ANALYSIS OF KALMAN FILTER PERFORMANCE ON LINEAR AND NON-LINEAR VEHICULAR MOTION

Nathaniel Gordon *Eric Grimaldi* *Siddhant Gupta*

Northeastern University

ABSTRACT

With the growing prevalence of autonomous vehicles, the need for methods to predict and model the motion of said vehicles is ever increasing. One such method for trajectory estimation is the Kalman filter. In this project, we derive and implement two versions of a Kalman filter that are able to harness RADAR and LIDAR data of a simulated autonomous vehicle to predict the vehicle's motion with an uncertainty of 4cm. This project demonstrates the effectiveness of the linear and lxtended variations of the Kalman filter, particularly when used in conjunction.

1. INTRODUCTION

Before proceeding to the applied Kalman filters, the algorithm will be defined in a general sense. The Kalman filter is built around two fundamental phases: the prediction phase, in which a state estimation is produced, and an update phase, during which uncertainty in the state space is propagated for use in the next iteration of the algorithm. The prediction phase can be defined as follows:

$$x' = Fx + u$$

Here, the state space and its projected prediction are denoted x and x' respectively, and u is the control term. F is a representation of the dynamics of the object's motion.

The update phase, in which the future state uncertainty P' is determined, can be represented as a function of the prior uncertainty and measurement noise Q:

$$P' = FPF^T + Q$$

Once again, the dynamics matrix F is present. Defining this quantity will be a crucial part of the implementation step.

2. METHODS

2.1. Linear Kalman Filter Implementation

First, the object dynamics for the autonomous vehicle must be discussed. Two values are pertinent to the prediction of the vehicle's motion: position and velocity. For the linear Kalman filter, linear motion will be assumed:

$$F = \begin{cases} p_x' = p_x + v_x \Delta t \\ p_y' = p_y + v_y \Delta t \\ v_x' = v_x \\ v_y' = v_y \end{cases}$$

Because there is no aspect of control in this application, we can then represent the prediction equation as:

$$\begin{pmatrix} p_x' \\ p_y' \\ v_x' \\ v_y' \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \end{pmatrix}$$

Now, the data may be applied to the algorithm. The data used for this project were borrowed from [1]. One source of the measurements present in the data is LIDAR. These readings contain a measurement of the vehicle's position in Cartesian space. Let these readings be denoted z, the measurement vector:

$$z = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

Because we are unable to directly reason about the vehicle's velocity from the LIDAR input, we will define a mask function H to apply the LIDAR data to only the positional aspects of the state vector:

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

Additionally, the instrument error for the LIDAR readings is 2.25 centimeters.

With these values defined, the update process for the LIDAR readings becomes the following:

$$P' = HPH^T + Q$$

The error, y, is the difference between the true reading and the masked projection:

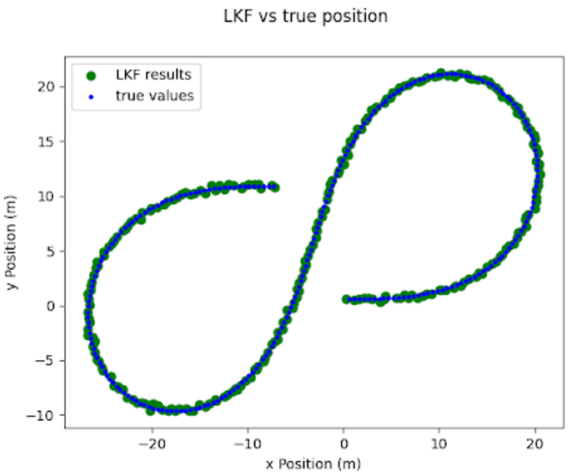

Fig. 1. Positional estimates generated from linear Kalman filter and LIDAR data.

$$y = z - Hx'$$

Finally, the Kalman gain, K, which describes the relative weight placed on the recent measurements (as opposed to older ones), can be derived as thus:

$$K = P'H^T P^I$$

2.2. Extended Kalman Filter Implementation

The other source of readings in the data is a RADAR system. These readings are given in polar form, and are a description of the vehicle's range ρ, perpendicular velocity $\dot{\rho}$, and bearing ϕ. To convert to Cartesian form, the following transformation is applied:

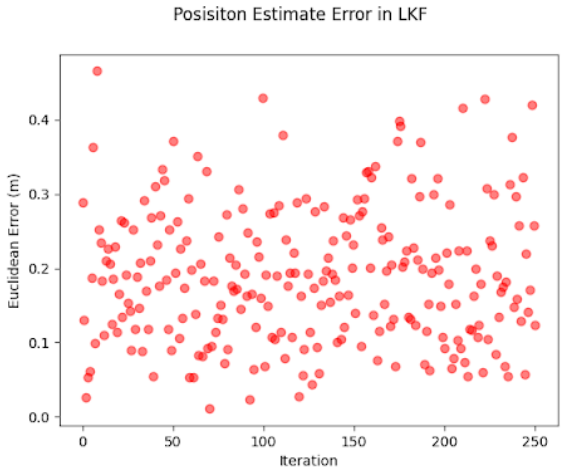

Fig. 2. Error (Euclidean distance) in linear Kalman filter estimation.

$$h(x') = \begin{pmatrix} \sqrt{p_x'^2 + [_y'^2} \\ arctan(p_y'/p_x') \\ \frac{p_x'v_x' + p_y'v_y'}{\sqrt{p_x'^2 + p_y'^2}} \end{pmatrix}$$

This transformation raises a pertinent issue: it introduces non-linearity into the system. To handle this linearity, the extended Kalman filter methodology is employed.

To adapt the algorithm to account for this non-linearity, the Jacobian of the transformation matrix h_j must be calculated, as derived by [1]:

$$h_j =$$

$$\begin{bmatrix} \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} & 0 & 0 \\ -\frac{p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 & 0 \\ \frac{p_y(v_x p_y - v_y p_x)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_y(v_y p_x - v_x p_y)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} \end{bmatrix}$$

The rest of the process of constructing the extended Kalman filter is identical to the methods in the previous section, where the linear variant was defined.

3. RESULTS

3.1. Linear Kalman Filter Results

To test the performance of the Kalman filter algorithms, a simplified trial using only LIDAR data was run to isolate the linear Kalman filter algorithm. This meant that while the procedure was unable to produce an estimate of the vehicle's velocity, it could predict its position. The trial contained 250 LIDAR readings of the vehicle traveling in a figure-eight pattern.

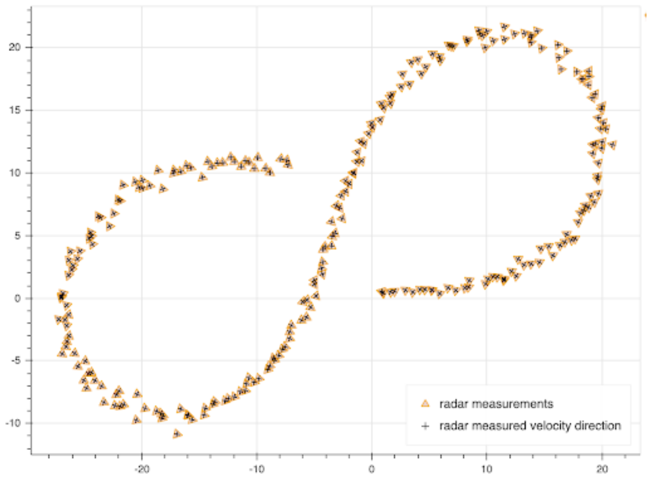

Fig. 3. Compiled LIDAR and RADAR readings from the dataset. The orientation of the triangle markers corresponds to the vehicle's measured bearing.

The results of this trial are visible in Figure 1, where the predicted positions are plotted alongside the corresponding true values. Additionally, in Figure 2, the error, found with Euclidean distance from the true value. For this trial, the empirically determined root mean squared error was found to be .21 meters.

While the linear Kalman filter generates accurate predictions of position, it is unable to capture the full state of the vehicle in motion. Thus, subsequent trials were run to determine if using pure RADAR data, or a mix of the two data streams, would improve the prediction quality.

3.2. Extended Kalman Filter Results

After the linear Kalman filter had been tested independently, a second trial was run, using only the RADAR data from the autonomous vehicle simulation. In this trial, the procedure was able to estimate the complete state vector, capturing both the position and velocity of the vehicle in the prediction. The results of this trial are visible in Figure 4.

