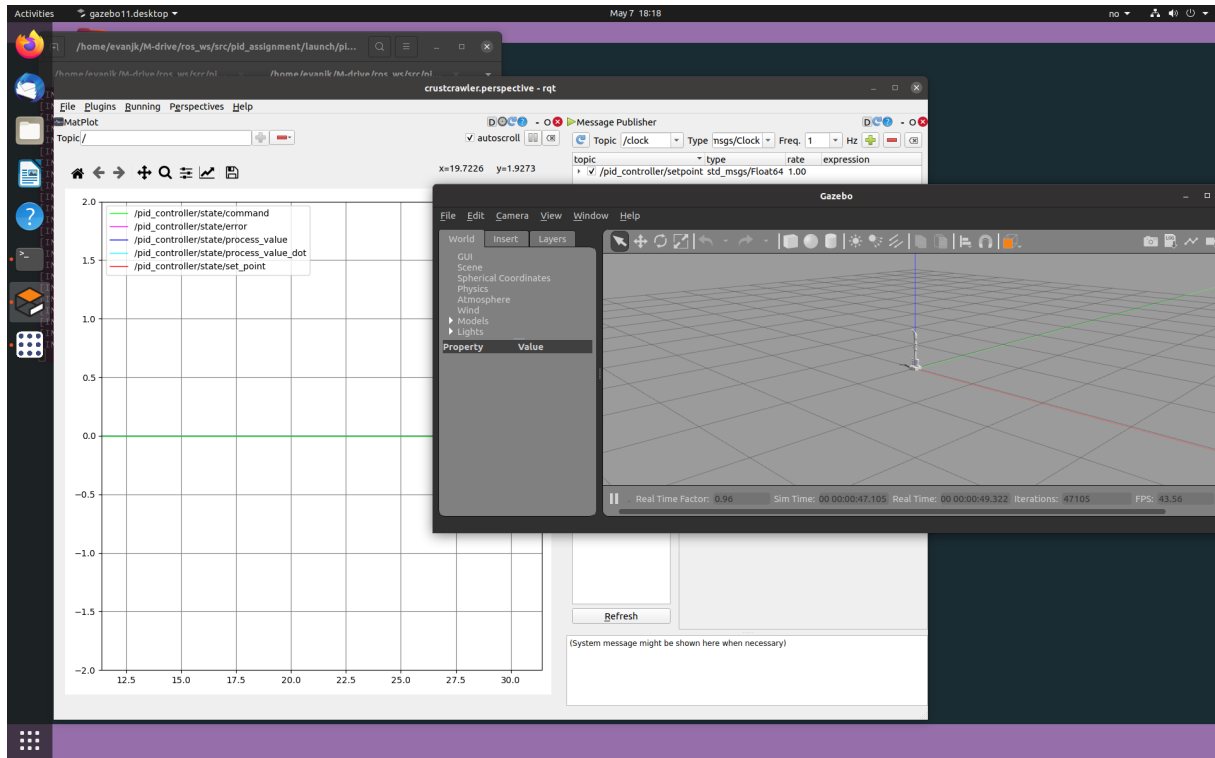
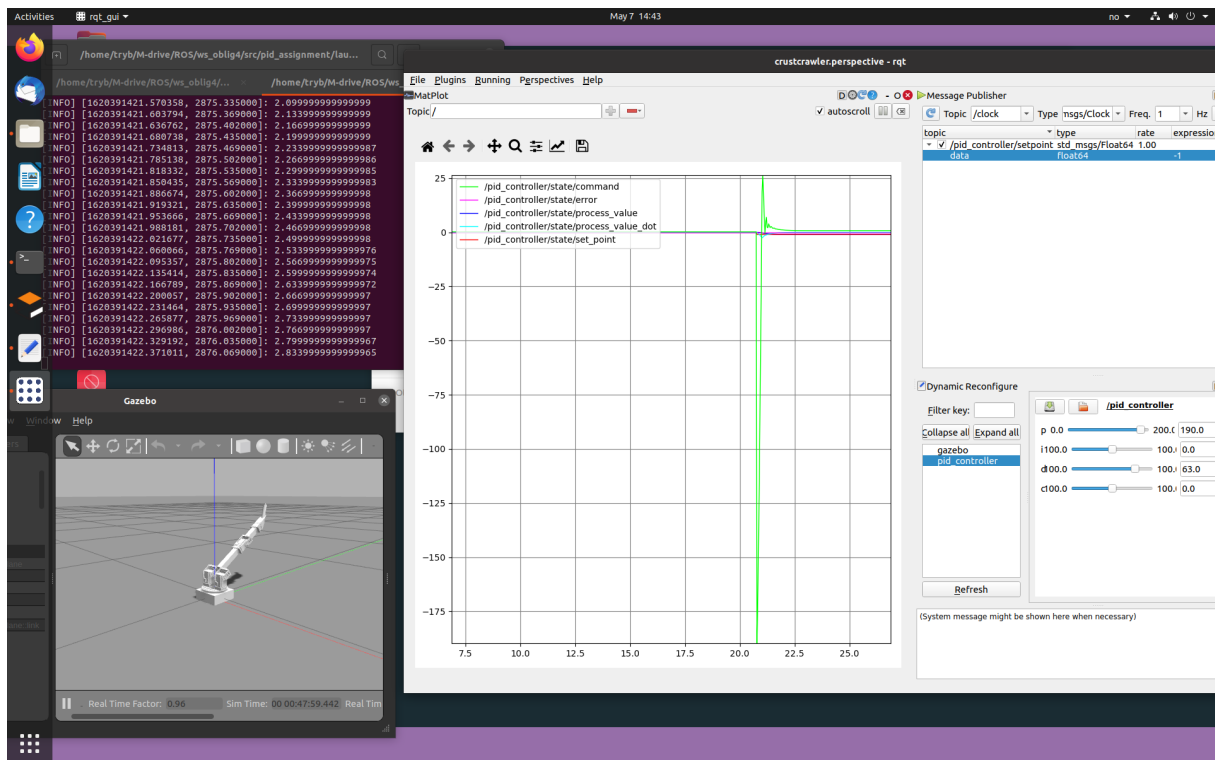


