Vehicle state estimation using EKF and Particle Filters

Dimitrios Triantafyllidis

August 10, 2022

1 Summary

The purpose of this work is to estimate the position and orientation of a vehicle moving on the level, while being monitored by two fixed radars. This process is implemented twice through the use of ExtendedKalman and Particle filters. Initially, the three states of the vehicle are estimated for each time using data from the first radar only. Then, the data of the second radar is used and the results are compared.

Keywords: ExtendedKalmanfilter, EKF, Particlefilter, filteringalgorithms, position estimation, nonlinear dynamic system.

2 Problem Description

Let a vehicle move in plane (2 dimensions). The vehicle is monitored by a fixed radar on the same plane as the vehicle. The estimator should calculate each time the best estimate for the position and orientation of the vehicle. The vehicle's motion model is given by Eq.

$$x_{t+1} = \begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ \theta_{t+1} \end{bmatrix} = f(x_t, u_t) = \begin{bmatrix} X_t + \cos(\theta_t) v_t dt \\ Y_t + \sin(\theta_t) v_t dt \\ \theta_t + \omega_t dt \end{bmatrix} \tag{1}$$

While the radar measurement model is given by the equation:

$$h(x_t) = \begin{bmatrix} d_t \\ \phi_t \end{bmatrix} = \begin{bmatrix} \sqrt{(X_t^o - X_t)^2 + (Y_t^o - Y_t)^2} \\ atan2(Y_t - Y_t^o, X_t - X_t^o) - \theta_t \end{bmatrix} \tag{2}$$

Where X^o and Y^o are the radar coordinates. The noise entering the measurement system is Gaussian with mean 0 and standard deviation σ. d_t denotes the distance of the radar from the vehicle and ϕ_t denotes the angle the vehicle is detected with respect to its position.

Question 1: Estimate the 3 states of the vehicle at each moment using radar data. The estimate should be done with Extended Kalman Filter and additionally with the Particle filter.

Question 2: A 2nd radar is added to the problem which is placed in a distance of 10m on the x-axis from the 1st radar. Repeat the estimates as in the 1st question but now using both radars at the same time.

Dataset: Sampling is done with a frequency of 10 Hz.

The file control contains the measured readings of velocity u and ω with noise with a standard recall of 0.05 and 0.1 respectively, while the file radar1 contains the noisy measurements from the vehicle (d1, ϕ1).

The radar2 file contains the noisy measurements of the vehicle from the second radar (d2, ϕ2).

The noise of the first radar has a mean value of 0 and a standard deviation of 0.2rad at angle and 1m in distance measurement.

The noise of the second radar has mean value 0 and standard deviation 0.05rad in the angle and 0.3m in the measurement of distance.

3 Extended Kalman Filter

3.1 Introduction

ExtendedKalman filter can be described as an "extension" of the Kalman filter for nonlinear dynamical systems. The problem is described with two system models: motion model and measurement model, whose general form is illustrated below respectively [Kal], [Ext].

$$
\begin{aligned}
x_t &= f(x_{(t-1)}, u_t) + w_t \\
z_t &= h(x_t) + v_t
\end{aligned}
$$

, where w_t and v_t are the process and observation noises which are both assumed to be zero mean multivariate Gaussian noises with covariance Q_t and R_t respectively and u_t is the control vector $f(x_{(t-1)}, u_t)$ and $h(x_t)$ are the nonlinear functions (one or both) describing the motion and measurement model.

