Filter and its Applications

Tian Chen

Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences

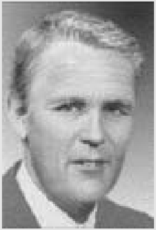
2017.05.25

Quick Overview

1. Introduction
2. Filter
   - State Space Model
   - Kalman Filter
   - Extended Kalman Filter
   - Particle Filter
3. Applications
   - Tracking
   - SLAM (Simultaneous Localization And Mapping)
   - Inertial Navigation

1. Introduction
2. Filter
   - State Space Model
   - Kalman Filter
   - Extended Kalman Filter
   - Particle Filter
3. Applications
   - Tracking
   - SLAM (Simultaneous Localization And Mapping)
   - Inertial Navigation

R. E. Kalman



Born 1930 in Hungry
Studied at MIT/Columbia
Developed filter in 1960/61

His passing not only brought about personal loss but also a sad reminder of the passing of a golden era in systems and control.

–Yu-Chi Ho

Kalman Filter

- Kalman filtering has proved useful in navigational and guidance systems, radar tracking, sonar ranging, and satellite orbit determination, to name just a few areas.
- Kalman and Bucy's original papers have generated thousands of other papers on aspects and applications of filtering. Their work has also stimulated mathematical research in such areas as numerical methods for linear algebra.

Particle Filter

- **First attempts-simulations of growing polymers**
  - Rosenbluth M N, Rosenbluth A W. Monte Carlo Calculation of the Average Extension of Molecular Chains[J]. Journal of Chemical Physics, 1955, 23(2):356-359.
- **First application in signal processing-1993**
  - Gordon N J, Salmond D J, Smith A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation[J]. IEE Proceedings F - Radar and Signal Processing, 2002, 140(2):107-113.

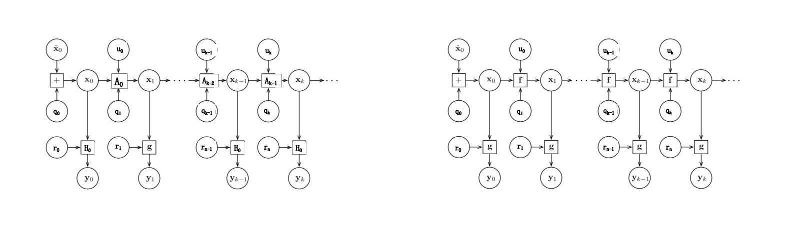
1. Introduction

2. **Filter**
   - State Space Model
   - Kalman Filter
   - Extended Kalman Filter
   - Particle Filter

3. Applications
   - Tracking
   - SLAM (Simultaneous Localization And Mapping)
   - Inertial Navigation

Linear State Space Model

process	Linear Process	Nonliear Process
motion model	$\mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k$	$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k)$
observation model	$\mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k$	$\mathbf{g}(\mathbf{x}_k, \mathbf{r}_k)$
Estimation	kalman Filter	EKF, Particle Filter



Linear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\ \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k \end{cases}$$

Linear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\ \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k \end{cases}$$

$$\begin{aligned} \text{system state:} \quad & \mathbf{x}_k \in \mathbb{R}^N \\ \text{input state:} \quad & \mathbf{u}_k \in \mathbb{R}^N \\ \text{process noise:} \quad & \mathbf{q}_k \in \mathbb{R}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \\ \text{measurement:} \quad & \mathbf{y}_k \in \mathbb{R}^M \\ \text{measurement noise:} \quad & \mathbf{r}_k \in \mathbb{R}^M \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \end{aligned}$$

Linear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\ \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k \end{cases}$$

Linear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\ \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k \end{cases}$$

$$\begin{aligned} \text{transition matrix:} \quad & \mathbf{A}_k \in \mathbb{R}^{N \times N} \\ \text{control-input matrix:} \quad & \mathbf{B}_k \in \mathbb{R}^{N \times N} \\ \text{observation matrix:} \quad & \mathbf{H}_k \in \mathbb{R}^{M \times N} \end{aligned}$$

Kalman Filter

Predict

$$\begin{aligned}
\text{Predicted state estimate} \quad & \hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\
\text{Predicted estimate covariance} \quad & \mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k \\
\text{measurement residual} \quad & \tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}
\end{aligned}$$

Kalman Filter

Predict

$$
\begin{aligned}
\text{Predicted state estimate} \quad & \hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{q}_k \\
\text{Predicted estimate covariance} \quad & \mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k \\
\text{measurement residual} \quad & \tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}
\end{aligned}
$$

Update

$$
\begin{aligned}
\text{residual covariance} \quad & \mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \\
\text{"Optimal" Kalman gain} \quad & \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1} \\
\text{Updated state estimate} \quad & \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\
\text{Updated estimate covariance} \quad & \mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}
\end{aligned}
$$

