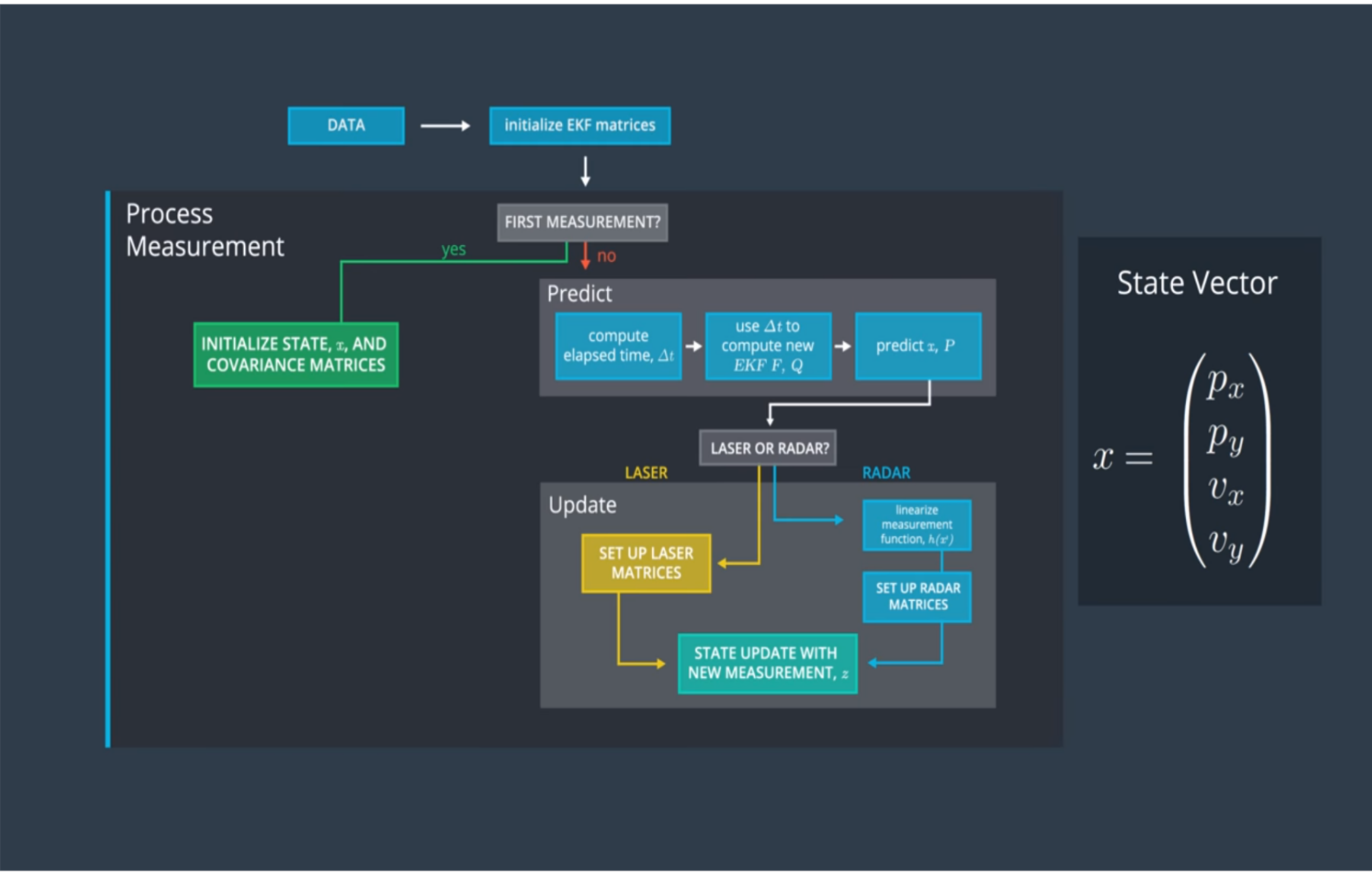
SENSOR FUSION SCHEMA (Object tracking)



