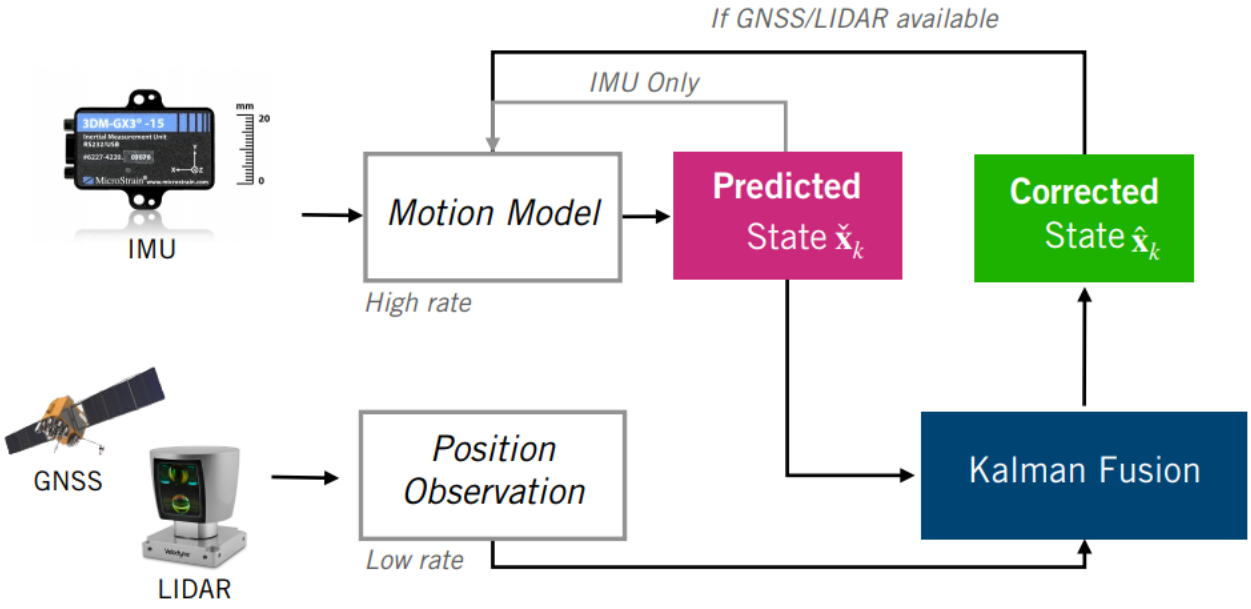
State Estimation and Localization for Self Driving Cars: Vehicle State Estimation on a Roadway (Project Report)

The final project in the second course in the Self Driving Car specialization offered by Coursera.

Objective: This project aims to implement an Error-State Extended Kalman Filter (ES-EKF) for vehicle localization using the data from the CARLA simulator.

Introduction: State estimation using the Kalman filter is a powerful technique used in various fields, such as control systems, robotics, and signal processing. It provides a mathematical framework to estimate the true state of a system by fusing noisy measurements with system dynamics. By utilizing a recursive algorithm, the Kalman filter optimally combines the current state estimate with new measurements to generate an improved estimate in a real-time fashion. It effectively handles uncertainties and can handle both linear and nonlinear systems. With its wide applicability, the Kalman filter has become a cornerstone of modern estimation and control theory.



The Kalman Filter algorithm updates a state estimate through two stages:

1. Prediction using the motion model
2. Correction using the measurement model

Methodology

1. Vehicle State Initialization

The vehicle state comprises position, velocity, and orientation, which is parameterized by a unit quaternion. The motion model relies on inputs derived from the measurements of specific force (acceleration) and angular rate obtained from the Inertial Measurement Unit (IMU). By incorporating these IMU measurements into the Kalman filter, it becomes possible to accurately estimate the dynamic state of the vehicle. This estimation process takes into account the noise and uncertainties inherent in the IMU measurements, resulting in reliable and precise estimates of the vehicle's motion parameters.

