udacity

Sensor Fusion

Kalman filters good for different sensors → combine strenghts of Radar & LIDAR

projects track pedestrian

Extended: More Complex Motion Models & Measurement Models

LIDAR → RADAR

first Measurement?

yes → Initialize State and Covariance Matrices

No → Predict

then Measurement Uptade: = correction step

LASER → Setup Laser Matrices

RADAR → Setup Radar Matrices

State update with new Measurement

Estimate state of pedestrian: 2D Position p_x, p_y

2D Velocity p'_x, p'_y

Update: Combine Measurement with Prediction

emphasis depending on uncertainty of each value (R & P)

Timestep K: Prediction

K+1: measurement uptade

→ new belief → posterior

Laser: Cartesian

Radar: Polar coordinate System

X: mean state Vector: position & velocity (v) → gaussian distribution with mean X

P state covariance matrix: info about uncertainty of pos. & Velocity (V) → like std. deviation

K: Timesteps: X_k → pos & v at time k

State transition function → show state changes from k-1 to k

Measurement function: how Measurement is calculated & how its related to predicted X

→ State transition: $x' = f(x) + v = Fx + \cancel{Bu}^{0} + v$

$x' = Fx + v$ ← stochastic noise part (random)

Measurement: $z = h'(x) + w = Hx' + w$

Prediction

Deterministic part

$\underset{\text{new pos}}{p'} = \underset{\text{old pos}}{p} + \underset{\text{velocity}}{v}\,\Delta t$ → Matrixform → $\begin{pmatrix} p' \\ v' \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}}_{F} \begin{pmatrix} p \\ v \end{pmatrix}$ while $v = \text{const.}$

Measurement lidar: $z = p'$ (no velocity measured)

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

Kalman filter prediction

$x' = Fx + v$ predict. calculation

$P' = FPF^T + Q$ → covariance

represents → increase in uncertainty

$v \sim N(0, Q)$ → gaussian

↳ mean

KF Update

$y = z - Hx'$

$S = HP'H^T + R$

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

$x = x' + Ky$

$P = (I - KH)P'$

EKF Variables

difference pos location

X: mean state Vector

P state covariance matrix → std.dev ✓

K: Timestep ✓

deterministic

X': new position Change in state ✓ $= f(x) + v = Fx + v$ → stochastic velocity?

?

$\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} = F$

Prediction

p': $p + v\Delta t$: new pos

change or new?

p: old pos ✓

P vs p

V: velocity ✓

t: time ✓

$V' = V$ new velocity: const

change or new

prediction

$Z = h'(x) + w = Hx' + w$ — stochastic

?

$Z = p'$ (Measurement) $= (1\ 0) \begin{pmatrix} p' \\ v' \end{pmatrix}$

?

w $w \sim N(0, R)$ → "measurement noise as gaussian"

Q: representation of uncertainty → or $P' = FPF^T + Q$

↳ covariance → acceleration noise

$Y = Z - Hx'$ measurement (Sensor) Difference Prediction & sensor "comparison: where we think we are & sensor data"

↳ Prediction? where we think we are

K Kf gain Matrix: combination of P' (uncertainty pos) & R (uncertainty measurement)

R uncertainty Measurement relative to P' → measurement covariance Matrix

B control input matrix

u control vector/external motion

v random noise of velocity?

$X' = Fx + Bu + v$

↳ Updated position due to acceleration

↳ here: unknown → $Bu = 0$ we represent it as random noise v

↳ pedestrian can chance acceleration randomly

$F = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$ new pos $x' = F \cdot x + \text{noise}$ F is matrix that, when multiplied with x, predicts pos after Δt

$P' = p + v * \Delta t$ new pos/location

H Matrix that will drop/forget velocity information from state vector x: $z = H * X + w$

Dropping ~~the~~ velocity needed to compare lidar measurement & belief

↳ $z = H * X + w$

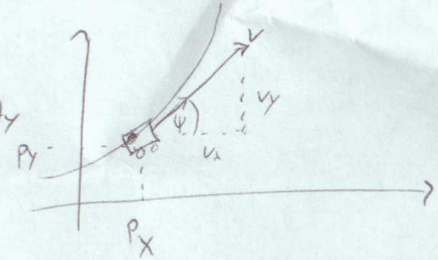
↳ Lidar has no velocity info; Radar does.

