

Extended Kalman Filter: Why do we need an Extended Version?



Harveen Singh

Apr 7, 2018 · 8 min read

This post is in continuation to my last post on Kalman Filter. So my colleague Larry is pretty excited to know about the working of Kalman Filter, but can he understand the concept of Non Linearity and Extended Kalman Filter. Let's find out.

. . .

Larry: I know Kalman Filters, I can now predict and update, I basically know an important prediction tool now.

Me: Can you tell me what were the assumptions we made while reading about Kalman Filter?

Larry: Assump... ? What do you mean? You only said that the Kalman Filter works with a Gaussian Only. Thats it. Isn't it?

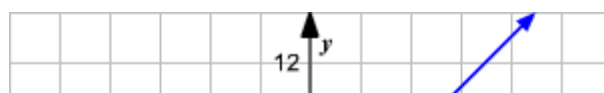
Me: Well, that is half right. The other important assumption that was hidden in the last article were Linear functions. So the two assumptions are:-

1. Kalman Filter will always work with **Gaussian Distribution**.
2. Kalman Filter will always work with **Linear Functions**.

Larry: Oh god! Now from where did Linear Functions come into picture?

Me: With Linear functions, I mean that the prediction and update step both will contain Linear Functions only. They were already there if you closely watch all the equations.

A linear function somewhat looks like this:



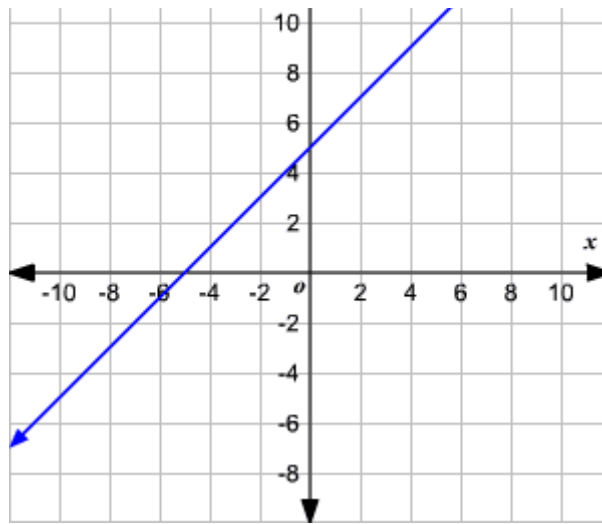


Figure 1. Linear Function. (Source)

On the other hand a non linear functions looks like this:

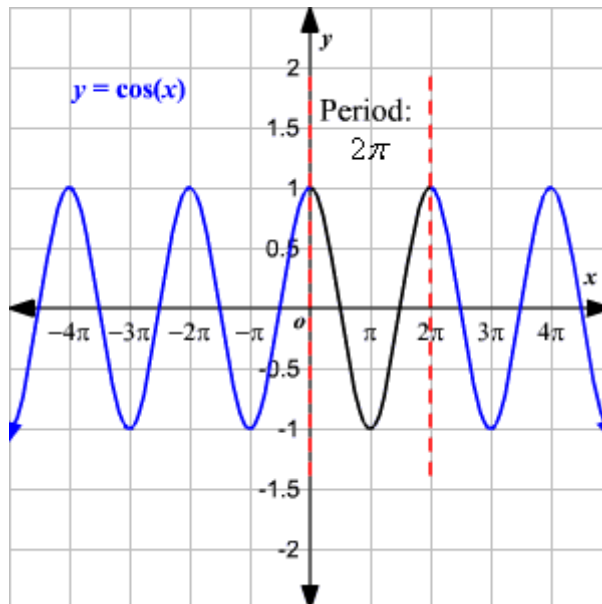


Figure 2. Non Linear Function. (Source)

So observing from these figures, an equation of a straight line is a linear function while a cos function is a non linear function.

Larry: Yup, that is ok. We do not have any angles in our equations, so they appear to be linear only. Then what is the problem now with KF?

Me: Most real world problems involve non linear functions. In most cases, the system is looking into some direction and taking measurement in another direction. This involves angles and sine, cosine functions which are non linear functions which then lead to problems.

Larry: Hmm, but still how does non linear function create problems?

