Introduction to Sequential Monte Carlo Methods

Sequential Importance Sampling (SIS)
Sequential Importance Sampling with Resampling (SISR)
State Space Model
Degeneracy, Convergence and Error Estimates

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

November 5, 2018

Contents

References

- C.P. Robert & G. Casella, Monte Carlo Statistical Methods, Chapter 11
- J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.
- A. Doucet, N. De Freitas & N. Gordon (eds), Sequential Monte Carlo in Practice, Springer-Verlag: 2001
- A. Doucet, N. De Freitas, N.J. Gordon, An introduction to Sequential Monte Carlo, in SMC in Practice, 2001
- D. Wilkison, Stochastic Modelling for Systems Biology, Second Edition, 2006
- E. Ionides, Inference for Nonlinear Dynamical Systems, PNAS, 2006
- J.S. Liu and R. Chen, Sequential Monte Carlo methods for dynamic systems, JASA, 1998
- A. Doucet, Sequential Monte Carlo Methods, Short Course at SAMSI
- A. Doucet, Sequential Monte Carlo Methods & Particle Filters Resources
- Pierre Del Moral, Feynman-Kac models and interacting particle systems (SMC resources)
- A. Doucet, Sequential Monte Carlo Methods, Video Lectures, 2007
- N. de Freitas and A. Doucet, Sequential MC Methods, N. de Freitas and A. Doucet, Video Lectures, 2010

References

- M.K. Pitt and N. Shephard, Filtering via Simulation: Auxiliary Particle Filter, JASA, 1999
- A. Doucet, S.J. Godsill and C. Andrieu, On Sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comp., 2000
- J. Carpenter, P. Clifford and P. Fearnhead, An Improved Particle Filter for Non-linear Problems, IEE 1999.
- A. Kong, J.S. Liu & W.H. Wong, Sequential Imputations and Bayesian Missing Data Problems, JASA, 1994
- O. Cappe, E. Moulines & T. Ryden, Inference in Hidden Markov Models, Springer-Verlag, 2005
- W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models, JRSS B, 2001
- G. Poyadjis, A. Doucet and S.S. Singh, Maximum Likelihood Parameter Estimation using Particle Methods, Joint Statistical Meeting, 2005
- N Gordon, D J Salmond, AFM Smith, Novel Approach to nonlinear non Gaussian Bayesian state estimation, IEE, 1993
- Particle Filters, S. Godsill, 2009 (Video Lectures)
- R. Chen and J.S. Liu, Predictive Updating Methods with Application to Bayesian Classification, JRSS B, 1996

References

- C. Andrieu and A. Doucet, Particle Filtering for Partially Observed Gaussian State-Space Models, JRSS B, 2002
- R Chen and J Liu, Mixture Kalman Filters, JRSSB, 2000
- A Doucet, S J Godsill, C Andrieu, On SMC sampling methods for Bayesian Filtering, Stat. Comp. 2000
- N. Kantas, A.D., S.S. Singh and J.M. Maciejowski, An overview of sequential Monte Carlo methods for parameter estimation in general state-space models, in Proceedings IFAC System Identification (SySid) Meeting, 2009
- C. Andrieu, A.Doucet & R. Holenstein, Particle Markov chain Monte Carlo methods, JRSS B, 2010
- C. Andrieu, N. De Freitas and A. Doucet, Sequential MCMC for Bayesian Model Selection, Proc. IEEE Workshop HOS, 1999
- P. Fearnhead, MCMC, sufficient statistics and particle filters, JCGS, 2002
- G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, IEEE Trans. Signal Processing, 2002

References

- C. Andrieu, A. Doucet and V.B. Tadic, Online EM for parameter estimation in nonlinear-non Gaussian state-space models, Proc. IEEE CDC, 2005
- G. Poyadjis, A. Doucet and S.S. Singh, Particle Approximations of the Score and Observed Information Matrix in State-Space Models with Application to Parameter Estimation, *Biometrika*, 2011
- C. Caron, R. Gottardo and A. Doucet, On-line Changepoint Detection and Parameter Estimation for Genome Wide Transcript Analysis, Technical report 2008
- R. Martinez-Cantin, J. Castellanos and N. de Freitas. Analysis of Particle Methods for Simultaneous Robot Localization and Mapping and a New Algorithm: Marginal-SLAM. International Conference on Robotics and Automation
- C. Andrieu, A.D. & R. Holenstein, Particle Markov chain Monte Carlo methods (with discussion), JRSS B, 2010
- A Doucet, Sequential Monte Carlo Methods and Particle Filters, List of Papers, Codes, and Viedo lectures on SMC and particle filters
- Pierre Del Moral, Feynman-Kac models and interacting particle systems

References

- P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo samplers, JRSSB, 2006
- P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo for Bayesian Computation, Bayesian Statistics, 2006
- P. Del Moral, A. Doucet & S.S. Singh, Forward Smoothing using Sequential Monte Carlo, technical report, Cambridge University, 2009
- P. Del Moral, Feynman-Kac Formulae, Springer-Verlag, 2004
- Sequential MC Methods, M. Davy, 2007
- A Doucet, A Johansen, Particle Filtering and Smoothing: Fifteen years later, in Handbook of Nonlinear Filtering (edts D Crisan and B. Rozovsky), Oxford Univ. Press, 2011
- A. Johansen and A. Doucet, A Note on Auxiliary Particle Filters, Stat. Proba. Letters, 2008.
- A. Doucet et al., Efficient Block Sampling Strategies for Sequential Monte Carlo, (with M. Briers & S. Senecal), JCGS, 2006.
- C. Caron, R. Gottardo and A. Doucet, On-line Changepoint Detection and Parameter Estimation for Genome Wide Transcript Analysis, Stat Comput. 2011.

Importance Sampling and its Application to Nonlinear Non-Gaussian Dynamic Models

Review: Importance Sampling

- Our goal is to compute an expectation value of the form :

$$\mathbb{E}_{\pi}\left[f(\boldsymbol{x}) \right] = \int_{A} f(\boldsymbol{x}) \pi(\boldsymbol{x}) d\boldsymbol{x}$$

where $\pi(\boldsymbol{x})$ is a probability distribution (posterior inference in Bayesian models, Bayesian model validation, etc.)

- We assume that $\pi(x) = \frac{\gamma(x)}{Z}$ where $Z = \int \gamma(x) dx$ is unknown and γ is known pointwise.

- The basic idea in Monte Carlo methods is to sample N i.i.d. random numbers $\boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$ and build an empirical measure

$$\hat{\pi}(\boldsymbol{x}) d\boldsymbol{x} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{X}^{(i)}} d\boldsymbol{x}$$

- Using this:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \frac{1}{N} \sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) \quad , \quad where \; \boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$$

J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.

Monte Carlo Methods

- Using the approximation of π:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) \quad , \quad where\, \boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$$

- The following hold:

$$\mathbb{E}[\mathbb{E}_{\hat{\pi}}(f)] = \mathbb{E}_{\pi}(f)\,,\, V[\mathbb{E}_{\hat{\pi}}(f)] = \frac{1}{N}\mathbb{E}_{\pi}\left(\left(f - \mathbb{E}_{\pi}(f)\right)^2\right),\, \sqrt{N}\left(\mathbb{E}_{\hat{\pi}}(f) - \mathbb{E}_{\pi}(f)\right) \sim \mathcal{N}\left(0, \mathbb{E}_{\pi}\left(\left(f - \mathbb{E}_{\pi}(f)\right)^2\right)\right)$$

- Similarly, marginalization is also simple:

$$\hat{\pi}(x_p)dx_p = \int \hat{\pi}(x_1, x_2, \ldots, x_n) d\boldsymbol{x}_{1:p-1}\, d\boldsymbol{x}_{p+1:n} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_p^{(i)}} dx_p$$

- In MC, the samples automatically concentrate in regions of high probability mass regardless of the dimension of the space.
- However, it is not always easy or effective to sample from the original probability distribution $\pi(\boldsymbol{x})$. A more effective strategy is to focus on the regions of "importance" in $\pi(\boldsymbol{x})$ so as to save computational resources.

J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.

Review: Importance Sampling

- We assume that $\pi(\boldsymbol{x})$ is only known up to a normalizing constant:

$$\pi(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{Z}$$

- For any distribution $q(\boldsymbol{x})$ such that $\pi(\boldsymbol{x})>0 \Rightarrow q(\boldsymbol{x})>0$, we can write:

$$\pi(\boldsymbol{x}) = \frac{w(\boldsymbol{x})q(\boldsymbol{x})}{\underbrace{\int w(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x}}_{Z}} = \frac{w(\boldsymbol{x})q(\boldsymbol{x})}{Z}, \text{ where } w(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})}$$

- The proposal distribution $q(\boldsymbol{x})$ is known as "importance density" or "trial density". $w(\boldsymbol{x})$ is called the importance weight.

- The importance density can be chosen arbitrarily as any proposal easy to sample from:

$$\boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} q(\boldsymbol{x}) \Rightarrow \hat{\boldsymbol{q}}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})$$

Review: Importance Sampling

- Substitution of $\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})$ in the importance sampling identity gives:

$$\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \frac{w(\boldsymbol{x})\hat{q}(\boldsymbol{x})}{\int w(\boldsymbol{x})\hat{q}(\boldsymbol{x})d\boldsymbol{x}}d\boldsymbol{x} = \frac{\frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right)\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})}{\frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{X}^{(i)})} = \sum_{i=1}^{N} W^{(i)}\,\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x}),$$

$$where\ W^{(i)} \propto w\left(\boldsymbol{X}^{(i)}\right)\ and \sum_{i=1}^{N} W^{(i)} = 1$$

- Similarly, we can approximate the normalization factor of our target distribution as follows:

$$\hat{Z} = \int \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})}\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \int w(\boldsymbol{x})\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right) = \frac{1}{N}\sum_{i=1}^{N} \frac{\gamma\left(\boldsymbol{X}^{(i)}\right)}{q(\boldsymbol{X}^{(i)})}$$

Review: Importance Sampling

$$\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{N} W^{(i)}\,\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x}),\ where\ W^{(i)} \propto w\left(\boldsymbol{X}^{(i)}\right)\ and\ \sum_{i=1}^{N} W^{(i)} = 1$$

- The distribution $\pi(\boldsymbol{x})$ is now approximated by a weighted sum of delta masses, where the weights compensate for the discrepancy between $\pi(\boldsymbol{x})$ and $q(\boldsymbol{x})$.

Review: Importance Sampling

- Similarly calculation of $\mathbb{E}_{\pi}[f(\boldsymbol{x})]$ using importance sampling gives:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \int_A f(\boldsymbol{x})\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) W^{(i)}$$

- The statistics of this estimate are given for *N>>1* as follows:

$$\mathbb{E}_{\pi}\left[\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})]\right] = \mathbb{E}_{\pi}[f(\boldsymbol{x})] - \frac{1}{N_{\pi}}\mathbb{E}_{\pi}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])]$$

$$V\left[\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})]\right] = \frac{1}{N_{\pi}}\mathbb{E}_{\pi}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])^2]$$

where as discussed in an earlier lecture we have some negligible bias:

$$\frac{1}{N_{\pi}}\mathbb{E}_{\pi}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])]$$

Statistics of the Normalization Constant

- We can similarly compute the statistics of the normalization constant:

$$\hat{Z} = \int \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})} \hat{q}(\boldsymbol{x}) d\boldsymbol{x} = \frac{1}{N} \sum_{i=1}^{N} \frac{\gamma\left(\boldsymbol{X}^{(i)}\right)}{q\left(\boldsymbol{X}^{(i)}\right)} = \frac{1}{N} \sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right)$$

- They are given as:

$$\mathbb{E}\left[\hat{Z}\right] = Z, \; and$$

$$V\left[\hat{Z}\right] = \frac{1}{N} \mathbb{E}_q \left[\left(\frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})} - Z \right)^2 \right]$$

Review: Importance Sampling

- We select $q(\boldsymbol{x})$ *as close as possible to* $\pi(\boldsymbol{x})$.

- The variance of the weights is bounded iff

$$\int \frac{\gamma^2(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} < \infty$$

- In practice, it is sufficient to ensure that the weights are bounded:

$$w(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})} < \infty$$

- This is equivalent to saying that $q(\boldsymbol{x})$ should have heavier tails than $\pi(\boldsymbol{x})$.

Monte Carlo for the State Space Model

- We are interested to estimate the high-dimensional density

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{p(\boldsymbol{y}_{1:n})} \propto p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})$$

- Start with a fixed n. A Monte Carlo approximation (empirical measure) of our target distribution is of the form:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}), \; where \; \boldsymbol{X}_{1:n}^{(i)} \sim p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

- For any function $\varphi(\boldsymbol{x}_{1:n}): \mathcal{X}^n \to \mathbb{R}$, we can use a Monte Carlo approximation of its expectation as:

$$\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) = \int_{\mathcal{X}^n} \varphi(\boldsymbol{x}_{1:n})\, \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n}$$

$$= \int_{\mathcal{X}^n} \varphi(\boldsymbol{x}_{1:n}) \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n} = \frac{1}{N}\sum_{i=1}^{N} \varphi\left(\boldsymbol{X}_{1:n}^{(i)}\right)$$

Monte Carlo for the State Space Model

- This earlier estimate is asymptotically consistent (converges towards $\mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$).

- The estimate is unbiased and its variance gives the following convergence properties:

$$Var_{\boldsymbol{X}_{1:n}^{(i)}}\left[\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right] = \frac{1}{N} Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$$

$$\sqrt{N}\left(\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) \xrightarrow{d} \mathcal{N}\left(0, Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right)$$

- The rate of convergence is independent of *n*. This does not imply that Monte Carlo bits the curse of dimensionality since it is possible that $Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$ increases (with time) *n*.

Monte Carlo for the State Space Model

- The Monte Carlo approximation can easily be used to compute any marginal distribution, e.g. $p(x_k \mid \boldsymbol{y}_{1:n})$

$$\hat{p}_N(x_k|\boldsymbol{y}_{1:n}) = \int_{\mathcal{X}^{n-1}} \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+1:n}$$

$$= \int_{\mathcal{X}^{n-1}} \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+1:n}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}}(x_k)$$

- Note that the marginal likelihood $p(\boldsymbol{y}_{1:n})$ cannot be estimated as easily using $\boldsymbol{X}_{1:n}^{(i)} \sim p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

Difficulties with Standard Monte Carlo Sampling

- It is difficult to sample from our target high-dimensional distribution:

$$\boldsymbol{X}_{1:n}^{(i)} \sim p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

- MCMC methods are not useful in this context.

- As *n* increases, we would like to be able to sample from $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ with an algorithm that keeps the computational cost fixed at each time step *n*.

Importance Sampling for our State Space Model

- Rather than sampling directly from our target distribution $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$, we should sample from an importance distribution $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$
- Note that in the notation here for q, $\boldsymbol{y}_{1:n}$ is used as a parameter – not to indicate any posterior distribution.
- The importance distribution needs to satisfy the following properties:
  - The support of $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ includes the support of $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ i.e.

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) > 0 \Rightarrow q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) > 0$$

  - It is easy to sample from $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$
- We use the following identity:

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{\int p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n}} = \frac{\left[p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) / q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\right] q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})}{\int \left[p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) / q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\right] q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n}}$$

$$= \frac{w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})}{\int w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n}}$$

Importance Sampling for our State Space Model

- Let us draw *N* samples from our importance distribution:

$$\boldsymbol{X}_{1:n}^{(i)} \sim q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}),\ \hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- Then using the identity in the earlier slide, we obtain the following approximation of our target distribution:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{\int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})d\boldsymbol{x}_{1:n}}$$

$$= \frac{w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})}{\int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n}}$$

$$= \sum_{i=1}^{N} W_n^{(i)}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),\ W_n^{(i)} = \frac{w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}{\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}$$

- Note that: $\hat{p}_N(\boldsymbol{y}_{1:n}) = \int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n} = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)$

Normalized Weights in Importance Sampling

- We defined earlier the unnormalized weights as follows:

$$Unnormalized\ weights: w\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) = \frac{p\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right)}{q\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)} = p\left(\boldsymbol{y}_{1:n}\right) \underbrace{\frac{p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}{q\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}}_{\substack{\text{Discrepancy between} \\ \text{target distribution and} \\ \text{importance distribution}}}$$

- The normalized weights were also introduced as:

$$Normalized\ weights: W_n^{(i)} = \frac{w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)}{\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)}$$

Optimal Importance Sampling Distribution

- $\hat{p}_N(\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)$ is an unbiased estimate of $p(\boldsymbol{y}_{1:n})$ with variance:

$$\frac{1}{N}\left[\int w^2(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n} - 1\right]\left(p(\boldsymbol{y}_{1:n})\right)^2$$

- You can bring this variance to zero with the selection

$$q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$$

  Of course this is what we wanted to avoid (we want to sample from an easier distribution).

- However, this result points to the fact that the choice of q needs to be as close as possible to the target distribution.

Importance Sampling Estimates

- We are interested in an importance sampling approximation of $\mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$.

$$\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) = \sum_{i=1}^{N} W_n^{(i)} \varphi\left(\boldsymbol{X}_{1:n}^{(i)}\right)$$

- This is a biased estimate for a finite *N* and we have shown in our earlier lecture on Importance Sampling that:

$$\lim_{N\to\infty} N\left[\mathbb{E}_{p_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right] = -\int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(\varphi(\boldsymbol{x}_{1:n}) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) d\boldsymbol{x}_{1:n}$$

$$\sqrt{N}\left(\mathbb{E}_{p_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) \xrightarrow{d} \mathcal{N}\left(0, \int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(\varphi(\boldsymbol{x}_{1:n}) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right)^2 d\boldsymbol{x}_{1:n}\right)$$

- The asymptotic bias is of the order *1/N* (negligible) and the MSE error is:

$$MSE = \underbrace{bias^2}_{O\left(N^{-2}\right)} + \underbrace{variance}_{O\left(N^{-1}\right)}$$

Selection of Importance Sampling Distribution

- As discussed before, the importance sampling distribution should be selected so that the weights are bounded or equivalently $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ has heavier tails than $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

$$w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) \leq C \; \forall \boldsymbol{x}_{1:n} \in \mathcal{X}^n$$

- To minimize the asymptotic bias, we aim for $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ that is as close as possible to $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

- Note that the selection of the importance sampling needs to be not only such that it covers the support of the target but also needs to be a clever one for the particular problem of interest.

- For numerical examples and MatLab implementations please see an earlier lecture on importance sampling.

Effective Sample Size

- In our importance sampling approximation from the target $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ using the importance distribution $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ (for a fixed n), we would like ideally to have

$$q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$$

- In this case, all the unnormalized importance weights will be equal and their variance equal to zero.

- To access the quality of the importance sampling approximation, note that for flat functions,

$$\frac{Variance\ of\ \ IS\ estimate}{Variance\ of\ \ Standard\ \ MC\ estimate} \approx 1 + Var_{q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})} w(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n})$$

- This is often interpreted as the effective sample size (N weighted samples from $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ are approximately equivalent to M unweighted samples from $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$)

$$M = \frac{N}{1 + Var_{q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})} w(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n})} \leq N$$

Effective Sample Size

- We often approximate the effective sample size *M* as follows:

$$ESS = \left(\sum_{i=1}^{N} W_n^{(i)2} \right)^{-1}$$

since

$$Var_{q(x_{1:n}|y_{1:n})} W\left(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{y}_{1:n} \right) \approx N \sum_{i=1}^{N} W^2 \left(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{y}_{1:n} \right) - 1$$

- We clearly can see that

$$1 \leq ESS = \left(\sum_{i=1}^{N} W_n^{(i)2} \right)^{-1} \leq N$$

- We can thus have
  - ESS=1 (one of the weights equal to 1, all other zero, very inefficient) to
  - ESS=*N* (all weights equal to *1/N*, excellent sampling).

Sequential Importance Sampling

- Let us return to our state space model and consider a sequential Monte Carlo approximation of $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) \propto p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})$

- The distributions $\{\pi_n = p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\}$ are known up to a normalizing constant:

$$\pi_n(\boldsymbol{x}_{1:n}) = \frac{\gamma_n(\boldsymbol{x}_{1:n})}{Z_n} = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{Z_n}$$

- We want to estimate the expectations of functions $f_n : \mathcal{X}^n \to \mathbb{R}$

$$\mathbb{E}_{\pi_n}(\varphi_n) = \int \varphi_n(\boldsymbol{x}_{1:n}) \pi_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n}$$

and/or the normalizing constants Z_n.

- One can use MCMC to sample from $\{\pi_n\}, n = 1, 2..$ This calculation will be slow and cannot compute

$$\{Z_n\}, n = 1, 2..$$

Sequential Importance Sampling

- We want to do these calculations sequentially starting with π_1 and Z_1 at step (time 1), then proceeding to π_2 and Z_2, etc.

- Sequential Monte Carlo (SMC) provides the means to do so as an alternative algorithm to MCMC.

The key idea is that if π_{n-1} does not differ a lots from π_n, we should be able to reuse our estimate of π_{n-1} to approximate π_n.

Sequential Importance Sampling

- We want to design a sequential importance sampling method to approximate

$$\{\pi_n\}_{n\geq 1} \text{ and } \{Z_n\}_{n\geq 1}$$

- Assume that `at time 1', we have approximations $\hat{\pi}_1(x_1) = \hat{p}_N(x_1|y_1)$, $\hat{Z}_1$ using an importance density $q_1(x_1 \mid y_1)$.

$$X_1^{(i)} \sim q_1(x_1|y_1),\ i = 1,2,\ldots,N$$

$$\hat{p}_N(x_1|y_1)dx_1 = \sum_{i=1}^{N} W_1^{(i)}\, \delta_{X_1^{(i)}}(dx_1),\ where\ W_1^{(i)} = \frac{w_1\left(X_1^{(i)}, y_1\right)}{\sum_{j=1}^{N} w_1\left(X_1^{(j)}, y_1\right)}$$

$$\hat{Z}_1 = \frac{1}{N}\sum_{i=1}^{N} w_1\left(X_1^{(i)}, y_1\right)\ with$$

$$w_1(x_1, y_1) = \frac{\gamma_1(x_1)}{q_1(x_1|y_1)} = \frac{p(x_1, y_1)}{q_1(x_1|y_1)}$$

Sequential Importance Sampling

- At `time 2', we want to approximate $\hat{\pi}_2(\boldsymbol{x}_{1:2}) = \hat{p}_N(\boldsymbol{x}_{1:2}|\boldsymbol{y}_{1:2})$, $\hat{Z}_2$ using an importance density $q_2\left(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2}\right)$.

- We want to reuse the samples $\boldsymbol{X}_1^{(i)}$ and $q_1(x_1|y_1)$ in building the importance sampling approximation for $\pi_2\left(\boldsymbol{x}_{1:2}\right), Z_2$.

