Exploring Kalman Filter in a V2I Implementation

Author: Ali Badreddine (UMID-47560952)
Affiliation: University of Michigan

ABSTRACT

Vehicle transportation has evolved significantly with the introduction of smart technology around us. "Driver assistance features" such as "Lane Keep Assist", "Adaptive Cruise Control", and "Adaptive Emergency Braking" have become common in today's vehicles. With Tesla's Auto Pilot Feature being a great hit, customers expect smarter features and want the vehicle to do more of the driving. As a result, we've seen the industry shift its research heavily to develop and explore partial automation that builds on the level 1 ADAS features with a hope of a future where full automation could become a reality. However, in order to implement such features many advancements are required.

Vehicles need to be more aware and receive measurements/data of their surroundings, this can be done through smart sensors installed on the vehicle, as well as information received from surrounding radar/traffic towers and other vehicles. The concept of "Vehicle to Vehicle (V2V)" and "Vehicle to Infrastructure (V2I)" [7][8] is at the core of imagining an autonomous future. Second, vehicles need to be able to use this data intelligently. This involves semantically labeling and mapping their surroundings, accurately estimating the vehicle state at any given moment, and path planning intelligently and dynamically in order to safely reach their destination while respecting traffic laws and regulations. This is no easy feet and requires knowledge and expertise in robotics and machine learning concepts as well as ample data and time to train a model that serves these functions. At the core of many ADAS and autonomous features is state estimation. This can be done using many techniques from, UKF ("Unscented Kalman Filter") [5], "Particle Filtering" [10], EKF ("Extended Kalman Filter") [2] etc.… This paper will explore one of the first and common technique used in state estimation; the basic yet effective Kalman Filter. A simulation toy example will be introduced to illustrate its feasibility and discuss the results and some of its short coming.

INTRODUCTION

State estimation is a common topic in robotics circles and one that has become very popular in autonomous applications. Without accurate state estimation autonomy features would not be feasible. One of the most commonly discussed state estimation techniques is the Kalman Filter. This method propagates a probability distribution through a non-linear function using Taylor Series Expansion. Linearizing using Taylor Series Expansion means we need to compute the first order Jacobian, evaluate the Jacobian around a point, and represent the non linear function in its linearized form. This allows us to treat the non-linear function as an affine transformation; and for a gaussian distribution we can extract the propagated mean and covariance according to $y \sim \mathcal{N}(F\mu + x_0, F\Sigma F^T)$ where F is the Jacobian, μ is the current state estimation, and Σ is the covariance.

In a Kalman Filter (KF) generally this process is done in two steps; both are based on recursive Bayes filter.

The first is the prediction step where the predicted state is estimated as propagated through the affine transformation of the vehicle motion model; where the motion of the vehicle is generally accompanied by noise that may result from sensor drift for example.

The second, is the correction step where we rely on the measurements received or recorded by the vehicle these measurements are generally noisy and are compared to the measurements acquired as a result of the predicted state as propagated through the Taylor Series Expansion of the measurement model. This then yields a state and covariance for the vehicle as estimated at this time by the Kalman Filter.

In this paper I will implement these techniques on a toy example. The goal is to explore the Kalman Filter in an application that represents a V2I environment. The trajectory data that will be estimated here will be based on the extracted 2D (x, y coordinates) ground truth data from the KITTI data [12] set 07 sequence as parsed by SLO (a past collaborative object I was involved in; referenced below [11]). This provides the Special Euclidian 3 trajectory data with the rotation and translation coordinates.

To simulate a V2I environment the data will be used to generate noisy measurements from 3 separate radar towers that will return the distance of the vehicle from each radar tower at each time step. These noisy measurements will then be propagated through a KF to estimate the current state of the vehicle in the 2D environment and then compare the results to the ground truth. I will assume no knowledge of the vehicle motion (random walk motion model) and start with an initial guess for the state.

WORK DONE

The 2D ground truth trajectory of the 07 KITTI data sequence [12], as well as the x, y coordinates for the 3 radar towers are displayed below.

