Date
Page No.

Camera calibration

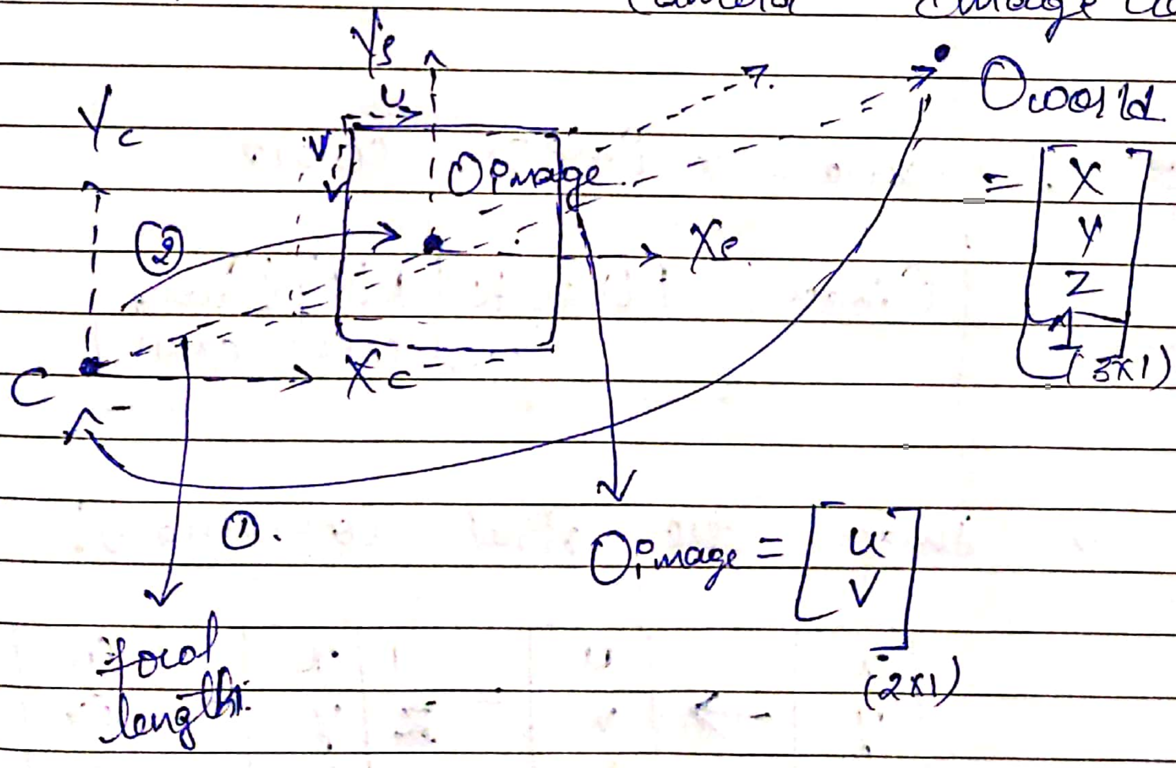
(2) Objective

find camera matrix 'P' through a process called camera calibration.

Extract the intrinsic & extrinsic parameters from the camera matrix 'P'

(1) projection : Convert world frame Co-ord to ~~camera~~ (image) frame co-ordinates.

Steps: 1) project from world to camera coord.
2) " " camera " image co-ord.



Date
Page No.

1) world → Camera co-ord.

$$O_{camera} = \underset{(3\times4)}{[R|t]} \underset{(4\times1)}{O_{world}}$$

2) Camera → Image Co-ord.

$$O_{image} = \underbrace{\begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{K \\ (3\times3)}} O_{camera} = K\,O_{camera}$$

∗ ∴ world → image co-ord.

$$\boxed{O_{image} = PO = \underset{(3\times3)}{K} \underset{(3\times4)}{[R|t]} \underset{(4\times1)}{O_{world}}}$$

3) Image to Pixel Coordinates:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} su \\ sv \\ s \end{bmatrix}$$

s → arbitrary scale
It is not possible to estimate depth

* we consider $s = Z_c$

Date
Page No.

