Sequential Kernel Herding: Frank-Wolfe Optimization for Particle Filtering

Simon Lacoste-Julien
INRIA / ENS, France
SIERRA Project Team

Fredrik Lindsten
University of Cambridge, UK
Department of Engineering

Francis Bach
INRIA / ENS, France
SIERRA Project Team

Journée MAS 2014 – Session statistique et optimisation

August 27th 2014

Summary in one slide

- Recent work [Bach et al. ICML 12] showed **how Frank-Wolfe optimization** could obtain **adaptive quadrature rules** with potentially better rates than Monte-Carlo (MC) or quasi-Monte-Carlo (QMC) integration

- Here we replace the random sampling phase in a **particle filter** with Frank-Wolfe optimization to get better locations of particles to approximate the distribution (a mixture of Gaussians)

- Our preliminary empirical study indicates that we can obtain improvements over MC or QMC in term of number of particles

Part I: Adaptive quadrature rule with Frank-Wolfe optimization

- Approximating integrals: $\int_{\mathcal{X}} f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$

  for **fixed** p, and multiple f's in a RKHS $\mathcal{H}$

  - Random sampling $x^{(i)} \sim p(x)$ yields $O(1/\sqrt{N})$ error
  - Kernel herding [Chen et al. 10] (can) yield $O(1/N)$ error!
    (need finite dim. $\mathcal{H}$) ← (like quasi-MC)
  - -> generalized to FW optimization [Bach et al. 12] and could even get $O(e^{-cN})$ error

- Trick: run Frank-Wolfe optimization on dummy objective:

$$\min_{g \in \mathcal{M}} \frac{1}{2}\|g - \mu(p)\|_{\mathcal{H}}^2$$

where $\mathcal{M} = \text{cl-conv}\,(\Phi(\mathcal{X}))$ is the *marginal polytope*

and $\mu(p) = \mathbb{E}_{p(x)}\Phi(x)$ is the *mean map*

→ representer: $k(x,\cdot) \in \mathcal{H}$

Approx. integrals in RKHS

- Why? Well, controlling **moment discrepancy** $\|\mu(\hat{p}) - \mu(p)\|_{\mathcal{H}}$ is enough to control **error of integrals** in RKHS $\mathcal{H}$:

- Reproducing property: $f \in \mathcal{H} \Rightarrow f(x) = \langle f, \Phi(x) \rangle$
- Define *mean map* : $\mu(p) = \mathbb{E}_{p(x)} \Phi(x)$
- Want to approximate integrals of the form:

$$\mathbb{E}_{p(x)} f(x) = \mathbb{E}_{p(x)} \langle f, \Phi(x) \rangle = \langle f, \mu(p) \rangle$$

- Use weighted sum to get approximated mean: $\hat{p} = \sum_{i=1}^{N} w_t^{(i)} \delta_{x^{(i)}}$

$$\mu(\hat{p}) = \mathbb{E}_{\hat{p}(x)} \Phi(x) = \sum_{i=1}^{N} w^{(i)} \Phi(x^{(i)}) \Rightarrow \mathbb{E}_{\hat{p}(x)} f(x) = \sum_{i=1}^{N} w^{(i)} f(x^{(i)})$$

- Approximation error is then bounded by:

$$|\mathbb{E}_{p(x)} f(x) - \mathbb{E}_{\hat{p}(x)} f(x)| \leq \|f\|_{\mathcal{H}} \, \|\mu(p) - \mu(\hat{p})\|_{\mathcal{H}}$$

Frank-Wolfe algorithm [Frank, Wolfe 1956]

(aka conditional gradient)

- alg. for constrained opt.: $\min_{\alpha \in \mathcal{M}} f(\alpha)$

  where:

  f convex & cts. differentiable

  $\mathcal{M}$ convex & compact

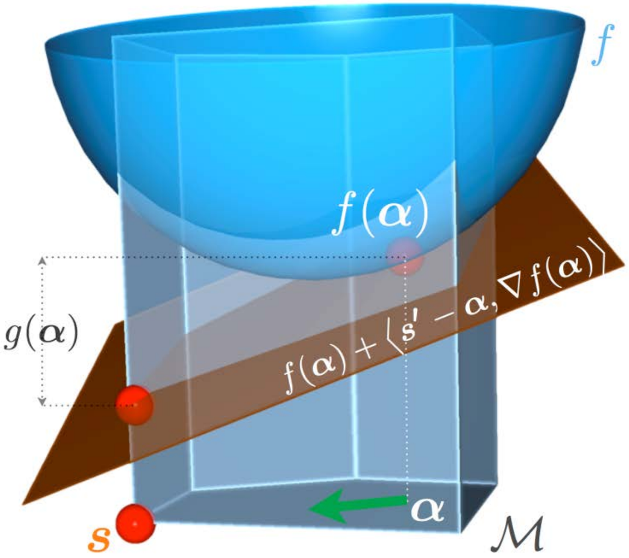
- FW algorithm – repeat:

1) Find good feasible direction by minimizing linearization of f :

$$\boldsymbol{s}_{k+1} \in \arg\min_{\boldsymbol{s}' \in \mathcal{M}} \langle \boldsymbol{s}', \nabla f(\boldsymbol{\alpha}_k) \rangle$$

2) Take convex step in direction:

$$\boldsymbol{\alpha}_{k+1} = (1 - \gamma_k)\, \boldsymbol{\alpha}_k + \gamma_k\, \boldsymbol{s}_{k+1}$$



- Properties: O(1/N) rate
  - sparse iterates
  - get duality gap $g(\alpha)$ for free
  - affine invariant
  - rate holds even if linear subproblem solved **approximately**

FW quadrature

repeat:

input: p

1) FW search:

$$x^{(k+1)} = \arg\min_{x \in \mathcal{X}} \; g_k(x) - \mu(p)(x)$$

e.g. minimum of a difference of mixture of Gaussian bumps!