The algorithm is divided into two steps:

1. Prediction (Computation of the predicted state from the previous estimate) [Kal], [Ext].

$$
\begin{aligned}
\hat{x}_t &= f(\hat{x}_{(t-1)}, u_t) \\
P_t &= F_t P_{t-1} F_t^\top + Q_t
\end{aligned}
$$

2. Update (First compute the predicted measurement from the predicted state and update state estimate taking into consideration the measurements from the radars) [Kal], [Ext].

$$
\begin{aligned}
\textit{Innovation or measurement residual:} \quad & \tilde{y}_t = z_t - h(\hat{x}_t) \\
\textit{Innovation or residual covariance:} \quad & S_t = H_t P_t H_t^\top + R_t \\
\textit{Near Optimal Kalman Gain:} \quad & K_t = P_t H_t S_t^{-1} \\
\textit{Update State Estimate:} \quad & \hat{x}_t = \hat{x}_t + K_t \tilde{y}_t \\
\textit{Update Covariance Estimate:} \quad & P_t = (I - K_t H_t) P_t
\end{aligned} \tag{3}
$$

where the state transition and observation matrices are defined to be the following Jacobians

$$
\begin{aligned}
F_t &= \frac{\partial f}{\partial x}\Big|_{\hat{x}_{t-1}, u_t} \\
H_t &= \frac{\partial h}{\partial x}\Big|_{\hat{x}_{t-1}}
\end{aligned}
$$

The partial derivatives of h and f (the Jacobian) are computed because they are needed for the calculation of the covariances (predicted and measurement). The Jacobian is evaluated with current predicted states. Using these matrices in the Kalman Filter the non-linear function is linearized around the current estimate.

3.2 Solution

1. The state transition and matrix can be found as following:

$$
F_t = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -dtv\sin(\theta_t) \\ 0 & 1 & dtv\cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \tag{4}
$$

2. The observation transition and matrix can be found as following:

$$
H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{-X^o + x}{\sqrt{(X^o - x)^2 + (Y^o - y)^2}} & \frac{-Y^o + y}{\sqrt{(X^o - x)^2 + (Y^o - y)^2}} & 0 \\ \frac{Y^o - y}{\sqrt{(X^o - x)^2 + (Y^o - y)^2}} & \frac{-X^o + x}{\sqrt{(X^o - x)^2 + (Y^o - y)^2}} & -1 \end{bmatrix} \tag{5}
$$

3. We are going to suggest two solutions with the one being the more precise and a heuristic one.

   (a) We may now concentrate on the noise. The noise in this instance is in control space since it is in our control input $u(v, \omega)$, In other words, we command a specific velocity and steering angle, but we need to convert that into errors in X, Y, θ. In a real system this might vary depending on velocity, so it will need to be recomputed for every prediction [Lab].

$$M = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega \end{bmatrix} \tag{6}$$

   If this was a linear problem we would convert from control space to state space using the by now familiar $FQF^\top$ form. Since our motion model is nonlinear we do not try to find a closed form solution to this, but instead linearize it with a Jacobian which we will name V. [Lab].

$$V = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \omega} \\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \omega} \\ \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \omega} \end{bmatrix} \tag{7}$$

   and then final form of the Predicted covariance estimate is as following:
   $P_t = F_t P_{t-1} F_t^\top + VMV^\top$

   (b) Another solution would be to define the array (3x3) giving values heuristically (maybe as following).

$$Q = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} = \begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.0025 & 0 \\ 0 & 0.0025 & 0.01 \end{bmatrix} \tag{8}$$

4. Predicted covariance estimate for the time $t = 0$. Since we are not sure about the starting point of the vehicle (the radar gave us the first measurement) it would be wise to use big values for the predicted covariance estimate, specially when only the first radar is taken into consideration because the noise of the first radar has a standard deviation of 0.2rad at angle and 1m in distance measurement (much bigger than that of the first radar). So we define the Predicted covariance estimate heuristically as illustrated below:

$$P_{t=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

5. The observation noises for the first and second radar can be calculated as below:

$$R_1 = \begin{bmatrix} \sigma_{1d}^2 & 0 \\ 0 & \sigma_{1\phi}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.04 \end{bmatrix} \tag{10}$$

$$R_2 = \begin{bmatrix} \sigma_{2d}^2 & 0 \\ 0 & \sigma_{2\phi}^2 \end{bmatrix} = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.0025 \end{bmatrix} \tag{11}$$