(2) cont -

when we convert world → Image coord scale info is lost.

points in 3D space along a ray from camera center all project the same location on image plane hence not possible to estimate depth only using image info.

hence we use arbitrary scale (s)

K → Intrinsic Parameter

[R|t] → Extrinsic rotational Parameter 'R' & translational parameter 't'

find least squares solution (non-linear soln) of the parameters P. (find unknown intrinsic & extrinsic parameters)

* $\theta = P\theta = \underset{\downarrow}{K[R|t]}$. θ

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & - & - & P_{24} \\ P_{31} & - & - & P_{34} \end{bmatrix}}_{P \text{ (camera matrix)}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Pixel co-ord. (Image) — world co-ord. — (*)

$su - P_{11}X - P_{12}Y - P_{13}Z - P_{14} = 0$ — (1)

$sv - P_{21}X - - - - P_{24} = 0$ — (2)

$s - P_{31}X - P_{32}Y - - - P_{34} = 0$ — (3)

Replace eqn (3) in (1) & (2).

Date
Page No.

$$P_{31}Xu + P_{32}Yu + P_{33}Zu + P_{34}u - P_{11}X - P_{12}Y - P_{13}Z - P_{14} = 0$$
$$P_{31}Xv + P_{32}Yv + P_{33}Zv + P_{34}v - P_{21}X - P_{22}Y - P_{23}Z - P_{24} = 0$$

factoring the 'P' matrix
RQ factorization

$$P = K\,[R|t]$$
$$= K\,[R\,I \;-\; RC]$$
$$= [KR\,I \;-\; KRC]$$

Let $KR = M$.

RQ-factorization i.e. $M \to RQ$

$$= [M\,I \;-\; MC]$$

'C' is the column vector representing camera's pos^n in world co-ord.

where, $M = \underset{(3\times3)}{R}\,\underset{(3\times3)}{Q} = KR$

M is an 3X3 matrix

✓ Intrinsic calibrat^n matrix
$$K = R$$

✓ Rotational matrix :
$$R = Q$$

✓ Translational vector :
$$t = -K^{-1}P[:,4] = -K^{-1}MC$$

* Any square matrix can be factored into upper triangular matrix "R" & orthogonal basis "Q"

$$P = K[R \mid -RC]$$

K = 3X3 matrix (it is upper triangular matrix) describes camera's parameters like focal length.

R = rotation matrix whose column are the direct^n of the world axes in the camera's reference frame.

C = is vector, representing camera's center in world co-ord

Vector, $t = -RC$ gives the pos^n of the world origin in camera coordinates.

$$[R \mid t] = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right] \text{ or } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$(u_0, v_0) \rightarrow$ image pixel coord. of origin

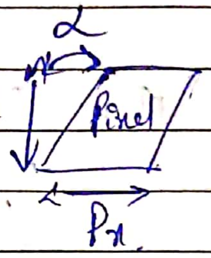
WKT. $P = K[R \mid t]$

$\therefore x = PX$

$$\boxed{\underset{(3X1)}{x} = \underset{(3X3)}{K}\,\underset{(3X4)}{[R \mid t]}\,\underset{(4X1)}{X}}$$

Intrinsic Parameters

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & C_y & 1 \end{bmatrix}$$



$(f_x, f_y) \rightarrow$ focal length in pixel

$f_x = F / P_x$

$f_y = F / P_y$

F = focal length in world units (millimeters)