2) convex combo:

$$g_{k+1} = (1 - \gamma_k)\, g_k + \gamma_k \, \Phi(x^{(k+1)})$$

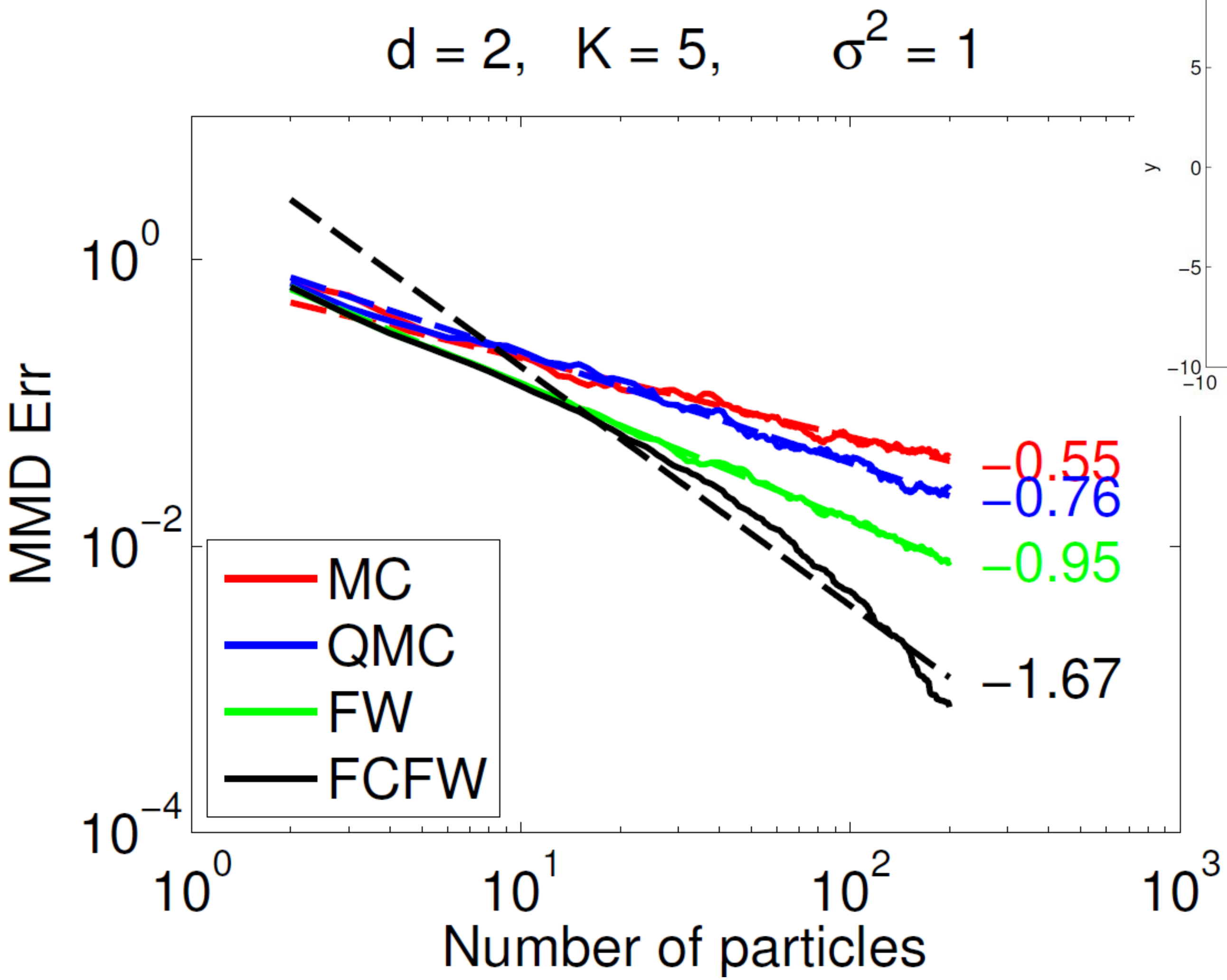
at end: $g_N = \sum_{i=1}^{N} w^{(i)} \Phi(x^{(i)})$

