Encoder Speed Measurement and Low Pass Filter

1-Encoder Speed Measurement

In this experiment, we will examine the encoder used to measure the speed of the motor and the functions of the low-pass filter used before taking the data. Encoder is an electromechanical device that generates a digital (digital) electrical (pulse) signal in response to the movement of the motor shaft to which it is connected. It is also a sensing device that monitors the current positions of the connected shaft and provides feedback. The motor used in Hello 1.0 DC Motor set is a Dc motor with encoder. Therefore, it is advantageous to use the encoder in speed measurement. The encoder and reducer part of the motor used is shown in **Figure 1**.

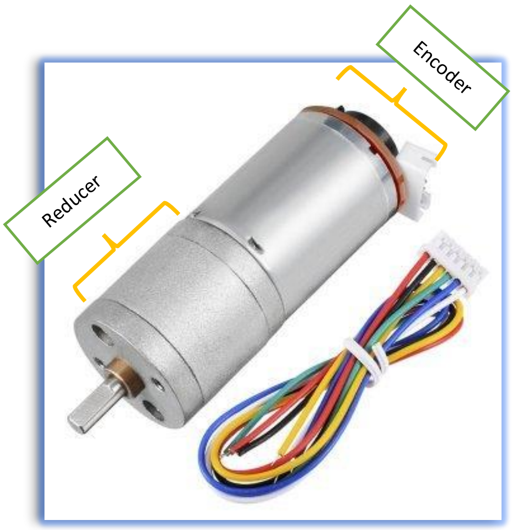

Figure 1: DC Motor

9.7: 1 Metal Gearmotor 25Dx63L mm HP 12V has an integrated CPR quad encoder on the motor shaft with 48 CPR Encoder providing 464.64 counts per revolution of the gearbox output shaft.

[redacted]

The box above contains technical information about the engine. CPR (Count Per Revolution) is the number counted per revolution. It is stated that this number is 48. The value to be read from the encoder as a result of one revolution of the motor is given as 464.64. Because of this the motor has a 9.7: 1 reducer. We can compare the gearbox structure to the wheel structure shown in **Figure 2**. When one wheel makes a full turn, the other will make 9.7 turns, which causes loss of speed and gain from force.

Our laboratory model is shown in **Figure 3**. First, double click on the model to enter the model.

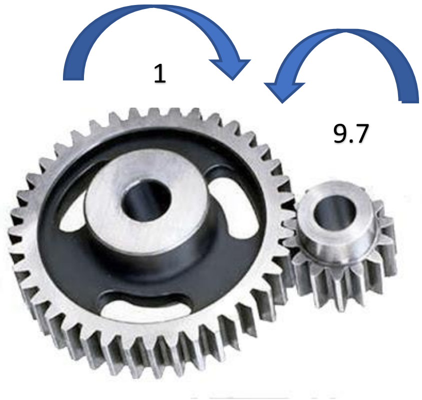

Figure 2: Gear System

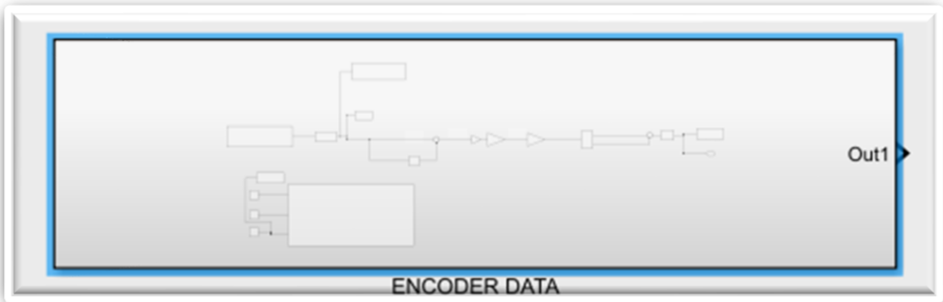

Figure 3: Lab Model

QUESTİONS

- There are 2 types of encoder types used in signal detection. One of them is the optical encoder and the other is the magnetic encoder shown in **Figure 4**. Hello 1.0 DC Motor uses magnetic encoder. Explain the working mechanism of the magnetic encoder by researching.