Me: If you feed a Gaussian with a linear function then the output is also a Gaussian

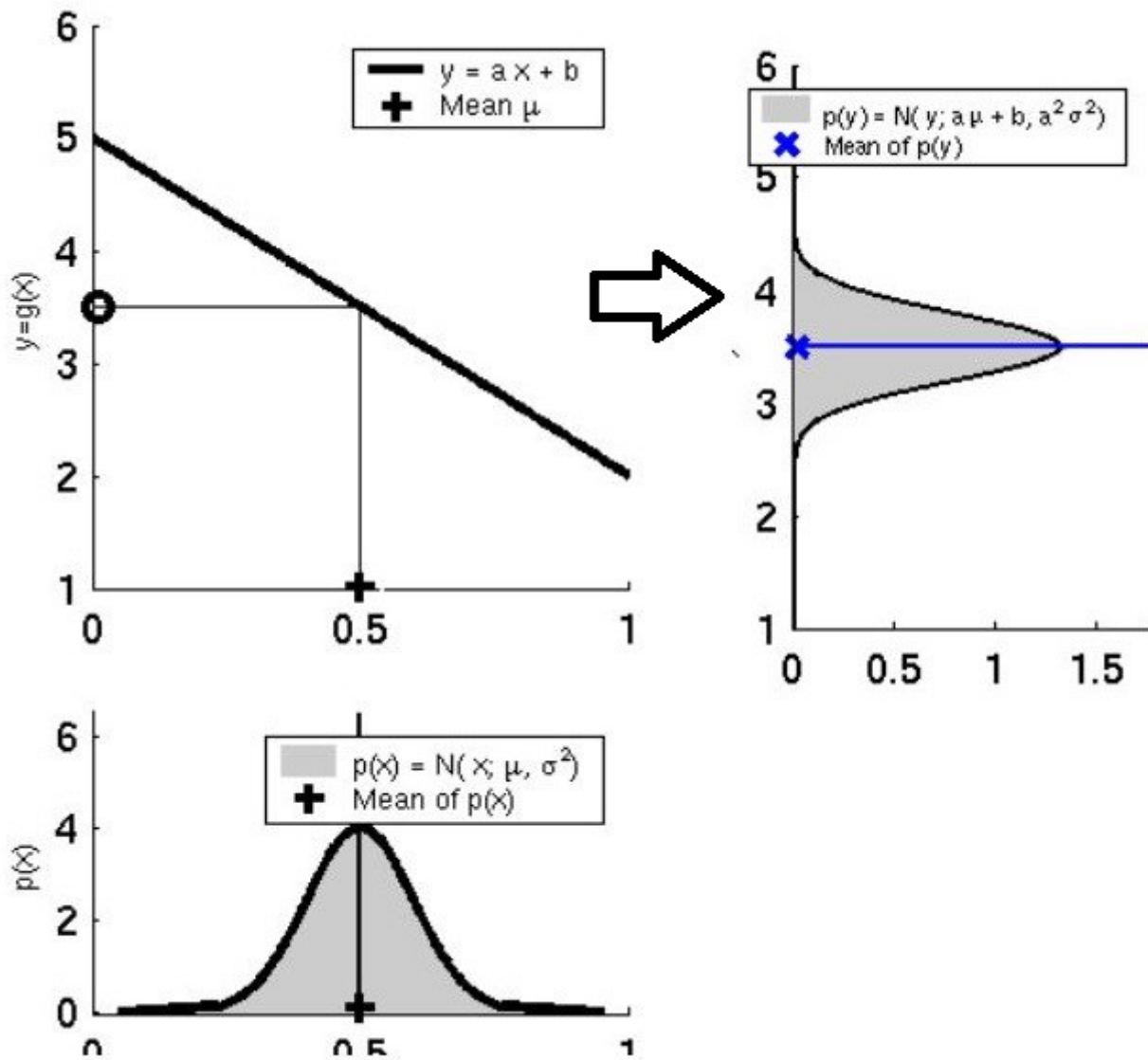
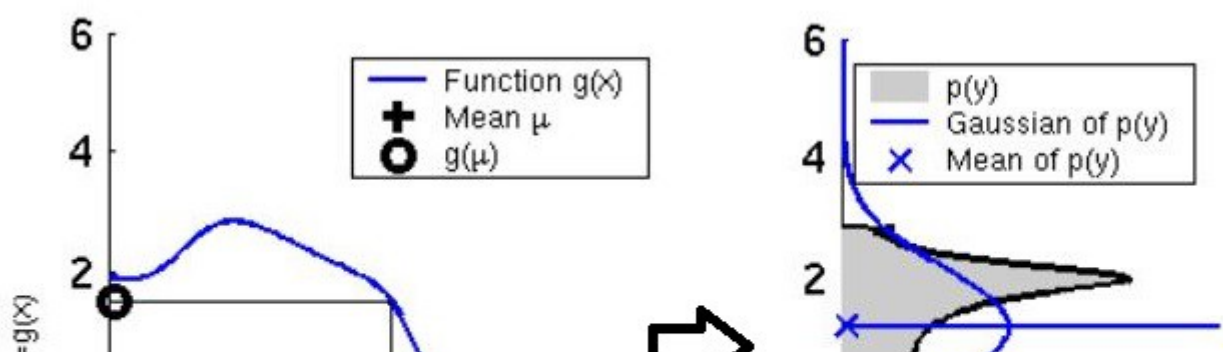


