



## Self-driving car: Tracking cars with Extended and Unscented Kalman Filter



Jonathan Hui

Apr 9, 2018 · 4 min read

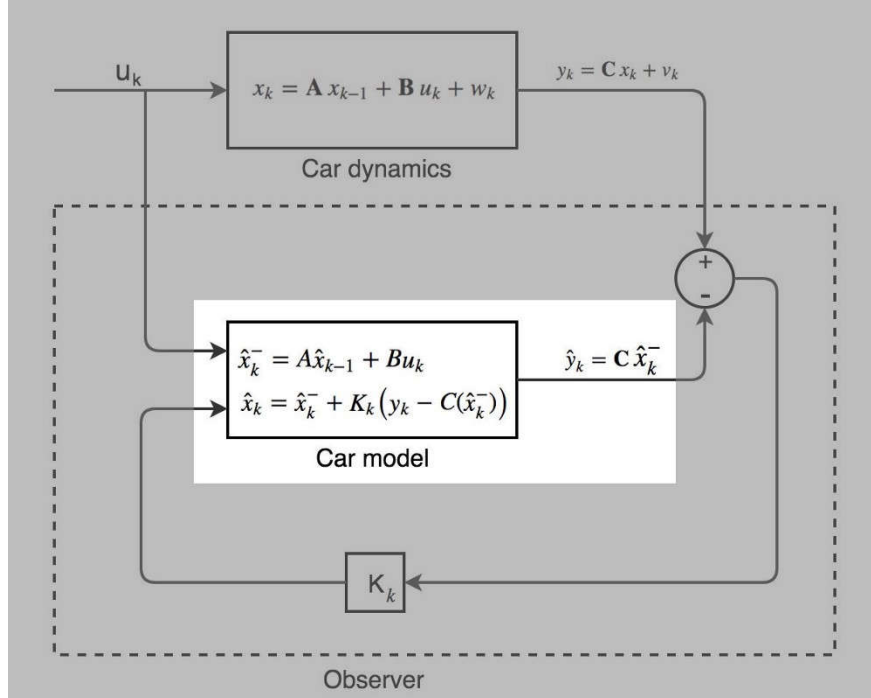
In the [previous article](#), we discuss how Kalman Filter works. The mathematical solution for Kalman Filter assumes states are Gaussian Distributed. However, this is not true sometimes. In this article, we extend our method to Extended Kalman Filter and the Unscented Kalman Filter which produce more accurate results than the Kalman Filter if the dynamic model is not linear.

This is part of a 5-series self-driving. Other articles includes

- Self-driving perception: [Sensor fusion with Kalman Filter](#).
- Self-driving perception: [Extended Kalman Filter and Unscented Kalman Filter](#).
- Self-driving localization: [Localization with Particle Filter](#).
- Self-driving control: [Control with Model Predictive Control & PID](#).
- Self-driving [Path finding](#).

### Extended Kalman Filter

Recall from the Kalman Filter, this is how the observer world looks like:



This assume our dynamic model is linear.

$$x_k = A x_{k-1} + B u_k + w_k$$

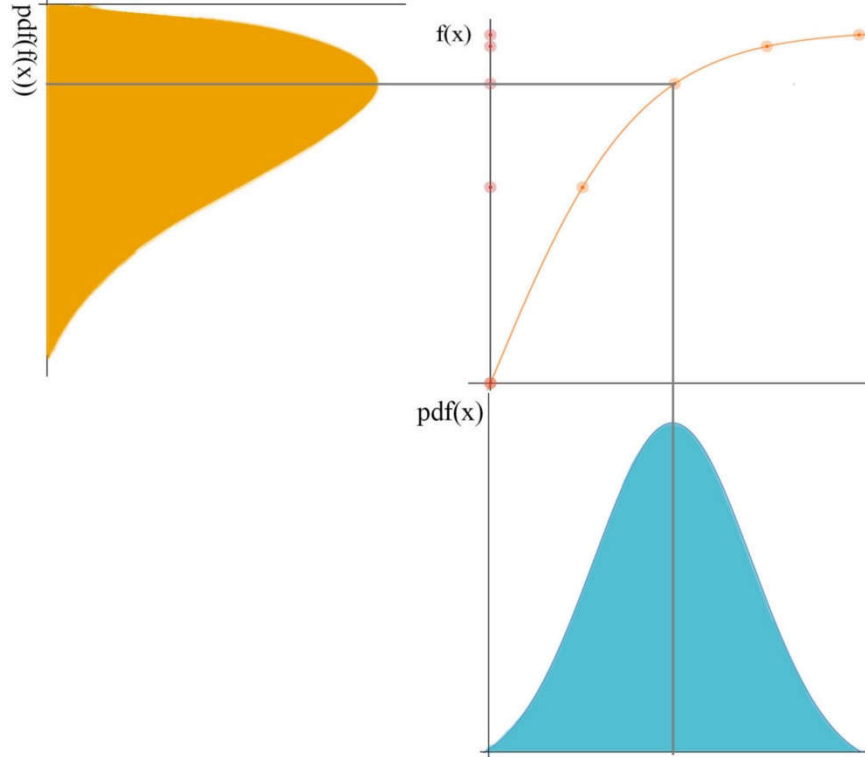
$$y_k = C x_k + v_k$$

where A, B and C are matrix. If the initial state is Gaussian distributed, the prediction is also Gaussian distributed. So Kalman Filter will work nicely. However, many dynamic models are non-linear. The model will be written as:

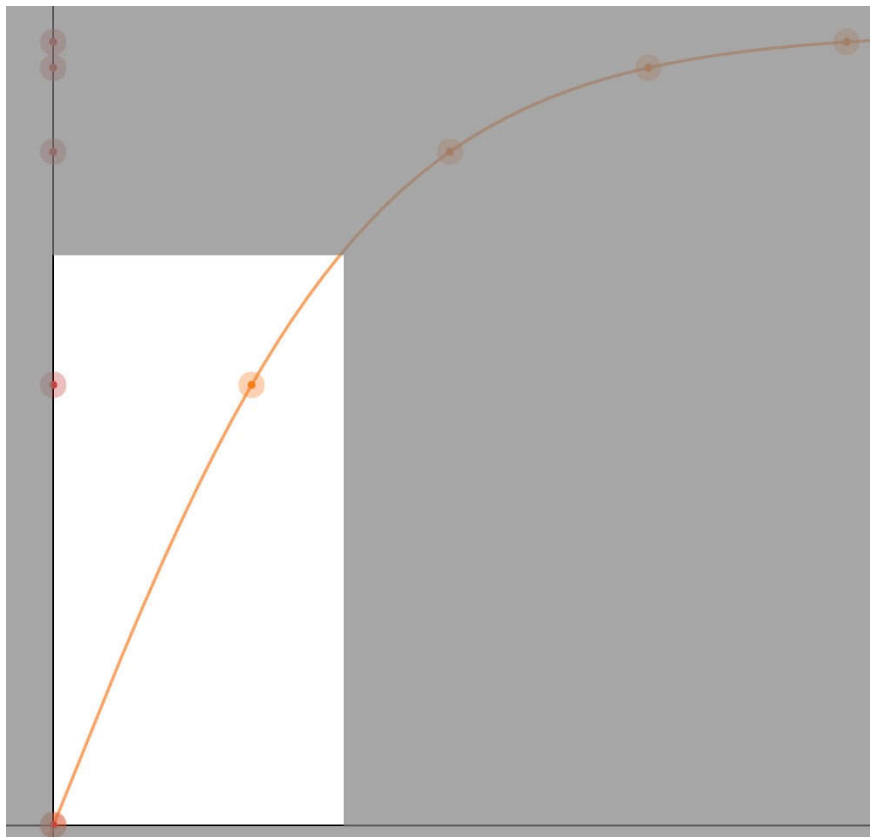
$$x_k = f(x_{k-1}, u_k) + w_k$$

$$y_k = h(x_k) + v_k$$

If  $f$  or  $h$  is not linear, will the output Gaussian distributed? In the bottom right below, the probability distribution function for our state  $x$  is Gaussian distributed. We apply a function  $f$  on  $x$ , and the plot on the top left is the probability distribution function for  $f(x)$ . As you can see, it is not Gaussian distributed anymore. Hence, if the function  $f$  is not linear, our output will not be Gaussian distributed.



However, if we limit the range of  $x$ , the function  $f$  can be approximated by a linear function.



If the function  $f$  is close to a linear function or  $f$  is close to linear for our target range of  $x$ , we can use Extended Kalman Filter to replace Kalman Filter.

In Extended Kalman Filter, we approximate the function  $f$  by Jacobian matrix.

$$f : \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Jacobian matrix

In the left below is the original equation for the Kalman Filter and the right is the Extended Kalman Filter. The difference is we replace A by the Jacobian matrix  $F$ , and C by the Jacobian matrix  $H$ .

Prediction:

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}u_k$$

$$P_k^- = \mathbf{A}P_{k-1}\mathbf{A}^T + Q$$

Update:

$$K_k = \frac{P_k^- \mathbf{C}^T}{\mathbf{C}P_k^- \mathbf{C}^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - \mathbf{C}\hat{x}_k^-)$$

$$P_k = (I - K_k \mathbf{C})P_k^-$$

Prediction:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k)$$

$$P_k^- = \mathbf{F}P_{k-1}\mathbf{F}^T + Q$$

$$\mathbf{F} = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1}, u_k}$$

Update:

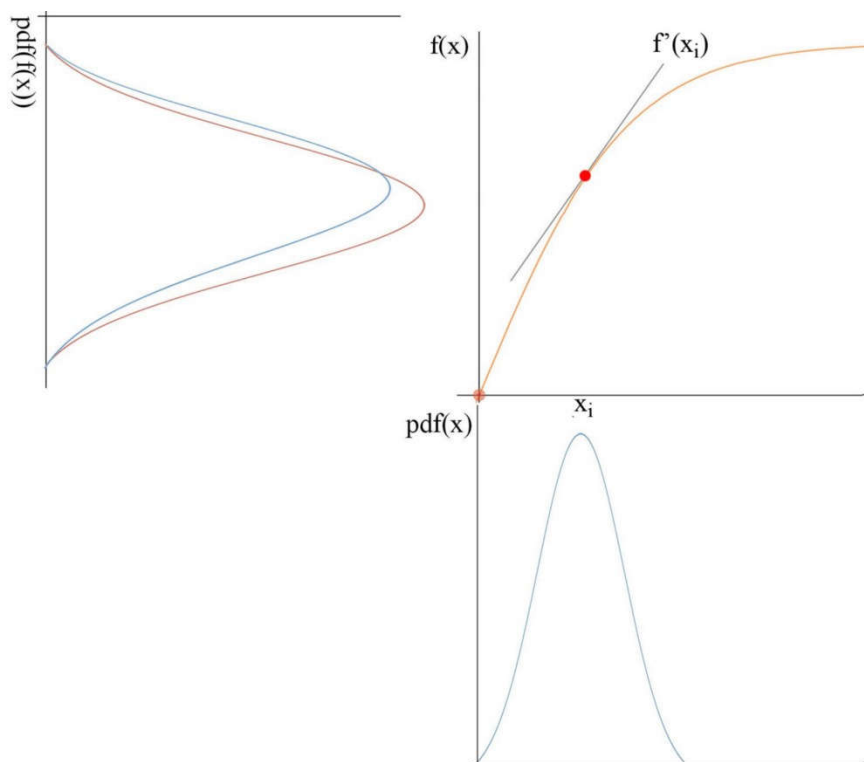
$$K_k = \frac{P_k^- H^T}{\mathbf{H}P_k^- \mathbf{H}^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-))$$

$$P_k = (I - K_k \mathbf{H})P_k^-$$

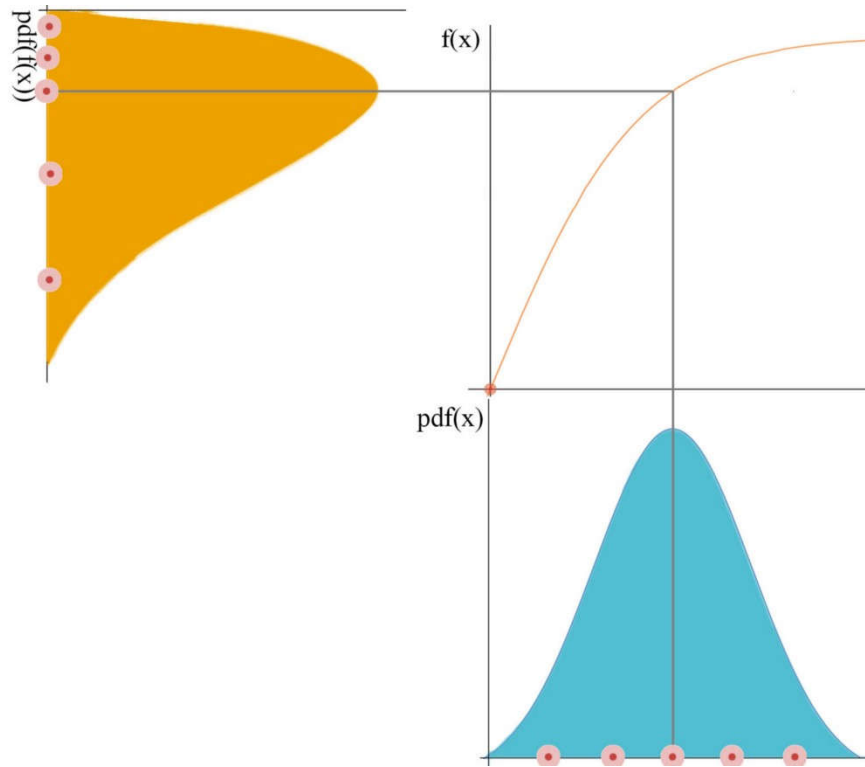
$$\mathbf{H} = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_k^-}$$

So instead of using  $f(x)$ , we replace it with  $f'(x_i)$ . Here is our visualization which the orange curve (computed from  $f'(x_i)$ ) in the top left is our new Gaussian Distribution which approximate the blue one (computed from  $f(x)$ ).



# Unscented Kalman Filter

Extended Kalman Filter handles cases where  $f$  is close to linear which we will use  $f'(x_i)$  to approximate  $f(x)$ . However, this is not feasible if  $f$  is not close to linear. In this case, we can sample values in  $x$  and compute  $f(x)$ . Then we use the sampled outputs to compute a Gaussian distribution which is used to approximate the probability distribution of  $f(x)$ . Since  $x$  is Gaussian distributed, instead of randomly sample  $x$ , we can just compute the output of predefined sigma points (on the bottom right).



Then we assign a weight to the output which is proportional to the probability distribution of  $x$ . We then compute the Gaussian Distribution based on the weighted output. This becomes our state prediction.

