“Fourier-Hermite Kalman Filter”

Juha Sarmavuori and Simo Sarkka (2011)

Shifra Abittan

ECE416 Adaptive Filters
Professor Fontaine
Summer 2019

Background: Kalman Filter

- Used to estimate a dynamic state from noisy measurements
- Discrete-time state space model used to represent
  - Internal dynamics
  - Measurement process

$$
\begin{aligned}
\text{Process equation:} \quad & x[n+1] &= \quad & Ax[n] + v[n] \\
\text{Observation equation:} \quad & y[n] &= \quad & Cx[n] + w[n]
\end{aligned}
$$

Background: Comparing Filters

- Kalman filter
    1. **Linear** internal dynamics and measurement process
    2. **Gaussian** noise
- Extended Kalman filter (EKF)
    1. **Nonlinear** internal dynamics and measurement process
    2. Approximate nonlinearity with first-order **Taylor series** approximation (Jacobian matrix)
    3. Issues:
        - (1) Nonlinearity not well approximated with taylor series, can use higher order
        - (2) Requires a smooth and differentiable nonlinear function f and h

Background: Comparing Filters

- Statistically Linearized filter (SLF)
    1. **Nonlinear** internal dynamics and measurement process
    2. Approximate nonlinearity by MMSE, Gaussian describing function
    3. Advantages:
        - Linearization is based on range of values/expectation. Global approximation (Taylor = local)
        - No continuity/differentiability requirement, don’t need to compute Jacobians.
    4. Disadvantage: Expectations must be able to be computed in closed form
- Sigma Point Filters
    1. Numerical approximation of SLF
    2. Computationally slow
    3. Example: Unscented Kalman Filter
    4. Simulate a few versions and taking weighted mean
    5. Not attempting to approximate the nonlinearity at all
    6. No closed form required

Fourier-Hermite Series

- Hermite polynomials

$$H_n(x) = (-1)^n \, e^{x^2/2} \, \frac{d^n}{dx^n} e^{-x^2/2}, \quad n = 0, 1, \ldots$$

$H_0(x) = 1$
$H_1(x) = x$
$H_2(x) = x^2 - 1$
$H_3(x) = x^3 - 3x$
$H_4(x) = x^4 - 6x^2 + 3$
…

- Orthogonal when inner product is defined by the expectation with respect to the standard normal distribution N(0,1) $\langle f, g \rangle = \mathrm{E}[f(x)g(x)]$

Fourier-Hermite Series

- A function g(x) has a Fourier-Hermite series representation if $E[g(x)^2]$ is finite

$$
\begin{aligned}
g(x) &= \sum_{k=0}^{\infty} \frac{1}{k!} \mathrm{E}[g(x) H_k(x)] H_k(x) \\
&= \sum_{k=0}^{\infty} \frac{1}{k!} \mathrm{E}[g^{(k)}(x)] H_k(x)
\end{aligned}
$$

- If the distribution is of the form N(mu,sigma^2), the above equations are modified. Replace x in H(x) with (x-mu)/sigma and multiply the terms in the second equation by simga^k

Fourier-Hermite Series

- Because of orthogonality, Parseval's theorem is satisfied
- Therefore, the expectation of a function squared is equivalent to summing the coefficients squared

$$E[g(x)*g(x)] = \sum^{\infty}_{k=0}(1/k!)*c_k$$

Fourier-Hermite Series

- Can be generalized to higher dimensions, i.e. x is a vector
- Multidimensional Hermite polynomials are vectors of polynomials
  - Elements are products of 1D Hermite polynomials

Fourier-Hermite Series Filter

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_k \\
\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k
\end{aligned}
$$

- *Prediction*

$$
\begin{aligned}
\mathbf{m}_k^- &= \mathrm{E}[\mathbf{f}(\mathbf{x}_{k-1})] \\
\mathbf{P}_k^- &= \mathrm{E}[(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T] + \mathbf{Q}_k
\end{aligned} \tag{2}
$$

where the expectations $\mathrm{E}[\cdot]$ are with respect to $\mathbf{x}_{k-1} \sim \mathrm{N}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1})$.