6. In case both radars are used the observation noises can be calculated as below:

$$R_{12} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0.09 & 0 \\ 0 & 0 & 0 & 0.0025 \end{bmatrix} \tag{12}$$

3.2.1 Using only the first radar

To find the initial position of the vehicle at t=0, the first readings of the first radar were used and the 2x2 system was solved, which is shown below:

$$\begin{bmatrix} d_0 \\ \phi_0 \end{bmatrix} = \begin{bmatrix} \sqrt{(X_t)^2 + (Y_t)^2} \\ atan2(Y_t, X_t) \end{bmatrix} \tag{13}$$

where X_0 and Y_0 the initial positions of the vehicle we are looking for, d_0, ϕ_0, the predictions of the first radar (0,0). The initial angle of the vehicle was assumed to be θ_0=0. Considering the d_0, ϕ_0 as found in dataset the initial coordinates for the vehicle are :

$$\begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 4.58 \\ 1.36 \\ 0 \end{bmatrix} \tag{14}$$

3.2.2 Using Both Radars

In this request, the same motion model and measurement model 2 are used. However, a second radar is added which is placed at a distance of 10 meters on the X-axis from the first radar. Basically, the procedure followed is similar to the previous query. The measurements of the second radar contain noise with covariance R_2:

As mentioned, the location of the second radar is described by the coordinates (10,0). Therefore, in the measurement model the coordinates X^o and Y^o take the new values 10 and 0 respectively.

Independent measurements of the two radars. In this case, the update step is done twice consecutively for the measurements of the two radars. It is important to note how this procedure was followed, as the two radars are considered independent of each other. First, estimates are calculated and then updated for the components of the first radar, and then the new updated estimates are updated again based on the measurements of the second radar.

If the measurements of the two radars depend to each other If this were not the case, the noises of the two radars would result in a 4x4 covariance matrix that would have this form:

$$R_{12} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0.09 & 0 \\ 0 & 0 & 0 & 0.0025 \end{bmatrix} \tag{15}$$

The state transition and observation matrices are now defined as follows:

$$F_{12_t} = \begin{bmatrix} F_1 t \\ F_2 t \end{bmatrix} \tag{16}$$

$$H_{12_t} = \begin{bmatrix} H_1 t \\ H_2 t \end{bmatrix} \tag{17}$$

,where F_1t , F_2t are the state matrices for the first and second radar respectively, and H_1t , H_2t are the observation matrices for the first and second radar respectively.

The covariance matrix Q remains the same as in subsection 3.1. To find the initial position of the vehicle, the first measurements of the second radar were used and the 2x2 system was solved. The second radar has less noise in its readings than the first, so it was chosen to find the initial state of the vehicle by taking into account the first readings of the second radar. Thus, the initial state of the system is obtained as:

$$\begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 3.7 \\ 3.85 \\ 0 \end{bmatrix} \tag{18}$$

4 Particle Filters

4.1 Introduction

Particle filters are a set of Monte-Carlo algorithms used to solve signal processing problems and have huge applications in robotics and engineering. The process is characterized by a set of elements (particles) that represent the distribution of some stochastic processes, given noisy measurements. The motion model can be both linear and non-linear.

The algorithm is briefly described in the following steps:

- Generation of random particles

  The particles that are "born" have the dimensions necessary to describe the motion model (position, angular velocity, etc.). Each particle gets a weight (a probability) that represents how likely it is to correspond to the current true state of the system.

- Prediction

  The particles move at the next time according to the motion model.

- Update

  After the measurements are taken into account, the weights are updated in the following way: the more the particle approaches the measurement, the more the value of its corresponding weight increases.

- Resampling

  This step is performed only when certain criteria are met, and not in every cycle of steps. With this method, particles that have a very low probability (weight) are rejected, and replaced by clones of particles that have a high enough probability.

- Estimate

  This step is optional. The average value and variance of the set of particles are calculated, in order to compare the estimate with the state of the system.