Figure 3. Gaussian + Linear Function = Gaussian (Source)

If you feed a Gaussian with a Non linear function then the output is not a Gaussian. Non Linear functions lead to Non Gaussian Distributions.



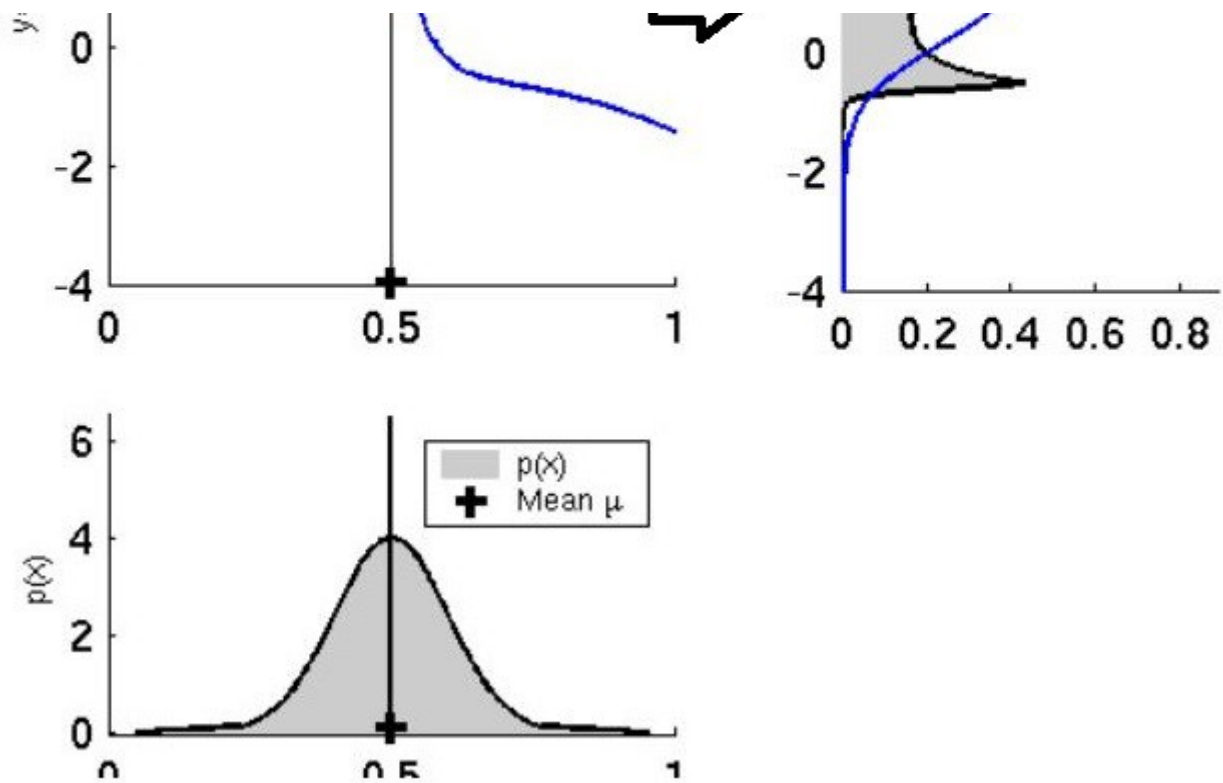


Figure 4. Gaussian + Non Linear Function = Non Gaussian (Source)

So if we apply a non linear function it will not end up as a Gaussian Distribution on which we can't apply Kalman Filter anymore. Non linearity destroys the Gaussian and it does not makes sense to compute the mean and variances.

