Markov Chain Monte Carlo Algorithms for Discrete-Time Filtering

Avishy Carmi

Department of Mechanical and Aerospace Engineering

Nanyang Technological University, Singapore

Email: acarmi@ntu.edu.sg

Mila Mihaylova

School of Computing & Communications, InfoLab21

United Kingdom

Email: mila.mihaylova@lancaster.ac.uk

Outline

I. Group and extended object tracking. Related works

II. Sequential Monte Carlo Filtering

III. A Sequential Markov Chain Monte Carlo Algorithm for High Dimensional Problems

IV. Performance Evaluation

V. Conclusions and Future Works

High Dimensional Problems

- Group object tracking
  - small groups
    - Koch et all, RAS 2010: approach of random matrices
    - Khan, Balsh Dellaert (2005)
    - Mahler (2010), Gning, Mihaylova, Maskell, Pang, Godsill (2011), Carmi et al (2010)
  - groups of a large number of targets
    - Carmi, Septier, Godsill (2011)
  - extended target tracking
    - Baum and Hannebeck (2009), Angelova and Mihaylova (2008), Boers and Driesen (2006)
    - Baum, Feldmann, Franken, Hanebeck and Koch (2010)
    - Vermaak, Ikoma, Godsill (2005)

The Dynamic System Model

- **State transition (motion) equation**

$x_k = f(x_{k-1}, v_{k-1})$

$f(.)$: evolution function (possibly nonlinear)

$x_k, x_{k-1} \in \Re^{n_x}$: current and previous state

$v_{k-1} \in \Re^{n_v}$: system noise (usually non-Gaussian)

The state depends only on the previous step: i.e. first order Markov process

- **Measurement equation**

$z_k = h(x_k, r_k)$

$z_k \in \Re^{n_z}$: measurement

$h(.)$: measurement function (possibly nonlinear)

r_k : measurement noise (usually non-Gaussian)

Bayesian Framework for Nonlinear Estimation Problems

- Estimate the posterior state probability density function (PDF)

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$$

given the data set

$$z_{1:k} \triangleq \{z_1, \ldots, z_k\}$$

- The sensor information updates recursively the state distribution.

- **Prediction**:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \int_{\mathbb{R}^{n_x}} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) d\boldsymbol{x}_k$$

- **Update**:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})}{p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1},)}$$

Sequential Monte Carlo Methods

- Sample based approximation of $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, with a set of particles and their weights

- **Prediction step**:

$$p(x_{k\text{-}1}|z_{1:k}) \longrightarrow p(x_k|z_{1:k})$$

spreads the state PDF due to noise

- **Update step:**

given $p(x_k|z_{1:k\text{-}1})$ and z_k, find $p(x_k|z_{1:k})$

Usually concentrates the state PDF by combining the likelihood of current measurement with the predicted state.

- **Resampling step**

Markov Chain Monte Carlo Schemes

- Metropolis Hastings (MH) : given N samples at time k-1, $\{\boldsymbol{x}_{k-1}^{(i)}\}_{i=1}^{N}$ drawn from $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1})$
- Generate new samples $\{\boldsymbol{x}_k\}_{i=1}^{N}$ representing $p(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k})$

from an a periodic and irreversible Markov chain with a stationary distribution

- First, we simulate a sample $\boldsymbol{x}'_k$ from the joint PDF

$$p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1})$$

- by drawing

$$\boldsymbol{x}'_k \sim p(\boldsymbol{x}_k \mid \boldsymbol{x}'_{k-1})$$

- where $\boldsymbol{x}'_{k-1}$ is uniformly drawn from the empirical approximation

$$\hat{p}(\boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1}) = N^{-1} \sum_{i=1}^{N} \delta(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{x}_{k-1})$$