4.2 Solution and Maths

- Generation of random particles

  Generation of random particles N particles normally distributed around the initial state are generated, with some standard deviation (radar noise) for each dimension.

$$Q = \begin{bmatrix} \sigma_u \\ \sigma_u \\ \sigma_\omega \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.01 \end{bmatrix} \tag{19}$$

  All particles have weights, the value of which is defined according to how "close" they are to the measurement. In the first step, all particles are initialized with the same weight, and in fact their sum is equal to unity, so that all particles form a probability distribution. So the weight of each particle is $1/N$ at the initialization step and they are stored in an (N,) array. To track the vehicle the states for X, Y, θ are need to be maintained. They will be stored in a (N, 3) array considering there are N particles.

$$x_{t=0} = \begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} mean(X_0) + randn(N) * Q[0] \\ mean(Y_0) + randn(N) * Q[1] \\ mean(\theta_0) + randn(N) * Q[2] \end{bmatrix} \tag{20}$$

  , where the X_0, Y_0, θ_0 are actually the starting potitions of the vehicle as calculated earlier.

- Prediction.

  Essentially, the prediction of a particle's state is based on its last estimate (x_{t-1}) and the information from the control vector(u_t). This is mathematically expressed as:

$$p(x_t|x_{t-1}, u_t) \tag{21}$$

The vehicle's controls are not perfect, so its motion will not executed exactly as ordered. Therefore, it is considered necessary to add noise to the movements of the particles, so that there is a reasonable possibility of capturing the real movement of the vehicle based on the Q matrix.

$$x_{t+1} = \begin{bmatrix} X_{t+1} \\ Y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} X_t + \cos(\theta_t) \cdot dt \cdot (\upsilon_t + randn(N) \cdot Q[0]) \\ Y_t + \sin(\theta_t) \cdot dt \cdot (\upsilon_t + randn(N) \cdot Q[1]) \\ \theta_t + dt \cdot (\omega + randn(N) \cdot Q[2]) \end{bmatrix} \tag{22}$$

The array above has a shape of (N, 3).

- Update

  After the measurements are taken into account, the weights are updated in the following way: the more the particle approaches the measurement, the more the value of its corresponding weight increases. This can be mathematically captured by the following Bayes formula [Lab]:

$$P(x|z) = \frac{P(z|x)P(x)}{p(z)} = \frac{\textit{likelihood x prior}}{\textit{normalization}} \tag{23}$$

  ,where

$$P(z|x) = \frac{e^{(\frac{-1}{2})(x-\mu)^T \Sigma^- 1(x-\mu)}}{\sqrt{2\pi^k |\Sigma|}} \tag{24}$$

  ,where μ are the estimates of the states (mean values) converted to (d, ϕ) using observation model equations to be compared with the radar measurements, x, and k the number of variables of the Gaussian normal distribution, which are two in our case (d, ϕ). Finally Σ is the covariance / covariance matrix which is equal to R(Σ = R)

  $P(x|z)$ is called importance factor/weight. The following procedure is followed for all particles. Each particle has a position and a weight that estimates how well it matches the measurement. Then, the weights are normalized so that sum equals to unity to turn into a probability distribution. Those particles that are closer to the vehicle will have greater weight than those that are further away from it.

- Resampling

  In this work, the stratified resample algorithm was used. This method makes selections relatively uniformly across the particles. They way it performs is that it divides the cumulative sum into equal sections, and then selects one particle randomly from each section. This can guarantee that each sample is between 0 and [2/N] apart. However, other resampling techniques based on the select with replacement logic were also tried. Also, techniques selected in evolutionary algorithms such as tournament selection and roulette selection could be applied. The resampling will be performed only if N_eff is less than a predefined value, which ranges in the interval [N/2, N], where N_eff can be calculated as follows [Lab].