5 Kalman Filter Equations In C++

The state transition function is

deterministic part

$$x' = f(x) + \nu = Fx + \underbrace{Bu}_{=0} + \nu \tag{1}$$

stochastic part

The motion control, Bu, is zero in our case, so our state transition function is simplified to:

We cannot know the control input of an uncooperative object

$$\boxed{x' = Fx + \nu} \tag{2}$$

The measurement function is

$$\boxed{z = h(x') + \omega = Hx' + \omega} \tag{3}$$

where,
ν is the process noise. It is a Gaussian with zero mean and covariance matrix Q:

$$\nu \sim N(0, Q) \tag{4}$$

ω is the measurement noise. It is a Gaussian with zero mean and covariance matrix R:

$$\omega \sim N(0, R) \tag{5}$$

The state vector:

$$x = \begin{pmatrix} p \\ v \end{pmatrix} \tag{6}$$

Linear motion:

$$p' = p + v\Delta t \tag{7}$$

$$v' = v \tag{8}$$

$$\rightarrow \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

The state prediction function:

$$\begin{pmatrix} p' \\ v' \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix} \tag{9}$$

The measurement function:

$$z = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} p' \\ v' \end{pmatrix} \tag{10}$$

5.1 Kalman Filter Algorithm

Prediction:

$$x' = Fx + u \tag{11}$$

$$P' = FPF^T + Q \tag{12}$$

project the previous covariance P into the current state space

Measurement Update:

$$y = z - Hx' \tag{13}$$

$$S = HP'H^T + R \tag{14}$$

Project the estimated covariance P' into the measurement space of the sensor.

$$K = P'H^TS^{-1} \tag{15}$$

It combines the uncertainty of where we think we are P' with the uncertainty of our sensor measurement R.

$$x = x' + Ky \tag{16}$$

$$P = (I - KH)P' \tag{17}$$

6 Kalman Filter Equation in C++ Programming Assignment

$$x = \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \end{pmatrix} \tag{18}$$

7 Measurement Update Quiz

No equations.

8 State Prediction - 4D Case

No equations.

9 Uncertainty Depending on Time Delay Quiz

No equations.

10 Uncertainty Depending on Acceleration Quiz

No equations.

11 Process Covariance Matrix

Linear Motion Model - 2D:

$$x' = Fx + \nu \tag{19}$$

$$\begin{cases} p'_x = p_x + v_x \Delta t + \nu_{px} \\ p'_y = p_y + v_y \Delta t + \nu_{py} \\ v'_x = v_x + \nu_{vx} \\ v'_y = v_y + \nu_{vy} \end{cases} \tag{20}$$

How could we determine these values?

Can be expressed in a matrix form as:

We can exploit the formulas

$$\begin{pmatrix} p'_x \\ p'_y \\ v'_x \\ v'_y \end{pmatrix} = \overbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{F} \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \end{pmatrix} + \begin{pmatrix} \nu_{px} \\ \nu_{py} \\ \nu_{vx} \\ \nu_{vy} \end{pmatrix} \tag{21}$$

11.1 Deriving the 2D velocity with a constant acceleration

Considering that we cannot measure a, we can consider it as a noise ν.

From the kinematic formulas we can define the acceleration as the change in the velocity over the elapsed time, so the current velocity minus previous velocity over the elapsed time.

$$a = \frac{\Delta v}{\Delta t} = \frac{v_{k+1} - v_k}{\Delta t} \tag{22}$$

Then we can derive the current velocity as the previous velocity plus the acceleration a times Δt.

$$v_{k+1} = v_k + a\Delta t \tag{23}$$

Generalized in 2D we have both speed components described as follows:

$$\begin{cases} v'_x = v_x + a_x \Delta t \\ v'_y = v_y + a_y \Delta t \end{cases} \tag{24}$$

11.2 Deriving the relation between the previous location and current position in 2D when having a motion with constant acceleration

Similarly we know that the displacement Δp equals the average velocity v_{avg} times Δt.

$$\Delta p = p_{k+1} - p_k = v_{avg} \Delta t \tag{25}$$

which is the previous velocity v_{k+1} plus current velocity v_k over two.