- Let us select $q_2\left(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2}\right) = q_1\left(x_1 \mid y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)$

- To obtain $\boldsymbol{X}_{1:2}^{(i)} \sim q_2\left(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2}\right)$ we need to sample as follows:

$$X_2^{(i)} \mid X_1^{(i)} \sim q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, X_1^{(i)}\right)$$

- The importance sampling weight for this step is then:

$$w_2\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right) = \frac{\gamma_2\left(\boldsymbol{x}_{1:2}\right)}{q_2\left(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2}\right)} = \frac{p\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right)}{q_1\left(x_1 \mid y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)} =$$

$$= \frac{p\left(x_1, y_1\right)}{q_1\left(x_1 \mid y_1\right)} \frac{p\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right)}{p\left(x_1, y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)} = \underbrace{w_1\left(x_1, y_1\right)}_{\text{Weight from step 1}} \underbrace{\frac{p\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right)}{p\left(x_1, y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)}}_{\text{Incremental weight}}$$

Sequential Importance Sampling

- The normalized weights for step 2 are then given as:

$$W_2^{(i)} \propto w_2\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right) = \underbrace{w_1\left(x_1, y_1\right)}_{\text{Weight from step 1}} \underbrace{\frac{p\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right)}{p\left(x_1, y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)}}_{\text{Incremental weight}}$$

- Generalizing to step n, we can write:

$$q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) = q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) q_n\left(x_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{x}_{1:n-1}\right)$$

$$= q_1\left(x_1 \mid y_1\right) \prod_{k=2}^{n} q_k\left(x_k \mid \boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}\right)$$

- Thus if

$$\boldsymbol{X}_{1:n-1}^{(i)} \sim q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)$$

we sample X_n from

$$X_n^{(i)} \mid \boldsymbol{X}_{1:n-1}^{(i)} \sim q_n\left(x_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{X}_{1:n-1}^{(i)}\right)$$

Sequential Importance Sampling

- The weights for step *n* are then given as:

$$w_n(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n})}{q_n(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{y}_{1:n})} = \underbrace{\frac{p(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1})}{q_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)} \mid \boldsymbol{y}_{1:n-1})}}_{w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1})} \frac{p(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n})}{p(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}) q_n(X_n^{(i)} \mid \boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n})}$$

$$= w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}) \frac{p(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n})}{p(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}) q_n(X_n^{(i)} \mid \boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n})}$$

- Similarly the normalized weights are as follows:

$$W_n^{(i)} \equiv W_n(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}) \propto w_n(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n})$$

- For our state space model, the above update formula takes the form:

$$w_n(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}) = w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}) \frac{f(X_n^{(i)} \mid X_{n-1}^{(i)}) g(y_n \mid X_n^{(i)})}{q_n(X_n^{(i)} \mid \boldsymbol{y}_{1:n}, \boldsymbol{X}_{1:n-1}^{(i)})}$$

- In general, we may need to store all the paths $\left\{\boldsymbol{X}_{1:n}^{(i)}\right\}$ even if our interest is to only compute $\pi_n(x_n) = p(x_n \mid \boldsymbol{y}_{1:n})$

Need for a Sequential Sampling Approach

- From practical perspective, we use proposal distributions of the form:

$$q_n\left(x_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{x}_{1:n-1}\right) = q_n\left(x_n \mid y_n, x_{n-1}\right)$$

- The idea here is that given x_{n-1}, $\boldsymbol{y}_{1:n-1}$ and $\boldsymbol{x}_{1:n-2}$ don't bring any new information about x_n.

- Our sequential importance sampling update now looks as follows:

$$\underbrace{q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}_{\text{Importance Samping at } n} = \underbrace{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)}_{\text{Distribution of the paths } \boldsymbol{X}_{1:n-1}^{(i)}} \underbrace{q_n\left(x_n \mid y_n, x_{n-1}\right)}_{\text{Conditional Distribution of } X_n^{(i)}}$$

$$= q\left(x_1\right)\prod_{k=2}^{n} q_k\left(x_k \mid y_k, x_{k-1}\right)$$

- Thus we assume that at $n-1$ we have sampled $\boldsymbol{X}_{1:n-1}^{(i)} \sim q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)$ and to obtain $\boldsymbol{X}_{1:n}^{(i)} \sim q\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$, we need to sample $X_n^{(i)} \sim q_n\left(x_n \mid y_n, X_{n-1}^{(i)}\right)$ and then set

$$\boldsymbol{X}_{1:n}^{(i)} = \left(\underbrace{\boldsymbol{X}_{1:n-1}^{(i)}}_{\text{Previously Sampled Paths}}, \underbrace{X_n^{(i)}}_{\text{Sampled Single Component at time } n} \right)$$

Sequential Importance Sampling

- We now need to show how we can recursively compute estimates of our target distribution $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ as well as of $p(\boldsymbol{y}_{1:n})$

- From our earlier Importance Sampling approximations:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),\ W_n^{(i)} = \frac{w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)}{\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)}$$

$$\hat{p}_N(\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)$$

- We can show the following recursions for calculations of these weights:

$$w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})} = \underbrace{\frac{p(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1})}{q(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1})}}_{w(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1})} \underbrace{\frac{f(x_n \mid x_{n-1}) g(y_n \mid x_n)}{q(x_n \mid y_n, x_{n-1})}}_{\text{Incremental Weight}}$$

- This suggests the following sequential Importance Sampling Algorithm.

Sequential Importance Sampling

At step n=1:

- Sample $X_1^{(i)} \sim q(x_1 \mid y_1), i = 1,\ldots,N$ and then approximate:

$$\hat{p}_N(x_1|y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1),\ W_1^{(i)}\left(X_1^{(i)}, y_1\right) \propto \frac{\mu\left(X_1^{(i)}\right) g\left(y_1, X_1^{(i)}\right)}{q\left(X_1^{(i)}|y_1\right)}$$

At step n≥2:

- Sample $X_n^{(i)} \sim q\left(x_n \mid y_n, X_{n-1}^{(i)}\right), n = 1,\ldots,N,$ and compute:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),$$

$$W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)}|X_{n-1}^{(i)}\right) g\left(y_n|X_n^{(i)}\right)}{q\left(X_n^{(i)}|y_n, X_{n-1}^{(i)}\right)}$$

- The algorithm has computational complexity $\mathcal{O}$(N) independent of n.

Sequential Importance Sampling

- Note that the complexity of the algorithm does not increase with *n*.
- The algorithm is fully parallelizable.
- Also note that if our interest is on computing the marginal posterior,

$\hat{p}_N(x_n|\boldsymbol{y}_{1:n})$ (posterior filtered density), then we only need to store $\boldsymbol{X}_{n-1:n}^{(i)}$ rather than all the $\boldsymbol{X}_{1:n}^{(i)}$ paths

$$\hat{p}_N(x_n|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}(x_n),$$

$$W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)}|X_{n-1}^{(i)}\right) g\left(y_n|X_n^{(i)}\right)}{q\left(X_n^{(i)}|y_n, X_{n-1}^{(i)}\right)}$$

- One can show that this approaches the true posterior as $N \to \infty$.
  - Crisan, D., P. D. Moral, and T. Lyons (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields 5*(3), 293–318.

Selection of the Importance Distribution

Selection of the Importance Distribution

- So far, we haven't discussed how to select an optimal or at least a reasonable importance distribution.
- The choice of the importance distribution is directly related to the efficiency of the proposed SIS method.
  - The importance distribution should be easy to take samples from. Thus the difficulty in direct sampling from the target distribution is bypassed.
  - Since the importance distribution is different from the target distribution, the importance weight is used to make a compensation.
  - However, introducing auxiliary distribution (importance density) increases the variance in the simulation.

Optimal Importance Distribution

- The variance of the importance weight can be estimated by

$$\mathrm{var}_{q_n(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(w_n\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)\right)=\int w_n^2 q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)d\boldsymbol{x}_{1:n}-\left(\int w_n q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)d\boldsymbol{x}_{1:n}\right)^2$$

- Since $w_n\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)=\dfrac{p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)}{q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}$, we have

$$\mathrm{var}_{q_n(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(w_n\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)\right)=\int w_n^2 q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)d\boldsymbol{x}_{1:n}-\left(\int w_n q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)d\boldsymbol{x}_{1:n}\right)^2$$

$$=\int\frac{\left(p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)\right)^2}{q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}d\boldsymbol{x}_{1:n}-\left(p\left(\boldsymbol{y}_{1:n}\right)\right)^2=$$

- Ideally when the proposal distribution is exactly the same as the target distribution, i.e. $q_n\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)=p\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)$, the variance is ZERO !

$$\mathrm{var}_{q_n}\left(w_n\right)=\int\frac{\left(p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)\right)^2}{p\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}d\boldsymbol{x}_{1:n}-\left(p\left(\boldsymbol{y}_{1:n}\right)\right)^2=p\left(\boldsymbol{y}_{1:n}\right)\int p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)d\boldsymbol{x}_{1:n}-\left(p\left(\boldsymbol{y}_{1:n}\right)\right)^2=0$$

Locally Optimal Importance Distribution

- A locally optimal IS distribution $q_n\left(x_n \mid x_{n-1}, y_n\right)$ is the one that minimizes the variance of the weights:

$$w_n\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) = \frac{p\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right)}{q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)} = \frac{p\left(\boldsymbol{y}_{1:n}\right) p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) q_n\left(x_n \mid x_{n-1}, y_n\right)} =$$

$$= \frac{p\left(\boldsymbol{y}_{1:n}\right) p\left(x_n, \boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) q_n\left(x_n \mid x_{n-1}, y_n\right)} = \frac{p\left(\boldsymbol{y}_{1:n}\right) p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n}\right) p\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) q_n\left(x_n \mid x_{n-1}, y_n\right)} =$$

$$= \frac{p\left(\boldsymbol{y}_{1:n}\right) p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)} \frac{p\left(x_n \mid x_{n-1}, y_n\right)}{q_n\left(x_n \mid x_{n-1}, y_n\right)}$$

- Conditional on $\boldsymbol{x}_{1:n-1}$ and $\boldsymbol{y}_{1:n}$, the optimal choice is given as:

$$q_n\left(x_n \mid x_{n-1}, y_n\right) = p\left(x_n \mid x_{n-1}, y_n\right) = \frac{p\left(x_n \mid x_{n-1}\right) p\left(y_n \mid x_n, x_{n-1}\right)}{p\left(y_n \mid x_{n-1}\right)} =$$

$$= \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{\int f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) dx_n}$$

Locally Optimal Importance Distribution

$$q_n\left(x_n \mid x_{n-1}, y_n\right) = \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{\int f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) dx_n}$$

- Our weights are now updated as follows:

$$w_n\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{q_n\left(x_n \mid x_{n-1}, y_n\right)} =$$

$$= w_{n-1}\left(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right) \int f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) dx_n$$

- Asymptotically as N→∞, the Monte Carlo approximation converges to the true values.

- Unfortunately the choice cannot always be used unless we make some approximations (e.g. Extended/Unscented Kalman filter).

Locally Optimal Importance Distribution

- Substituting the locally optimal proposal distribution into the recursive form for the importance weight,

$$w_n^{(i)} = w_{n-1}^{(i)} \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{q_n\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}$$

we obtain the following

$$w_n^{(i)} = w_{n-1}^{(i)} \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{p\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)} = w_{n-1}^{(i)} \frac{p\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) p\left(y_n \mid X_n^{(i)}, X_{n-1}^{(i)}\right)}{p\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}$$

$$= w_{n-1}^{(i)} \frac{p\left(X_n^{(i)}, y_n \mid X_{n-1}^{(i)}\right)}{p\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)} \Rightarrow$$

$$w_n^{(i)} = w_{n-1}^{(i)} p\left(y_n \mid X_{n-1}^{(i)}\right) = w_{n-1}^{(i)} \int f\left(x_n \mid X_{n-1}^{(i)}\right) g\left(y_n \mid x_n\right) dx_n$$

Locally Optimal Importance Distribution

- **Proposition**

$q^{opt}\left(x_n \mid x_{n-1}, y_n\right) = p\left(x_n \mid x_{n-1}, y_n\right)$ is the importance density which minimizes the variance of the importance weight w_n conditional upon $x_{1:n-1}$ and $y_{1:n}$

$$\operatorname{var}_{q_n\left(x_n \mid x_{n-1}, y_n\right)}\left[w_n\right] = w_{n-1}^2 \int \frac{\left(g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right)\right)^2}{q_n\left(x_n \mid x_{n-1}, y_n\right)} dx_n - \left(w_{n-1} \int g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right) dx_n\right)^2$$

When $q_n^{opt}\left(x_n \mid x_{n-1}, y_n\right) = p\left(x_n \mid x_{n-1}, y_n\right) = \dfrac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{\int f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) dx_n}$, *we have*:

$$\operatorname{var}_{q_n^{opt}\left(x_n \mid x_{n-1}, y_n\right)}\left[w_n\right] = w_{n-1}^2 \int g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right) dx_n \int g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right) dx_n - \left(w_{n-1} \int g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right) dx_n\right)^2 \Rightarrow$$

$$\operatorname{var}_{q_n^{opt}\left(x_n \mid x_{n-1}, y_n\right)}\left[w_n^{opt}\right] = 0$$

Locally Optimal Importance Distribution

- In general, it is intractable to sample from $p(x_n \mid x_{n-1}, y_n)$ and to evaluate the integral needed to compute the predictive density $p\left(y_n \mid x_{n-1}^{(i)}\right)$.
- The optimal proposal distribution can be used:
  - When $\mathbf{x}_t$ is discrete, so the integral becomes a sum. If the entire state space is discrete, we can use an HMM filter instead, but in some cases, some parts of the state are discrete, and some continuous.
  - When $p(x_n \mid x_{n-1}, y_n)$ is Gaussian. This occurs when the dynamics are nonlinear but the observations are linear.
- In cases where the model is not linear-Gaussian, we may still compute a Gaussian approximation to $p(x_n \mid x_{n-1}, y_n)$ using the unscented transform as a proposal (**unscented particle filter)**.
- In general, we can use other kinds of **data-driven proposals**, perhaps based on discriminative models. Unlike MCMC, we do not need to worry about the proposals being reversible.

- van der Merwe, R., A. Doucet, N. de Freitas, and E. Wan (2000). The unscented particle filter. In *NIPS-13*.

Application to State-Space Model

- Consider the state-space model
  - state equation
$$f\left(x_n \mid x_{n-1}\right) = \mathcal{N}\left(x_n; \alpha\left(x_{n-1}\right), \beta\left(x_{n-1}\right)\right)$$
  - observation equation
$$g\left(y_n \mid x_n\right) = \mathcal{N}\left(y_n; x_n, \sigma_w^2\right)$$
  - to abbreviate the notation, we use
$$\alpha = \alpha\left(x_{n-1}\right), \quad \beta = \beta\left(x_{n-1}\right)$$

- The optimal proposal distribution

$$\begin{aligned} p\left(x_n \mid y_n, x_{n-1}\right) &\propto f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) \\ &= \mathcal{N}\left(x_n; \alpha\left(x_{n-1}\right), \beta\left(x_{n-1}\right)\right) \times \mathcal{N}\left(y_n; x_n, \sigma_w^2\right) \\ &\propto \exp\left(-\frac{\left(x_n-\alpha\right)^2}{2\beta}\right) \exp\left(-\frac{\left(y_n-x_n\right)^2}{2\sigma_w^2}\right) \\ &= \exp\left(-\frac{1}{2}\left(x_n^2\left(\frac{1}{\beta}+\frac{1}{\sigma_w^2}\right) - 2x_n\left(\frac{\alpha}{\beta}+\frac{y_n}{\sigma_w^2}\right) + \left(\frac{\alpha^2}{\beta}+\frac{y_n^2}{\sigma_w^2}\right)\right)\right) \end{aligned}$$

Application to State-Space Model

- Obviously, the proposal distribution is subject to a Gaussian distribution with

$$p\left(x_n \mid y_n, x_{n-1}\right) \propto \exp\left(-\frac{1}{2}\left(x_n^2\left(\frac{1}{\beta}+\frac{1}{\sigma_w^2}\right)-2x_n\left(\frac{\alpha}{\beta}+\frac{y_n}{\sigma_w^2}\right)+\left(\frac{\alpha^2}{\beta}+\frac{y_n^2}{\sigma_w^2}\right)\right)\right)$$

$$\propto \exp\left(-\frac{\left(\frac{1}{\beta}+\frac{1}{\sigma_w^2}\right)}{2}\left(x_n-\frac{\left(\frac{\alpha}{\beta}+\frac{y_n}{\sigma_w^2}\right)}{\left(\frac{1}{\beta}+\frac{1}{\sigma_w^2}\right)}\right)^2\right)$$

  - Variance

$$\sigma^2\left(x_{n-1}\right)=\left(\frac{1}{\beta}+\frac{1}{\sigma_w^2}\right)^{-1}$$

  - Mean

$$m\left(x_{n-1}\right)=\sigma^2\left(x_{n-1}\right)\left(\frac{\alpha}{\beta}+\frac{y_n}{\sigma_w^2}\right)$$

Application to State-Space Model

- Consider the simpler model

$$X_n = \alpha X_{n-1} + V_n,\ X_1 \sim \mathcal{N}(0,1),\ V_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$$

$$Y_n = X_n + \sigma_w W_n,\ W_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$$

- The state and observation equations are

$$f\left(x_n \mid x_{n-1}\right) = \mathcal{N}\left(x_n; \alpha x_{n-1}, 1\right)$$

$$g\left(y_n \mid x_n\right) = \mathcal{N}\left(y_n; x_n, \sigma_w^2\right)$$

- Using the results from our previous example

$$\alpha\left(x_{n-1}\right) = \alpha x_{n-1},\ \beta\left(x_{n-1}\right) = 1$$

- Substitute them into the equation for the optimal proposal distribution, we obtain:

$$\sigma^2\left(x_{n-1}\right) = \left(1 + \frac{1}{\sigma_w^2}\right)^{-1} \qquad m\left(x_{n-1}\right) = \left(1 + \frac{1}{\sigma_w^2}\right)^{-1}\left(\alpha x_{n-1} + \frac{y_n}{\sigma_w^2}\right)$$

State-Space Model: Linear Observation

- Consider a generalized Gaussian state-space model with linear observation equation

$$\boldsymbol{x}_t = f\left(\boldsymbol{x}_{t-1}\right) + \boldsymbol{v}_t \quad , \quad \boldsymbol{v}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_v\right)$$

$$\boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{w}_t \quad , \quad \boldsymbol{w}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_w\right)$$

- The optimal importance distribution

$$q^{opt}\left(x_t \mid x_{1:t-1}, y_{1:t}\right) = p\left(x_t \mid x_{t-1}, y_t\right) = \frac{f\left(x_t \mid x_{t-1}\right) g\left(y_t \mid x_t\right)}{\int p\left(x_t, y_t \mid x_{t-1}\right) dx_t}$$

- It can easily be shown with substitution above that

$$p\left(x_t \mid x_{t-1}, y_t\right) = \mathcal{N}\left(\boldsymbol{m}, \Sigma\right)$$

  is a Gaussian distribution with

$$\Sigma^{-1} = \Sigma_v^{-1} + C^T \Sigma_w^{-1} C$$

$$\boldsymbol{m} = \Sigma\left(\Sigma_v^{-1} f\left(\boldsymbol{x}_{t-1}\right) + C^T \Sigma_w^{-1} \boldsymbol{y}_t\right)$$

- Also you can show that:

$$p\left(y_t \mid x_{t-1}\right) = \mathcal{N}\left(y_t; Cf\left(\boldsymbol{x}_{t-1}\right), \Sigma_v + C\Sigma_w C^T\right)$$

Optimal Importance Distribution

- We stated that we will like importance distributions $q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ with heavier tails than $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ and s.t. $q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) \approx p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$

- The introduced structure of the SIS distribution limits us as e.g.

$$X_1 \sim q_1\left(x_1 \mid y_1\right)$$

even though at time *n > 1*, we might be able to sample from

$$X_1 \sim p\left(x_1 \mid \boldsymbol{y}_{1:n}\right)$$

- At time *n*, the only degree of freedom we have is $q_n\left(x_n \mid y_{n,} x_{n-1}\right)$. We can select it to minimize the variance of the weight w($\mathbf{x}_{1:n}$,$\mathbf{y}_{1:n}$) so that

$$q_{opt}\left(x_n \mid y_n, x_{n-1}\right) = p\left(x_n \mid y_n, x_{n-1}\right) = \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{p\left(y_n \mid x_{n-1}\right)}$$

- Sampling from $p\left(x_n \mid y_{n,} x_{n-1}\right)$ maybe difficult as $p\left(y_n \mid x_{n-1}\right)$ may not be easy to evaluate with some key important exceptions.

Optimal Importance Distribution

- At time *n*, we are interested at the optimal choice of the importance distribution that minimizes the variance of the weights

$$w_n\left(X_n^{(i)}, \boldsymbol{y}_{1:n}\right)$$

- The optimal choice of the proposal distribution is obviously

$$q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) = p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

- This is not a possible choice and even an approximation of $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ is not available due to the sequential nature of the proposal distribution.

- For example, remember that $X_1^{(i)} \sim q_1\left(x_1 \mid y_1\right)$ whereas at time *n*, we would love to have

$$X_1^{(i)} \sim p\left(x_1 \mid \boldsymbol{y}_{1:n}\right)$$

Suboptimal Importance Distribution

- A simpler choice of an importance sampling distribution $q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ is derived based on the following:

$$q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) = p\left(\boldsymbol{x}_{1:n}\right)$$

that is

$$q_1\left(x_1 \mid y_1\right) = \mu\left(x_1\right)$$

and

$$q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right) = \frac{q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)} = \frac{p\left(\boldsymbol{x}_{1:n}\right)}{p\left(\boldsymbol{x}_{1:n-1}\right)} = f\left(x_n \mid x_{n-1}\right)$$

- We also have:

$$w_n\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) = \frac{p\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right)}{q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)} = \frac{p\left(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right)}{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)} \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right)} = w_{n-1}\left(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right) \times g\left(y_n \mid x_n\right)$$

- This choice is extremely poor if the data are very informative (peaky likelihood), since the proposal distribution doesn't include any information from the data $\boldsymbol{y}_{1:n}$.

$$w_n\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right) \times g\left(y_n \mid x_n\right) = \prod_{k=1}^{n} g\left(y_k \mid x_k\right)$$

Algorithm: Suboptimal Importance Density

- One selects

$$q_1(x_1) = \mu(x_1) \text{ and } q_n(x_n \mid x_{1:n-1}) = q_n(x_n \mid x_{n-1}) = f(x_n \mid x_{n-1})$$

- At time *n=1*, we sample $X_1^{(i)} \sim \mu(.)$ and set $w_1^{(i)} = g\left(y_1 \mid X_1^{(i)}\right)$

- At time *n* (*n>1*)

  - sample $X_n^{(i)} \sim f(. \mid \boldsymbol{X}_{1:n-1}^{(i)})$ and set $\boldsymbol{X}_{1:n}^{(i)} = \left(\boldsymbol{X}_{1:n-1}^{(i)}, X_n^{(i)}\right)$

  - evaluate the importance weights

$$w_n^{(i)} = w_{n-1}^{(i)} g\left(y_n \mid X_n^{(i)}\right) \quad \text{or normalized}: \quad W_n^{(i)} = w_n^{(i)} / \sum_{i=1}^{N} w_n^{(i)}$$

- At any time *n* we have:

$$\boldsymbol{X}_{1:n}^{(i)} \sim \mu(x_1) g(y_1 \mid x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1}) g(y_k \mid x_k), \; w_n\left(\boldsymbol{X}_{1:n}^{(i)}\right) = \prod_{k=2}^{n} g(y_k \mid X_k^{(i)})$$

Importance Distribution Obtained by Local Linearization

- A simple choice selects the importance function $p\left(x_t \mid x_{t-1}, y_t\right)$ a parametric distribution

$$p\left(x_t \mid \theta\left(x_{t-1}, y_t\right)\right)$$

with a finite-dimensional parameter θ determined by x_{t-1} and y_t.