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} w_i^2} \tag{25}$$

- Estimate In most applications the estimated state must be computed after each update, but the filter contains nothing more than a collection of particles. Assuming that we are tracking an object (vehicle, airplane) the next estimate of the states can be calculated as the weighted average of the particles, calculated with the following formula [Lab]:

$$\mu^2_{particles} = \frac{1}{N} \sum_{i=1}^{N} x_i w_i \tag{26}$$

The Variance is also calculated with the following formula [Lab]:

$$\sigma^2_{particles} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{particles})^2 w_i \tag{27}$$

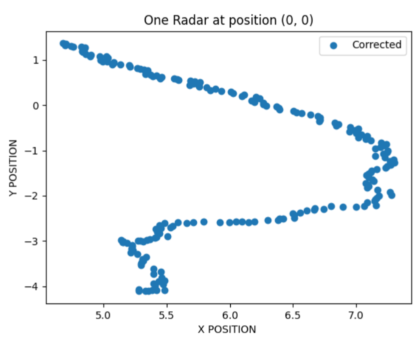
5 Results

To visualize the results, the diagram of the vehicle's path was created, according to its coordinates (X_t, Y_t).

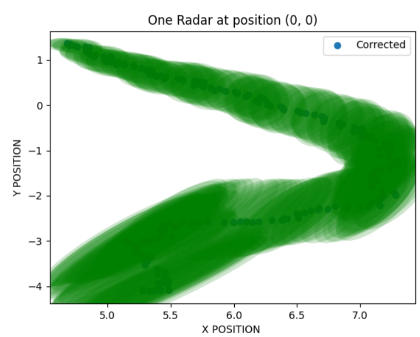
5.1 EKF

5.1.1 Using one Radar

Fig 1a shows the path of the vehicle, through the Kalman estimations. Its coordinates (X_t, Y_t) are those obtained after the update stage. In the figure 1b , the uncertainty ellipses around each point are also plotted. The ellipses reflect the size of the uncertainty of each estimate, or otherwise called areas of uncertainty. Essentially, the dimensions of an ellipse depend on the covariance matrix of the variables X_t and Y_t.



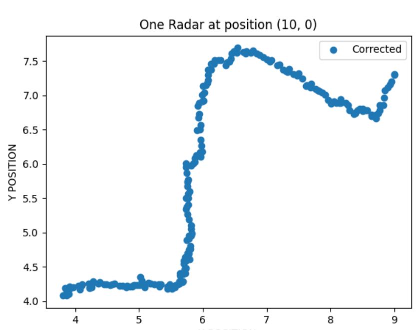
(a) Without error ellipses.



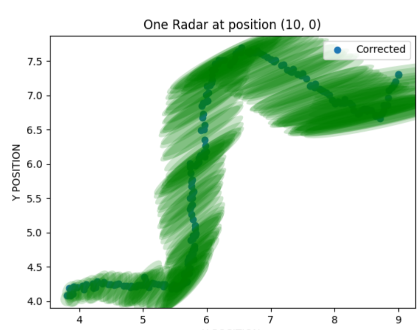
(b) With error ellipses.

Figure 1: [EKF] Vehicle tracking using the first radar

Similars Figures can be derived using the second radar and can be depicted in Figure 2.



(a) Without error ellipses.

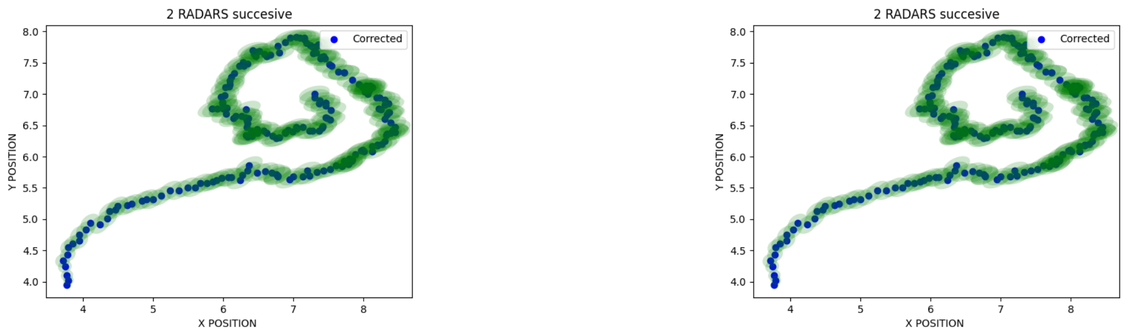


(b) With error ellipses.