EKF-Nonlinear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k) \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{r}_k) \end{cases}$$

$$\begin{aligned} \text{transition model:} \quad & \mathbf{f} \\ \text{observation model:} \quad & \mathbf{g} \end{aligned}$$

EKF-Nonlinear State Space Model

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k) \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{r}_k) \end{cases}$$

$$\text{transition model:} \quad \mathbf{f}$$

$$\text{observation model:} \quad \mathbf{g}$$

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k) \approx \check{\mathbf{x}}_k + \mathbf{A}_{k-1}(\mathbf{x}_{k-1} - \check{\mathbf{x}}_{k-1}) + \mathbf{q}'_k$$
$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{r}_k) \approx \check{\mathbf{y}}_k + \mathbf{H}_k(\mathbf{x}_k - \check{\mathbf{x}}_k) + \mathbf{r}'_k$$

EKF

Predict

$$\text{Predicted state estimate} \quad \hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k)$$

$$\text{Predicted estimate covariance} \quad \mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}'_k$$

$$\text{measurement residual} \quad \tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{r}_k)$$

EKF

Predict

$$\begin{aligned}
\text{Predicted state estimate} \quad \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{q}_k) \\
\text{Predicted estimate covariance} \quad \mathbf{P}_{k|k-1} &= \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}'_k \\
\text{measurement residual} \quad \tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{r}_k)
\end{aligned}$$

Update

$$\begin{aligned}
\text{residual covariance} \quad \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \\
\text{"Optimal" Kalman gain} \quad \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1} \\
\text{Updated state estimate} \quad \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\
\text{Updated estimate covariance} \quad \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}
\end{aligned}$$

Particle Filter

- Sequential Monte Carlo Algorithm Based on Bayesian Criterion
- Looking for a set of discrete sample points to approximate the probability density function

Particle Filter-Monte Carlo

- Estimate integral values (p is a gaussian distribution)

$$v = \int_{-\infty}^{+\infty} x^2 p(x)\, dx$$

- Monte Carlo:
  1. Simulate M random variables

$$x^{(m)} \sim N(0, \sigma^2)$$

  2. Calculate the mean

$$v = \frac{1}{M} \sum_{m=1}^{M} (x^{(m)})^2$$

Particle Filter-Importance sampling

- Integral calculation

$$E(f(x)) = \int_{-\infty}^{+\infty} f(x)p(x)dx = \int_{-\infty}^{+\infty} f(x)\frac{p(x)}{\pi(x)}\pi(x)dx$$

- Monte Carlo:

1. Simulate M random variables $(\pi(x))$

$$x^{(m)} \sim \pi(x)$$

2. Calculate the expectation

$$E(f(x)) \approx \frac{1}{M}\sum_{m=1}^{M} f(x^{(m)}) \underbrace{\frac{p(x^{(m)})}{\pi(x^{(m)})}}_{w^{(m)}}$$

Particle Filter-Importance sampling

- Integral calculation

$$E(g_t(x_{0:t})) = \int g_t(x_{0:t}) p(x_{0:t}|y_{1:t}) dx_{0:t}$$

$$E(g_t(x_{0:t})) = \int g_t(x_{0:t}) \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} q(x_{0:t}|y_{1:t}) dx_{0:t}$$

- Monte Carlo:
  1. Simulate M random variables $(q(x))$
  2. Calculate the mean

$$E(g_t(x_{0:t})) = \frac{\frac{1}{M}\sum_{i=1}^{M} g_t(x_{0:t}^{(i)}) w_t(x_{0:t}^{(i)})}{\frac{1}{M}\sum_{i=1}^{M} w_t(x_{0:t}^{(i)})} = \sum_{i=1}^{M} g_t(x_{0:t}^{(i)}) \tilde{w}_t(x_{0:t}^{(i)})$$

$$\hat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{M} \int \tilde{w}_t^{(i)} \delta_{x_{0:t}^{(i)}}(dx_{0:t})$$

Particle Filter-Sequential Importance Sampling

$$p(x_{0:t}) = p(x_0)\prod_{j=1}^{t} p(x_j|x_{j-1}), \quad p(y_{1:t}|x_{0:t}) = \prod_{j=1}^{t} p(y_j|x_j)$$