Other MCMC Particle Methods for High Dimensional Systems

- The MH algorithm accepts the new candidate $(\boldsymbol{x}'_k, \boldsymbol{x}'_{k-1})$

as the next realisation from the chain with probability

$$\alpha = \min\{1, p(\boldsymbol{z}_k \mid \boldsymbol{x}'_k)/p(\boldsymbol{z}_k \mid \boldsymbol{x}_k)\}$$

$$(\boldsymbol{x}_k^{(i+1)}, \boldsymbol{x}_{k-1}^{(i+1)}) = \begin{cases} (\boldsymbol{x}'_k, \boldsymbol{x}'_{k-1}), & \text{if } u \leq \alpha \\ (\boldsymbol{x}_k^{(i)}, \boldsymbol{x}_{k-1}^{(i)}), & \text{otherwise} \end{cases}$$

- MH, followed by Gibbs refinement: better mixing

- The new approach: use the (sub)gradient information of the likelihood. Very promising in high dimensional state spaces.

The Proposed Subgradient Projection (SP), SP-MCMC Particle Filter

- The samples $\boldsymbol{x}'_k$ are propagated through $p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1})$

and pushed towards high probability areas based on the subgradient of the likelihood

$$\bar{\boldsymbol{x}}_k^{(i)} = \boldsymbol{x}'^{(i)}_k - \lambda^{(i)} \frac{\log p(\boldsymbol{z}_k \mid \boldsymbol{x}'^{(i)}_k)}{\| t^{(i)} \|_2^2} t^{(i)}, \quad i = 1, \ldots, N$$

$\lambda^{(i)}$ relaxation parameter sampled for every i from some distribution, e.g. uniform. The associated (sub)gradient

$$t^{(i)} := \partial \log p(\boldsymbol{z}_k \mid \boldsymbol{x}_k)$$

with respect to $\boldsymbol{x}_k$ is computed at $\boldsymbol{x}'^{(i)}_k$

Improved Subgradient Projection MCMC Particle Filter

- Construct a regularised proposal based on

$$\bar{\mathbf{y}}_k^{(i)} = [(\bar{\mathbf{x}}_k^{(i)})^T, (\mathbf{x}_{k-1}^{(i)})^T]^T, \quad \bar{\mathbf{y}}_k = [\bar{\mathbf{x}}_k^T, \mathbf{x}_{k-1}^T]^T,$$

$$q(\bar{\mathbf{y}}_k) \propto \sum_{i=1}^{N} \mathcal{N}(\bar{\mathbf{y}}_k \mid \bar{\mathbf{y}}_k^{(i)}, \sigma^2)$$

- The new candidate pair is selected based on:

$$\alpha = \min\left\{1, \frac{p(\mathbf{z}_k \mid \bar{\mathbf{x}}_k)\hat{p}(\bar{\mathbf{x}}_k, \mathbf{x}'_{k-1} \mid \mathbf{z}_{1:k-1})q(\mathbf{y}_k^{(i)})}{p(\mathbf{z}_k \mid \mathbf{x}_k^{(i)})\hat{p}(\mathbf{x}_k^{(i)}, \mathbf{x}_{k-1}^{(i)} \mid \mathbf{z}_{1:k-1})q(\bar{\mathbf{y}}_k)}\right\}$$

The Proposed SP-MCMC Particle Filter

- A regularised proposal is constructed in the following way

$$q(\boldsymbol{x}_k) \propto \sum_{i=1}^{N} \mathcal{N}(\boldsymbol{x}_k \mid \bar{\boldsymbol{x}}_k^{(i)}, \sigma^2)$$

- The MH acceptance probability of the new candidate move $\bar{\boldsymbol{x}}_k \sim q(\boldsymbol{x}_k)$ is

$$\alpha = \min\left\{1, \frac{p(\boldsymbol{z}_k \mid \bar{\boldsymbol{x}}_k)\hat{p}(\bar{\boldsymbol{x}}_k, \boldsymbol{x}'_{k-1} \mid \boldsymbol{z}_{1:k-1})q(\boldsymbol{x}_k)}{p(\boldsymbol{z}_k \mid \boldsymbol{x}_k)\hat{p}(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} \mid z_{1:k-1})q(\bar{\boldsymbol{x}}_k)}\right\}$$

