1.

a. Degree of accuracy of the encoder largely depends on how well the controller has been tuned. By considering the error anything less than or equal to 5 counts as acceptable, the minimum degree that the motor reference can be set is 4 degree, that is 28 counts. This result was obtained using the following values of Kp and Kd.

Kp = 16.0

Kd = 3.5

Following results are obtained

Before the trajectory:

Reference position set-point in degree: 4
Measured position 'Pm' in count: 24

Current value of 'Kp': 16.5
Current value of 'Kd': 3.5

After completing the trajectory:

Measured position 'Pm' in count: 25
Reference value 'Pr' in count: 28

b. In the PWM configuration, the top value (OCR1A) is set to 10000 and the minimum value of OCR1B needed for the motor to move from the stopped position is 340. So the duty cycle will be 3.4%.

In other words, the motor won't start moving from its stopped postion as along as the duty cycle applied to the dc motor is less than 3.4%

c. The frequency of the encoder task can be calculated by dividing number of counts by the time taken for that many number of counts. And this is calculated as shown below.
We are using an inbuilt timer to record the time before the start of the motor and at the end of rotation of the motor.

I. For duty cycle 25% - To obtain the duty cycle of 25%, we configured following values
```
OCR1A = 10000
OCR1B = 2500
```
Hence the duty cycle is 25%

Now, from the experiment,
Time when motor started the rotation = 5771 ms
Time when motor finished 360 degree rotation = 7381 ms
Number of interrupts in this duration = 2520 counts

Therefore, we can calculate the frequency of encoder task = number of interrupts/time taken
Where,
Number of interrupts = 2520
Time duration = (7381 – 5771) ms = 1610 ms = 1.61 seconds
Frequency = 2520/1.61 = 1.565 kHz

II. Similarly for 75% duty ratio,

```
OCR1A = 10000
OCR1B = 7500
```

Now, from the experiment,
Time when motor started the rotation = 2539 ms
Time when motor finished 360 degree rotation = 3092 ms
Number of interrupts in this duration = 2520 counts
Therefore, we can calculate the frequency of encoder task = number of interrupts/time taken
Where,
Number of interrupts = 2520
Time duration = (3092 – 2539) ms = 553 ms = 0.553 seconds
Frequency = 2520/0.553 = 4.556 kHz

d. When we scale the clock with the preclear above $CLK_{I/O}/64$ we can notice the on and off portion of the wave.
In the current scenario, the PWM TOP setting is 10000 and with the clock of $CLK_{I/O}/64$, we have the frequency,
Considering, $CLK_{I/O}$ = 16000000 Hz

PWM Frequency = 16000000 / (64*10000)
= 25 Hz

We can clearly feel on and off portion of the wave.

2. PD controller has been implemented to maintain the position and it has been tuned by using the following procedure.
Tuning starting with KP=0 and KD=0. Then, KP is increased until the output started overshooting and later KD is increased until the overshoot is reduced to an acceptable level. Here the acceptable level is less than 5 counts.

Note that the method followed in this case is very similar to Ziegler–Nichols tuning, which is used to tune a PD control performing following steps.
- Initially, set Kp (proportional) and Kd (derivative) gains to zero.
- Increase Kp until it reaches the ultimate gain (gain at which the output of the control loop has stable and consistent oscillations called ultimate gain), say Ku.

- Note down the oscillation period Tu.
- Now, set Kp and Kd gains using following equations.
- Kp = 0.8*Ku
- Kd = Tu/8

Observations:

1. Observations while tuning Kp:
    a. Increasing Kp will reduce the steady state error up to some extent.
    b. But after certain limit, increasing Kp will only increase overshoot.
    c. Increasing Kp reduces rise time
2. By keeping Kp fixed, we started increasing Kd and the following observations made:
    a. Kd decreases the overshoot caused by Kp.
    b. Kd also reduces settling time.
    c. The response of the PD controller gives a faster rising system process value than only the P controller
    d. The derivative term (D) gives an addition from the rate of change in the error to the system control input.
    e. A rapid change in the error will give an addition to the system control input.
    f. This improves the response to a sudden change in the system state or reference value.
    g. D term essentially behaves as a high pass filter on the error signal and thus easily introduces instability in a system and make it more sensitive to noise.
3. P and D gain values obtained at which the error was less than 5 counts and the there was no oscillation in the behavior. Values obtained are
    - **Kp = 20**
    - **Kd = 3.5**

