① State transition func^n

$$\boxed{x_{k_p} = F x_{k-1} + u_k}$$

PAGE:
DATE:

$V = u + at$

$V_x \Delta t \rightarrow$ Displacement

State transition Eqn's. or Kinematic Eqn's

$$\begin{cases} P_x' = P_x + V_x \Delta t + V_{px} \\ P_y' = P_y + V_y \Delta t + V_{py} \\ V_x' = V_x + V_{vx} \\ V_y' = V_y + V_{vy} \end{cases}$$

$S = ut + \frac{1}{2} at^2$

$V = u + at$

$$\underset{(4\times1)}{\begin{pmatrix} P_x' \\ P_y' \\ V_x' \\ V_y' \end{pmatrix}} = \overbrace{\underset{(4\times4)}{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}}^{F} \underset{(4\times1)}{\begin{pmatrix} P_x \\ P_y \\ V_x \\ V_y \end{pmatrix}} + \underset{(4\times1)}{\begin{pmatrix} V_{px} \\ V_{py} \\ V_{vx} \\ V_{vy} \end{pmatrix}}$$

($u_k \rightarrow$ noise vector)

$$u_k = \begin{pmatrix} V_{px} \\ V_{py} \\ V_{vx} \\ V_{vy} \end{pmatrix} = \underset{(4\times1)}{\begin{pmatrix} a_x \Delta t^2/2 \\ a_y \Delta t^2/2 \\ a_x \Delta t \\ a_y \Delta t \end{pmatrix}}$$

$s = ut + \frac{1}{2}at^2$

$v = u + at$

$a = \frac{\Delta V}{\Delta t} = \frac{V_{k+1} - V_k}{\Delta t}$

We are predicting the object state x

Scanned by CamScanner

② $P_{K_p} = F P_{K-1} F^T + Q_K$ | Predicting the object covariance (P)

{from ①}

$$\underset{\substack{\text{noise}\\\text{vector}\\(4\times1)}}{u_K} = \begin{pmatrix} a_x \Delta t^2/2 \\ a_y \Delta t^2/2 \\ a_x \Delta t \\ a_y \Delta t \end{pmatrix} = \underbrace{\underset{(4\times2)}{\begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix}}}_{G} \underbrace{\underset{(2\times1)}{\begin{pmatrix} a_x \\ a_y \end{pmatrix}}}_{a}$$

$$Q_K = E[u_K u_K^T]$$
$$= E[G a G^T a^T] = G\{E[a a^T]\} G^T$$
$$= G \underbrace{\begin{pmatrix} \sigma^2_{ax} & \sigma_{ax} \\ \sigma_{ay} & \sigma^2_{ay} \end{pmatrix}}_{Q_v} G^T$$

$$\therefore \boxed{Q_K = G Q_v \cdot G^T}$$

$$Q_v = \underset{(2\times2)}{\begin{pmatrix} \sigma^2_{ax} & 0 \\ 0 & \sigma^2_{ay} \end{pmatrix}}$$

$$Q_K = (4\times2)(2\times2)(2\times4)$$
$$= (4\times4)$$

$$\therefore \underset{(4\times4)}{P_{K_p}} = (4\times4)(4\times4)(4\times4) + (4\times4)$$
$$= (4\times4)$$

→ 'F' matrix (from ①)

3. $\underset{(2\times1)}{y_K} = \underset{(\text{sensor } 1)}{Z_K} - H x_{K_p}$

$$= \underset{(2\times1)}{\begin{pmatrix} P_x \\ P_y \end{pmatrix}} - \underset{(2\times4)}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}} \underset{(4\times1)}{\begin{pmatrix} P_x' \\ P_y' \\ V_x' \\ V_y' \end{pmatrix}}$$

$Z_K \rightarrow$ Lidar measurements
i.e. Sensor 1

4. $S_K = H P_{K_p} H^T + R$

$$\underset{(2\times2)}{} = \underset{\downarrow \, 3}{(2\times4)} \, \underset{\downarrow \, 2}{(4\times4)} \, \underset{\downarrow \, 3}{(4\times2)} + (2\times2)$$

Eg. $R = E[w w^T]$

$$= \begin{pmatrix} \sigma_{P_x}^2 & 0 \\ 0 & \sigma_{P_y}^2 \end{pmatrix}_{2\times2}.$$