- Many strategies are possible based upon this idea. Here, we present a scheme that results in a Gaussian importance function whose parameters are evaluated using local linearization.

- **Local linearization of the state-space model**
  The state-space model is linearized locally with the aim of obtaining an importance function and the algorithm obtained still converges asymptotically towards the required filtering distribution under the usual assumptions for importance functions.

Importance Distribution Obtained by Local Linearization

- For example, let us consider the following non-linear model

$$\boldsymbol{x}_t = f\left(\boldsymbol{x}_{t-1}\right) + \boldsymbol{v}_t \quad , \quad \boldsymbol{v}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_v\right)$$
$$\boldsymbol{y}_t = g\left(\boldsymbol{x}_t\right) + \boldsymbol{w}_t \quad , \quad \boldsymbol{w}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_w\right)$$

- Performing an approximation up to first-order of the observation equation gives

$$y_t = g(x_t) + w_t$$
$$\simeq g\big(f(x_{t-1})\big) + \frac{\partial g(x_t)}{\partial x_t}\Big|_{x_t = f(x_{t-1})}\big(x_t - f(x_{t-1})\big) + w_t$$

which is a first-order Taylor expansion at $\boldsymbol{x}_t = f\left(\boldsymbol{x}_{t-1}\right)$.

One-dimensional Nonlinear Example

- Consider a one-dimensional nonlinear example

$$x_n = 0.5x_{n-1} + 25\frac{x_{n-1}}{1+x_{n-1}^2} + 8\cos\left[1.2(n-1)\right] + \sigma_v V_n$$

$$y_n = x_n^2/20 + \sigma_w W_n$$

$$\sigma_w^2 = 1, \sigma_v^2 = 10, W_n \sim \mathcal{N}(0,1), V_n \sim \mathcal{N}(0,1)$$

- The state and observation equations are two Gaussian distributions

$$x_n \mid x_{n-1} \sim \mathcal{N}\left(0.5x_{n-1} + 25\frac{x_{n-1}}{1+x_{n-1}^2} + 8\cos\left[1.2(n-1)\right], 10\right)$$

$$y_n \mid x_n \sim \mathcal{N}\left(x_n^2/20, 1\right)$$

R.Casarin. Bayesian Monte Carlo filtering for stochastic volatility models.

One-dimensional Nonlinear Example

- In this occasion, it is impossible to sample from the locally optimal importance distribution $q^{opt}\left(x_n \mid x_{n-1}, y_n\right)$

- We simply use the prior distribution $x_n \mid x_{n-1}$ as the proposal distribution

- At each time step, we sample $X_n^{(i)}$ from

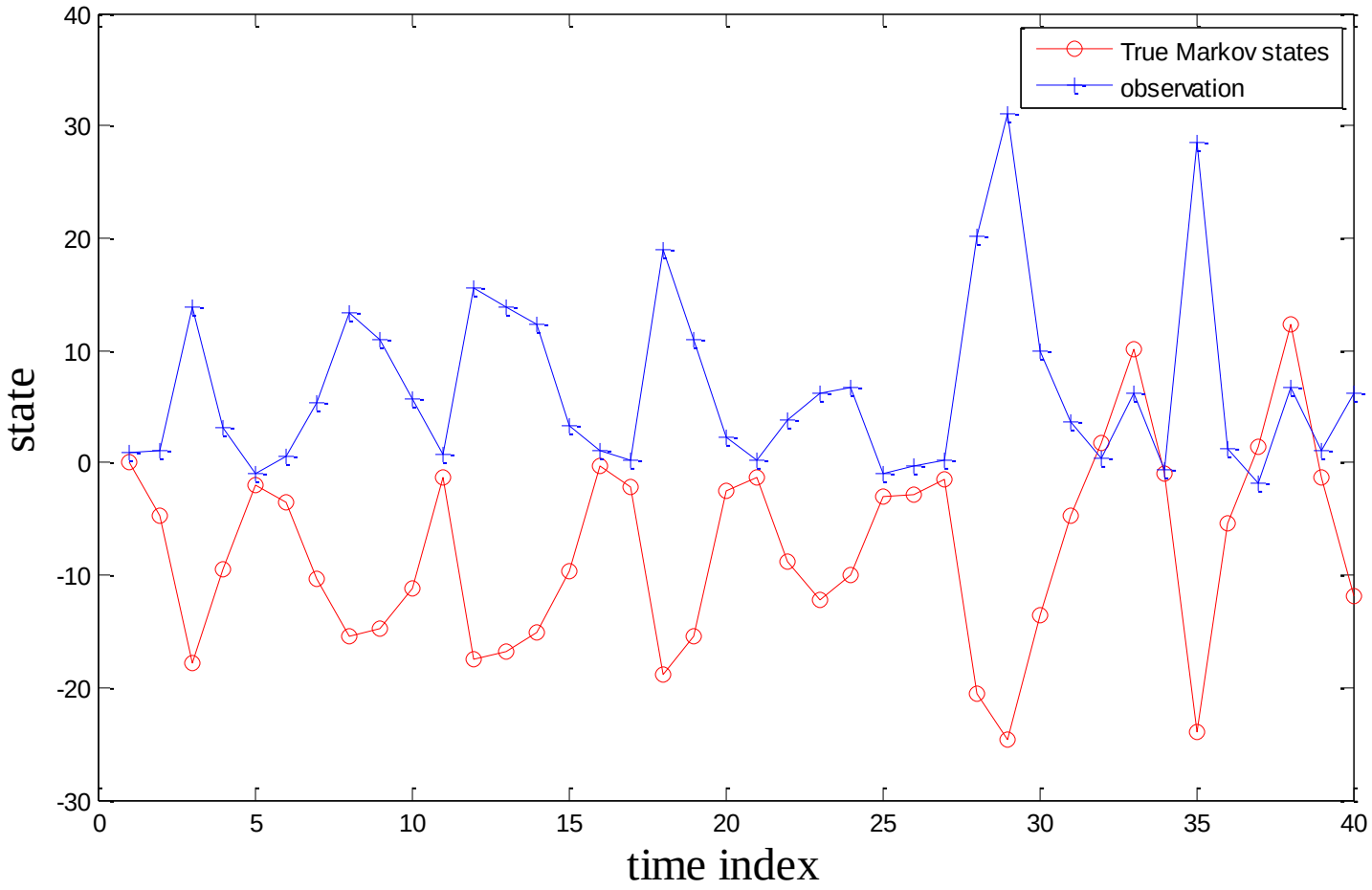
$$x_n \mid X_{n-1}^{(i)} \sim \mathcal{N}\left(0.5 X_{n-1}^{(i)} + 25\frac{X_{n-1}^{(i)}}{1+\left(X_{n-1}^{(i)}\right)^2} + 8\cos\left[1.2\left(n-1\right)\right], 10\right)$$

and compute the weights

$$w_n^{(i)} = w_{n-1}^{(i)} \times \mathcal{N}\left(x_n^2/20, 1\right)$$

Nonlinear Example

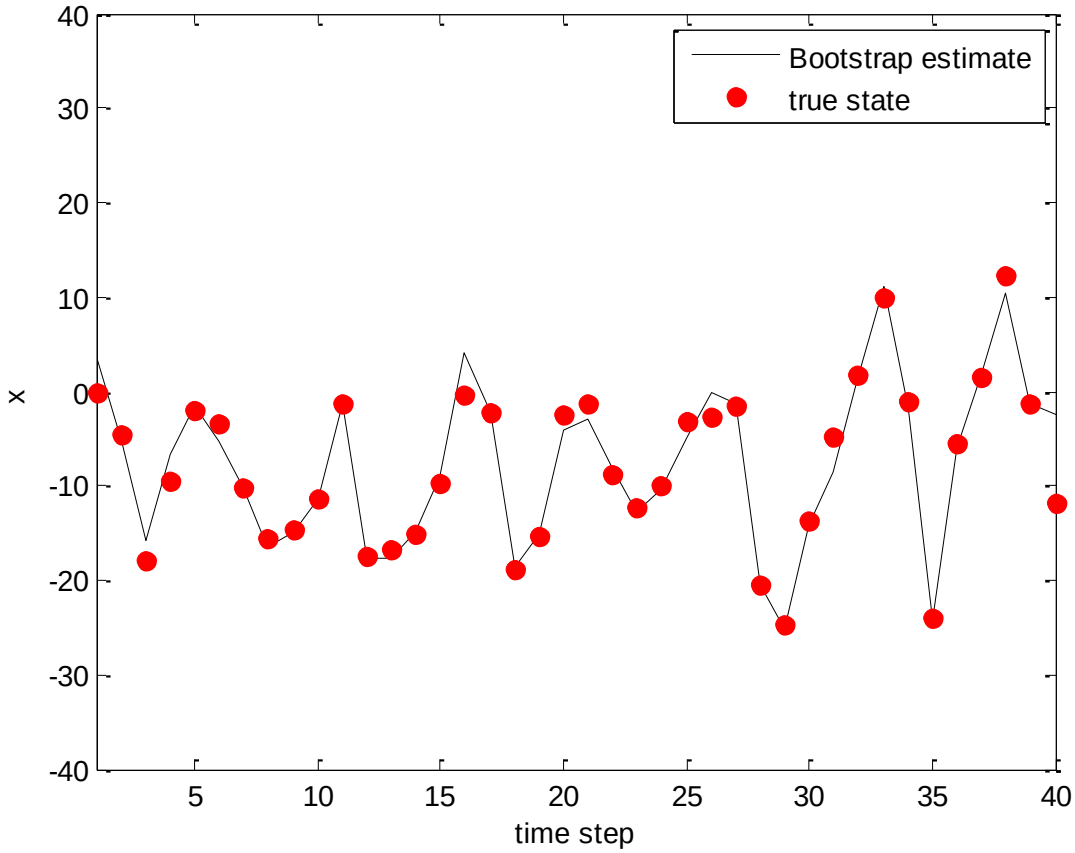
- Non-linear system observations and true Markov states



For implementation, see MatLab code Bootstrap

Nonlinear Example

- This figure shows the result of directly applying the algorithm with a sample size of N. The solid line represents the mean of the 200 posterior samples.



Importance Distribution from Local Linearization

- As an alternative, we can use importance distribution obtained from local linearization.
- First, we get a locally linearized observation equation

$$y_t \simeq g\big(f(x_{t-1})\big) + \frac{\partial g(x_t)}{\partial x_t}\Big|_{x_t = f(x_{t-1})}\big(x_t - f(x_{t-1})\big) + w_t$$

$$= -\frac{f^2(x_{t-1})}{20} + \frac{f(x_{t-1})}{10}x_t + w_t$$

$$x_n = 0.5x_{n-1} + 25\frac{x_{n-1}}{1+x_{n-1}^2} + 8\cos\Big[1.2\big(n-1\big)\Big] + \sigma_v V_n$$

$$y_n = x_n^2/20 + \sigma_w W_n$$

Nonlinear Example

- Linearized importance function

$$p\left(x_t \mid x_{t-1}, y_t\right) = \mathcal{N}\left(x_t; m_t, \sigma_t^2\right)$$

where

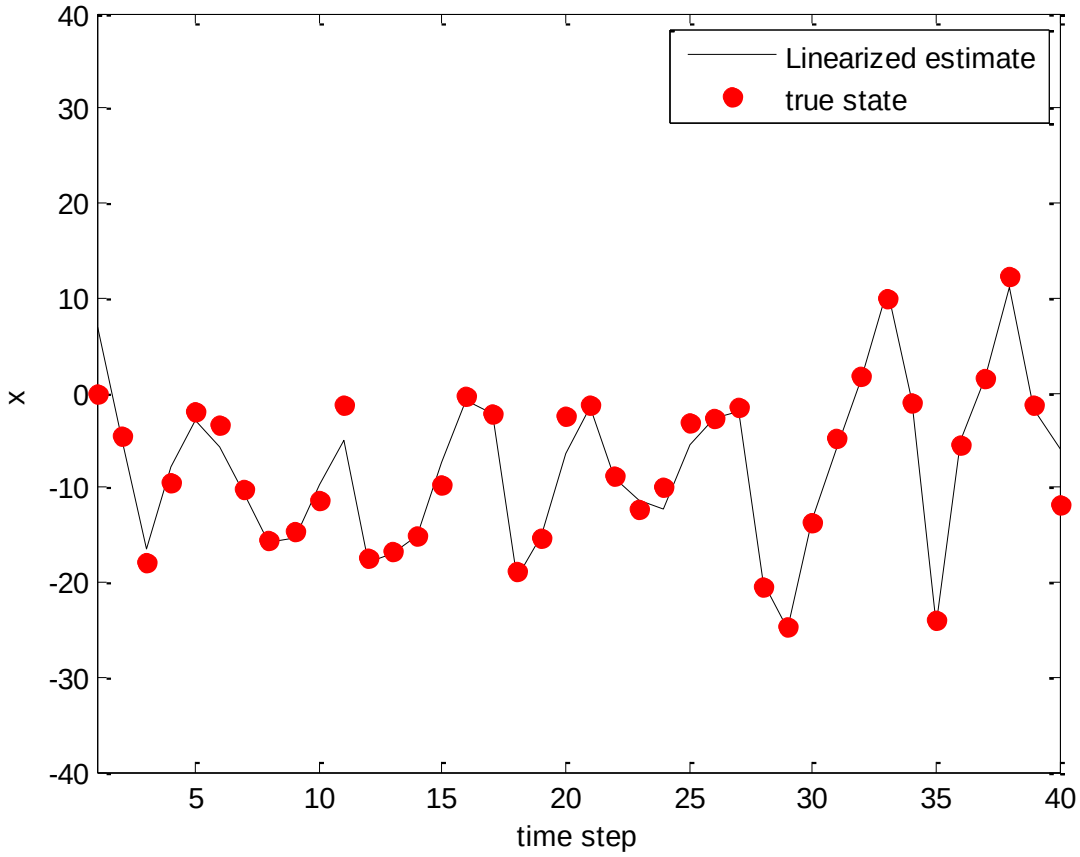
$$\sigma_t^{-2} = \sigma_v^{-2} + \sigma_w^{-2} \frac{f^2\left(x_{t-1}\right)}{100}$$

$$m_t = \sigma_t^2 \left[\sigma_v^{-2} f\left(x_{t-1}\right) + \sigma_w^{-2} \frac{f\left(x_{t-1}\right)}{10} \left(y_t + \frac{f^2\left(x_{t-1}\right)}{20} \right) \right]$$

- We take it as the importance weight for the nonlinear Gaussian state-space model

Nonlinear Example

- N = 200 particles are used. The posterior mean of the linearized estimate is presented.



For implementation, see MatLab code Linearize

Importance Distribution Obtained by Local Linearization

- The locally linearized Gaussian state-space model can be formulated as

$$\boldsymbol{x}_t = f\left(\boldsymbol{x}_{t-1}\right) + \boldsymbol{v}_t \quad , \quad \boldsymbol{v}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_v\right)$$

$$\boldsymbol{y}_t - g\left(f\left(\boldsymbol{x}_{t-1}\right)\right) + \left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)} f\left(\boldsymbol{x}_{t-1}\right) = \left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)} \boldsymbol{x}_t + \boldsymbol{w}_t \quad , \quad \boldsymbol{w}_t \sim \mathcal{N}\left(\boldsymbol{0}, \Sigma_w\right)$$

- Recall the generalized Gaussian state-space model with linear observation equation, the optimal importance distribution for the linearized state-space model is

$$q^{opt}\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t}\right) = p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_t\right) = \mathcal{N}\left(\boldsymbol{m}, \Sigma\right)$$

where

$$\Sigma^{-1} = \Sigma_v^{-1} + \left[\left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)}\right]^T \Sigma_w^{-1} \left[\left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)}\right]$$

$$\boldsymbol{m} = \Sigma\left(\Sigma_v^{-1} f\left(\boldsymbol{x}_{t-1}\right) + \left[\left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)}\right]^T \Sigma_w^{-1}\left(\boldsymbol{y}_t - g\left(f\left(\boldsymbol{x}_{t-1}\right)\right) + \left.\frac{\partial g\left(\boldsymbol{x}_t\right)}{\partial \boldsymbol{x}_t}\right|_{x_t = f\left(x_{t-1}\right)} f\left(\boldsymbol{x}_{t-1}\right)\right)\right)$$

Stochastic Volatility Example

- To clearly describe the application of the SIS algorithm, let us take the stochastic volatility dynamical model as an example

$$x_0 \sim \mathcal{N}\left(0, \sigma^2\right)$$

$$x_n = \nu + \phi x_{n-1} + \sigma v_n, \; v_n \overset{i.i.d.}{\sim} \mathcal{N}\left(0,1\right)$$

$$y_n = \exp\left(\frac{x_n}{2}\right) w_n, \; w_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0,1\right)$$

where x_n denotes the unobserved Markov states, and y_n denotes the observations.

- We use simulated data: First generate the states x_n from the first 2 equs above (these are what we call `true state'), and then using x_n, generate y_n from the 3nd equation.

Stochastic Volatility Example

- Write the state and observation equations

$$x_n \sim \mathcal{N}\left(v + \phi x_{n-1}, \sigma^2\right)$$

$$y_n \sim \mathcal{N}\left(0, \exp\left(x_n\right)\right)$$

- The recursive form of the posterior distribution is

$$p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) \propto p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) \times f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)$$

- Here we cannot sample from the locally optimal importance distribution $q_n^{opt}\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right) = p\left(x_n \mid x_{n-1}, y_n\right)$. Thus, we use the simple (suboptimal) choice: Set

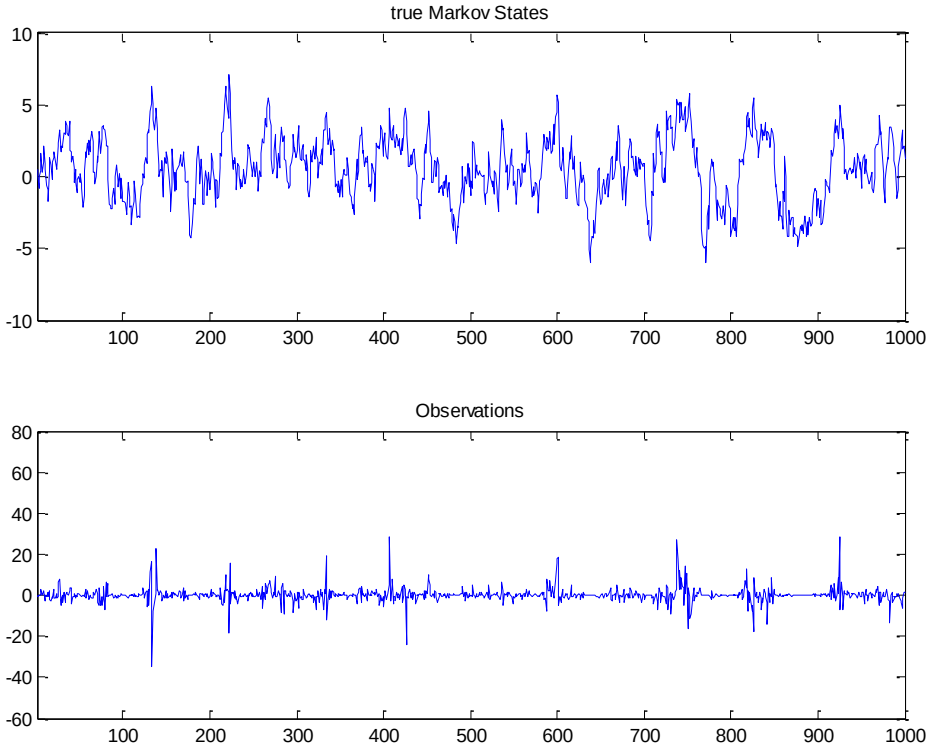
$$q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right) = f\left(x_n \mid x_{n-1}\right) \sim \mathcal{N}\left(v + \phi x_{n-1}, \sigma^2\right)$$

- Importance weight

$$w_n = w_{n-1} \times g\left(y_n \mid x_n\right), \quad \text{where } g\left(y_n \mid x_n\right) = \mathcal{N}\left(0, \exp\left(x_n\right)\right)$$

Stochastic Volatility Example

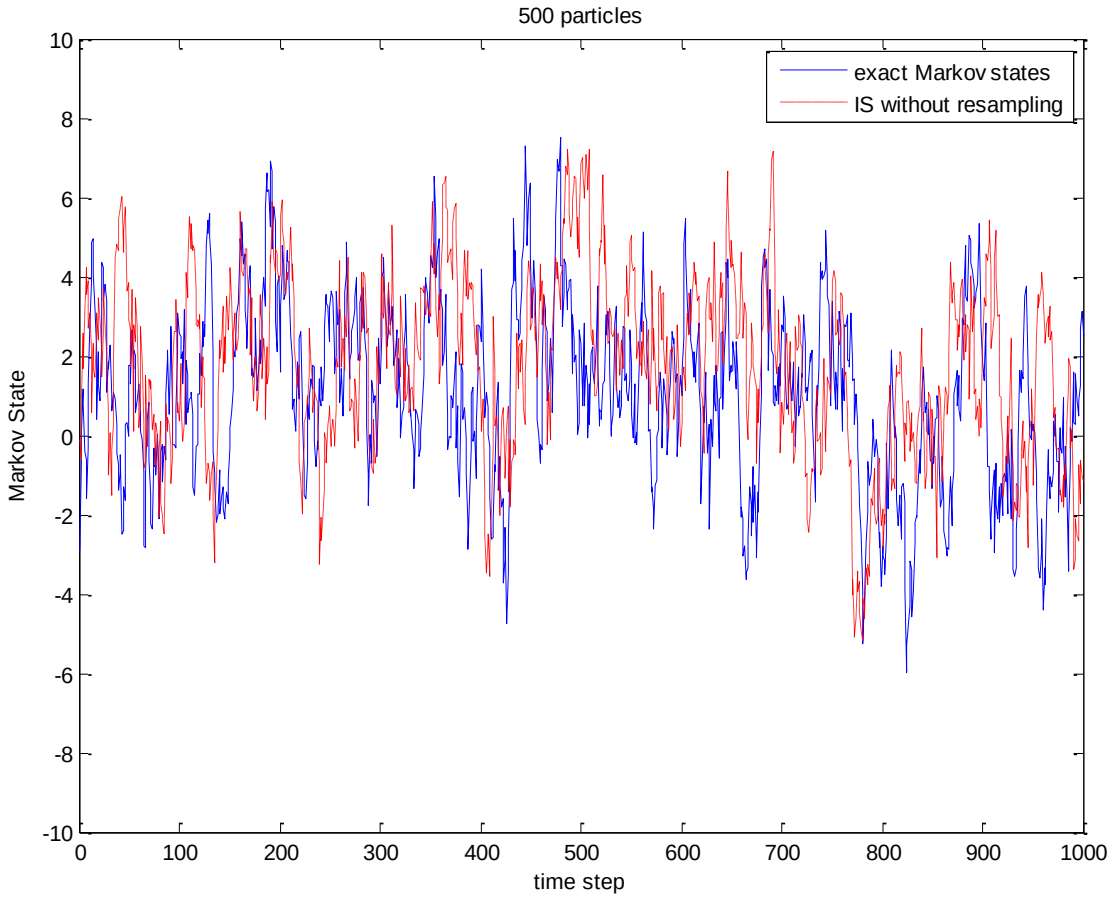
- We simulate the model by setting $\phi = 0.9$, $v = 0.1$, $\sigma = 1$
- A 1000 time steps realization
  - upper picture depicts the unobserved Markov States
  - lower picture depicted corresponding observations