A much efficient approach targets the joint pdf of both the improved and propagated states (Pang and Godsill, 2011)

- S. K. Pang and J. Li and S. J. Godsill, Detection and Tracking of Coordinated Groups, IEEE Transactions on Aerospace and Electronic Systems, 2011

Relation to Langevin MCMC

- The intermediate subgradient projection step is similar to the Langevin random-walk-like MCMC (targeting the likelihood)

- We, however, allow the relaxation parameter to take negative values

- If we restrict the relaxation parameter to the interval (0,2) then convergence towards the mode is guaranteed for convex log likelihoods

MCMC Particle Filtering Algorithm 1

1: Given previous time samples $\mathbf{x}_{k-1}^{(i)}, \;\; i = 1, \ldots, N$ perform the following steps.

2: Draw $\mathbf{x}_k'^{(i)} \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}), \;\; i = 1, \ldots, N$.

3: Use (9) for producing $\bar{\mathbf{x}}_k^{(i)}, \quad i = 1, \ldots, N$. The set $\{\mathbf{x}_k'^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^N$ simulates $\hat{p}(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$, whereas $\{\bar{\mathbf{y}}_k^{(i)}\}_{i=1}^N$, $\bar{\mathbf{y}}_k^{(i)} = [(\bar{\mathbf{x}}_k^{(i)})^T, (\mathbf{x}_{k-1}^{(i)})^T]^T$ simulates $q(\bar{\mathbf{y}}_k) = \mathcal{N}(\bar{\mathbf{y}}_k \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ where

$$\boldsymbol{\mu}_k = N^{-1} \sum_{i=1}^{N} \bar{\mathbf{y}}_k^{(i)}, \quad \boldsymbol{\Sigma}_k = N^{-1} \sum_{i=1}^{N} \left[\bar{\mathbf{y}}_k^{(i)} - \boldsymbol{\mu}_k\right] \left[\bar{\mathbf{y}}_k^{(i)} - \boldsymbol{\mu}_k\right]^T$$

4: **for** i=1, ..., $N + N_{Burn-in}$ **do**

5: Draw $(\bar{\mathbf{x}}_k, \mathbf{x}_{k-1}) \sim q(\bar{\mathbf{y}}_k)$.

6: Accept the new move as a sample in the chain $\mathbf{x}_k^{(i)} = \bar{\mathbf{x}}_k$ with probability α given in (11).

7: **end for**

8: Retain only N samples $\mathbf{x}_k^{(i)}$ subsequent to the end of the burn-in period.

Alternating Steering MCMC Algorithm

1: Simulate $q_1(\bar{\mathbf{y}}_k)$ and $q_2(\bar{\mathbf{y}}_k)$ using two distinct steering distributions p_λ^1 and p_λ^2, respectively.
2: **for** i=1, $\ldots$, $N + N_{Burn-in}$ **do**
3: Draw $(\bar{\mathbf{x}}_k, \mathbf{x}_{k-1}) \sim q(\bar{\mathbf{y}}_k)$ where $q(\bar{\mathbf{y}}_k) = q_1(\bar{\mathbf{y}}_k)$ if $(i \bmod 2) = 1$, and $q(\bar{\mathbf{y}}_k) = q_2(\bar{\mathbf{y}}_k)$, otherwise.
4: Accept the new move as a sample in the chain $\mathbf{x}_k^{(i)} = \bar{\mathbf{x}}_k$ with probability α given in (11).
5: **end for**
6: Retain only N samples $\mathbf{x}_k^{(i)}$ subsequent to the end of the burn-in period.