5. $K_K = P_{K_p} H^T S_K^{-1}$

$$(4\times2) = \underset{\downarrow \, \text{from } 2}{(4\times4)} \; \underset{\downarrow \, 4}{(4\times2)} \; \underset{\downarrow \, \text{from } 4}{(2\times2)}$$

⑥ $x_k = x_{kp} + K_k \cdot y_k$.

$$\underset{\text{from } ①}{(4\times1)} = \underset{\text{from } ①}{(4\times1)} + \underset{\text{from } ⑤}{(4\times2)}\ \underset{\text{from } ③}{(2\times1)}$$

⑦. $P_k = (I - K_k H) P_{kp}$.

$$= P_{kp} I - K_k H \cdot P_{kp}$$

$$(4\times4) = \underset{\text{from } ②}{(4\times4)}(4\times4) - \underset{\text{from } ⑤}{(4\times2)}\ \underset{\text{from } ③}{(2\times4)}\ \underset{\text{from } ②}{(4\times4)}$$

⑧. $x_{(k+1)p} = F\,\cancel{x_k} + u_k$

$$\underset{(4\times1)}{\begin{pmatrix} P_x' \\ P_y' \\ V_x' \\ V_y' \end{pmatrix}} = \underset{\text{from } ①}{(4\times4)}\ \underset{\text{from } ⑥}{(4\times1)} + \underset{\text{from } ①}{(4\times1)}$$

⑨. $P_{(k+1)_p} = F P_k F^T + Q_{k}$

$$(4\times4) = (4\times4)\ (4\times4)\ (4\times4) + (4\times4)$$

$\downarrow$ from ⑦. $\downarrow$ from ⑦. $\downarrow$ from ⑦. $\downarrow$ from ②.

⑩. $y_{k+1} = Z_{k+1} - h(x')_{k+1}$
(Sensor 2)

$$\underset{(3\times1)}{y_{k+1}} = \underset{(3\times1)}{\begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix}} - \underset{(3\times1)}{\begin{pmatrix} \sqrt{P_x'^2 + P_y'^2} \\ \arctan\left(P_y'/P_x'\right) \\ \dfrac{P_x' V_x' + P_y' V_y'}{\sqrt{P_x'^2 + P_y'^2}} \end{pmatrix}} \rightarrow \text{Non-linear equations}$$

Sensor 2 measurement (RADAR)

from ⑧.

$\rho \rightarrow$ range
$\varphi \rightarrow$ bearing
$\dot{\rho} \rightarrow$ radial velocity

(11). $S_{k+1} = H_j P_{(k+1)p} H_j^T + R$.

$(3\times3) = (3\times4)\ (4\times4)\ (4\times3) + (3\times3)$

↓

from (9).

$$R = \begin{pmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\psi^2 & 0 \\ 0 & 0 & \sigma_{\dot\rho}^2 \end{pmatrix}$$

↓

Radar measurement Covariance matrix.

wrt, $Z_{(k+1)} = \begin{pmatrix} \rho \\ \psi \\ \dot\rho \end{pmatrix}_{(3\times1)}$ → from (10).

where, $\rho = \sqrt{P_x'^2 + P_y'^2}$

$\psi = \arctan\left(P_y'/P_x'\right)$

$$\dot\rho = \frac{P_x' V_x' + P_y' V_y'}{\sqrt{P_x'^2 + P_y'^2}}$$

→ from (8) &

from (10)

$$\therefore H_j = \begin{pmatrix} \dfrac{\partial \rho}{\partial p'_x} & \dfrac{\partial \rho}{\partial p'_y} & \dfrac{\partial \rho}{\partial v'_x} & \dfrac{\partial \rho}{\partial v'_y} \\[2ex] \dfrac{\partial \varphi}{\partial p'_x} & \dfrac{\partial \varphi}{\partial p'_y} & \dfrac{\partial \varphi}{\partial v'_x} & \dfrac{\partial \varphi}{\partial v'_y} \\[2ex] \dfrac{\partial \dot\rho}{\partial p'_x} & \dfrac{\partial \dot\rho}{\partial p'_y} & \dfrac{\partial \dot\rho}{\partial v'_x} & \dfrac{\partial \dot\rho}{\partial v'_y} \end{pmatrix}_{(3 \times 4)}$$

