Nonlinear Filtering

2015 ACC Workshop on Uncertainty Analysis & Estimation

Raktim Bhattacharya

Intelligent Systems Research Laboratory
Aerospace Engineering, Texas A&M University.

isrlab.github.io

Nonlinear Filtering

With Polynomial Chaos

Raktim Bhattacharya
Aerospace Engineering, Texas A&M University
isrlab.tamu.edu

Nonlinear Filtering with PC

Problem Setup

Dynamics: $\dot{x} = f(x, \Delta)$

Sensor Model: $\tilde{y} = h(x) + \nu$

where ν is measurement noise with

$$\mathsf{E}\left[\nu\right] = 0, \text{ and } \mathsf{E}\left[\nu\nu^T\right] = R.$$

Measurements available at times $t_k, t_{k+1}, \cdots$

Parametric uncertainty

$$\Delta := \begin{pmatrix} \Delta_{x_0} \\ \Delta_\rho \end{pmatrix}$$

Dynamics transformed to

$$\dot{x}_{pc} = F_{pc}(x_{pc})$$

Initial Condition at time t_k

$$x_{pc_i}(t_k) = \int_{\mathcal{D}_\Delta} \Delta_{x_0} \phi_i(\Delta) p(k, \Delta) d\Delta$$

Filtering with Higher Order Moment Updates

1. Propagation

$$\boldsymbol{x}_{pc}(t_{k+1}) = \boldsymbol{x}_{pc}(t_k) + \int_{t_k}^{t_{k=1}} \boldsymbol{F}_{pc}(\boldsymbol{x}_{pc}(\tau))d\tau$$

2. Get Prior Moments from $x_{pc}(t_{k+1})$

recall $\boldsymbol{x}_{pc} := \textbf{vec}\,(\boldsymbol{X})$ and $\boldsymbol{X} := [\boldsymbol{x}_0\ \boldsymbol{x}_1\ \cdots\ \boldsymbol{x}_N]$

$$\begin{aligned}
\boldsymbol{M}_i^{1^-} &= \boldsymbol{X}_{i0} \\
\boldsymbol{M}_{ij}^{2^-} &= \sum_p \sum_q \boldsymbol{X}_{ip}\boldsymbol{X}_{jq} \left\langle \phi_p \phi_q \right\rangle \\
\boldsymbol{M}_{ijk}^{3^-} &= \sum_p \sum_q \sum_r \boldsymbol{X}_{ip}\boldsymbol{X}_{jq}\boldsymbol{X}_{kr} \left\langle \phi_p \phi_q \phi_r \right\rangle \quad \text{and so on ...}
\end{aligned}$$

Inner products are with respect to $p(t_k, \boldsymbol{\Delta})$

Filtering with Higher Order Moment Updates

3. Update

- Incorporate measurements $\tilde{\boldsymbol{y}} := \tilde{\boldsymbol{y}}(t_{k+1})$ and prior moments to get posterior estimates
- Consider prior state estimate to be $\hat{\boldsymbol{x}}^- := \mathbf{E}[\boldsymbol{x}] = \boldsymbol{M}^{1^-}$
- Let

$$\boldsymbol{v} := \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}}^- = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\nu} - \boldsymbol{h}(\hat{\boldsymbol{x}}^-)$$

- Use linear gain $\boldsymbol{K}$ to update moments as

$$\boldsymbol{K} = \boldsymbol{P}^{\boldsymbol{xv}}\left(\boldsymbol{P}^{\boldsymbol{vv}}\right)^{-1}, P_{ij}^{\boldsymbol{xv}} = \mathbf{E}\left[\boldsymbol{x}_i \boldsymbol{v}_j^T\right], P_{ij}^{\boldsymbol{vv}} = \mathbf{E}\left[\boldsymbol{v}_i \boldsymbol{v}_j^T\right]$$

$$\boldsymbol{M}^{1^+} = \boldsymbol{M}^{1^-} + \boldsymbol{K}\boldsymbol{v}$$

$$\boldsymbol{M}^{2^+} = \boldsymbol{M}^{2^-} + \boldsymbol{K}\boldsymbol{P}^{\boldsymbol{vv}}\boldsymbol{K}^T$$

$$\boldsymbol{M}^{3^+} = \boldsymbol{M}^{3^-} + 3\boldsymbol{K}^2\boldsymbol{P}^{\boldsymbol{xvv}} - 3\boldsymbol{K}\boldsymbol{P}^{\boldsymbol{xxv}} - \boldsymbol{K}^3\boldsymbol{P}^{\boldsymbol{vvv}}$$

