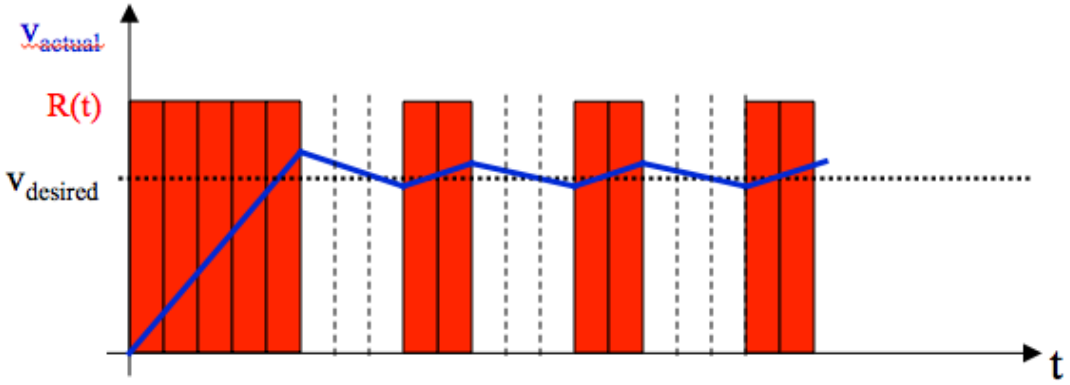
Digital and Embedded Systems
Professor Thomas Bräunl
Associate Lecturer Kieran Quirke-Brown

Tutorial 7 – PID Control in C

1. Implement a complete C program for PID motor control.

- The motor output pin is GPIO1.
- The encoder input pin is GPIO2.
- The motor is spinning only in one direction, so no motor direction pin is required and a single encoder input is sufficient.



Note:

- The main program has to initialize pins
- The encoder needs to be setup as a interrupt to avoid wasting the CPUusage
- The motor routine reads the encoder, calculates the current speed, then decides whether the motor should be switched on or off.

Example Solution:

```
int count = 0;

void tickCount() {
    count++;
    return;
}

void setMotor() {
    static int r_old = 0, enc_old = 0, enc_old2 = 0;
    const int v_des = 80;   //assumed speed is 80 ticks per second
    int Kp = 5, Ki = 2, Kd = 1; //values randomly chosen
    int v_act = (count - enc_old) * 100; //this gives ticks/sec as the function runs at 100 times a second

    int error = v_des - v_act;
    int r_mot = r_old + Kp * (error - e_old) + Ki * (error - e_old) / 2 + \
        Kd * (error - 2 * e_old + e_old2);

    r_mot = max(min(r_mot, 100), 0);

    enc_old2 = enc_old;
    enc_old = count;
    r_old = r_mot;
    return;
}


void setup() {
    pinMode(1, OUTPUT);
    pinMode(2, INPUT);

    //setup tick count
    attachInterrupt(2, tickCount, RISING);  //updates count when there is a rising edge on encoder input

    //setup speed calculation
    hw_timer_t *My_timer = timerBegin(0, 80, true);
    timerAttachInterrupt(My_timer, &setMotor, true);
    timerAlarmWrite(My_timer, 100000, true);    //run 100 times a second
    timerAlarmEnable(My_timer);

}

void loop() {

}
```

2. Implement a C program that controls the temperature in the room using Hysteresis control.

- The heater output pin is GPIO1.
- The heater is controlled through an SCR which requires PWM control, set this up as a separate function.
- The cooling system uses an analogue controlled Chilled water valve onoutput pin GPIO3.
- The temperature input pin is GPIO2 and is an analogue value.
- The requirements for heater setpoint is controlled as follows:
    - if the room temperature falls below 20 degrees the heater setpoint should start at 20%. The heater setpoint should increase as the temperature gets lower than 20 degrees where it will be running at 100% by 17 degrees.
    - The system should use PID to control the heater to this setpoint
- The chilled water valve should be controlled as follows:
    - If the temperature gets to 25 degrees then the chilled water valve setpoint should be to 30%. The setpoint should continue to increase until it reaches 100% at 27 degrees.
    - The system should use PID to control the valve to the setpoint

Example Solution:

```
int calcOutput(int setpoint, int currentValue) {
    static int o_old = 0, set_old = 0, set_old2 = 0;
    const int v_des = 80;   //assumed speed is 80 ticks per second
    int Kp = 5, Ki = 2, Kd = 1; //values randomly chosen
    int error = (setpoint - currentValue); //this gives ticks/sec as the function runs at 100 times a second

    int output = o_old + Kp * (error - set_old) + Ki * (error - set_old) / 2 + \
        Kd * (error - 2 * set_old + set_old2);

    output = max(min(output, 100), 0);

    set_old2 = set_old;
    set_old = setpoint;
    o_old = output;
    return output;
}

void setup() {
    ledcAttach(1, 100, 8);  //use ledc to write a PWM output easily
    pinMode(2, INPUT);
    pinMode(3, OUTPUT);
}

void loop() {
    int temp = analogRead(2);
    static bool heating = false, cooling = false;
    static int lastOutput = 0;
    int output = 0;

    if (temp > 25) {
        output = min(max(30 + (temp - 27.0 / 2.0) * 70, 30), 100);  //ensure value is between 30 and 100
        analogWrite(3, max(calcOutput(output, lastOutput), 30));
        cooling = true;
        lastOutput = output;
    } else if (temp < 20) {
        output = min(max(20 + (20 - temp / 3.0) * 80, 20), 100);    //ensure value is between 20 and 100
        output = 255 * (max(calcOutput(output, lastOutput), 30) / 100.0);    //convert output value to duty cycle.
        ledcWrite(1, output);
        heating = true;
        lastOutput = output;
    } else {
        if (temp > 23 && heating) {
            output = 0;
            ledcWrite(1, output);
            heating = false;
        } else if (temp < 23 && cooling) {
            output = 0;
            analogWrite(3, output);
            cooling = false;
        }
        lastOutput = output;
    }
}
```