$H_j \rightarrow$ Jacobian matrix

$$H_j = \begin{bmatrix} \dfrac{p'_x}{\sqrt{p'^2_x + p'^2_y}} & \dfrac{p'_y}{\sqrt{p'^2_x + p'^2_y}} & 0 & 0 \\[3ex] -\dfrac{p'_y}{\sqrt{p'^2_x + p'^2_y}} & \dfrac{p'_x}{\sqrt{p'^2_x + p'^2_y}} & 0 & 0 \\[3ex] \dfrac{p'_y(v'_x p'_y - v'_x p'_y)}{(p'^2_x + p'^2_y)^{3/2}} & \dfrac{p'_x(v'_y p'_x - v'_x p'_y)}{(p'^2_x + p'^2_y)^{3/2}} & \dfrac{p'_x}{\sqrt{p'^2_x + p'^2_y}} & \dfrac{p'_y}{\sqrt{p'^2_x + p'^2_y}} \end{bmatrix}$$

(3×4)

(12) $K_{(k+1)} = P_{(k+1)p} H_j^T S_{k+1}^{-1}$

$(4\times3) = (4\times4)\ (4\times3)\ (3\times3)$

$\downarrow$ from (9). $\downarrow$ from (11). $\downarrow$ from (11).

(13). $x_{k+1} = x_{(k+1)p} + K_{(k+1)}\, y_{k+1}$

$(4\times1) = (4\times1) + (4\times3)\ (3\times1)$

$\downarrow$ from (8). $\downarrow$ from (12). $\downarrow$ from (10).

(14). $P_{k+1} = (I - K_{k+1} H_j)\, P_{(k+1)p}$

$= P_{(k+1)p}\, I - K_{(k+1)} H_j$

$(4\times4) = (4\times4)\ (4\times4) - (4\times3)\ (3\times4)$

$\downarrow$ from (9) $\downarrow$ from (12). $\downarrow$ from (11).

PAGE :
DATE : / /

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

Jacobian matrix

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

for calculating S, K and P.

$x = x' + Ky$

$P = (I - KH)P'$

* Drawbacks of EKF

- ✓ Difficult to calculate Jacobians
- ✓ High computational cost to calculate Jacobians.
- ✓ EKF works only on the systems that has differentiable model.

$$R_{(Lidar)} = \begin{bmatrix} 0.0225 & 0 \\ 0 & 0.0225 \end{bmatrix}$$

$$R_{(Radar)} = \begin{bmatrix} 0.09 & 0 & 0 \\ 0 & 0.0009 & 0 \\ 0 & 0 & 0.09 \end{bmatrix}$$

Scanned by CamScanner

① x_{kp} → predicted object state @ k
x_{k-1} → previous object state @ k−1

DATE: PAGE:

F → State transition matrix
→ written based on state transition Eq^n's ie (Kinematic Eq^n's).

We assume that the object is moving at constant velocity, so we predict that after time Δt, the velocity is same in each direction.

u_k → external motion
It gives information about how external actions affecting our object.

Eg: Driver of the car we're tracking is braking

{if we dont know anything, then $u_k = 0$ }.

② P_{kp} → Predicted object Covariance (or Predicted uncertainity)
P_{k-1} → (previous) object covariance matrix
object covariance → Uncertainity of object's state
∵ sensors often have measurement error.
(or measured uncertainity @ k−1)

F → State transition matrix

→ It's important to know approximately how

DATE: PAGE:

precise our measurement is.

If we are highly uncertain about ~~measurement~~ pos^n of car in front of us, we may not stay away from it to avoid crash.

→ FPF^T → we do this ∵ if we scale a variable (which is basically what a matrix does ∵ its mapping), we multiply its covariance by the scaling squared.

Eg: If F is a constant, then $Cov(F*x) = F^2 Cov(x)$. $\underline{FPF^T}$

Let is similar to that i.

Q_k → Process Covariance matrix

process covariance → It's the uncertainity about the true velocity of the object.

we've assumed constant velocity, but the object might be accelerating.