Task 2)

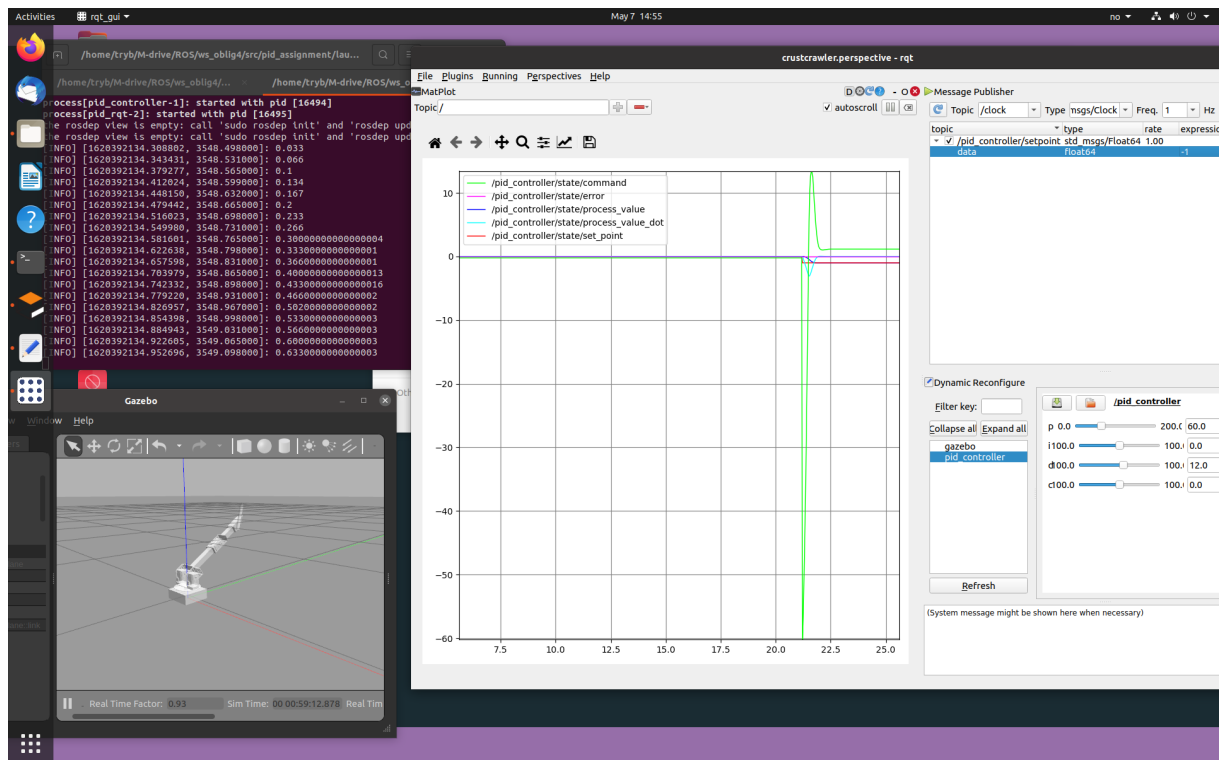


Task 3a)



