e-Yantra Robotics Competition - 2017

Chaser Drone

1712

Team leader name	Rachen R
College	St. Joseph's College of Engineering
Email	rachenindia2020@gmail.com
Date	28.11.2017

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?**

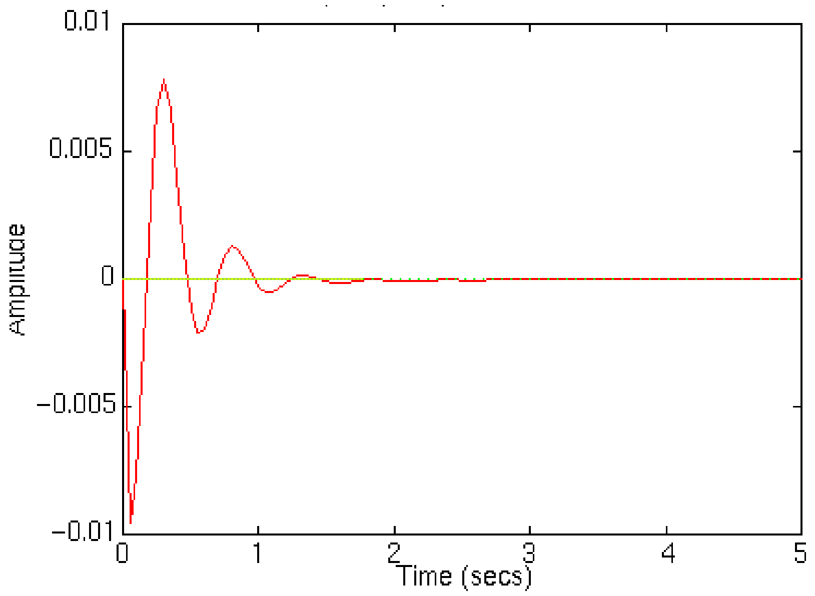


Figure 1: Graph of a PID Controller

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)**

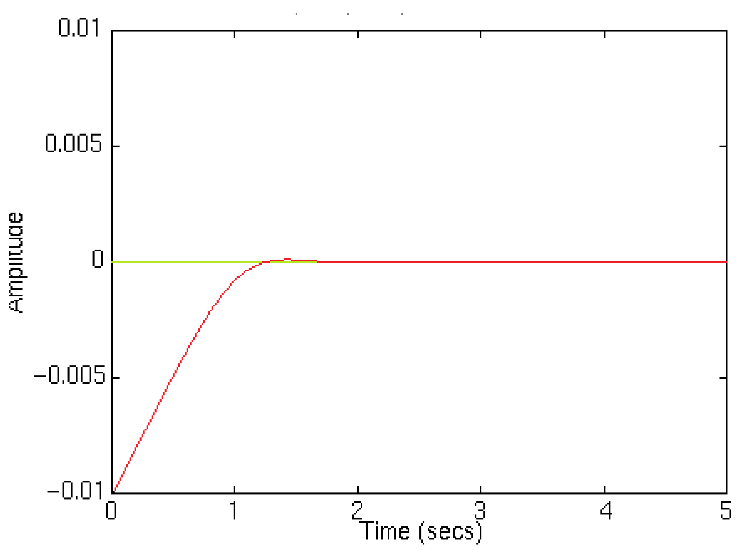


Figure 2: Graph of a PID Controller

Answer:

The red curve represents the output of PID controller and green line represents the set point given to a PID controller. Here, set point at 0 indicates that output of PID controller should follow the set point. Hence, PID output oscillates about set point and finally reaches the set point.

Increasing Kp value decreases the rise time and increases the overshoot in the graph, and vice versa. If Kp value is higher, the PID output will reach a closer value to the set point quickly. However, it can never reach the zero steady state error (error at infinite tine), as it oscillates about the set point and proportional output can exist only if there is an error.

Increasing Ki value eliminates the steady state error. But, increasing it to a larger value increases overshoot in the PID output and settling time (time to reach the steady state value).

Kd value affects only for changes in error. Increasing Kd reduces the overshoot due to Kp and Ki values and is called damping. It also reduces the settling time. However, a large value in Kd can cause over damping.