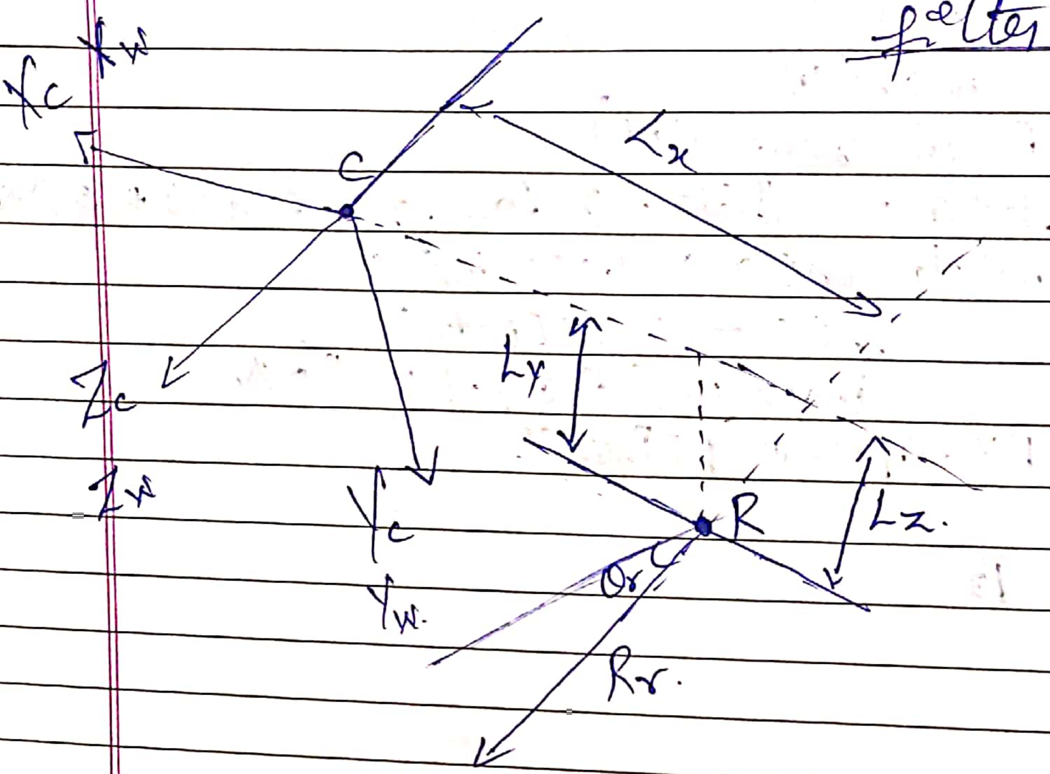
(P_x, P_y) = size of pixel in world ~~units~~ in X & Y axis

s = skew co-eff.

$s = f_x \tan(\alpha)$

$(x_0, y_0) \rightarrow$ Image pixel co-ord. of origin. is the intersection point of the optic axis Z_c & the image plane.

Radar and Camera Data fusion using Extended Kalman filter



C → Camera.
R → Millimeter wave radar

X_c, Y_c, Z_c → Camera co-ordinates
X_w, Y_w, Z_w → World co-ordinates
R_r, θ_r, V_r → Radar coordinates

R_r → radial distance
θ_r → Yaw angle (azimuthal)
V_r → radial velocity

} Radar

L_x, L_y, L_z → Distance between camera coordinate system & radar coordinate system.

* Radar & camera calibration.

1) Transformation relation ~~between~~ of Radar to camera coordinate system.

$$\underset{(4\times1)}{\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}} = \underbrace{\underset{(4\times4)}{\begin{bmatrix} 0 & -1 & 0 & L_x \\ 0 & 0 & 0 & L_y \\ 1 & 0 & 0 & L_z \\ 0 & 0 & 0 & 0 \end{bmatrix}}}_{G} \underset{(4\times1)}{\begin{bmatrix} R_r \cdot \cos\theta_r \\ R_r \sin\theta_r \\ 0 \\ 1 \end{bmatrix}}$$

2) Transformation relation ~~between~~ of camera to pixel coord s/m.

$$Z_c \underset{(3\times1)}{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}} = \underbrace{\underset{(3\times4)}{\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}}_{F} \underset{(4\times1)}{\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}}$$

3) Transformation relation of Radar to pixel coordinate system.