Filtering with Higher Order Moment Updates

4. Estimation of Posterior PDF

$$\max_{p^{k+1}(\boldsymbol{\Delta})} \; -\int_{\mathcal{D}_{\boldsymbol{\Delta}}} p^{k+1}(\boldsymbol{\Delta}) \log(p^{k+1}(\boldsymbol{\Delta}))\, d\boldsymbol{\Delta},$$

subject to

$$\int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{\Delta} p^{k+1}(\boldsymbol{\Delta})\, d\boldsymbol{\Delta} = \boldsymbol{M}^{1+} \qquad \int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{Q}_2(\boldsymbol{\Delta}) p^{k+1}(\boldsymbol{\Delta})\, d\boldsymbol{\Delta} = \boldsymbol{M}^{2+}$$

$$\int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{Q}_3(\boldsymbol{\Delta}) p^{k+1}(\boldsymbol{\Delta})\, d\boldsymbol{\Delta} = \boldsymbol{M}^{3+} \qquad \int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{Q}_4(\boldsymbol{\Delta}) p^{k+1}(\boldsymbol{\Delta})\, d\boldsymbol{\Delta} = \boldsymbol{M}^{4+}$$

Approximate

$$p^{k+1}(\boldsymbol{\Delta}) = \sum_{i}^{M} \alpha_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

with

$$\int_{\mathcal{D}_{\boldsymbol{\Delta}}} p^{k+1}(\boldsymbol{\Delta}) d\boldsymbol{\Delta} = 1 \Rightarrow \sum_{i}^{M} \alpha_i = 1, \qquad p^{k+1}(\boldsymbol{\Delta}) \geq 0 \Rightarrow \alpha_i \geq 0.$$

Filtering with Higher Order Moment Updates

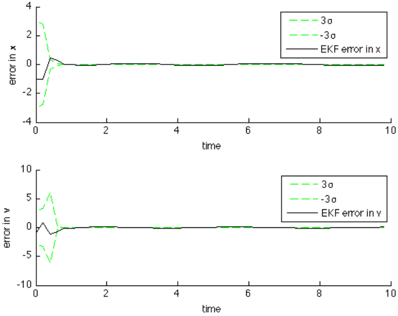
Example

- Classical duffing oscillator
- Two state system, $\boldsymbol{x} = [x_1, x_2]^T$,
- Dynamics

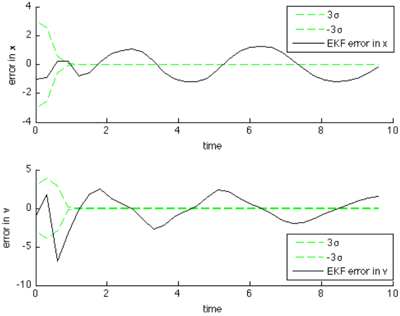
$$\dot{x}_1 = x_2,\ \dot{x}_2 = -x_1 - \frac{1}{4}x_2 - x_1^3.$$

- Uncertainty in $\boldsymbol{x}_0 \sim \mathcal{N}([1,1], \mathsf{diag}(1,1))$
- Simulation $\boldsymbol{x}_0 = [2,2]^T$
- Scalar measurement model $\tilde{y} = \boldsymbol{x}^T\boldsymbol{x} + \nu$,
- $\mathbf{E}[\nu] = 0$ and $\mathbf{E}\left[\nu\nu^T\right] = 0.006$.

Results



(a) EKF based estimator with 0.2s update.



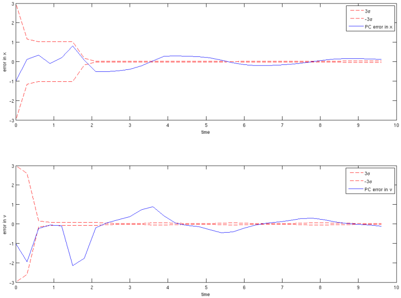
(b) EKF based estimator with 0.3s update.

Figure: Performance of EKF estimators.

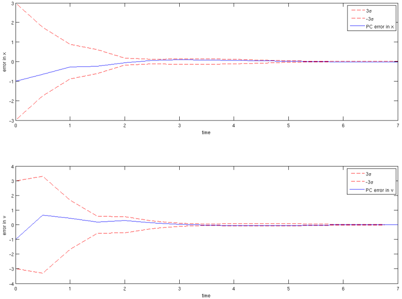
If data rate is high, no need for nonlinear estimators!!

EKF works with faster updates

Otherwise we need nonlinear non Gaussian algorithms



(a) gPC based estimator with first two moments updated every 0.3s.



(b) gPC based estimator with first three moments updated every 0.5s.

Figure: Performance of gPC estimators.

