1. My camera was sending 10 frames per second while my lidar was about 1 update per second. I updated each component of the twist as its sensor sent a new value

2. I used a Proportional controller because it is the simplest one and it worked. If the proportional controller had not worked, I would have moved on to a PI controller, but since the P controller worked, I stopped there. My controller is susceptible to disturbances not taken into account while modeling the system.

3. There is no systematic way of taking care of disturbances, but in order to make sure that the distance reported by the lidar is correct, I averaged 10 rays (5 on each side of the angle reported by the pixel from the camera)

4. If I make my gains too high, the robot oscillates. It overshoot the target, then tries to come back, but it goes much farther when trying to do so. Thus, it never stops into one place. Moreover, it foes too fast. This happens because it is a proportional controller

5. The robot start spinning on itself or it start going straight without stopping no matter what the lidar and the camera sense. It accelerated without stopping, and I had to lift it off the ground.

6. The field of view of the camera is 62.2 degrees, so when the object is centered, its center of mass is located at 31.1 degrees. However, that 31.1 degree corresponds to the 0 degree of the lidar, and the lidar's degree go counter-clockwise while the camera's degree go clock-wise. Thus correspondence had to be found.
Each pixel correspond to 62.2/360 because I used the 360 by 240 frame. I got the x pixel position from the camera, and translated it to degrees. The I took distances reported by 5 rays on each side of that angle, and I averaged them. Since there is an offset between the two frames and their angles go in opposite direction, the correspondence where established as:

if the angle of interest is greater than 31 degrees, then the corresponding lidar angle is found as: lidar_angle = 360-(camera_angle-31). The angles for which the distances were averaged were found as: lidar_angle, (lidar_angle – 1), (lidar_angle – 2), (lidar_angle – 3), (lidar_angle – 4), (lidar_angle – 5) and ((lidar_angle+1)%360), ((lidar_angle+2)%360), ((lidar_angle+3)%360), ((lidar_angle+4)%360), ((lidar_angle+5)%360).

If the angle of interest is less than 31 degrees, then the corresponding lidar angle is found as: lidar_angle = 31-camera_angle. The angles for which the distances were averaged were found as: lidar_angle, (lidar_angle+1), (lidar_angle+2), (lidar_angle+3), (lidar_angle+4), (lidar_angle+5), (lidar_angle-1), (lidar_angle-2), (lidar_angle-3), (lidar_angle-4), (lidar_angle-5).
Given that lidar_angle could be less than 5, which would lead to negative angles when computing lidar_angle-5, we checked if the angles were less than 0, and in those cases the negative angle is added to 360 in order to get the corresponding angle between 0, and 360 degrees.