e-Yantra Robotics Competition - 2018

Theme and Implementation Analysis – Hungry Bird

#2099

Team leader name	Aniket Sharma
College	Bharati Vidyapeeth's College of Engineering
Email	aniket965.as@gmail.com
Date	01-01-2019

Scope and Preparing the Arena

Q1 a. State the scope of the theme assigned to you. (5)

The Hungry Bird (drone), should perform task of navigating in 3D space which is done by birds routinely, in this theme we are given task to navigate drone avoiding obstacles and passing through fruit trees (hoops) in shortest possible time, which is very common task in air transportation, it can be used in factories to transfer material from one place to another in efficient way. If we make vision sensor moveable as show in below figure(for example vision sensor on ground bot)It can be used in surveillance. It can also be used for gauging a situation of unrest. It can also be used collect data on weather, pollution if proper sensors are attached to it. It can go through areas which are of unknown dangers.

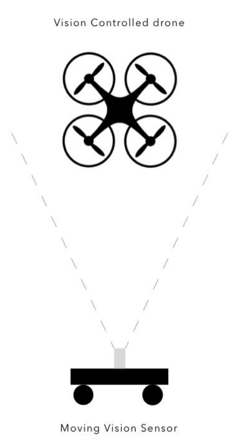
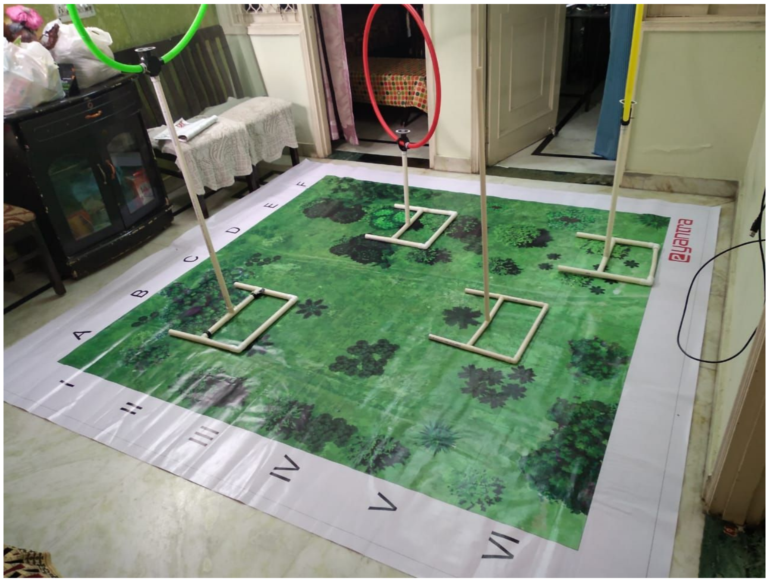
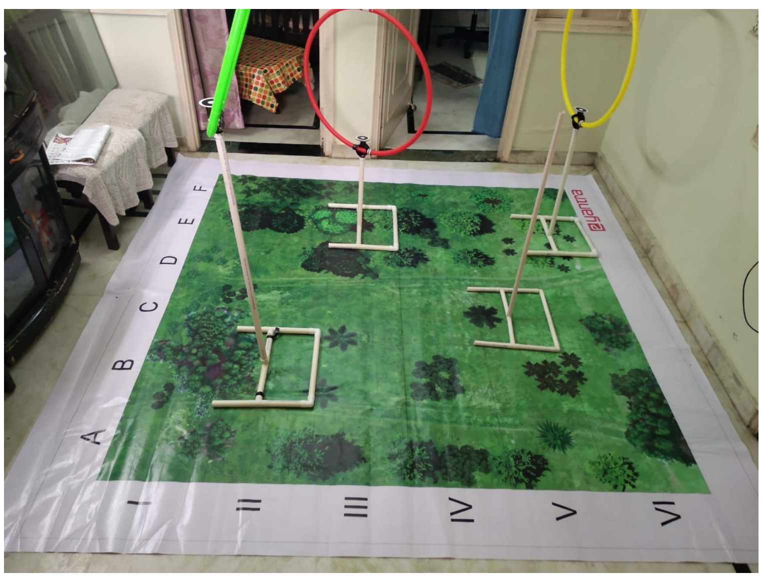