Figure 2: [EKF] Vehicle tracking using the second radar

5.1.2 Using both Radars

Using both the radars the vehicles trajectory follows more that of the second radar (that was expected considering that the second's radar measurements are less noisy than the first's). The results can be decipted in Figure 3.



(a) Without error ellipses.

(b) With error ellipses.

Figure 3: [EKF] Vehicle tracking using both radars

5.2 Particle filters

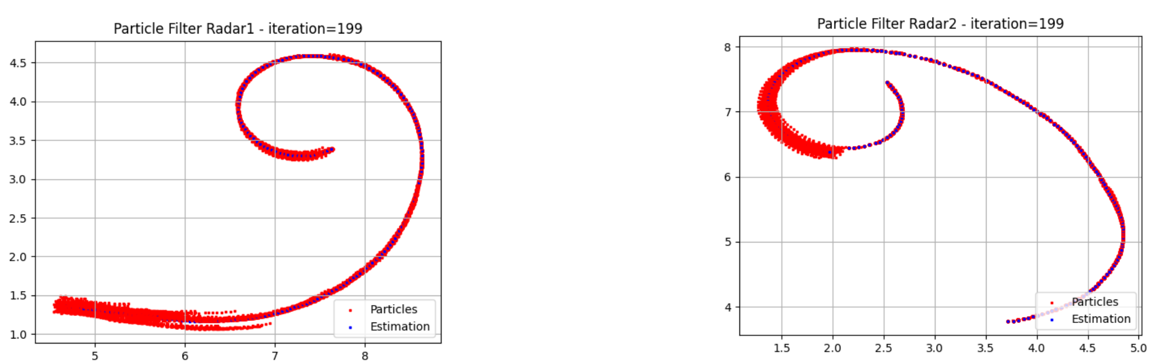
5.2.1 Using one Radar

As previsously explained, to find the initial position of the vehicle, the first measurements of each radar were taken into consideration and the 2x2 system of (Eq. 13) was solved. The results of the vehicles trajectory using only one of the radars can be depicted in Fig. 4

$$\begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 4.58 \\ 1.36 \\ 0 \end{bmatrix} \tag{28}$$

$$\begin{bmatrix} X_0 \\ Y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 3.7 \\ 3.85 \\ 0 \end{bmatrix} \tag{29}$$

the steps of Section 4 were performed. To visualize the results, the diagram of the vehicle's path (Figure 9).



(a) First radar

(b) Second Radar

Figure 4: [PF] Vehicle tracking using only one radar

5.2.2 Using both Radars

Using both the radars the vehicles trajectory follows more that of the second radar (that was expected considering that the second's radar measurements are less noisy than the first's). The results can be decipted in Figure 5.

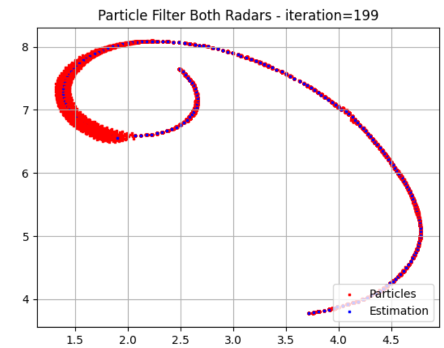


Figure 5: [PF] Vehicle tracking using both radars

References

[Ext] Extended kalman filter. https://en.wikipedia.org/wiki/Extended_Kalman_filter.

[Kal] Kalman filter. https://en.wikipedia.org/wiki/Kalman_filter.

[Lab] Roger Labbe. Kalman-and-bayesian-filters-in-python. https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.