e-Yantra Robotics Competition - 2018

Pollinator Bee

eYRC#1841

Team leader name	**Aakash Hegde**
College	**PES University**
Email	**hegdeaakash@gmail.com**
Date	**13/11/2018**

Q1. Figure 1 is a graph of a PID Controller. (7.5)

a. What do the red line and the green line at 0 signify in the graph?

b. What effects do the Kp, Ki and Kd values have on the wave shown in the graph?

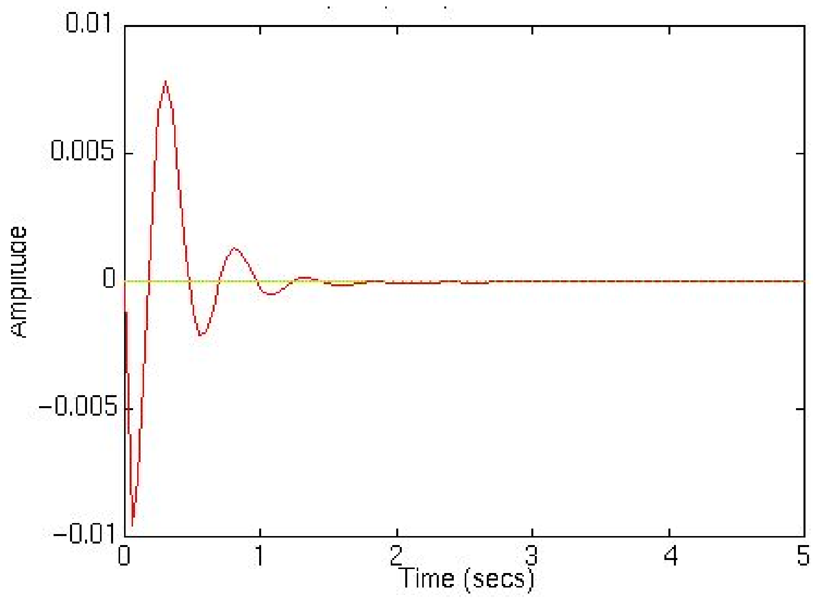

Figure 1: Graph of a PID Controller

Ans: a) The red line indicates the error signal generated.
The green line is the reference or the actual value that the object should tend to ideally.

b)Kp is the linear gain, it decreases the steady state error but it can never eliminate it. This factor creates the oscillations in the graph.

Kd is the differential gain,it predicts the error,hence it can tolerate sudden changes in the error signal. This factor reduces the amplitude of the oscillations of the error signal.

Ki is the integral gain. It takes into consideration the sum of all the previous errors,hence it overcomes steady state error. This factor zeroes in on the reference,it makes the graph ‘settle’ into the correct position.

A combination of all the 3 makes the error equal to zero,the ideal state the object has to achieve.

Figure 2 is a graph of a PID Controller. Notice how the red line sets itself immediately to the desired set point. For a PID controller with a response as shown in Figure 1, what changes should be made to the Kp, Ki and Kd values in order to achieve the graph in Figure 2? Explain your answer in detail.
(7.5)

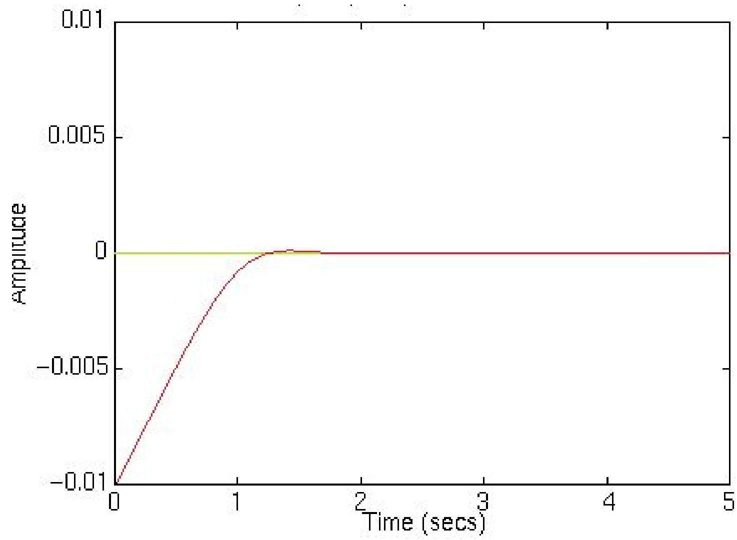

Figure 2: Graph of a PID Controller

Answer:
We run the simulation in V-REP Simulator first.
Using the PID Tune GUI ,we increase the Kp value from zero to a small value and try to obtain a smooth curve by altering this value.
Kp is the linear gain. Kp creates the oscillations in the graph.
Compared to figure 1, there is no oscillation in figure 2, hence no change in value of Kp.