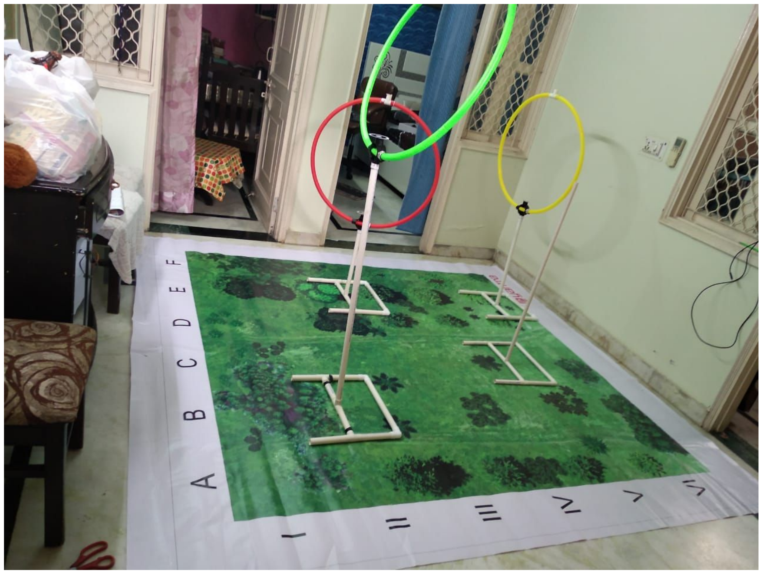
Fig. 0 Example usage of theme

B . Attach the Final Arena Images. **(5)**



Different Images of Arena





(Note we have two more non-food tree just not added in these images to make it less cluttered)

Testing your knowledge (theme analysis and rulebook-related)

Q2. How will you ensure that while tuning the PID value, Drone will not crash? (5)

We are using a string as jugad, we tied a string at bottom of drone with a length such that it can no leave the arena, it is helping in preventing drone to fly away and crashing into walls of the room,

Fig1. Yaw tuning

For Yaw, we hanged drone from upside and kept it close to its set point and then armed drone and changed kp, ki, kd constants for yaw axis.we rotated drone while hanging and used plotjuglar and pid tunner.

Fig 2. Throttle, pitch, roll tuning

For Throttle Drone we tied string at bottom of the drone which helped in preventing drone to fly too up.
For Pitch and roll, we attached 2 strings in both side which prevent drone from moving out from arena since the throttle is tuned the first drone don't move up down too often which helps in tuning

We tuned PID values using PID tuner and visualized using plot juggler provided by eyantra team which helped in tuning in runtime and we used key Events to prevent drone from crashing we created, emergency stop, arming, disarming functions and bind them to key Events to help in decreasing chance of crashing.

```
def key_callback(self,data):
     if data == 70:
          self.akp += 1 # for changing kp
     elif data == 60:
          self.akd -= 1 # for changing kd
     elif data == 10:
          self.disarm() # shortcut to disarm
     elif data == 20:
```

```
self.arm() # shortcut to arm
```

Q3. How will you navigate the drone through the hoops. Do you anticipate to lose input from the overhead camera while the drone is crossing the hoop? If yes, how are you planning to deal with this scenario? (5)

Yes we think it will lose whycon marker while crossing hoop, We will make a different function/state for passing hoops as position and orientation of hoop is not going to change in runtime, so we will take 2 points one just before hoop one just after hoop through the line of center