3. One of the main thing we note here is that the `OCR1B,` which decides the duty cycle is 16-bit register and the value assigned to it cannot exceed 65536. If exceeds, causes an integer overflow which should be avoided. This scenario is possible, especially for higher values of Kp which causes undefined behavior. To handle this, we need to limit the values that are assigned to `OCR1B` by using cutoff limit.

a. Experiment with **LOW** gain:
Kp = 2.5 and **Kd = 0.5**

Observation:

i. Rotation starts relatively slow and as the error between the target and the current position becomes less and less, the speed reduces and the motor stops with relatively high error counts – 88 counts. Values logged as recorded below.

```
Current value of kp: 2.5
Current value of kd: .5
Measured position 'Pm' in count: 4952
Reference position in count: 5040
Reference position in degree: 720
```

ii. Controller doesn't have enough gain to reach the reference position. So motor stops with considerable amount of error.
iii. Similar behavior has been observed for both the case
    a. With reference position of 2 full rotation from the current position, I.e. 720 degree or 5040 count.
    b. With reference position of 5 degree from the current position or 35 encoder count.
iv. The gain values should be increased to reduce the error.

b. Experiment with **HIGH** gain:
Kp = 50 and **Kd =50**

Observation:

i. Using a too large P term with Kd gives an unstable system and it results in continue oscillation which never settles.
ii. The motor moves back and forth about the reference position.
iii. Similar behavior has been observed for both the case but the amplitude of oscillation was less for the reference position of 5 degree.
    a. With reference position of 2 full rotation from the current position, I.e. 720 degree or 5040 count.
    b. With reference position of 5 degree from the current position or 35 encoder count.
iv. One of the main thing we note here is that the `OCR1B,` which decides the duty cycle is 16-bit register and the value assigned to it cannot exceed 65536. If exceeds, causes an integer overflow. To avoid this, maximum cutoff limit has been assigned to the controller.
v. Gain values should be brought down to reduce the oscillation.

c. Experiment with **IDEAL** gain:
Kp = 20 and **Kd =3.5**

Observation:

i. Above values result in a very good response of the system and following values are obtained
    a. 720 as set point:
    Reference position in degree: 720
    Reference position in count: 5040
    Current value of kp: 20.0
    Current value of kd: 3.5
    Measured position 'Pm' in count: 5039

    b. 5 degree from the current position
    Reference position in degree: 5
    Reference position in count: 35
    Current value of kp: 20.0
    Current value of kd: 3.5
    Measured position 'Pm' in count: 34

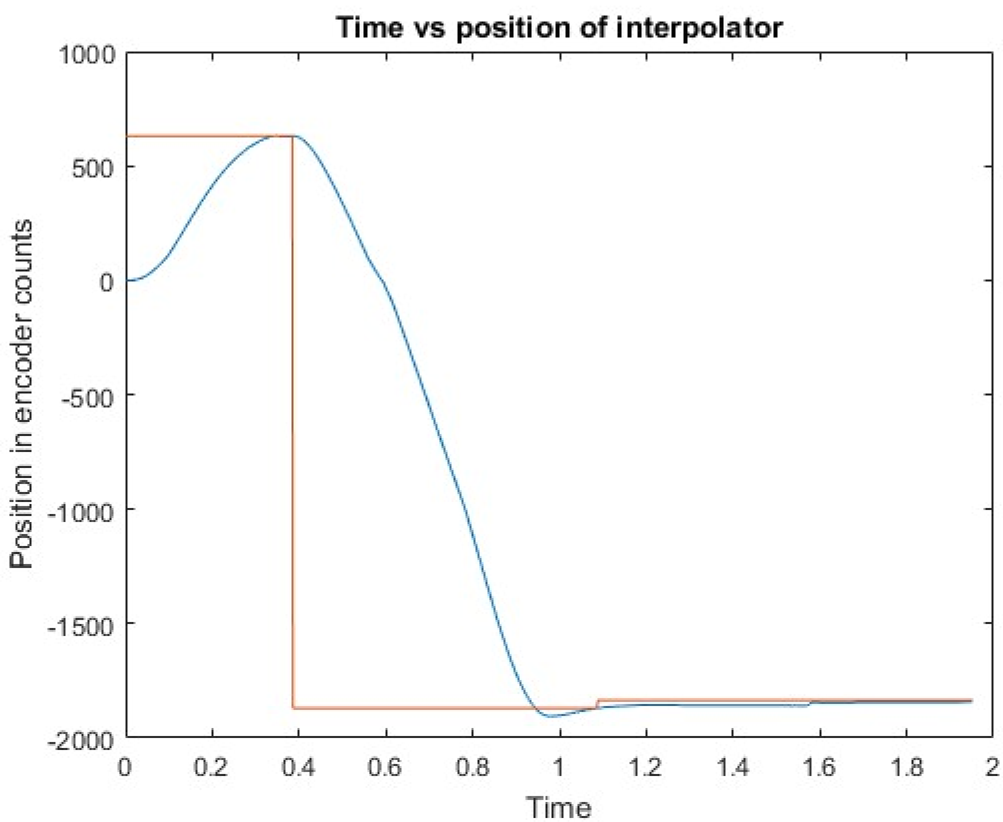
ii. Error in reference position and measured position is only 1 count, which is less than 0.15 degree. So these values are IDEAL for this application.