Nonlinear Filtering Example

Example

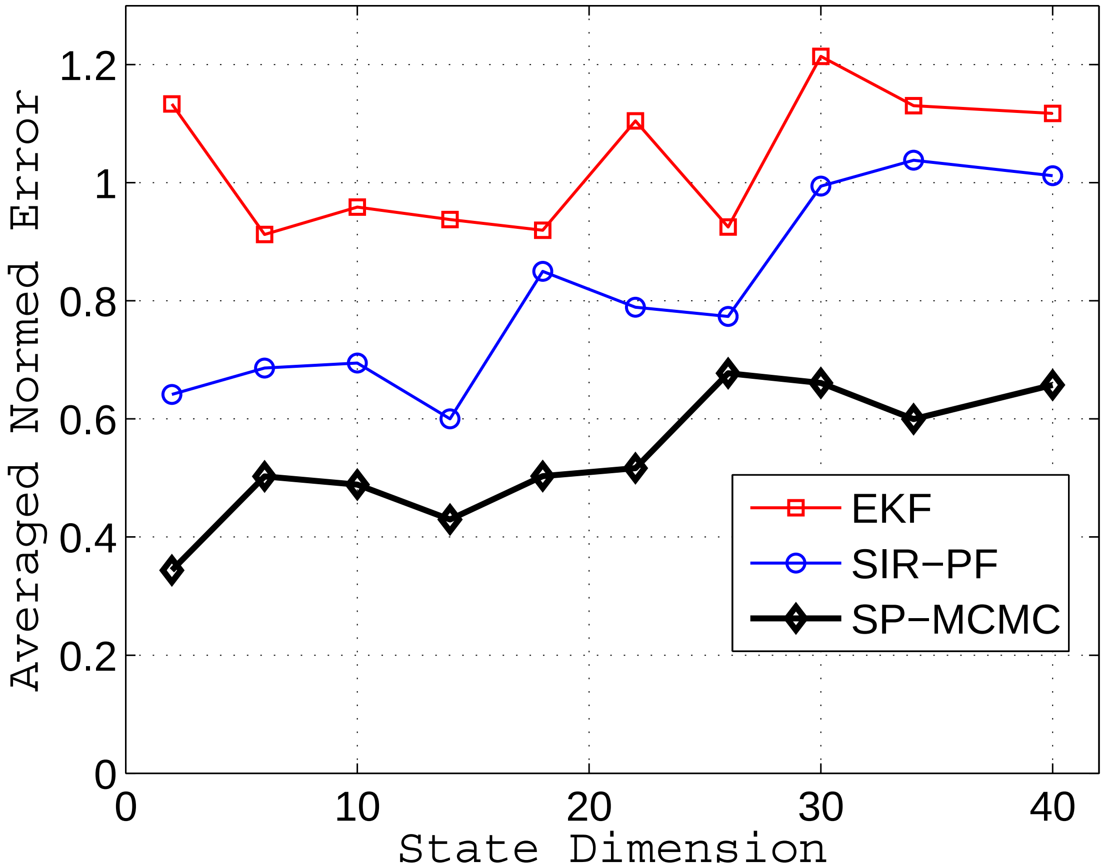
$$\boldsymbol{x}_k^j = \boldsymbol{x}_{k-1}^j + \frac{25\sum_{j=1}^{n_x}\boldsymbol{x}_{k-1}^j}{1+\left(\sum_{j=1}^{n_x}\boldsymbol{x}_{k-1}^j\right)^2} + \cos(1.2k) + \boldsymbol{v}_{k-1}^j$$

$$z_k^j = \frac{\left(\boldsymbol{x}_k^j\right)^2}{20} + r_k^j, \quad j = 1, \ldots, n_x$$