Similarly Kd value is increased from zero and adjusted accordingly to reduce the amplitude of oscillation .
Kd is the differential gain.This factor reduces the amplitude of the error signal. Kd takes into account the rate of change of the error.
Only the Kd value is varied such that there is no overshoot in figure 2 compared to that of figure 1,which has overshoot.

Finally Ki value is changed such that the red line matches the desired response.
Ki is the integral gain,it takes into the sum of all the previous error,hence it overcomes steady state error. This step ensures that there is zero error is attained.**As there is no**

steady state error to correct,the Ki value remains the same as previous

Q2. Given a static set point in a 3D space defined by (x,y,z) coordinates, answer the following questions: (i) how would you move the drone from its current position to this set point and (ii) how would you ensure that the drone is on the set point and it is ready to go to the next way point?
Explain the pseudocode you would implement in detail.
(15)

Answer:
i) At first the error is calculated,which is the difference between the set position(x,y,z) and the current position(x1,y1,z1).
Error=[(x-x1),(y-y1),(z-z1)]
This error term basically comprises of the three values kp,ki,kd which are tuned as per requirements in order to achieve stability.
The pseudo code for output response calibration is given by the formulae below:-

Iterm = (Iterm + error) * Ki
output = Kp*error + Iterm + Kd*(error - previous error)

Here, the Error corresponding to ki and kp are directly multiplied by them respectively.
Whereas the error corresponding to kd also constitutes the previous error. For this the difference between current and previous error is taken and this is multiplied with the corresponding kd value.

The above values are generated until the error becomes zero with the help of a loop, once the difference tends to zero- it signifies that the drone has reached the set position assigned to it and will try to be stable by hovering over the same position with a very less tolerance.

ii) When the drone reaches the set position,the error is zero(minimal) as the current position and set position is the same,the difference between them is almost tends to zero.

When the next way point is set,the error suddenly peaks,this signals that the drone has a new waypoint to go towards,using the PID formula ,it calibrates to achieve zero error,i.e: it has reached the point set.

Q3. In order to achieve the task in Problem Statement 1.1, did your team implement a cascaded PID loop or a parallel PID loop for maintaining the roll, pitch and yaw of the drone? Explain the reason of your choice with advantages and disadvantages over the other option. Use a block diagram to explain your answer in detail.
(20)

Answer: Parallel PID is used to maintain the roll,pitch and yaw of the drone.

In the case of parallel PID;the Kp,Ki and Kd values can be set independently.
This method is easy to implement and gives the freedom to experiment.
The disadvantage is that to produce the factors,it’s very time consuming.