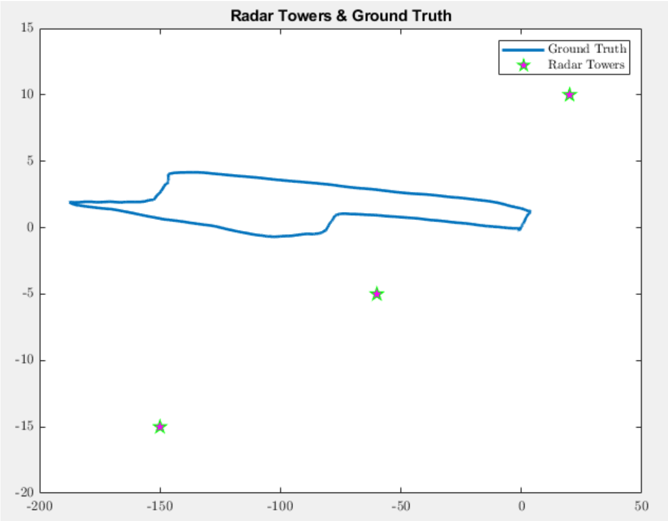

Figure 1 Ground truth trajectory & Radar tower positions

The set of measurements to be used are generated by first computing the distance from the ground truth states at each time instant to each of the respective radar towers by using ℓ_2 norm. Next noise is applied to the measurements by sampling from a zero mean gaussian distribution of noise covariance defined as $\begin{bmatrix} \sigma_{r1r1}^2 & 0 & 0 \\ 0 & \sigma_{r2r2}^2 & 0 \\ 0 & 0 & \sigma_{r3r3}^2 \end{bmatrix}$ where $\sigma_{r1r1}^2 = \sigma_{r2r2}^2 = \sigma_{r3r3}^2 = 0.04^2$, are the variances for each of the radar towers. This yields a set of noisy measurement that will represent the radar tower data communicated to the vehicle.

In order to successfully run the Kalman Filter a "good enough" initial guess of the vehicle state is needed. This can be done by using the first set of radar tower measurements received by triangulation of the robot in relation to the three reference points (radar towers). Thus, solving the two equations $C_1 = C_2$, and $C_2 = C_3$for the x, and y coordinates yields the initial vehicle state; where $C_{1,2,3}$ are the respective equations of the circle between the vehicle state and each radar tower.

This initial guess can then be used to propagate the state through the first step of the Kalman Filter; the prediction step.

In the prediction step; the vehicle motion is assumed as a random walk motion model, and thus the state is propagated as is in this step with a motion Jacobean of $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then the predicted state $\mu_{pred} = \mu$, where $\mu_{pred} = \begin{bmatrix} x \\ y \end{bmatrix}$, and μ is the previously estimated state. The motion noise covariance Q is $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ and then the predicted covariance is $\Sigma_{\mathrm{pred}} = \mathrm{A}\Sigma\mathrm{A}^{\mathrm{T}}$. In summary, we end up with an adjusted predicted state and covariance defined as $\mathcal{N}(\mu_{pred}, \Sigma_{\mathrm{pred}}) = \mathcal{N}(A\mu, Q\Sigma Q^T)$.

The second step of the Kalman Filter is the measurement step. Since the measurements are defined as the distance between the vehicle and the respective radio towers we can define the measurement model as $z' = \begin{bmatrix} z_1' \\ z_2' \\ z_3' \end{bmatrix} = \begin{bmatrix} norm(\mu_{pred} - r_1) \\ norm(\mu_{pred} - r_2) \\ norm(\mu_{pred} - r_3) \end{bmatrix}$; where $r_{1,2,3} = \begin{bmatrix} x_{r1,r2,r3} \\ y_{r1,r2,r3} \end{bmatrix}$ respectively (i.e. the radio tower coordinated). To linearize this model using Taylor series expansion we then must evaluate the measurement model Jacobian:

$$H(\mu_{pred}, r_{1,2,3}) = \begin{bmatrix} \frac{d(norm(\mu_{pred}-r_1))}{dx_{pred}} & \frac{d(norm(\mu_{pred}-r_1))}{dy_{pred}} \\ \frac{d(norm(\mu_{pred}-r_2))}{dx_{pred}} & \frac{d(norm(\mu_{pred}-r_2))}{dy_{pred}} \\ \frac{d(norm(\mu_{pred}-r_3))}{dx_{pred}} & \frac{d(norm(\mu_{pred}-r_3))}{dy_{pred}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{x_{pred}-x_{r1}}{q_1} & \frac{y_{pred}-y_{r1}}{q_1} \\ \frac{x_{pred}-x_{r2}}{q_2} & \frac{y_{pred}-y_{r2}}{q_2} \\ \frac{x_{pred}-x_{r3}}{q_3} & \frac{y_{pred}-y_{r3}}{q_3} \end{bmatrix}$$

Where $q_{1,2,3} = \sqrt{(x_{pred} - x_{r1,2,3})^2 + (y - y_{r1,2,3})^2}$

With the propagated measurements z' we can compute the innovation (v) which is defined as the difference between the propagated measurements and noisy measurements (z) → $v = z - z'$.

Next, we evaluate the Kalman gain as:

$K = \Sigma_{\mathrm{pred}}\mathrm{H} + \mathrm{R}$ [10]

Where R is the measurement noise covariance $\begin{bmatrix} 0.04^2 & 0 & 0 \\ 0 & 0.04^2 & 0 \\ 0 & 0 & 0.04^2 \end{bmatrix}$, and $S = H\Sigma_{\mathrm{pred}}H^T + R$ [10]

Finally, we can compute the final corrected state at this time instant as $\mu = \mu_{pred} + Kv$,

and covariance $\Sigma = (\mathrm{I} - \mathrm{KH})\Sigma_{\mathrm{pred}}(\mathrm{I} - \mathrm{KH})^T + KRK^T$ [10]

This KF output is the state as estimated at the next time instant; we repeat this process for all noisy measurements to end up with an estimated trajectory.

RESULTS

To evaluate the performance of the filter a plot of the trajectory of the Kalman Filter State approximation vs the ground truth is displayed in Fig. 2 below.