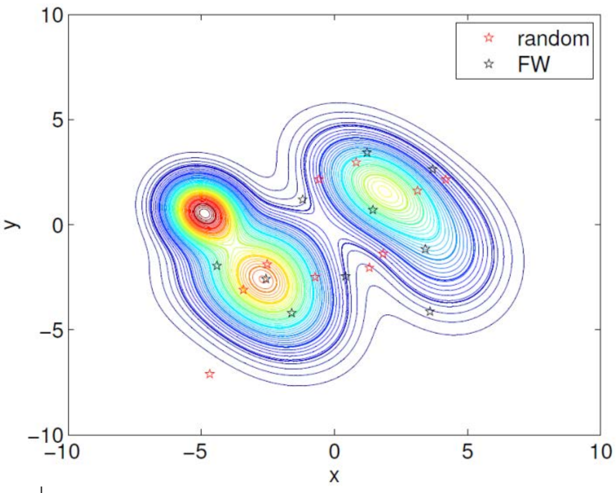
- Theoretical rates for $\|\mu(\hat{p}) - \mu(p)\|_{\mathcal{H}}$

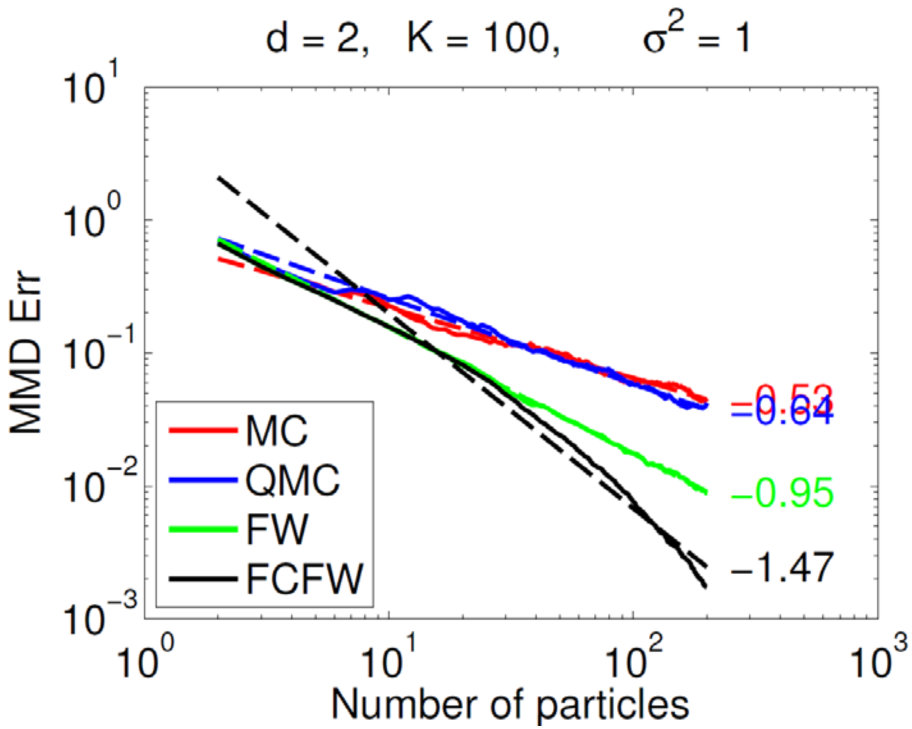
- variations:
  - kernel herding: $\gamma_k = \frac{1}{k+1}$
  - line-search FW
  - fully-corrective FW (FCFW)

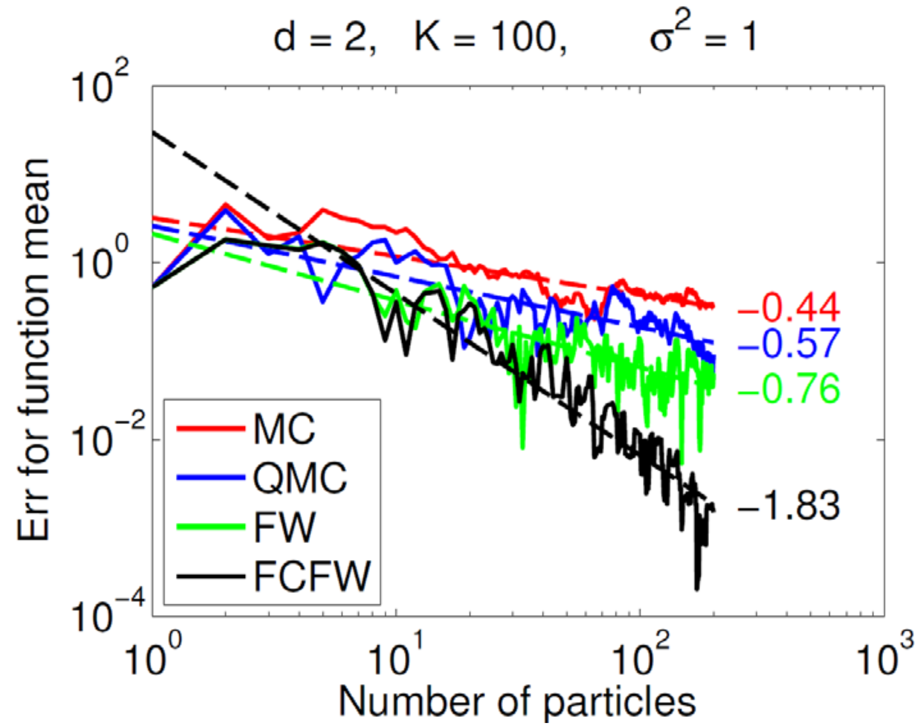
$\dim(\mathcal{H})$:	finite	infinite
kernel herding	$O(1/N)$	$O(1/\sqrt{N})$
line-search FW	$O(e^{-cN})$	$O(1/\sqrt{N})$
fully-corrective FW (FCFW)	$O(e^{-cN})$	$O(1/\sqrt{N})$