A C++ implementation can be found here

Stochastic Volatility Example

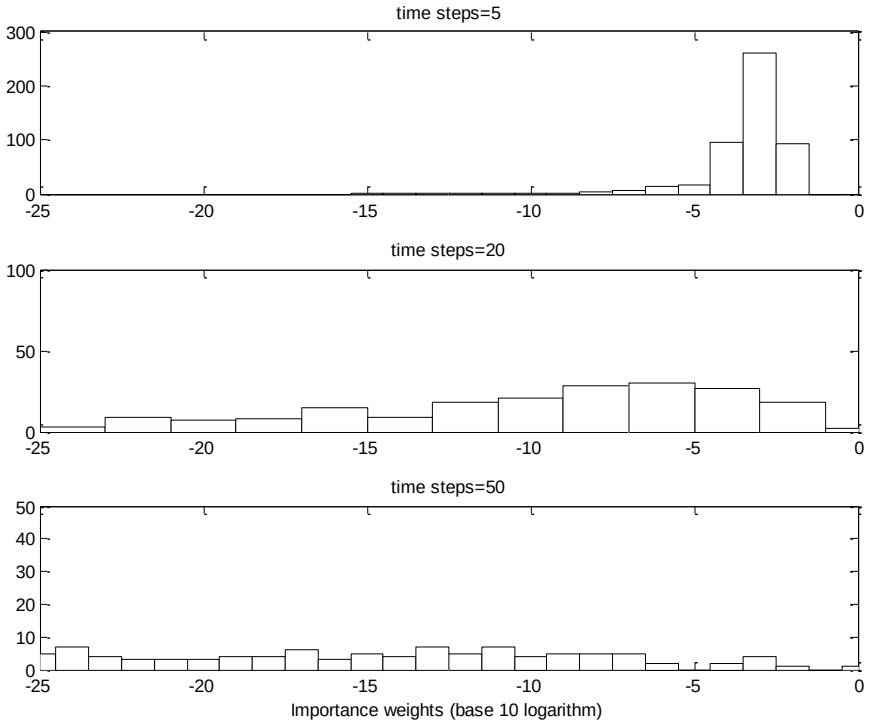
- Estimation (without resampling – to be introduced later on)
  - blue line represents the true (unobserved) Markov states
  - red line represents estimations (posterior means)



Stochastic Volatility Example

- Obviously, the estimation is unsatisfactory. There is significant difference between the estimated Markov states and the true values.

- The algorithm performs extremely poorly. Here are the histograms of the importance weights at time steps *5, 20, 50*. After a few time steps, only a few particles have non negligible weights.



SEQUENTIAL IMPORTANCE SAMPLING/RESAMPLING

Problems with Sequential Imp. Sampling

- We can see that after a few time steps, only a very small number of samples have non-negligible weights.
- This means only small part of samples contribute to the estimation. Therefore the accuracy is very low.
- As the dimension of the target distribution increases over time, the problem is becoming increasingly more difficult.
- In practice, the discrepancy between the target $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ and the importance sampling distribution $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ increases. Even using the locally optimal importance sampling distribution will make the algorithm to eventually collapse.
- As the dimension *n* increases, the variance of the weights increases (typically geometrically) and the algorithm finally fails.

Problems with Sequential Imp. Sampling

- Since importance sampling is not working in high dimensions, we expect the same to be true for sequential importance sampling.

- Consider the following target distribution.

$$p(\boldsymbol{x}_{1:n}) = \prod_{k=1}^{n} p(x_k)$$

- We consider the following importance sampling distribution:

$$q(\boldsymbol{x}_{1:n}) = \prod_{k=1}^{n} q(x_k) = \prod_{k=1}^{n} \mathcal{N}(x_k; 0, \sigma^2)$$

- Then the variance of the weights is given as follows:

$$Var_{q(\boldsymbol{x}_{1:n})}\left(\frac{p(\boldsymbol{x}_{1:n})}{q(\boldsymbol{x}_{1:n})}\right) = \left(\mathbb{E}_{p(x)}\left(\frac{p(x)}{q(x)}\right)\right)^n - 1 = \left(Var_{q(x)}\left(\frac{p(x)}{q(x)}\right) + 1\right)^n - 1 = \sum_{i=1}^{n}\binom{n}{i}\left[Var_{q(x)}\left(\frac{p(x)}{q(x)}\right)\right]^i$$

$$where: Var_{q(x)}\left(\frac{p(x)}{q(x)}\right) = \mathbb{E}_{p(x)}\left(\frac{p(x)}{q(x)}\right) - 1 > 0$$

Problems with Sequential Imp. Sampling

- Let us assume that we want to compute using this importance distribution, the expectation $\mathbb{E}_{p(x_n)}(\varphi(x_n))$

- Then the asymptotic variance of the SIS estimator is:

$$\int \frac{p^2(\boldsymbol{x}_{1:n})}{q(\boldsymbol{x}_{1:n})}\left(\varphi(x_n)-\mathbb{E}_{p(x_n|\boldsymbol{y}_{1:n})}(\varphi)\right)^2 d\boldsymbol{x}_{1:n} =$$

$$\left[\underbrace{\mathbb{E}_{p(x)}\left(\frac{p(x)}{q(x)}\right)}_{>1,\ p\neq q}\right]^{n-1} \int \frac{p^2(x_n)}{q(x_n)}\left(\varphi(x_n)-\mathbb{E}_{p(x_n|\boldsymbol{y}_{1:n})}(\varphi)\right)^2 dx_n$$

- We can see that the variance of the weights increases exponentially fast with n.

- In high dimensions, the discrepancy between the proposal distribution $q(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})$ and the target distribution $p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})$ increases.

- SIS works for moderate dimensions. Can we do something about it?

Resampling

- As n increases, the mass of our approximation to the target distribution concentrates on a few particles (degeneracy problem):

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}) \approx \delta_{\boldsymbol{X}_{1:n}^{(i_0)}}(\boldsymbol{x}_{1:n})$$

- Here a single delta mass is used (weight 1 for particle i_0).

- When the variance of the weights $\{W_n^{(i)}\}$ is high, the resampling idea essentially is to kill the samples with low weights $W_n^{(i)}$ $(relative\ to\ 1/N)$ and multiply the particles with higher weights.

- Of course the assumption here is that particles with low weights (relative to 1/N) at step n will have even lower weights at later on steps.

Ref : J.S.Liu, R.Chen. Blind deconvolution via sequential imputations. Journal of the American Statistical Association. 1995, 90:p567.

Resampling

- When the variance of the weights $\{W_n^{(i)}\}$ is high, we would like to get rid of the samples with low weights (relative to *1/N*) and multiply the samples with large weights.
- If a particle *i* has a low weight $\{W_n^{(i)}\}$ at time *n*, we anticipate that it would also have a low weight at n+1.
- To compute the samples with low weights is a waste of computational resources, since such samples have no contribution to the estimation of the target distribution.
- Resampling techniques are a key ingredient of SMC methods which can partially address the problem of degeneracy of the SIS algorithm.

Ref : J.S.Liu, R.Chen. Blind deconvolution via sequential imputations. Journal of the American Statistical Association. 1995, 90:p567.

Resampling

- Let us assume that at time n the following approximation holds:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- With resampling, we sample N times from the above distribution

$$\tilde{X}_n^{(i)} \sim \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}),\ i = 1, \ldots, N$$

to build a new approximation:

$$\tilde{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- Note that the resampled particles $\tilde{\boldsymbol{X}}_{1:n}^{(i)}$ are approximately distributed according to $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ but they are statistically dependent (so CLT approximations, etc. are not holding!)

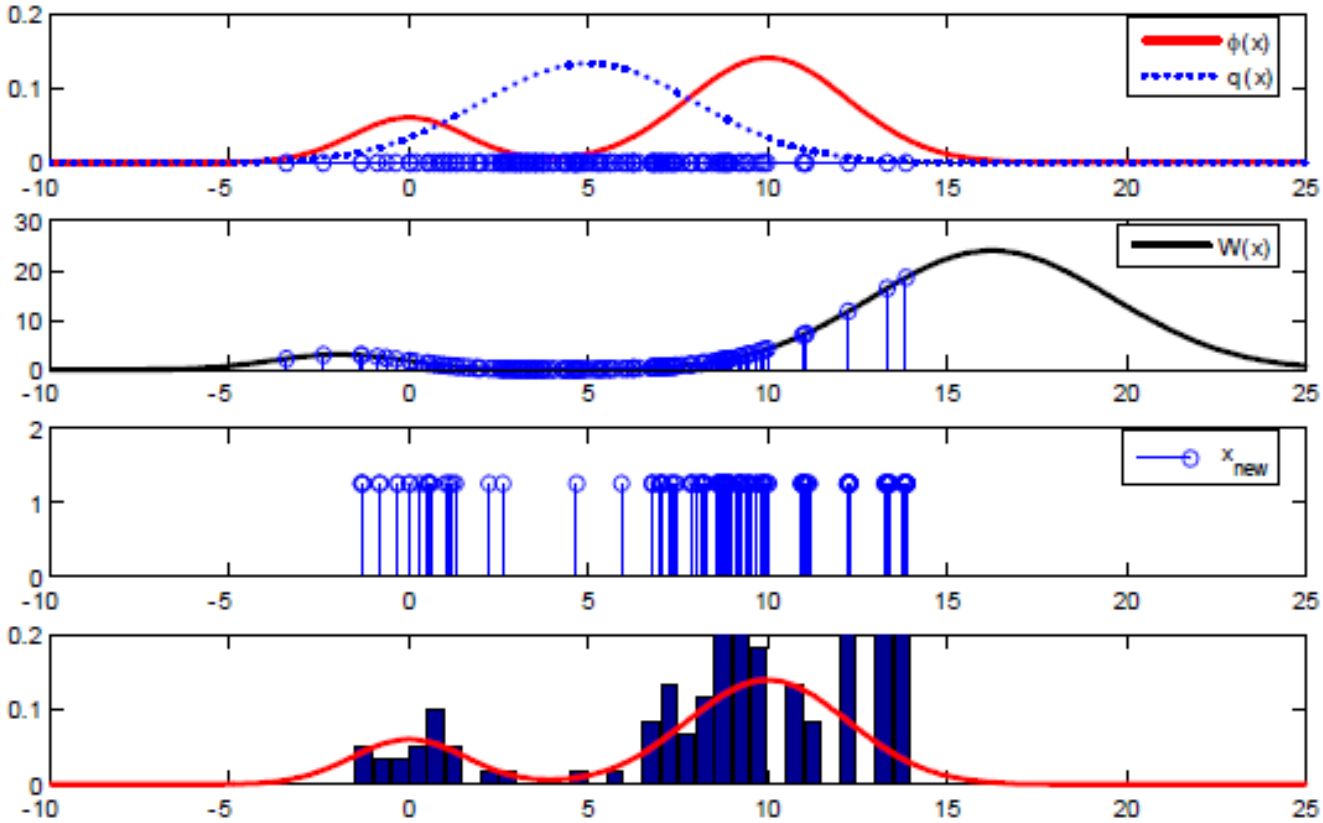
Importance Sampling with Resampling

- Importance Sampling Approximation is shown followed by Resampling (from A. Doucet)

$$\hat{\varphi}_N(x) = \sum_{i=1}^{N} W^{(i)} \delta_{X^{(i)}}(x) \Rightarrow$$

$$\tilde{X}^{(i)} \sim \tilde{\varphi}_N(x) \Rightarrow$$

$$\tilde{\varphi}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\tilde{X}^{(i)}}(x) \Rightarrow$$



Resampling

- Resampling can be done at any stage of the process.
  - However, resampling too often adds computational burden.
  - On the other hand, resampling too rarely may result in loss of efficiency.
- We need to measure the variance of the importance weights before we decide whether to carry out resampling. If the variation of the weights is too high based on the measures below, then we resample.
- There are several parameters for this purpose
  - ESS : effective sample size
  - CV : coefficient of variation
  - Entropy

Resampling: Coefficient of Variation

- Suppose the normalized importance weight is $W(x)$ in which x is drawn from a proposal distribution $q(x)$.
- The coefficient of variation, CV is computed as

$$CV^2(W) = \frac{1}{N}\sum_{j=1}^{N}\left\{NW_n^{(j)} - 1\right\}^2$$

- When N is large, CV is a reasonable approximation of the variance of "true weight" of particle s $\text{var}_q\left\{W_t^{*s}(x)\right\}, W_t^{*s}(x) = p\left(x_t^s \mid \boldsymbol{y}_{1:t}\right) / q\left(x_t^s \mid, x_{t-1}^s y_t\right)$
- We use this approximation because generally the variance cannot be computed directly.
- If the variance of the weights is large, we are wasting resources updating particles with low weight which don't contribute to the posterior estimate.

Coefficient of Variation (CV)

- Suppose that N samples have been drawn, each associated with a weight $W^{(k)}$ that is either zero or $1/M$; where M is the number of the non-zero weights.

- Substitute these values into the expression for the coefficient of variation, we have

$$CV^2(W) = \frac{1}{N}\sum_{j=1}^{N}\left\{NW_n^{(j)} - 1\right\}^2$$

$$= \frac{1}{N}\left\{N - M + M\left(\frac{N}{M} - 1\right)^2\right\}$$

$$= \frac{N}{M} - 1$$

- We see that:

$$CV = 0 \; if \; W_n^{(i)} = 1/N \;\; for \; all \; i \; and$$

$$CV = \sqrt{N-1} \; if \; W_n^{(i)} = 1 \; and \;\; W_n^{(j)} = 0 \; \forall j \neq i$$

Effective Sample Size (ESS)

- The effective sample size can be estimated based on the coefficient of variation CV

$$ESS = M = \frac{N}{1 + CV^2(w)}$$

or estimated directly by

$$ESS = \left(\sum_{j=1}^{N} \left(W_n^{(j)} \right)^2 \right)^{-1}$$

  - The effective sample size estimates the number of "non-zero" weights (i.e. number of effective particles)
  - In practical application, we can set a threshold M_{thres} for M, if
    - $M >= M_{thres}$, no resampling
    - $M < M_{thres}$, start resampling in the current time step

- We have that:

$$ESS = N \; if \; W_n^{(i)} = 1/N \;\; for \; all \; i \; and$$

$$ESS = 1 \; if \; W_n^{(i)} = 1 \; and \;\; W_n^{(j)} = 0 \; \forall j \neq i$$

Entropy

- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_n^{(i)} \log_2 \left(W_n^{(i)} \right)$$

- We have

$$Ent = \log_2 \left(N \right) \quad \text{if } W_n^{(i)} = 1/N \text{ for any } i$$

and

$$Ent = 0 \quad \text{if } W_n^{(i)} = 1 \text{ and } W_n^{(j)} = 0 \text{ for } j \neq i$$

Resampling

- Consider an IS approximation $\hat{p}^N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ of the target distribution $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$. To obtain approximate samples from $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$, we can simply sample from its IS approximation $\hat{p}^N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$; that is we select $\boldsymbol{X}_{1:n}^{(i)}$ with probability $W_n^{(i)}$ (a process known as rejuvenation).

- The idea is to eliminate particles with low weight and then create replicas of the surviving particles (survival of the fittest)

- This operation is called "Resampling" as it corresponds to sampling from an approximation $\hat{p}^N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ which was itself obtained by sampling.

- Although the degeneracy of the algorithm is still inevitable, resampling techniques can effectively reduce the variance due to the importance sampling. These methods take $\mathcal{O}$(N) time.

- Kanazawa, K., D. Koller, and S. Russell (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In *UAI*.
- Doucet, A., N. de Freitas, and N. J. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Resampling

- To describe this procedure using more rigorous mathematical terms, we have

$$\widehat{\boldsymbol{X}}_{1:n} \sim \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

where

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

and $\boldsymbol{X}_{1:n}^{(i)}$ denote the random samples.

- We make resampling from the probability mass $\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$

  i.e. we resample *N* times from the current approximation.

- This corresponds to an approximation of $\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ that we write as

$$\tilde{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),\ \left(N_n^{(1)},\ldots,N_n^{(N)}\right) \sim \mathcal{M}\left(N; W_n^{(i)},\ldots,W_n^{(N)}\right),\ \mathbb{E}\left[N_n^{(i)}\right] = NW_n^{(i)},$$

$$Var\left[N_n^{(i)}\right] = NW_n^{(i)}\left(1 - W_n^{(i)}\right),\ \sum_{i=1}^{N} N_n^{(i)} = N,\ and\ N_n^{(i)}: offsprings\ of\ \boldsymbol{X}_{1:n}^{(i)}$$

Resampling

- Note that resampling introduces additional errors – in particular, we can show that for any estimator/test function, the variance increases:

$$Var_{\tilde{p}_N}[\varphi(\boldsymbol{x}_{1:n})] \geq Var_{\hat{p}_N}[\varphi(\boldsymbol{x}_{1:n})]$$

- Even though you loose in terms of variance, resampling can be beneficial at future time steps n.

- Finally, it is interesting to note that the resampling operation is an unbiased one, i.e.

$$\mathbb{E}[\tilde{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})|\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})] = \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

Multinomial Resampling

- Here is a simple resampling technique called "Multinomial Resampling"
  - suppose at certain time step, we already have *N* samples $\left\{ \boldsymbol{X}_{1:n}^{(i)} \right\}$ with normalized importance weights $\left\{ W_n^{(i)} \right\}$
  - The resampling procedure is
    - consider a random variable *u* which takes integer values *1,2,3,...,N*
    - the probability corresponding to value *i* is $W_n^{(i)}$
    - generate *N* realizations from the random variable and get u_1, u_2, ..., u_N

Multinomial Resampling

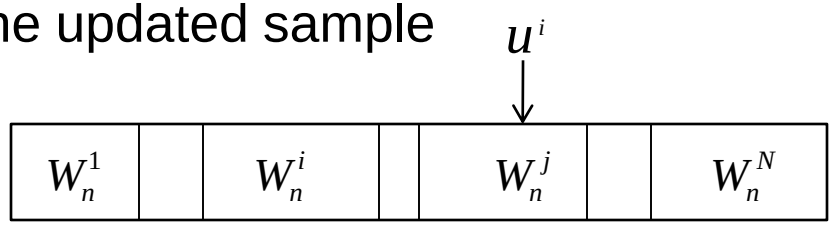
Continue

- select *N* elements from $\left\{ \boldsymbol{X}_{1:n}^{(i)} \right\}$ according to indices $u_1, u_2, \ldots, u_N$, i.e. we get *N* random vectors

$$\boldsymbol{X}_{1:n}^{u_1}, \boldsymbol{X}_{1:n}^{u_2}, \boldsymbol{X}_{1:n}^{u_3}, \ldots, \boldsymbol{X}_{1:n}^{u_m}$$

  which are the resampled results. The importance weights corresponding to them are identical equal to *1/N*

- A realization of the algorithm is
  - In practice, we sample $u^i \sim \mathcal{U}[0,1]$ i.i.d.
  - $\hat{X}_n^i$ denotes the updated sample



$$\hat{X}_n^i \leftarrow X_n^j$$

$$\hat{\imath}^i \leftarrow j$$

Resampling

- Resampling corresponds to the approximation:

$$\sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{\mathbf{x}_{1:n}^{(i)}}(\mathbf{x}_{1:n}) \approx \sum_{i=1}^{N} W_n^{(i)} \delta_{\mathbf{x}_{1:n}^{(i)}}(\mathbf{x}_{1:n})$$

where $N_n^{(i)}$ is the number of offsprings of the particle $\mathbf{x}_{1:n}^{(i)}$ and $\sum_{i=1}^{N} N_n^{(i)} = N.$

- The resampling scheme is essentially equivalent to sampling from a multinomial distribution:

$$\left(N_n^{(1)}, N_n^{(2)}, ..., N_n^{(N)}\right) \sim \mathcal{M}\left(N; W_n^{(1)}, W_n^{(2)}, ..., W_n^{(N)}\right)$$

- We know that the mean and variance of the multinomial is:

$$\mathbb{E}\left[N_n^{(i)}\right] = NW_n^{(i)}, Var\left[N_n^{(i)}\right] = NW_n^{(i)}\left(1 - W_n^{(i)}\right)$$

- Other resampling algorithms have the same $\mathbb{E}\left[N_n^{(i)}\right]$ but differ on

$$Var\left[N_n^{(i)}\right] < NW_n^{(i)}\left(1 - W_n^{(i)}\right) \text{ and } Cov\left[N_n^{(i)}, N_n^{(j)}\right]$$

Stratified Resampling Method

- We still make sampling from $\hat{p}^N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$ but we use alternative steps. The procedure is

  - Partition the interval (0,1] into N disjoint sets

$$(0,1] = (0, \frac{1}{N}] \cup \cdots \cup (\frac{N-1}{N}, 1]$$

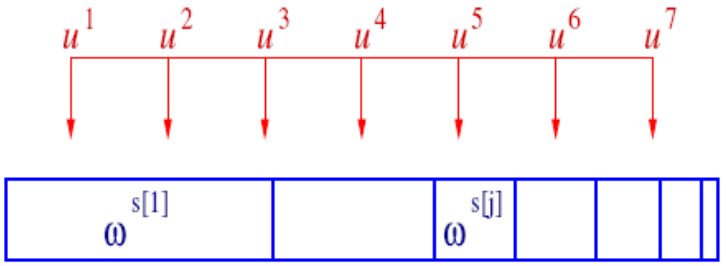
  - Draw independently in each of the sub-intervals

$$U_i \sim \mathcal{U}\left(\frac{i-1}{N}, \frac{i}{N}\right) \text{ for each } i = 1, 2, ..., N$$

  - The subsequent procedure is the same to multinomial resampling. We divide the interval [0,1] into subintervals with length $W_1, W_2, \ldots, W_N$. For each U_i, we determine which subinterval it follows and then take the corresponding raw samples as the new ones.