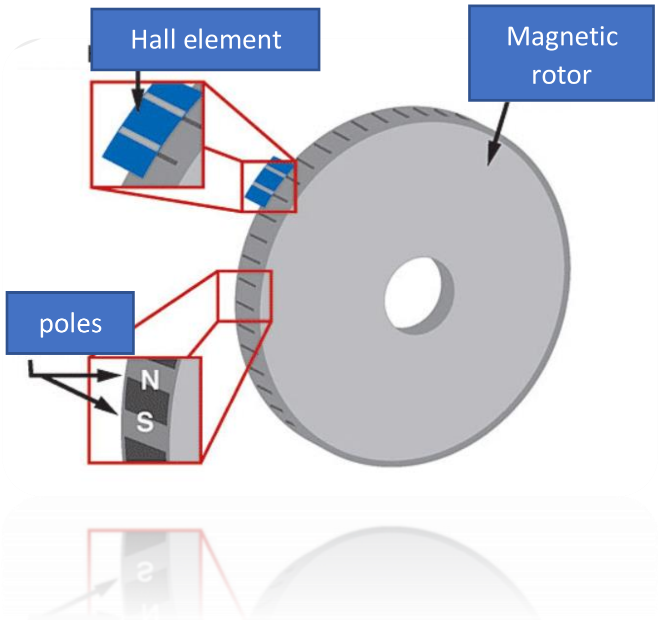

Figure 4: Magnetic Encoder

- Bring the Hello 1.0 block to the location shown in **Figure 5**. This is the part where data is read from the encoder. Thus, when you start your engine, you will be able to read the data from the encoder according to the rotation number. After giving the left input1, the right input 0 and the motor input 50, run the simulation for 5 seconds. Look at the total number of encoders per second from the graph and calculate how many rotations the motor takes. Add your chart.

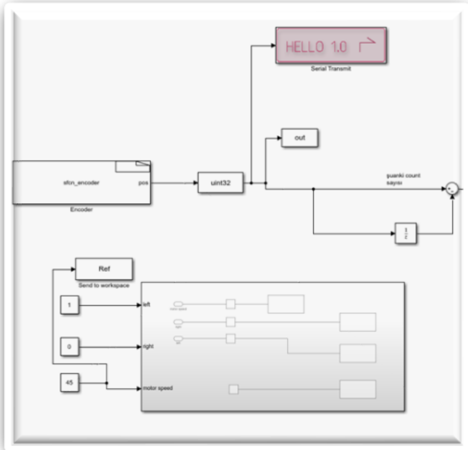

Figure 5: Block Diagram

As an example, 45 is given to the engine input and the simulation is run for 5 seconds and the graphic shown in **Figure 6** is obtained. Encoder data came as 6030. It is also available that 464.64 cpr occurs in one revolution of the engine. In that case;
6030 = 464.64 * number of turns
number of turns = 12.97 revolutions

