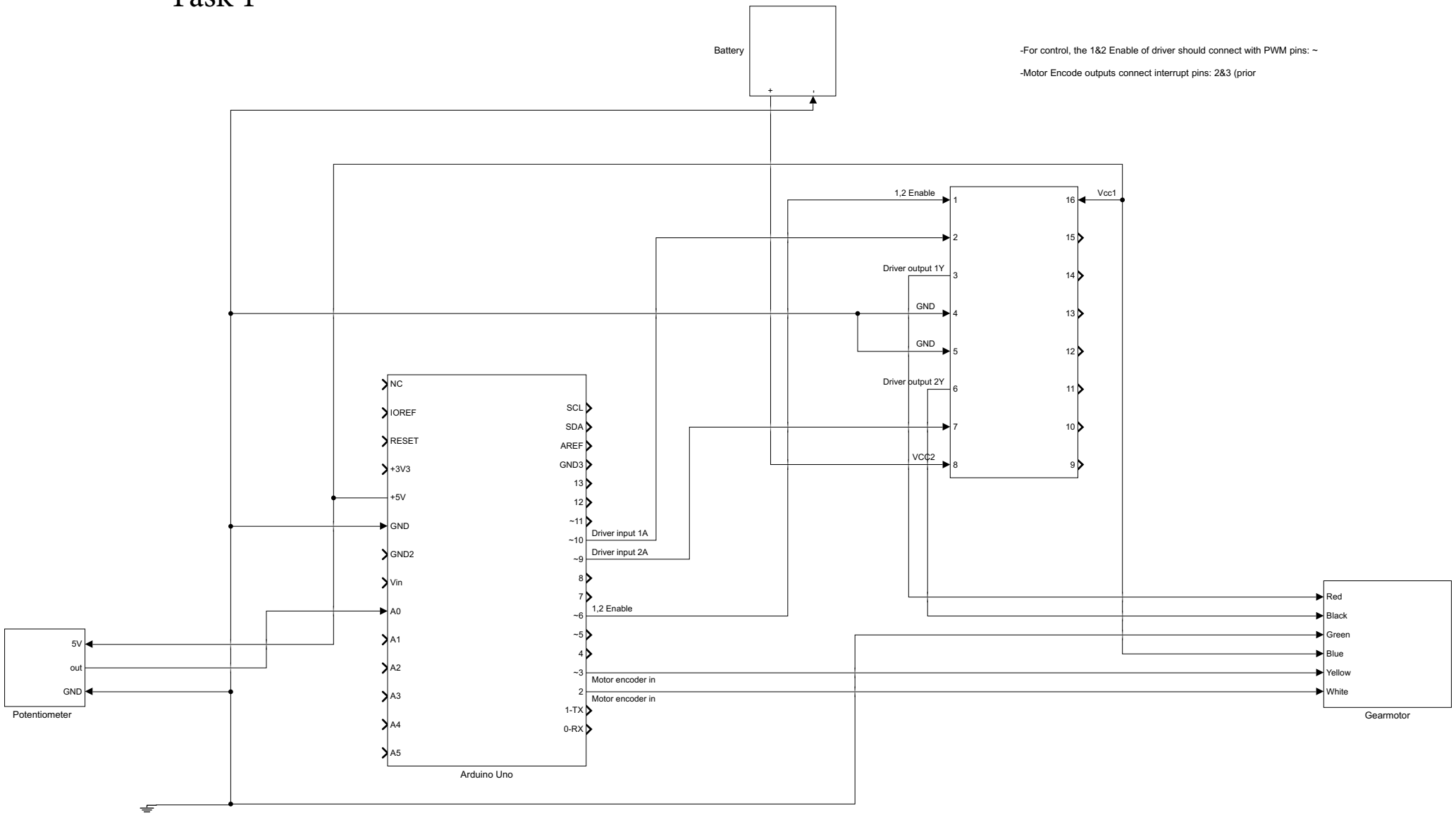
Circuit Diagram
Task 1



Main Tasks-Arduino

Task 2,3,4,6

Task 2:

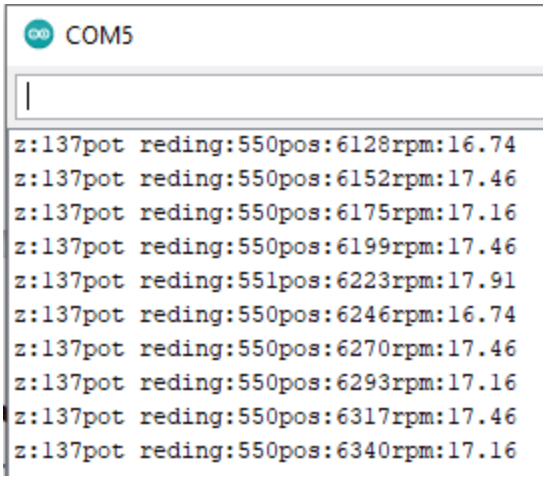


Figure 1: Arduino monitor reading

The task 2 requires the

1. Encoder reading: pos variable from fig.1. The pos variable changed when encoderA is rising. When encoderB reading shows 0, the rotation is CW. When encoder reading shows1 the rotation is CWW.

$$\omega_{rpm} = \frac{pos - posPrev}{deltaT} * \frac{60 * 4}{48 * 171.79}$$

2. The PWM output voltage can be calculated by duty cycle: $\frac{137}{255}$
   The output voltage can be seen as $3.2235V$.
   The motor transfer function is:

$$tf\ function: \frac{\omega}{V} = \frac{kt}{RJs + Rb + kbkt}$$

   When J=0.018256, b=0.168599, kt=1.595349, kb=1.56749, R=1.746171, and L is assumed to be 0. All SI units. V = 6v DC.

   After calculating:

$$\omega_{rpm\ arduino\ mean} = 17.40$$
$$\omega_{rpm\ ideal} = 17.56$$
$$error = 0.91\%$$

Task 3: The position growing speed can be known as motor ω

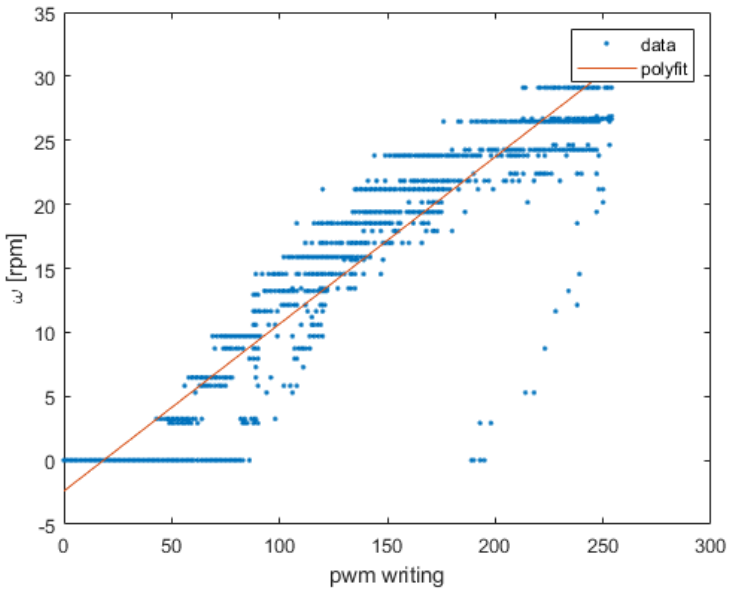


Figure 2: motor speed with different duty cycle

With pwm writing increasing, the speed ω increases, indicating the motor position grows fast.

Assume tf function as $\frac{\omega}{V} = \frac{K}{\tau * S+1}$, when K is DC gain, τ is time constant. From the experimental results, $\frac{\omega_{rpm}}{V} = \frac{18.64}{S+3.377}$, when $K = 5.5197$, $\tau = 1/3.377$.

The simulation result is shown.