Fitting a mixture of Gaussian

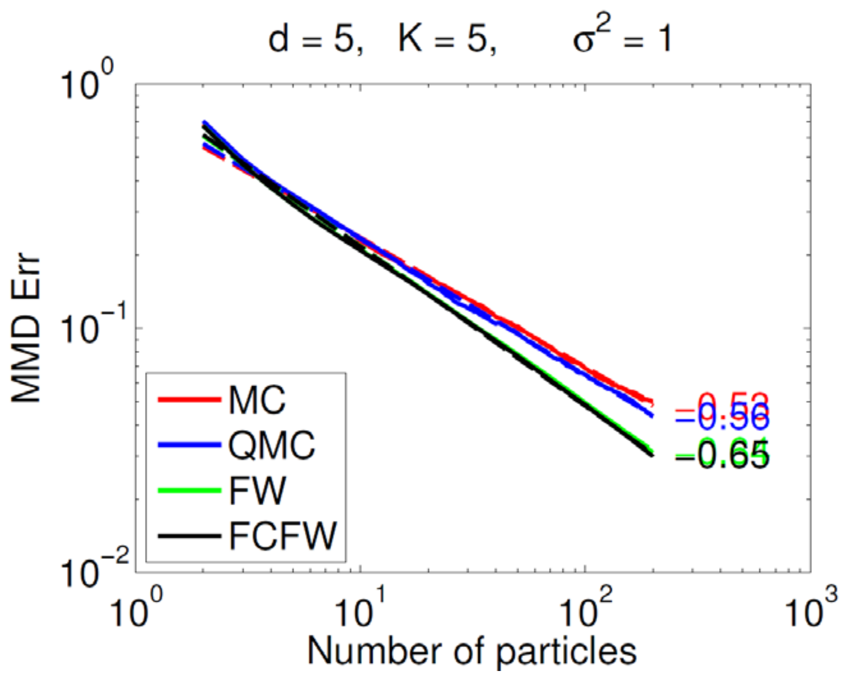





d = 2, K = 100, σ² = 1
MMD Err
Number of particles
MC
QMC
FW
FCFW
−0.52
−0.64
−0.95
−1.47


d = 2, K = 100, σ² = 1
Err for function mean
Number of particles
MC
QMC
FW
FCFW
−0.44
−0.57
−0.76
−1.83

higher d:


d = 5, K = 5, σ² = 1
MMD Err
Number of particles
MC
QMC
FW
FCFW
−0.52
−0.56
−0.65

Part II: Particle filtering

- HMM / state-space model: $p(x_{1:T}, y_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t-1})\, p(y_t|x_t)$
- goal: approximate filtering distribution $p(x_{1:t}|y_{1:t})$
  with weighted set of N 'particles' $\{x_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$:

$$p(x_{1:t}|y_{1:t}) \approx q_t(x_{1:t}) := \sum_{i=1}^{N} w_t^{(i)}\, \delta(x_{1:t}^{(i)}, x_{1:t})$$

E.g. a mixture of Gaussians!

- One view of PF algorithm:

Propagate approximation forward in time by:

1) Sample new particles from: $\bar{q}_{t+1}(x_{1:(t+1)}) := p(x_{t+1}|x_t) q_t(x_{1:t})$

$$x_{1:(t+1)}^{(i)} \sim \bar{q}_{t+1} \qquad = \sum_{i=1}^{N} w_t^{(i)}\, \delta(x_{1:t}^{(i)}, x_{1:t}) p(x_{t+1}|x_t^{(i)})$$

2) Reweight particles according to observation:

$$w_{t+1}^{(i)} \propto p(y_{t+1}|x_{t+1}^{(i)})$$

New weighted set gives:
$q_{t+1}(x_{1:(t+1)})$

Sequential Kernel Herding

- **Main idea:** replace the random sampling step to approximate $\bar{q}_{t+1}$ with **FW-quadrature**
  - (aside: if use quasi-random sampling from $\bar{q}_{t+1}$ instead, we get the previously proposed QMC particle filters)

[Philomin et al. ECCV 00, Ormoneit et al. UAI 01]

1) $\{x^{(i)}_{1:(t+1)}, \bar{w}^{(i)}_{t+1}\}_{i=1}^{N}$ obtained from FW-quadrature on $\bar{q}_{t+1}(x_{1:(t+1)})$

$:= p(x_{t+1}|x_t)q_t(x_{1:t})$

2) $w^{(i)}_{t+1} \propto \bar{w}^{(i)}_{t+1} p(y_{t+1}|x^{(i)}_{t+1})$

- Modular algorithm! Can add FW-quadrature anywhere need to get particles to approximate distribution
- Conditions to run:
  - need to be able to compute expectation of kernel with $\bar{q}_{t+1}$
  - need to be able to (approx.) optimize this function
- In our experiments: $\bar{q}_{t+1}$ is a mixture of Gaussians; we use Gaussian kernel; optimize non-convex problem using exhaustive search over **random sample** from $\bar{q}_{t+1}$

