EL2425 - Slip Control - Meeting agenda 2016-11-16

November 15, 2016

1 Done

- Microcontroller messaging. Teensy was flushed with new firmware that allows only messages of type `slip_control_communications/input_drive` to be passed to it. (Previously, a message type of `f1tenth_msgs/drive_values` was hardcoded into Arduino's firmware, making communication within package `slip_control` impossible due to incompatibility.)
- Time constant for velocity found, although significant differences between time constants have been found for different velocity references.
- ROS infrastructure set.
- MPC python package found: `cvxopt`. It seems to be able to capture the essence of what our goal is. See `http://nbviewer.jupyter.org/github/cvxgrp/cvx_short_course/blob/master/intro/control.ipynb`
- Theoretical solution involving MPC and traveling on the circumference of a circle found.

2 Ongoing

- Theoretical solution involving MPC and traveling in the middle of two walls found. The problem can be decomposed into two separate and independent components involving a translation and a rotation of the vehicle. Given the pose of the vehicle at time t as (x_c, y_c, v_c, ψ_v) and two range scans at $-90°$ and $+90°$ with respect to the longitudinal axis of the vehicle which are denoted as CL and CR respectively, the error in translational terms is

$$e_x = \frac{CR - CL}{2} sin\psi \tag{1}$$

$$e_y = \frac{LC - CR}{2} cos\psi \tag{2}$$

In other words, at time t the vehicle should have been at point

$$x_o = x_c + \frac{CR - CL}{2} sin\psi \tag{3}$$

$$y_o = y_c + \frac{LC - CR}{2} cos\psi \tag{4}$$

With regard to rotation, given the pose of the vehicle at time t as (x_c, y_c, v_c, ψ_v) and three range scans at $-90°$, $0°$ and $+90°$ with respect to the longitudinal axis of the vehicle which are denoted as CL, CF and CR respectively, the heading angle error is

$$\phi = \frac{\pi}{2} - tan^{-1}\frac{CF}{CR} \tag{5}$$

since $\mu + \phi + \frac{\pi}{2} = \pi$ and $tan(\mu) = \frac{CF}{CR}$.
In other words, at time t the vehicle should have a heading angle of

$$\psi_o = \psi_c + \phi = \psi_c + \frac{\pi}{2} - tan^{-1}\frac{CF}{CR} \tag{6}$$

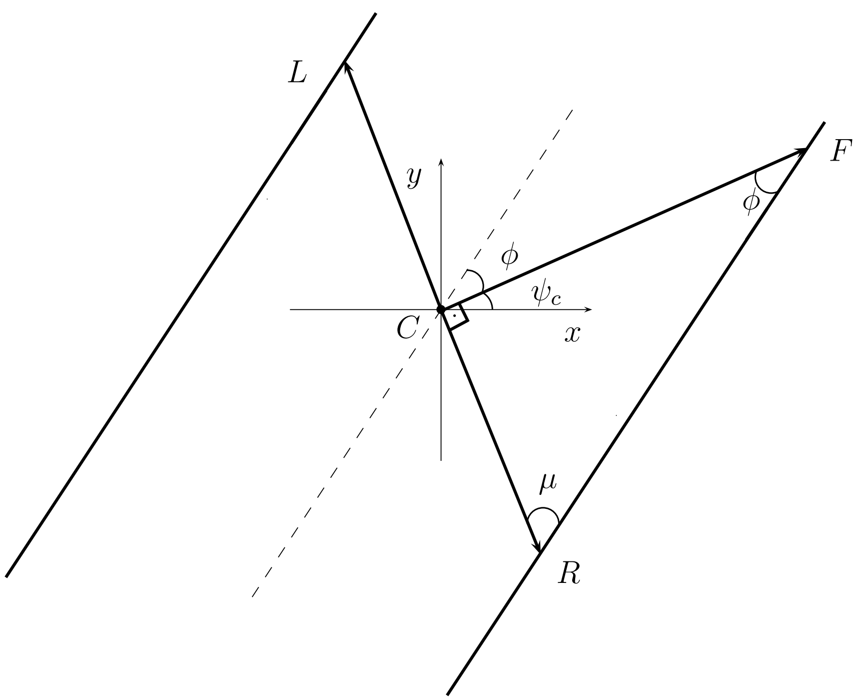

Figure 1

Hence we can formulate the optimization problem as

$$min \sum_{k=0}^{k=N} (X - X_o)^T Q (X - X_o) + U^T R U \tag{7}$$

$$\text{subject to } X[t+1] = AX[t] + BU[t] \tag{8}$$

$$-U_{min} \leq U \leq U_{max} \tag{9}$$

where $X = [x_c, y_c, v_c, \psi_c]^T$ and $X_o = [x_o, y_o, v_o, \psi_o]^T$. Positive definite matrices Q, R will have to be adjusted experimentally.

However, the vehicle's velocity is not measurable since the vehicle does not have encoders connected to its wheels and MOCAP or range scans cannot provide measurements of velocity. Either a Kalman filter will have to be employed in order to estimate the vehicle's velocity, or the ESC feature of the vehicle will have to be investigated with regard to its ability to ensure that the input velocity is indeed the vehicle's velocity.

3 Issues

- The ethernet adapter for the lidar is broken and needs to be replaced. This means that packages `circular_mpc` and `centerline_mpc` cannot be tested until communication with the lidar is fixed.

- The SML lab is booked for the week 14/11-18/11 (what about the weekend?), hence no MOCAP. This means that packages `circular_pid` and `centerline_pid` (gains need adjusting) cannot be tested until at least Saturday 19/11.

- Package `circular_pid`, which was to be working out-of-the-box, does not work. The fault lies somewhere inside ROS: it appears that when ROS_MASTER runs outside Jetson, sometimes communication between Jetson and the nodes running outside it is not established. When it is established, no messages are getting through to teensy.