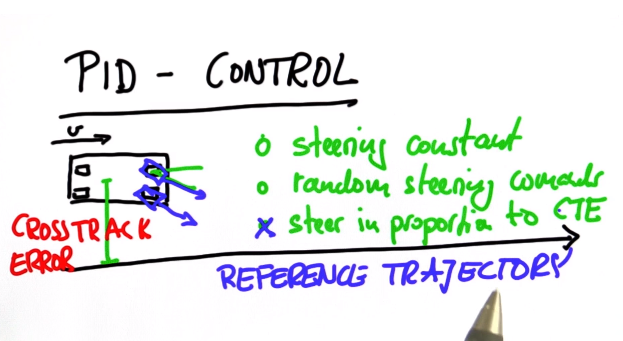
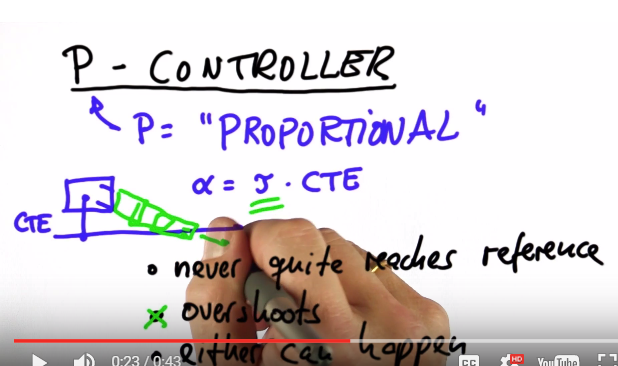
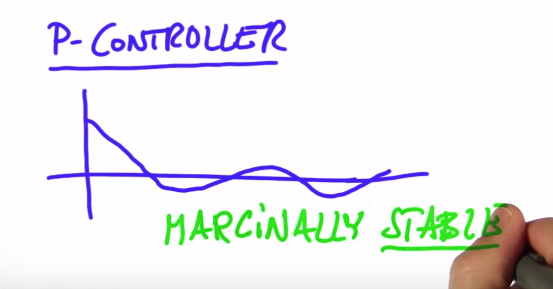
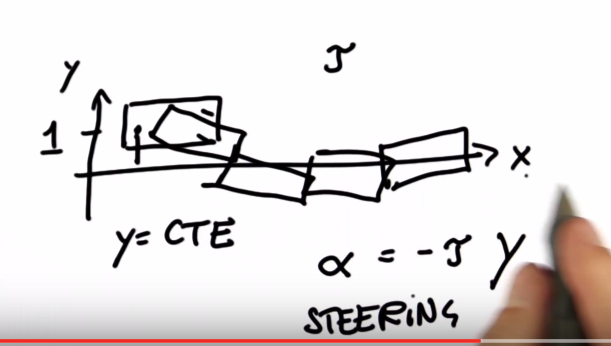
16.3 proportional control



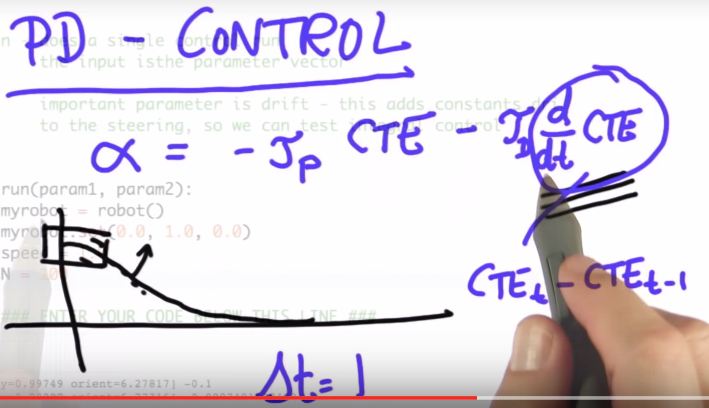


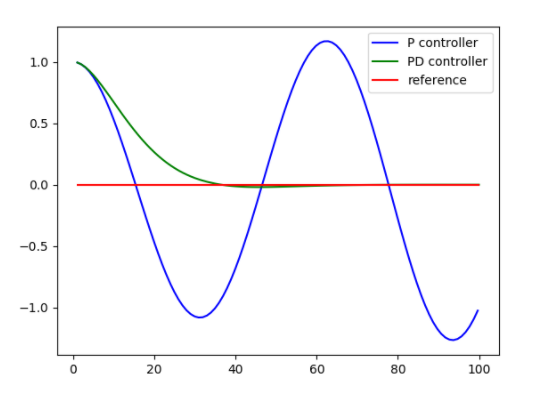
Marginally stable



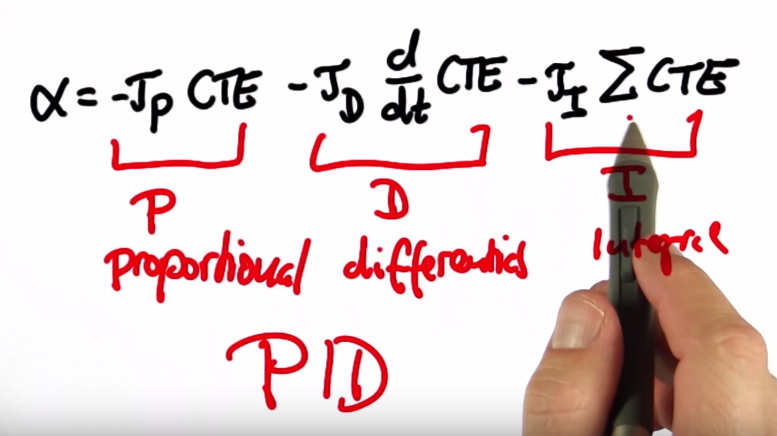


Use PD control to avoid overshoot

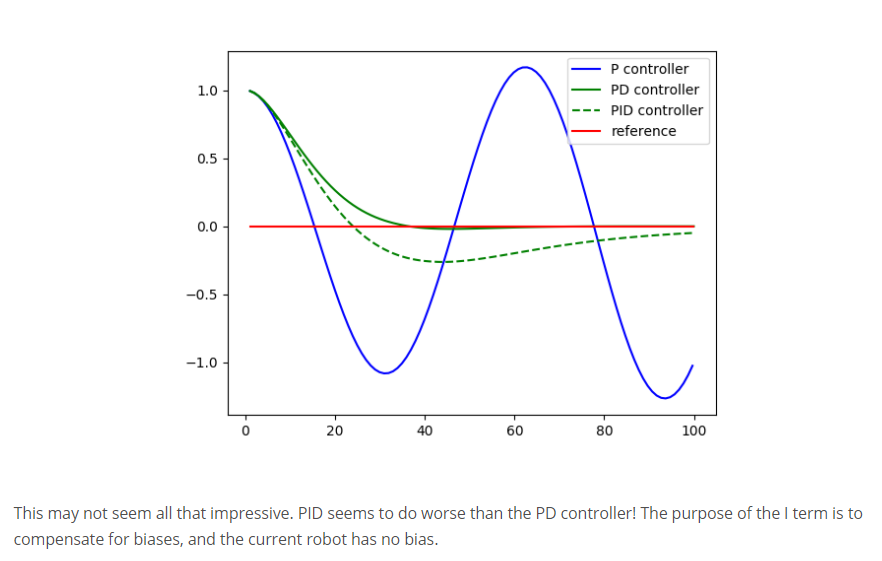




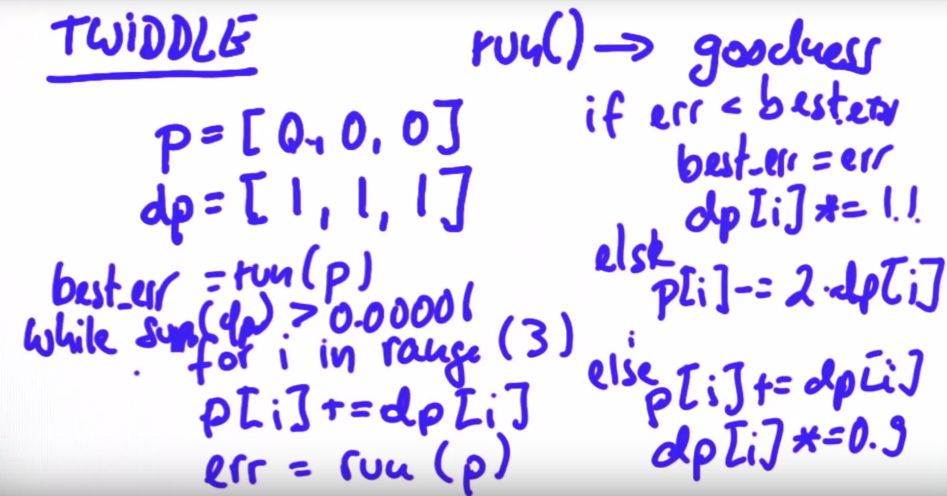
16.11 Use sum/integral of all observed crosstrack errors to compensate bias and correct robot’s motion



When no bias, PD > PID

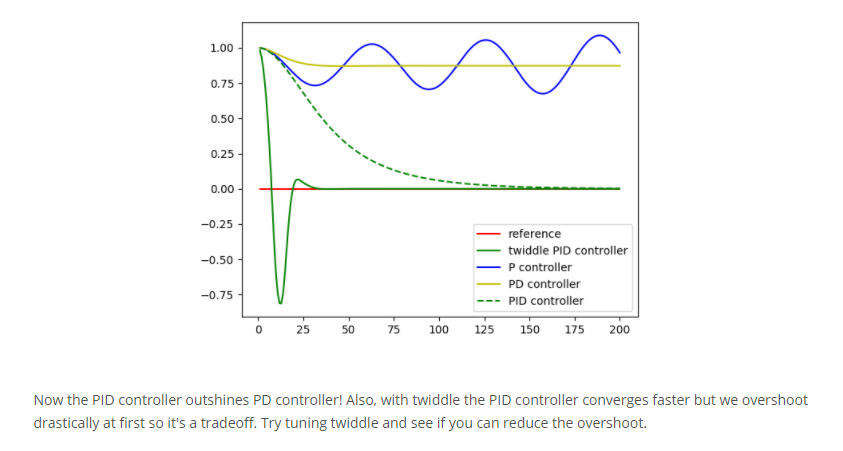


16.13 twiddle



With bias:

robot.set\_steering\_drift(10 / 180 \* np.pi)



2 solution for P4:

1. ***PID manual parameter tuning***

By suboth :

Here are some general tips to get the manual tuning to work:

1. Try to find a value for Kp which keeps you car on the track atleast for 2-3 seconds. The car might oscillate but that is fine.
2. To tune down the oscillation gradually try with increased Kd values.

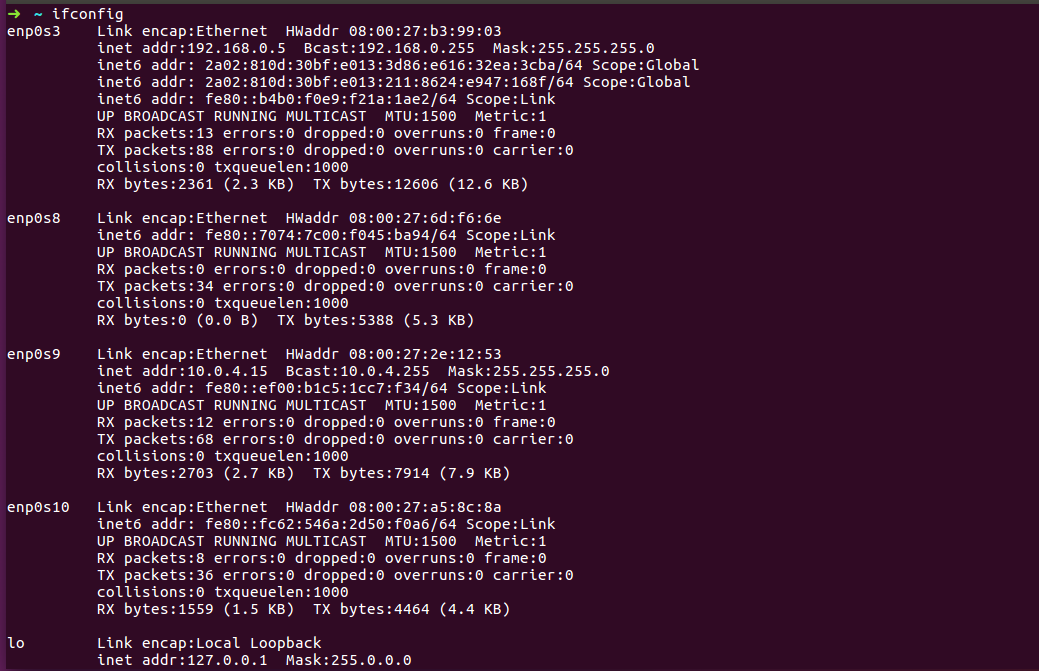
Increasing Kd reduces oscillations, whereas increasing Kp increases the magnitude of turning. You need to find a sweet spot where the car turns sufficiently but does not oscillate. Once you find Kp and Kd values which work well, try very small values of Ki. Don’t be surprised if Ki is many orders of magnitude lower than Kp or Kd.

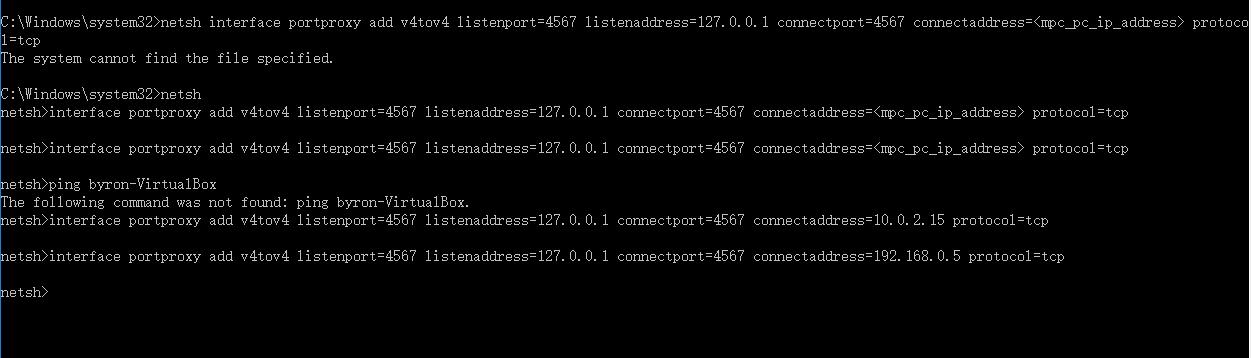
It may also help to have another PID controller to control the speed. Having a nearly constant speed makes controlling steering a lot easier.

Let us know if you are not able to get it to work using the above tips.

1. ***Twiddle automatic tuning***

How to run win sim, and generate signal from linux:





Netsh > interface portproxy add v4tov4 listenport=4567 listenaddress=127.0.0.1 connectport=4567 connectaddress=192.168.178.74 protocol=tcp