Stratified Resampling Method

- Given U_i, if $\sum_{k=1}^{j-1} W_n^{(k)} \le U_i \le \sum_{k=1}^{j} W_n^{(k)}$ (find in which subinterval U_i belongs), we pick up the $s(j)$ sample $X_{1:n}^{s(j)}$ in that subinterval to be the resampled result. It is easy to check that :



$$\hat{i}^i \leftarrow s[j]$$

$$N_n^{(j)} = \#\left\{ U_i : \sum_{k=1}^{i-1} W_n^{(k)} \le U_i \le \sum_{k=1}^{j} W_n^{(k)} \right\}, \text{where} : \sum_{k=1}^{0} = 0$$

$$\mathbb{E}\left[N_n^{(j)} \right] = N W_n^{(j)}$$

R. Doucet, et al, Comparison of resampling techniques for particle filtering

Sequential Importance Sampling (SIS) with Resampling

- Incorporating the resampling procedure, we present the framework for Sequential Monte Carlo

SIS/Resampling Monte Carlo Filter

1. Importance Sampling

For times *n=0,1,2,...*

- For *i=1,...,N*, sample $X_n^{(i)} \sim q\left(x_n \mid X_{n-1}^{(i)}, y_n\right)$ and set $\boldsymbol{X}_{1:n}^{(i)} = \left(\boldsymbol{X}_{1:n-1}^{(i)}, X_n^{(i)}\right)$
- For *i=1,...,N*, evaluate the importance weights up to a normalizing constant

$$w_n^{(i)} = w_{n-1}^{(i)} \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{q_n\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}$$

- For *i=1,...,N*, normalize the importance weights

$$W_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{N} w_n^{(j)}}$$

- Evaluate effective Sample Size N_{eff}

SIS with Resampling

SIS/Resampling Monte Carlo Filter

2. Resampling

If $N_{eff}>=N_{thres}$,

- No resampling

Otherwise

- For *i=1,…,N*, sample an index *j(i)* distributed according to the discrete distribution with *N* elements satisfying $\Pr\{j(i)=l\}=W_n^{(l)}$ for l=1,…,N.

- For *i=1,…N*, $X_n^{(i)}=X_n^{j(i)}$ and $W_n^{(i)}=1/N$

Sequential Importance Sampling (SIS) with Resampling

- Time 1:

  Sample N particles $X_1^{(i)} \sim q(x_1 \mid y_1)$ and compute:

$$w\left(X_1^{(i)}, y_1\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 \mid X_1^{(i)}\right)}{q\left(X_1^{(i)} \mid y_1\right)}, W_1^{(i)} \propto w\left(X_1^{(i)}, y_1\right)$$

  Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles $\left\{X_1^{(i)}\right\}$

- Time n, n≥2

  Sample N particles $X_n^{(i)} \sim q(x_n \mid y_n, X_{n-1}^{(i)})$ and compute:

$$w\left(\boldsymbol{X}_{n-1:n}^{(i)}, y_n\right) = \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{q\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}, W_n^{(i)} \propto w\left(\boldsymbol{X}_{n-1:n}^{(i)}, y_n\right)$$

  Resample $\left\{\boldsymbol{X}_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles $\left\{\boldsymbol{X}_{1:n}^{(i)}, \frac{1}{N}\right\}$

Here we consider resampling at each step. Thus that in our weight update formulas there are no more incremental weights!

Sequential Importance Sampling (SIS) with Resampling

- Resampling should take place when the effective sample size (ESS) is less than a threshold (typically N/2).

- Our update formula without (or before) resampling is:

$$w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{q\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}, W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right)$$

$$\hat{p}_N(\boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- With (after) resampling the update formula is as follows:

$$w\left(\boldsymbol{X}_{n-1:n}^{(i)}, y_n\right) = \frac{f\left(X_n^{(i)} \mid X_{n-1}^{(i)}\right) g\left(y_n \mid X_n^{(i)}\right)}{q\left(X_n^{(i)} \mid X_{n-1}^{(i)}, y_n\right)}$$

$$\tilde{p}_N(\boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- Note that in this last case, we only need to keep track of $\left\{\boldsymbol{X}_{n-1:n}^{(i)}\right\}$.

Computing $p(y_n|\boldsymbol{y}_{1:n-1})$

- Similarly, one can approximate other distributions of interest, e.g.

$$p(y_n \mid \boldsymbol{y}_{1:n-1}) = \int p(y_n, x_n, x_{n-1} \mid \boldsymbol{y}_{1:n-1})\, d\boldsymbol{x}_{n-1:n} = \int p(y_n, x_n \mid x_{n-1}, \boldsymbol{y}_{1:n-1})\, p(x_{n-1} \mid \boldsymbol{y}_{1:n-1})\, d\boldsymbol{x}_{n-1:n}$$

$$= \int \frac{g(y_n \mid x_n)\, f(x_n \mid x_{n-1})}{q(x_n \mid y_n, x_{n-1})}\, q(x_n \mid y_n, x_{n-1}) \underbrace{p(x_{n-1} \mid \boldsymbol{y}_{1:n-1})}_{\sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{X_{n-1}^{(i)}}(x_{n-1})} d\boldsymbol{x}_{n-1:n}$$

$$= \sum_{i=1}^{N} W_{n-1}^{(i)} \int \frac{g(y_n \mid x_n)\, f(x_n \mid X_{n-1}^{(i)})}{q(x_n \mid y_n, X_{n-1}^{(i)})} \underbrace{q(x_n \mid y_n, X_{n-1}^{(i)})}_{\delta_{X_n^{(i)}}(x_n)} dx_n$$

$$= \sum_{i=1}^{N} W_{n-1}^{(i)} \frac{g(y_n \mid X_n^{(i)})\, f(X_n^{(i)} \mid X_{n-1}^{(i)})}{q(X_n^{(i)} \mid y_n, X_{n-1}^{(i)})}$$

Computing $\hat{p}(\boldsymbol{y}_{1:n})$

- The particle approximation of $p(y_n \mid \boldsymbol{y}_{1:n-1})$ is then:

$$\hat{p}_N(y_n|\boldsymbol{y}_{1:n-1}) = \sum_{i=1}^{N} W_{n-1}^{(i)} \frac{g\left(y_n|X_n^{(i)}\right) f\left(X_n^{(i)}|X_{n-1}^{(i)}\right)}{q\left(X_n^{(i)}|y_n, X_{n-1}^{(i)}\right)}$$

- These approximations are useful in computing the marginal likelihood

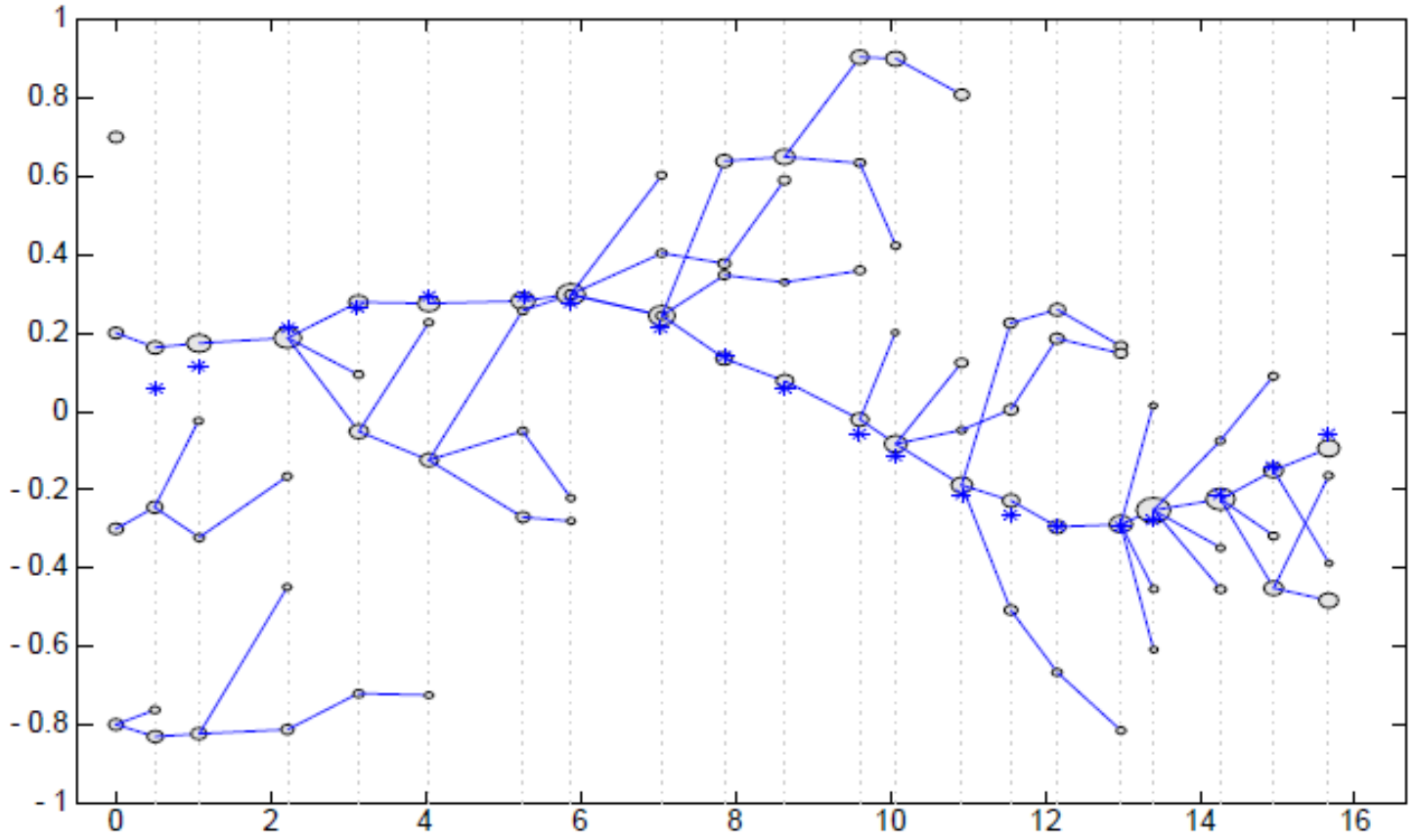
$$\hat{p}(\boldsymbol{y}_{1:n}) = \hat{p}(y_1) \prod_{k=2}^{n} \hat{p}(y_k|\boldsymbol{y}_{1:k-1})$$

- Note that this is an unbiased estimate:

$$\mathbb{E}[\hat{p}(\boldsymbol{y}_{1:n})] = p(\boldsymbol{y}_{1:n})$$

Resampling

- Demonstration of Importance Sampling/Resampling for four particles (from A. Doucet). The size of the particles is proportional to their weight. The bigger the dot size, the more offsprings you will get.



Sample Impoverishment

- When resampling the particles with high weight will be selected many times, thus there is a loss of diversity amongst the population.
- In the extreme case of no process noise (e.g., if we have static but unknown parameters as part of the state space), then all the particles will collapse to a single point within a few iterations.
- To mitigate this problem, several solutions have been proposed.

- Gordon, N. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceeding (F) 140*(2), 107–113.
- Gilks, W. and C. Berzuini (2001). Following a moving target – Monte Carlo infernece for dynamic Bayesian models. *J. of Royal Stat. Soc. Series B 63*, 127–146.
- Andrieu, C., A. Doucet, and V. Tadicc(2005). Online EM for parameter estimation in nonlinear-non Gaussian state-space models. In *Proc.cIEEE CDC*.

Sample Impoverishment

To mitigate this problem, several solutions have been proposed.

- (1) Only resample when necessary, not at every time step.

  The original **bootstrap filter** (Gordon 1993) resampled at every step, but this is suboptimal.

- (2) After replicating old particles, sample new values using an MCMC step which leaves the posterior distribution invariant

  See e.g., the **resample-move** algorithm in (Gilks and Berzuini 2001)

- Gordon, N. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceeding (F) 140*(2), 107–113.
- Gilks, W. and C. Berzuini (2001). Following a moving target – Monte Carlo infernece for dynamic Bayesian models. *J. of Royal Stat. Soc. Series B 63*, 127–146.

Sample Impoverishment

To mitigate this problem, several solutions have been proposed.

- (3) Create a kernel density estimate on top of the particles,

$$p(x_t|\boldsymbol{y}_{1:t}) \approx \sum_{s=1}^{N} w_t^s \kappa(x_t - x_t^s)$$

Here κ is some smoothing kernel. We then sample from this smoothed distribution.

This is known as a **regularized particle filter** (Musso et al. 2001).

- (4) When performing inference on static parameters, add some artificial process noise.

If this is undesirable, other algorithms must be used for online parameter estimation, e.g., (Andrieu et al. 2005)

- Andrieu, C., A. Doucet, and V. Tadicc(2005). Online EM for parameter estimation in nonlinear-non Gaussian state-space models. In *Proc.cIEEE CDC*.
- Musso, C., N. Oudjane, and F. LeGland (2001). Improving regularized particle filters. In A. Doucet, J. F. G. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*. Springer.

APPLYING SISR TO A LINEAR GAUSSIAN MODEL

Example: Linear Gaussian Model

- Let us use a linear Gaussian model to illustrate the SMC method.
- Consider a linear Gaussian model

$$x_0 \sim \mathcal{N}(0,1)$$

$$x_n = Ax_{n-1} + \sigma_v v_n, \;\; v_n \sim \text{i.i.d.}\, \mathcal{N}(0,1)$$

$$y_n = Bx_n + \sigma_w w_n, \;\; w_n \sim \text{i.i.d.}\, \mathcal{N}(0,1)$$

- Here, A = 0.69, σ_v = 1.12, B = 0.89 and σ_w = 0.78
- The state and observation equations are

$$x_n \sim \mathcal{N}\left(Ax_{n-1}, \sigma_v^2\right)$$

$$y_n \sim \mathcal{N}\left(Bx_n, \sigma_w^2\right)$$

Linear Gaussian Model

- The recursive form of the posterior distribution is

$$p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) \propto p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) \times f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)$$

- The suboptimal proposal distribution is

$$q\left(x_n \mid \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n}\right) = f\left(x_n \mid x_{n-1}\right) \sim \mathcal{N}\left(Ax_{n-1}, \sigma_v^2\right)$$

- The corresponding importance weight is

$$w_n = w_{n-1} \times g\left(y_n \mid x_n\right), \quad \text{where } g\left(y_n \mid x_n\right) \sim \mathcal{N}\left(Bx_n, \sigma_w^2\right)$$

Linear Gaussian Model

Recall the general SIS method with resampling

- At time *n=1*, sample $X_1^{(i)} \sim \mathcal{N}\left(AX_0^{(i)}, \sigma_v^2\right)$

  where $X_0^{(i)} \sim \mathcal{N}(0,1)$

  compute the importance weight $w_1^{(i)} = \mathcal{N}\left(BX_1^{(i)}, \sigma_w^2\right)$

- At step *n* (n>1)

  - Draw a random number $X_n^{(i)}$ from $\mathcal{N}\left(Ax_{n-1}^{(i)}, \sigma_v^2\right)$, and let $\boldsymbol{X}_{1:n}^{(i)} = \left(\boldsymbol{X}_{1:n-1}^{(i)}, X_n^{(i)}\right)$
  - Compute the importance weight

$$w_n^{(i)} = w_{n-1}^{(i)} \mathcal{N}\left(BX_n^{(i)}, \sigma_w^2\right)$$

    and normalize them by

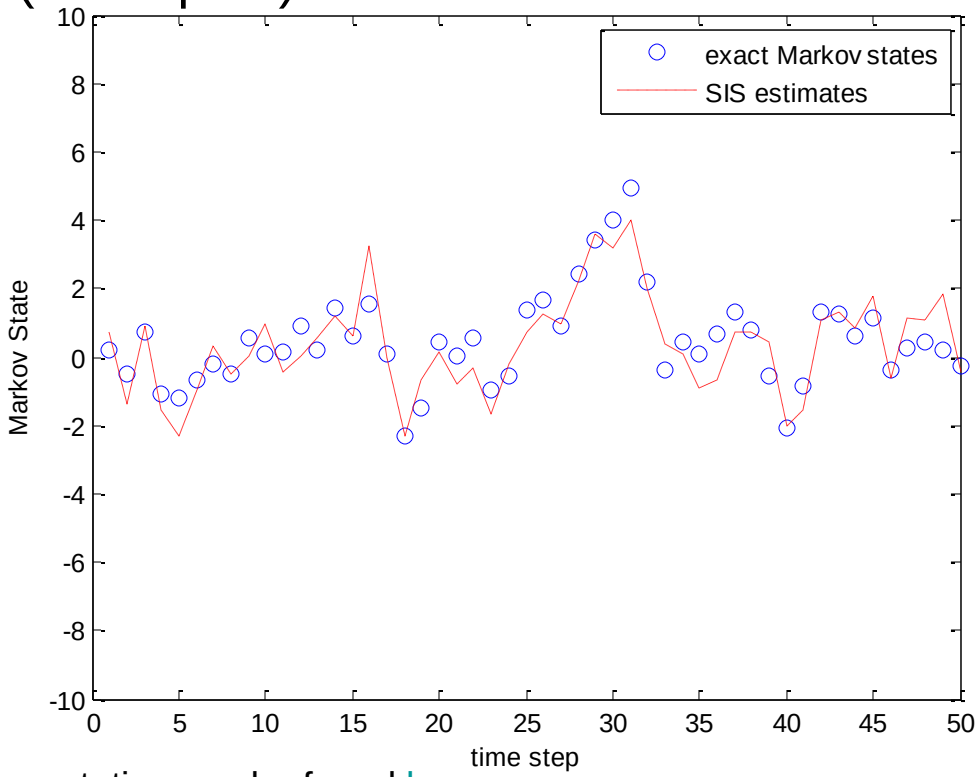
$$W_n^{(i)} = w_n^{(i)} / \sum_{i=1}^{N} w_n^{(i)}$$

  - Make resampling according to the normalized weights $W_n^{(i)}$

Linear Gaussian Model

- We simulate 2000 observations from the linear Gaussian model for estimation

- We compare the estimated random states (red line) with exact Markov states (blue spots)



A MatLab implementation can be found here

Linear Gaussian Example

- We now consider the Linear Gaussian State Space Model with

$$x_0 \sim \mathcal{N}(0,1) \; and$$
$$x_n = Ax_{n-1} + Bv_n$$
$$y_n = Cx_n + Dw_n, \; where$$
$$v_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1) \; and \; w_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$$

where A = 0.9, B = 0.2, C = 0.1, D = 0.2

- The observation is taken through 24 time steps.
- An optimal importance distribution is used
- Resampling is performed at every step

Linear Gaussian Model with Locally Optimal Importance Distrib.

- We may have other choices for the proposal distribution
- In this linear Gaussian model, we have the posterior distribution as

$$p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) \propto p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) \times f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)$$
$$= p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right) \times \mathcal{N}\left(Ax_{n-1}, \sigma_v^2\right) \mathcal{N}\left(Bx_n, \sigma_w^2\right)$$

- For this problem, the locally optimal importance distribution is a Gaussian distribution

$$q^{opt}\left(x_n \mid x_{n-1}, y_n\right) = \mathcal{N}\left(m_n, \sigma_n^2\right)$$

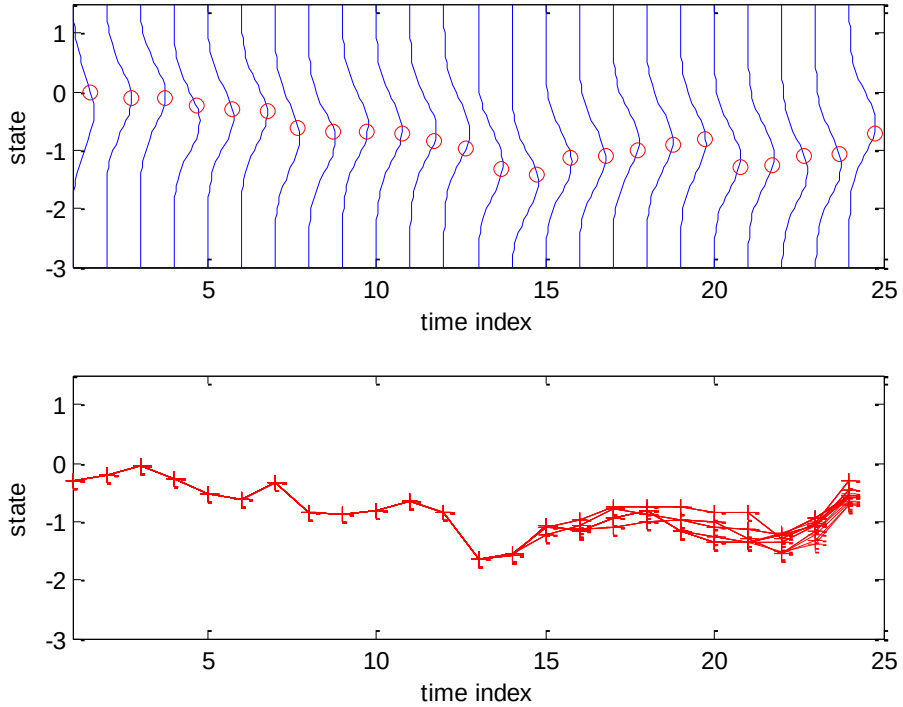
where

$$\sigma_n^{-2} = B^{-2} + \left(\frac{D}{C}\right)^{-2}$$

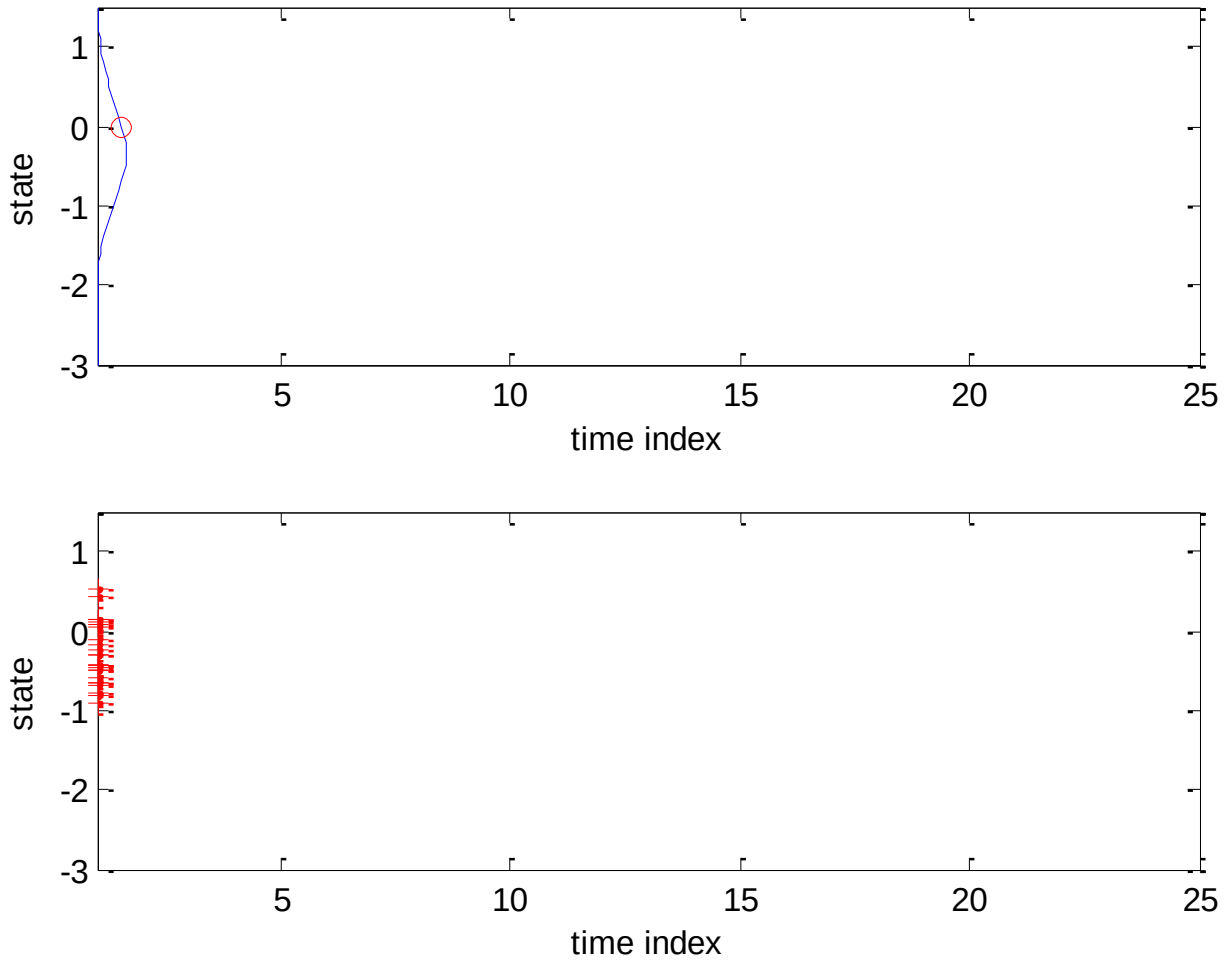
$$m_n = \sigma_n^2 \left(\frac{Ax_{n-1}}{B^2} + \frac{Cy_n}{D^2}\right)$$

Linear Gaussian Example

- The upper figure depicts the particle approximation of $p(x_n \mid \boldsymbol{y}_{1:n})$ where n = 1,2, ..., 24. The red circles denote the true Markov states $\{\boldsymbol{x}_{1:24}\}$
- The lower figure depicts the paths of 25 particles. The resampling makes the paths of the particles partially coalesce. This is due to the resampling which removes the particles with negligible weights and multiplies those with large weights.



