Intro to Robotics Research Lab - 3
Team 9 (Kurma)
Prajwal Bhaskar Bharadwaj
Abhinandan Krishnan

Q1: What is the sampling time of your system(hint: how fast can you get sensor data)? How does this matter with a subscriber based controller?
The camera sends out data at 10Hz while LiDAR does it at 5Hz frequency. Subscribing to the data from each of these components, we effectively end up with a 5Hz sampling frequency (0.2 s sampling time) since we are combining the camera and LiDAR data to check for the location and angle of the object with respect to the robot. We aren't adding any delays in our code to reduce the frequency further.

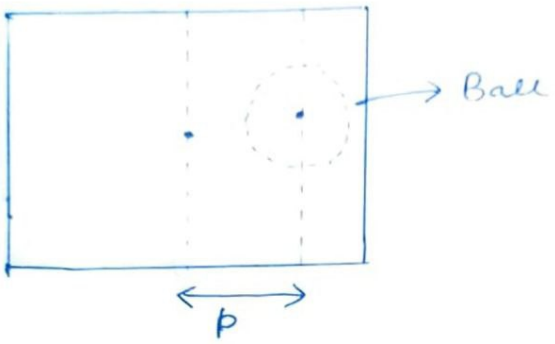
2. What variant of PID control did you use? Why?
We used a simple P controller. It's easy to debug a P controller in case the robot is not behaving in the desired manner. We were also able to find suitable Kp values for both angular and linear velocities which reached the desired state within the required settling time. So although we tried with I and D, we finally ended up using just P as it was doing the job well. In addition, the low-level motor controller takes care of any internal disturbances with it's own PID control. So it made sense to use a mid-level P control.

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 do you deal with disturbances?
We spent enough time understanding the robot system and were able to iterate through Kp values such that the overshoot was minimal. As mentioned before, the low-level motor controller takes care of any disturbances with it's own PID control. However, a major lesson we learned was to ensure that the linear control wasn't more aggressive than the angular control. Otherwise, the bot would have lost sight of the object even before it oriented itself to the object.

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?
In simple terms, a system is unstable (asymptotically unstable) if the error (reference state - current state) doesn't decrease with time. In terms of our robot, we can define stability as maintaining a particular distance and angle from a reference object. If our Kp gain is high, then the bot keeps oscillating around the object but never reaches the desired state - that would be unstable, then we will need to reduce the Kp value. In case we use a Ki, there are chances that the physical/ motor limitation of the robot makes it accumulate the error and continue to increase as this accumulated error is unwound. It leads to overshooting and instability. In that case, we need to limit our output value to the motor's limitation. Kd is sensitive to noise and hence, a large value of Kd will lead to a disproportionate input to the system. PID controllers often go with an N - filter which is supposed to cancel out the negative effects of the Kd term. Making linear control more aggressive than angular control can also lead to instability as the robot might lose sight of the goal before it even orients itself to the goal as the linear velocity control output is too high.

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 the desired velocity vector? Include mathematical expressions used and supportive figures where appropriate.

We used a mask in the image processing code of the camera data. It detects objects of 'Yellow' color and gives the centroid location (using cv2.moments) of the object contour. We use this to estimate the angle difference between the center of the robot camera frame and the center of the ball. We then average the ±5° instantaneous LIDAR data for the corresponding angle at which the ball center is located.
(In the figure, 320 corresponds to the horizontal resolution of the image)



p → pixel difference b/w the centre of the frame & ball centre.

$\theta_t = p \times \frac{62.2}{320}$, field of view of Pi camera = 62.2°

The ranges array holds the distance from 0° to 360° with 1° step.

$\therefore \quad d_t = \frac{\sum ranges_t [\theta_t \pm i]}{10}$, $\forall\ i = 1,2,3,4,5.$