Convergence result

- current result (roughly):
  - assume that: $\mathcal{H}_t = \mathcal{H} \quad \forall t$

$$f_t(x_{t+1}, \cdot) := p(x_{t+1}|\cdot)\, p(y_t|\cdot) \;\in \mathcal{H} \quad \forall x_{t+1}$$

and regularity condition on norm of f_t

  - then:

for fixed t, MMD error on **predictive** $p(x_{t+1}|y_{1:t})$ is $O(\epsilon)$

where ϵ is bound on FW MMD error at each t

- so in if $\mathcal{H}$ is finite dimensional:
  - can get provably faster rates than PF (for integrals of members of $\mathcal{H}$)
  - compare with $o(\frac{1}{\sqrt{N}})$ for sequential QMC in [Garber & Chopin 14]

Synthetic experiments

- Evaluated in simulation study on different models:
  - Linear Gaussian models (orders d=3 and d=15)
  - Jump Markov linear model

$$P(r_t = l | r_{t-1} = k) \sim \Pi_{kl}$$
$$x_t = A(r_t)x_{t-1} + v_t$$
$$y_t = C(r_t)x_t + e_t$$

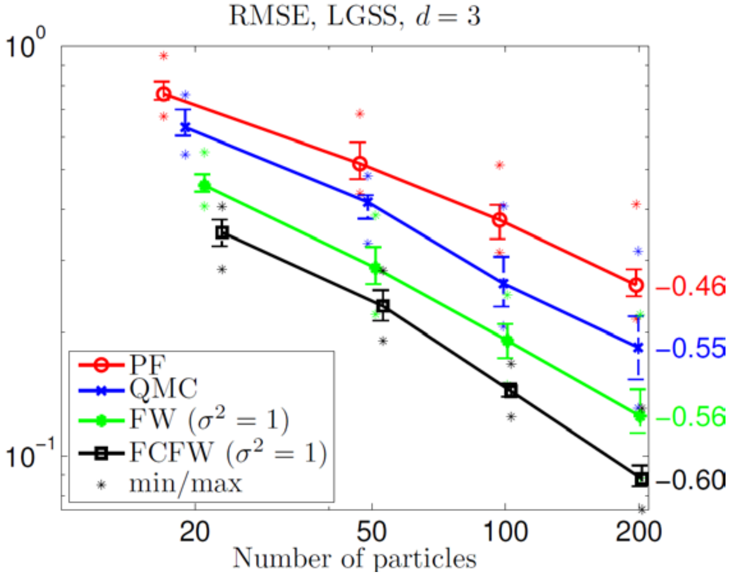
  - Nonlinear time series model

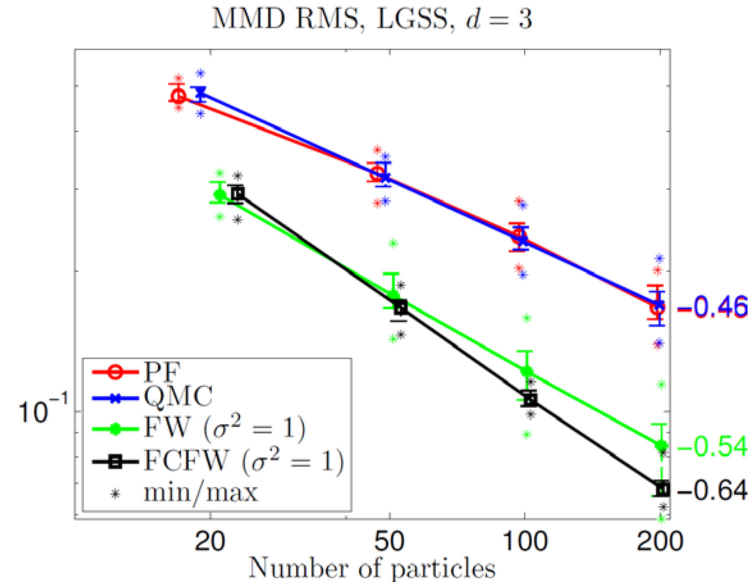
$$x_t = \frac{1}{2}x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} \quad 8\cos\big(1.2(t-1)\big) + v_t$$
$$y_t = \frac{1}{20}x_t^2 + e_t$$

- T=100 time steps for all models
- $\sigma^2 \in \{0.01, 0.1, 1\}$ (variance of Gaussian kernel)
- FW quadrature points for mixture of Gaussians chosen by optimizing through 50k random samples