Linear Gaussian Example



The data and a MatLab implementation can be found here

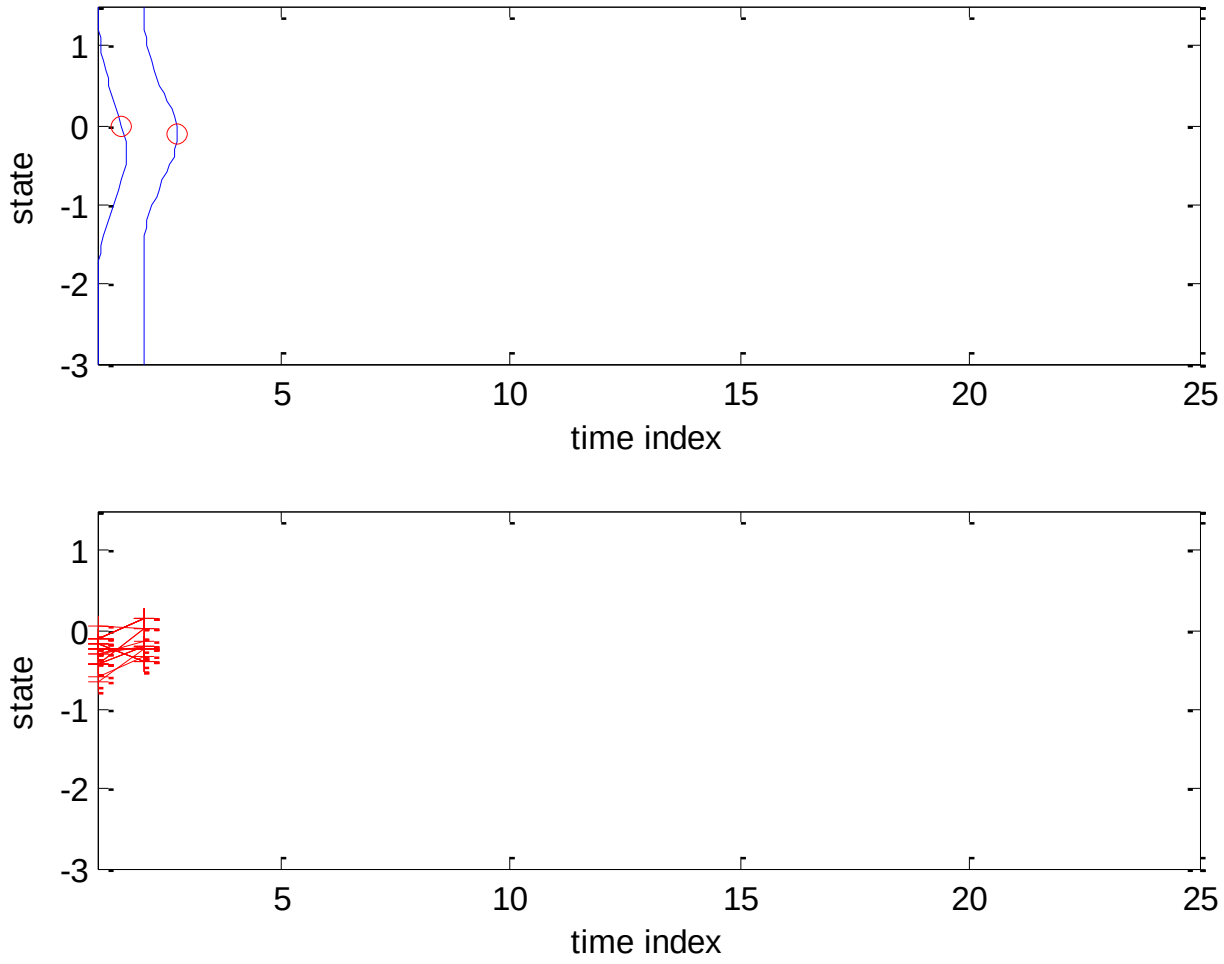
Linear Gaussian Dynamic Model

- At step 1, the target distribution is $p(x_1 \mid y_1)$. The blue curve in the upper figure represents the empirical distribution of $\hat{p}(x_1|y_1)$ obtained from particles $\left\{X_1^{(i)}, W_1^{(i)}\right\}$; while the red circle denotes the true Markov state x_1.

- The empirical distribution $\hat{p}(x_1|y_1)$ is obtained in the following way
  - resample from $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to get a new population $\left\{\tilde{X}_1^{(i)}, 1/N\right\}$, i.e. $\tilde{X}_1^{(i)} \sim \hat{p}(x_1|y_1)$
  - plot the distribution of samples $\left\{\tilde{X}_1^{(i)}\right\}$

- The particles $\left\{X_1^{(i)}\right\}$ are plotted in the bottom figure marked by symbol "+".

Linear Gaussian Example



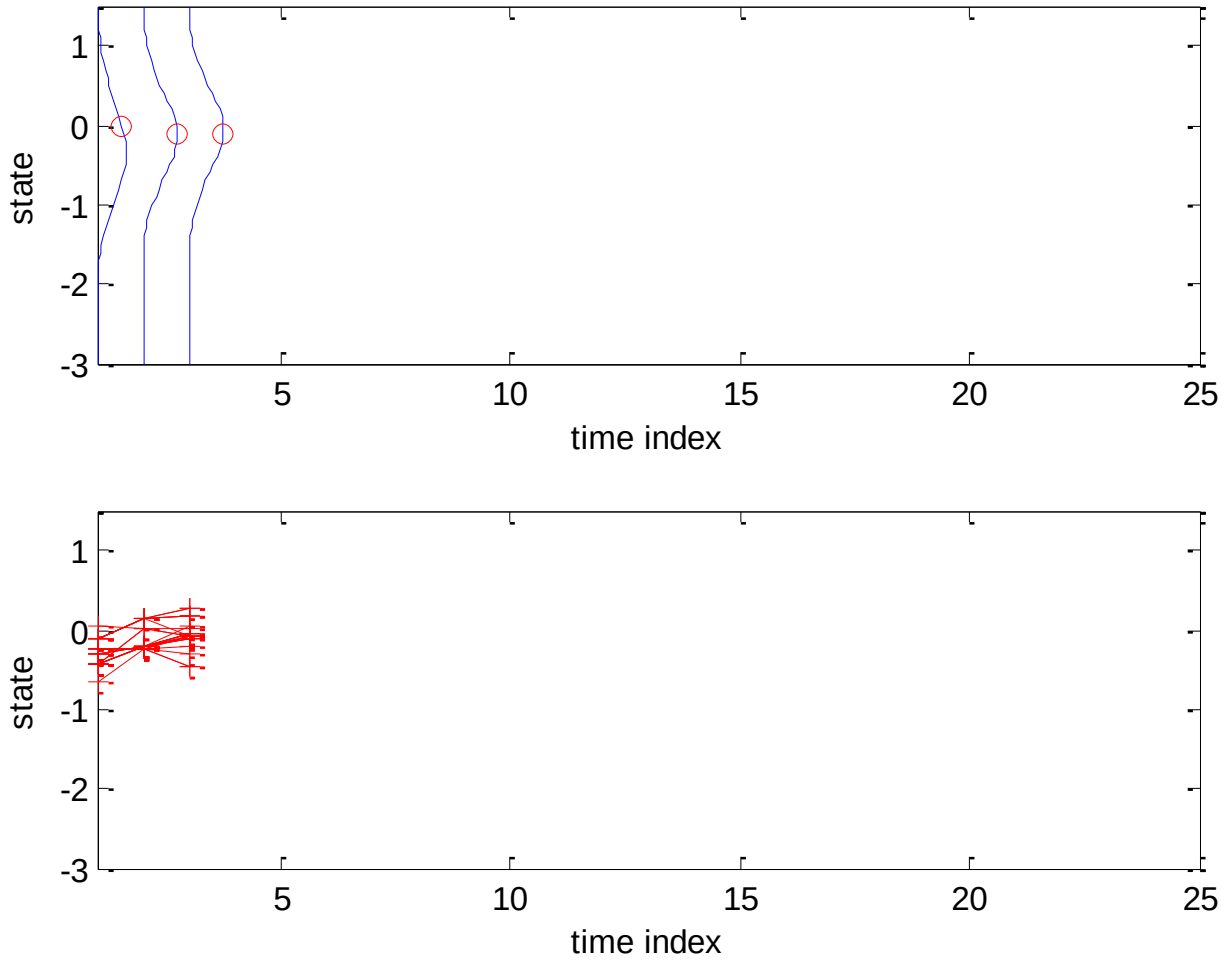
Linear Gaussian Dynamic Model

- At step 2, the target distribution is $p(x_{1:2} \mid y_{1:2})$. The blue curve in the upper figure represents the empirical distribution of $\hat{p}(x_2|\boldsymbol{y}_{1:2})$ obtained from particles $\left\{X_{1:2}^{(i)}, W_2^{(i)}\right\}$; while the red circle denotes the true Markov state $x_{1:2}$.

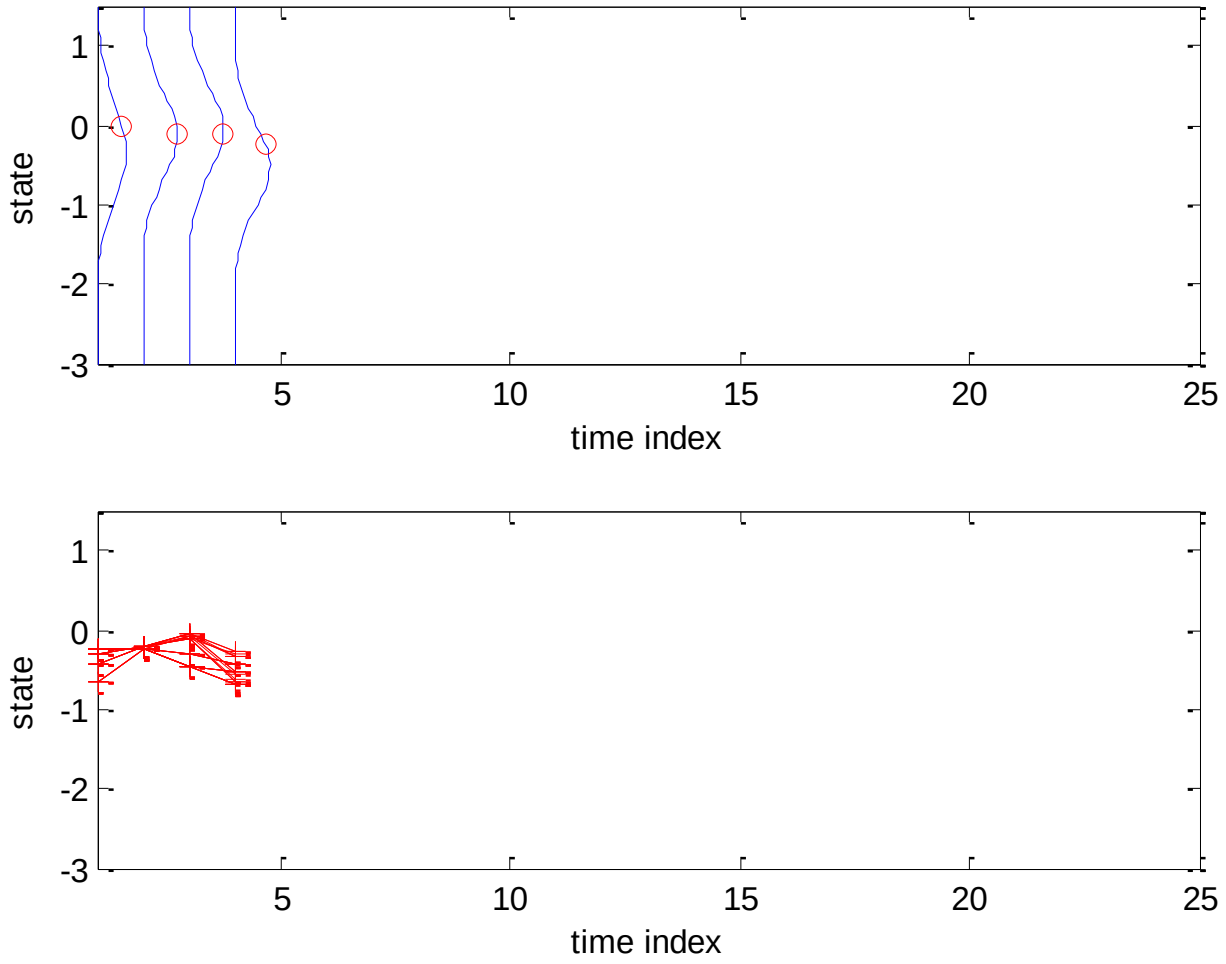
- The empirical distribution $\hat{p}(x_2|\boldsymbol{y}_{1:2})$ is obtained in the following way
  - resample from $\left\{X_{1:2}^{(i)}, W_2^{(i)}\right\}$ to get a new population $\left\{\tilde{X}_{1:2}^{(i)}, 1/N\right\}$, i.e. $\tilde{X}_{1:2}^{(i)} \sim \hat{p}(x_{1:2}|y_{1:2})$
  - plot the distribution of samples $\left\{\tilde{X}_2^{(i)}\right\}$

- The particles $\left\{X_{1:2}^{(i)}\right\}$ are plotted in the bottom figure marked by symbol "+".