$a = \frac{\Delta V}{\Delta t} = \frac{V_{K-1} - V_K}{\Delta t}$

DATE: PAGE:

→ Q_K = depends on Δt & how variable our random acceleratⁿ is (ie ΔV)

(6) x_K → updated object state.

x_{Kp} → predicted object state.

(3) $y_K \Rightarrow \underset{\text{(sensor 1)}}{Z_K} - H x_{Kp}$

y_K → is the difference between sensor measurement (Z_K) & the predicted measurement (x_{Kp}),

H → Measurement matrix

H → it is changing unit of x, so it matches z.

y_K → Innovation variable
It is the predicted measurement error.

Z_K → Sensor measurement Vector

(5) $K_K \rightarrow$ Kalman Gain.

$P_{Kp} \rightarrow$ predicted object covariance (or predicted uncertainty)

$R \rightarrow$ measurement uncertainty matrix {Given by Manufacturer}

$H \rightarrow$ measurement matrix.

$$K_K = \frac{P_{Kp} H^T}{P_{Kp} H^T H + R}$$

✓ when R (measurement {sensor} uncertainty) is very large & the P_{Kp} (predicted ") is very small, K_K is close to 0.

Hence we give big weight to Prediction step & small weight to measurement update step.

✓ when R is very small & P_{Kp} is very large, K_K is close to 1.

Hence we give big weight to measurement update step & small weight to prediction step.

Scanned by CamScanner

(7) $P_k \rightarrow$ updated object covariance.

$P_{k|-1} \rightarrow$ predicted " "

DATE: PAGE:

$K_k \rightarrow$ Kalman Gain.

$H \rightarrow$ Measurement matrix

Imp. * Object covariance (P_k) decreases after every update step.

$I \rightarrow$ Identity matrix.

(2) We've assumed constant velocity, But the object didn't maintain same constant velocity, maybe the object change direction, accelerated or decelerated, So when we predict object's pos after one second, our uncertainity increases.

(8) & (10) Our predicted state $X_{k+1|k}$ represented by Gaussian distribution which is mapped to Non Linear function (Z_k sensor) & Radar measurement & then the resulting function will not be Gaussian funct.

So, the Kalman Filter is not applicable.

Solution DATE: PAGE:

→ One of the ~~feature~~ is to linearize $h(x')$ & non-linear func^n^s.

EKF uses method called I order Taylor Expansion to linearize the func^n^.

Hence we use H_j.

✓ I order Taylor Series

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

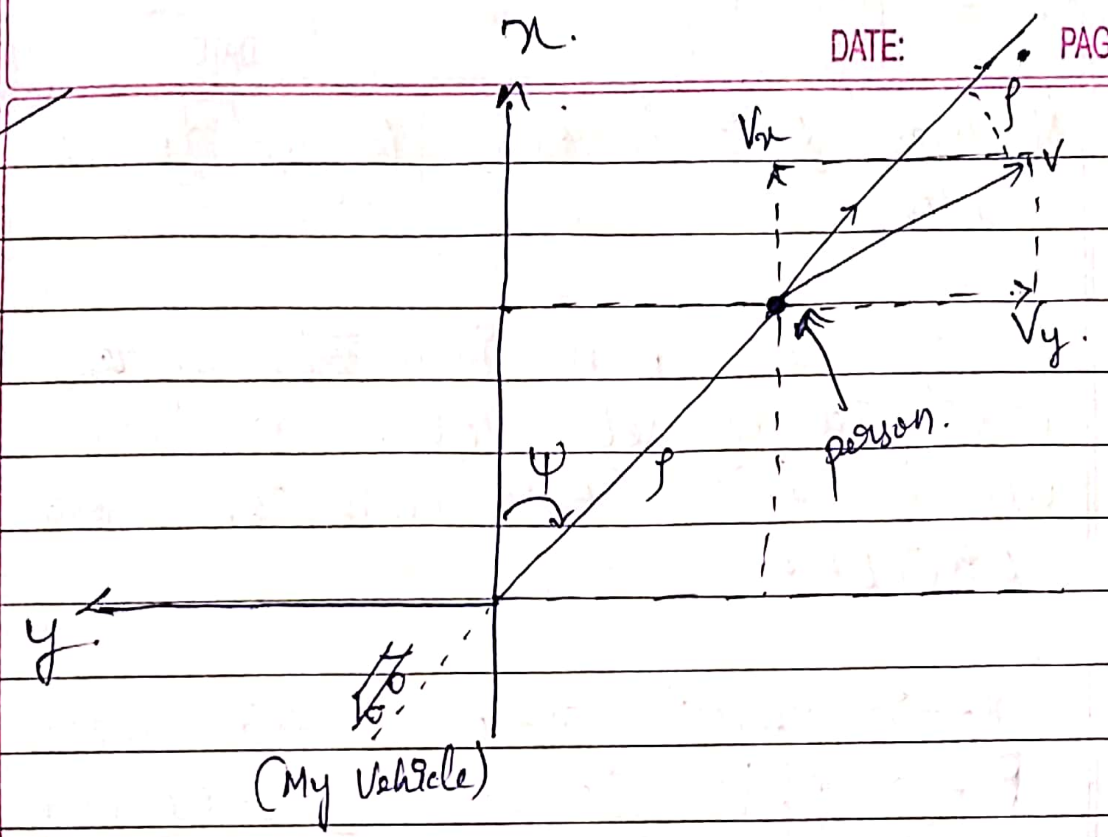
Jacobian.