Vehicle State: $\mathbf{x}_k = [\mathbf{p}_k, \mathbf{v}_k, \mathbf{q}_k]^T \in R^{10}$

IMU Measurements: $\mathbf{u}_k = [\mathbf{f}_k, \boldsymbol{\omega}_k]^T \in R^6$

2. Prediction Motion Model Vehicle's predicted states using the motion model would be as follows:

Predicted State: $\check{\mathbf{x}}_k = [\check{\mathbf{p}}_k, \check{\mathbf{v}}_k, \check{\mathbf{q}}_k]^T$ |

Predicted Position: $\check{\mathbf{p}}_k = \mathbf{p}_{k-1} + \Delta t \mathbf{v}_{k-1} + \frac{\Delta t^2}{2}(\mathbf{C}_{ns}\mathbf{f}_{k-1} + \mathbf{g})$ |

Predicted Velocity: $\check{\mathbf{v}}_k = \mathbf{v}_{k-1} + \Delta t(\mathbf{C}_{ns}\mathbf{f}_{k-1} + \mathbf{g})$ |

Predicted Orientation: $\check{\mathbf{q}}_k = \mathbf{q}_{k-1} \otimes \mathbf{q}(\boldsymbol{\omega}_{k-1}\Delta t)$ |

Error State Linearization

In the error-state Kalman filter,th rather than applying the filter to the while nominal stae, the filter will be applied only to the error state.

The error states will be first estimated and then it will be used to correct the nominal state.

The linearized error dynamics is used now to get the error states.

Error State: $\delta\mathbf{x}_k = [\delta\mathbf{p}_k, \delta\mathbf{v}_k, \delta\boldsymbol{\phi}_k]^T \in R^9$

Error Dynamics: $\delta\mathbf{x}_k = \mathbf{F}_{k-1}\delta\mathbf{x}_{k-1} + \mathbf{L}_{k-1}\mathbf{n}_{k-1}$

Measurement Noise: $\mathbf{n}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$

Where,

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \cdot \Delta t & 0 \\ 0 & \mathbf{I} & -[\mathbf{C}_{ns}\mathbf{f}_{k-1}]_{\times}\Delta t \\ 0 & 0 & \mathbf{I} \end{bmatrix}$$ is the motion model jacobian,

$$\mathbf{L}_{k-1} = \begin{bmatrix} 0 & 0 \\ \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix}$$ is the Motion Model Noise Jacobian, and

$$\mathbf{Q}_k = \Delta t^2 \begin{bmatrix} \mathbf{I} \cdot \sigma_{acc}^2 & 0 \\ 0 & \mathbf{I} \cdot \sigma_{gyro}^2 \end{bmatrix}$$ is the IMU Noise Covariance

Uncertainity Propagation

The Jacobian Matrices computed will be used to propagate the uncertainities forward in time. Uncertainity is captured by state covariance matrix. Untill we obtain a measurement, the uncertainity grows.

Predicted State Covariance: $\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^T$

3.Correction Stage

Measurement Model

Measurement Model:

$$\begin{aligned} \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \\ &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \\ &= \mathbf{p}_k + \mathbf{v}_k \end{aligned}$$

Measurement Noise (GNSS): $\mathbf{v}_k \sim N(0, \mathbf{R}_{\text{GNSS}})$

LIDAR Measurement Noise: $\mathbf{v}_k \sim N(0, \mathbf{R}_{\text{LIDAR}})$

Measurement Updte

Extended Kalman filter is used to process the measurements. The measurements in this case come from GNSS and LIDAR. They both provide position updates.

Measurement Model Jacobian: $\mathbf{H}_k = \begin{bmatrix} \mathbf{I} & 0 & 0 \end{bmatrix}$

Sensor Noise Covariance: $\mathbf{R} = \mathbf{I} \cdot \sigma_{\text{sensor}}^2$

Kalman Gain: $\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R})^{-1}$

The Kalman Gain is used to compute the error state. The error state consideres the difference between the predicted vehicle position and measured position ($\delta \mathbf{x}_k = \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{p}}_k)$).

The error state is used then to update the nominal state vector. Also, the corrected nominal state covariance vector is calculated.

Corrected Position: $\hat{\mathbf{p}}_k = \check{\mathbf{p}}_k + \delta \mathbf{p}_k$

Corrected Velocity: $\hat{\mathbf{v}}_k = \check{\mathbf{v}}_k + \delta \mathbf{v}_k$

Corrected Orientation: $\hat{\mathbf{q}}_k = \mathbf{q}(\delta \boldsymbol{\phi}_k) \otimes \check{\mathbf{q}}_k$

The corrected states and state covariance values are obtained in this way.

