Controlled Interacting Particle Systems for Nonlinear Filtering

SIAM UQ 2018
Orange County, CA, Apr 16-19, 2018

Prashant G. Mehta[†]

Joint work with Amirhossein Taghvaei[†] and Sean Meyn[+]

[†]Coordinated Science Laboratory
Department of Mechanical Science and Engg., U. Illinois
[+]Department of Electrical and Computer Engg., U. Florida

April 17, 2018

ILLINOIS

Feedback Particle Filter

A numerical algorithm for nonlinear filtering

Problem:

Signal model: $\mathrm{d}X_t = a(X_t)\,\mathrm{d}t + \mathrm{d}B_t \qquad X_0 \sim p_0^*$

Observation model: $\mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t$

Posterior distribution of X_t given $\mathcal{Z}_t := \sigma(Z_s : 0 \leq s \leq t)$?

Yang, **M.**, Meyn. Feedback particle filter. *IEEE Trans. Aut. Control* (2013)
Yang, Laugesen, **M.**, Meyn. Multivariable Feedback particle filter. *Automatica* (2016)

Problem:

Signal model: $\mathrm{d}X_t = a(X_t)\,\mathrm{d}t + \mathrm{d}B_t \qquad X_0 \sim p_0^*$

Observation model: $\mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t$

Posterior distribution of X_t given $\mathcal{Z}_t := \sigma(Z_s : 0 \le s \le t)$?

Solution: Feedback particle filter

$$\mathsf{P}(X_t|\mathcal{Z}_t) \approx \text{empirical dist. of } \{X_t^1, \ldots, X_t^N\}$$

Mean-fld FPF: $(N=\infty)$

$$\mathrm{d}X_t^i = \underbrace{a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i}_{\text{Propagation}} + \underbrace{\mathsf{K}_t(X_t^i) \circ \left(\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t\right)}_{\text{Update}}, \quad X_0^i \sim p_0^*$$

Yang, **M.**, Meyn. Feedback particle filter. *IEEE Trans. Aut. Control* (2013)
Yang, Laugesen, **M.**, Meyn. Multivariable Feedback particle filter. *Automatica* (2016)

Feedback Particle Filter
A numerical algorithm for nonlinear filtering

Problem:

$$\textbf{Signal model:} \quad \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + \mathrm{d}B_t \qquad X_0 \sim p_0^*$$

$$\textbf{Observation model:} \quad \mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t$$

Posterior distribution of X_t given $\mathcal{Z}_t := \sigma(Z_s : 0 \le s \le t)$?

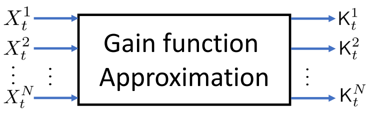
Solution: Feedback particle filter

$$\mathsf{P}(X_t|\mathcal{Z}_t) \approx \text{empirical dist. of } \{X_t^1, \dots, X_t^N\}$$

Mean-fld FPF: $(N=\infty)$

$$\mathrm{d}X_t^i = \underbrace{a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i}_{\text{Propagation}} + \underbrace{\mathsf{K}_t(X_t^i) \circ \left(\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t\right)}_{\text{Update}}, \qquad X_0^i \sim p_0^*$$

Finite-N implementation:



Yang, **M.**, Meyn. Feedback particle filter. *IEEE Trans. Aut. Control* (2013)

- Fokker-Planck equation for the conditional density of X_t^i:

$$\mathrm{d}p_t = \mathcal{L}p_t\,\mathrm{d}t - \nabla\cdot(p_t\mathsf{K}_t)\,\mathrm{d}Z_t + (\ldots)\,\mathrm{d}t, \quad p_0 = p_0^*$$

- Nonlinear filtering equation for the conditional density of X_t:

$$\mathrm{d}p_t^* = \mathcal{L}p_t^*\,\mathrm{d}t + p_t(h-\hat{h}_t)(\,\mathrm{d}Z_t - \hat{h}_t\,\mathrm{d}t), \quad p_0^* = p_0^*$$

The easy part

If K_t satisfies the following linear pde

$$\nabla\cdot(p_t\mathsf{K}_t) = -(h-\hat{h}_t)p_t \quad \forall\ t>0$$

then

$$p_t = p_t^* \quad \forall\ t>0$$

Laugesen, **M.**, Meyn and Raginsky. Poisson equation in nonlinear filtering. *SIAM J. Control and Optimization* (2015).

Why it works?
Exactness

- Fokker-Planck equation for the conditional density of X_t^i:

$$\mathrm{d}p_t = \mathcal{L}p_t\,\mathrm{d}t - \nabla\cdot(p_t\mathsf{K}_t)\,\mathrm{d}Z_t + (\ldots)\,\mathrm{d}t, \quad p_0 = p_0^*$$

- Nonlinear filtering equation for the conditional density of X_t:

$$\mathrm{d}p_t^* = \mathcal{L}p_t^*\,\mathrm{d}t + p_t(h - \hat{h}_t)(\mathrm{d}Z_t - \hat{h}_t\,\mathrm{d}t), \quad p_0^* = p_0^*$$

The easy part

If K_t satisfies the following linear pde

$$\nabla\cdot(p_t\mathsf{K}_t) = -(h - \hat{h}_t)p_t \quad \forall\ t > 0$$

then

$$p_t = p_t^* \quad \forall\ t > 0$$

Laugesen, **M.**, Meyn and Raginsky. Poisson equation in nonlinear filtering. *SIAM J. Control and Optimization* (2015).

- Fokker-Planck equation for the conditional density of X_t^i:

$$\mathrm{d}p_t = \mathcal{L}p_t \,\mathrm{d}t - \nabla \cdot (p_t \mathsf{K}_t)\,\mathrm{d}Z_t + (\ldots)\,\mathrm{d}t, \quad p_0 = p_0^*$$

- Nonlinear filtering equation for the conditional density of X_t:

$$\mathrm{d}p_t^* = \mathcal{L}p_t^* \,\mathrm{d}t + p_t(h - \hat{h}_t)(\mathrm{d}Z_t - \hat{h}_t\,\mathrm{d}t), \quad p_0^* = p_0^*$$

The easy part

If K_t satisfies the following linear pde

$$\nabla \cdot (p_t \mathsf{K}_t) = -(h - \hat{h}_t)p_t \quad \forall \; t > 0$$

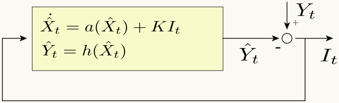
then

$$p_t = p_t^* \quad \forall \; t > 0$$

The hard part: Numerical approximation of the gain function

Laugesen, **M.**, Meyn and Raginsky. Poisson equation in nonlinear filtering. *SIAM J. Control and Optimization* (2015).