I identity Matrix

Unscented Kalman Filters

CTRV Const. turn rate and velo. Mag. Model

State vec.: $X = \begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot{\psi} \end{bmatrix}$ Pos x, Pos y, Magnitude of velocity, Yaw angle, Yaw rate



Process model f $\quad x_{k+1} = f(x_k, \nu_k)$

change Rate of state $\dot{X} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos(\psi) \\ v\sin(\psi) \\ 0 \\ \dot{\psi} \\ 0 \end{bmatrix}$

$$t_{k+1} = \Delta t + t_k \Rightarrow X_{k+1} = X_k + \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{X}(t) \end{bmatrix} dt = \begin{bmatrix} \frac{v_k}{\dot{\psi}_k}\left(\sin(\psi_k + \dot{\psi}_k \Delta t) - \sin(\psi_k)\right) \\ \frac{v_k}{\dot{\psi}_k}\left(-\cos(\psi_k + \dot{\psi}_k \Delta t) + \cos(\psi_k)\right) \\ 0 \\ \dot{\psi}_k \Delta t \\ 0 \end{bmatrix}$$

Longitudinal accel. noise
$\nu_{a,k} \sim N(0, \sigma_a^2)$

Yaw accel. Noise

$\nu_{\ddot{\psi},k} \sim N(0, \sigma_{\ddot{\psi}}^2)$

~~[illegible]~~

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \cos(\psi_k)\, \nu_{a,k} \\ \frac{1}{2}(\Delta t)^2 \sin(\psi_k) \cdot \nu_{a,k} \\ \Delta t \cdot \nu_{a,k} \\ \frac{1}{2}(\Delta t)^2 \cdot \nu_{\ddot{\psi},k} \\ \Delta t \cdot \nu_{\ddot{\psi},k} \end{bmatrix}$$

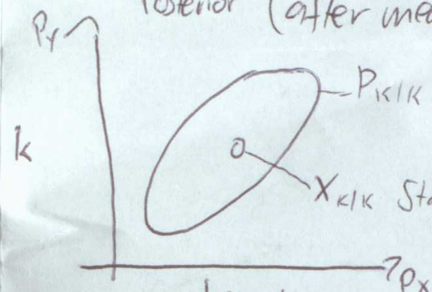
grows linearly with time

+Noise

$$\Rightarrow \boxed{X_{k+1} = X_k + \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{X}(t) \end{bmatrix} dt + \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}}$$

Sigma Points Generation Predict

Posterior (after measurement)

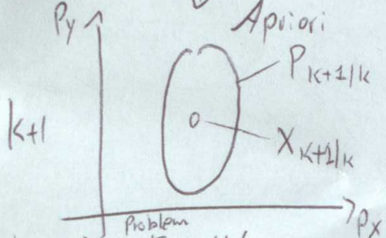


$P_{k|k}$ state covariance

$X_{k|k}$ State Vector Mean (Mittelwert/Durchschnitt)

↓ Predict

Apriori



$P_{k+1|k}$

$X_{k+1|k}$

(after prediction) estimation

covariance:
if Normal distributed: Ellipse
↳ only when linear

Problem
when linear: $X_{k+1} = F x_k + \nu_k$

$Q = E\{\nu_k \cdot \nu_k^T\}$ (Process noise)(Covariance Matrix)

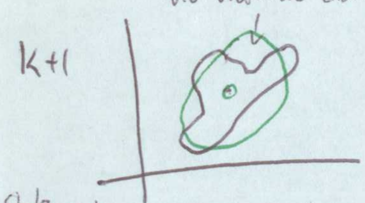
Solution
$X_{k+1|k} = F X_{k|k}$
$P_{k+1|k} = F P_{k|k} F^T + Q$

non linear: $X_{k+1} = f(x_k + \nu_k)$ nonliner Function

$Q = E\{\nu_k \cdot \nu_k^T\}$

Solution: Numerical! with Particle Filter

no normal distribution: UKF: predict as if normal distributed
↳ not analytical → aprox.



Find normal dist. that represents real dist. as close as possible.
↳ normal that has same mean and covariance matrix as the real dist.

State vector

$X = \begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot{\psi} \end{bmatrix}$

State dimension
$n_x = 5$

number of sigma Points

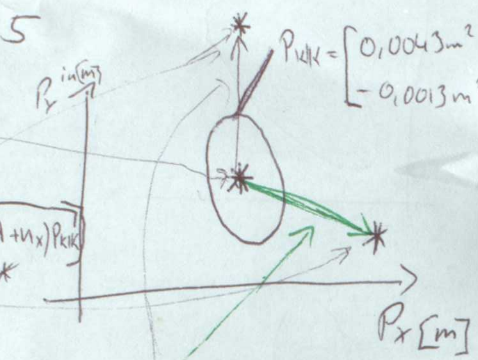
$n_\sigma = 2n_x + 1 = 11$

first point is the mean of state
2 points for Dimension

easier example with 2 Dimensions:

3

$x = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$ $n_x = 2$ $n_\sigma = 2n_x + 1 = 5$



$P_{k|k} = \begin{bmatrix} 0{,}0043m^2 & -0{,}0013m^2 \\ -0{,}0013m^2 & 0{,}0077m^2 \end{bmatrix}$

Matrix with σ-Points

$X_{k|k} = [X_{k|k} \quad X_{k|k} + \sqrt{(\lambda + n_x) P_{k|k}} \quad X_k - \sqrt{(\lambda + n_x) P_{k|k}}]$

for mean

λ: Design Parameter

$\lambda = 3 - n_x$ (Example often used)

