Technitute_

Advanced Kalman Filtering and Sensor Fusion

Linear Vehicle Tracker: Update Step

LKF Exercise 2

Linear Vehicle Tracker: Update Step Exercise

Overview

Implement the ***Kalman Filter Update Equations*** and the ***GPS Measurement Model***.

Step 1 (Setup)

- Open your last kalman filter file from the previous exercise which had the prediction step completed.

Step 2 (Setup the H Matrix and R Matrix)

- Modify the function ***handleGPSMeasurement()***
- Assume the position measurement std is GPS_POS_STD

```
void KalmanFilter::handleGPSMeasurement(GPSMeasurement meas)
{
    if(isInitialised())
    {
        VectorXd state = getState();
        MatrixXd cov = getCovariance();

        // Implement The Kalman Filter Update Step for the GPS Measurements in the
        // section below.
        // Hint: Assume that the GPS sensor has a 3m (1 sigma) position uncertainty.
        // Hint: You can use the constants: GPS_POS_STD
        // ----------------------------------------------------------------------- //
        // ENTER YOUR CODE HERE


        // ----------------------------------------------------------------------- //

        setState(state);
        setCovariance(cov);
    }
    else
```

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} \sigma^2_{meas} & 0 \\ 0 & \sigma^2_{meas} \end{bmatrix}$$

Linear Vehicle Tracker: Update Step Exercise

Step 3 (Implement the Kalman Filter Update Step Equations)

- Modify the function ***handleGPSMeasurement()***

```
void KalmanFilter::handleGPSMeasurement(GPSMeasurement meas)
{
    if(isInitialised())
    {
        VectorXd state = getState();
        MatrixXd cov = getCovariance();

        // Implement The Kalman Filter Update Step for the GPS Measurements in the
        // section below.
        // Hint: Assume that the GPS sensor has a 3m (1 sigma) position uncertainty.
        // Hint: You can use the constants: GPS_POS_STD
        // ----------------------------------------------------------------------- //
        // ENTER YOUR CODE HERE


        // ----------------------------------------------------------------------- //

        setState(state);
        setCovariance(cov);
    }
    else
```

$$\tilde{y}_k = z_k - \mathbf{H}_k \hat{x}_k^-$$
$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$$
$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k \tilde{y}_k$$
$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\, \mathbf{P}_k^-$$

Linear Vehicle Tracker: Update Step Exercise

Step 4 (Run the Simulation in the following configurations)

- Set initial state and covariance to zero along with the process model noise. See that the Kalman Filter Estimate does not change or is updated.
- Change the initial velocity std (INIT_VEL_STD) to 10 and re-run the simulation.
- Test out different values of INIT_POS_STD, INIT_VEL_STD and ACCEL_STD.
- What happens if you run with:
  - Profile 2 (No-Zero Initial Conditions)
  - Profile 3 (Constant Speed, Changing Headings)
  - Profile 4 (Changing Speed, Changing Headings)

X Position RMSE: 0.88 m
Y Position RMSE: 0.98 m
Heading RMSE: 5.31 deg
Velocity RMSE: 0.50 m/s