- *Update*

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \mathrm{E}[\mathbf{h}(\mathbf{x}_k)] \\
\mathbf{S}_k &= \mathrm{E}[(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T] + \mathbf{R}_k \\
\text{Gain} \longleftarrow \mathbf{K}_k &= \mathrm{E}[(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T]\, \mathbf{S}_k^{-1} \\
\mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\, (\underline{\mathbf{y}_k - \boldsymbol{\mu}_k}) \longrightarrow \text{Innovations} \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\, \mathbf{S}_k\, \mathbf{K}_k^T
\end{aligned} \tag{3}
$$

where the expectations $\mathrm{E}[\cdot]$ are with respect to $\mathbf{x}_k \sim \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)$.

Fourier-Hermite Series Filter

There are three different types of expectations that include the nonlinear function to be solved:

(A) Mean: $E[g(x)]$

(B) Covariance: $E[(g(x) - mean(g))(g(x) - mean(g))^T]$

(C) Cross covariance: $E[(x - mean(x))(g(x) - mean(g))^T]$

Fourier-Hermite Series Filter

Theorem 1. If the Fourier-Hermite series representation for a function g(x) exists,

(A) The **expectation** of the function is the **zeroth** coefficient. $E[g(x)] = c_0$

- Proof: When k=0, the Fourier-Hermite series term reduces to the above trivially because $H_0(x) = 1$

(B) The **covariance** of the function is the **sum** of the Fourier-Hermite **coefficients squared**. $Cov[g(x)*g(x)] = \sum^{\infty}_{k=0}(1/k!)*c_k$

- Proof: Parseval

(C) The **cross-covariance** between the random variable and the function is the variance of the random variable times the **first** coefficient. $Cov[x,g(x)] = sigma^2*c_1$

- Proof: When k=1, the Fourier-Hermite series term reduces to the above trivially because $H_1(x) = x$

Fourier-Hermite Series Filter

Recall that $c_k = a_k$*sigma^k where c_k = E[g(x)*H((x-mu)/sigma)] and $a_k = E[g^{(k)}(x)]$.

Theorem 2. Assume that the following integral, i.e. E[g(x)]=c_0, can be computed in closed form:

$$\hat{\mathbf{g}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int_{\mathbb{R}^n} \mathbf{g}(\mathbf{x}) \mathrm{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{x}.$$

Then, the a_k term can also be expressed as d^k(E[g(x)])/dx^k. Order of expectation and differentiation does not matter. Therefore, instead of computing the derivative of g, instead take the derivative of the expectation of g, i.e. the zeroth coefficient.

Ex: c2 = a2*sigma = second derivative of the above integral*sigma;

Fourier-Hermite Series Filter

- Using Theorem 1 and Theorem 2, the Fourier-Hermite series based approximation for a nonlinear function, of **any order**, can be constructed in terms of just the integral

$$\hat{\mathbf{g}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int_{\mathbb{R}^n} \mathbf{g}(\mathbf{x}) \mathrm{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{x}.$$

  and its derivatives.

- If the closed form integration is not possible, can use a table or use a powerful computer algebra system (CAS) to solve.
- This method is significantly simpler than EKF

Example: FHSF up to third order

1. Compute mean functions = c_o = integral

$$\hat{\mathbf{f}}(\mathbf{m}, \mathbf{P}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\, \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\, \mathrm{d}\mathbf{x}$$

$$\hat{\mathbf{h}}(\mathbf{m}, \mathbf{P}) = \int_{\mathbb{R}^n} \mathbf{h}(\mathbf{x})\, \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\, \mathrm{d}\mathbf{x}.$$

2. Compute Jacobians = first derivatives = c_1

$$[\hat{\mathbf{F}}_{\mathbf{m}}(\mathbf{m}, \mathbf{P})]_{i,i'} = \frac{\partial}{\partial m_{i'}} \hat{f}_i(\mathbf{m}, \mathbf{P})$$

$$[\hat{\mathbf{H}}_{\mathbf{m}}(\mathbf{m}, \mathbf{P})]_{i,i'} = \frac{\partial}{\partial m_{i'}} \hat{h}_i(\mathbf{m}, \mathbf{P}).$$