$$p_{k+1} - p_k = \frac{v_{k+1} + v_k}{2} \Delta t \tag{26}$$

$$p_{k+1} = p_k + \frac{v_{k+1} + v_k}{2} \Delta t \tag{27}$$

And as we already saw, the current velocity v_{k+1} is our old velocity v_k plus the acceleration a times Δt.

$$v_{k+1} = v_k + a\Delta t \tag{28}$$

so we can replace the v_{k+1} with $v_k + a\Delta t$ in previous equations:

$$p_{k+1} = p_k + \frac{v_k + a\Delta t + v_k}{2} \Delta t \tag{29}$$

$$p_{k+1} = p_k + \frac{2v_k \Delta t}{2} + \frac{a\Delta t^2}{2} \tag{30}$$

And finally we get the full relation between the previous location p_k and current location p_{k+1}:

$$p_{k+1} = p_k + v_k \Delta t + \frac{a\Delta t^2}{2} \tag{31}$$

Going in 2D, we'll have both horizontal and vertical directions:

$$\begin{cases} p'_x = p_x + v_x \Delta t + \frac{a_x \Delta t^2}{2} \\ p'_y = p_y + v_y \Delta t + \frac{a_y \Delta t^2}{2} \end{cases} \tag{32}$$

11.3 The 2D motion model with constant acceleration

Combining both 2D position and 2D velocity equations previously deducted formulas we have:

$$\boxed{\begin{cases} p'_x = p_x + v_x\Delta t + \frac{a_x\Delta t^2}{2} \\ p'_y = p_y + v_y\Delta t + \frac{a_y\Delta t^2}{2} \\ v'_x = v_x + a_x\Delta t \\ v'_y = v_y + a_y\Delta t \end{cases}} \tag{33}$$

this motivate the calculation

11.4 Process Noise in 2D

Since the acceleration is unknown we can add it to the noise component, and this random noise would be expressed analytically as the last terms in the equation derived above. So, we have a random acceleration vector ν in this form:

$$\boxed{\nu = \begin{pmatrix} \nu_{px} \\ \nu_{py} \\ \nu_{vx} \\ \nu_{vy} \end{pmatrix} = \begin{pmatrix} \frac{a_x\Delta t^2}{2} \\ \frac{a_y\Delta t^2}{2} \\ a_x\Delta t \\ a_y\Delta t \end{pmatrix}} \tag{34}$$

which is described by a zero mean and a covariance matrix Q.

$$\nu \sim N(0, Q) \tag{35}$$

NOTE: we need to compute Q

The vector ν can be decomposed into two components a 4 by 2 matrix G which does not contain random variables and a 2 by 1 matrix a which contains the random acceleration components:

$$\nu = \begin{pmatrix} \frac{a_x\Delta t^2}{2} \\ \frac{a_y\Delta t^2}{2} \\ a_x\Delta t \\ a_y\Delta t \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} a_x \\ a_y \end{pmatrix}}_{a} = Ga \tag{36}$$

Δt is computed at each Kalman Filter step and the acceleration is a random vector with zero mean and standard deviations σ_{ax} and σ_{ay}.

Based on our noise vector we can define now the new covariance matrix Q. The covariance matrix is defined as the expectation value of the noise vector ν times the noise vector ν transpose. So let's write this down:

$$Q = E[\nu\nu^T] = E[Gaa^TG^T] \tag{37}$$

As G does not contain random variables, we can put it outside the expectation calculation.

$$Q = GE[aa^T]G^T = G\underbrace{\begin{pmatrix} \sigma_{ax}^2 & \sigma_{axy} \\ \sigma_{axy} & \sigma_{ay}^2 \end{pmatrix}}_{Q_\nu} G^T = GQ_\nu G^T \tag{38}$$

This leaves us with three statistical moments:

- the expectation of ax times ax, which is the variance of ax: σ_{ax}^2.
- the expectation of ay times ay, which is the variance of ay: σ_{ay}^2 squared.
- And the expectation of ax times ay, which is the covariance of ax and ay: σ_{axy}.