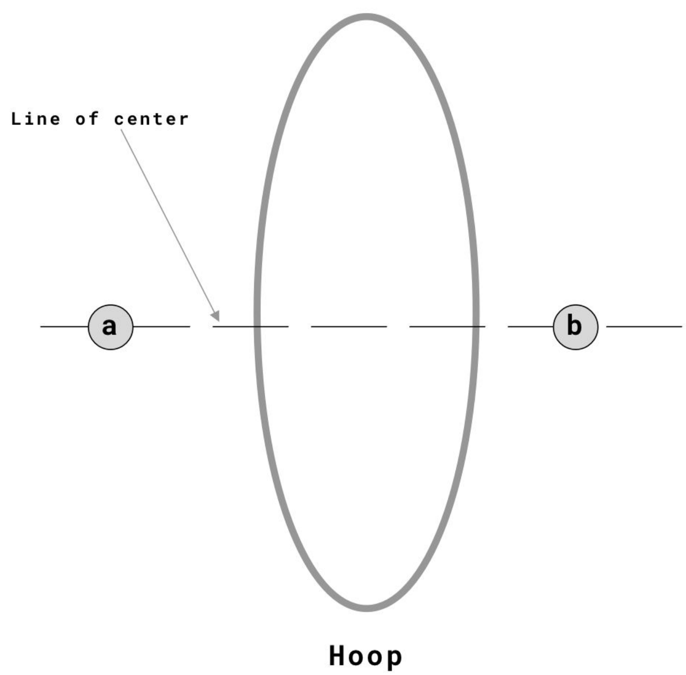


Fig 3. Illustration of strategy

As shown in fig. 3 we will calculate A, and B point and through the center of hoop perpendicular to face of the hoop, and create a function to do this passing accordingly as shown in fig. 4