Kalman filter (KF): $$\mathrm{d}\hat{X}_t = a(\hat{X}_t)\,\mathrm{d}t + \underbrace{\mathsf{K}_t(\,\mathrm{d}Z_t - h(\hat{X}_t)\,\mathrm{d}t)}_{\text{update}}$$

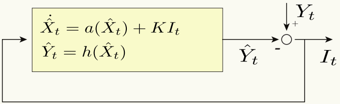


Kalman Filter

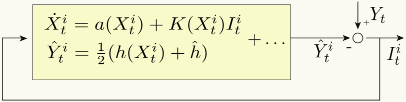
Zhang, Taghvaei, **M.** Feedback particle filter on Riemannian manifolds and Matrix Lie Groups. *IEEE Trans. Aut. Cntrl.* (2018)
Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

Kalman filter (KF): $\mathrm{d}\hat{X}_t = a(\hat{X}_t)\,\mathrm{d}t + \underbrace{\mathsf{K}_t(\,\mathrm{d}Z_t - h(\hat{X}_t)\,\mathrm{d}t)}_{\text{update}}$

FPF: $\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \,\mathrm{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ (\,\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t)}_{\text{update}}$



Kalman Filter



Feedback Particle Filter

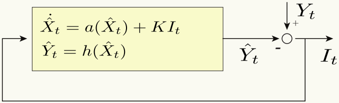
Zhang, Taghvaei, **M.** Feedback particle filter on Riemannian manifolds and Matrix Lie Groups. *IEEE Trans. Aut. Cntrl.* (2018)
Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

Why is it useful?
Analogy with the Kalman filter and the ensemble Kalman filter

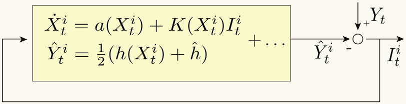
Kalman filter (KF): $$\mathrm{d}\hat{X}_t = a(\hat{X}_t)\,\mathrm{d}t + \underbrace{\mathsf{K}_t(\,\mathrm{d}Z_t - h(\hat{X}_t)\,\mathrm{d}t)}_{\text{update}}$$

FPF: $$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ (\,\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t)}_{\text{update}}$$

Ensemble KF: (sqrt-form) $$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i + \hat{\mathsf{K}}_t(\,\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t)$$



Kalman Filter

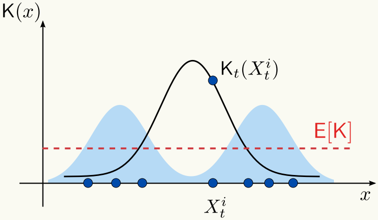


Feedback Particle Filter

Zhang, Taghvaei, **M.** Feedback particle filter on Riemannian manifolds and Matrix Lie Groups. *IEEE Trans. Aut. Cntrl.* (2018)
Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

FPF = EnKF in two limits:

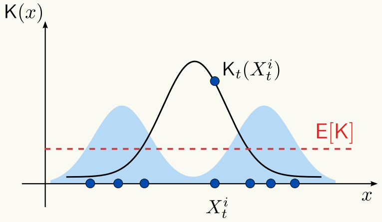
1. Linear Gaussian where gain function = Kalman gain
2. Approximation of the gain function by its average (constant) value



Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

FPF = EnKF in two limits:

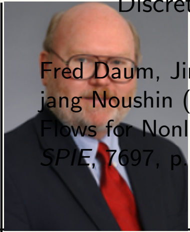
1. Linear Gaussian where gain function = Kalman gain
2. Approximation of the gain function by its average (constant) value



Question: Can we improve this approximation?

Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

Discrete-time



Fred Daum, Jim Huang and Arjang Noushin (2010). Exact Particle Flows for Nonlinear Filters. *Proc. SPIE*, 7697, p. 769704.

Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

Continuous-time

Dan Crisan and Jie Xiong (2010). Approximate McKean–Vlasov Representations for a Class of SPDEs. *Stochastics*, 82(1), pp. 5368.

Discrete-time

Fred Daum, Jim Huang and Arjang Noushin (2010). Exact Particle Flows for Nonlinear Filters. *Proc. SPIE*, 7697, p. 769704.

Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

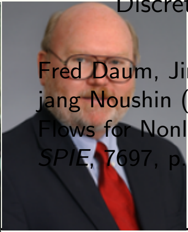
Literature

Interacting Particle Representations

Continuous-time

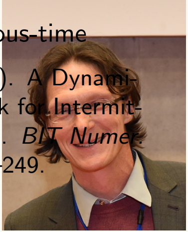
Dan Crisan and Jie Xiong (2010). Approximate McKean–Vlasov Representations for a Class of SPDEs. *Stochastics*, 82(1), pp. 5368.

Discrete-time



Fred Daum, Jim Huang and Arjang Noushin (2010). Exact Particle Flows for Nonlinear Filters. *Proc. SPIE*, 7697, p. 769704.

Continuous-time



Sebastian Reich (2011). A Dynamical Systems Framework for Intermittent Data Assimilation. *BIT Numer. Anal.*, 51(1), pp. 235–249.

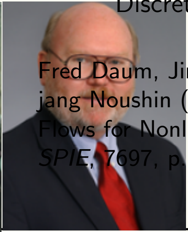
Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).

Literature
Interacting Particle Representations

Continuous-time

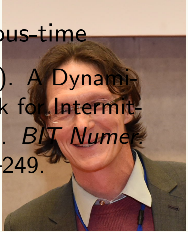
Dan Crisan and Jie Xiong (2010). Approximate McKean–Vlasov Representations for a Class of SPDEs. *Stochastics*, 82(1), pp. 5368.

Discrete-time



Fred Daum, Jim Huang and Arjang Noushin (2010). Exact Particle Flows for Nonlinear Filters. *Proc. SPIE*, 7697, p. 769704.

Continuous-time

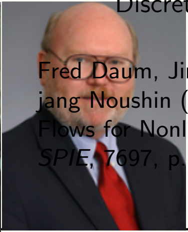


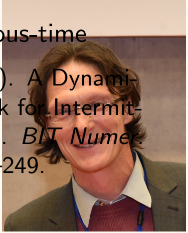
Sebastian Reich (2011). A Dynamical Systems Framework for Intermittent Data Assimilation. *BIT Numer. Anal.*, 51(1), pp. 235–249.

Continuous-time

Tao Yang, **M.**, and Sean Meyn (2011). Feedback Particle Filter with Mean-field Coupling. *IEEE Conf. on Decision and Control*, pp. 7909–16.

Taghvaei, de Wiljes, **M.**, and Reich, Kalman Filter and its Modern Extensions for the Continuous-time Nonlinear Filtering Problem, *ASME J. of Dynamic Systems, Measurement, and Control* (2018).





Continuous-time

Dan Crisan and Jie Xiong (2010). Approximate McKean–Vlasov Representations for a Class of SPDEs. *Stochastics*, 82(1), pp. 5368.

Discrete-time

Fred Daum, Jim Huang and Arjang Noushin (2010). Exact Particle Flows for Nonlinear Filters. *Proc. SPIE*, 7697, p. 769704.

Continuous-time

Sebastian Reich (2011). A Dynamical Systems Framework for Intermittent Data Assimilation. *BIT Numer. Anal.*, 51(1), pp. 235–249.

Continuous-time

Tao Yang, **M.**, and Sean Meyn (2011). Feedback Particle Filter with Mean-field Coupling. *IEEE Conf. on Decision and Control*, pp. 7909–16.

Connections/Extensions: Moselhy and Marzouk (2012); Reich (2013); Heng, Doucet and Pokern (2015); de Wiljes and Reich (2016-); Halder, Georgiou (2018); **Applications:** Neural particle filtering (Surace and Pfister (2017)); Satellite tracking (Berntrop, Berntrop and Grover 2015-); Dredging (Stano, 2013); Motion sensing (Tilton, 2013);· · ·

Outline

1. Numerics
   - Kernel algorithm (based on a diffusion map approximation)

2. Theory
   - Uniqueness

Poisson equation:

$$-\Delta_\rho \phi := -\frac{1}{\rho(x)} \nabla \cdot (\rho(x) \underbrace{\nabla\phi}_{\mathsf{K}}(x)) = (h(x) - \hat{h}) \quad \text{on} \quad \mathbb{R}^d$$

$$\int_{\mathbb{R}^d} \phi(x)\rho(x)\,\mathrm{d}x = 0$$

Numerical problem:

Given: $\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho$

Compute: $\{\mathsf{K}(X^1), \ldots, \mathsf{K}(X^N)\}$

Laugesen, **M.**, Meyn and Raginsky. Poisson equation in nonlinear filtering. *SIAM J. Control and Optimization* (2015).
Yang, Laugesen, **M.**, Meyn. Multivariable Feedback particle filter. *Automatica* (2016).