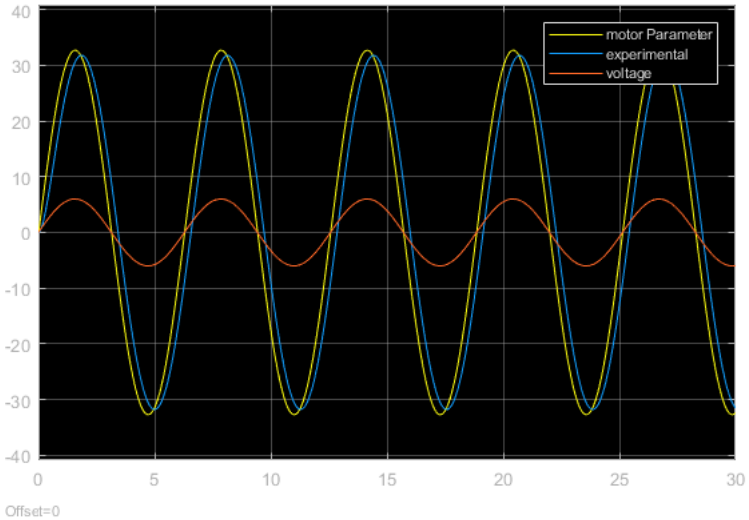


Figure 3: simulation result for both experimental parameters and motor parameters

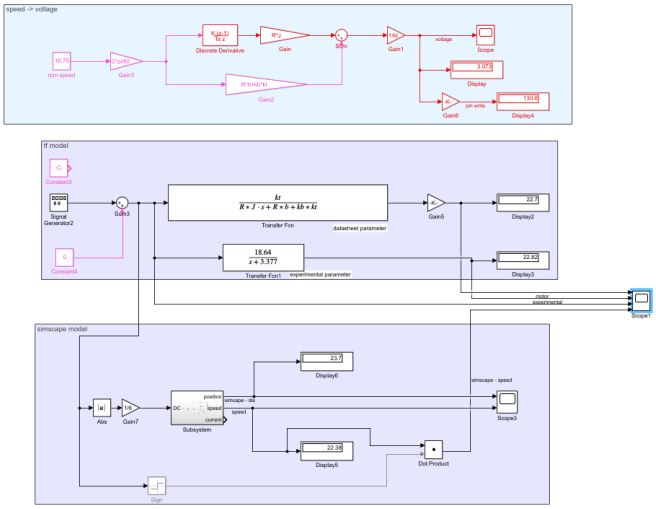


Figure 4: TF model and Simscape model

The simulation results of TF model and Simscape model are shown:

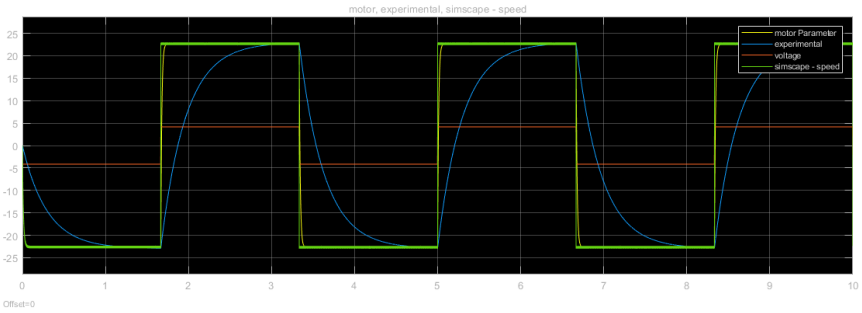


Figure 5: tf model and Simscape model simulation results

Task 4:

Figure1 has shown the reading from potentiometer: parameters named pot reading.

Task 6:

Arduino speed tracking result:

See attached video [speedControl.mp4]

Simulink simulation result:

The Arduino dataset is [aD2.xlsx]

The experimental TF model has already been proven it is a good model for the Arduino-motor system, seen figure 2,3 and5.

Main Tasks-Simulink

Task 2,4,5

Task 4:

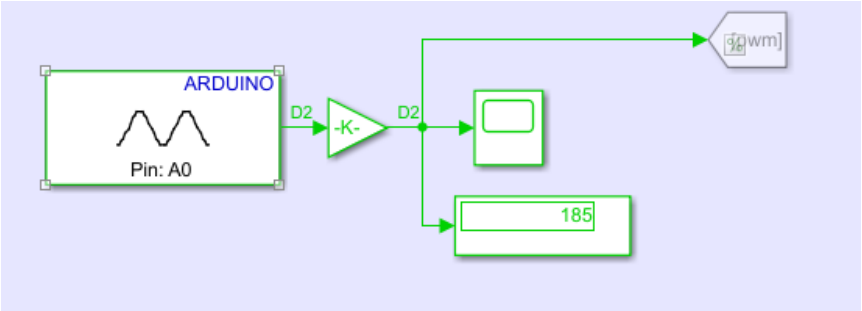


Figure 6: simulink potentiometer reading

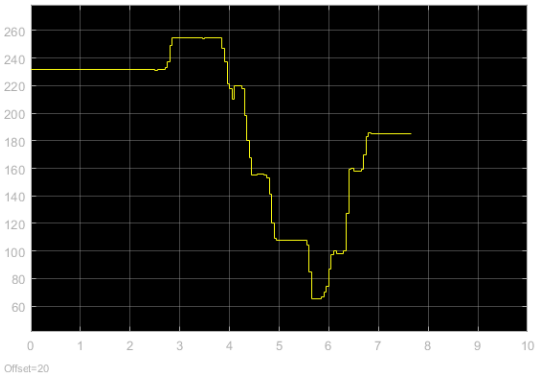


Figure 7: the potentiometer reading after gain block [uint8]

Task 5:

Simulink position tracking result:

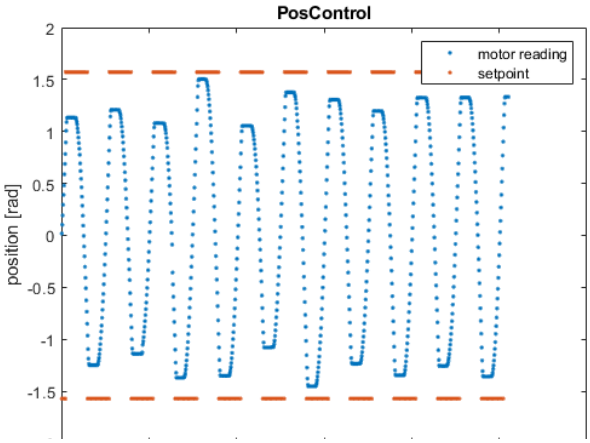


Figure 8: position track

Also see attached video [posControl.mp4]

The are some delay happening the Simulink running process. I am using IO mode in this project. The possible reason is IO mode is slower than external mode ie deploy

The MATLAB dataset is [posControlData2.mat]

The parameter file is [simulinkParameters.m]

Simulink simulation result:

The H bridge block are not working on my Simulink so I try integral speed products voltage sign them getting bias also integral forever :)

Task 2:

Task 6 shown the encoder reading working well.

The error is close to 10%.

Bouns Tasks

1. FFT:

Since my sampling time is large as 0.09s in Arduino, ie 11Hz. The possible high frequency noise is not shown in this FFT plot.

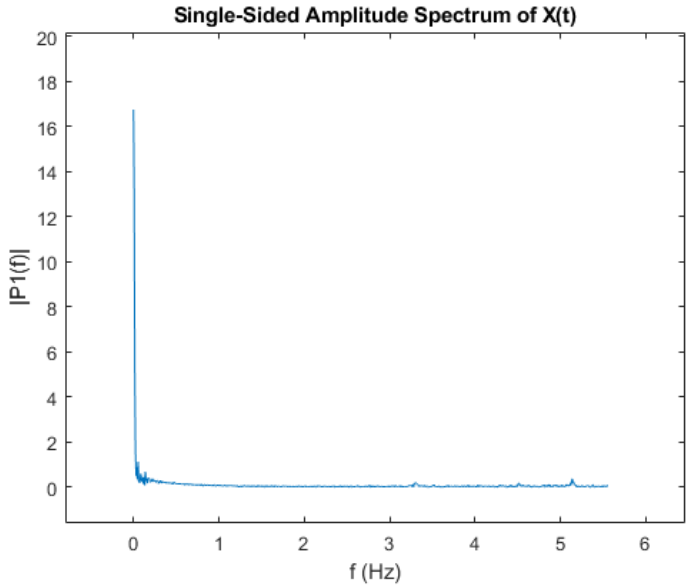


Figure 9: frequency analysis

2. Low pass filter

The average filter will work fine:

$$value_f = value[N] * 0.926 + value[N-1] * 0.074$$

3.4 PID control

All the previous Controller are with PID controller:

For Arduino: kp = 0.2, ki = 3, kd = 0;

For Simulink : kp = 0.12, ki = 0.71, kd = 0;