✓ RMSE (Root mean square error)

Kalman filter Performance

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

DATE: PAGE:



$$Z_k = \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} \begin{matrix} \rightarrow \text{range} \\ \rightarrow \text{Bearing} \\ \rightarrow \text{radial Velocity} \end{matrix}$$
(sensor 2)

$\rho \rightarrow$ Radial dist. from the Origin
$\varphi \rightarrow$ Angle betn ρ & x.
$\dot{\rho} \rightarrow$ Rate of change of ρ.

{ suppose, ρ = distance then, $\dot{\rho}$ = rate of change of distance, which is Velocity. }

$$Z_k = \begin{pmatrix} P_x \\ P_y \end{pmatrix} \rightarrow \text{position}$$
(Sensor 1)

$$H_{expected} = H_k x_k$$
$$\Sigma_{expected} = H_k P_k H_k^T$$

DATE: PAGE:

- ✓ Kalman filter is only designed for linear systems.

- ✓ If we have non-linear ~~state~~ systems & want to estimate state, we need to use non-linear state estimators like EKF, UKF.

- ✓ $x \rightarrow$ object state or state vector
  $P \rightarrow$ object covariance or state covariance matrix

- ✓ Lidar $\rightarrow$ Gives information about current position of the object.
  Measurement have more resolution.

Radar $\rightarrow$ Gives info about current position & velocity of the objects

Measurements have less resolution

Previous state

(A) x_{K-1}, P_{K-1}

$x_{K_p} = F \cdot x_{K-1} + U_{K-1}$ — (1)

$P_{K_p} = F \cdot P_{K-1} F^T + Q_{K-1}$ — (2)

Predict @ K (KF)

Sensor 1 = LiDAR

Update @ K (KF)

$y_K = \underset{(\text{sensor } 1)}{Z_K} - H x_{K_p}$ — (3)

$S_K = H P_{K_p} H^T + R$ — (4)

$\underset{(0\text{-}1)}{K_K} = \dfrac{P_{K_p} H^T}{\underset{(S_K)}{H P_{K_p} H^T + R}}$ — (5)

$x_K = x_{K_p} + K_K \cdot y_K$ — (6)

$P_K = (I - K_K H) P_{K_p}$ — (7)

Predict @ K+1 (KF)

$x_{(K+1)_p} = F x_K + U_K$ — (8)

$P_{(K+1)_p} = F P_K F^T + Q_K$ — (9)

update @ K+1 (EKF)

Sensor 2 = RADAR

$y_{K+1} = \underset{(\text{sensor } 2)}{Z_{K+1}} - h(x')_{K+1}$ — (10)

$S_{K+1} = H_j P_{(K+1)_p} H_j^T + R$ — (11)

$K_{K+1} = \dfrac{P_{(K+1)_p} H_j^T}{S_{K+1}}$ — (12)

$x_{K+1} = x_{(K+1)_p} + K_{(K+1)} y_{K+1}$ — (13)

$P_{K+1} = (I - K_{K+1} H_j) P_{(K+1)_p}$ — (14)