$$Z_c \underset{(3\times1)}{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}} = \underbrace{\underset{(3\times4)}{\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}}_{F} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{W\ (4\times4)} \underbrace{\underset{(4\times4)}{\begin{bmatrix} 0 & -1 & 0 & L_x \\ 0 & 0 & 0 & L_y \\ 1 & 0 & 0 & L_z \\ 0 & 0 & 0 & 0 \end{bmatrix}}}_{G} \begin{bmatrix} R_r\cos\theta_r \\ R_r\sin\theta_r \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = F \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times G \times \begin{bmatrix} R_r \cdot \cos\theta_r \\ R_r \cdot \sin\theta_r \\ 0 \\ 1 \end{bmatrix}$$

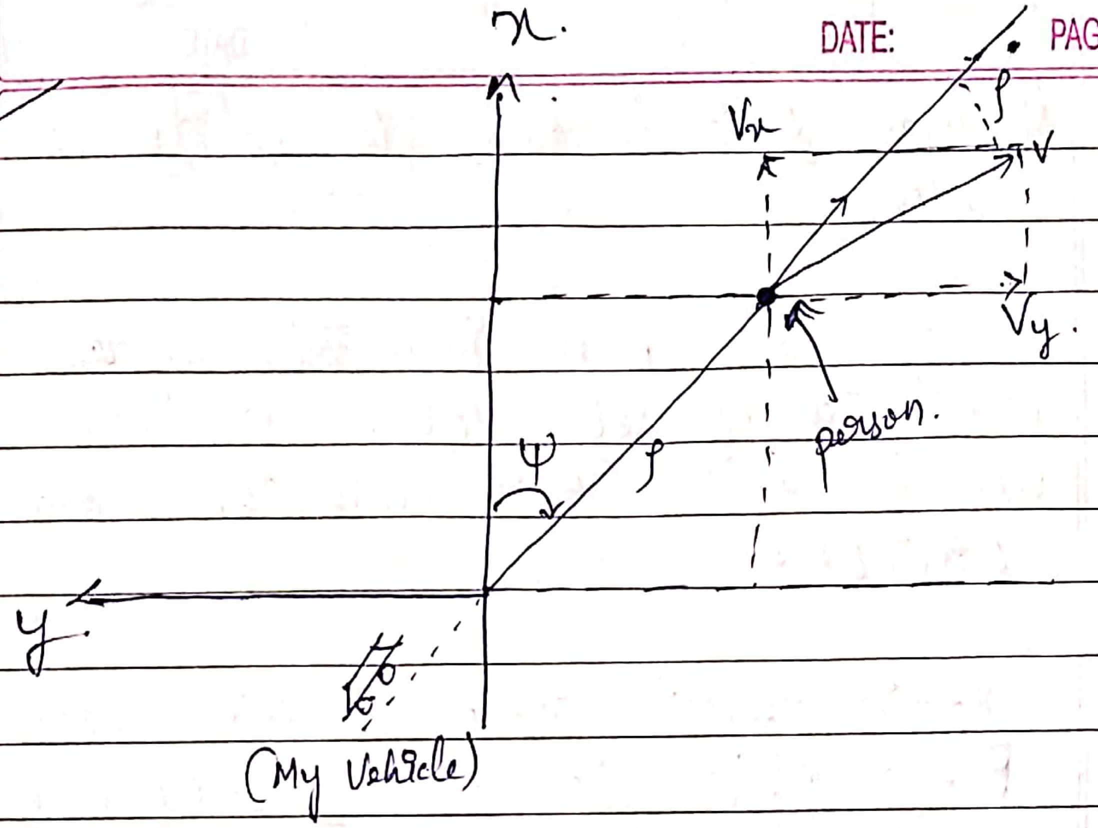
Date
Page No.

F → intrinsic parameter of camera

W → Transformation matrix from world coord s/m to camera co-ord system

G → Transformation matrix from radar co-ord system to world coordinate system.