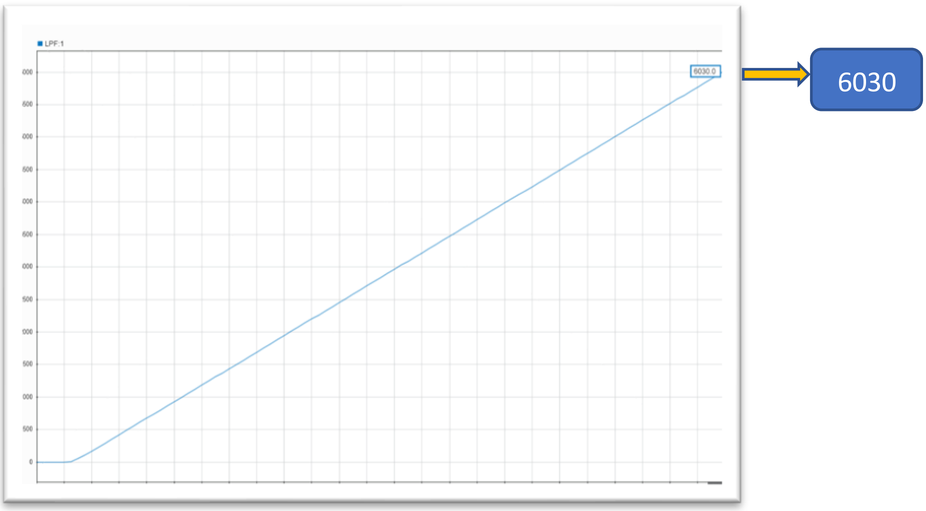

Figure 6: Data from Encoder

- Place the Hello 1.0 block in the place shown in **Figure 7**. The block here is the differentiation block. Run the simulation for 5 seconds without changing any settings and add your graphical output and interpret what the derivation block is for. While commenting;
  Instantaneous count number = You can take advantage of the connotation in the present - past moment expression
  The number 0.05 that we entered in the Serial Recive block in the Settings A section represents the unit time. You observed the count number in 5 seconds. How many counts do you expect to be in a time frame of 0.05 seconds?

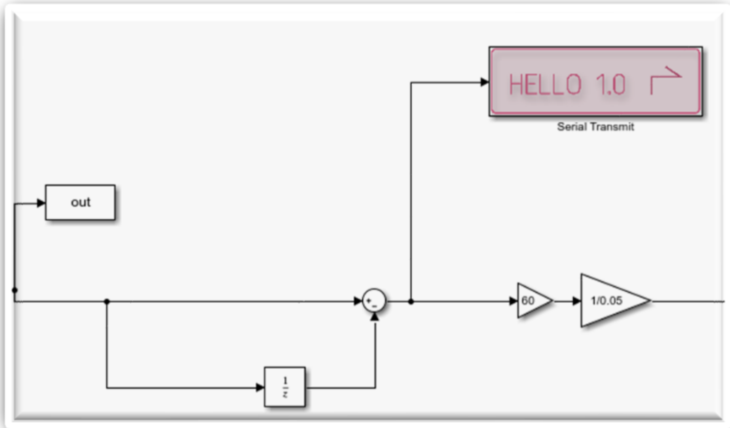

Figure 7: Block Daigram

- Bring the Hello 1.0 block to the location shown in **Figure 8**. This block shows the count value in a minute. Run the simulation without any changes and add your chart. In the first question of the experiment sheet, you found the count value that comes in 5 seconds. Estimate how many counts will be in 1 minute by making a transaction and compare it with your graphical output.

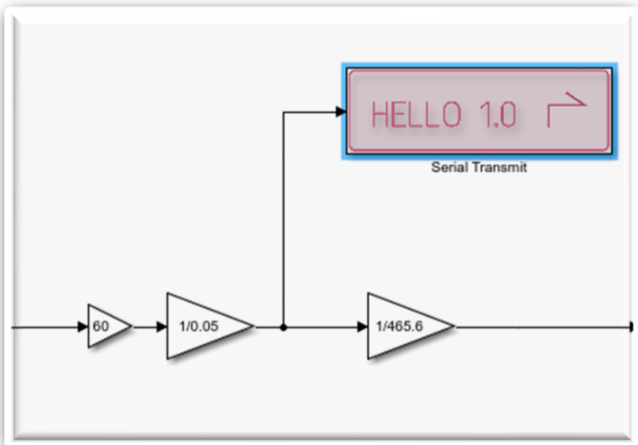

Figure 8: Block Diagram

For 45 pwm motor input, 6030 count count was obtained in 5 seconds. It is expected to come 60/5 * 6030 counts in 1 minute, that is, 72360. As seen in **Figure 9**, the number 77198 has arrived. This number can be considered close to the hand calculated number 72360. The engine's opening time was effective in the occurrence of this difference. When the 6030 value was calculated, since the engine was not used before, it was opened a little later than the next attempts, and therefore the total number of counts was less. **Note that you will repeat the same process for 50 pwm motor input.**