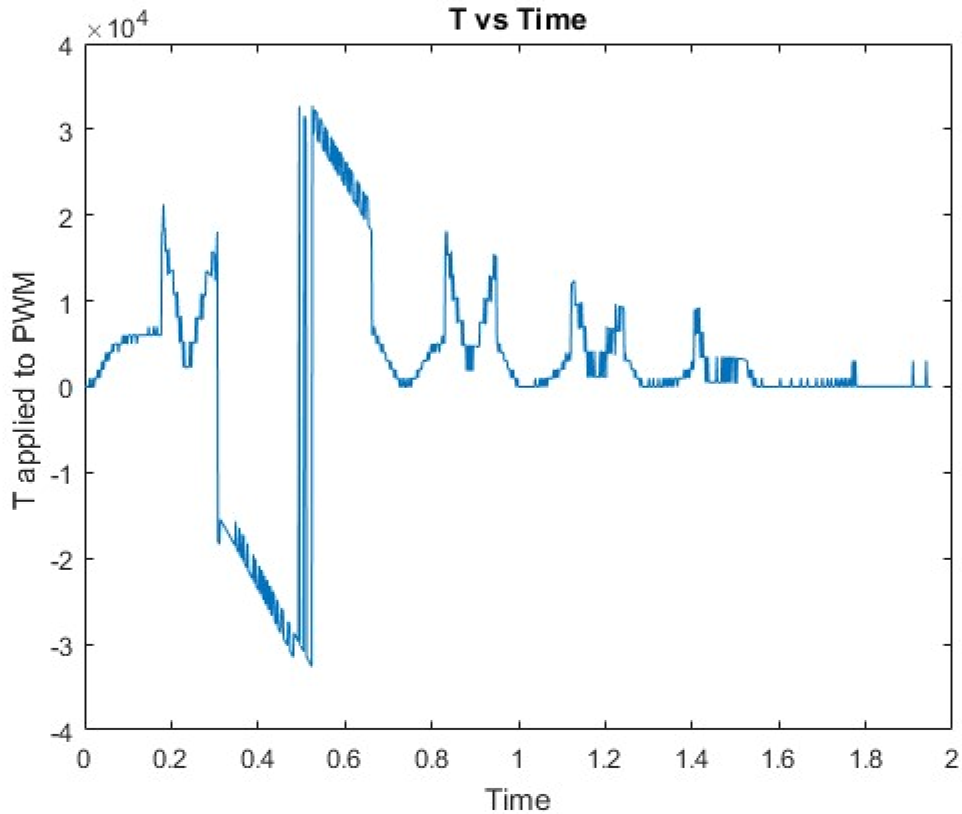
d. Now, by slowing down the controller task to 10 Hz both 720 degree and 5 degree as set point: Motor is able to reach the reference position successfully but there is a disruption in the motion of the shaft. We can clearly feel the on and off portion of the waveform. However, for the case of 720 degree as a set point, there is no change in the behavior as long as the error is concerned.

4. Interpolator is implemented by to perform following steps
a. Step 1 : Rotate the motor forward 90 degrees
b. Step 2 : Held for 0.5 seconds
c. Step 3 : Rotate in reverse for 360 degree
d. Step 4 : Held for 0.5 seconds
e. Step 5 : Rotated forward for 5 degrees

Measured position, Reference position vs time graph is plotted from the data obtained and they are as shown below