Larry: Oh no, in that case our Kalman Filter is now broken. So what is the solution?

Me: What can the most trivial solution you can think of?

Larry: Me? Umm. I would say use Linear functions only :D

Me: Even it does not makes sense, but this is exactly the solution.

Larry: What? You mean I was right?? How come?

Me: Yup, you were right. We will work with Linear functions only.

Larry: But what about that cos, sin functions you were talking about? They are still Non Linear, right?

Me: Absolutely. They are non linear but we will make them Linear by approximation. Here, we will take help of a powerful tool called Taylor Series, which will help us to get a Linear Approximation of the Non Linear function. After applying the approximation what we get is an Extended Kalman Filter.

Larry: New tools keep on coming! How does this Taylor works?

Me: We take a point and perform a bunch of derivatives on that point. In case of an

EKF, we take mean of the Gaussian on the Non Linear Curve and perform a number of derivatives to approximate it.

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

Figure 5. Taylor Series

Suppose we want to approximate $\sin(x)$ at $x=0$.

Lets assume that we want to find a polynomial function $P(x) = c_0 + c_1 * x + c_2 * x^2 + c_3 * x^3$ to approximate $\sin(x)$. So we need to find out the values for c_0 , c_1 , c_2 and c_3

At $x=0$, $\sin(x) = 0$, $P(x) = c_0 + 0 + 0$

If our approximation has to be even a little bit near to $\sin(x)$, then the value of $\sin(x)$ must be equal to value of $P(x)$ at $x=0$. So, $c_0 = 0$

It will also be good if our approximation has the same tangent slope as that of $\sin(x)$ at $x=0$

At $x=0$, $d \sin(x)/dx = \cos(x) \cos(0) = 1$

$d P(x)/dx = c_1 + 2*c_2*x + 3*c_3*x^2 = c_1 + 0 + 0$

If our approximation has to be precise then the value of derivative of $P(x)$ must be equal to derivative of x at $x=0$. So $c_1 = 1$

Going on.. we can find that the approximation of $\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! \dots$

Larry: Cool! That was really cool. But this again will give a curve right which is non linear, aren't we just interested in Linearizing?

Me: Exactly, we are interested in linearizing, so we are just interested in the first derivative of Taylor series. For every non linear function, we just draw a tangent around the mean and try to approximate the function linearly.

Larry: Hmm. Ok. KF worked only on linear functions, but in real life we have non linear functions which destroy our Gaussians, so we try to approximate those functions linearly by Taylor Series and this comes under Extended Kalman Filter. Right?

Me: Absolutely. Spot on! Check this out!

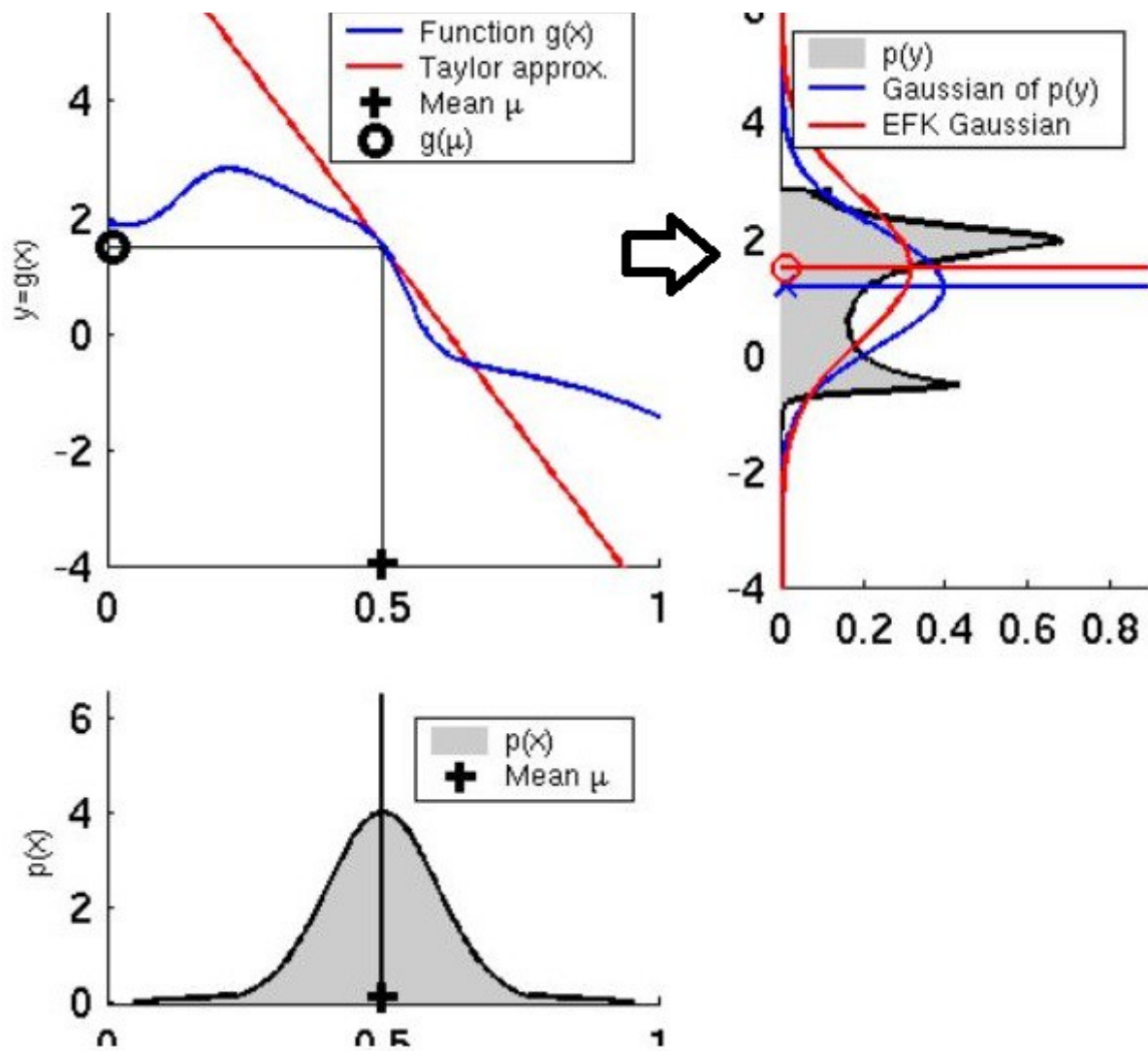


Figure 6. Scenario after applying Taylor's Approximation to Linearize our function

Larry: I want to know that how does that effect our equations that we wrote for Kalman Filter but before that what all are the sensors that provide data?

Me: Suppose we have two sensors LIDAR and RADAR. A LIDAR provides us the distance in the form of Cartesian coordinate system. On the other hand, a RADAR provides us the distance and velocity in Polar coordinate system.

Lidar => {px, py}

Radar => { ρ , Φ , ρ_{dot} }

px, py -> Coordinates of object in Cartesian System

ρ -> is the distance to the object

Φ -> is the counter clockwise angle between ρ and x- axis

ρ_{dot} -> is the change of ρ

The x-axis is always in the direction where the car is heading.

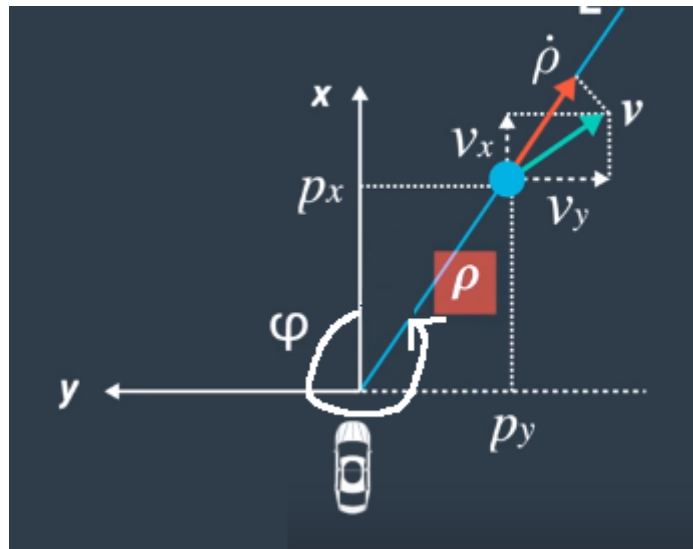


Figure 7. Polar Coordinates as reported by Radar. (Source)

Larry: Can we take data from both the sensors?