Poisson equation:

$$-\Delta_\rho \phi := -\frac{1}{\rho(x)} \nabla \cdot (\rho(x) \underbrace{\nabla\phi}_{\mathsf{K}}(x)) = (h(x) - \hat{h}) \quad \text{on} \quad \mathbb{R}^d$$

$$\int_{\mathbb{R}^d} \phi(x)\rho(x)\,\mathrm{d}x = 0$$

Numerical problem:

Given: $\{X^1, \ldots, X^N\} \overset{\text{i.i.d}}{\sim} \rho$

Compute: $\{\mathsf{K}(X^1), \ldots, \mathsf{K}(X^N)\}$

Assumptions/Notation:

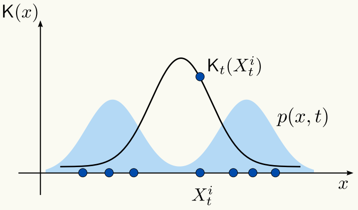
- Density $\rho = e^{-V}$ where $\lim_{|x|\to\infty} [-\Delta V(x) + \frac{1}{2}|\nabla V(x)|^2] = \infty$ and $D^2 V \in L^\infty$
- Function h is given with $h,\ \nabla h \in L^2(\rho; \mathbb{R}^d)$
- $\hat{h} := \int_{\mathbb{R}^d} h(x)\rho(x)\,\mathrm{d}x$

Laugesen, **M.**, Meyn and Raginsky. Poisson equation in nonlinear filtering. *SIAM J. Control and Optimization* (2015).
Yang, Laugesen, **M.**, Meyn. Multivariable Feedback particle filter. *Automatica* (2016).

(1) Non-Gaussian density, (2) Gaussian density

(1) Nonlinear gain function, (2) Constant gain function = Kalman gain

(1) FPF: $$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ (\,\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t)}_{\text{update}}$$



(1) Non-Gaussian density, (2) Gaussian density

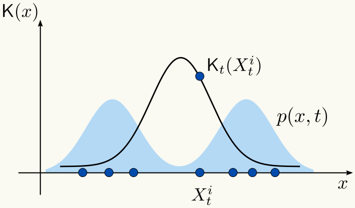
(1) Nonlinear gain function, (2) Constant gain function = Kalman gain

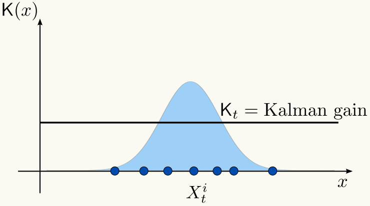
(1) FPF:

$$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ \left(\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t\right)}_{\text{update}}$$

(2) Linear Gaussian:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t\left(\mathrm{d}Z_t - \frac{HX_t^i + H\hat{X}_t}{2}\,\mathrm{d}t\right)}_{\text{update}}$$



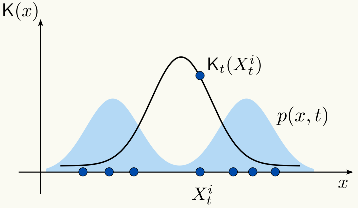


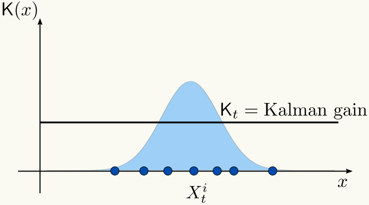
(1) Non-Gaussian density, (2) Gaussian density

(1) Nonlinear gain function, (2) Constant gain function = Kalman gain

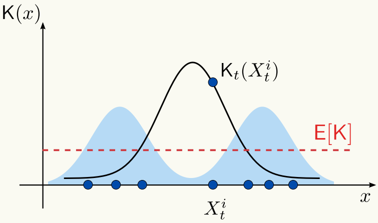
(1) FPF: $$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i) \circ \left(\mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t\right)}_{\text{update}}$$

(2) Linear Gaussian: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t\left(\mathrm{d}Z_t - \frac{HX_t^i + H\hat{X}_t}{2}\,\mathrm{d}t\right)}_{\text{update}}$$

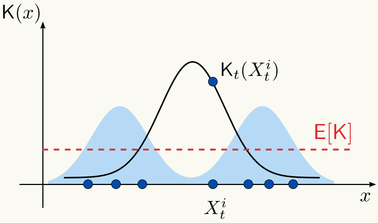




The blow-up of gain (on the left) is real! Leads to stiff numerical integration.



$$\mathsf{E}[\mathsf{K}] = \int (h(x) - \hat{h}) x \rho(x)\, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} (h(X^i) - \hat{h}) X^i$$

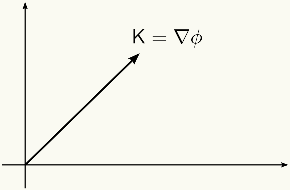


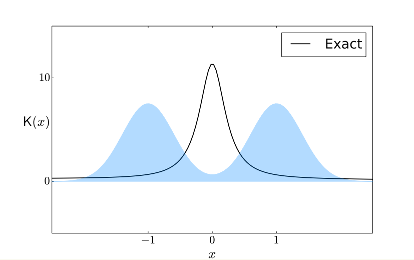
$$\mathsf{E}[\mathsf{K}] = \int (h(x) - \hat{h}) x \rho(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} (h(X^i) - \hat{h}) X^i$$