Example: FHSF up to third order

3. Compute higher derivatives

$$\hat{\mathbf{f}}''_{i,j}(\mathbf{m}, \mathbf{P}) = \frac{\partial^2}{\partial m_i \partial m_j} \hat{\mathbf{f}}(\mathbf{m}, \mathbf{P})$$

$$\hat{\mathbf{f}}'''_{i,j,u}(\mathbf{m}, \mathbf{P}) = \frac{\partial^3}{\partial m_i \partial m_j \partial m_u} \hat{\mathbf{f}}(\mathbf{m}, \mathbf{P})$$

$$\vdots$$

$$\hat{\mathbf{h}}''_{i,j}(\mathbf{m}, \mathbf{P}) = \frac{\partial^2}{\partial m_i \partial m_j} \hat{\mathbf{h}}(\mathbf{m}, \mathbf{P})$$

$$\hat{\mathbf{h}}'''_{i,j,u}(\mathbf{m}, \mathbf{P}) = \frac{\partial^3}{\partial m_i \partial m_j \partial m_u} \hat{\mathbf{h}}(\mathbf{m}, \mathbf{P})$$

$$\vdots$$

Example: FHSF up to third order

4. Plugging the FHS approximations into the gaussian filter:

- *Prediction*

$$\begin{aligned}\mathbf{m}_k^- &= \mathrm{E}[\mathbf{f}(\mathbf{x}_{k-1})]\\ \mathbf{P}_k^- &= \mathrm{E}[(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T] + \mathbf{Q}_k \end{aligned} \quad (2)$$

where the expectations $\mathrm{E}[\cdot]$ are with respect to $\mathbf{x}_{k-1} \sim \mathrm{N}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1})$.

- *Update*

$$\begin{aligned}\boldsymbol{\mu}_k &= \mathrm{E}[\mathbf{h}(\mathbf{x}_k)]\\ \mathbf{S}_k &= \mathrm{E}[(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T] + \mathbf{R}_k\\ \mathbf{K}_k &= \mathrm{E}[(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T]\,\mathbf{S}_k^{-1}\\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\,(\mathbf{y}_k - \boldsymbol{\mu}_k)\\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}_k^T \end{aligned} \quad (3)$$

where the expectations $\mathrm{E}[\cdot]$ are with respect to $\mathbf{x}_k \sim \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)$.

$$\begin{aligned}\mathbf{m}_k^- &= \hat{\mathbf{f}}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1})\\ \mathbf{P}_k^- &= \mathbf{Q}_k + \hat{\mathbf{F}}_{\mathbf{m}}\,\mathbf{P}_{k-1}\,\hat{\mathbf{F}}_{\mathbf{m}}^T + \frac{1}{2!}\sum_{\substack{i,j\\u,v}} \hat{\mathbf{f}}''_{i,u}\,P_{i,j}P_{u,v}\,\hat{\mathbf{f}}''^T_{j,v}\\ &\quad + \frac{1}{3!}\sum_{\substack{i,j\\u,v\\a,b}} \hat{\mathbf{f}}'''_{i,u,a}\,P_{i,j}P_{u,v}P_{a,b}\,\hat{\mathbf{f}}'''^T_{j,v,b} + \cdots \end{aligned} \quad (17)$$

where we used shorthand notation $P_{i,j} = [\mathbf{P}_{k-1}]_{i,j}$ and the derivatives are evaluated at $\mathbf{m}_{k-1}$ and $\mathbf{P}_{k-1}$.

Update

$$\begin{aligned}\mathbf{S}_k &= \mathbf{R}_k + \hat{\mathbf{H}}_{\mathbf{m}}\,\mathbf{P}_k^-\,\hat{\mathbf{H}}_{\mathbf{m}}^T + \frac{1}{2!}\sum_{\substack{i,j\\u,v}} \hat{\mathbf{h}}''_{i,u}\,P_{i,j}P_{u,v}\,\hat{\mathbf{h}}''^T_{j,v}\\ &\quad + \frac{1}{3!}\sum_{\substack{i,j\\u,v\\a,b}} \hat{\mathbf{h}}'''_{i,u,a}\,P_{i,j}P_{u,v}P_{a,b}\,\hat{\mathbf{h}}'''^T_{j,v,b} + \cdots \end{aligned} \quad (18)$$

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^-\,\hat{\mathbf{H}}_{\mathbf{m}}^T\,\mathbf{S}_k^{-1}\\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\,(\mathbf{y}_k - \hat{\mathbf{h}}(\mathbf{m}_k^-, \mathbf{P}_k^-))\\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}_k^T \end{aligned} \quad (19)$$

where $P_{i,j} = [\mathbf{P}_k^-]_{i,j}$ and the derivatives are evaluated at $\mathbf{m}_k^-$ and $\mathbf{P}_k^-$.

