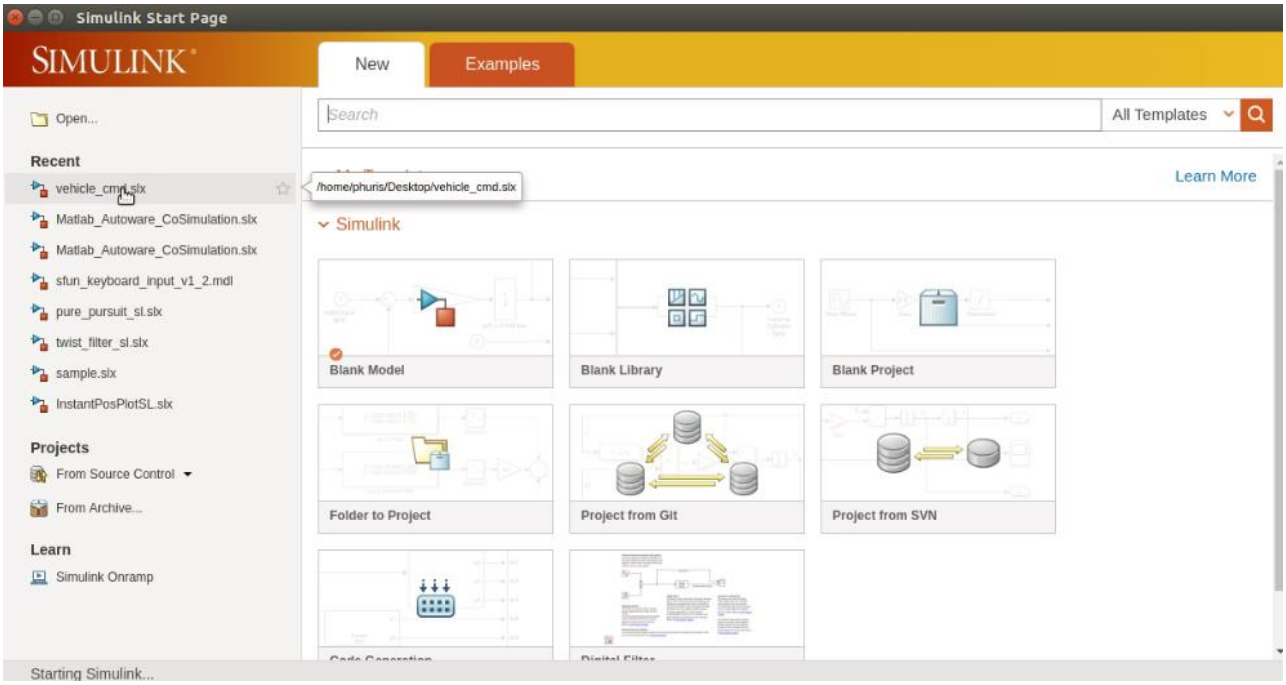

Figure 4: vehicle_cmd.slx

Step 10: Now, we are ready to control the car via keyboard.

- press 'up-arrow' to accelerate
- press 'down-arrow' to decelerate
- press 'left-arrow' to steer left
- press 'right-arrow' to steer right
- press 'spacebar' to stop turning the car

Some remarks on how to use the model:

- In the model, there exists two gain blocks for left steering angle and right steering angle which can be adjusted so that the response can be quicker.

- In addition to steering, there exists two gain blocks that determines until which velocity the vehicle should accelerate and decelerate. Those values are currently set to 60 km/h and 0 km/h respectively.

- Key control can be thought of as a switch. For instance, when up-arrow key is pressed, the vehicle will accelerate continuously even though key is not pressed anymore. Same condition applies to the left and right steering of the vehicle as well.

The video demonstration is found in our Git repository for the project and the link is provided below.