Big λ: further from mean in relation to error ellipse

Square Root of Matrix

$A = \sqrt{P_{k|k}} \Leftarrow A^T A = P_{k|k} \quad \rightarrow A = \sqrt{P_{k|k}} = \begin{bmatrix} 0{,}00656m & 0m \\ -0{,}0191m & 0{,}0855m \end{bmatrix}$ maybe wrong

Augmentation:

$x_{a,k} = \begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot\psi \\ v_a \\ v_{\ddot\psi} \end{bmatrix}$ $n_a = 7$ $n_\sigma = 2n_a + 1 = 15$ $P_{a,k|k} = \begin{bmatrix} P_{k|k} & 0 \\ 0 & Q \end{bmatrix}$

5 state, 2 noise

Predict Sigma Points Insert sigmapoints in process model

Predicted state

$X_{a,k|k} = \begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot\psi \end{bmatrix}$ $x_{k+1} = f(x_k, v_k)$ $\rightarrow x_{k+1|k}$

Predict Mean & Covariance Mean: $x_{k+1|k} = \sum_{i=1}^{n_\sigma} w_i X_{k+1|k,i}$ weights

Covariance: $P_{k+1|k} = \sum_{i=0}^{2n_\sigma} w_i (X_{k+1|k,i} - x_{k+1|k})(X_{k+1|k,i} - x_{k+1|k})^T$

Weights: $w_i = \frac{\lambda}{\lambda + n_\sigma}, i = 0$ | $w_i = \frac{1}{2(\lambda + n_\sigma)}, i = 2 \dots n_\sigma$

4

Update Measurement Prediction

Shortcut: Reuse Sigmapoints: Transform old σ-Points into measurement space

↳ calculate mean & covariance matrix S of predicted measurement

→ Store transformed measurement σ-Points as colums in Matrix

↳ in Radar $z_{k+1|k} = \begin{bmatrix} \rho \\ \varphi \\ \dot\rho \end{bmatrix}$ 3. Dim → Matrix: 3 Rows, 15 Colums ↳ 15 σ-P $Z_{k+1|k}$

↳ Predicted σ-P

↓

Measurementmodel: $z_{k+1} = h(x_{k+1}) + w_{k+1}$

↓

Measurement σ-P

Predicted measurement mean

$z_{k+1|k} = \sum_{i=1}^{n_\sigma} w_i Z_{k+1|k,i}$

Measurement model

$z_{k+1} = h(x_{k+1}) + w_{k+1}$

Predicted measurement covariance

$S_{k+1|k} = \sum_{i=0}^{2n_\sigma} w_i (Z_{k+1|k,i} - z_{k+1|k})(Z_{k+1|k,i} - z_{k+1|k})^T + R$ → $R = E\{w_k \cdot w_k^T\}$

Measurement noise covariance

Update State

Kalman Gain $K_{k+1|k} = T_{k+1|k} S_{k+1|k}^{-1}$

State update $x_{k+1|k+1} = x_{k+1|k} + K_{k+1|k}(z_{k+1} - z_{k+1|k})$

Covariance matrix update $P_{k+1|k+1} = P_{k+1|k} - K_{k+1|k} S_{k+1|k} K_{k+1|k}^T$

New: Cross-correlation betw. σ-P in state space and measurement space

$T_{k+1|k} = \sum_{i=0}^{2n_\sigma} w_i (X_{k+1|k,i} - x_{k+1|k})(Z_{k+1|k,i} - z_{k+1|k})^T$

Parameters Radar measurement Noise $w_k = \begin{bmatrix} w_{\rho,k} \\ w_{\varphi,k} \\ w_{\dot\rho,k} \end{bmatrix}$ $R = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & \sigma_{\dot\rho}^2 \end{bmatrix}$ Measurement covariance

Sensor manual: $\sigma_\rho = 0{,}3m$

↳ $\sigma_\rho^2 = 0{,}09 m^2$

Process noise covariance:

$Q = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_{\ddot\psi}^2 \end{bmatrix}$ → guess max acceleration → $\sigma_a = \frac{a_{max}}{2} = \frac{6}{2} = 3$, $\sigma_a^2 = 9 \frac{m^2}{s}$

z.B. Urban $a_{max} \approx 6\frac{m}{s^2}$

react fast on changes?
↳ process noise ↑
smooth estimations?
↳ proc. noise ↓

Consistency

Consistency: check if Noise Parameters are good

Normalized Innovation Squared (NIS): $\varepsilon = (z_{k+1} - z_{k+1|k})^T \cdot S_{k+1|k}^{-1} \cdot (z_{k+1} - z_{k+1|k})$

Difference, Measurement, Prediction ~~S~~

Normalize

↳ Chi χ^2 - Distribution

df: degrees of freedom
↳ dim measurement space
↳ 3dim Radar

df	$\chi^2_{.950}$	...	$\chi^2_{.050}$
1			
2			7.815
3			

↑5% of all cases NIS will be higher than 7.815

→ looks great ≈5% above 95%