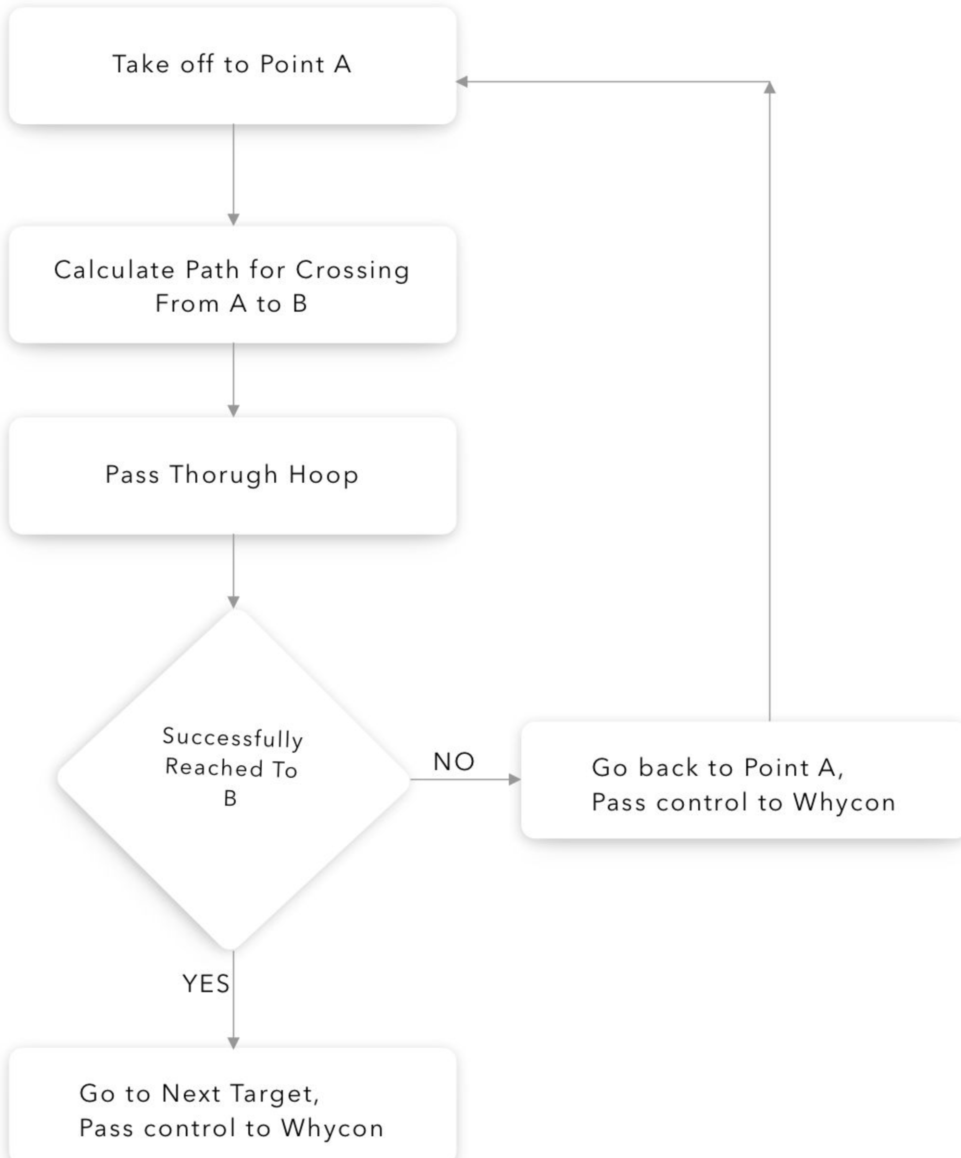


Fig. 4 Flow Chart for passing through Hoops

Firstly our drone will move to a point A in front of the hoop which will be directly in front of a predetermined point which is calculated using path planning algorithm. Then drone will trace the calculated path to point B and if it fails to go through the hoop then it will return to a point calculated using tuned path planner and perform above operation again.

Q4. What is the trend of WhyCon co-ordinates in a single real-world z-plane? (5)

Whycon coordinates in real-world achieve millimeter precision with very high performance but fluctuate in position with a little error continuously. It is capable of efficient real-time detection and precise position estimation of several circular markers in a video stream. It can be used both off-line, as a source of ground-truth for robotics experiments, or online as a component of robotic systems that require real-time, precise position estimation. WhyCon is meant as an alternative to widely used and expensive localization systems.

