# Change the values of dt in the simulator and Kv and dt\_controller in the minimal controller, Comment on the influence of these values, and recommend preferred values:

1. For the dt in the simulator, it controls how “reactive” are the simulator, hence how real the data get. If we set the dt in a large number, we will see that especially when plotting sin, it is no longer smooth but chunky as illustrated. However, decreasing dt seems not have a huge effect unless change the sine commander as well. However, due to human eye limitation, it seems not a huge different**. So I would recommend not to change but leave the original 0.01**.

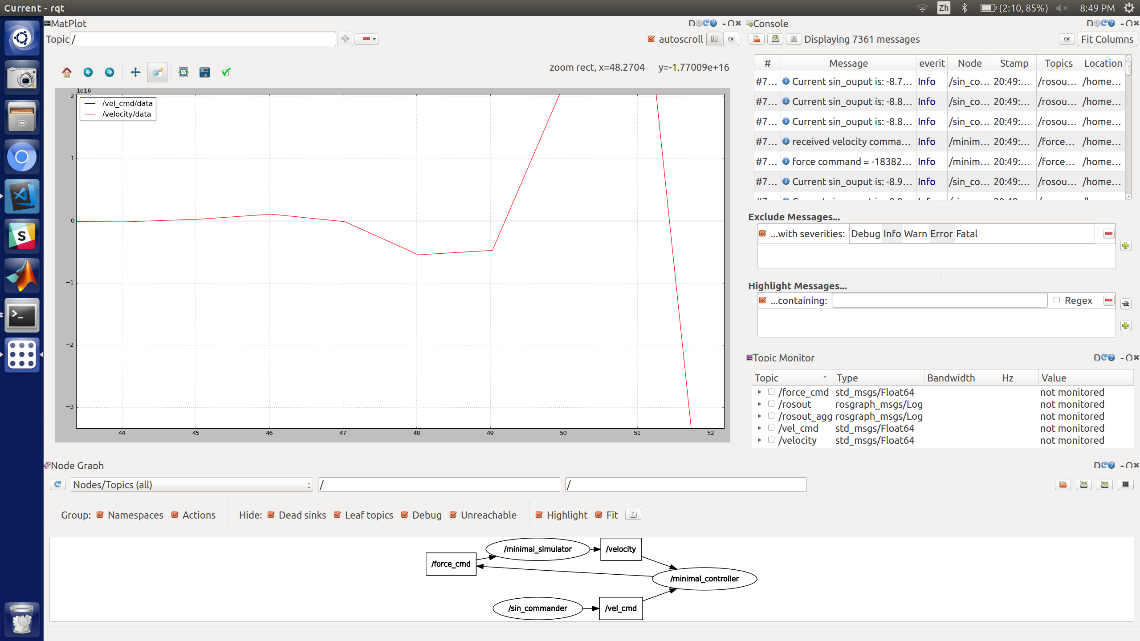


Figure Change in dt to a larger number

1. Change in controller kv changes how fast the controller responds to the change of velocity. A larger kv will improve the response, but an extra-large value of kv will cause some overshoot hence we will have an unstable system, as illustrated below. **I found that value of 5 seems working**, and over that overshoot starts to happen.

A screenshot of a computer

Description generated with very high confidence

Figure Unstable system

1. Change the dt in controller affects how fast it announces its change of signals. When value is increased, it slows down the controller. Considering the fact that the current plot looks fine**, I do not recommend to change it. However, a value of 0.01 can be a choice to sync up all the dt across all files.**

# Plot out the commanded velocity and the simulated velocity (superimposed) using rqt\_plot.

1. Below is using**: dt=0.01 for simulator, kv = 5 and dt = 0.01 for controller, dt = 0.01 for commander. The frequency is set to be 0.1 Hz and amplitude is 50.**

**A screenshot of a computer

Description generated with very high confidence**

Figure Ideal Scenario

# Comment on limits to performance.

If the given frequency is set to be large, the velocity will lag behind. Although on the previous “ideal” picture it didn’t show much, the below illustrated if I increase my frequency to 1hz, it will have a larger lag behind. It is due to the limitation of how fast each node is communicating to each other as well as kv, dt we set in file.

A screenshot of a computer

Description generated with very high confidence

Figure lagging