EL2425 - Slip Control
Meeting agenda 2016-11-16

November 15, 2016

1   Done

- **Microcontroller messaging**. Teensy was eventually 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 hard-coded into Arduino's firmware, making communication within package `slip_control` impossible due to incompatibility.)

- **Time constant** pertaining to the velocity response of the vehicle found, although significant differences between time constants for different velocity references have been found.

- **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 PID and traveling on the circumference of a circle found, and code for it written. However, the involved gains have to be tuned experimentally.

- Theoretical solution involving MPC and tracking the centerline of a lane circle found, however code for it has not been written yet.

2   Ongoing

- **Theoretical solution involving MPC and tracking the centerline of a lane**.

  The problem can be decomposed into two separate and independent components involving a translation and a rotation of the vehicle.

  Translational component 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

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

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

since $CL + CR = L = OL + OR$, and $CL = OC + OL$, which means that

$$OC = OR - CR = \frac{L}{2} - CR = \frac{CL + CR}{2} - CR = \frac{CL - CR}{2} \tag{3}$$

where L is the width of the lane whose centerline the vehicle is to track.

Furthermore, in the COM triangle:

$$O'C = OC\cos\mu = OC\cos(\frac{\pi}{2} - \psi) = OC\sin\psi = \frac{CL - CR}{2}\sin\psi \tag{4}$$

$$O'O = OC\sin\mu = OC\sin(\frac{\pi}{2} - \psi) = OC\cos\psi = \frac{CL - CR}{2}\cos\psi \tag{5}$$

In other words, at time t the vehicle should have been at point $O(x_o, y_o)$:

$$x_o = x_c - \frac{CL - CR}{2}\sin\psi \tag{6}$$
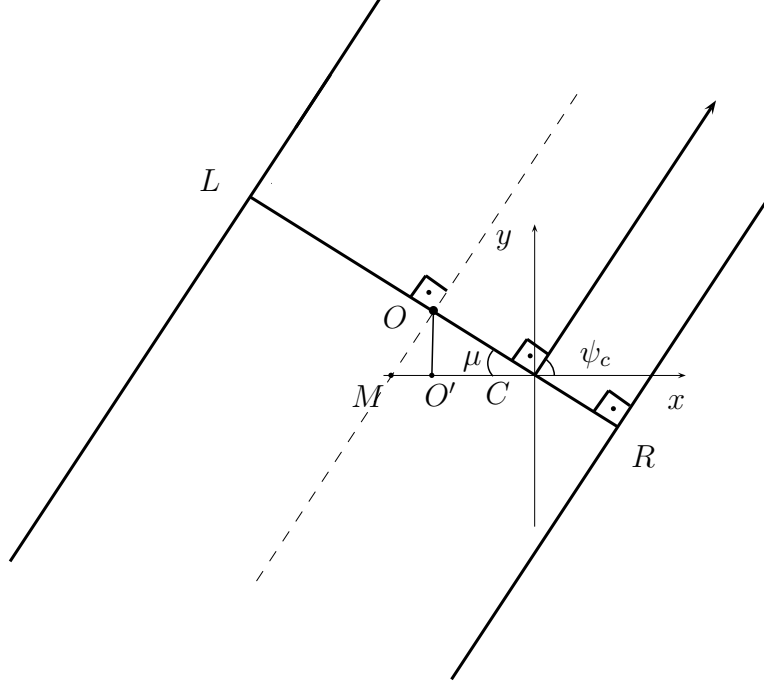
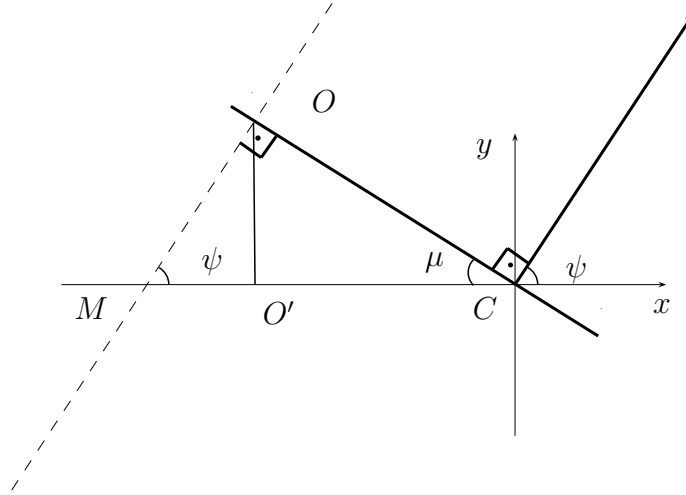$$y_o = y_c + \frac{CL - CR}{2}\cos\psi \tag{7}$$