Me: Of course, you have to take. Taking data from different sensors and combining them together is called Sensor Fusion.

Larry: Ok, as far as I can guess, the measurements coming from RADAR are non linear as they involve angles. Now I am interested to know the equations of Extended Kalman Filter!

Me: Right! Sure.

Prediction Step

$$\mathbf{x}' = \mathbf{F} \cdot \mathbf{x} + \mathbf{B} \cdot \boldsymbol{\mu} + \mathbf{v}$$

$$\mathbf{P}' = \mathbf{F} \mathbf{P} \mathbf{F}^T + \mathbf{Q}$$

The prediction step is exactly the same as that of Kalman Filter. It does not matter whether the data is coming from LIDAR or RADAR the prediction step is exactly the same.

Update Step (Only in case of EKF i.e. Non Linear Measurements coming from RADAR)

Equation 1:

$$\mathbf{y} = \mathbf{z} - \mathbf{h}(\mathbf{x}')$$

\mathbf{z} -> actual measurement in polar coordinates

\mathbf{h} -> function that specifies how our speed and position are mapped to polar coordinates

\mathbf{x}' -> Predicted Value

\mathbf{y} -> Difference between Measured Value and Actual Value

$h(x')$

This is a function that specifies the mapping between our predicted values in Cartesian coordinates and Polar coordinates. **This mapping is required because we are predicting in Cartesian coordinates but our measurement (z) that is coming from the sensor is in Polar Coordinates.**

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

Figure 8. Mapping between Cartesian and Polar coordinates (Source)

Equation 2:

$$S = H_j P' H_j^T + R$$

$$K = P' H_j^T S^{-1}$$

R -> Measurement Noise

K -> Kalman Gain

S -> Total Error

S^{-1} -> The inverse of S

H_j -> The Jacobian Matrix

H_j

H_j is the Jacobian Matrix. The Jacobian matrix is the first order derivative that we just discussed in Taylor Series. Since here we are dealing with matrices, we need to find differential in the form of a matrix.

$$J_{kl} = dF_k / dX_l$$

J_{kl} is the k, l element of the Jacobian matrix, F_k is the k th element of the vector function F , and X_l is the l th element of the vector variable X .

Here $F_k = \{\rho, \Phi, \rho_{\dot{}}\}$

$X_l = \{p_x, p_y, v_x, v_y\}$

Since in case of RADAR we have 4 measurements, 2 for distance and 2 for velocity.

$$H_j = \begin{bmatrix} \frac{\partial \rho}{\partial p_x} & \frac{\partial \rho}{\partial p_y} & \frac{\partial \rho}{\partial v_x} & \frac{\partial \rho}{\partial v_y} \\ \frac{\partial \varphi}{\partial p_x} & \frac{\partial \varphi}{\partial p_y} & \frac{\partial \varphi}{\partial v_x} & \frac{\partial \varphi}{\partial v_y} \\ \frac{\partial \dot{\rho}}{\partial p_x} & \frac{\partial \dot{\rho}}{\partial p_y} & \frac{\partial \dot{\rho}}{\partial v_x} & \frac{\partial \dot{\rho}}{\partial v_y} \end{bmatrix}$$

Figure 9. Jacobian matrix (Source)

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

Figure 10. Jacobian Matrix after applying derivatives

Equation 3:

$$x = x' + K.y$$

$$P = (I - KH_j)P'$$

Larry: Oh got it! So in case of a LIDAR we will apply a Kalman Filter because the measurements from the sensor are Linear. But in case of a Radar we need to apply Extended Kalman Filter because it includes angles that are non linear, hence we do an approximation of the non linear function using first derivative of Taylor series called Jacobian Matrix (H_j). Then we convert our cartesian space to polar space using $h(x')$ and finally we replace H with H_j in all further equations of a KF.

Me: 10/10. Thank You!

• • •

Surely the Jacobian Matrix is a bit of Magic because it transforms a non linear space into a linear space.

But believe me it is no magic, it is all Maths. Please do connect with me in case you find any errors. You can find me on LinkedIn here.

You can read the basics of Kalman Filter here.
Unscented Kalman Filter here.

[Self Driving Cars](#)[Kalman Filter](#)[Udacity](#)[Robotics](#)[Artificial Intelligence](#)[About](#) [Help](#) [Legal](#)