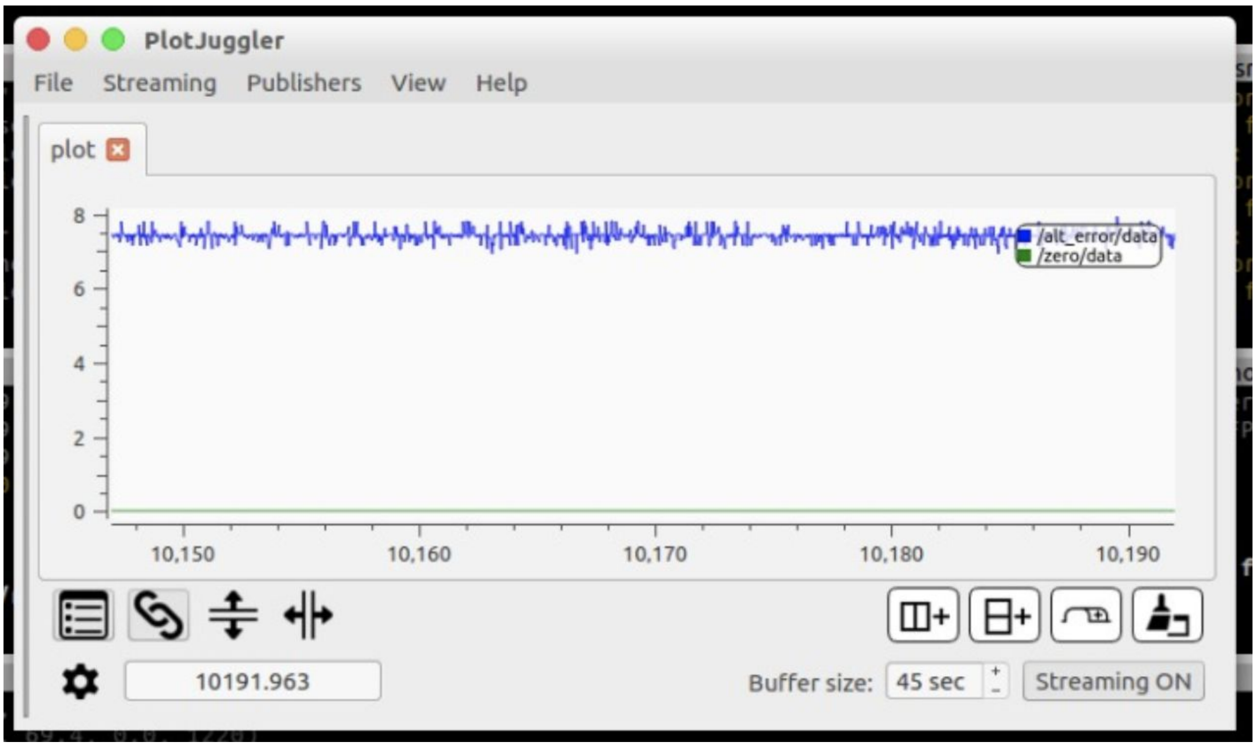


Fig 5. Error fluctuation in steady state of drone in Z axis(alt)

In Real-world z-plane Whycon shows more fluctuation in Z axis than simulation as show in fig 5 it shows error in steady state of drone, to solve this issue we not applying to much ki,kd (very less as compared to kp)

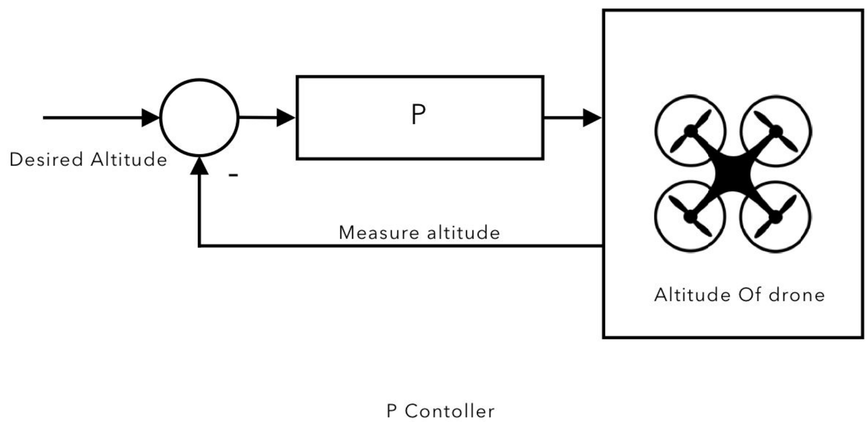


Fig 6. P controller for altitude

This results in only P controller of Drone Rather PID controller for Altitude, also we will try to implement different curve smoothing functions and different control algorithms to see if we can get better performance than only P controller.

Q5. What will be your strategy to earn maximum points in a run? (10)

1. Tune PID as accurate as possible. This will provide better accuracy in localization and hence will decrease chances of collision.
2. Implement different control algorithms and compare in speed and accuracy, choose the best algorithm with best speed and most accuracy, Therefore decreasing T and CP (collision penalty) hence increasing points.
3. Choosing the shortest path among all possible paths in between trees. To achieve this we will use a python node while the drone is moving through waypoints, hence in this ideal time, we will compute next path waypoints, while the drone is moving in the previous path.
4. We will provide required offset to obstacles in Vrep so that it can easily pass trees by avoiding collisions.
5. We will try to implement an algorithm that can classify sudden changes (which can lead drone to go outside the arena and stuck state of the drone to slowly try to go in the previous position.
6. Using optimal time, shortest path and avoiding collisions will save us time hence we will ensure proper landing hence increasing landing bonus point.
7. We will make robust hoop crossing algorithm,hence decreasing Collision penalty hence increasing points.