Figure 1

Figure 2

Rotational component 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

$$\phi = \frac{\pi}{2} - tan^{-1}\frac{CF}{CR} \tag{8}$$

3

since $\mu + \phi + \dfrac{\pi}{2} = \pi$ and $tan(\mu) = \dfrac{CF}{CR}$.

In other words, at time t the vehicle should have a heading angle of

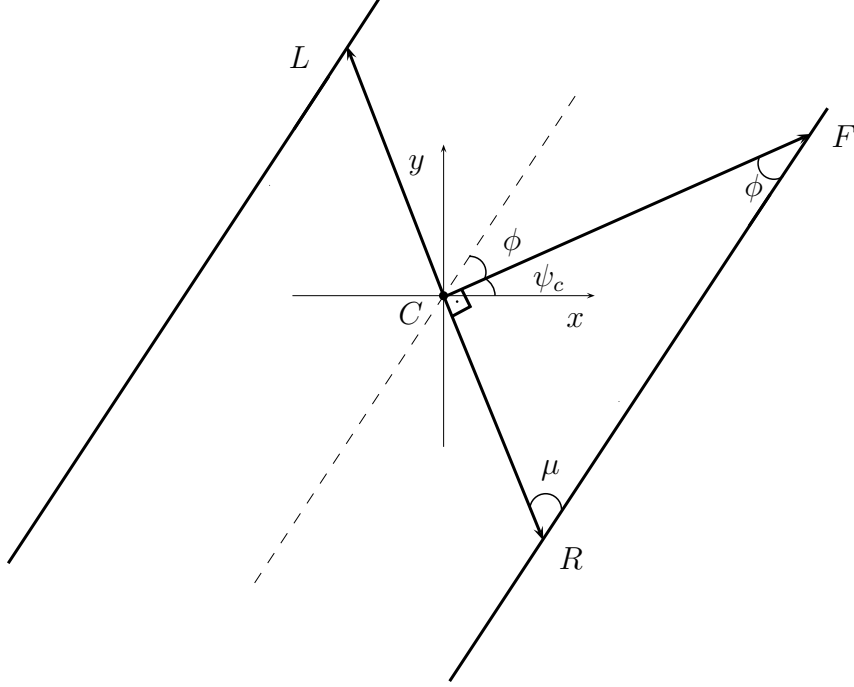$$\psi_o = \psi_c + \phi = \psi_c + \frac{\pi}{2} - tan^{-1}\frac{CF}{CR} \tag{9}$$

Figure 3

Hence we can formulate the optimization problem as

$$min \sum_{k=0}^{N}(X - X_o)^T Q(X - X_o) + U^T R U \tag{10}$$

$$\text{subject to } X[t+1] = AX[t] + BU[t] \tag{11}$$

$$-U_{min} \leq U \leq U_{max} \tag{12}$$

where $X = [x_c, y_c, v_c, \psi_c]^T$, $X_o = [x_o, y_o, v_o, \psi_o]^T$, N is the horizon and $U = [v, \delta]^T$ is the input vector. 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 there is no access to 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 within 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.

4 To do

- Tune gains of the PID concerning package `centerline_pid`.

- Implement `centerline_mpc`.

- Implement `circular_pid`.

- A node that handles the linearization of the kinematic model of the vehicle has to be written in ROS.

5 Misc.

The progress of the project can be observed in `trello` and `github`:

- `https://trello.com/b/uEP0jl0B/slip-control`

- `https://gits-15.sys.kth.se/alefil/HT16_P2_EL2425`

- `https://gits-15.sys.kth.se/alefil/HT16_P2_EL2425_resources`