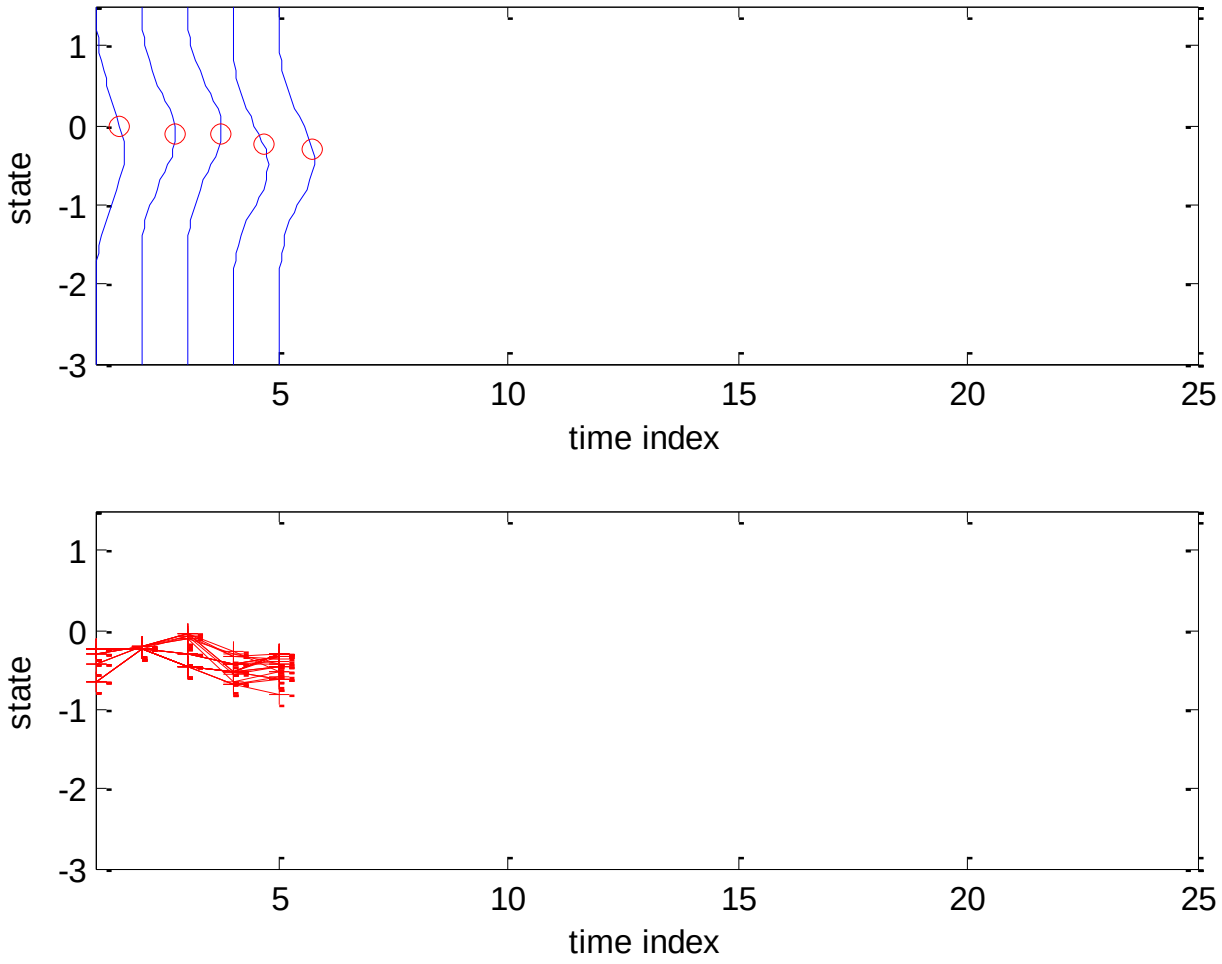
Linear Gaussian Example



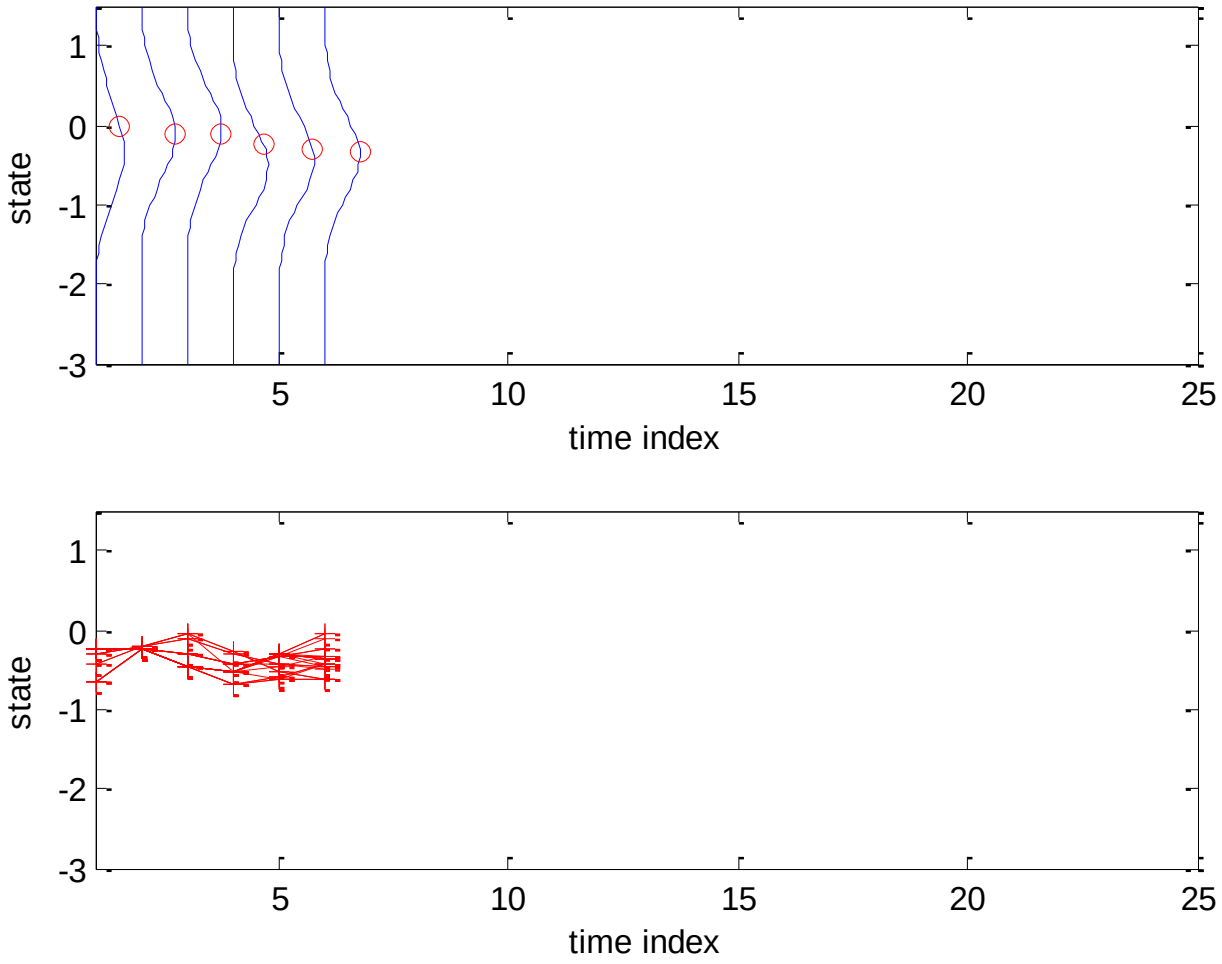
Linear Gaussian Example



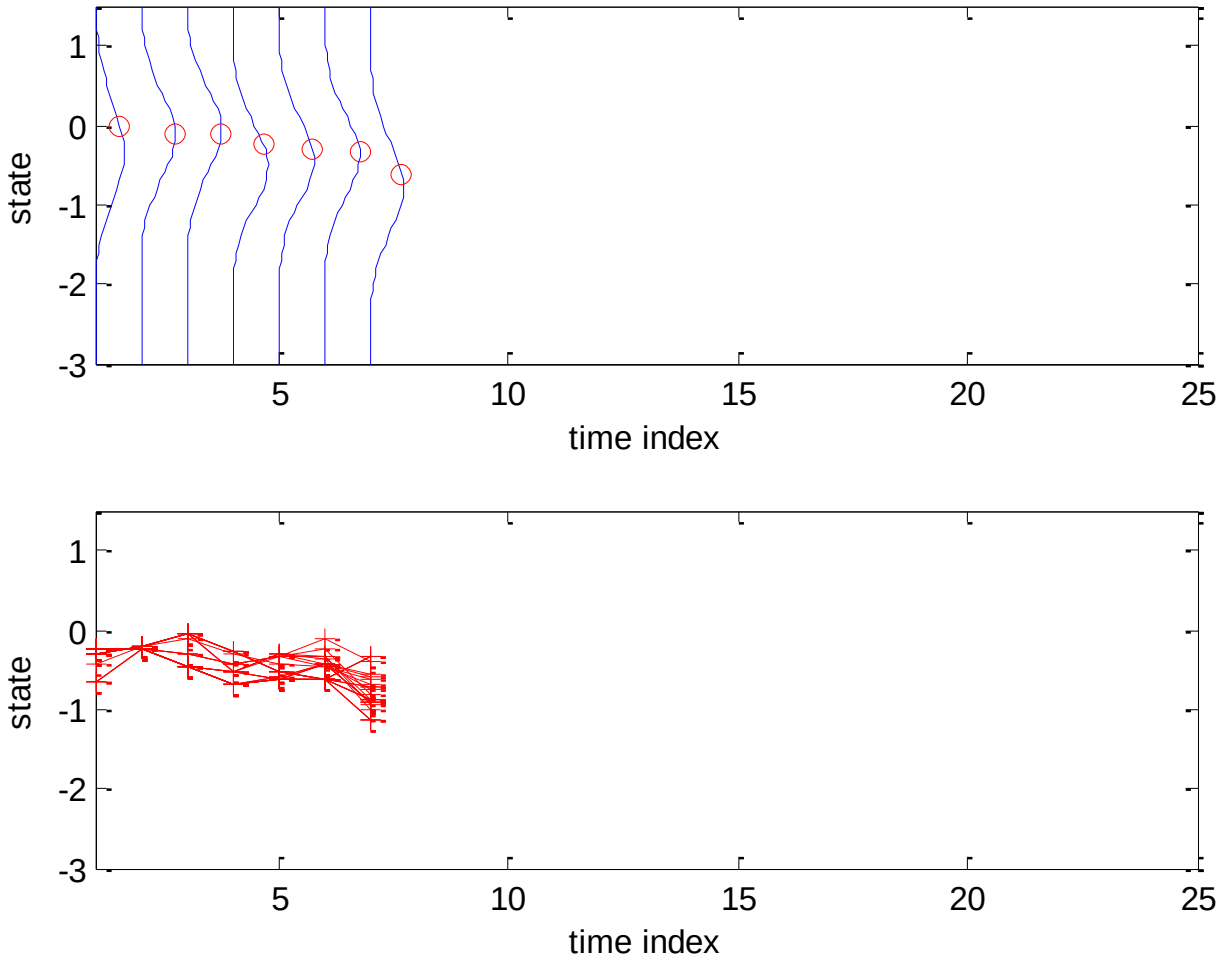
Linear Gaussian Example



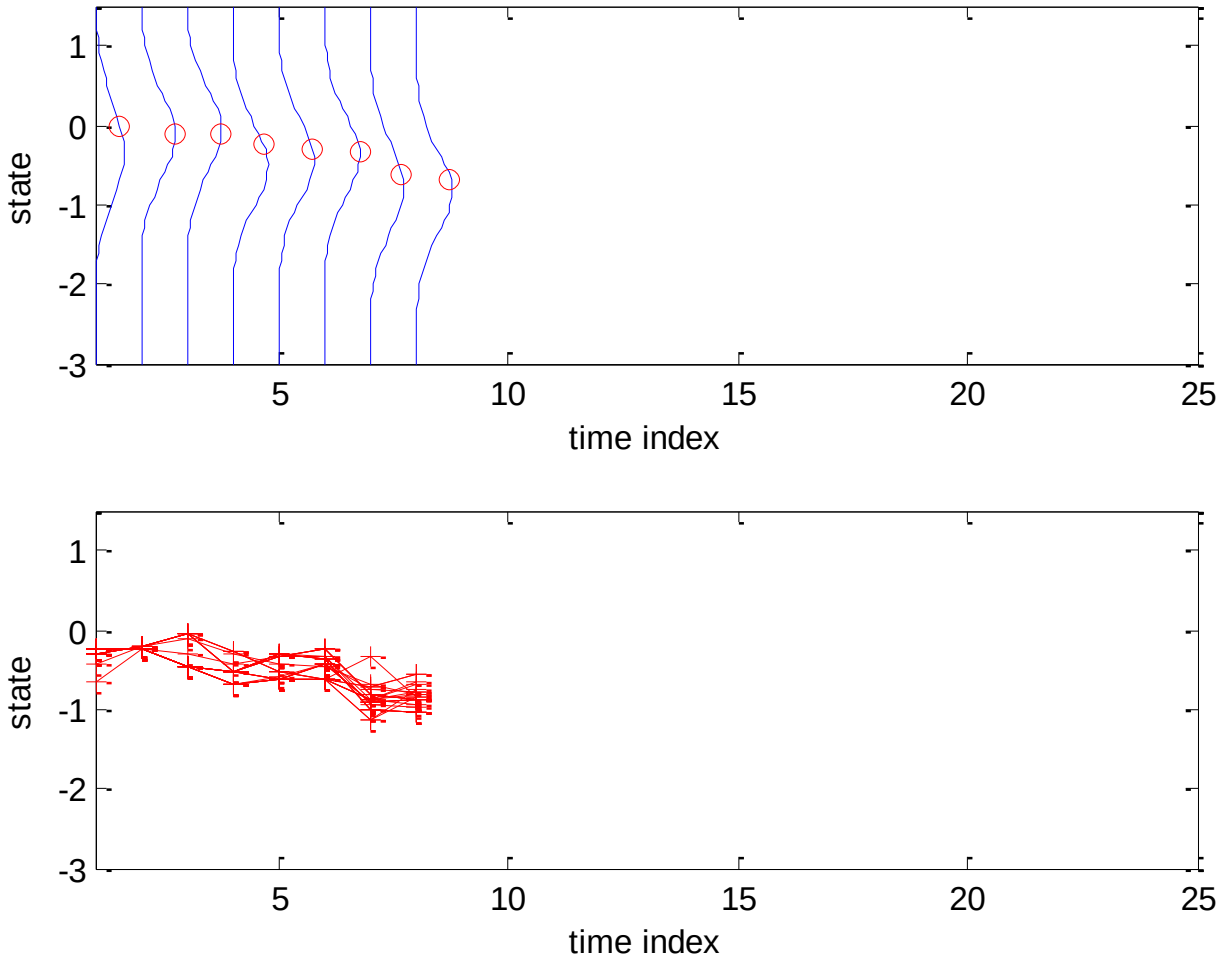
Linear Gaussian Example



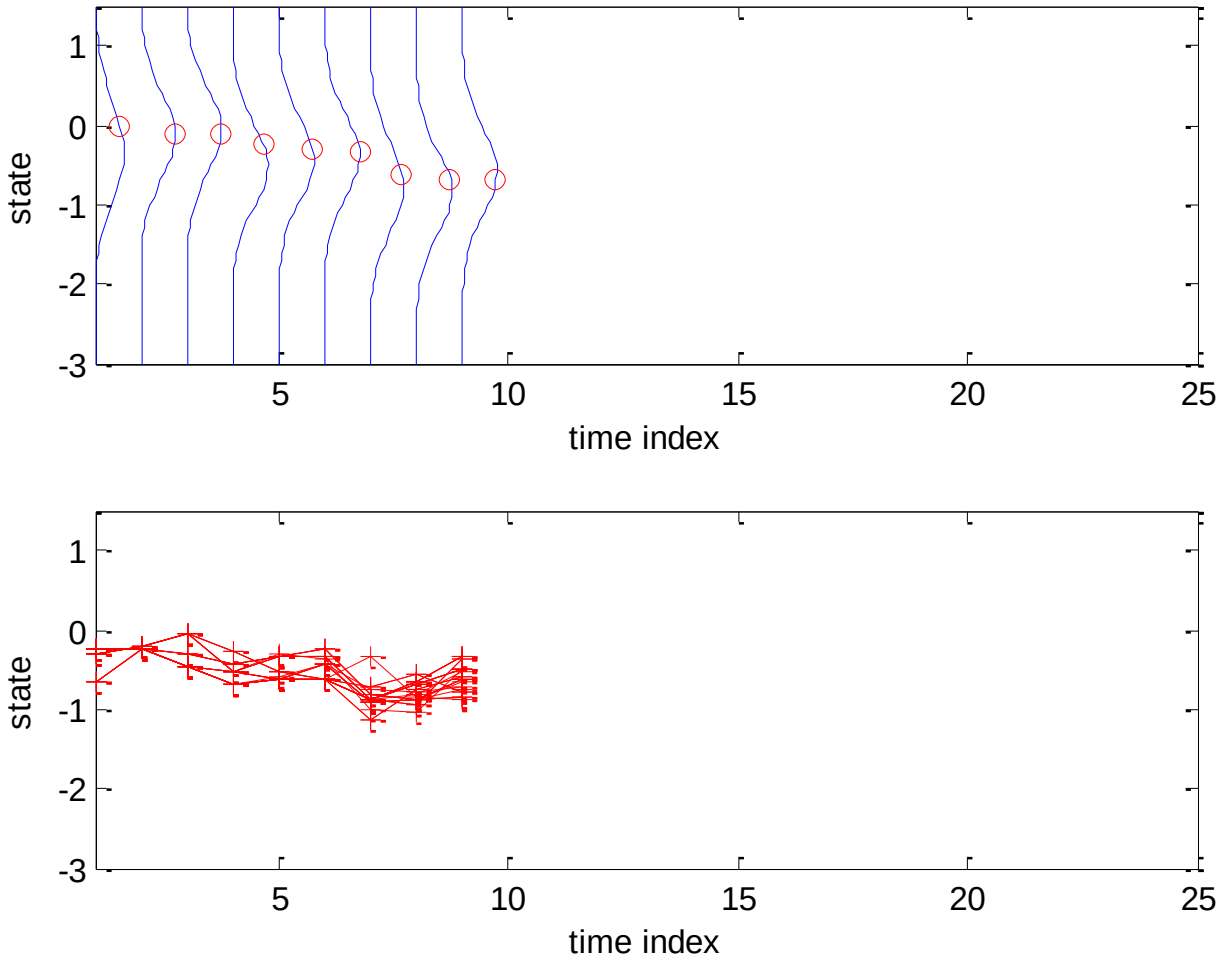
Linear Gaussian Example



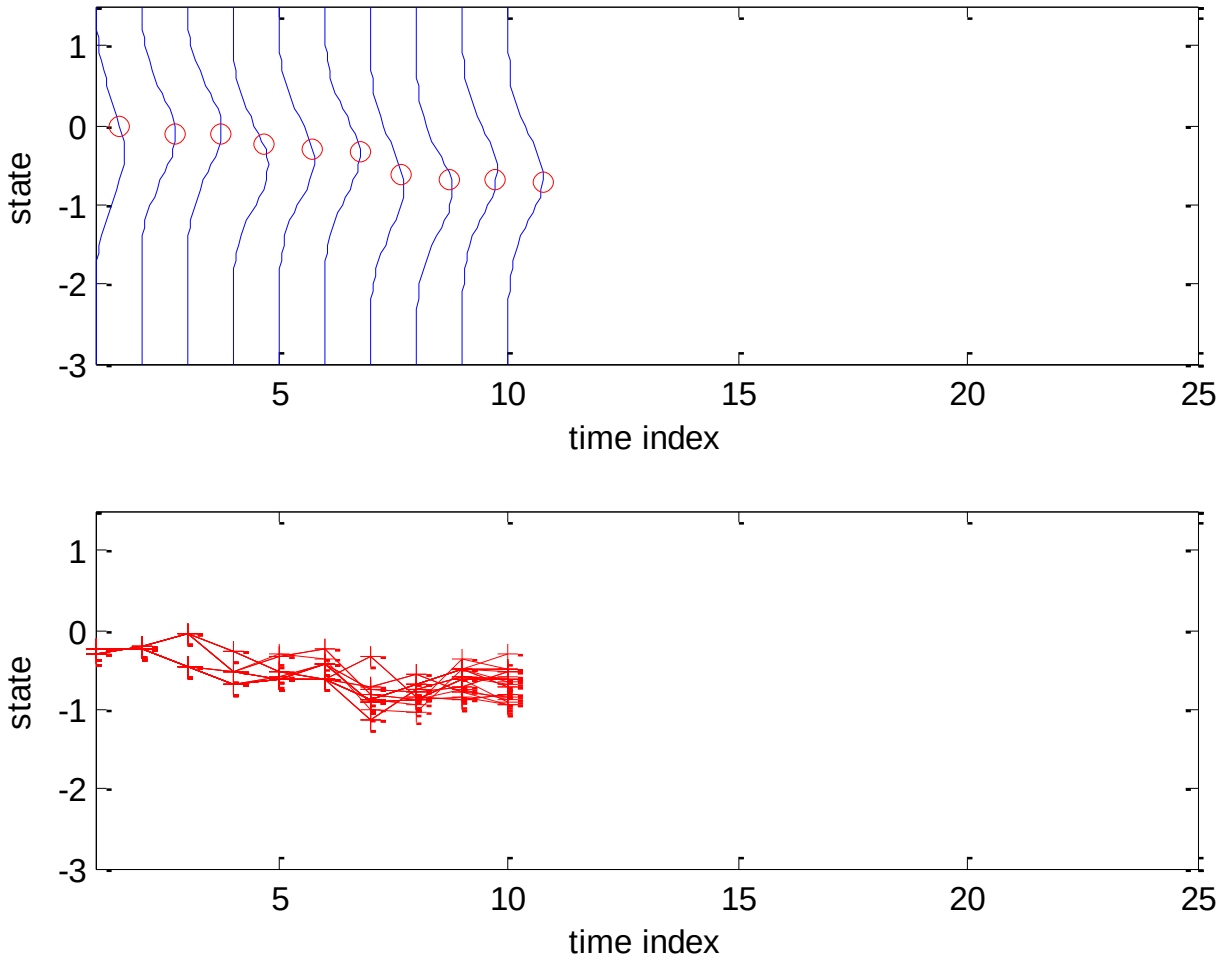
Linear Gaussian Example



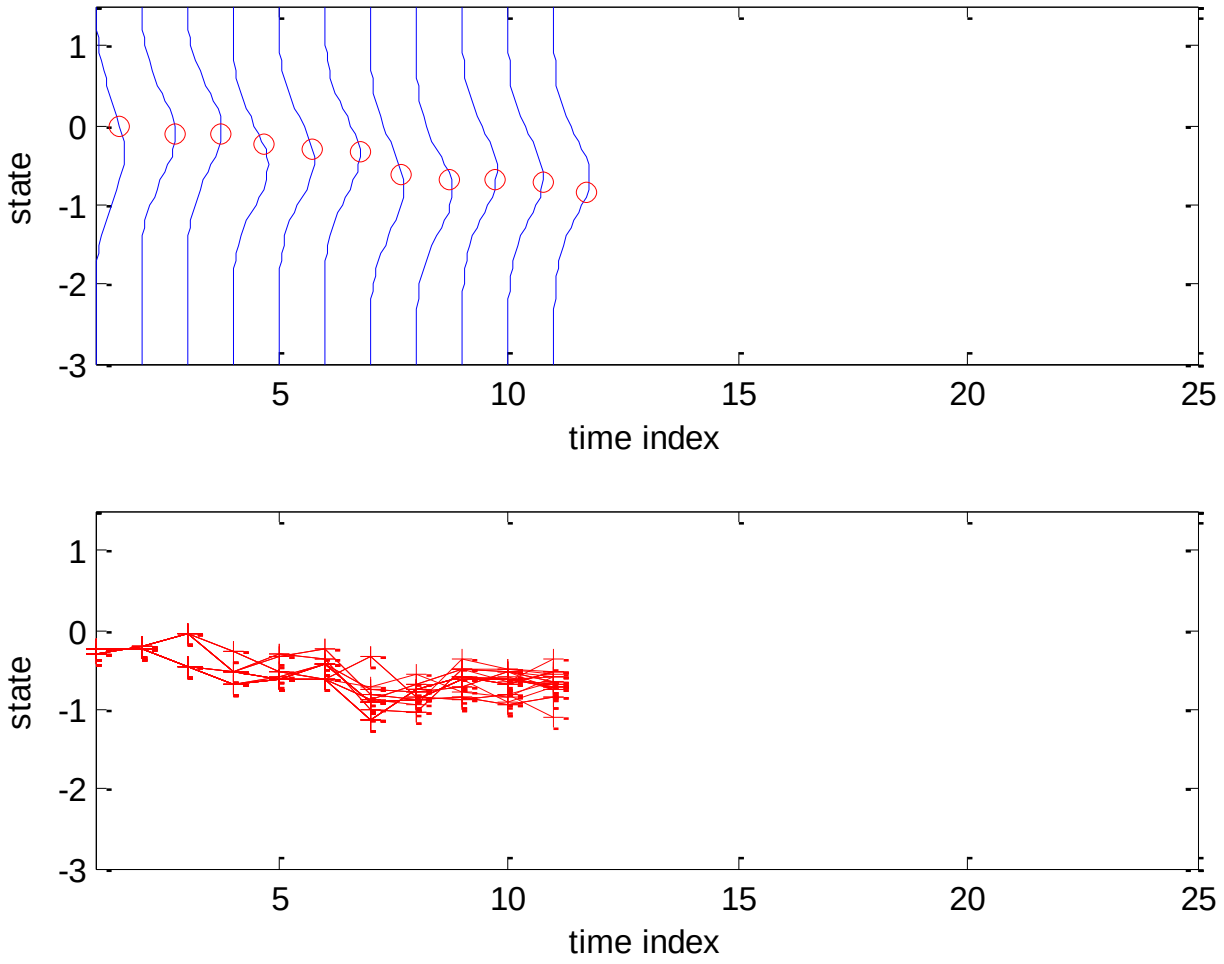
Linear Gaussian Example