- With normal speed Kp = 12 and kd = 3.0





n

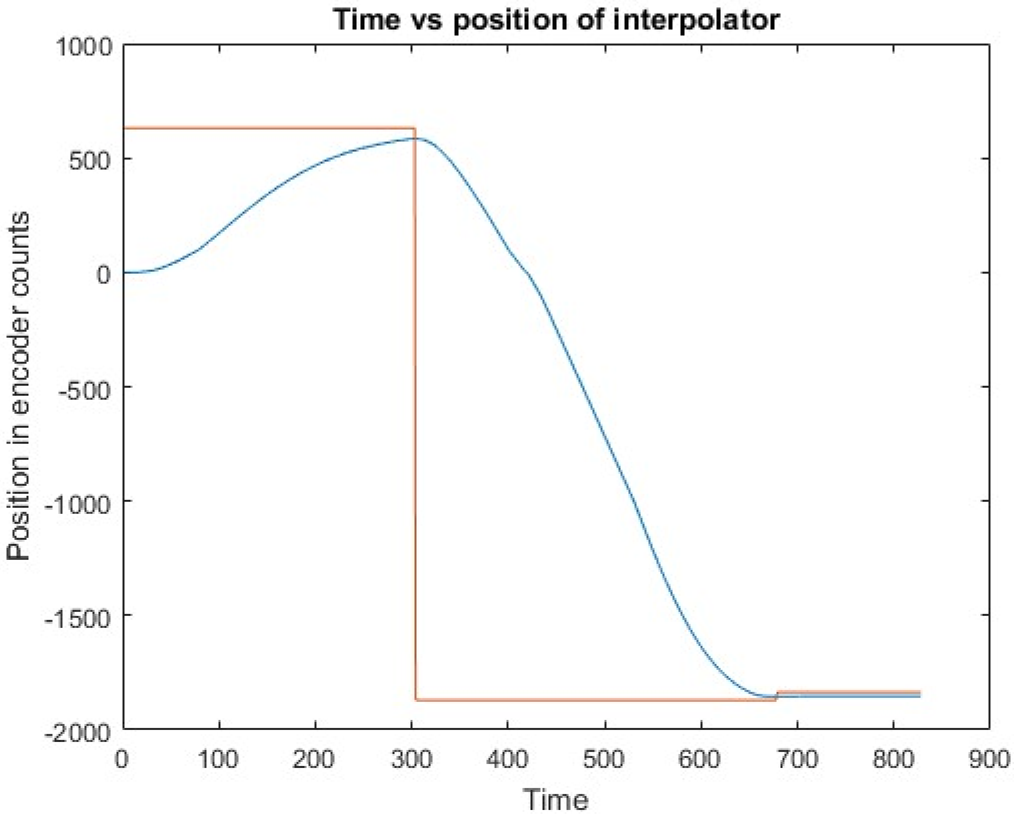
As we can see from the above two plots, initially when the error is high, the controller output starts increasing. The presence of Kp makes sure that the motor speeds up in the beginning and the position approaches the set point.

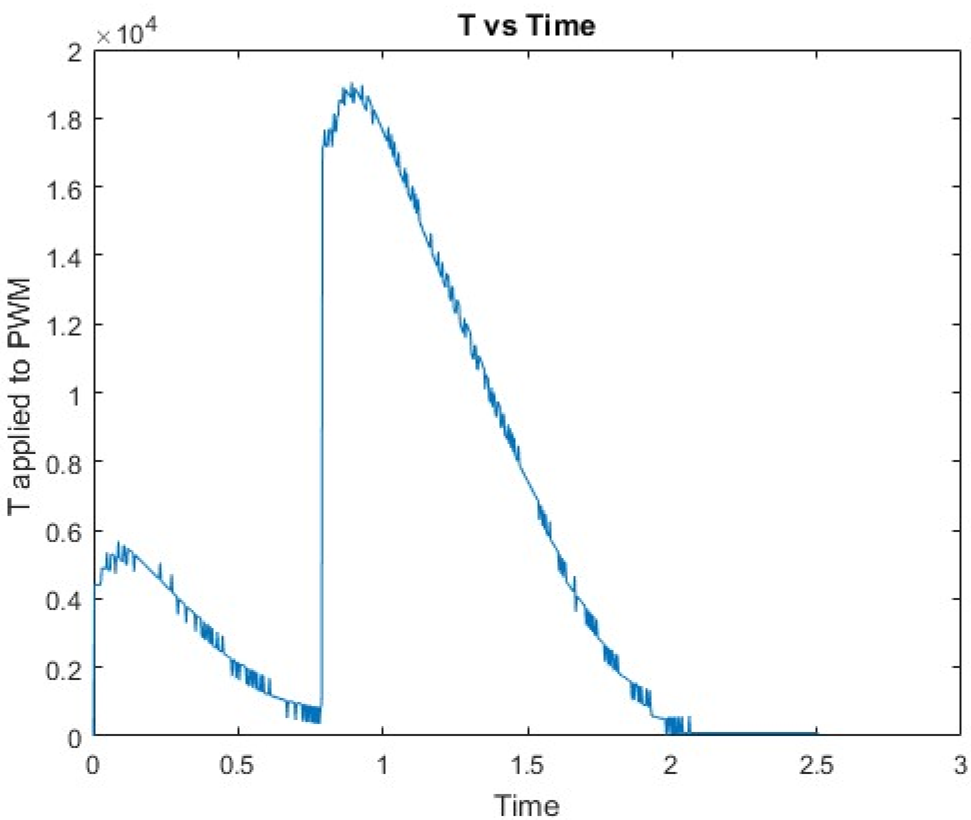
Since the subsequent target are stored in buffer, the next target executes only after the first target is reached. With very slow controllers and low gain values, the position sometimes doesn't reach the target with acceptable amount of error. In that case the motor stops with higher amount of error which causes motor to stop and not continue with the interpolation.