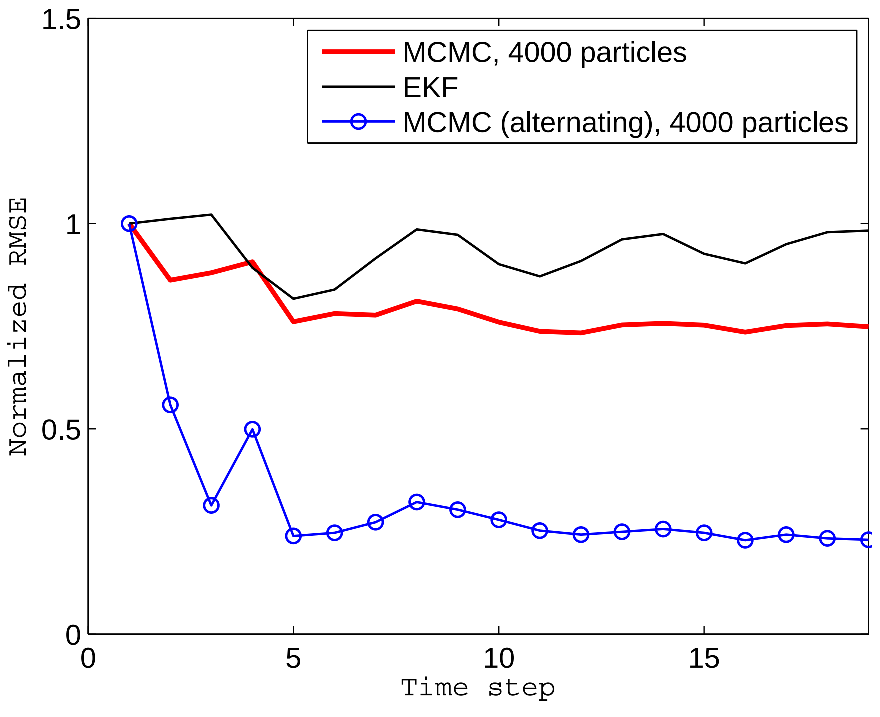
Curse of Dimensionality

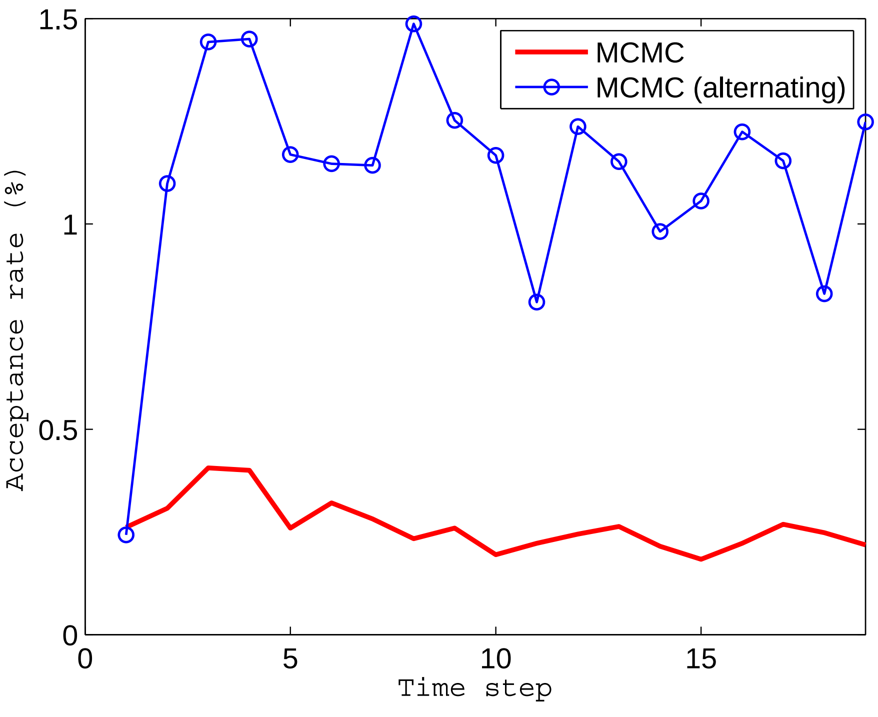
- The averaged normalised estimation errors (based on 100 runs)

$$e_k := \| \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \|_2 \; / \; \| \boldsymbol{x}_k \|_2$$