a_x and a_y are assumed uncorrelated noise processes. This means that the covariance σ_{axy} in Q_ν is zero:

$$Q_\nu = \begin{pmatrix} \sigma_{ax}^2 & \sigma_{axy} \\ \sigma_{axy} & \sigma_{ay}^2 \end{pmatrix} = \begin{pmatrix} \sigma_{ax}^2 & 0 \\ 0 & \sigma_{ay}^2 \end{pmatrix} \tag{39}$$

So after combining everything in one matrix we obtain our 4 by 4 Q matrix:

Complete process covariance

$$Q = GQ_\nu G^T = \begin{pmatrix} \frac{\Delta t^4}{4}\sigma_{ax}^2 & 0 & \frac{\Delta t^3}{2}\sigma_{ax}^2 & 0 \\ 0 & \frac{\Delta t^4}{4}\sigma_{ay}^2 & 0 & \frac{\Delta t^3}{2}\sigma_{ay}^2 \\ \frac{\Delta t^3}{2}\sigma_{ax}^2 & 0 & \Delta t^2\sigma_{ax}^2 & 0 \\ 0 & \frac{\Delta t^3}{2}\sigma_{ay}^2 & 0 & \Delta t^2\sigma_{ay}^2 \end{pmatrix} \tag{40}$$

Cov. matrix of the individual noise processes matrix

11.5 other equations

Note: Some authors describe Q as the complete process noise covariance matrix. And some authors describe Q as the covariance matrix of the individual noise processes. In our case, the covariance matrix of the individual noise processes matrix is called Q_ν. So we should be aware of that.

12 Process Covariance Matrix Text

No equations.

13 Laser Measurements

No equations.

14 Find Out the H Matrix

14.1 Quiz question

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = (?) \begin{pmatrix} p'_x \\ p'_y \\ v'_x \\ v'_y \end{pmatrix} \tag{41}$$

14.2 Quiz answer

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p'_x \\ p'_y \\ v'_x \\ v'_y \end{pmatrix} \tag{42}$$

15 Find out the Dimension of R

15.1 Quiz question

Now, can you determine what is the dimensionality of the measurement noise covariance matrix R.

15.2 Quiz answer

Generally, provided by the sensor manufacturer

$$R = E[\omega\omega^T] = \begin{pmatrix} \sigma^2_{px} & 0 \\ 0 & \sigma^2_{py} \end{pmatrix} \tag{43}$$

16 Programming Assignment

$$R = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix} \tag{44}$$

$$R = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \tag{45}$$

16.1 Other Equations

$$a_x \sim N(0, \sigma^2_{ax}) \tag{46}$$

$$a_y \sim N(0, \sigma^2_{ay}) \tag{47}$$

17 Radar Measurements

The state transition function:

$$x' = f(x) + \nu \tag{48}$$

The measurement function:

$$z = h(x') + \omega \tag{49}$$

The state transition function will be exactly what you've designed in the lidar case.

However our new Radar sees the world differently. Instead of a 2D pose (p_x, p_y), the radar can directly measure the object range ρ, bearing φ and range rate $\dot{\rho}$:

$$\boxed{z = \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix}} \tag{50}$$

- The range ρ is the radial distance from the origin to our pedestrian. So we can always define a ray which extends from the origin to our object position.
- The bearing φ is the angle between the ray and x direction.
- And the range rate $\dot{\rho}$, also known as Doppler or radial velocity is the velocity along this ray.

And similar to our motion model, the radar observations are corrupted by a zero-mean random noise $\omega \sim N(0, R)$. Considering that the three measurement components of the measurement vector z are not cross-correlated, the radar measurement covariance matrix R becomes a 3 by 3 diagonal matrix:

$$\boxed{R = \begin{pmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & \sigma_{\dot{\rho}}^2 \end{pmatrix}} \tag{51}$$

And the next question is what is the measurement function $h(x')$ that maps the predicted state x' into the measurement space?

$$\begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} \xleftarrow{h(x')} \begin{pmatrix} p'_x \\ p'_y \\ v'_x \\ v'_y \end{pmatrix} \tag{52}$$