If data rate is low, we need nonlinear estimators, with higher order updates!

Publications

1. P. Dutta, R. Bhattacharya, *Nonlinear Estimation of Hypersonic Flight Using Polynomial Chaos*, AIAA GNC, 2010.
2. P. Dutta, R. Bhattacharya, *Nonlinear Estimation with Polynomial Chaos and Higher Order Moment Updates*, IEEE ACC 2010.
3. P. Dutta, R. Bhattacharya, *Nonlinear Estimation of Hypersonic State Trajectories in Bayesian Framework with Polynomial Chaos*, Journal of Guidance, Control, and Dynamics, vol.33 no.6 (1765-1778), 2011.

Nonlinear Filtering

With Frobenius-Perron Operator

Raktim Bhattacharya

Aerospace Engineering, Texas A&M University

uq.tamu.edu

Frobenius-Perron Operator

Given dynamics $\dot{\boldsymbol{x}} = \boldsymbol{F}(t, \boldsymbol{x})$,

- x is augmented state variable – captures state and system parameters (including those from KL expansion)
- $p(t_0, \boldsymbol{x})$ as the initial state density function.

Evolution of density

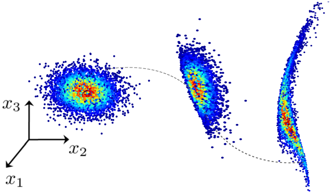
$$p(t, \boldsymbol{x}) := \mathcal{P}_t \, p(t_0, \boldsymbol{x}).$$

$\mathcal{P}_t$ is defined by

$$\frac{\partial p}{\partial t} + \boldsymbol{\nabla} \cdot (p\boldsymbol{F}) = 0$$

Equations

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(t, \boldsymbol{x})$$
$$\dot{p} = -p \boldsymbol{\nabla} \cdot \boldsymbol{F}$$



Assumptions

- Measurements are available at times $t_1, \cdots, t_{k-1}, t_k, t_{k+1}, \cdots$
- $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are state and measurement at t_k
- Measurement model
$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\nu}$$
- $\mathsf{E}\left[\boldsymbol{\nu}\right] = \mathbf{0}$, $\mathsf{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^T\right] = \boldsymbol{R}$
- $p_k(\cdot) := p(t_k, \cdot)$
- $p_k^-(\cdot)$ is prior at t_k
- $p_k^+(\cdot)$ is posterior at t_k
- $\mathcal{D}_{\boldsymbol{x}}$ is domain of state augmented

A Particle Filter Based Algorithm

1. Initialize

- Domain $\mathcal{D}_{\boldsymbol{x}}$ is sampled according to $p_0(\boldsymbol{x})$
- $\boldsymbol{x}_{0,i}$ samples of r.v. $\boldsymbol{x}_0$
- $p_{0,i} := p_0(\boldsymbol{x}_{0,i})$
- Recursively apply steps $2, \cdots, 6$ for $k = 1, \cdots$

A Particle Filter Based Algorithm (contd.)

2. Propagate

$$\begin{pmatrix} \boldsymbol{x}_{k,i} \\ p_{k,i}^{-} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{k-1,i} \\ p_{k-1,i} \end{pmatrix} + \int_{t_{k-1}}^{t_k} \begin{pmatrix} \boldsymbol{F} \\ -p\boldsymbol{\nabla} \cdot \boldsymbol{F} \end{pmatrix} dt$$

$p_{k,i}^{-}$ because it is prior state PDF

A Particle Filter Based Algorithm (contd.)

3. **Determine likelihood function** $p(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}_k = \boldsymbol{x}_{k,i})$
   - for each grid point i,
   - using Gaussian measurement noise and sensor model

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\nu}$$

   - $|R|$ is the determinant of the covariance matrix of measurement noise

$$l(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}_k = \boldsymbol{x}_{k,i}) = \frac{1}{\sqrt{(2\pi)^m|R|}} e^{-0.5(\tilde{\boldsymbol{y}}_k - \boldsymbol{h}(\boldsymbol{x}_{k,i}))^T R^{-1} (\tilde{\boldsymbol{y}}_k - \boldsymbol{h}(\boldsymbol{x}_{k,i}))},$$

4. **Update:** Get Posterior

$$p^{+}_{k,i} := p_k(\boldsymbol{x}_k = \boldsymbol{x}_{k,i}|\tilde{\boldsymbol{y}}_k) = \frac{l(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}_k = \boldsymbol{x}_{k,i}) p^{-}_k(\boldsymbol{x}_k = \boldsymbol{x}_{k,i})}{\sum_{j=1}^{N} l(\tilde{\boldsymbol{y}}_k|\boldsymbol{x}_k = \boldsymbol{x}_{k,j}) p^{-}_k(\boldsymbol{x}_k = \boldsymbol{x}_{k,j})}$$

A Particle Filter Based Algorithm (contd.)