With small values of Kp and Kd, all the target positions reached with some amount of error.

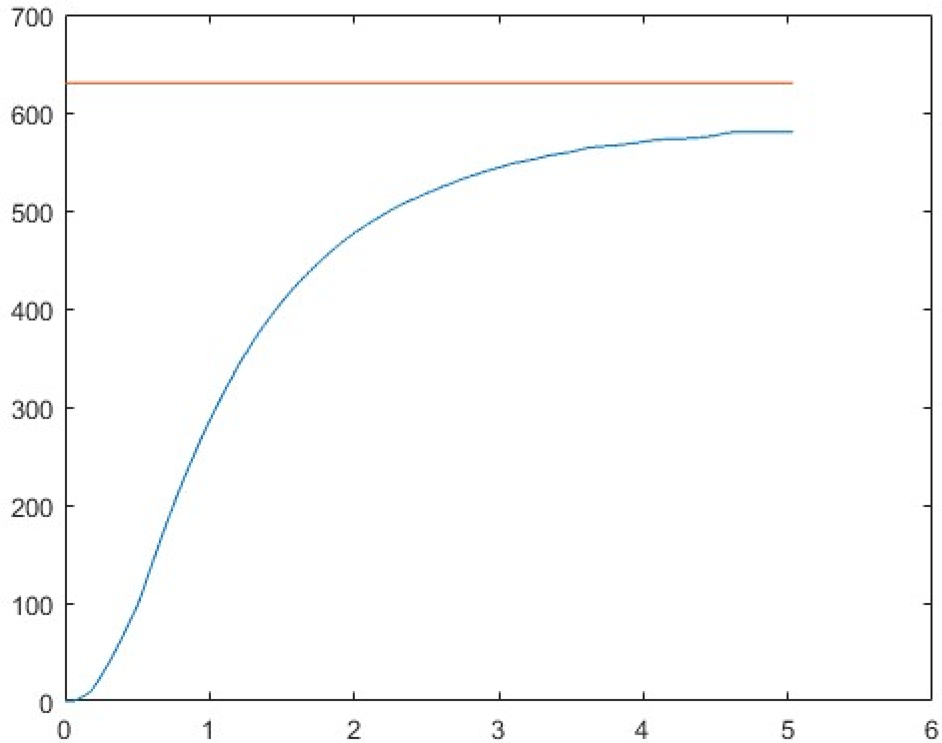
Also, we must note that the controller needs to be tuned as per the application. The controller tuned for one particular scenario might not perfectly match the other scenario. So, the controller that we tuned for earlier question was slightly adjusted to remove some overshoots.

Response with Kp = 12 and kd = 3.0


Time vs position of interpolator
Position in encoder counts
Time
1000
500
0
-500
-1000
-1500
-2000
0
100
200
300
400
500
600
700
800
900


T vs Time
T applied to PWM
Time
×10^4
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.5
1
1.5
2
2.5
3

Finally for very low rate of PD controller,



Similarly, T will have similar behavior as the previous document graph where T increase when error is high and it decreases as the position approaches s target.

So we can conclude from the graph that the T value supplied to the PWM of the motor is high when the error is high and it is low when the measured position is close to the target position/reference position.

Also, we need to note that, for lower values of the gain, the error will be higher compared to the ideally tuned controller.