Since, the response in figure 2 has reached the set point quickly, Kp value has to be increased. This allows a faster rise time. Then, to dampen or reduce the overshoots caused by Kp value, Kd value has to be increased. Then, Ki value has to be increased to reduce the steady state error such that set point is reached more accurately. The order of tuning is:

Kp -> Kd -> Ki

These constants are again fine tuned in the same order until Kp causes sufficient reduction in rise time, Kd reduces overshoot due to Kp and Ki, and Ki eliminates the steady state error.

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:

Pseudo code:

```
Kp, Ki, Kd;
Previous_Error[3], Integral_Error[3];
//list of 3 elements for integral errors and previous errors in x, y, z axis
Set_Point[2][3];            //list of 2 lists each representing a set point, e.g., SetPoint[0]=(x,y,z)
w = 0;                      //Set Point Number
dt = 0.01 ;                 //change in time
tolerance = 0.2;
while(1)
    flag = 0;                              //flag for identifying if current set point is reached
    Current_Position[3] = getWhyconPoses();           //gets current Whycon marker position
    for axis_no in loop from 0 to 2:                  //axis_no := x = 0  ; y = 1;  z = 2
        Error = Set_Point[w][axis_no] – Current_Position[axis_no];
        Integral_Error[axis_no] += Error * dt;
        Derivative_Error = (Error – Previous_Error[axis_no])/dt;
        PID_Output[axis_no] = Kp * Error + Ki * Integral_Error[axis_no] + Kd * Derivative_Error;
        Previous_Error[axis_no] = Current_Position[axis_no];
        if (Error < tolerance):
            flag += 1        //flag incremented if error in each axis position is less than tolerance
    pidPublish(PID_Output);                //Publish PID output to twist message of cmd_vel topic
    if(flag == 3):
        w +=1;     //indicates next set point when flag is 3 as error is less than tolerance for each
axis
    sleep(dt);
```

Explanation:

(i) We can obtain the current position of the drone using its marker coordinates obtained from /whycon/poses topic by subscribing to it. Now, using a PID controller we can control the error in current position of the drone from its target set point (after tuning Kp, Ki, Kd). This PID output is published as twist message to /cmd_vel topic which controls the velocity of drone in their respective axis.

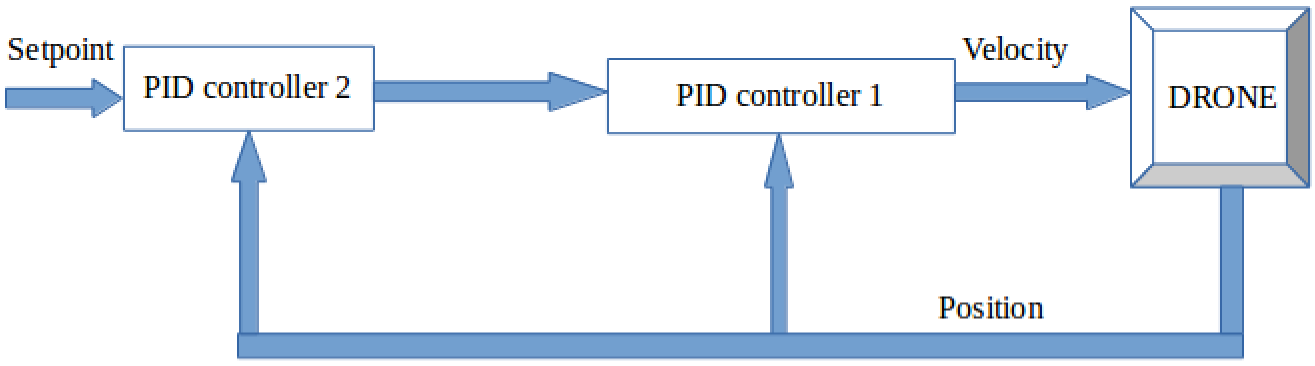
(ii) To ensure that a set point is reached, we check whether error is lesser than a tolerance value in all the 3 axis. If error is lesser, the drone has reached the set point and is ready to go to the next way point. Now, PID starts to control the error for the next set point.

