Goal 1:

For reaching goal1 there are two files:

1 - turtle_controller_pid.py (which controls the pid and helps to reach the goal)

2 - turtle_spawner.py (which makes the turtle spawn at random locations from where it should reach the goal)

.....

Goal 2: turtle_grid_goal2.py

Used the pid gains from the goal 1 and define the grid points to the robot.

Yes it is required to change the gains every time, but with further hit and trial I came up to some value which works everywhere.

.....

Goal 3: turtle_rotate_goal3.py

Passed the linear.x and angular.z with little bit mathematics for converting the speed and radius, in terms of linear and angular values.

.....

Goal 4: turtle_rotate_chase_goal4.py

By using everything used in above goals first spawned a police_turtle at random location, then the turtle1 which is the robber_turtle starts rotating.

Then using the data published from robber_turtle the police_turtle moves towards the robber_turtle, considering each data received as the goal point.

.....

Goal 5: turtle_rotate_chase_goal5.py

By using the goal4 algo and limiting the speed of police_turtle to half.

Now to catch the robber_turtle , I had applied the estimated pose of the robber_turtle after 5 secs using the mathematics calculation

```
initial_angle_rad = math.atan2(self.dist_y, self.dist_x) #this line finds the current angle of robber_turtle
initial_angle_deg = math.degrees(initial_angle_rad)
new_angle_deg = initial_angle_deg + (12 * self.speed * 5)
# using the equation of the circle estimating the position of the turtle based on its speed,
as the ratio when turtle moving with linear 1.0 it takes 30 sec for 1 rotation at radius 5,
so using this by multiplying with ratio*speed*sec  we get the estimated pose after 5 sec.
Now the slow police_turtle will go for this position to catch the turtle
```

.....

Goal 6: turtle_rotate_chase_goal6.py

By using goal5 algo and adding the gaussian noise to the position and angular data of the TurtlePose.

Now to find out the estimated turtle_pose we can use the estimation technique and to tackle the problem of gaussian error,

I used considering mean values that is coming at each 5sec pose read from the topic, its like putting all the points into a graph,

and applying linear regression over it to the the central line and then using the points of that line estimating the position after 5 secs.

.....

1. If submission time was not a constraint as I had to complete this whole assignment in 1 day,
then we can perform more hit and trial over the pid values to find out the best pid,
by plotting their plots and finding relations.
2. We can develop a better estimator by researching as in this case I find this estimator good enough.
3. To tackle the problem of noisy data we can apply filters over the topics in real hardware applying filters like kalman filter,
on the data coming from sensor helps to reduce the noise.