Lab 03 Questions

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?

For the lidar data we obtain the data at an average rate of 9.5 Hz with standard deviation of 0.077. And for the camera the frame is refreshed at a rate of 9.9 Hz with standard deviation of 0.059.

For subscriber-based controllers if the sampling frequency is lower than than the controller output then there will be data loss which will result in erratic behaviour.

What variant of PID control did you use? Why?

For this lab, we used a proportional controller because this controller generates a smooth response proportional to the error and the output disturbances are rejected by the PID controller in the motor.

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 reject output disturbances?

The integral windup problem can be mitigated by halting the accumulation of integral error when the control output reaches 100% of system capacity. If we use a derivative term for the controller, we need to utilize filters like the Kalman filter in order to deal with noise in the input data. When we just use a proportional control, the output disturbances are rejected by the PID controller in the motor.

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?

If the angle controller is unstable, the robot keeps spinning till the battery lasts. If the linear controller is unstable, the robot will oscillate at one position without coming to a stop.

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.

The algorithm has following parts:

Obtain the Orientation of the object using the Camera

We get the centroid of the object using Object Detection. From this we obtain the angle using following formula:

$$(X - XCent / Width) * FOV * (pi / 180) = angle\ in\ radians$$

Obtain the Range of the object from the Lidar

From the Lidar msg we obtain the $angle_{min}$, $angle_{max}$, $angle_{increment}$ and the $ranges$. Using the following formula we obtain the index:

$$index = round(goal_{ang} - angle_{min} / angle_{inc})$$

The threshold to obtain is 5 so we take index - 5 and index + 5 to get the range and then average it to the distance of the object.

Controller Algorithm

We get the current_angle and current_dist, the linear_setpoint is 15 cm, the angular setpoint is 0 rad and the tolerance is 0.001. Based on this we calculate the errors and use a proportional controller to get the current twist.

The controller first tries to align the angle and then the linear distance. If the object is removed the default behaviour is to rotate the robot with velocity of 0.05 rad/s.