

Advanced Kalman Filtering and Sensor Fusion

2D Vehicle Extended Kalman Filter: Update Step

EKF Exercise 2





Overview

Implement the Kalman Filter Update Equations and the Lidar Measurement Model.

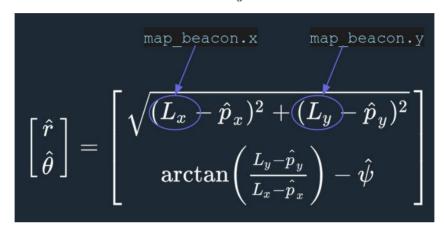
Step 1 (Setup)

- Open your last kalman filter file from the previous exercise which had the prediction step completed.
- Review the code that is already been written for the Update Steps as part of the EKF exercises.
 - handleLidarMeasurements() is called after the prediction step whenever LIDAR measurements have been made by the sensor. This function then sequentially calls the handleLidarMeasurement() function which is the main function you will modify to fuse the Lidar measurements one at a time.



Step 2 (Implement the Lidar Measurement Model)

Modify the function handleLidarMeasurement()



NOTE: The code assumes that the state vector has the following form!!
$$\hat{x} = egin{bmatrix} p_x \\ p_y \\ \psi \\ V \end{bmatrix}$$

Step 3 (Implement the Innovation Calculations)

- Modify the function handleLidarMeasurement()
- Calculate the Measurement Innovation (make sure you normalise the angle innovation!)
- Calculate the Measurement Jacobian Matrix (H matrix)
- Calculate the Measurement Innovation Covariance (S matrix)
- Assume the range and theta measurement std are LIDAR_RANGE_STD and LIDAR_THETA_STD



Step 3 (Implement the Innovation Calculations)

- Modify the function handleLidarMeasurement()
- Calculate the Measurement Innovation (make sure you normalise the angle innovation!)

y(1) = wrapAngle(y(1)); // Wrap the Heading Innovation

- Calculate the Measurement Jacobian Matrix (H matrix)
- Calculate the Measurement Innovation Covariance (S matrix)
- Assume the range and theta measurement std are LIDAR_RANGE_STD and LIDAR_THETA_STD.

$$u = egin{bmatrix} r \ heta \end{bmatrix}_{ ext{meas}} - egin{bmatrix} \hat{r} \ \hat{ heta} \end{bmatrix} \ \mathbf{R} = egin{bmatrix} \sigma_r^2 & 0 \ 0 & \sigma_ heta^2 \end{bmatrix}$$

$$\mathbf{H} =
abla h_x = egin{bmatrix} rac{1}{d}(\hat{p}_x - L_x) & rac{1}{d}(\hat{p}_y - L_y) & 0 & 0 \ rac{-1}{d^2}(\hat{p}_y - L_y) & rac{1}{d^2}(\hat{p}_x - L_x) & -1 & 0 \end{bmatrix} \ d = \sqrt{(L_x - \hat{p}_x)^2 + (L_y - \hat{p}_y)^2}$$

$$\mathbf{S} =
abla h_x \mathbf{P}_k^-
abla h_x^T + \mathbf{R}_k^T$$



Step 4 (Implement the Kalman Filter Update Step Equations)

$$egin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + \mathbf{K}_k
u_k \ \mathbf{K}_k &= \mathbf{P}_k^-
abla h_x^T \mathbf{S}_k^{-1} \ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k
abla h_x) \, \mathbf{P}_k^- \end{aligned}$$



Step 5 (Run the Simulation in the following configurations)

- Profile 1 (Constant Speed/Heading, Zero Initial Conditions)
- Profile 2 (No-Zero Initial Conditions)
- Profile 3 (Constant Speed, Changing Headings)
- Profile 4 (Changing Speed, Changing Headings)
- Profile 5 (Profile 1 + LIDAR)
- Profile 6 (Profile 2 + LIDAR)
- Profile 7 (Profile 3 + LIDAR)
- Profile 8 (Profile 4 + LIDAR)

Step 6 (Compare the Simulation Results between the LKF and EKF with/without Lidar)