And that's how I defined the h function which basically specifies how the predicted position and speed can be related to the observed range ρ, bearing φ and range rate $\dot{\rho}$:

$$h(x') = \begin{pmatrix} \sqrt{p_x'^2 + p_y'^2} \\ \arctan(p'_y/p'_x) \\ \frac{p'_x v'_x + p'_y v'_y}{\sqrt{p_x'^2 + p_y'^2}} \end{pmatrix} \tag{53}$$

Non linear equation that maps the measurements into the state space

17.1 Deriving the Radar Measurement Function

The measurement function is composed of three components which show how the predicted state $x' = (p'_x, p'_y, v'_x, v'_y)^T$ is mapped into the measurement space $z = (\rho, \varphi, \dot{\rho})^T$:

1. The range ρ is the distance to the pedestrian which can be defined as:

$$\rho = \sqrt{p_x^2 + p_y^2} \tag{54}$$

2. φ is the angle between ρ and x direction and can be defined as:

$$\varphi = \arctan(p_y/p_x) \tag{55}$$

3. Range rate $\dot{\rho(t)}$ derivation:

- Method 1: Generally we can explicitly describe the range ρ as a function of time:

$$\rho(t) = \sqrt{p_x(t)^2 + p_y(t)^2} \tag{56}$$

The range rate $\dot{\rho(t)}$ is defined as time rate of change of the range ρ, and it can be described as the time derivative of ρ:

$$\dot{\rho} = \frac{\partial \rho(t)}{\partial t} = \frac{\partial}{\partial t}\sqrt{p_x(t)^2 + p_y(t)^2} = \frac{1}{2\sqrt{p_x(t)^2 + p_y(t)^2}}(\frac{\partial}{\partial t}p_x(t)^2 + \frac{\partial}{\partial t}p_y(t)^2) =$$
$$\frac{1}{2\sqrt{p_x(t)^2 + p_y(t)^2}}(2p_x(t)\frac{\partial}{\partial t}p_x(t) + 2p_y(t)\frac{\partial}{\partial t}p_y(t))$$

$\frac{\partial}{\partial t}p_x(t)$ is nothing else than $v_x(t)$, similarly $\frac{\partial}{\partial t}p_y(t)$ is $v_y(t)$. So we have:

$$\dot{\rho} = \frac{\partial \rho(t)}{\partial t} = \frac{1}{2\sqrt{p_x(t)^2 + p_y(t)^2}}(2p_x(t)v_x(t) + 2p_y(t)v_y(t)) = \frac{2(p_x(t)v_x(t) + p_y(t)v_y(t))}{2\sqrt{p_x(t)^2 + p_y(t)^2}} =$$
$$\frac{p_x(t)v_x(t) + p_y(t)v_y(t)}{\sqrt{p_x(t)^2 + p_y(t)^2}}$$

For the simplicity we just use the following notation:

$$\dot{\rho} = \frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}} \tag{57}$$

- Method 2: The range rate $\dot{\rho}$ can be seen as a scalar projection of the velocity vector $\vec{v}$ onto $\vec{\rho}$. Both $\vec{\rho}$ and $\vec{v}$ are 2D vectors defined as:

$$\vec{\rho} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}, \vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{58}$$

The scalar projection of the velocity vector $\vec{v}$ onto $\vec{\rho}$ is defined as:

$$\dot{\rho} = \frac{\vec{v}\vec{\rho}}{|\vec{\rho}|} = \frac{\begin{pmatrix} v_x \\ v_y \end{pmatrix}\begin{pmatrix} p_x \\ p_y \end{pmatrix}}{\sqrt{p_x^2 + p_y^2}} = \frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}} \tag{59}$$

where $|\vec{\rho}|$ is the length of $\vec{\rho}$. In our case it is actually the range, so $\rho = |\vec{\rho}|$

18 Mapping with a Nonlinear Function Quiz

No equations.

19 Extended Kalman Filter

No equations.