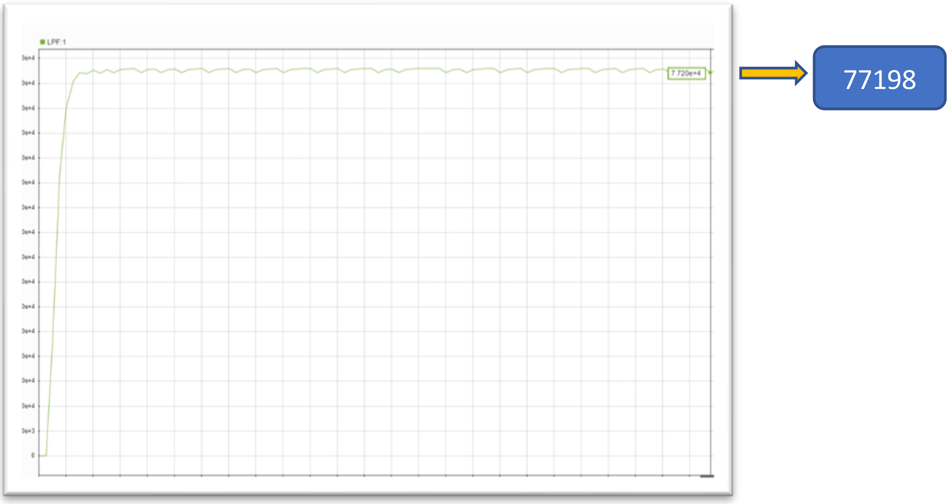

Figure 9: Counts per Minute for 45 PWM

- Bring the Hello 1.0 block to the location shown in **Figure 10**. This block output gives us the number of turns in a minute. In the first question of the experiment sheet, you were asked to calculate the number of rounds in 5 seconds. Using this calculated number, find the number of laps in 1 minute by hand calculation. Run the simulation without making any changes and add your graph and compare the graph result with the manual calculation. The blocks added in the continuation of the simulation are for control purposes.

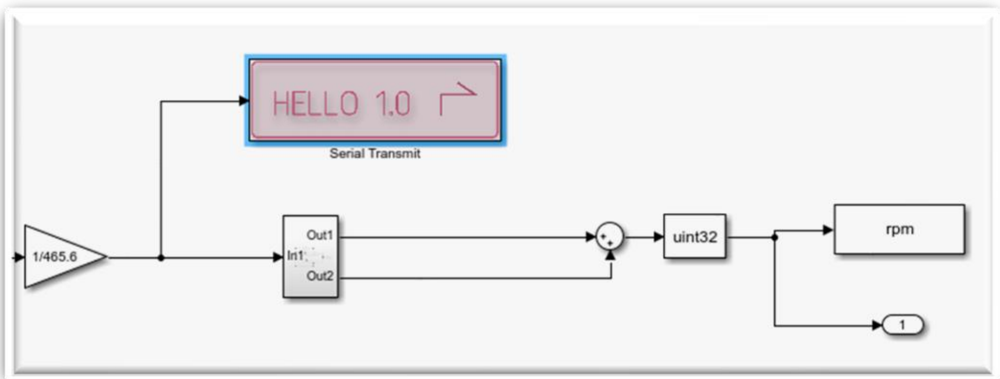

Figure 10: Block Diagram

For 45 pwm motor input, the number of rotations in 5 seconds was calculated as 12.97. In this case, it is calculated that the number of laps in 1 minute is 60/5 * 12.97 and 155.64 laps. As seen in **Figure 11**, this number came as 164. These small deviations between theoretical and experimental results are considered reasonable. **Note that you will repeat the same process for 50 pwm motor input.**

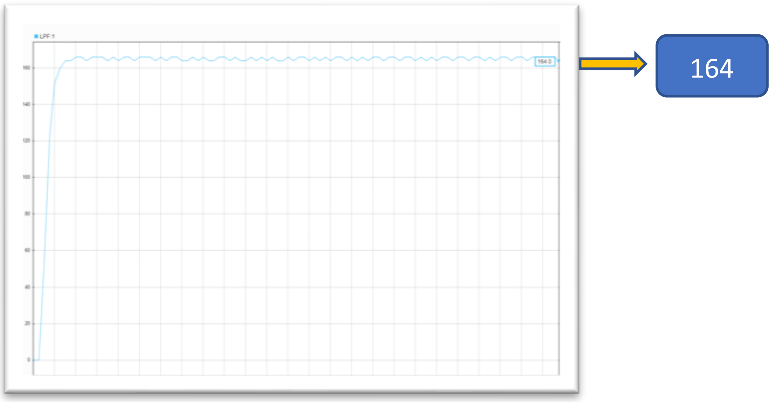

Figure 11: Laps per Minute for 45 PWM

- RPM: Revolutions per Minute, that is the number of cycles in a minute. In the previous question, you found the number of turns in 1 minute by means of an encoder against 50 PWM motor speed input. Thus, you performed the speed reading process with the encoder. In the previous experiment, you learned to read the speed of the motor directly by giving the motor input a constant value. You are asked to give 50 constant to the speed input of the motor, read the speed directly, and compare the resulting graph with the graph you obtained in the previous question.