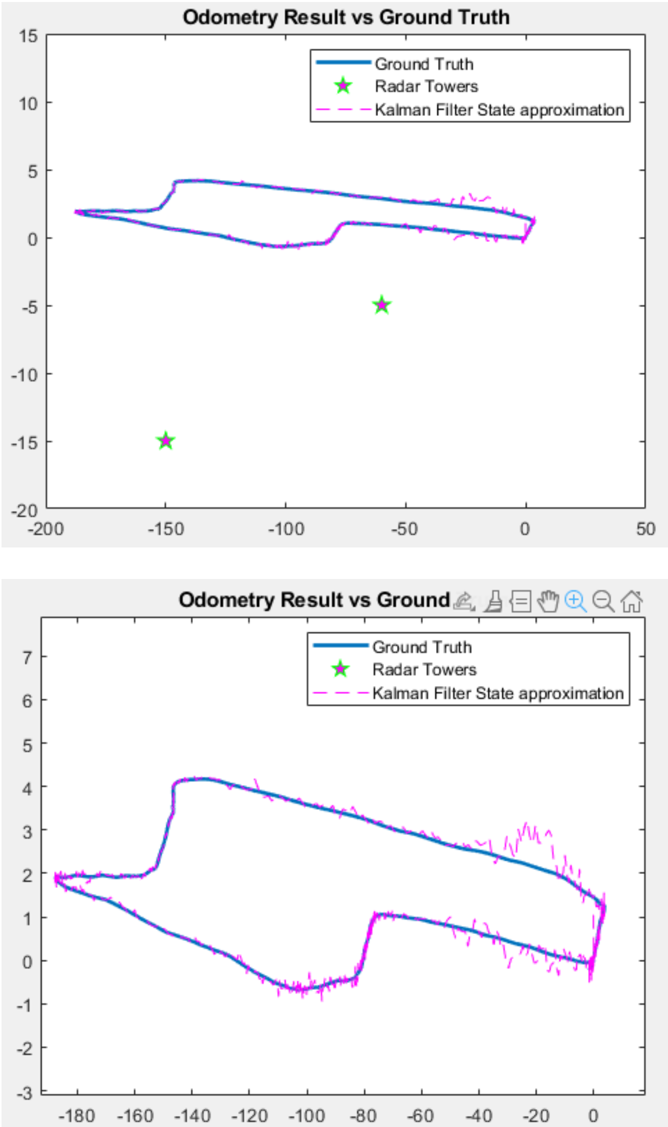

Figure 2 Kalman Filter State Approximation vs Ground Truth

Visually the Kalman Filter approximation mirrors the trajectory reasonably well with some noisy state approximations that result in oscillations away from the ground truth.

Next, Fig. 3 displays the deviation of the state estimate vs the ground truth for the x coordinate as well as the y coordinate. It is apparent that the deviation error remains within the 3-sigma contours evaluated as $3\sigma_x$ for the x coordinate and $3\sigma_y$ for the y coordinate. Where $\Sigma = \begin{bmatrix} \sigma_x^2 \\ \sigma_y^2 \end{bmatrix}$ is the covariance of the state approximated by the Kalman Filter at each time instant.

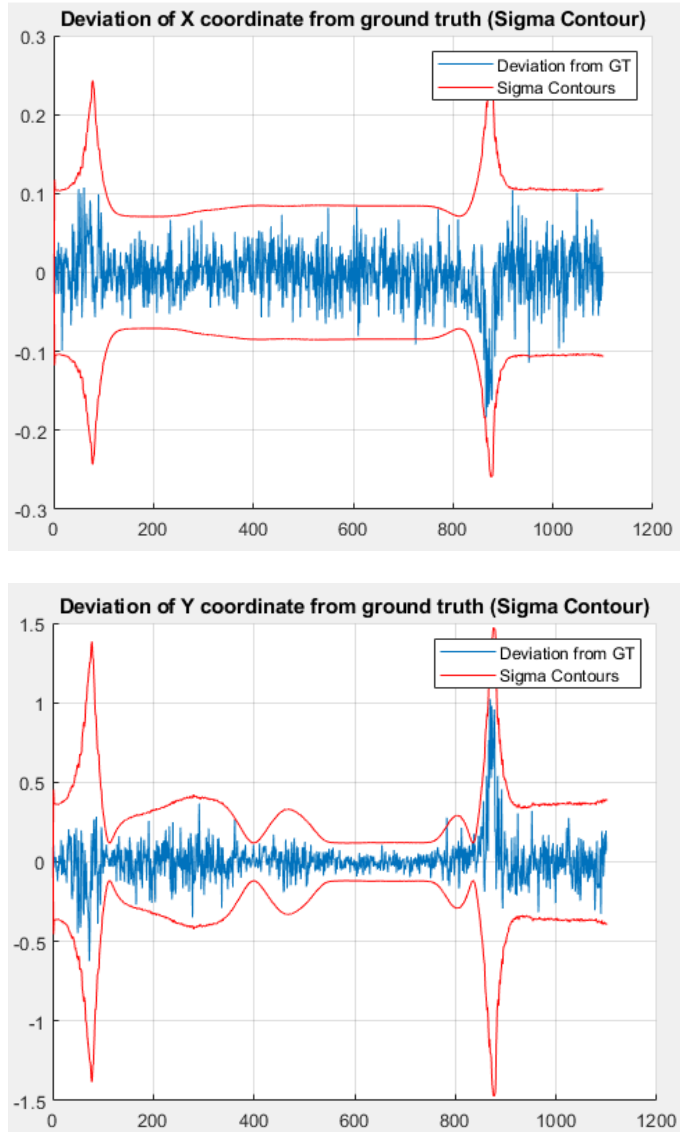

Figure 3 Deviation from ground truth for x, and y coordinates

Lastly the MSE yielded is 4.5384.

