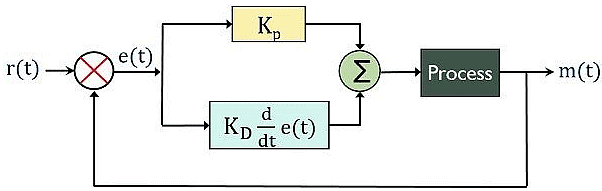
1. **What is the sampling time of your system (hint: how fast can you get sensor data)? Why does sampling time matter, even with a subscriber based controller?**The sampling rate of the LIDAR on the turtlebot3 is 1.8kHz, so its sampling time is 0.55ms. The camera information states that the implementation has a max of 90 fps. So the sampling time for this is 11.1ms. Since both sensors are being used for the lab, the limiting factor is the camera, so the system sampling rate is 11.1ms. This matters even with a subscriber-based controller because the robot is only able to detect changes the centroid of the ball every 11.1ms. If this sampling time is too low, the control loop may react too slowly to changes.
2. **What variant of PID control did you use? Why?**A PD controller was chosen. The proportional component would directly adjust the present error, and the damping gain was used to prevent overshooting the desired state, especially for rotation. Adding the integral gain was tested to reduce the steady state error, however it repeatedly led to an unstable system.
3. **If you use an integral term, how do you deal with windup? If you use a derivative term how do you deal with noise/fast changes in the object’s location? If you just used a purely proportional control, how does your reject output disturbances (like friction)?**Because we’re using the derivative term, we deal with noise/fast changes in the object’s location by averaging the detected LIDAR distances around the angular position.
4. **What does it mean for this system to be unstable? A helicopter/plane will fall out of the sky if it uses an unstable controller, what does your robot do when your controller is unstable?**When the system is unstable, it means that it’s unable to converge to the desired state. This can manifest as the controller keeps overcorrecting for the error, leading to repeated overshoot that increases the amplitude of the oscillations around the desired position in the robot.
5. **Describe your algorithm to determine where the object is relative to the robot. Specifically, how do you use the camera and LIDAR data to produce a desired velocity vector? Include mathematical expressions used and supportive figures where appropriate.**To determine where the object is angularly relative to the robot, we convert the pixel location of the centroid to an angular position relative to the center of the camera screen. The frame is 320px across with a field of view of 62.2º so the following equation was used:

To determine the distance the object is relative to the robot, the range at the resulting angular position are taken from the LIDAR data, where ‘index’ is the location within the vector, ranges, produced by laserScan.ranges:

To gain some resistance to noise, the distance values around the exact angular position were averaged:

To produce the desired velocity vector, the z-component of the angular velocity is the angular position produced from the PD controller and the x-component of the linear velocity is the calculated distance from the LIDAR produced by the second PD controller.

PD controller implemented:



For angular velocity PD controller gains:

Kp=0.24

Kd=0.005

For linear velocity PD controller gains:

Kp=2

Kd=0.5