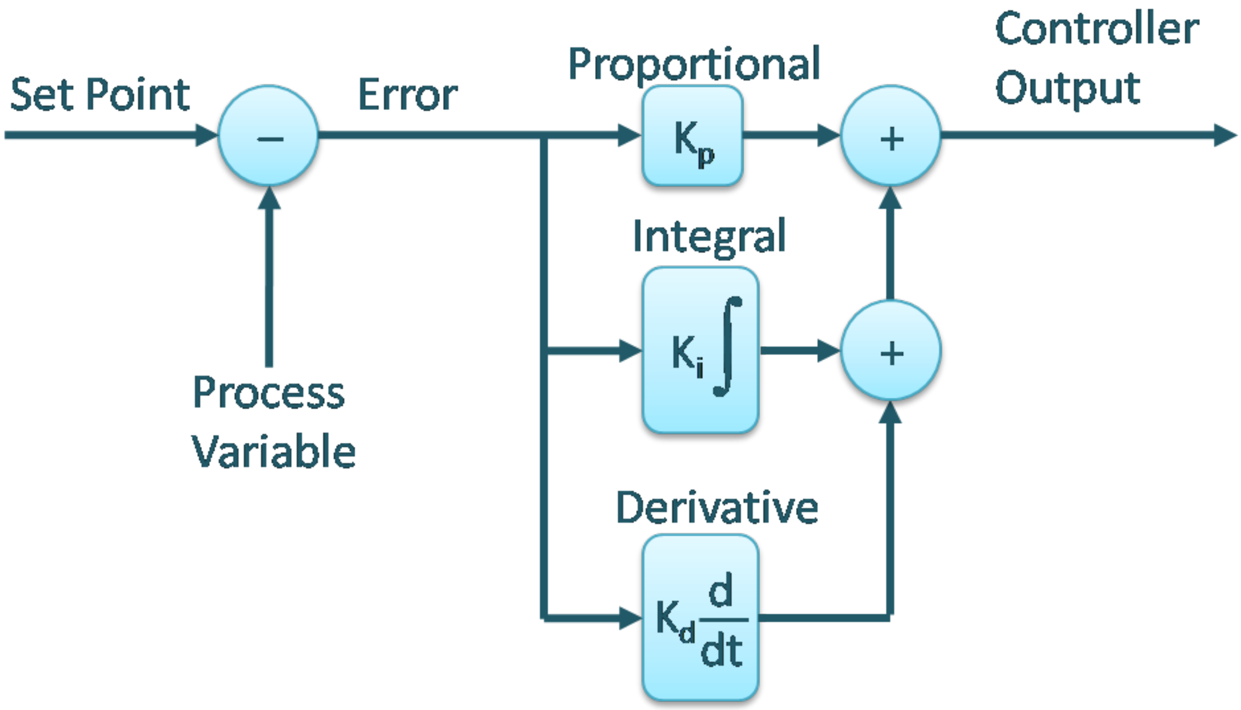

Figure 1

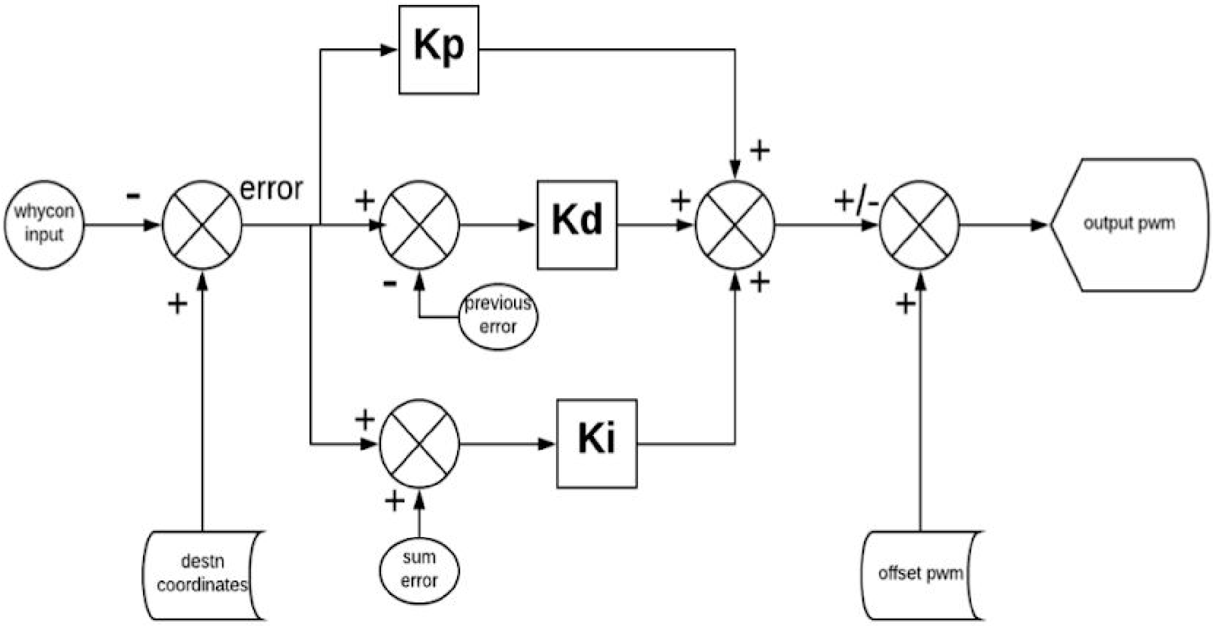

Figure -2

In figure -1 ,we can observe that the Kp,Ki and Kd values are independent of each other , therefore they can be set one at a time,irrespective of the value in the other block(this basically highlights the parallel nature). The adjustment due to this factor is summed together and the path of the drone is corrected to attain a zero error, i.e- the drone reaches the set point from the present point.

In figure 2,the previous error obtained is taken into account, the availability of feedback is what enables for the drone correction to be possible. It is a more detailed explanation of figure 1.
Here we also obtain the destination coordinates, and then compute the error.
The correction factor is determined by the three values kp,ki and kd
With respect to kp, the error value is directly multiplied. For kd, the previous error is also considered and the relative error is multiplied proportionately.
Finally the sum error is multiplied with ki.
These three values are summed up and compared with the offset Pwm to generate the output pwm.