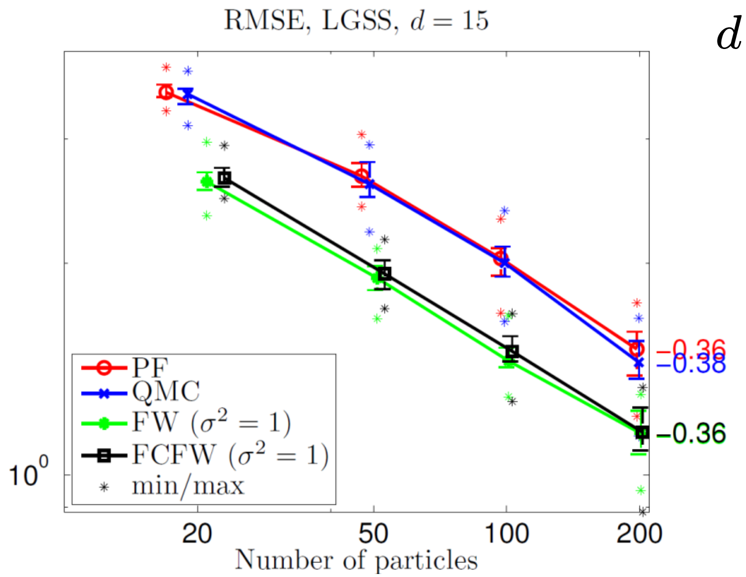
Results: Linear Gaussian system

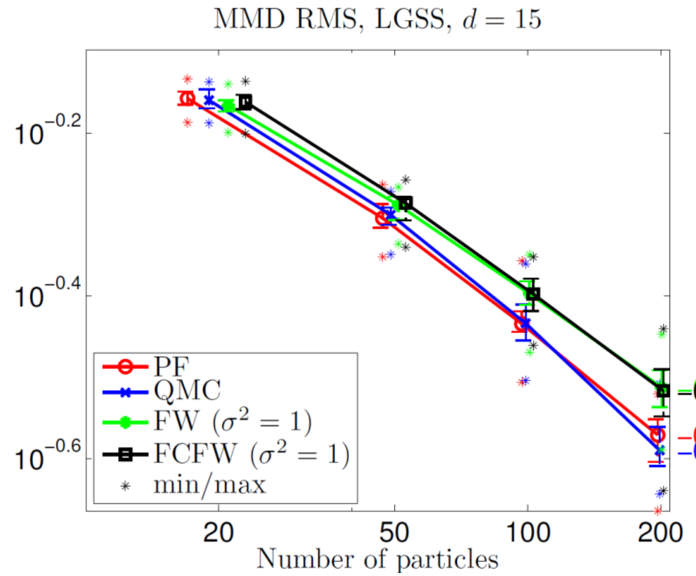
$d = 3$





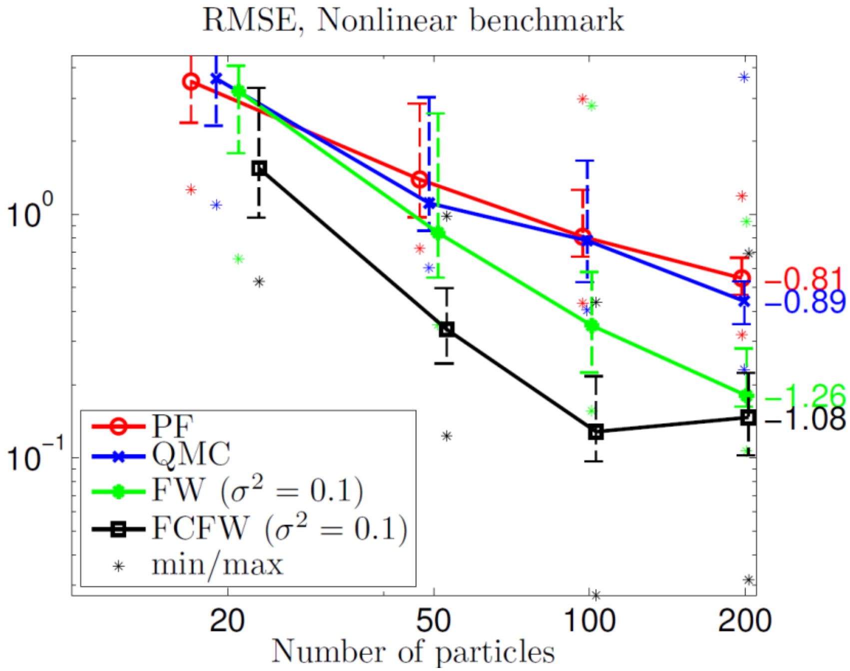
$d = 15$

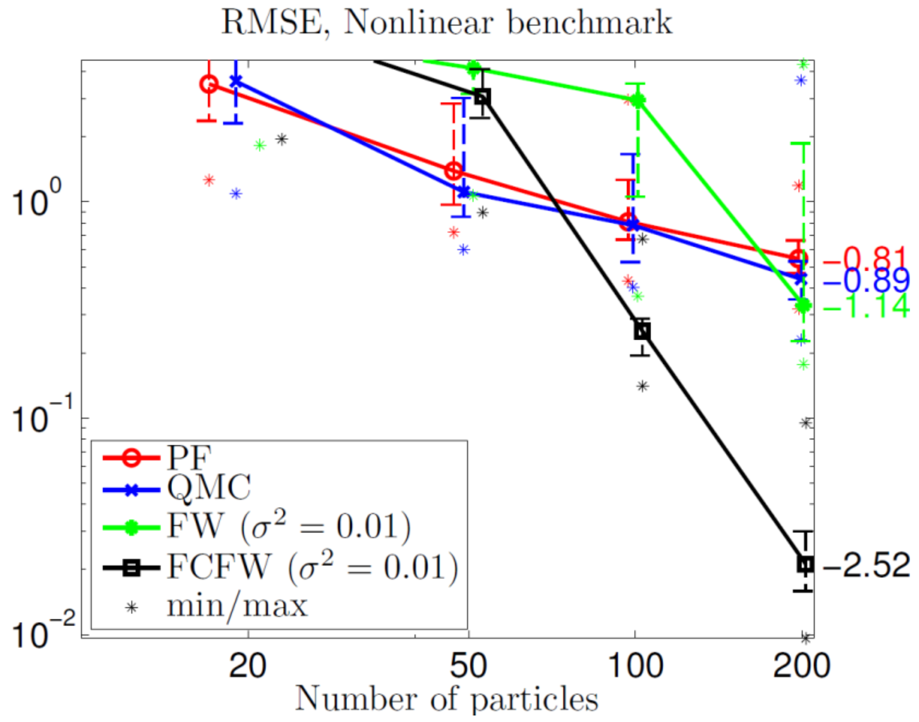




$\sigma^2 = 1$

Nonlinear 1d time series results:





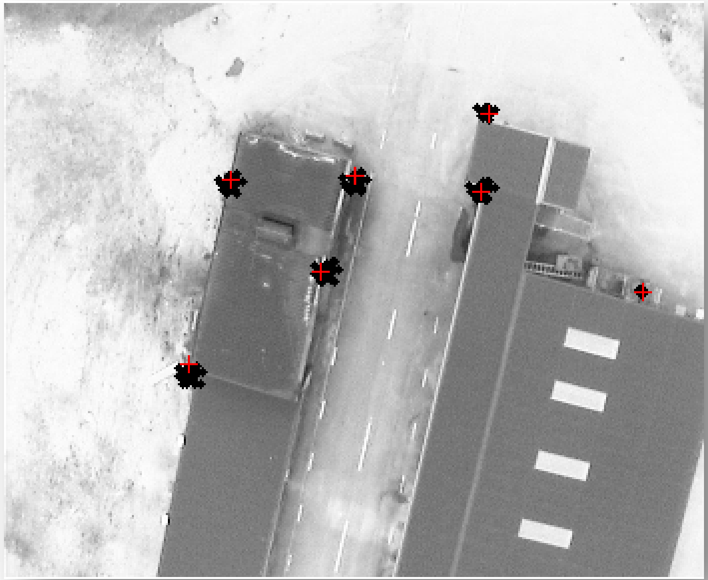