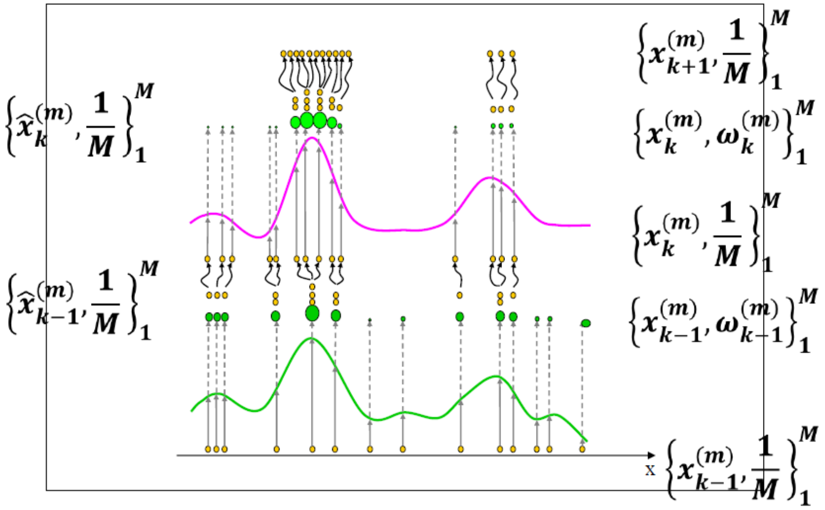
1. Predicted state estimate $x_k^{(m)} = p(x_k|x_{k-1})$

2a. Weight computation $w_k^{*(m)} = w_{k-1}^{*(m)} p(y_k|x_k^{(m)})$

2b. Weight normalization $w_k^{(m)} = \dfrac{w_k^{*(m)}}{\sum_{m=1}^{M} w_k^{*(m)}}$

3. Estimation $E(g(x_k|y_{1:k})) = \sum_{m=1}^{M} g(x_k^{(m)}) w_k^{(m)}$

Particle Filter-Resampling



Particle Filter

Advantages

- Global approximation.
- Wide adaptability.

Particle Filter

Advantages

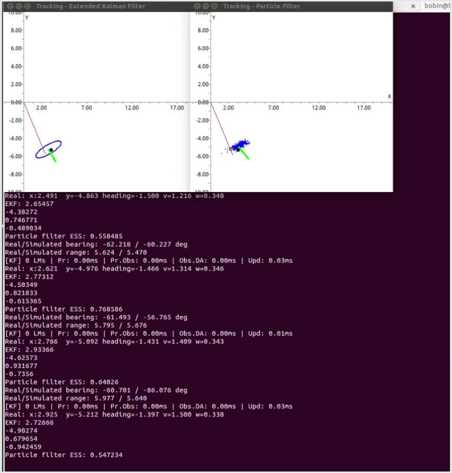
- Global approximation.
- Wide adaptability.

Disadvantages

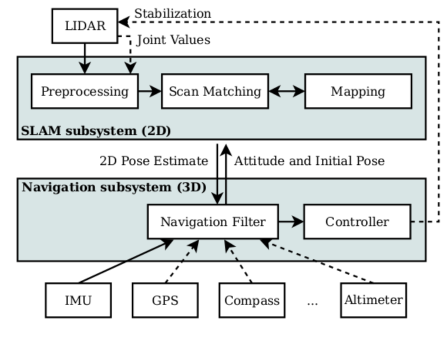
- Computational requirements much higher than of the Kalman filters.
- Problems with nearly noise-free models, especially with accurate dynamic models.
- The selection of particle numbers and important density function.

Tracking

red-observationbearing green-velocty blue(left)-EKF covariance blue(right)-Particles Comparison: Accuracy performance



Hector SLAM



- Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation[C]//Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on. IEEE, 2011: 155-160.

Hector SLAM

3D state

$$\mathbf{x} = \begin{bmatrix} \Omega^T & \mathbf{p}^T & \mathbf{v}^T \end{bmatrix}$$

where

$$\begin{aligned} \Omega &= [\phi, \theta, \varphi] && \text{roll, pitch and yaw Euler angles} \\ \mathbf{p} &= [\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z] && \text{posotion} \\ \mathbf{v} &= [\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z] && \text{velocity} \end{aligned}$$

Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation[C]//Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on. IEEE, 2011: 155-160.

Hector SLAM

Dynamic system

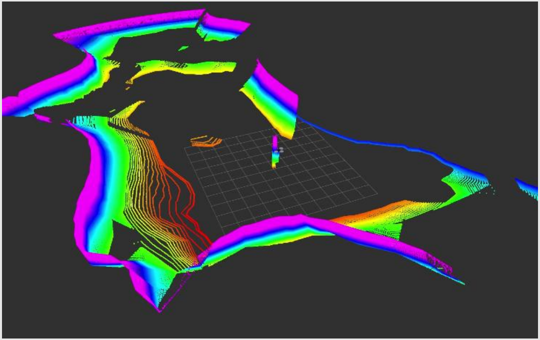
$$\dot{\Omega} = \mathbf{E}_\omega \cdot \omega$$
$$\dot{\mathbf{p}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{R}_\omega \cdot \mathbf{a} + \mathbf{g}$$

where

$\mathbf{R}_\omega$	Rotation matrix from Sensor to world
$\mathbf{E}_\omega$	maps angular rates to the derivatives of the Euler angles
$\mathbf{g}$	constant gravity vector

- Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation[C]//Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on. IEEE, 2011: 155-160.

Hector SLAM



- Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation[C]//Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on. IEEE, 2011: 155-160.

Thank you