Let, $Z_c \times u = P_x$.
$Z_c \times V = P_y$.



$$\underset{(\text{sensor2})}{Z_k} = \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} \begin{matrix} \rightarrow \text{range} \\ \rightarrow \text{Bearing} \\ \rightarrow \underline{\text{radial Velocity}} \end{matrix}$$

$\rho \rightarrow$ 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 <u>Velocity</u>. }

$$\underset{(\text{Sensor1})}{Z_k} = \begin{pmatrix} P_x \\ P_y \end{pmatrix} \rightarrow \text{position}$$

Previous state

(A)

$$x_{K-1}, \quad P_{K-1}$$

Predict @ K (KF)

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

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

Sensor 1 = LiDAR

update @ K (KF)

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

$$S_K = H P_{K_p} H^T + R \quad \text{— (4)}$$

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

$$x_K = x_{K_p} + K_K \cdot y_K \quad \text{— (6)}$$

$$P_K = (I - K_K H) P_{K_p} \quad \text{— (7)}$$

Predict @ K+1 (KF)

$$x_{(K+1)_p} = F x_K + U_K \quad \text{— (8)}$$

$$P_{(K+1)_p} = F P_K F^T + Q_K \quad \text{— (9)}$$

update @ K+1 (EKF)

sensor 2 = RADAR

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

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

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

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

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

PAGE :
DATE :

1. State transition func^n

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

State transition Eqn's. or Kinematic Eqn's

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

$V = u + at$

$V_x \Delta t \rightarrow$ Displacement

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

$V = u + at$

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

($u_k \rightarrow$ 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}

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

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

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

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

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

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

→ 'F' matrix (from ①)

③. $y_K = Z_K - H x_{K_p}$
(sensor 1)

(2×1)

$$= \begin{pmatrix} P_x \\ P_y \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_x' \\ P_y' \\ V_x' \\ V_y' \end{pmatrix}$$

(2×1) (2×4) (4×1)

$Z_K \to$ <u>Lidar</u> measurements i.e. Sensor 1

④. $S_K = H P_{K_p} H^T + R$

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

↓ ③ ↓ ② ↓ ③

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

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

⑤. $K_K = P_{K_p} \cdot H^T S_K^{-1}$

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

↓ from ②. ↓ ④. ↓ from ④.

Scanned by CamScanner

⑥ $x_k = x_{kp} + K_k \cdot y_k$.

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

⑦. $P_k = (I - K_k H) P_{kp}$.

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

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

⑧. $x_{(k+1)p} = F \cancel{X_k} + U_k$

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

⑨. $P_{(k+1)_p} = F P_k F^T + Q_{k+1}$

PAGE :
DATE : / /

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

from ①. from ⑦. from ①. from ②.

⑩. $y_{k+1} = Z_{k+1} - h(x')_{k+1}$

(Sensor 2)

$$y_{k+1} = \begin{pmatrix} \rho \\ \varphi \\ \dot{\rho} \end{pmatrix} - \begin{pmatrix} \sqrt{P_x'^2 + P_y'^2} \\ \arctan(P_y'/P_x') \\ \dfrac{P_x' V_x' + P_y' V_y'}{\sqrt{P_x'^2 + P_y'^2}} \end{pmatrix}$$

(3×1) (3×1) (3×1)

→ Non-linear equations

Sensor 2 measurement (RADAR)

from ⑧.

$\rho \rightarrow$ range

$\varphi \rightarrow$ bearing

$\dot{\rho} \rightarrow$ radial velocity

Scanned by CamScanner

(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.

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

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

(3X4)

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

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

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

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

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

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

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

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

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

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

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 uncertainty)

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 $Cov(F * x) = F^2 Cov(x)$. FPF^T

Let is similar to that is

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 acceleration ω (ie ΔV)

⑥ x_K → updated object state.

x_{Kp} → predicted object state.

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

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

H → Measurement matrix
H → it is changing units of x, such it matches z.

y_K → Innovation variable
It is the predicted measurement error.

Z_K → Sensor measurement vector

Scanned by CamScanner

(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.

(7) P_k → updated object covarience.

P_{k+1} → predicted " "

DATE: PAGE:

K_k → Kalman Gain.

H → Measurement matrix

Imp.

* Object covarience (P_k) decreases after every update step.

I → Identity matrix.

(9) 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.

Scanned by CamScanner

Solution DATE: PAGE:

→ One of the ~~feature~~ is to Linearise $h(x')$. 2 non-linear func^n.

EKF uses method called I order Taylor Expansion to linearise 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

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