ME333 Introduction to Mechatronics
2022 Quiz 7: PID control

Name Sean Morton

1. Give pseudocode for a basic PID controller (without integrator anti-windup).
   There are functions get_ref() and get_sensor() to call, and you can make others if you want.
   There are already global variables, and you can add more:

```
static volatile float eint = 0;
static volatile float eprevious = 0;
```

```
#define SCALAR <something>
static volatile float e = 0;
static volatile float edot = 0;
static volatile float sensorval;
static volatile float ref;
```

   The ISR is already setup to run at 1kHz:

```
__ISR(timer at 1kHz) {
    // your code here:
    sensorval = get_sensor();
    ref = get_ref();
    e = ref - sensor_val;
    edot = e - eprevious;

    //setup the output of the PID controller
    u = Kp * e + Ki * eint + Kd * edot;
    if (u > 100.0) {
    u = 100.0;
    } else if (u < 0.0) {
    u = 0.0;
    }

    //setup output, may or may not be like OC1RS in the hw
    output = u / 100.0 * SCALAR;
    send_output(output);

    //set variable values for next iteration
    eint += e;
    eprevious = e;

    interrupt_flag = 0;
}
```

2. Explain what integrator anti-windup is:

   In a case like where you're controlling the position of a robot arm, and there's an obstacle blocking the arm from getting to where it wants to go, the integral of error can increase by a lot during that time. When that obstacle is moved out of the way, the PID controller wants to counteract that high integral of error, so the arm might spring forward, way past the desired position.

   Integrator anti-windup sets a cap on how large eint is allowed to get. This prevents eint from growing out of control, in cases like I described above.

3. You have picked Kp, Ki, and Kd gains.
   a. The response has too much overshoot. Which gain could you increase to reduce the overshoot?
      You could increase Kd to prevent the response from increasing past the desired value too rapidly.
   b. The response has too much overshoot. Which gain could you decrease to reduce the overshoot?
      You could decrease Kp to prevent the Kp * e from being too large at each timestep
   c. The response has the right overshoot and settling time characteristics, but too much steady-state error. Which gain could you increase to reduce the steady-state error?
      Increasing Ki may reduce the steady state error for the response.