Lecture: Mapping and Perception for an autonomous robot ,0510-7951

Roy Orfaig, Prof Ben-Zion Bobrovsky. Spring 2021

Last Date to submit- 2.5

VERSION C

Project 2

Topics:

- Kalman filter
- Extended Kalman filter
- EKF-SLAM

1. Kalman Filter
   a) Load KITTI GPS sequence (OXT)
   kitti_root_dir: 2011_9_30
   KITTI drive=0033

   b) Extract 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)
   - 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] coordinates in order to enable the Kalman filter to handle them. Plot ground-truth GPS trajectory
   - World coordinate (LLA)
   - Local coordinate (ENU)

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

   i. standard deviation of observation noise of x and y in meter (σ_x =3 , σ_y =3)
   ii. plot on the same graph original GT and observations noise.

   <u>Apply Kalman filter!</u>
   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: (10%)

$$minimize\ RMSE\ while\ maxE < 2m$$

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

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

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

$$100 \le i \le N$$

N is last sample.

Find which calibration has the best performance according to the above criteria.

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. Find the best state transition noise covariance R. Which values of standard deviation σ_a 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: (10%)
i. Ground-truth and estimated results
ii. Show the Kalman filter's performance.

iii. Plot the final trajectory and show it in an animation:
a. Trajectory of GT and KF results.
b. After 5 seconds try estimate the trajectory based on the prediction without observation (dead reckoning). compare results to GT and explain.
c. Plot covariance matrix of state vector as an ellipse.
iv. Plot and analyze the estimated x-y values separately and corresponded sigma value along the trajectory. (e.g. show in same graph $x_{estimated}$-x_{GT} and σ_x values and explain your results).

g. Performance comparison (10%)
   i. Change the gaussian observation noise of the GPS data to a different noise level. As follows,
      - $\sigma_x = \sigma_y = 5$
      - $\sigma_x = \sigma_y = 10$

      Check the performance of your filter on these noise levels and compare results to f.

   ii. Implement constant-acceleration model and compare the results with constant-velocity model (bonus! 5%).

2. Extended Kalman Filter

a. Load KITTI GPS/IMU sequence
   kitti_root_dir: 2011_9_30
   KITTI drive=0033

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 [lon, lat, alt] to local [x, y, z] cord in order to enable the Kalman filter can handle it. Plot ground-truth GPS trajectory: (2.5%)
   - 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
   $\sigma_x = \sigma_y = 3$

<u>Apply Extended Kalman filter!</u>

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 ($\sigma_x = 3$, $\sigma_y = 3$)
   - Implement an EKF based on velocity-based model and compare results to the constants-velocity model (set the same initial conditions). Analyze

and explain the results (compare to constant-velocity model) (5%)

f. Add gaussian noise to the IMU data: (5%)
   - Add noise to 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

f) The goal: (10%)

$$minimize\ RMSE\ while\ maxE < 2m$$

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

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

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

$$100 \le i \le N$$

N is last sample.

Find which calibration has the best performance according to the above criteria.

i. Initial conditions: Which values of standard deviations should be initialized according to your first observation? Explain.
ii. Jacobians G, V and C
iii. Covariance (Q and R)

Implement and describe EKF main routine.
Show and explain your results:

h. Result analysis: (15%)
   i. Ground-truth and estimated results
   ii. Show Kalman filter performance.

   iii. Plot final trajectory and show it also in animation:
      a. Trajectory of GT and EKF results.
         Plot covariance matrix of state vector as ellipse
      b. After 5 seconds try estimate the trajectory based on the prediction without observation (dead reckoning). compare results to GT and explain.

   iv. Plot and analyze the estimated x-y-θ values separately and corresponded sigma value along the trajectory. (e.g. show in one graph $x_{estimated}$-x_{GT} and σ_x and explain your results).

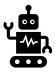
3. EKF-SLAM

   a) Load attached inputs and m 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 (σ_{rot1}^2=0.01, σ_{trans}^2=0.04, σ_{rot2}^2=0.01).

Apply Extended Kalman SLAM filter! (15%) 

The goal:

$$minimize\ RMSE\ while\ maxE < 1.5$$

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

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

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

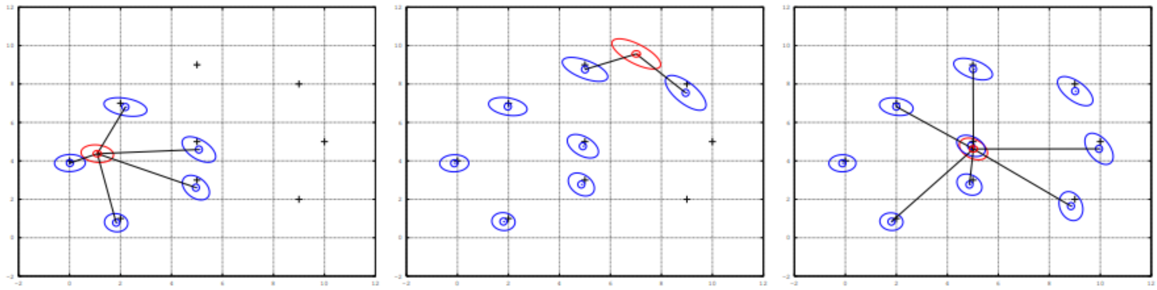
$$20 \le i \le N$$

N is last sample.

   d) Initialize initial conditions μ_0, Σ_0
   e) Implement the prediction step of the EKF SLAM algorithm in the file prediction_step.m. 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



$t = 50$ $t = 150$ $t = 330$

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

h) Implement the correction step in the file correction_step.m.

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

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

You should compute a block Jacobian matrix $\boldsymbol{H_t}$ by stacking the $\boldsymbol{H_t^i}$ 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 $\boldsymbol{Q_t}$ is a diagonal square matrix as follows
(σ_r^2=0.5, $\sigma_\emptyset^2$=0.3).

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 of state vector as ellipse.
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.

Simulate EKF-SLAM by yourself! (5% bonus!)

j) Create your own ground truth:
- Use the landmarks given in the previous section.
- Move the robot from the initial location [0,0] based on the odometry motion model $(rot1, trans, rot2)$. The trajectory should pass by the landmarks.
  Save the odometer motion commands in each iteration and the

trajectory waypoints [x,y] as the GT.
- The distance of the trajectory should be at least 10 pixels during at least 100 steps.

k) Simulate trajectory including motion and sensor data:

- During the movement of the robot search for landmarks inside known gating radius around the robot. (e.g radius=5m)

  Save the observed landmarks (ID) and corresponded measurements [range,bearing] in each iteration.

l) . Run EKF-SLAM on your own simulation!
Assume noises in the motion model and sensors model as you wish.
Repeat section 3.f

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 suggest solutions 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 ,movies, 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.

Reoprt Format.docx