Discussion

- First order truncation of the FH series → statistically linearized filter (SLF)
- Primary advantage: Because Hermite polynomials are orthogonal, the FHKF is easy to compute for any order. The Taylor series used in EKF does not have this property.
- Best possible polynomial approximation in the mean squared error sense
- Fourier Hermite series expansions can be derived for non-differentiable functions

Numerical Example: Simulated pendulum

Internal dynamic model:

$$x_{k,1} = x_{k-1,1} + x_{k-1,2}\, \Delta t$$
$$x_{k,2} = x_{k-1,2} - g\, \sin(x_{k-1,1})\, \Delta t + q_{k-1}$$

Where sampling period = delta t = 1/1000; g = 9.81; q is N(0,Q); Q = delta t/100

Measurement model:

$$y_k = h(x_{k,1}) + r_k, \quad r_k \sim N(0, R)$$

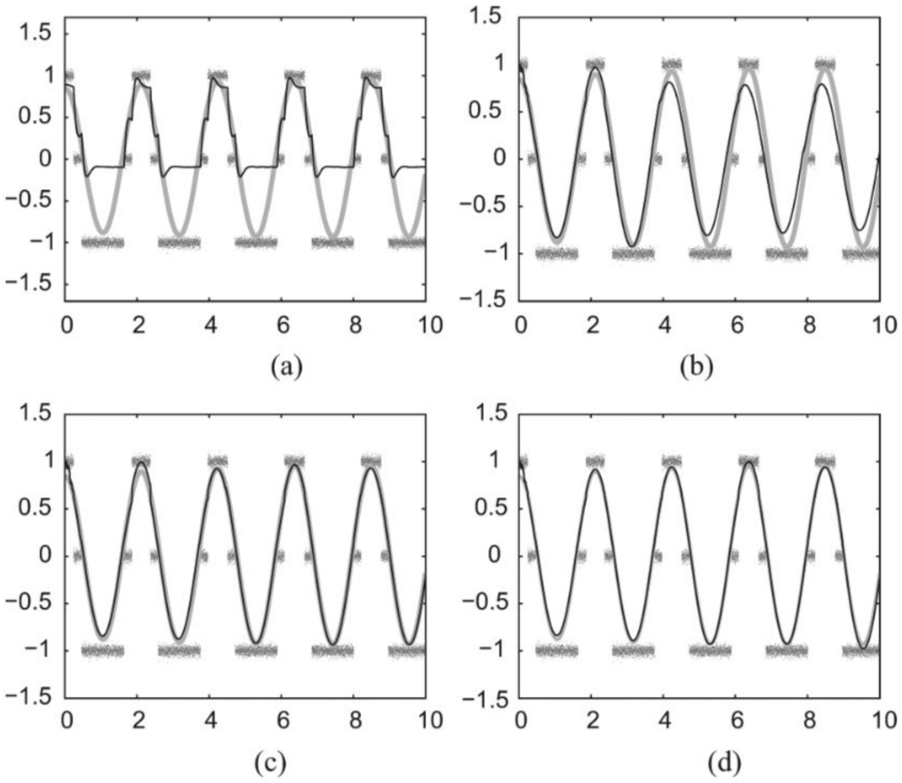
$$h(x) = \begin{cases} -1, & \text{if } x < \frac{-a}{2+b} \\ 0, & \text{if } \frac{-a}{2+b} < x < \frac{a}{2+b} \\ 1, & \text{if } x > \frac{a}{2+b} \end{cases}$$

Where R = 1/1000; a = 0.5; b = 0.4

Numerical Example: Simulated pendulum

- Implementing EKF is extremely difficult because of the piecewise constant measurement model function
- For FHKF, the function itself does not have to be differentiable. The expectation of the function will be differentiable even if the function itself is not.
- In order to compare to EKF, h(x) can be approximated as (x-b)/a

Numerical Example: Results



Model	RMSE
(a) EKF	0.41
(b) first order FHKF	0.16
(c) second order FHKF	0.04
(d) third order FHKF	0.03

Sigma point methods: UKF, Cubature KF = second order FHKF