Trent Ziemer

Dr. Newman

EECS 397

September 9, 2016

Homework 1: Minimal Sine Controller

Part 1: Adjusting Parameters

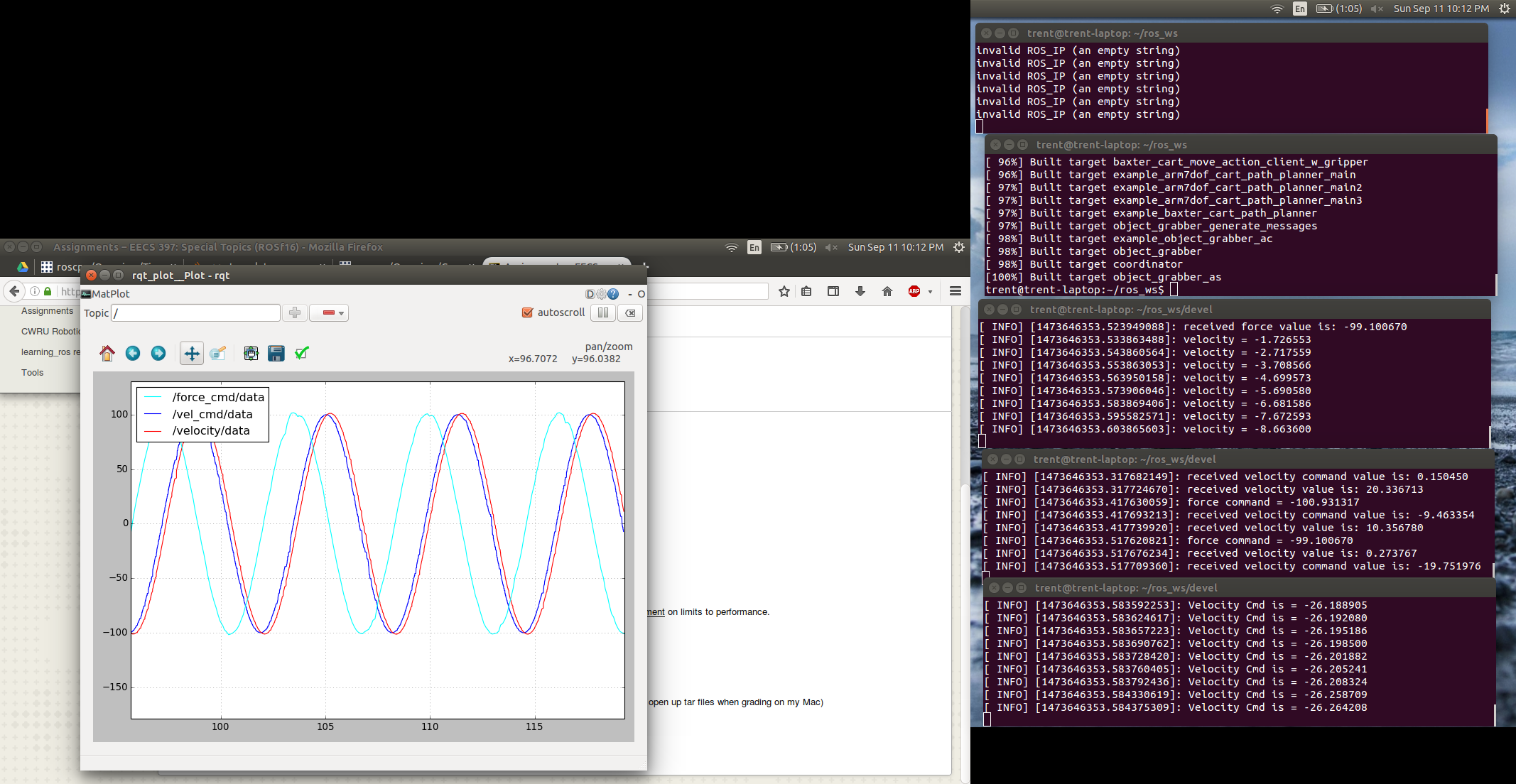
The controller gain, Kv, controls how (and how quickly) the actual velocity responds to the commanded velocity. The integration time step, dt, controls how often the value of the actual velocity is recalculated (lower dt => more accurate calculations). The controller time step, controller\_dt, governs how often the controller sends the desired (commanded) force signal to the simulation (lower controller\_dt => more responsive velocity following).

The initial Kv value of 1 was too slow (taking about 10 seconds to reach the desired value), and a value of 10 produced a sinusoidal oscillation that never reached the desired value. A value of 3.5 was much better, reaching the desired value in about 2 seconds, and then a value of Kv = 5 took about 1.5 seconds to reach the desired value. Going up any more, such as to Kv = 6 made it slow down again. We settled that a Kv of about 5 would be optimal for this simulation because going either higher or lower would cause the response to slow down.

Changing values of dt or controller\_dt any lower than their default values (0.01 and 0.1, respectively) made little change in the simulation results because the default values are already very accurate. However, any additional decrease in their values would cause the simulation to be more accurate, if, for example, we were using a different controller than changed the velocity on the order of milliseconds rather than seconds.

Part 2: Sine Commander

The below image shows the velocity command published by the sin commander (blue), the transmitted force between min\_ctl and min\_sim (teal), and the actual produced velocity in the simulator (red). The frequency is about 0.2 Hz, the amplitude is about 50, and the delay time is a few hundred milliseconds.



The performance of the simulator is limited by the propagation time between calculating the desired sinusoidal velocity and the simulator receiving the force command, and also how frequently each node sends messages. While with this frequency the sinusoids closely match each other, if we were to increase the frequency by a factor of ten or so, then we would notice that one lags behind the other and has a greater amplitude (due to overshooting during that lag time).