With a constant gain approximation, one obtains an ensemble Kalman filter

Non-Gaussian case
Galerkin approximation

$$\phi \in H_0^1(\rho, \mathbb{R}^d)$$

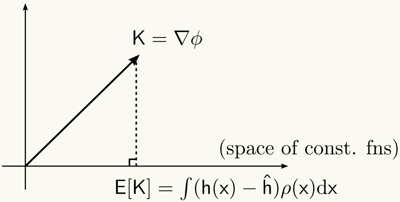


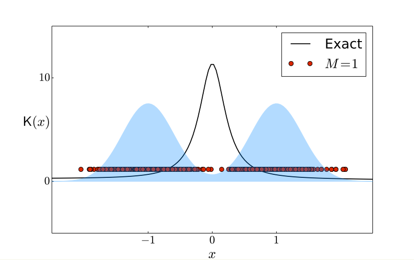


Non-Gaussian case

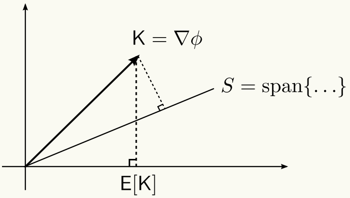
Galerkin approximation

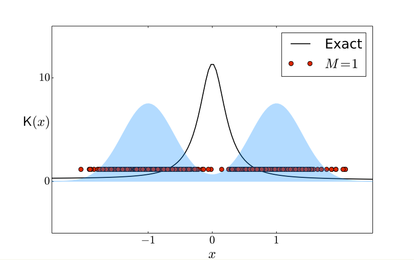
$$\phi \in H_0^1(\rho, \mathbb{R}^d)$$



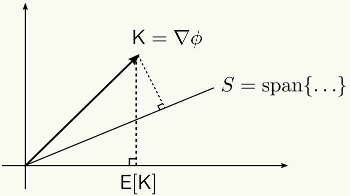


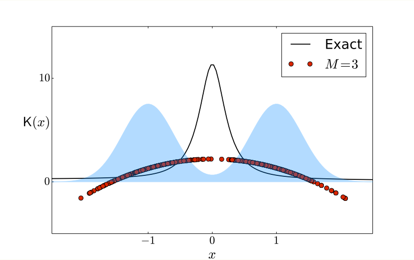
$$\phi \in H_0^1(\rho, \mathbb{R}^d)$$





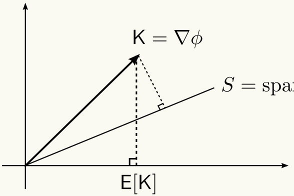
$\phi \in H_0^1(\rho, \mathbb{R}^d)$

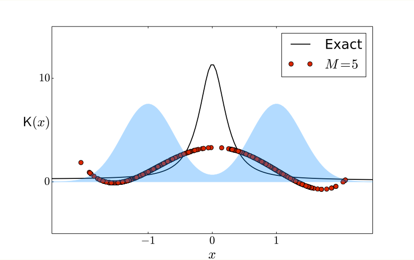




$$\psi \in \{1, x, \ldots, x^M\}$$

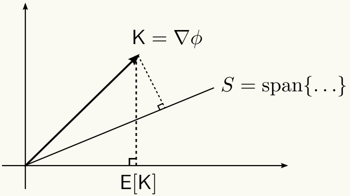
$\phi \in H_0^1(\rho, \mathbb{R}^d)$

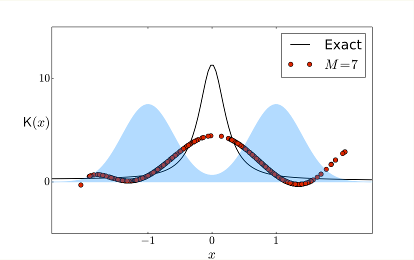




$$\psi \in \{1, x, \ldots, x^M\}$$

$$\phi \in H_0^1(\rho, \mathbb{R}^d)$$

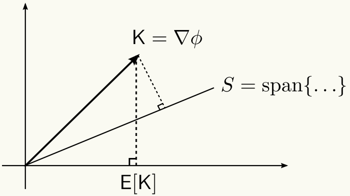


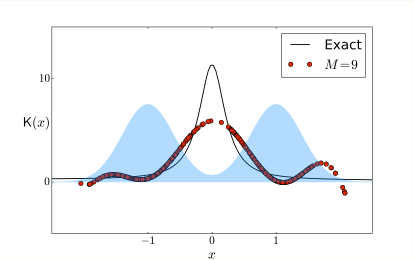


$$\psi \in \{1, x, \ldots, x^M\}$$

Non-Gaussian case
Galerkin approximation

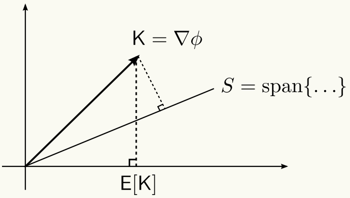
$\phi \in H_0^1(\rho, \mathbb{R}^d)$

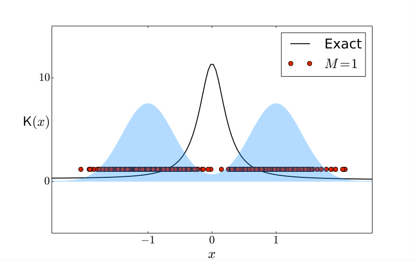




$$\psi \in \{1, x, \ldots, x^M\}$$

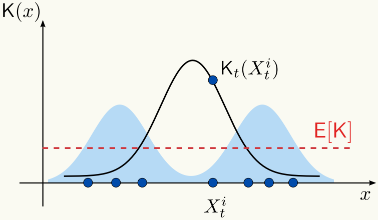
$\phi \in H_0^1(\rho, \mathbb{R}^d)$





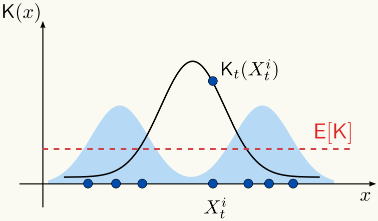
Moral of the story: basis function selection is non-trivial!

What are we looking for?
Ensemble Kalman filter +



$$\mathsf{E}[\mathsf{K}] = \int (h(x) - \hat{h}) x \rho(x) \, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} (h(X^i) - \hat{h}) X^i$$

What are we looking for?
Ensemble Kalman filter +



$$\mathsf{E}[\mathsf{K}] = \int (h(x) - \hat{h}) x \rho(x) \, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} (h(X^i) - \hat{h}) X^i$$

Question: Can we improve this approximation?

Kernel Algorithm (based on diffusion maps)

First the punchline

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N} \sum_{j=1}^{N} (h(X^j) - \hat{h}^{(N)}) X^j$$

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

Kernel Algorithm (based on diffusion maps)

First the punchline

1. No basis function selection!
2. Simple formula[a]

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N}\sum_{j=1}^{N}(h(X^j) - \hat{h}^{(N)})X^j$$

[a] Reminiscent of the ensemble transform (Reich, A non-parametric ensemble transform method for Bayesian inference, *SIAM J. Sci. Comput.*, (2013))

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N} \sum_{j=1}^{N} (h(X^j) - \hat{h}^{(N)}) X^j$$

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

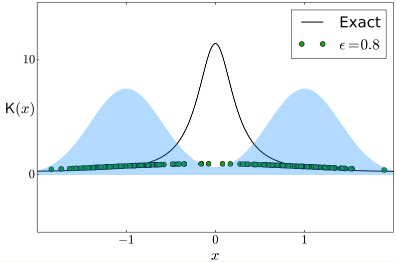
Kernel Algorithm (based on diffusion maps)
First the punchline

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N} \sum_{j=1}^{N} (h(X^j) - \hat{h}^{(N)}) X^j$$



Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

Kernel Algorithm (based on diffusion maps)

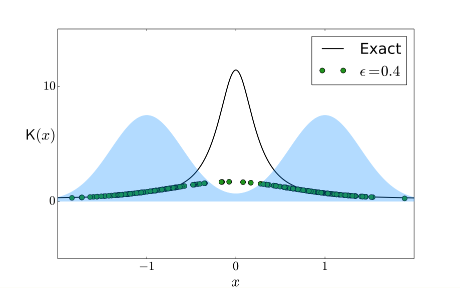
First the punchline

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N} \sum_{j=1}^{N} (h(X^j) - \hat{h}^{(N)}) X^j$$



Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

Kernel Algorithm (based on diffusion maps)

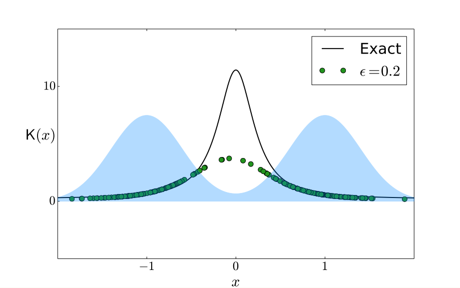
First the punchline

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N}\sum_{j=1}^{N}(h(X^j) - \hat{h}^{(N)})X^j$$



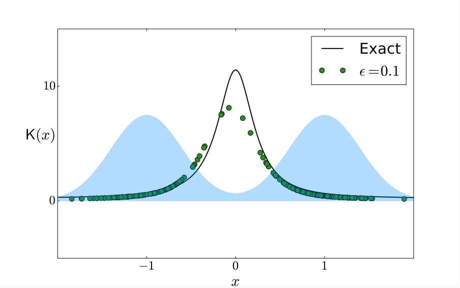
Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

1. No basis function selection!
2. Simple formula

$$\mathsf{K}^i = \sum_{j=1}^{N} s_{ij} X^j$$

3. Reduces to the constant gain in a certain limit

$$\mathsf{K}^i = \frac{1}{N}\sum_{j=1}^{N}(h(X^j) - \hat{h}^{(N)})X^j$$



Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).