5. Get State Estimate

a Maximum-Likelihood Estimate: Maximize the probability that $\boldsymbol{x}_{k,i} = \hat{\boldsymbol{x}}_k$

$$\hat{\boldsymbol{x}}_k = \textbf{ mode } p_k^+(\boldsymbol{x}_{k,i})$$

b Minimum-Variance Estimate: The estimate is the **mean** of $p_k^+(\boldsymbol{x}_{k,i})$

$$\hat{\boldsymbol{x}}_k = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{N} ||\boldsymbol{x} - \boldsymbol{x}_{k,i}||^2 \, p_k^+(\boldsymbol{x}_{k,i}) = \sum_{i=1}^{N} \boldsymbol{x}_{k,i} \, p_k^+(\boldsymbol{x}_{k,i})$$

c Minimum-Error Estimate: Minimize maximum $|\boldsymbol{x} - \boldsymbol{x}_{k,i}|$

$$\hat{\boldsymbol{x}} = \textbf{ median } p_k^+(\boldsymbol{x}_{k,i})$$

All same for Gaussian $p(t_k, \boldsymbol{x})$

A Particle Filter Based Algorithm (contd.)

6. Resample

- Detect degeneracy from $p_k^+(\boldsymbol{x}_{k,i})$

$$p_k^+(\boldsymbol{x}_{k,i}) < \epsilon \Rightarrow \boldsymbol{x}_{k,i} \text{ is degenerate}$$

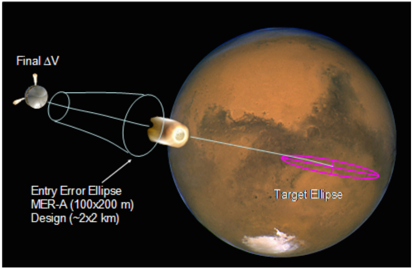
- Use existing methods for resampling from the new distribution $p_k^+(\boldsymbol{x}_{k,i})$.
  - *Importance sampling* *moderate size problems*
  - *Resampling – simple random, multinomial, stratified, systematic*

Qualitatively, since histogram techniques are not used in determining density functions, this method is less sensitive to the issue of degeneracy.

Example

3 DOF Vinh's Equation Models motion of spacecraft during planetary entry

$$\dot{h} = V\sin(\gamma)$$
$$\dot{V} = -\frac{\rho R_0}{2B_c}V^2 - \frac{gR_0}{v_c^2}\sin(\gamma)$$
$$\dot{\gamma} = \frac{\rho R_0}{2B_c}\frac{C_L}{C_D}V + \frac{gR_0}{v_c^2}\cos(\gamma)\left(\frac{V}{R_0+h} - \frac{1}{V}\right)$$



R_0 – radius of Mars

ρ – atmospheric density

v_c – escape velocity

$\frac{C_L}{C_D}$ – lift over drag

B_c – ballistic coefficient

h – height

V – velocity

γ – flight path angle

Measurement Model

$$\tilde{\boldsymbol{y}} = \left[\bar{q} = \frac{1}{2}\rho V^2, Q = k\rho^{\frac{1}{2}}V^{3.15}, \gamma\right]$$
$$\mathbf{E}\left[\boldsymbol{\nu}\right] = \mathbf{0}_{3\times 1}, \mathbf{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^T\right] = 6\times 10^{-5}\boldsymbol{I}_3 \text{ scaled}$$

Gaussian initial condition uncertainty

$$\boldsymbol{\mu}_0 = [54\text{ km}, 2.4\text{ km/s}, -9^\circ]^T$$
$$\boldsymbol{\Sigma} = \textbf{diag}[5.4\text{ km}, 240\text{ km/s}, -0.9^\circ]$$

Example (contd.)

- Compared with generic particle filter and Bootstrap filter
- All 3 perform equally well – FP requires much less number of samples
  - *Particle Filter:* 25000 *samples*
  - *Bootstrap Filter:* 20000 *samples*
  - *Frobenius-Perron Filter:* 7000 *samples*

Generic Particle filter	Bootstrap filter	FP operator based filter
207.96 s	168.06 s	57.42 s

Table: Computational time for each filter

Details

1. P. Dutta and R. Bhattacharya, *Hypersonic State Estimation Using Frobenius-Perron Operator*, AIAA Journal of Guidance, Control, and Dynamics, Volume 34, Number 2, 2011.