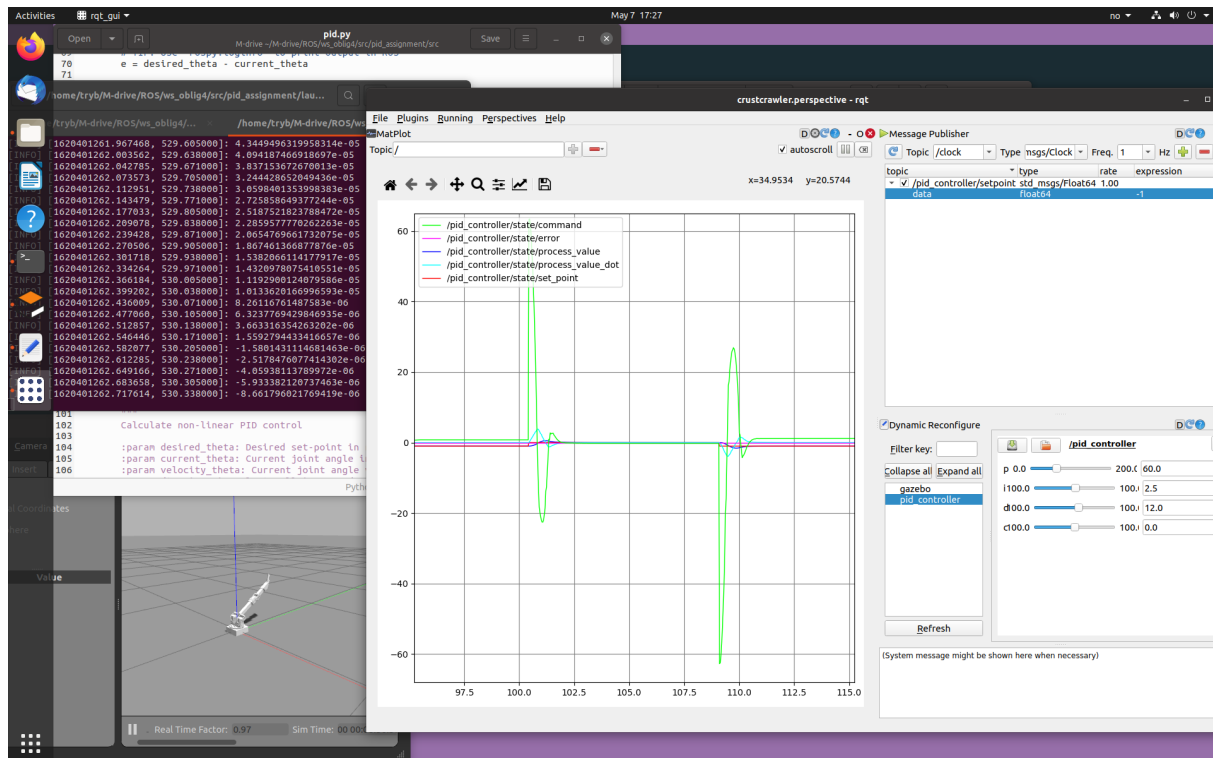
The PD-controller is now implemented and is functionally. The values used in the controller is the calculated values from Task 1d). The response of the settling time is not as expected. The formula tells us it should be around 4/6 s but in reality it turned out to use over 2.8 s. This is much higher than anticipated and for optimization we have to find new values for K_d and K_p . The reason why the settling time is wrong may be due to incorrect center of mass as we have assumed that joint 3 extends joint 2.

Task 3b)



For tuning of the controller we did a bit of trying and failing. We tried to find the best combination of both the derivative constant and the propositional constant, without causing too much overshooting or oscillation. We settles with the values as seen in the screenshot from above ($K_p = 60$ and $K_d = 12$). This combination caused it to settle within 5% of the steady state error around 0.6 seconds. This is much more similar to the expected settle time.

Task 3c)



For this task we had a steady state error on about 0.019 radians. This means that the tip of our joint would be 6.8 mm of the desired destination. As 6.8 mm isn't precise enough, this had to be optimized. Especially if you consider a robot with several joints. If there is this big of an error in all links, it would be drastically off at the tool tip (i.e. a gripper). Therefore we implemented an integral to the controller, a constant K_1 . Now that we have a PID-controller, a steady state error would be minimal if we configure it correctly. A PID-controller causes the robot to overshoot a bit easier but as mentioned it will result in a more precise manner.

We got a steady state error of 1×10^{-6} radians which is 0.00035 mm error at the tool tip. This is a result we are more than happy with.

Task 4)

We decided to not go through with this task as the VM Ware horizon client was really slow. We tried it at different times at different days but it was still challenging. Instead we used more time to optimize task 3 and implement a PID-controller as well as focusing on understanding the controller theory.