(1) Poisson equation: $$-\Delta_\rho \phi = h - \hat{h}$$

(2) Semigroup formulation: $$\phi = e^{\epsilon\Delta_\rho}\phi + \int_0^\epsilon e^{s\Delta_\rho}(h-\hat{h})\,\mathrm{d}s$$

(3) Kernel approximation: $$\phi_\epsilon = T_\epsilon \phi_\epsilon + \epsilon(h - \hat{h}_\epsilon)$$

(4) Empirical approximation $$\phi_\epsilon^{(N)} = \mathsf{T}_\epsilon^{(N)} \phi_\epsilon^{(N)} + \epsilon(h - \hat{h}^N)$$

- $\mathsf{T}_\epsilon^{(N)}$ is a $N \times N$ Markov matrix,

$$\mathsf{T}_\epsilon^{(N)}{}_{ij} = \frac{k_\epsilon^{(N)}(X^i, X^j)}{\sum_{l=1}^N k_\epsilon^{(N)}(X^i, X_l)}$$

- $k_\epsilon^{(N)}(x,y)$ is the diffusion map kernel

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).
R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, et. al., Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

(1) Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

(2) Semigroup formulation: $\phi = e^{\epsilon\Delta_\rho}\phi + \int_0^\epsilon e^{s\Delta_\rho}(h-\hat{h})\,\mathrm{d}s$

(3) Kernel approximation: $\phi_\epsilon = T_\epsilon \phi_\epsilon + \epsilon(h - \hat{h}_\epsilon)$

(4) Empirical approximation $\phi_\epsilon^{(N)} = \mathsf{T}_\epsilon^{(N)}\phi_\epsilon^{(N)} + \epsilon(h - \hat{h}^N)$

- $\mathsf{T}_\epsilon^{(N)}$ is a $N \times N$ Markov matrix,

$$\mathsf{T}_\epsilon^{(N)}{}_{ij} = \frac{k_\epsilon^{(N)}(X^i, X^j)}{\sum_{l=1}^N k_\epsilon^{(N)}(X^i, X_l)}$$

- $k_\epsilon^{(N)}(x,y)$ is the diffusion map kernel

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).
R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, et. al., Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

(1) Poisson equation: $$-\Delta_\rho \phi = h - \hat{h}$$

(2) Semigroup formulation: $$\phi = e^{\epsilon\Delta_\rho}\phi + \int_0^\epsilon e^{s\Delta_\rho}(h - \hat{h})\,\mathrm{d}s$$

(3) Kernel approximation: $$\phi_\epsilon = T_\epsilon \phi_\epsilon + \epsilon(h - \hat{h}_\epsilon)$$

(4) Empirical approximation $$\phi_\epsilon^{(N)} = \mathsf{T}_\epsilon^{(N)}\phi_\epsilon^{(N)} + \epsilon(h - \hat{h}^N)$$

- $\mathsf{T}_\epsilon^{(N)}$ is a $N \times N$ Markov matrix,

$$\mathsf{T}_\epsilon^{(N)}{}_{ij} = \frac{k_\epsilon^{(N)}(X^i, X^j)}{\sum_{l=1}^N k_\epsilon^{(N)}(X^i, X_l)}$$

- $k_\epsilon^{(N)}(x,y)$ is the diffusion map kernel

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).
R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, et. al., Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

(1) Poisson equation: $$-\Delta_\rho \phi = h - \hat{h}$$

(2) Semigroup formulation: $$\phi = e^{\epsilon \Delta_\rho} \phi + \int_0^\epsilon e^{s\Delta_\rho}(h - \hat{h})\,\mathrm{d}s$$

(3) Kernel approximation: $$\phi_\epsilon = T_\epsilon \phi_\epsilon + \epsilon(h - \hat{h}_\epsilon)$$

(4) Empirical approximation $$\phi_\epsilon^{(N)} = \mathsf{T}_\epsilon^{(N)} \phi_\epsilon^{(N)} + \epsilon(h - \hat{h}^N)$$

- $\mathsf{T}_\epsilon^{(N)}$ is a $N \times N$ Markov matrix,

$$\mathsf{T}_\epsilon^{(N)}{}_{ij} = \frac{k_\epsilon^{(N)}(X^i, X^j)}{\sum_{l=1}^N k_\epsilon^{(N)}(X^i, X_l)}$$

- $k_\epsilon^{(N)}(x, y)$ is the diffusion map kernel

Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).
R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, et. al., Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

(1) Poisson equation: $-\Delta_\rho \phi = h - \hat{h}$

(2) Semigroup formulation: $\phi = e^{\epsilon \Delta_\rho} \phi + \int_0^\epsilon e^{s\Delta_\rho}(h - \hat{h})\,\mathrm{d}s$

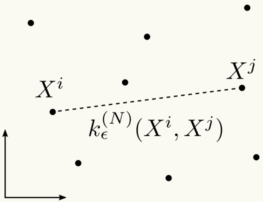
(3) Kernel approximation: $\phi_\epsilon = T_\epsilon \phi_\epsilon + \epsilon(h - \hat{h}_\epsilon)$

(4) Empirical approximation $\phi_\epsilon^{(N)} = \mathsf{T}_\epsilon^{(N)} \phi_\epsilon^{(N)} + \epsilon(h - \hat{h}^N)$

- $\mathsf{T}_\epsilon^{(N)}$ is a $N \times N$ Markov matrix,

$$\mathsf{T}_\epsilon^{(N)}{}_{ij} = \frac{k_\epsilon^{(N)}(X^i, X^j)}{\sum_{l=1}^N k_\epsilon^{(N)}(X^i, X_l)}$$

- $k_\epsilon^{(N)}(x,y)$ is the diffusion map kernel

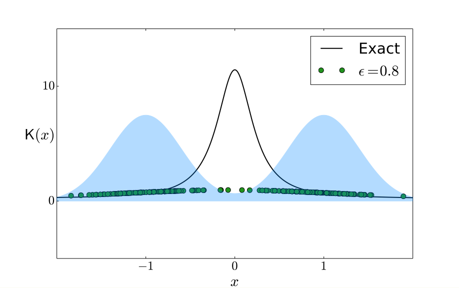


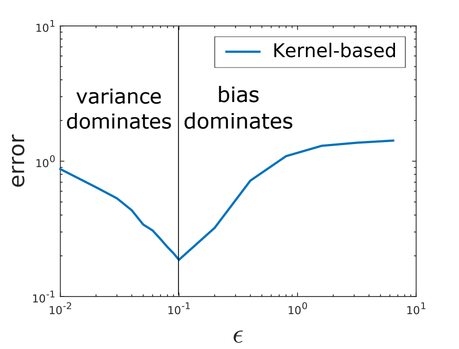
Taghvaei and **M.**, Gain Function Approximation for the Feedback Particle Filter, *IEEE Conference on Decision and Control*, (2016).
Taghvaei, **M.**, and Meyn, Error Estimates for the Kernel Gain Function Approximation in the Feedback Particle Filter, *American Control Conference*, (2017).
R. Coifman, S. Lafon, Diffusion maps, *Applied and computational harmonic analysis*, 2006,
M. Hein, et. al., Convergence of graph Laplacians on random neighborhood graphs, *JLMR*, 2007

Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

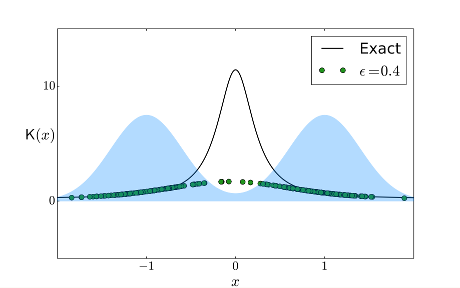
$$\underbrace{\mathsf{E}\left[\|\mathsf{K}-\mathsf{K}_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \sim \underbrace{O(\frac{1}{\epsilon^{1+d/2}\sqrt{N}})}_{\text{Variance}} + \underbrace{O(\epsilon)}_{\text{Bias}}$$

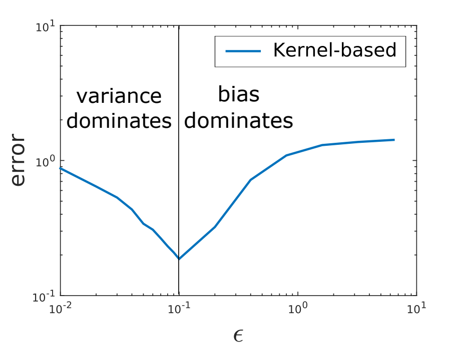




Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

$$\underbrace{\mathsf{E}\left[\|\mathsf{K}-\mathsf{K}_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \sim \underbrace{O(\frac{1}{\epsilon^{1+d/2}\sqrt{N}})}_{\text{Variance}} + \underbrace{O(\epsilon)}_{\text{Bias}}$$

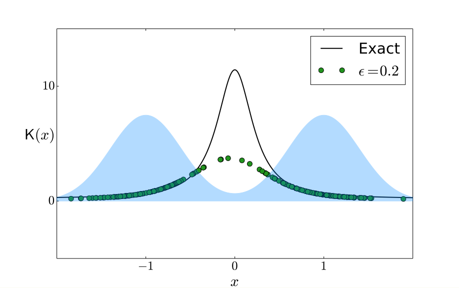


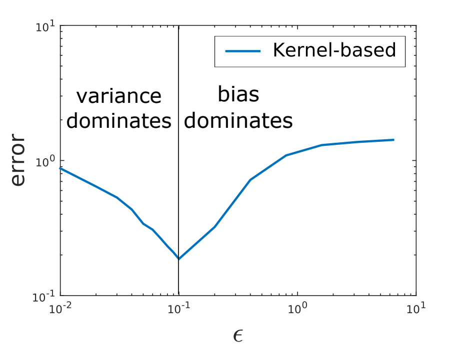


Convergence analysis
Numerics

Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

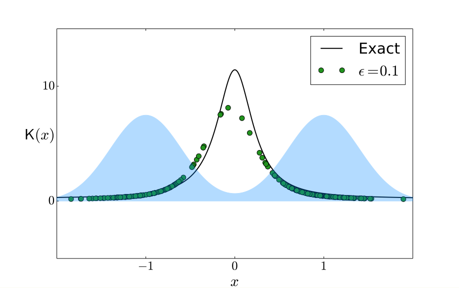
$$\underbrace{\mathsf{E}\left[\|\mathsf{K}-\mathsf{K}_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \sim \underbrace{O(\frac{1}{\epsilon^{1+d/2}\sqrt{N}})}_{\text{Variance}} + \underbrace{O(\epsilon)}_{\text{Bias}}$$

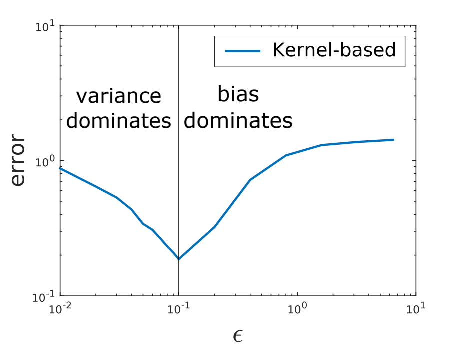




Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

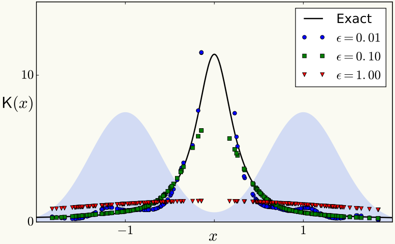
$$\underbrace{\mathsf{E}\left[\|\mathsf{K} - \mathsf{K}_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \sim \underbrace{O(\frac{1}{\epsilon^{1+d/2}\sqrt{N}})}_{\text{Variance}} + \underbrace{O(\epsilon)}_{\text{Bias}}$$

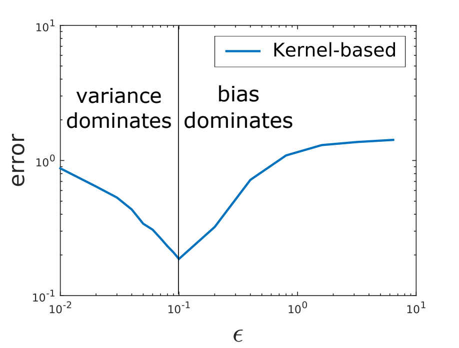




Convergence analysis: $\phi_\epsilon^{(N)} \xrightarrow[\text{variance}]{N\uparrow\infty} \phi_\epsilon \xrightarrow[\text{bias}]{\epsilon\downarrow 0} \phi$

$$\underbrace{\mathsf{E}\left[\|\mathsf{K}-\mathsf{K}_\epsilon^{(N)}\|_2\right]}_{\text{Total error}} \sim \underbrace{O(\frac{1}{\epsilon^{1+d/2}\sqrt{N}})}_{\text{Variance}} + \underbrace{O(\epsilon)}_{\text{Bias}}$$





Outline

1. Theory
   - Uniqueness
2. Numerics
   - Kernel algorithm (based on a diffusion map approximation)

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0)$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t$$

Feedback Particle Filter:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t}_{\text{Kalman gain}}\left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right)$$

- **Exactness:** The mean and covariance of X_t^i evolve according to Kalman filter
- **Non-uniqueness:** For *all* choices of skew-symmetric matrix Ω_t, the filter is exact!

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0)$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t$$

Feedback Particle Filter:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t}_{\text{Kalman gain}}\left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right)$$

- **Exactness:** The mean and covariance of X_t^i evolve according to Kalman filter
- **Non-uniqueness:** For *all* choices of skew-symmetric matrix Ω_t, the filter is exact!

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0)$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t$$

Feedback Particle Filter:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t}_{\text{Kalman gain}} \left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right) + \Omega_t \Sigma_t^{-1}(X_t^i - \hat{X}_t)$$

- **Exactness:** The mean and covariance of X_t^i evolve according to Kalman filter
- **Non-uniqueness:** For *all* choices of skew-symmetric matrix Ω_t, the filter is exact!

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0)$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t$$

Feedback Particle Filter:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}B_t^i + \underbrace{\mathsf{K}_t}_{\text{Kalman gain}} \left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right) + \Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)$$

- **Exactness:** The mean and covariance of X_t^i evolve according to Kalman filter
- **Non-uniqueness:** For *all* choices of skew-symmetric matrix Ω_t, the filter is exact!

Uniqueness issue: There are infinitely many ways to construct X_t^i!

Model:

$$dX_t = AX_t\,dt + dB_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

$$dZ_t = CX_t\,dt + dW_t,$$

Objective: Construct a unique process X_t^i s..t

$$X_t^i \sim N(\hat{X}_t, \Sigma_t)$$

where $\hat{X}_t$ and Σ_t are given by Kalman Filter.

Taghvaei and **M.**, An Optimal Transport Formulation of the Linear Feedback Particle Filter. *American Control Conference* (2016).

