CMPE 565 AUTONOMOUS ROBOTS
Assignment 6: State Estimation

Deadline: 22 March 2018

In this assignment, you will implement a Kalman Filter to estimate the position of a moving object.

Update your course repository using `git pull`.

For this assignment, you will use scene assignment6 and package state_estimation. Note that you can run state_estimation packages as follows:
`rosrun state_estimation state_estimation` Also, to be able to compile the state estimation package, you need to install `ros-kinetic-tf-conversions` package.

In the simulation, the ball moves at a constant speed to a constant direction. You may modify the movement of the object as you like, however the object should disappear behind an obstacle for some time. Note that, we actually add certain x and y value to the position of the object, and additional noise at every time step. We also add noise to depth sensor to make sure that your sensor provides noisy estimation about the position of the object.

For this assignment, you will implement an object detection algorithm using point cloud data and kalman filter for tracking the position of the object.

1 Object Detection

You are going to detect a ball using sensor_msgs::PointCloud2 data. You are going to implement detection code in pointCloudCallback method. This method reads point cloud, changes the frame of the point cloud to robot frame and changes the type of the point cloud to PCL point cloud.

You can develop your own code for plane removal since scene has planes and the ball. When you remove the planes, you can detect the position of the ball.

As an alternative, you can use pcl library to remove planes from the point cloud. You can check these two tutorials for plane extraction:

- `http://pointclouds.org/documentation/tutorials/extract_indices.php`
- `http://pointclouds.org/documentation/tutorials/planar_segmentation.php`

2 Kalman Filter

You are going to implement a linear Kalman filter. You can use state vector as x position, y position, x speed, and y speed (as shown in the class). Note that at every time step position of the ball is increased in both direction according to the constant speed with an added noise. To estimate the error covariance matrices you can use actual ball position read in ballPoseCallback method, and point cloud data by analysing data coming from the floor or the wall.

To be able to implement Kalman Filter:

- Define the state, control and observation matrices.
- Define state transition and observation matrices.
- Define the initial belief state, state transition probability and observation probability. (e.g. define noise parameters like covariance matrices)

3 Deliverables

- Put your object detection code
- State your Kalman Filter definitions/matrices explicitly in your report.
- Put your kalman filter implementation code
- Plot the estimated, perceived, and real position values starting from the beginning of the scene until the ball leaves the scene.