  As seen in **Figure 12**, when the speed is read directly, 164 rpm output is obtained at 45 pwm speed input. As can be seen in **Figure 11**, it is exactly the same as the speed read on the encoder. **Do not forget to repeat the same process for 50 pwm motor speed input.**

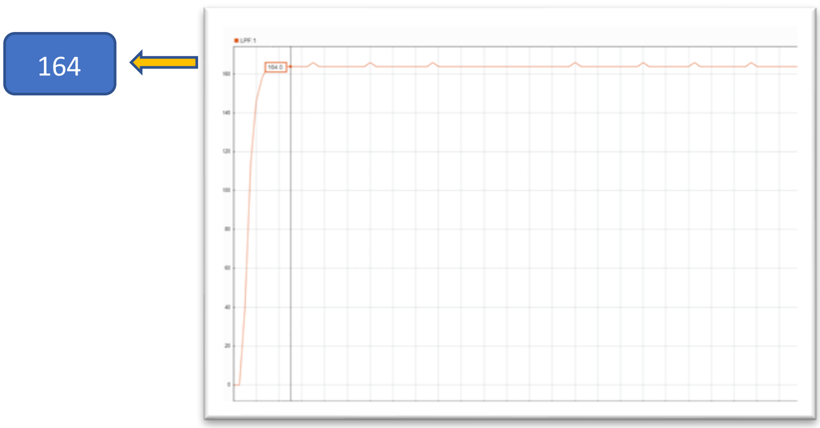

Figure 12: Direc Reading Speed for 45 PWM

2-Low Pass Filter

Electronic filters are circuits that pass some parts of signals with different frequencies and suppress some parts. In this way, unwanted parts and interference on the signal are eliminated. Low-pass filters pass signals below a certain frequency. The frequency response graph of a low-pass filter is shown in **Figure 13**.

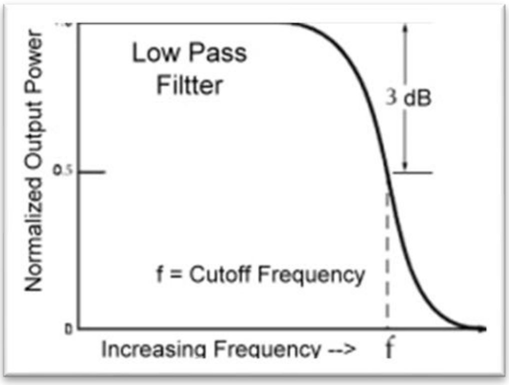

Figure 13: Low Pass Filter Frequency Response

Before the data is presented to the user, it passes through the low pass filter shown in **Figure 14** in the Settings A section.

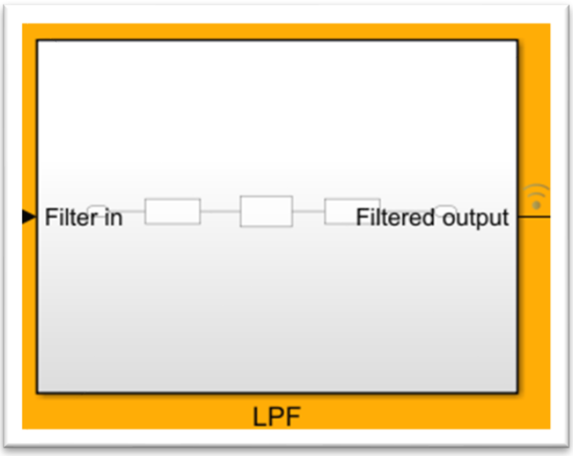

Figure 14: Low Pass Filter

Let's enter it by double-clicking on the low-pass filter.