Robot localization experiment

- The UAV is tracked using IMU and visual odometry
- High-dimensional vehicle state:
  - pose, velocities, accelerations
  - sensor biases
  - landmark positions
- Four filters:
  - PF, QMC, FW-SKH, FCFW-SKH
  - all Rao-Blackwellized

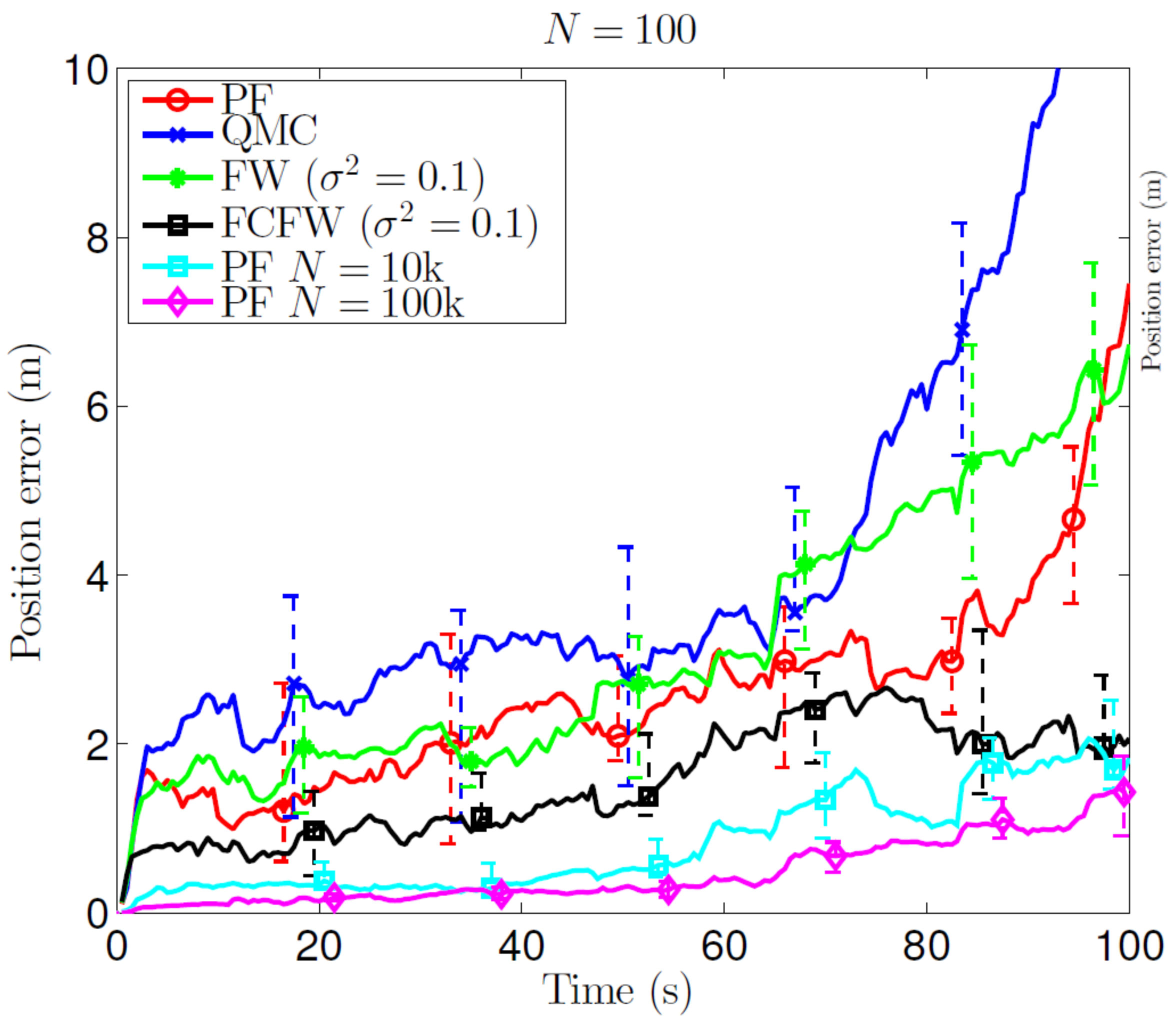
  [particles on 7d state:

  3d space + quaternion rotation]
- Compare position errors relative to a reference trajectory (mean of 10 PF with N = 100k)

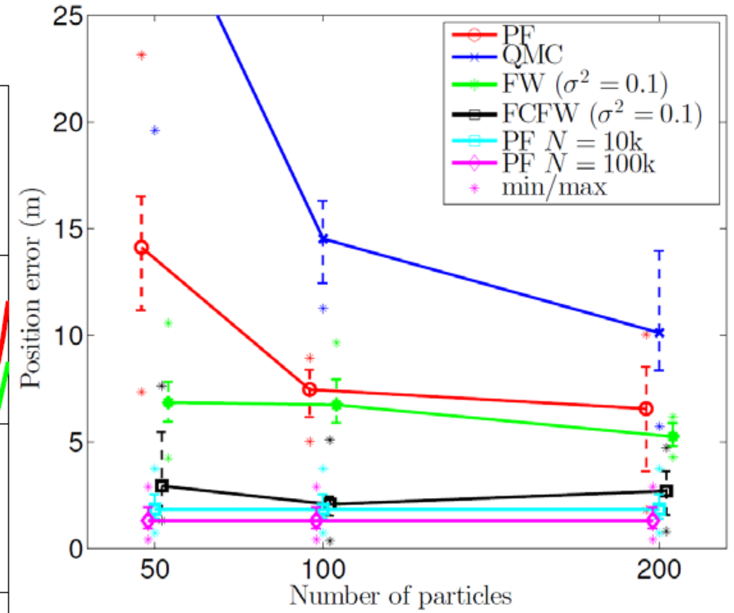
Yamaha RMAX UAV



Robot localization results



error last time step



Conclusion

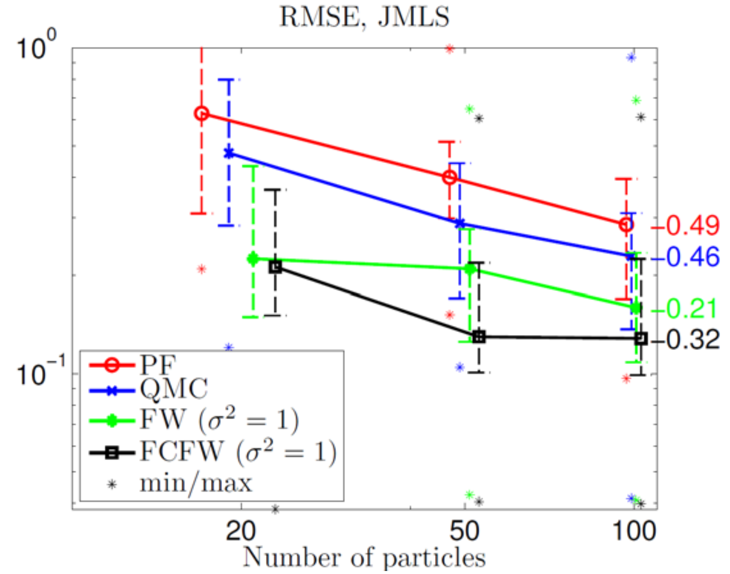
- Tools from optimization to help deterministic sampling!
- With FW-quadrature, getting each particle is more costly, but empirically, we need less particles to get a good error
  - -> this could be useful when evaluating $p(y_{t+1}|x_{t+1}^{(i)})$ is very expensive (e.g. in robot localization problem)
  - [e.g. 0.2 s for N=50 PF; overhead of 0.1 s for N=50 FW]
- Current work:
  - refine convergence theory
  - results somewhat sensitive to kernel bandwidth parameter -> find ways to adaptively choose it
  - understand better relationship between kernel and error propagation for class of functions
    - (e.g. introduce a kernel on past histories as well – changing $\mathcal{H}_t$)

Thank you! Any question?

Jump Markov Gaussian linear model results:

- RMSE computed on mean predicted position vs. good approximation from Rao-Blackwellized Discrete PF with 10k particles

$d = 2$, 3 modes, $\sigma^2 = 1$



Nonlinear 1d time series results: