Motion Controller and Odometry Interface for the RCV

Lars Svensson & Rui Oliveira

February 2016

Develop a useful and easy to use interface that will allow the RCV to be used reliably and easily through ROS. A simple diagram of the expected outcome is as follows:

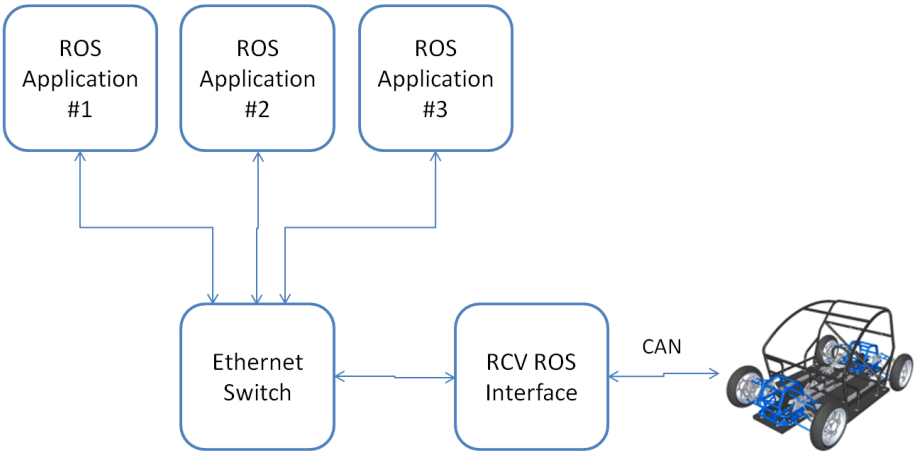

Figure 1: System Structure Draft

There are many aspects that can be developed in the RCV, however it is necessary to choose only those which are required for the correct usage of the RCV by the other parties interested. Several functionalities will be described in this proposal, and they will be divided into Mandatory and Optional functionalities. The division made is based on a first thought of the authors on what might be, or not, required from the other students. It is not a final decision, as it might still be changed upon request from the other course participants.

An important point to make is that the Velodyne sensors are not a concern of this work. They should be easy to interface with by making use of Andreas' work.

1 Mandatory

1.1 Path Following Controller

The Path Following Controller will be able to receive path requests and will be responsible for following them. In order for proper functioning of the controller, these paths need to be sent to the controller at a refresh rate above 10 Hz, and they must be defined on the car local frame (figure 2).

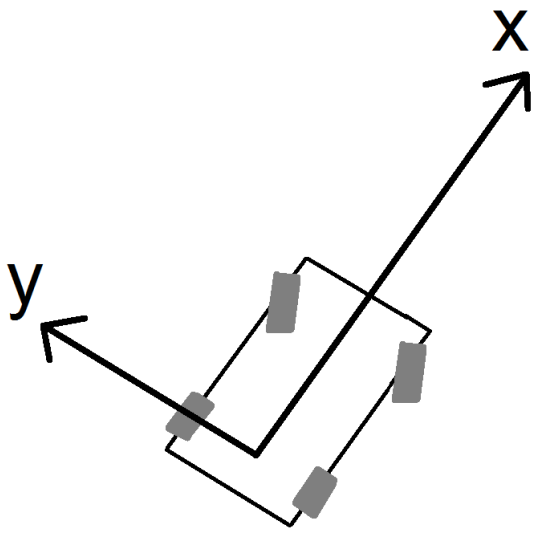

Figure 2: Local (moving) Referential

1.2 Low Level Sensor Readings Interface

With this task we will present the ROS user with an easy way to have access to all of the low level information of the RCV. This includes, among others, steering wheel angles, wheel encoder readings and gas pedal position.

2 Optional

2.1 IMU Readings Interface

This will allow the ROS user to have access to the information transmitted from the IMU.

2.2 GPS Readings Interface

This will allow the ROS user to have access to the information transmitted from the GPS.

2.3 GPS Based Path Following

Based on the GPS measurements, the RCV will be able to follow a path defined of GPS waypoints.

2.4 Lane Marking Camera Interface

The RCV will soon be equipped with a Lane Detection camera from Scania. This camera is currently using CAN for message transmission. A possible ROS interface can be done such that the camera information arrives easily to computers in the Network.

2.5 Lane Keeping Controller

If possible (or required), a lane keeping command can also be implemented. This will make use of the Lane Tracking camera that will be implemented in the RCV. Very high level commands can then be sent to the RCV:

- Keep on lane with a fixed velocity or fixed acceleration
- Change to a different lane
- More complicated commands, such as those happening in intersections

2.6 Radar Safety Checking

If possible a Radar will be integrated and will serve as a safety check, overriding a commanded path in case it shows a collision according to the radar measurements.

2.7 Odometry

A simple odometry/Information system can be implemented, it will be responsible from broadcasting general information about the car such as its position and velocities relative to a user defined fram. Not much state estimation effort will be put into it, it will simply make use of the existing GPS/IMU to generate odometry information. The generated odometry will be processed and displayed in a simple way to the user, and be consistent with the desired coordinate system.

This will allow the user to define fixed frames (figure 3) and send path commands in this easier to use frame.

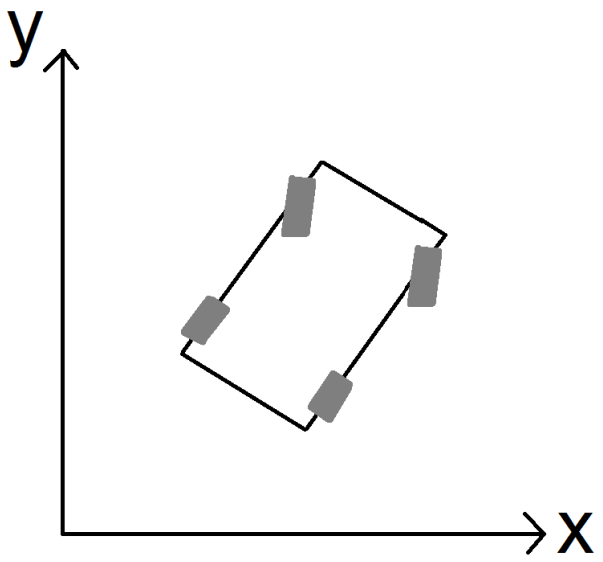

Figure 3: Fixed Referential

2.8 Platform Health

Typical sources of errors will be shown to the user (battery check, etc)