Algorithm Analysis

Q6. Draw a flowchart illustrating the algorithm you propose to use for theme implementation.
(10)

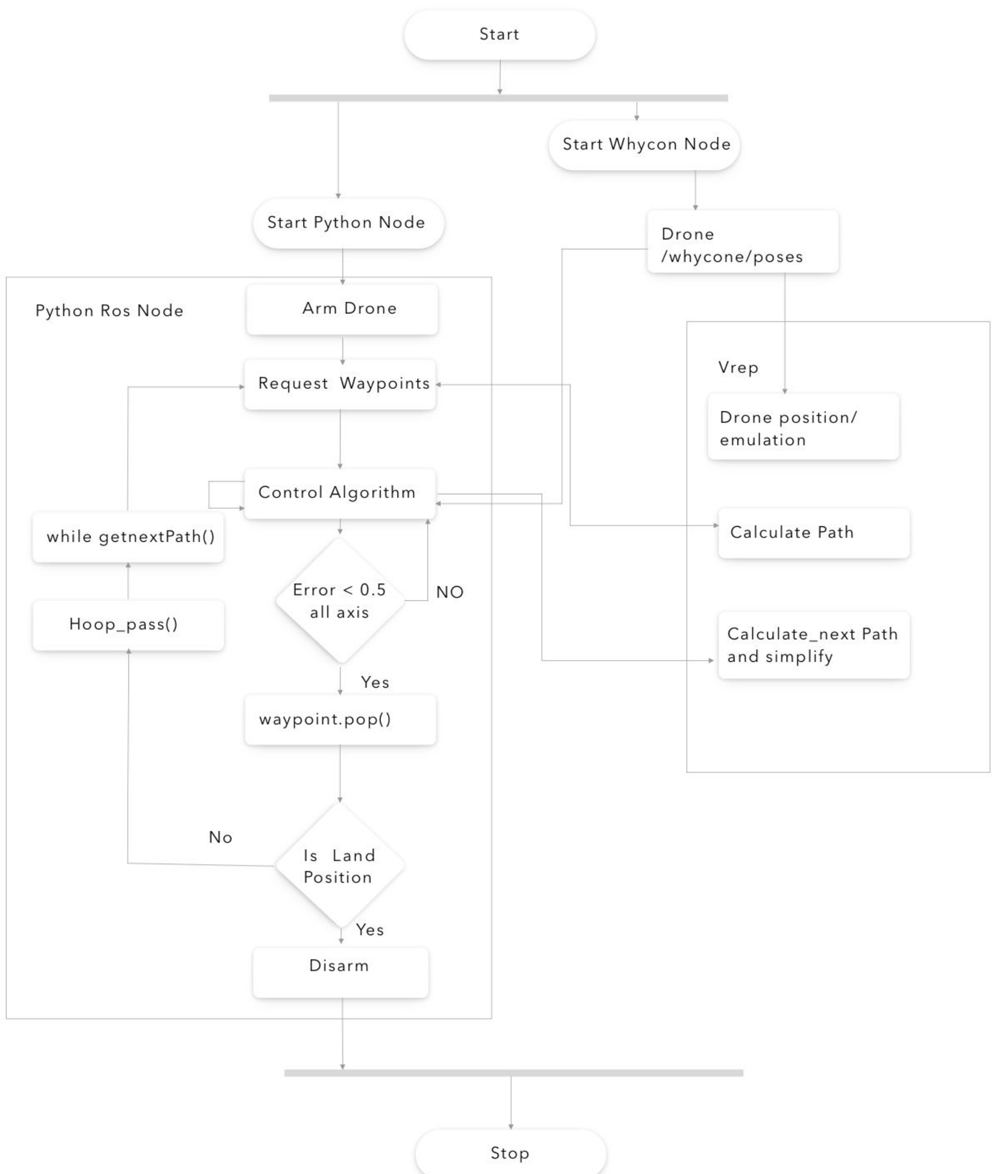


Fig 7. Flowchart for Algorithm theme implementation

Challenges

Q7. What are the major challenges that you can anticipate in addressing this theme and how do you propose to tackle them? (5)

1. Evaluation of different PID algorithms. There are 3 types of PID algorithms namely Type-A, Type-B, and Type-C.

Type-A PID: The problem with conventional PID controllers is their reaction to a step change in the input signal which produces an impulse function in the controller action. There are two sources of the violent controller reaction, the proportional term, and derivative term. Therefore, there are two other PID controllers that avoid this issue.