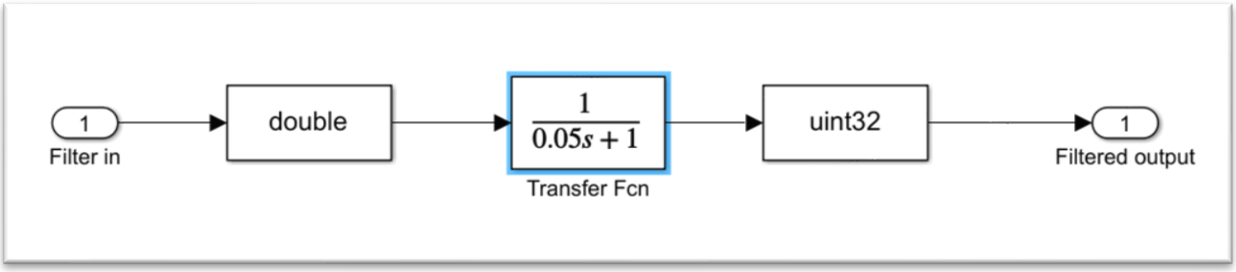

Figure 15: Insede of Low Pass Filter

The transfer function of the low pass filter is given as $\frac{1}{\tau s+1}$. τ means time constant. If τ is increased, the resolution of the data decreases, but there is a delay. If τ is reduced, the data obtained will be closer to the original, but this time the noise may distort the data. Therefore, it is very important to set the correct τ value in filtering. In the filter in the lab model, the value of τ is set as 0.05. This is because the number 0.05 is the sampling time. Thus, there is neither delay nor noise.

QUESTİONS

- First, place the Hello 1.0 block at the output of the encoder data block as seen in **Figure 16**. Run the simulation for 10 seconds by entering 50 for motor speed input, 1 for left input and 0 for right input. Come to Settings A, click on the transfer function twice and change the τ value with 0.2 as shown in **Figure 17** and run the simulation again. Repeat the same process by setting the value of τ to 1. Collect the three graphs you have obtained in a single graph and compare and interpret them.

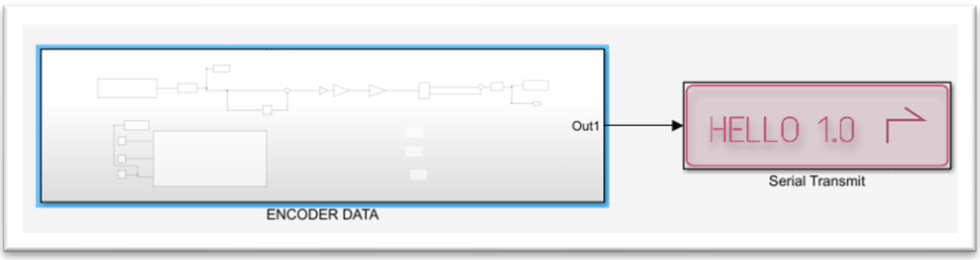

Figure 16: Encoder Block

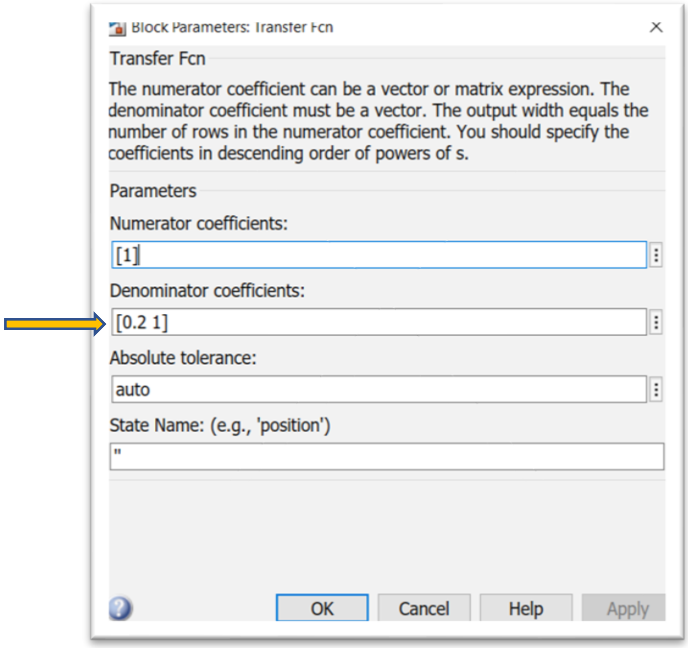

Figure 17: Parameter Settings