Project 2

Topics:

- Kalman filter
- Extended Kalman filter
- EKF-SLAM

1. Kalman Filter

a) Download an INS sequence (OXT) recorded from the KITTI dataset:
Please use the following table to download your example, based on your ID:

Last ID	Recorded data
0-1	2011_09_26_drive_0061
2-3	2011_10_03_drive_0042
4-5	2011_09_30_drive_0020
6-7	2011_09_26_drive_0087
8-9	2011_09_29_drive_0071
Example	2011_09_30_drive_0033

b) Extract the vehicle GPS trajectory from KITTI OXTS senser packets.

These are treated as ground-truth trajectory in this experiment.

- lat: latitude of the oxts-unit (deg)
- long: longitude of the oxts-unit (deg)
- Timestamps- Extract timestamps from KITTI data and convert them to seconds elapsed fromthe first one.

c) Transform GPS trajectory from global coordinates (LLA) to local coordinates (ENU→ x/y/z) in order to enable the Kalman filter to handle them. Plot ground-truth GPS trajectory (2.5%):

- World coordinate (LLA)
- Local coordinate

d) Add a gaussian noise to the ground-truth GPS data which will be used as noisy observations fed to the Kalman filter later. (2.5%):

- standard deviation of observation noise of x and y in meter (σ_x =3 , σ_y =3)
- Plot the trajectory in local coordinates without noise (ENU) and observations noise (ENU+noise) on the same graph.

Apply Kalman filter!!

Apply a linear Kalman filter to the GPS sequence in order to estimate vehicle 2D pose based on constant velocity model:

Suppose initial 2D position [x, y] estimation starts with the first GPS observation. GPS observation noise of X and Y is known (σ_x =3, σ_y =3)

Be aware that the ground-truth data is used only for the evaluation.i.e., only noised GPS sequence is given to the filter.

e) The goal: (15%):

Calibration of Kalman filter based on the following RMSE and maxE criteria:

$$RMSE \triangleq \sqrt{\frac{1}{N}\sum_{i=100}^{N}[e_x^2(i) + e_y^2(i)]}$$

$$e_x(i) \triangleq x_{GT}(i) - x_{Estimate}(i)$$

$$e_y(i) \triangleq y_{GT}(i) - y_{Estimate}(i)$$

$$maxE \triangleq max\{|e_x(i)| + |e_y(i)|\} \quad ,$$

$$100 \leq i \leq N$$

N is last sample.

Note- the RMSE calculation does not start for the first frame due to the convergence time of the model

Conduct an empirical test to determine which calibration of Kalman filter (i.e, initial condition and noise covariance matrix R) achieves a maxE criterion of less than ~7m.

i. Initial conditions: Which values of standard deviations should be initialized. according to your first observation? Explain.
ii. Matrixes: A ,B ,C
iii. Measurement covariance (Q)
iv. Noise covariance R- which values of standard deviation σ_n should be set in this case? Show your analysis.

Implement and describe the Kalman filter main routine.

Show and explain your results.

f) Result analysis: (15%)

i. Ground-truth and estimated results

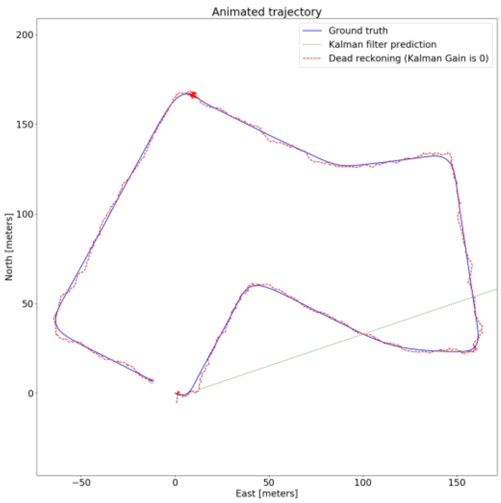
ii. What are the minimum values of maxE and RMSE achieved? Explain.

iii. Show the Kalman filter's performance.

iv. Plot the final trajectory and show it in an animation:

- Trajectory of GT and KF results.
- After ~5 seconds try estimate the trajectory based on the prediction without observation (dead reckoning, Kalman gain=0).
- Optional- plot covariance matrix of state vector as an ellipse (ellipse printing function is part of the package).

