

Hi tmladek,

Your desired speed is the PID controller's "setpoint" and your encoder is the "process variable" that's measuring the actual speed. Both your desired and actual speed should be the same unit of measurement, be it degrees/s, radians/s, RPM or whatever.

The proportional part of the PID controller simply subtracts the actual speed from the desired speed (to give your error) and multiplies it by a gain factor:

PID proportional output = (desired speed - actual speed) * gain

What took me some time to understand was the fact that the PID output is just a value, it doesn't have to be the same units as the inputs. It's just used to drive your "process" or in your case a motor. You just need to map and constrain the PID output to drive whatever it is you're trying to control, again in your case mapping and constraining it to a value between 0 and 255.

The thing to think about here is feedback. The PID system is trying to drive the actual motor speed as measured by your encoder so that it matches your desired speed, in effect it's trying to minimise the error. This error is always about and with respect to your setpoint (desired speed). Naturally the error will always go slightly positive and negative about your setpoint, as the PID system never quite reaches equilibrium. In an extreme case, if you set your gain too high you can get overshoot or even worse oscillations, as your error swings uncontrollably between negative and positive.