The results are promising and show the effectiveness of the Kalman Filter in state estimation for a linear example (no rotation) as well as exposing the feasibility of Kalman Filter in V2I applications and smart vehicles.

However, one of the major downsides of KF is sensor error and drift which can be witnessed in the beginning and end of the trajectory where the oscillations worsen. KF also struggles with non-linear motion and measurement models. There exist the EKF Extended Kalman Filter which builds on KF for nonlinear models and works for multimodal distributions (translations & rotations). Additionally, the robotics community has already explored solutions for drift, and one of the major industry leading techniques is “SLAM: Simulations Localization and Mapping” [10]. SLAM attempts to correct drift errors by observing landmarks and establishing loop closures that correct odometry effectively.

CONCLUSION

The world of smart vehicles and smart highways is in its infancy and welcomes a lot of creativity and imagination when it comes to possible implementations. However, it is not without its challenges, the considerations required to implement autonomous and smart features safely in vehicle systems are extensive. One necessary technique used in partial and full autonomy application is state estimation and “V2X (Vehicle to Everything) communication” [8]. In this paper I explored one of the most common techniques used in state estimation; the Kalman Filter. I evaluated the predicted state and corrected state to come up with a final trajectory based on noisy measurements generated by the 2D equivalent of 07 KITTI data set [12]. I also utilized a set of radar towers to provide the vehicle measurements to explore a basic implementation of a V2I system. Results were promising but KF is not the most optimal and suffers from drift as well as limitations for non-linear models.

REFERENCES

1. Q. Li, R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application," 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Tianjin, 2015, pp. 74-77, doi: 10.1109/ICINIS.2015.35.
2. H. Wang, J. Wang, L. Qu and Z. Liu, "Simultaneous localization and mapping based on multilevel-EKF," 2011 IEEE International Conference on Mechatronics and Automation, Beijing, 2011, pp. 2254-2258, doi: 10.1109/ICMA.2011.5986290.
3. P. J. Hargrave, "A tutorial introduction to Kalman filtering," IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments, London, UK, 1989, pp. 1/1-1/6.
4. H. Khazraj, F. Faria da Silva and C. L. Bak, "A performance comparison between extended Kalman Filter and unscented Kalman Filter in power system dynamic state estimation," 2016 51st International Universities Power Engineering Conference (UPEC), Coimbra, 2016, pp. 1-6, doi: 10.1109/UPEC.2016.8114125.
5. M. G. Jadidi and E. Hashemi, "Optimal preview control of the Nao biped robot using a UKF-based state observer," 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, 2016, pp. 52-57, doi: 10.1109/AIM.2016.7576742.
6. Q. He, C. Wei and Y. Xu, "An improved adaptive Kalman filtering algorithm for balancing vehicle," 2017 Chinese Automation Congress (CAC), Jinan, 2017, pp. 5721-5725, doi: 10.1109/CAC.2017.8243804.
7. Steven E. Underwood, “Social and Institutional Considerations in Intelligent Vehicle-Highway Systems,” SAE Technical Paper 901505, 1990, doi: 10.4271/901505
8. N. Varga, L. Bokor, A. Takács, J. Kovács and L. Virág, "An architecture proposal for V2X communication-centric traffic light controller systems," 2017 15th International Conference on ITS Telecommunications (ITST), Warsaw, 2017, pp. 1-7, doi: 10.1109/ITST.2017.7972217.
9. Author's Last Name, First Name Middle Name or Initial. *Title of Book: Subtitle of Book*. Edition, Publisher, Year of print publication.
10. Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic Robotics*. Cumberland: MIT Press.
11. https://github.com/tglenn28/SLO/tree/master/SLAM_ON_LIDAR
12. http://www.cvlibs.net/datasets/kitti/

CONTACT

Ali Badreddine
abadredd@umich.edu