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

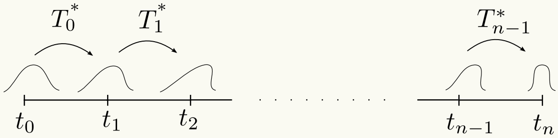
$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Objective: Construct a unique process X_t^i s..t

$$X_t^i \sim N(\hat{X}_t, \Sigma_t)$$

where $\hat{X}_t$ and Σ_t are given by Kalman Filter.

Solution methodology: A time-stepping procedure based on optimal transportation



Taghvaei and **M.**, An Optimal Transport Formulation of the Linear Feedback Particle Filter. *American Control Conference* (2016).

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

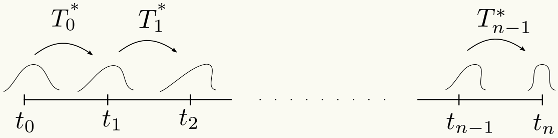
$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Objective: Construct a unique process X_t^i s..t

$$X_t^i \sim N(\hat{X}_t, \Sigma_t)$$

where $\hat{X}_t$ and Σ_t are given by Kalman Filter.

Solution methodology: A time-stepping procedure based on optimal transportation



This idea appears in other constructions of particle flow algorithms as well!

Taghvaei and **M.**, An Optimal Transport Formulation of the Linear Feedback Particle Filter. *American Control Conference* (2016).

Model:

$$\mathrm{d}X_t = aX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

$$\mathrm{d}Z_t = cX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Scalar case:

Opt. FPF: $$\mathrm{d}X_t^i = aX_t^i\,\mathrm{d}t + \frac{1}{2\Sigma_t}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t(\,\mathrm{d}Z_t - \frac{cX_t^i + c\hat{X}_t}{2}\,\mathrm{d}t)$$

This deterministic version of the ensemble Kalman filter also appears in the work of Jana de Wiljes and Sebastian Reich.

Model:

$$\mathrm{d}X_t = aX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$
$$\mathrm{d}Z_t = cX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Scalar case:

FPF: $$\mathrm{d}X_t^i = aX_t^i\,\mathrm{d}t + \mathrm{d}B_t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{cX_t^i + c\hat{X}_t}{2}\,\mathrm{d}t)$$

Opt. FPF: $$\mathrm{d}X_t^i = aX_t^i\,\mathrm{d}t + \frac{1}{2\Sigma_t}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{cX_t^i + c\hat{X}_t}{2}\,\mathrm{d}t)$$

This deterministic version of the ensemble Kalman filter also appears in the work of Jana de Wiljes and Sebastian Reich.

Model:

$$\mathrm{d}X_t = aX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

$$\mathrm{d}Z_t = cX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Scalar case:

FPF: $$\mathrm{d}X_t^i = aX_t^i\,\mathrm{d}t + \mathrm{d}B_t + \mathsf{K}_t\Big(\mathrm{d}Z_t - \frac{cX_t^i + c\hat{X}_t}{2}\,\mathrm{d}t\Big)$$

Opt. FPF: $$\mathrm{d}X_t^i = aX_t^i\,\mathrm{d}t + \frac{1}{2\Sigma_t}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t\Big(\mathrm{d}Z_t - \frac{cX_t^i + c\hat{X}_t}{2}\,\mathrm{d}t\Big)$$

Opt. FPF is a deterministic filter - Process noise is replaced by a deterministic term!

This deterministic version of the ensemble Kalman filter also appears in the work of Jana de Wiljes and Sebastian Reich.

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Vector case:

FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}\tilde{B}_t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t),$$

Opt. FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{\Sigma_t^{-1}}{2}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t)$$

$$+ \Omega_t \Sigma_t^{-1}(X_t^i - \hat{X}_t)\,\mathrm{d}t,$$

- Ω_t is the (skew symmetric) solution to the matrix equation:

$$\Omega_t \Sigma_t^{-1} + \Sigma_t^{-1}\Omega_t = A^T - A + \frac{1}{2}(\mathsf{K}_t C - C^T \mathsf{K}_t^T)$$

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$

$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Vector case:

FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}\tilde{B}_t + \mathsf{K}_t\Big(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\Big),$$

Opt. FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{\Sigma_t^{-1}}{2}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t\Big(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\Big) + \Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)\,\mathrm{d}t,$$

- Ω_t is the (skew symmetric) solution to the matrix equation:

$$\Omega_t\Sigma_t^{-1} + \Sigma_t^{-1}\Omega_t = A^T - A + \frac{1}{2}(\mathsf{K}_tC - C^T\mathsf{K}_t^T)$$

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$
$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Vector case:

FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}\tilde{B}_t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t),$$

Opt. FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{\Sigma_t^{-1}}{2}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t) + \Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)\,\mathrm{d}t,$$

- Ω_t is the (skew symmetric) solution to the matrix equation:

$$\Omega_t\Sigma_t^{-1} + \Sigma_t^{-1}\Omega_t = A^T - A + \frac{1}{2}(\mathsf{K}_t C - C^T \mathsf{K}_t^T)$$

Model:

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + \mathrm{d}B_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0),$$
$$\mathrm{d}Z_t = CX_t\,\mathrm{d}t + \mathrm{d}W_t,$$

Vector case:

FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \mathrm{d}\tilde{B}_t + \mathsf{K}_t\left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right),$$

Opt. FPF: $$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{\Sigma_t^{-1}}{2}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t\left(\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t\right) + \Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)\,\mathrm{d}t,$$

- Ω_t is the (skew symmetric) solution to the matrix equation:

$$\Omega_t\Sigma_t^{-1} + \Sigma_t^{-1}\Omega_t = A^T - A + \frac{1}{2}(\mathsf{K}_tC - C^T\mathsf{K}_t^T)$$

The skew-symmetric matrix term ($\Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)$) serves to cancel the curl!

Model:

$$dX_t = AX_t\,dt + dB_t, \quad X_0 \sim \mathcal{N}(\hat{X}_0, \Sigma_0)$$

$$dZ_t = CX_t\,dt + dW_t$$

Feedback Particle Filter:

$$dX_t^i = AX_t^i\,dt + dB_t^i + \underbrace{\mathsf{K}_t}_{\text{Kalman gain}} \left(dZ_t - \frac{CX_t^i + C\hat{X}_t}{2}\,dt \right) + \Omega_t \Sigma_t^{-1} (X_t^i - \hat{X}_t)$$

Uniqueness issue suggests:

1. A better model is needed to derive the filter;
2. Optimal transport may not be a suitable framework;
3. An optimal control-type formulation may be better suited.

Outline

1. Theory
   - Uniqueness
2. Numerics
   - Kernel algorithm (based on a diffusion map approximation)
3. Backup
   - Error analysis

Mean-field system:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{\Sigma_t^{-1}}{2}(X_t^i - \hat{X}_t)\,\mathrm{d}t + \mathsf{K}_t(\,\mathrm{d}Z_t - \frac{CX_t^i + C\hat{X}_t}{2}\,\mathrm{d}t)$$
$$+ \Omega_t\Sigma_t^{-1}(X_t^i - \hat{X}_t)\,\mathrm{d}t$$

Finite-N system:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \frac{1}{2}\Sigma_t^{(N)^{-1}}(X_t^i - m_t^{(N)})\,\mathrm{d}t + \mathsf{K}_t^{(N)}(\,\mathrm{d}Z_t - \frac{HX_t^i + Hm_t^{(N)}}{2}\,\mathrm{d}t)$$
$$+ \Omega_t^{-1}\Sigma_t^{(N)^{-1}}(X_t^i - m_t^{(N)})\,\mathrm{d}t, \quad \text{for} \quad i = 1, \ldots, N$$

where

Empirical mean: $\quad m_t^{(N)} := \dfrac{1}{N}\displaystyle\sum_{j=1}^{N} X_t^i$

Empirical covariance: $\quad \Sigma_t^{(N)} := \dfrac{1}{N-1}\displaystyle\sum_{j=1}^{N}(X_t^i - m_t^{(N)})(X_t^i - m_t^{(N)})^\top$

Assumptions:

(I) The system (A, H) is detectable, and (A, σ_B) is stabilizable

(II) The initial covariance $\Sigma_0^{(N)}$ is invertible

Main result:

$$\mathsf{E}[|m_t^{(N)} - m_t|^2] \leq (\text{const.})\frac{e^{-2\lambda_0 t}}{N}$$

$$\mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|_F^2] \leq (\text{const.})\frac{e^{-4\lambda_0 t}}{N}$$

where m_t and Σ_t are the true conditional mean and the error covariance, respectively.

J. de Wiljes, S. Reich, W. Stannat, Long-time stability and accuracy of the ensemble Kalman-Bucy filter for fully observed processes and small measurement noise (2016)
A. Taghvaei, P. G. Mehta, Error analysis of the linear feedback particle filter (ACC 2018)

Assumptions:

(I) The system (A, H) is detectable, and (A, σ_B) is stabilizable

(II) The initial covariance $\Sigma_0^{(N)}$ is invertible

Main result:

$$\mathsf{E}[|m_t^{(N)} - m_t|^2] \leq (\text{const.})\frac{e^{-2\lambda_0 t}}{N}$$

$$\mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|_F^2] \leq (\text{const.})\frac{e^{-4\lambda_0 t}}{N}$$

where m_t and Σ_t are the true conditional mean and the error covariance, respectively.

Proof idea:

- Evolution of $m_t^{(N)}$ and $\Sigma_t^{(N)}$ are exactly like Kalman filter equations
- Stability theory of Kalman filter applies!

J. de Wiljes, S. Reich, W. Stannat, Long-time stability and accuracy of the ensemble Kalman-Bucy filter for fully observed processes and small measurement noise (2016)
A. Taghvaei, P. G. Mehta, Error analysis of the linear feedback particle filter (ACC 2018)

Mean-field process:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \sigma_B\,\mathrm{d}B_t^i + \mathsf{K}_t(\,\mathrm{d}Z_t - \frac{HX_t^i + H\hat{X}_t}{2}\,\mathrm{d}t)$$

Finite-N system:

$$\mathrm{d}X_t^i = AX_t^i\,\mathrm{d}t + \sigma_B\,\mathrm{d}B_t^i + \mathsf{K}_t^{(N)}(\,\mathrm{d}Z_t - \frac{HX_t^i + H\bar{m}_t^{(N)}}{2}\,\mathrm{d}t), \quad \text{for} \quad i = 1, \ldots, N$$

Problem statement:

- Convergence $m_t^{(N)} \to m_t, \quad \Sigma_t^{(N)} \to \Sigma_t$
- Convergence of the empirical distribution

Related literature: Error analysis of the Ensemble Kalman filter

- **discrete time:** F. Le Gland, et. al. (2009), J. Mandel, et. al. (2011), D. Kelly, et. al. (2014), X. T. Tong, et. al. (2016)
- **continuous time:** Del Moral, et. al. (2016,2017), J. de Wiljes, et. al. (2016)

This remains an active area of research

Given: $\{X_t^1, \ldots, X_t^N\} \overset{\text{i.i.d}}{\sim} \rho \quad \overset{\text{(BVP)}}{\Longrightarrow}$ **Compute:** $\{\mathsf{K}_t(X_t^1), \ldots, \mathsf{K}_t(X_t^N)\}$

FPF sde: $$\mathrm{d}X_t^i = \ldots + \mathsf{K}_t(X_t^i)\,\mathrm{d}Z_t + u_t^i\,\mathrm{d}t$$

And its analysis:

Mean-field model

1. BVP: $\exists!$, regularity estimates — $\mathsf{E}[|\mathsf{K}_t|^2] < \infty$, $\mathsf{E}[|u_t|] < \infty$
2. Numerical methods — (This talk)
3. Optimality

Finite-N model

4. $\exists!$ of McKean-Vlasov sde
5. Prop. of chaos + error estimates
6. Simulation variance estimates

Non-uniqueness of the gain function

$$\nabla \cdot (\rho(x)\mathsf{K}(x)) = -(h(x) - \hat{h})\rho(x) \quad \text{on} \quad \mathbb{R}^d$$

- Non-uniqueness:

$$\nabla \cdot (\rho \, J \nabla \log \rho) = 0, \qquad \forall \ \text{skew-symmetric matrices} \ \ J$$

- Scalar case

$$\mathsf{K}(x) = \frac{1}{\rho(x)} \int_{-\infty}^{x} (h(y) - \hat{h}(y))\rho(y)\,\mathrm{d}y \approx \frac{1}{\rho(x)} \frac{1}{N} \sum_{\{i: X^i < x\}} (h(X_i) - \hat{h})$$

Non-uniqueness of the gain function

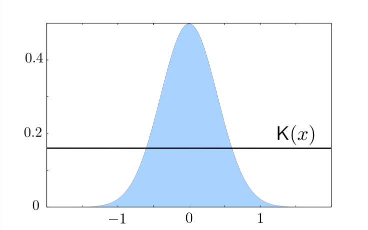
$$\nabla \cdot (\rho(x)\mathsf{K}(x)) = -(h(x) - \hat{h})\rho(x) \quad \text{on} \quad \mathbb{R}^d$$

- Non-uniqueness:

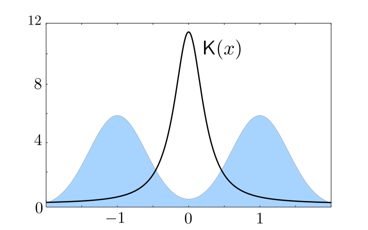
$$\nabla \cdot (\rho\, J \nabla \log \rho) = 0, \qquad \forall \text{ skew-symmetric matrices } J$$

- Scalar case

$$\mathsf{K}(x) = \frac{1}{\rho(x)} \int_{-\infty}^{x} (h(y) - \hat{h}(y))\rho(y)\,\mathrm{d}y \approx \frac{1}{\rho(x)} \frac{1}{N} \sum_{\{i : X^i < x\}} (h(X_i) - \hat{h})$$



Unimodal distribution



Bimodal Distribution

Non-uniqueness of the gain function

$$\nabla \cdot (\rho(x)\mathsf{K}(x)) = -(h(x) - \hat{h})\rho(x) \quad \text{on} \quad \mathbb{R}^d$$

- Non-uniqueness:

$$\nabla \cdot (\rho\, J \nabla \log \rho) = 0, \qquad \forall \text{ skew-symmetric matrices } J$$

- Scalar case

$$\mathsf{K}(x) = \frac{1}{\rho(x)} \int_{-\infty}^{x} (h(y) - \hat{h}(y))\rho(y)\,\mathrm{d}y \approx \frac{1}{\rho(x)} \frac{1}{N} \sum_{\{i : X^i < x\}} (h(X_i) - \hat{h})$$

- Vector case: (particular soln.)

$$\begin{aligned} \rho\mathsf{K} = \nabla\phi \quad &\Rightarrow \quad \nabla \cdot (\nabla\phi) = -\rho(h - \hat{h}) = r \quad \text{on} \quad \mathbb{R}^d \\ &\Rightarrow \quad \phi(x) = \int g(x-y) r(y)\,\mathrm{d}y \\ &\Rightarrow \quad \mathsf{K}(x) = \frac{1}{\rho(x)} \int \nabla g(x-y) r(y)\,\mathrm{d}y \end{aligned}$$