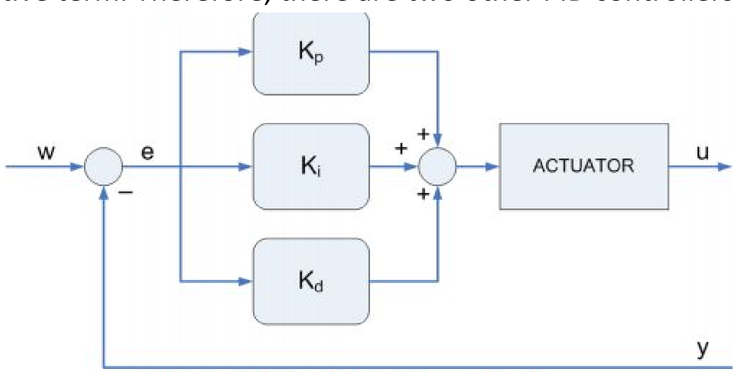


Fig. 8 Type-A PID controller

Type-B PID: It is more suitable in practical implementation to use "derivative of output controller form".

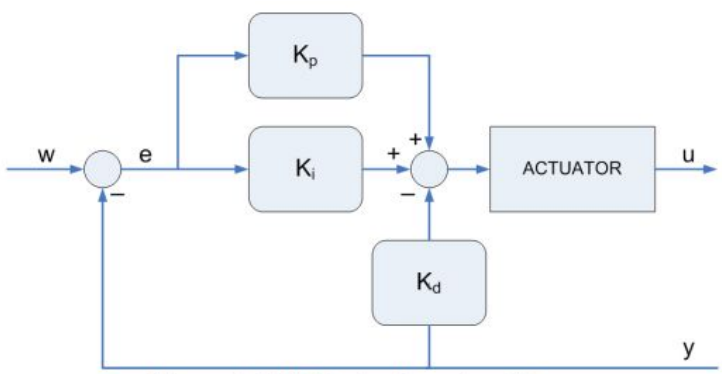


Fig. 9 Type-B PID controller

Type-C PID: With this structure transfer of reference value discontinuities to control signal is completely avoided. Control signal has less sharp changes than with other structures.

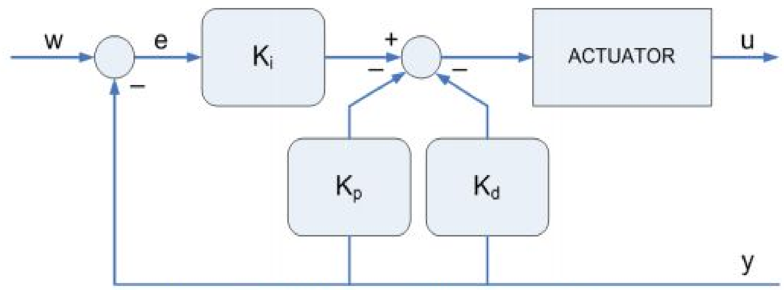


Fig. 10 Type-C PID controller

So we will have to select one that suits the task and have comparatively smaller search space for tuning the PID values. We will be balancing time-accuracy trade-off over here.

2. Tuning PID perfectly, we are trying to make an auto tuner for this.
3. Tackling with Differently Oriented hoops in the Z axis, we will need to use Transformation Matrix for taking this account to do **ellipse fitting**. As a tilted circle becomes an ellipse. So our drone will have to traverse using the much more cautious algorithm.
4. The inertia of drone while flight has to be maintained and taken account of in order to design accurate and precise algorithm. This will help us in avoiding the collision and help drone in stick to the math planned for its traversal throughout the task.
5. Since the camera is not of very high quality and wide aperture it may result in distorted marker detection, we will have to take account of this and find a mapping matrix that can map distorted pixels to undistorted pixels.

$$(X_{Corrected}, Y_{Corrected}) = \text{mapping}(X, Y)$$