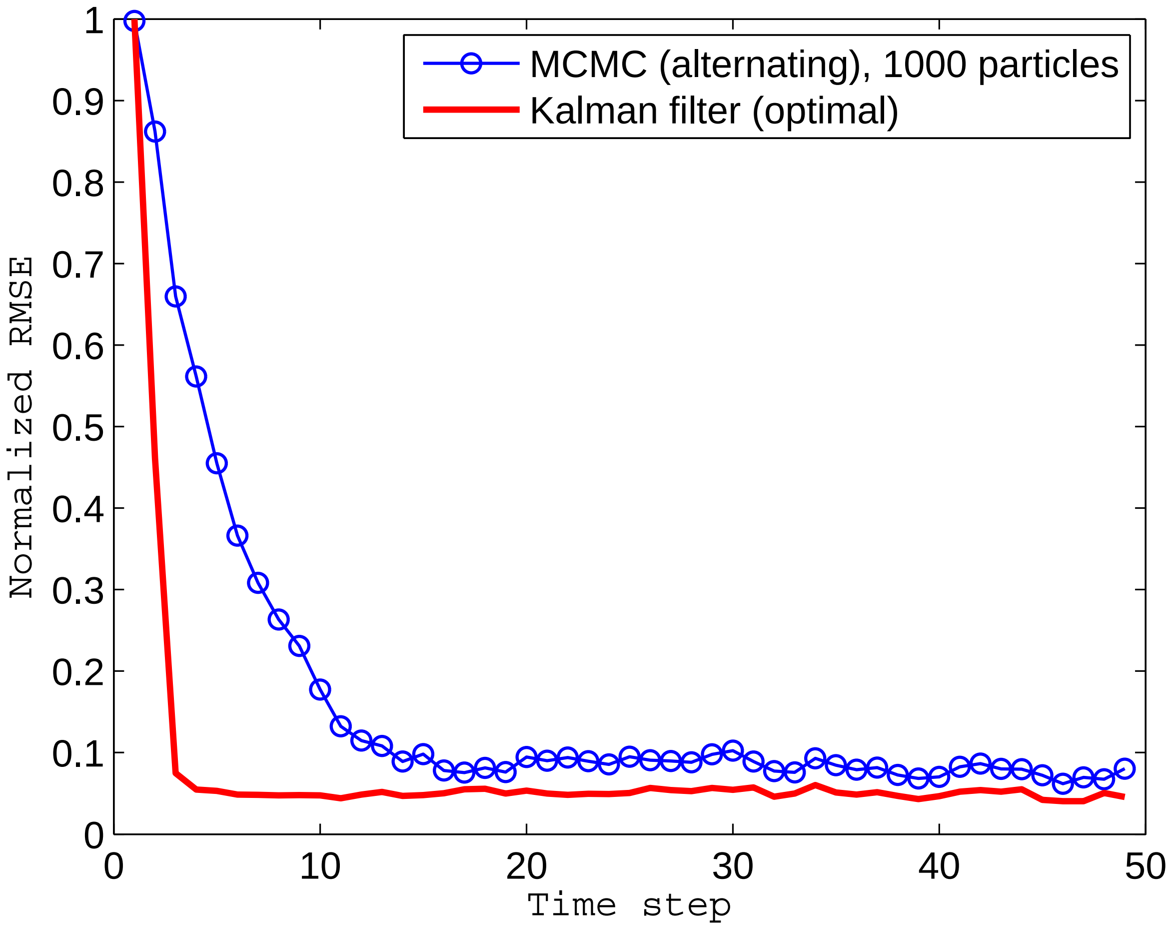
Filtering Performance – 100 states





Normalised RMSE (left) and mean acceptance rate (right)