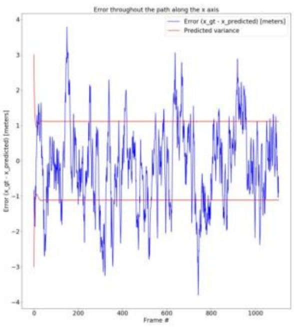
Example:

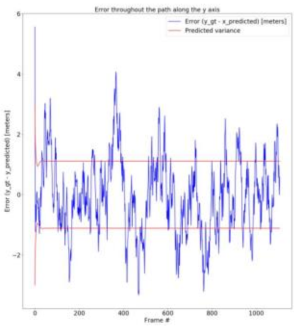


v. Plot and analyze the estimated error of x-y values separately and corresponded sigma value along the trajectory

- Plot GT-estimated values.
- Plot on the graph sigma value (+/- sigma)

Example:





(bonus! 5%). Implement <u>constant-acceleration model</u> and compare theresults with constant-velocity model

2. **Extended Kalman Filter**

a. Use the same KITTI GPS/IMU sequence from last section.

b. Extract vehicle GPS trajectory, yaw angle, yaw rate, and forward velocity from KITTI senser packets (OXT).

These are treated as ground-truth trajectory in this experiment:

- lat: latitude of the oxts-unit (deg)
- lon: longitude of the oxts-unit (deg)
- yaw: heading (rad)
- vf: forward velocity, i.e. parallel to earth-surface (m/s)
- wz: angular rate around z (rad/s)
- Extract timestamps from KITTI data and convert them to seconds elapsed. from the first one.

c. Transform GPS trajectory from [lat, long, alt] to local [x, y, z] cord in order to enable the Kalman filter can handle it. Plot ground-truth GPS trajectory:

- World coordinate (LLA)
- Local coordinate (ENU)
- Plot ground-truth yaw angles, yaw rates, and forward velocities

d. Add gaussian noise to the ground-truth GPS/IMU data.
Those are used as noisy observations given to Kalman filter later.standard deviation of observation noise of x and y in meter: $\sigma_x = \sigma_y = 3$

Apply Extended Kalman filter!

e. Let's apply an Extended Kalman filter to the GPS sequence in order to estimate the vehicle’s 2D pose velocity-based model (non-linear model):

- Suppose initial 2D position [x, y] estimation begins with the first GPS observation
- GPS observation noise of X and Y is known (σ_x =3, σ_y =3)

f. Adding gaussian noise to Forward Velocity and Angular Rate: (5%)

- Add noise to angular yaw rates-
  standard deviation of yaw rate in rad/s (σ_w =0.2)
  plot graphs of GT+ noise yaw rate

- Add noise to forward velocities-
  add standard deviation of forward velocity in m/s (σ_{fv} =2)
  plot graphs of GT+ noise velocities

In this exercise, we are not introducing additional noise to the yaw measurements because the estimation is already affected by the noise added to the yaw rate and velocities. However, in reality pure noise could also be added to yaw measurements.

g. The goal: (15%): $minimize\ RMSE\ while\ maxE < \sim 5$
(similar criteria such as 2.e)

Conduct an empirical test to determine which calibration of Extended Kalman filter (i.e, initial condition and noise covariance matrix R) achieves a maxE criterion of less than ~5m.

i. Initial conditions.
ii. Jacobians G, V and C
iii. Covariance (Q and R)

Please implement and describe the main routine of the EKF.

h. Result analysis: (15%)

i. Ground-truth and estimated results
ii. Show Kalman filter performance.
iii. Based on your empirical tests- what are the minimum values of maxE and RMSE were able to you achieved? Please provide an explanation

iv. Plot final trajectory and show it also in animation (like section 1.f.iv):
- Trajectory of GT and EKF results.
- Optional- plot covariance matrix of state vector as ellipse
- After 5 seconds estimate the trajectory based on the prediction without observation (dead reckoning). compare results to GT and explain.

v. Plot and analyze the estimated error of x-y-θ values separately and corresponded sigma value along the trajectory. (Like section 1.f.v).
- Plot GT-estimated values.
- Plot on the graph sigma value (+/- sigma)

3. EKF-SLAM

a) Load attached inputs and code Python files.
   a. Landmarks location
   b. Odometry and sensor data
   c. Fill the missing parts inside the attached code. ("TODO" comment)

b) Run Odometry data according to odometry model and plot the GT trajectory.

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \delta_{trans}\cos(\theta_{t-1}+\delta_{rot1}) \\ \delta_{trans}\sin(\theta_{t-1}+\delta_{rot1}) \\ \delta_{rot1}+\delta_{rot2} \end{pmatrix}$$

c) Add gaussian noise in the motion model assume ($\sigma_{rot1} = 0.01, \sigma_{trans} = 0.1, \sigma_{rot2} = 0.01$)

<u>Apply Extended Kalman SLAM filter!</u> (15%)

The goal: $minimize\ RMSE\ while\ maxE < 1.5$

$$RMSE \triangleq \sqrt{\frac{1}{N}\sum_{i=20}^{N}[e_x^2(i)+e_y^2(i)]}$$

$$e_x(i) \triangleq x_{GT}(i) - x_{Estimate}(i)$$
$$e_y(i) \triangleq y_{GT}(i) - y_{Estimate}(i)$$

$$maxE \triangleq max\{|e_x(i)| + |e_y(i)|\} \quad ,$$

$$20 \le i \le N$$

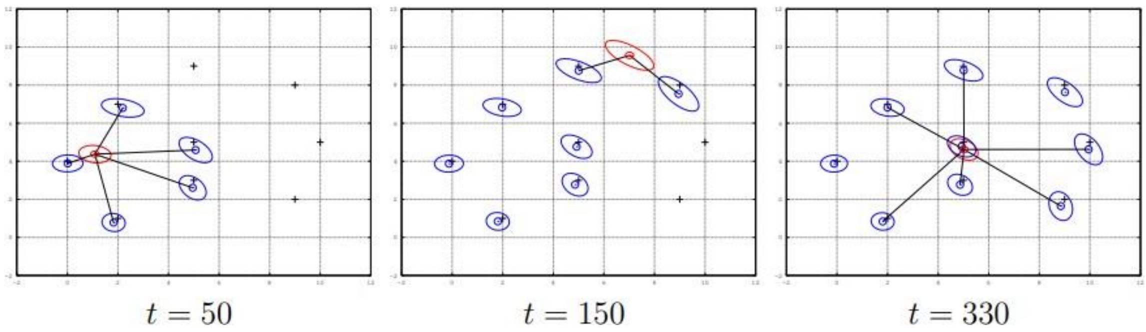
N is last sample.

Note- The RMSE calculation does not start for the first frame due to the convergence time of the model

d) Initialize initial conditions μ_0, Σ_0
e) Implement the prediction step of the EKF SLAM algorithm in the function "predict" Use the odometry motion model.

f) Compute its Jacobian G_t^x to construct the full Jacobian matrix G_t:

g) Compute its Jacobian V to construct the full Jacobian matrix R_t^x and R_t



Example images of the state of the EKF at certain time indices.

h) Implement the correction step in the function “update”.

The argument z of this function is a struct array containing m landmark observations made at time step t.

Each observation $z(i)$ has an id $z(i).id$, a range $z(i).range$, and a bearing $z(i).\,bearing$. Iterate over all measurements $(i = 1, \ldots, m)$ and compute the Jacobian H_t^i

You should compute a block Jacobian matrix H_t by stacking the H_t^i matrices corresponding to the individual measurements. Use it to compute the Kalman gain and update the system mean and covariance after the for-loop. For the noise in the sensor model assume that Q_t is a diagonal square matrix as follows
$(\sigma_r = 0.3, \sigma_\emptyset = 0.035)$.

i) Analyze results: (15%)
1. Plot on the same figure
   a. Trajectory of EKF-SLAM results.
   b. show the above results in an animation, plot covariance matrix ofstate vector as ellipse.
   c. What is the minimum values of maxE and RMSE achieved? Explain.

2. Analyze estimation error of X, Y and Theta
   a. Plot GT-estimated values.
   b. Plot on the graph sigma value (+/- sigma)

3. Select 2 landmarks and analyze.
   a. Plot GT-estimated values.
   b. Plot on the graph sigma value (+/- sigma)

4. Explain your results.

Appendix

A. Please be honest, you may automatically lose points if you are caught copying including from the internet (code, results). The work is personal.

B. See instructions about the recorded data in the Appendix.

C. You are required to read the following paper for better understanding.
Vision meets Robotics: The KITTI Dataset/ Andreas Geiger

D. The final grade is given according to the quality of your analyses, descriptions, conclusions, explanations, the form of the results (plot, graphs, animations), understandable code with comments and explanations. It is possible that the final performance and results will not be as perfect as you desired as this is real data and is part of the challenge of the autonomous driving field Feel free to suggestsolutions that could improve your results if this is the case.

E. Your final package should contain the following folders:

- **Code** contains all functions + sub-functions
- **Results** stores the resulting figures ,animations, etc.
- Please save the package as zip file. The name of the should be your ID.

The report should be separated from the package, please use the attached format and read the comments therein.