Q3. In order to achieve the task in Problem Statement 1.2, 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:

A cascaded PID control is a form of control where the required output is obtained from the current point using a PID controller whose setpoint is controlled by another PID controller.It consists of two loops-the primary and the secondary loops.The primary loop acts as the outer loop which determines the setpoint for the other loop in accordance to a reference.The secondary loop or the inner loop receives the above determined setpoint and issues the output signal to the system under consideration by comparing with the output of the system.

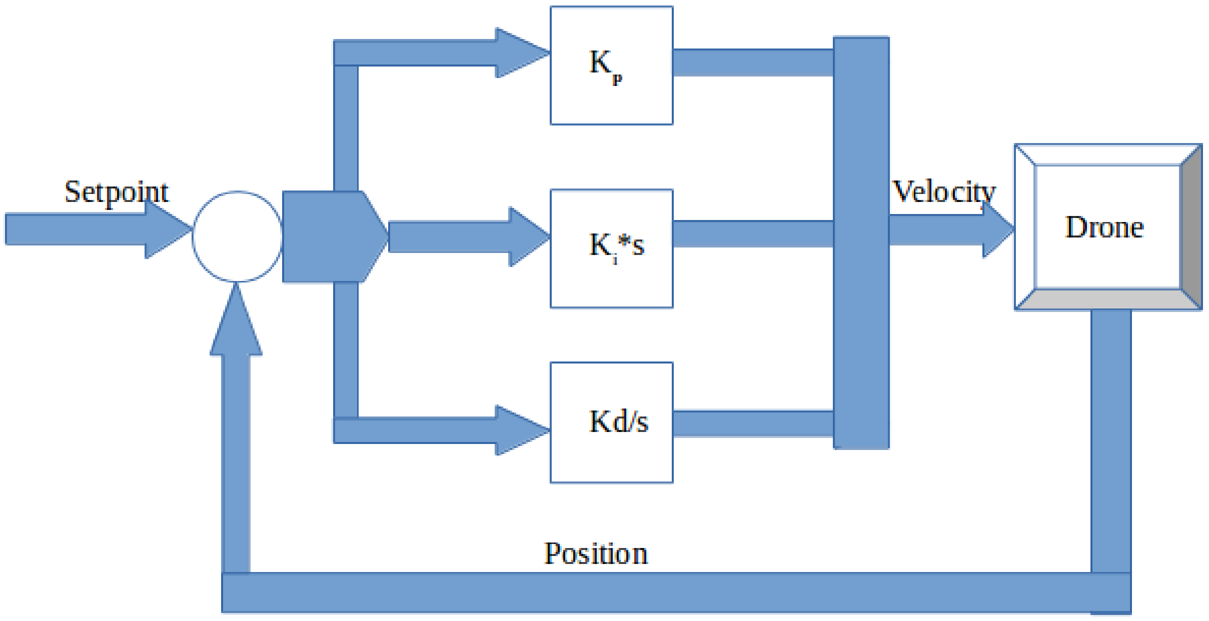
Block diagram:



A cascade PID control is normally used for systems where the required output is a slowly varying quantity and has to be controlled by a dynamically altering input paramaters.The requirement of Cascade control is that the inner loop has to be atleast 5 times faster than the outer loop so that the effect of secondary disturbances are corrected by the inner loop before they affect the primary process.Considering the drone,the primary loop is the one that receives the current position of the drone and determines the next position to be traversed by the drone.The secondary loop publishes the required velocity to reach the next point based on the setpoint provided by the primary loop.The disadvantage with cascade control are that it requires two controllers and twice the tuning process than required for a single PID control mechanism.One more disadvantage is that if the inner and outer loops are operating on the same time frame,then they may compete with each other,thereby tending the drone to unstability.

The type of control implemented in our task is the Parallel PID loop.In this,all the three terms-Proportional,Integral and Derivative are computed independently and the control output is obtained by substituting them in the PID equation.

Block Diagram:



The current coordinates of the drone are compared with the setpoint and the proportional,integral and derivative terms are calculated.The control output which is the velocity is fed to the drone,thus enabling the drone to move to the specified co-ordinates.The parallel PID control is very simple and easy to tune when compared with Cascade PID control.K_p is varied till the drone reaches the setpoint.Next,K_i is varied to reduce the offset.Finally K_d can be adjusted to reduce the oscillations of the drone around the desired point.Thus all the three parameters are varied independently of each other.