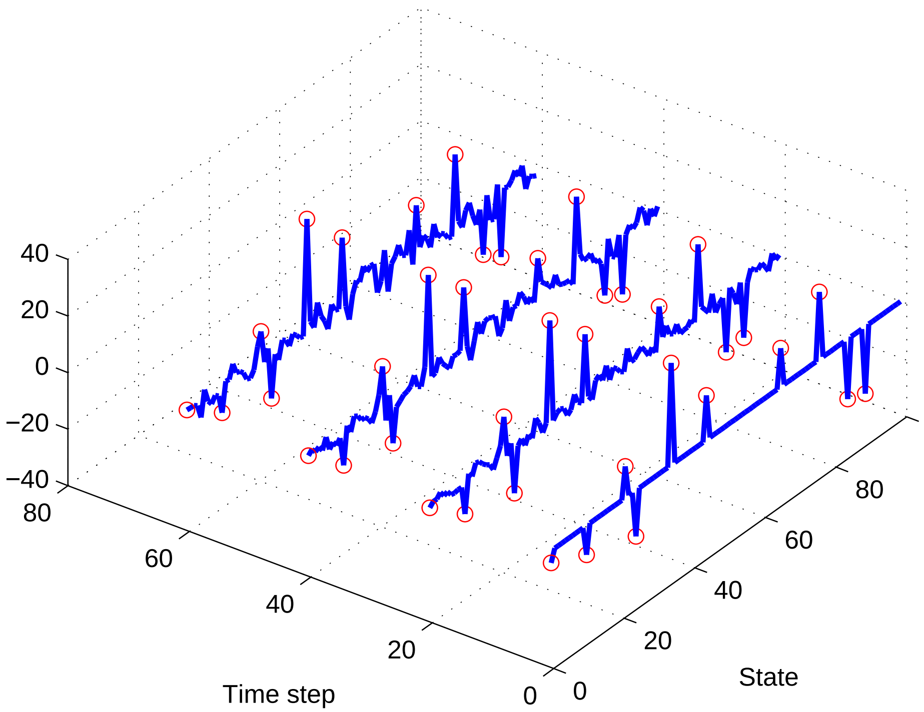
Linear Example – 100 states

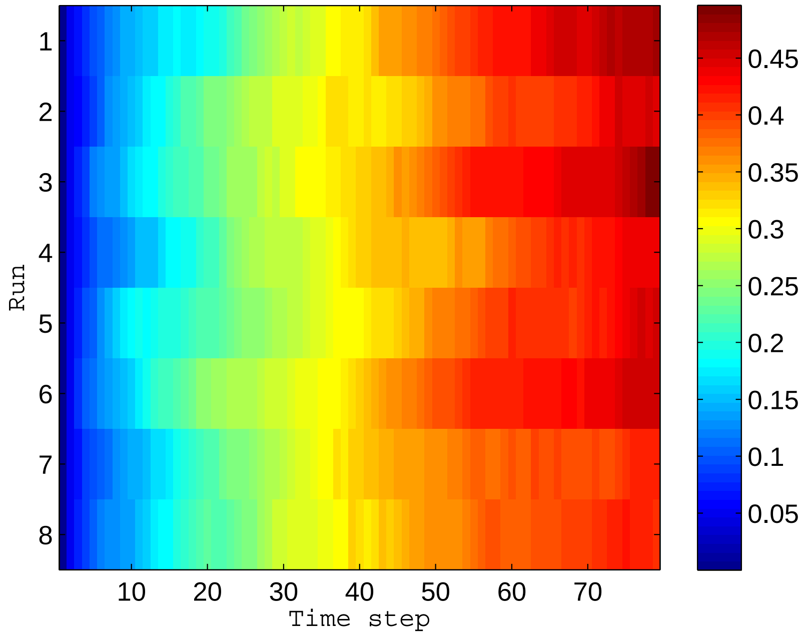


Dynamic Compressed Sensing

- Unobservable linear system with 100 states and 50 observations per time step. The signal is nearly sparse with 10 significant entries.
- The complexity rises with time as the signal becomes less compressible due to accumulated noise.
- Regular Kalman filter fails to work as the system is non estimable.
- We compare the SP-MCMC (which employs an L1-constrained likelihood) to the CSKF of Carmi et al. (2010)
- Carmi, A.; Gurfil, P.; Kanevsky, D.; , "Methods for Sparse Signal Recovery Using Kalman Filtering With Embedded Pseudo-Measurement Norms and Quasi-Norms," Signal Processing, IEEE Transactions on , vol.58, no.4, pp.2405-2409, April 2010