- https://gitlab.lrz.de/ge73xoh/software-lab---autoware.git

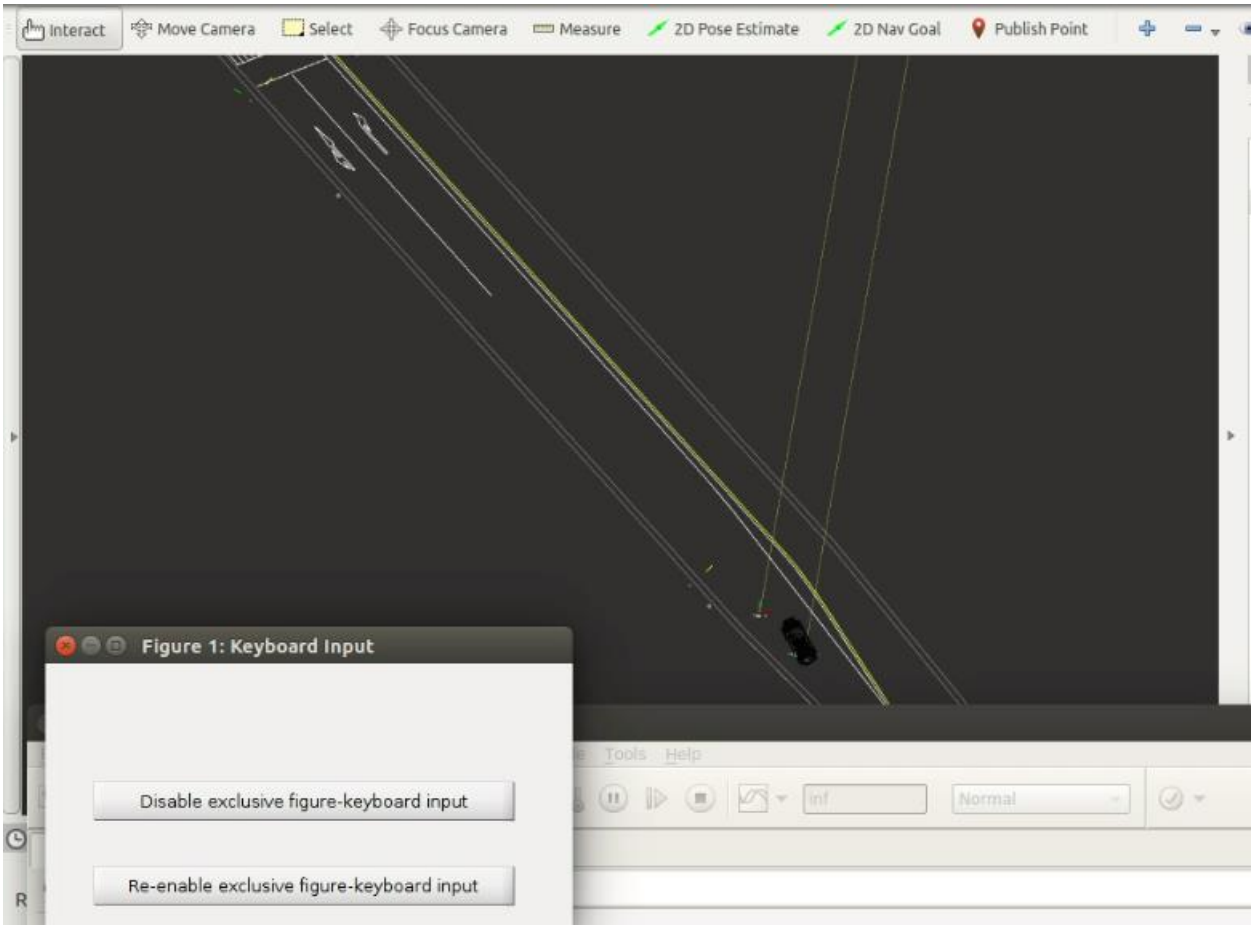

Figure 5: Keyboard Control