ECE**5780** Inv Pend Robot

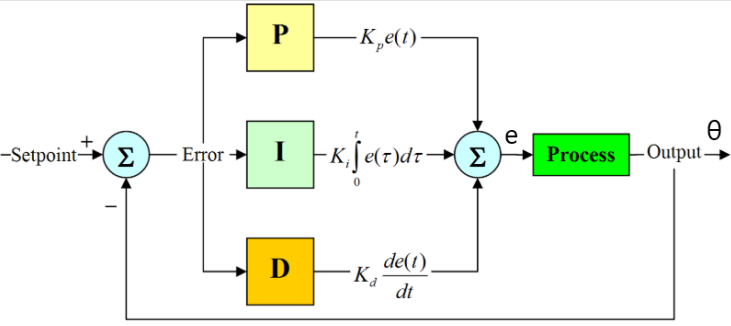
# PID in C

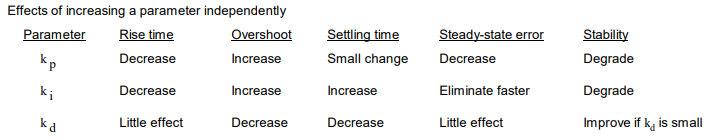
The robot is driven by monitoring one variable – θ. This will be the deviation from our set equilibrium point, and whose raw measured data is read from the Gyroscope. Think of the feedback control as a black box:

INPUT – This will come from our BT signal and can be read as a quantized voltage.

OUTPUT – This will come from our gyroscope reading (1 axis, depending PCB orientation)

A brief overview of what each part does:





PID control in the frequency domain (open loop):

The controller multiplies these weighted gains by the error signal whose value is then multiplied by the Process (robot). Ideally, we will **eliminate steady state error (SSE)** and **reach a settling time as fast, and smooth as possible**. Settling time is defined as getting and staying within 2% error, and “smooth” means we aren’t rapidly accelerating to achieve our goal (slamming on brakes vs. gradual pressure).

# Gyroscope

**High-Pass Filter** LD3GD20HTR is provided with a high-pass filter for our digital signal. We use this filtering to prevent angle “walk” and improve the accuracy of our signal. The walk is due to low frequency noise from a small DC bias.

**Angle Measurement**

To get a measurement of the current angle, you use the previous measurement with scaled angular velocities from previous readings. This will require holding and updating 6 variables for each loop.

Note: It has been suggested that accuracy can be improved by the **fusion** if accelerometer and gyroscope sensors such as in the **MPU6050**. Fusion is done using a **Kalman** or **Complementary Filter**.

# Links of Interest

<https://www.mathworks.com/matlabcentral/fileexchange/58358-inverted-pendulum-robot-project-matlabhw2k16>

[MATLAB to STM32 Tut](https://www.youtube.com/watch?v=hnZ622Fkwkg)

[MPC on Embedded](https://www.researchgate.net/publication/259671909_Implementation_Aspects_of_Model_Predictive_Control_for_Embedded_Systems)

[PID Arduino INV Pend Bot](https://www.youtube.com/watch?v=VxpMWncBKZc)

<http://www.brokking.net/yabr_downloads.html>

[Generating uC MPC code](https://link.springer.com/chapter/10.1007%2F978-3-319-60699-6_30)