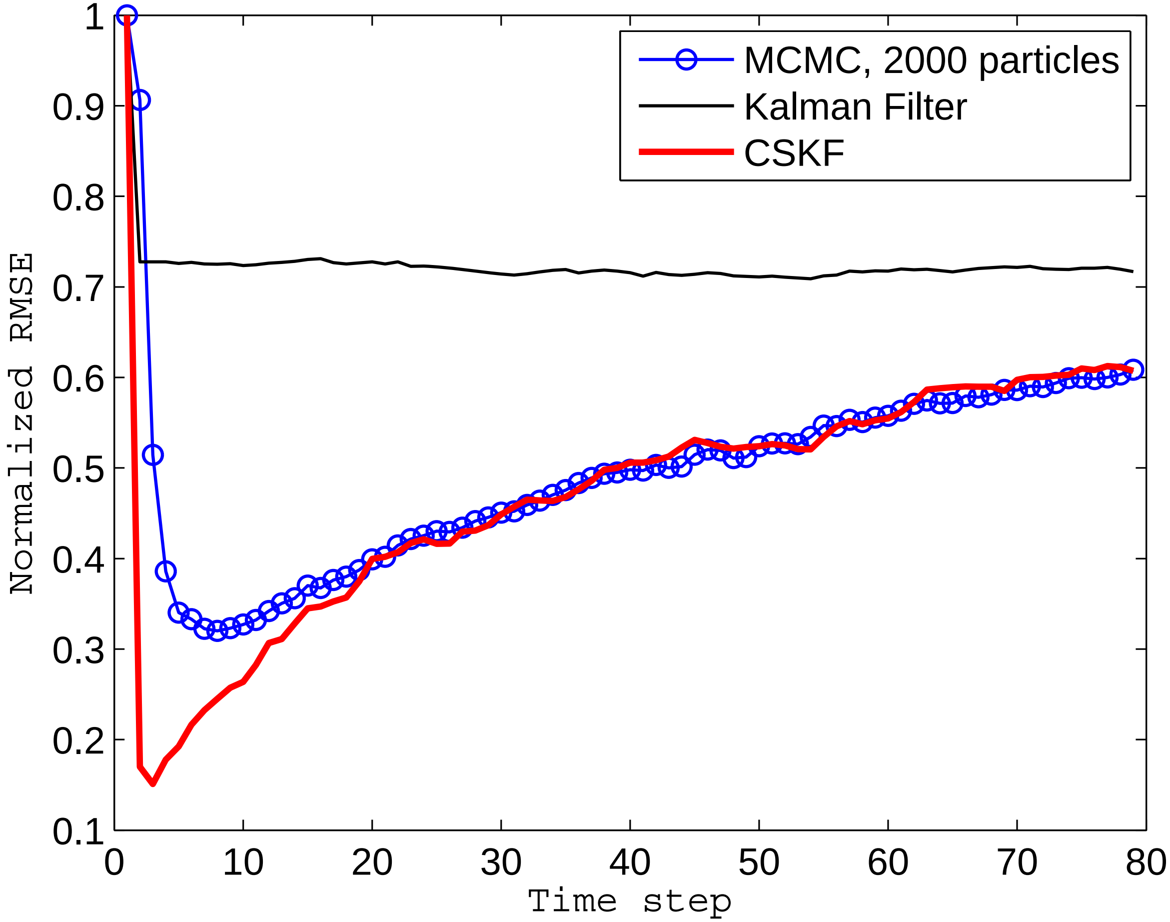
Dynamic Compressible Signal





Evolution of a compressible signal (left) and a measure of complexity in 8 samples runs (right). From perfectly sparse (blue) to compressible (red)

Filtering Performance – 100 states



Conclusions and Future Plans

- A new MCMC approach for state estimation using sub-gradient information is developed
- New proposal function constructed, based on samples from more likely regions
- A Metropolis Hasting step refines further the particles
- High accuracy achieved
- Future work: on extended target tracking and group tracking for more complex scenarios

Thank you for your attention ☺ !