20 Linearization Example

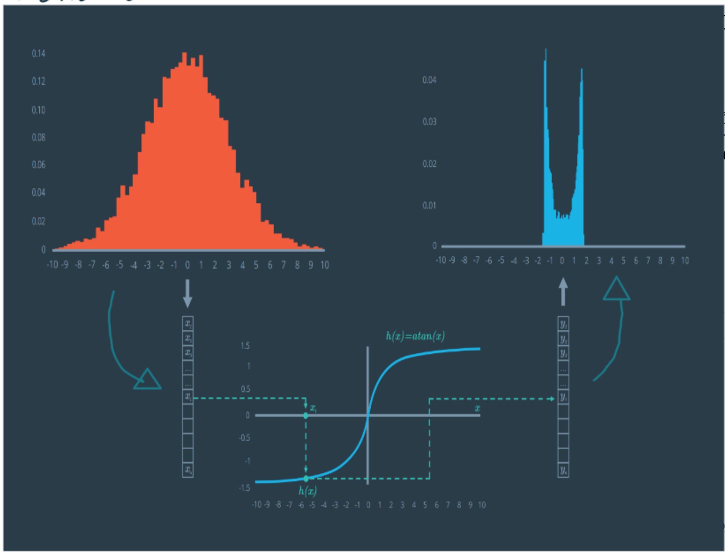
20.0.1 Linearization of a function

The Extended Kalman Filter uses a method called first order Taylor expansion

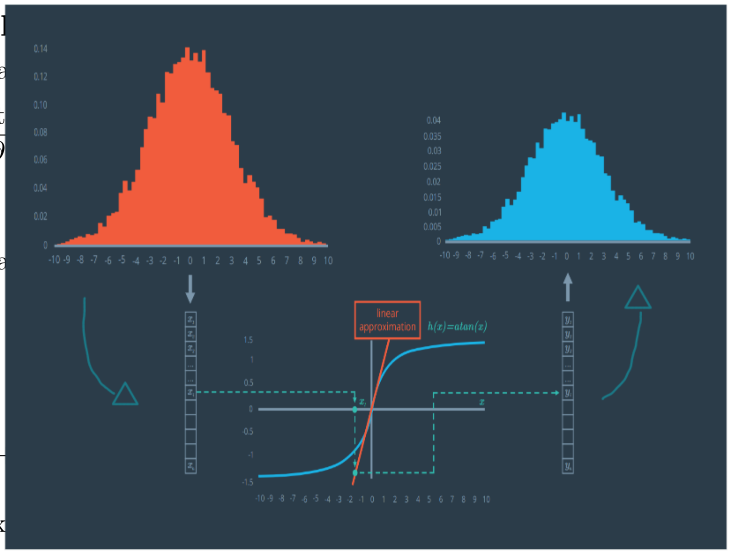
$$h(x) \approx h(\mu) + \frac{\partial h(\mu)}{\partial x}(x - \mu) \tag{60}$$

Reason to linearize.

Without linearization we lose the Gaussian distribution.



With linearization we preserve it.



$$h(x) \approx x \tag{66}$$

21 Jacobian Matrix

State transition function:

$$x' = f(x) + \nu \tag{67}$$

Measurement function:

$$z = h(x') + \omega \tag{68}$$

where we consider that $f(x)$ and $h(x)$ are nonlinear and can be linearized as:

$$f(x) \approx f(\mu) + \underbrace{\frac{\partial f(\mu)}{\partial x}}_{F_j}(x - \mu) \tag{69}$$

$$h(x) \approx h(\mu) + \underbrace{\frac{\partial h(\mu)}{\partial x}}_{H_j}(x - \mu) \tag{70}$$

The derivative of $f(x)$ and $h(x)$ with respect to x are called Jacobians:

$$\frac{\partial f(\mu)}{\partial x} = F_j \tag{71}$$

$$\frac{\partial h(\mu)}{\partial x} = H_j \tag{72}$$

and it is going to be a matrix containing all the partial derivatives:

$$H_j = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix} \tag{73}$$

To be more specific, I know that my function h describes three components: range ρ, bearing φ and range-rate $\dot{\rho}$, and my state is a vector with four components p_x, p_y, v_x, v_y. Then the Jacobian matrix H_j is going to be a matrix with 3 rows and four columns:

$$H_j = \begin{bmatrix} \frac{\partial \rho}{\partial p_x} & \frac{\partial \rho}{\partial p_y} & \frac{\partial \rho}{\partial v_x} & \frac{\partial \rho}{\partial v_y} \\ \frac{\partial \varphi}{\partial p_x} & \frac{\partial \varphi}{\partial p_y} & \frac{\partial \varphi}{\partial v_x} & \frac{\partial \varphi}{\partial v_y} \\ \frac{\partial \dot{\rho}}{\partial p_x} & \frac{\partial \dot{\rho}}{\partial p_y} & \frac{\partial \dot{\rho}}{\partial v_x} & \frac{\partial \dot{\rho}}{\partial v_y} \end{bmatrix} \tag{74}$$

After computing all these partial derivatives we have:

$$H_j = \begin{bmatrix} \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} & 0 & 0 \\ -\frac{p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 & 0 \\ \frac{p_y(v_x p_y - v_y p_x)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x(v_y p_x - v_x p_y)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} \end{bmatrix} \tag{75}$$

If you're interested to see more details about how I got this results have a look at details below.

21.0.1 Derivation of the Jacobian H_j

We're going to calculate, step by step, all the partial derivatives in H_j:

$$H_j = \begin{bmatrix} \frac{\partial \rho}{\partial p_x} & \frac{\partial \rho}{\partial p_y} & \frac{\partial \rho}{\partial v_x} & \frac{\partial \rho}{\partial v_y} \\ \frac{\partial \varphi}{\partial p_x} & \frac{\partial \varphi}{\partial p_y} & \frac{\partial \varphi}{\partial v_x} & \frac{\partial \varphi}{\partial v_y} \\ \frac{\partial \dot{\rho}}{\partial p_x} & \frac{\partial \dot{\rho}}{\partial p_y} & \frac{\partial \dot{\rho}}{\partial v_x} & \frac{\partial \dot{\rho}}{\partial v_y} \end{bmatrix} \tag{76}$$

So all the matrix H_j elements are calculated as follows:

1.
$$\frac{\partial \rho}{\partial p_x} = \frac{\partial}{\partial p_x}(\sqrt{p_x^2 + p_y^2}) = \frac{2p_x}{2\sqrt{p_x^2 + p_y^2}} = \frac{p_x}{\sqrt{p_x^2 + p_y^2}} \tag{77}$$

2.
$$\frac{\partial \rho}{\partial p_y} = \frac{\partial}{\partial p_y}(\sqrt{p_x^2 + p_y^2}) = \frac{2p_y}{2\sqrt{p_x^2 + p_y^2}} = \frac{p_y}{\sqrt{p_x^2 + p_y^2}} \tag{78}$$

3.
$$\frac{\partial \rho}{\partial v_x} = \frac{\partial}{\partial v_x}(\sqrt{p_x^2 + p_y^2}) = 0 \tag{79}$$

4.
$$\frac{\partial \rho}{\partial v_y} = \frac{\partial}{\partial v_y}(\sqrt{p_x^2 + p_y^2}) = 0 \tag{80}$$

5.
$$\frac{\partial \varphi}{\partial p_x} = \frac{\partial}{\partial p_x} arctan(p_y/p_x) = \frac{1}{(\frac{p_y}{p_x})^2 + 1}(-\frac{p_y}{p_x^2}) = -\frac{p_y}{p_x^2 + p_y^2} \tag{81}$$

6.
$$\frac{\partial \varphi}{\partial p_y} = \frac{\partial}{\partial p_y} arctan(p_y/p_x) = \frac{1}{(\frac{p_y}{p_x})^2 + 1}(\frac{1}{p_x}) = \frac{p_x^2}{p_x^2 + p_y^2}\frac{1}{p_x} = \frac{p_x}{p_x^2 + p_y^2} \tag{82}$$

7.
$$\frac{\partial \varphi}{\partial v_x} = \frac{\partial}{\partial v_x} arctan(p_y/p_x) = 0 \tag{83}$$

8.
$$\frac{\partial \varphi}{\partial v_y} = \frac{\partial}{\partial v_y} arctan(p_y/p_x) = 0 \tag{84}$$

9.
$$\frac{\partial \dot{\rho}}{\partial p_x} = \frac{\partial}{\partial p_x}\left(\frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}\right) \tag{85}$$

In order to calculate the derivative of this function we use the quotient rule.