Corrected State Covariance

Vehicle Trajectory Comparison The ground truth vehicle trajectory was compared with the trajectory estimated by the algorithm.

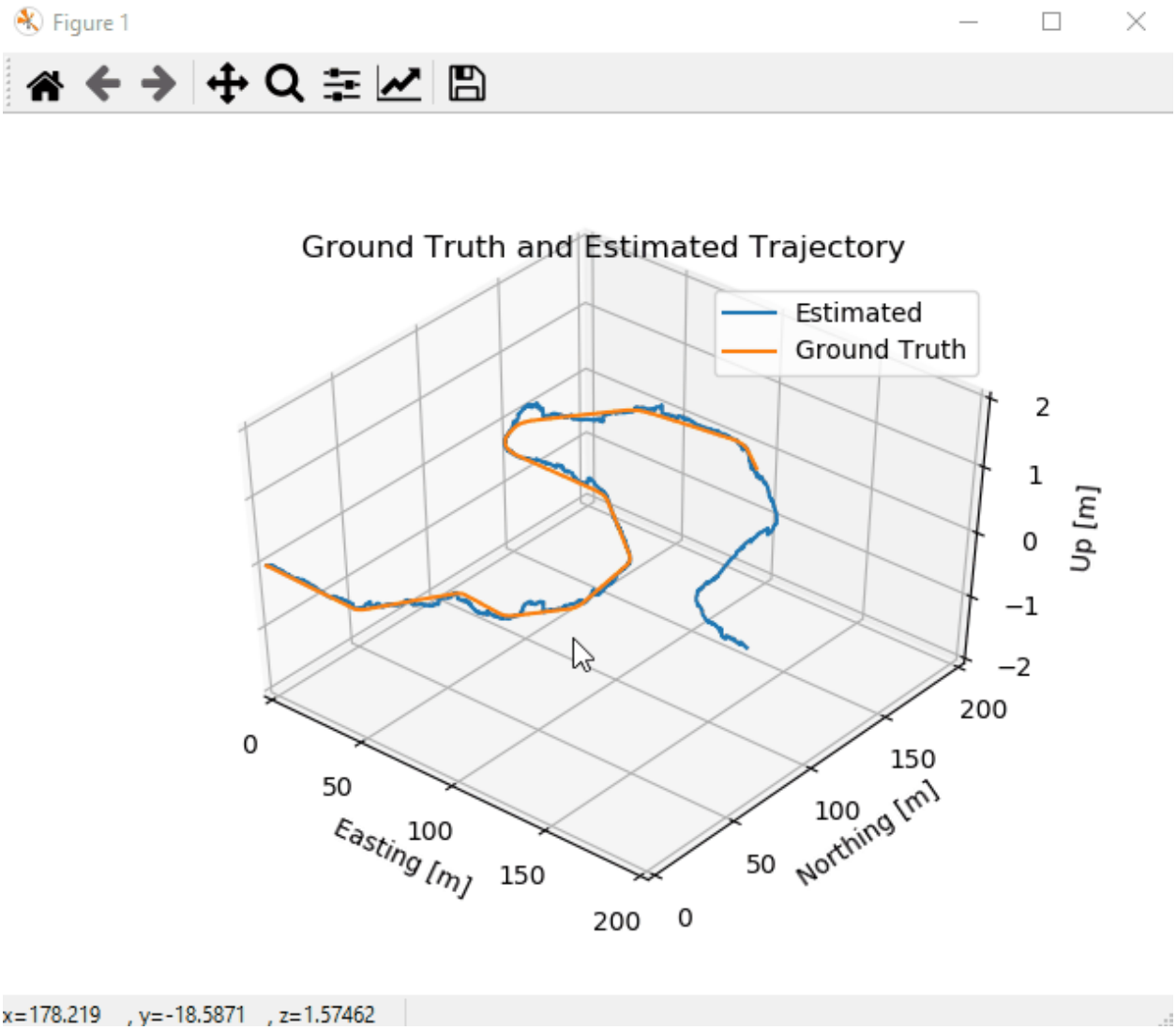


Figure 1
Ground Truth and Estimated Trajectory
Estimated
Ground Truth
Easting [m]
Northing [m]
Up [m]
0
50
100
150
200
2
1
0
−1
−2
x=178.219 , y=-18.5871 , z=1.57462