Finally, a trial was run where both Kalman filters were run in tandem, analyzing both LIDAR and RADAR data. In estimating the vehicle's motion, the linear Kalman filter serves as the more accurate arbiter of positional data, as it draws from the less noisy LIDAR instrument. However, this model also has the benefit of ascertaining velocity information from the extended Kalman filter. While drawing on the RADAR data is necessary for projecting the vehicle's future state, the graph in Figure 5 demonstrates how the positional data gathered from the extended Kalman filter is much less accurate due to the RADAR's lesser precision. In conjunction, the two filters provide a robust estimation of vehicle movement, drawing from both linear and non-linear data streams.

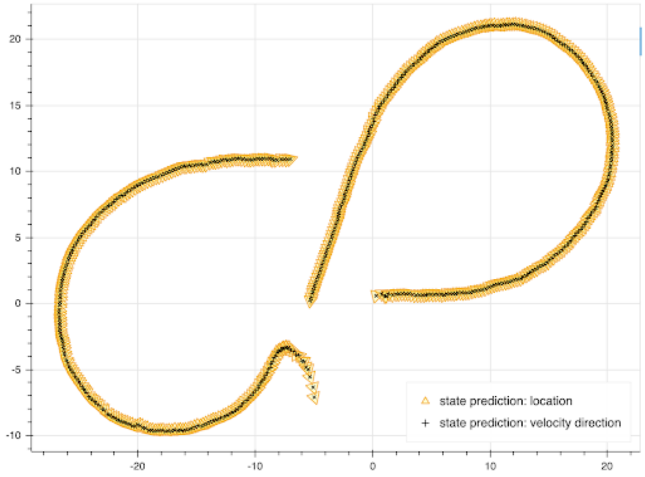

Fig. 4. Predicted vehicle states collected from using only the extended Kalman filter on the RADAR data.

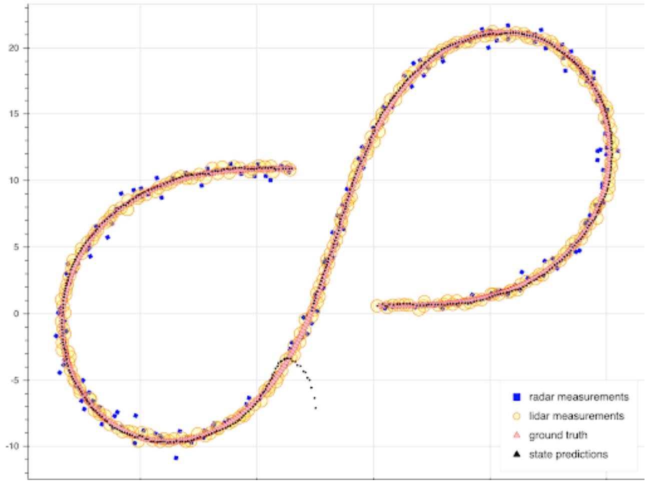

Fig. 5. Overlaid results of drawing predictions from both linear and extended Kalman filters. The state predictions draw on both LIDAR and RADAR data.

4. CONCLUSION

The outcomes of the three trials indicate that the Kalman filter is a highly flexible tool, and can be used in tandem with other filters to process data from multiple sources. Furthermore, the uncertainty measurement the filter maintains enables it to rely on measurements of different inputs for parts of the state vector, minimizing overall error in the output.

An future extension to this project could entail observing the behavior of the system with changes to the degree of noise in each measuring instrument. Alternatively, the use of an unscented Kalman filter for processing the non-linear data could be investigated.

This project served as an excellent opportunity for all three group members to gain experience implementing and testing the filtering techniques demonstrated in this report. The source code for this report can be found at [2].

5. REFERENCES

[1] O. Gaskey, "Extended kalman filter," https://github.com/OanaGaskey/Extended-Kalman-Filter, 2019.

[2] Grimaldi E. Gordon, N. and S. Gupta, "EECE5644 kalman filters," https://github.com/gordonng123/EECE5644_Kalman_Filters, 2020.