Given a function z that is quotient of two other functions f and g:

$$z = \frac{f}{g} \tag{86}$$

its derivative with respect to x is defined as:

$$\frac{\partial z}{\partial x} = \frac{\frac{\partial f}{\partial x} g - \frac{\partial g}{\partial x} f}{g^2} \tag{87}$$

in our case:

$$f = p_x v_x + p_y v_y \tag{88}$$

$$g = \sqrt{p_x^2 + p_y^2} \tag{89}$$

Their derivatives are:

$$\frac{\partial f}{\partial p_x} = \frac{\partial}{\partial p_x}(p_x v_x + p_y v_y) = v_x \tag{90}$$

$$\frac{\partial g}{\partial p_x} = \frac{\partial}{\partial p_x}\left(\sqrt{p_x^2 + p_y^2}\right) = \frac{p_x}{\sqrt{p_x^2 + p_y^2}} \tag{91}$$

Putting everything together into the derivative quotient rule we have:

$$\frac{\partial \dot{\rho}}{\partial p_x} = \frac{v_x \sqrt{p_x^2 + p_y^2} - \frac{p_x}{\sqrt{p_x^2+p_y^2}}(p_x v_x + p_y v_y)}{p_x^2 + p_y^2} = \frac{p_y(v_x p_y - v_y p_x)}{(p_x^2 + p_y^2)^{3/2}} \tag{92}$$

10. Similarly

$$\frac{\partial \dot{\rho}}{\partial p_y} = \frac{\partial}{\partial p_y}\left(\frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}\right) = \frac{p_x(v_y p_x - v_x p_y)}{(p_x^2 + p_y^2)^{3/2}} \tag{93}$$

11.

$$\frac{\partial \dot{\rho}}{\partial v_x} = \frac{\partial}{\partial v_x}\left(\frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}\right) = \frac{p_x}{\sqrt{p_x^2 + p_y^2}} \tag{94}$$

12.

$$\frac{\partial \dot{\rho}}{\partial v_y} = \frac{\partial}{\partial v_y}\left(\frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}\right) = \frac{p_y}{\sqrt{p_x^2 + p_y^2}} \tag{95}$$

So now, after calculating all the partial derivatives our resulted Jacobian H_j is:

$$H_j = \begin{bmatrix} \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} & 0 & 0 \\ -\frac{p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 & 0 \\ \frac{p_y(v_x p_y - v_y p_x)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x(v_y p_x - v_x p_y)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} \end{bmatrix} \tag{96}$$

22 EKF Algorithm Generalization

No equations.

23 EKF with Linear Functions Quiz

No equations.

24 Sensor Fusion General Processing Flow

No equations.

25 General EKF Algorithm

No equations.

26 Evaluating the Performance

26.1 RMSE

$$\boxed{RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_t^{est} - x_t^{true})^2}} \tag{97}$$

KF vs. EKF

Kalman Filter	Extended Kalman Filter
Prediction	
$x' = Fx + u$	$x' = f(x, u)$ $u = 0$
$P' = FPF^T + Q$	use F_j instead of F
Measurement update	
$y = z - Hx'$	$y = z - h(x')$
$S = HP'H^T + R$	use H_j instead of H
$K = P'H^TS^{-1}$	
$x = x' + Ky$	
$P = (I - KH)P'$	

- F will be replaced with F_j when calculating P'
- H will be replaced by H_j when calculating S, K and P
- To calculate x', the prediction update function, f, is used instead of the F matrix
- To calculate y, the h function is used instead of the H matrix.

NOTE: one important point to reiterate is that the equation $y = z - H'x$ for the KF does not become $y = z - H_j x$ for the EKF. Instead, for EKF, we'll use the h function directly to map predicted locations x' from Cartesian to Polar coordinates, so $y = z_{radar} - h(x')$