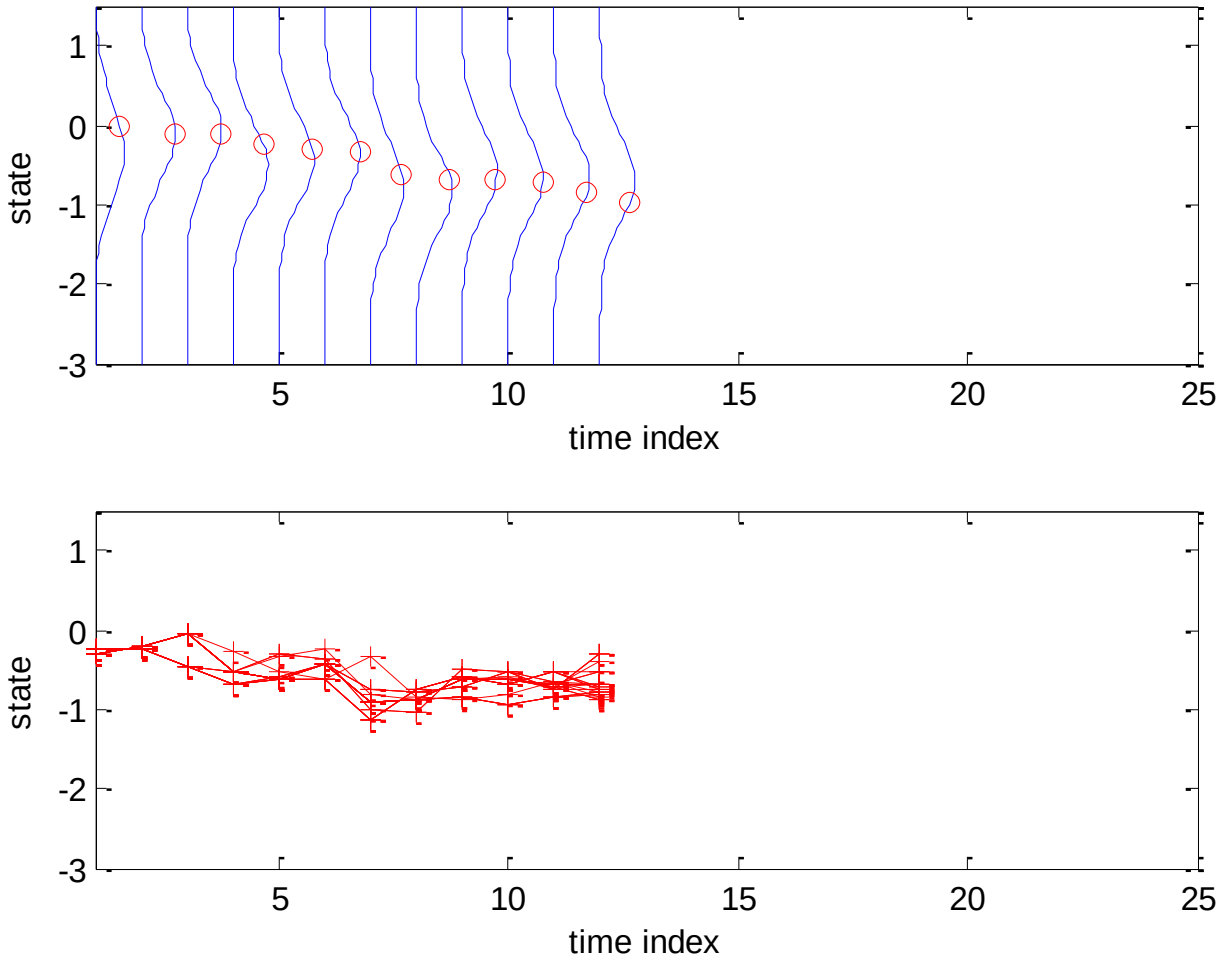
Linear Gaussian Example



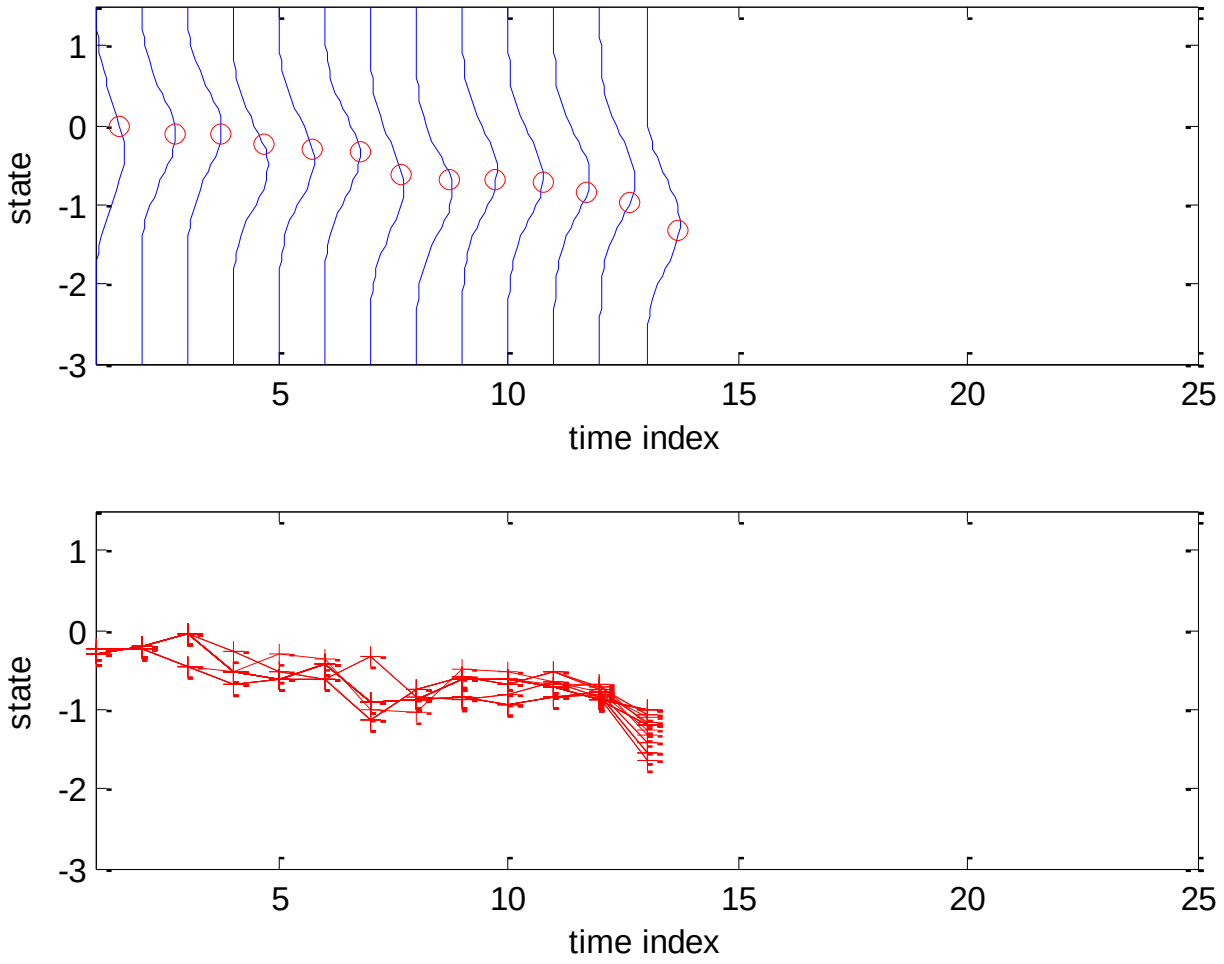
Linear Gaussian Example



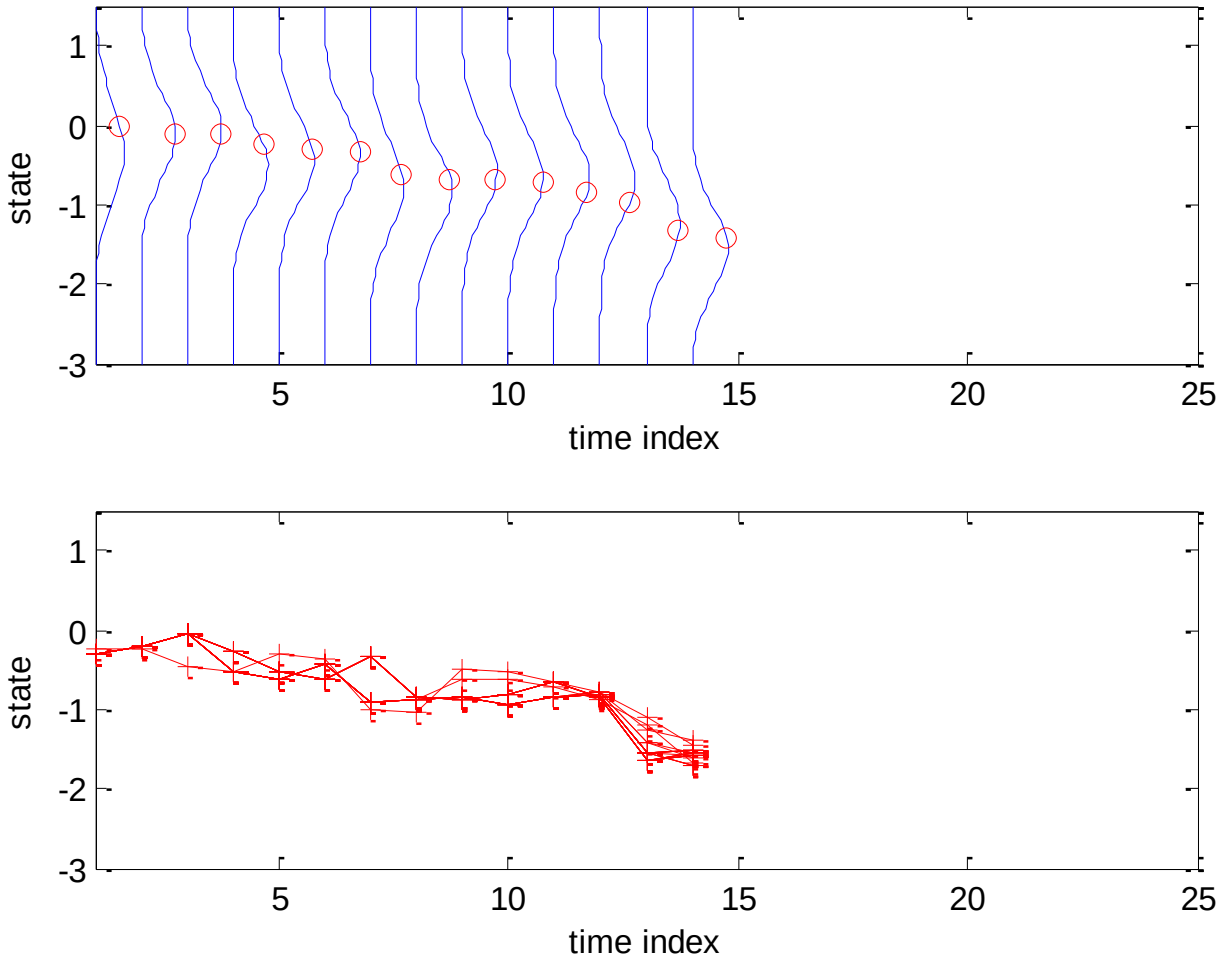
Linear Gaussian Example



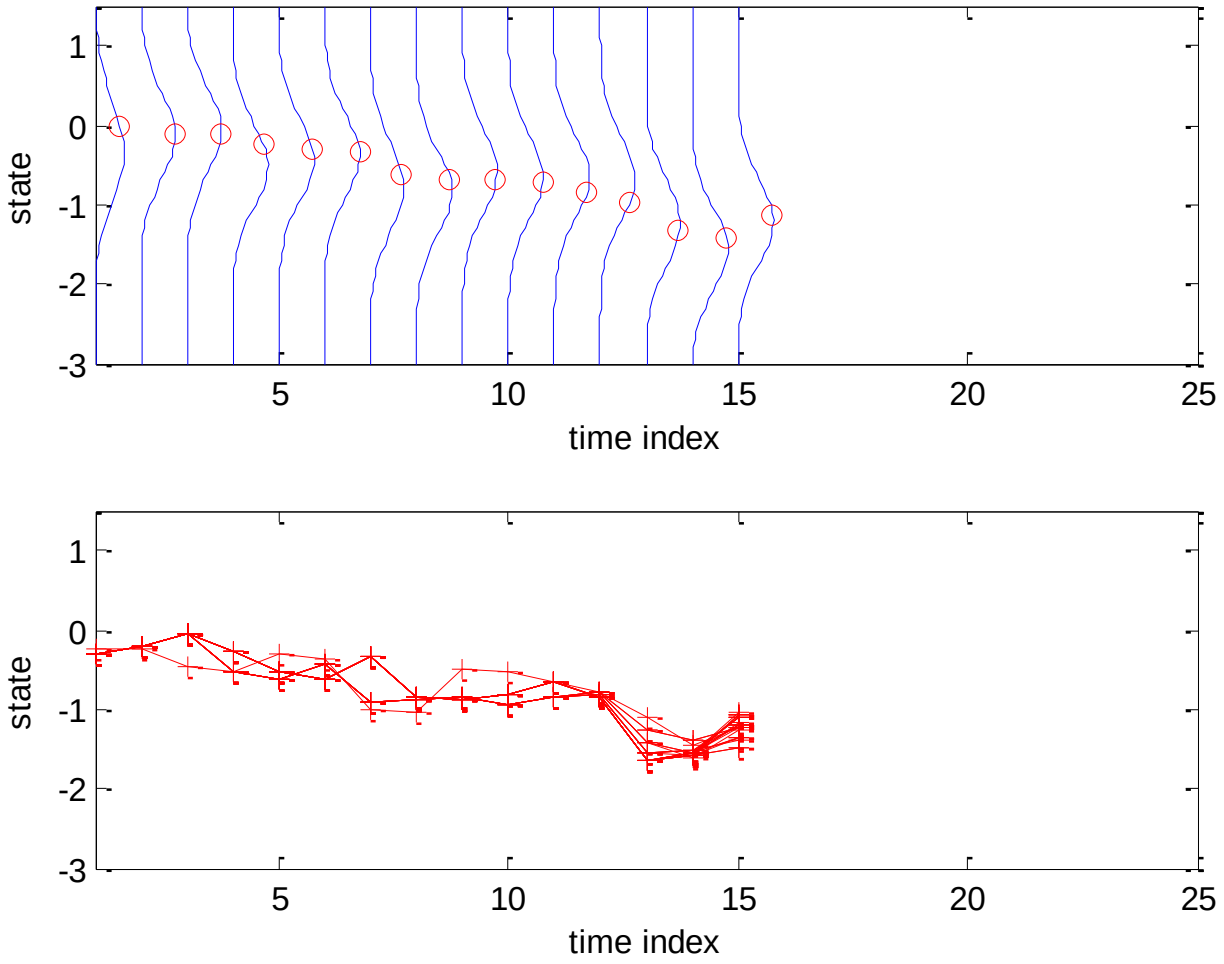
Linear Gaussian Example



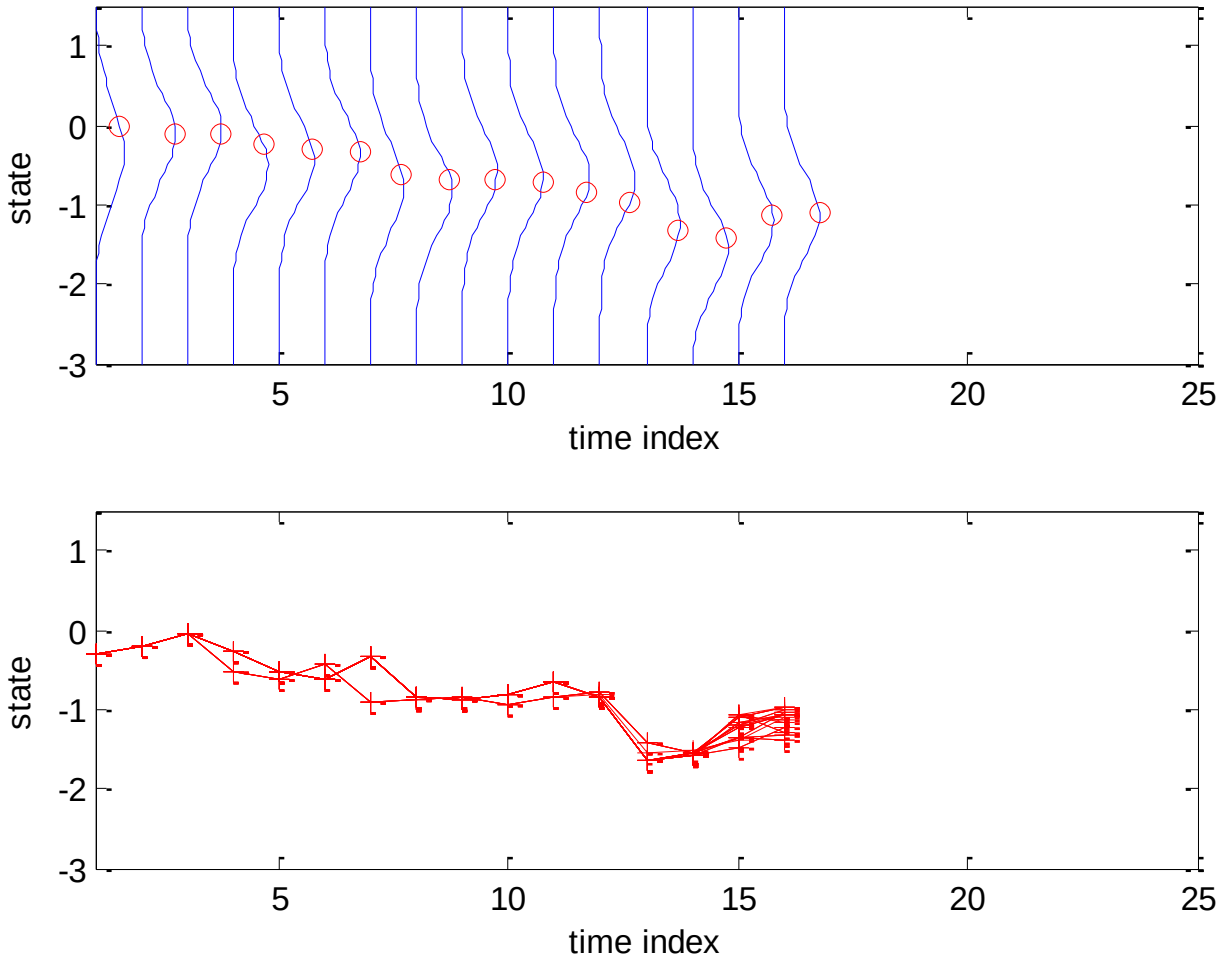
Linear Gaussian Example



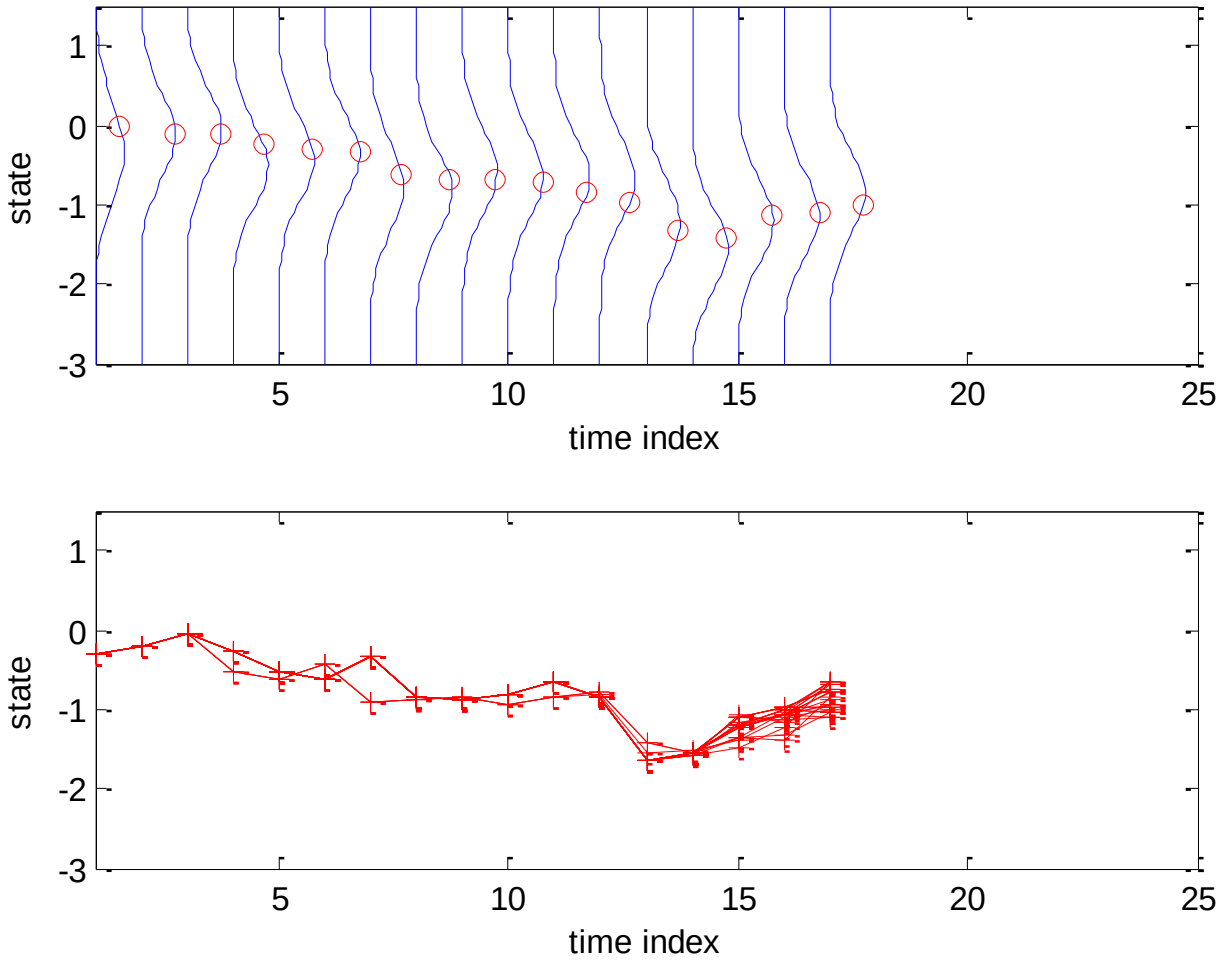
Linear Gaussian Example



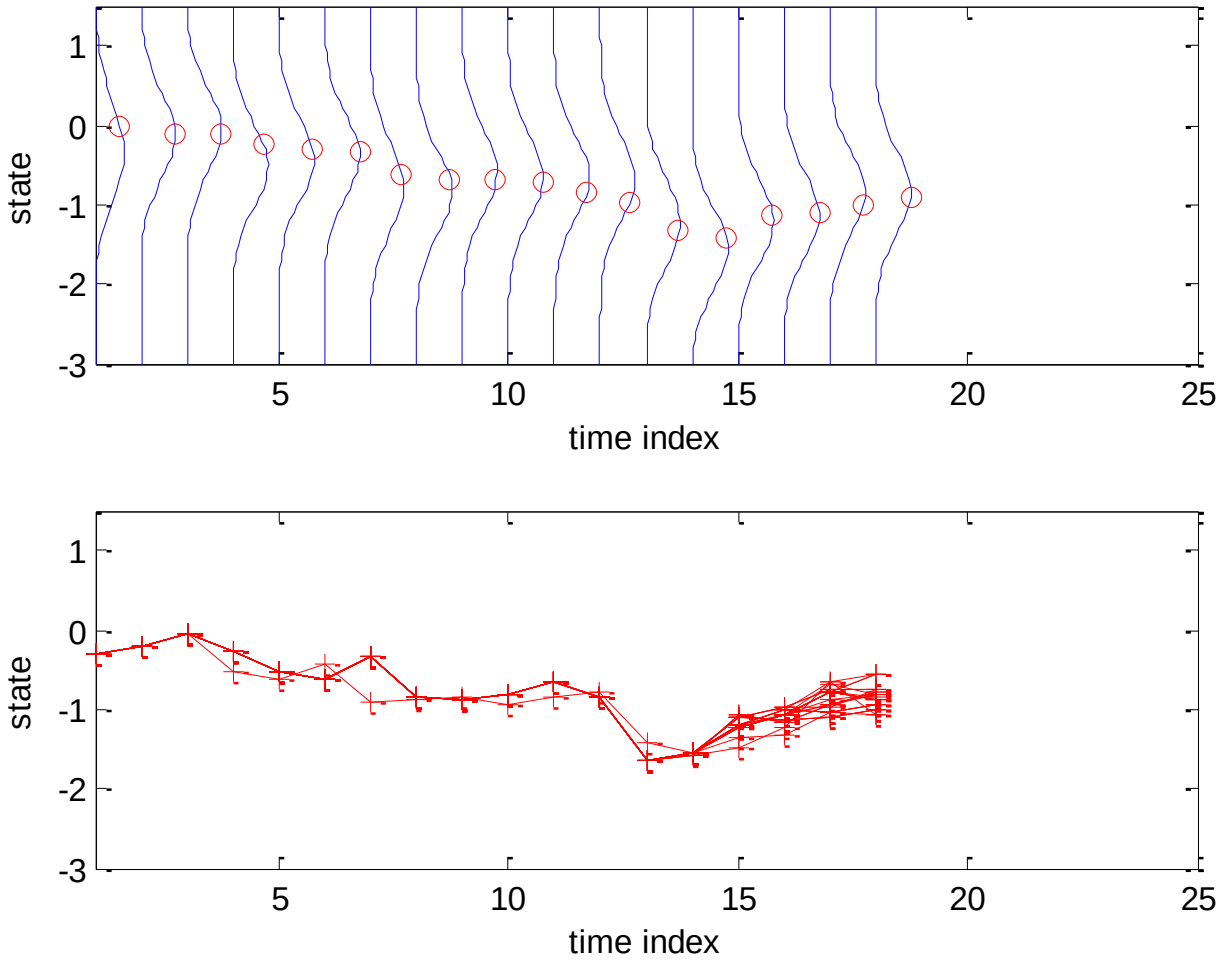
Linear Gaussian Example



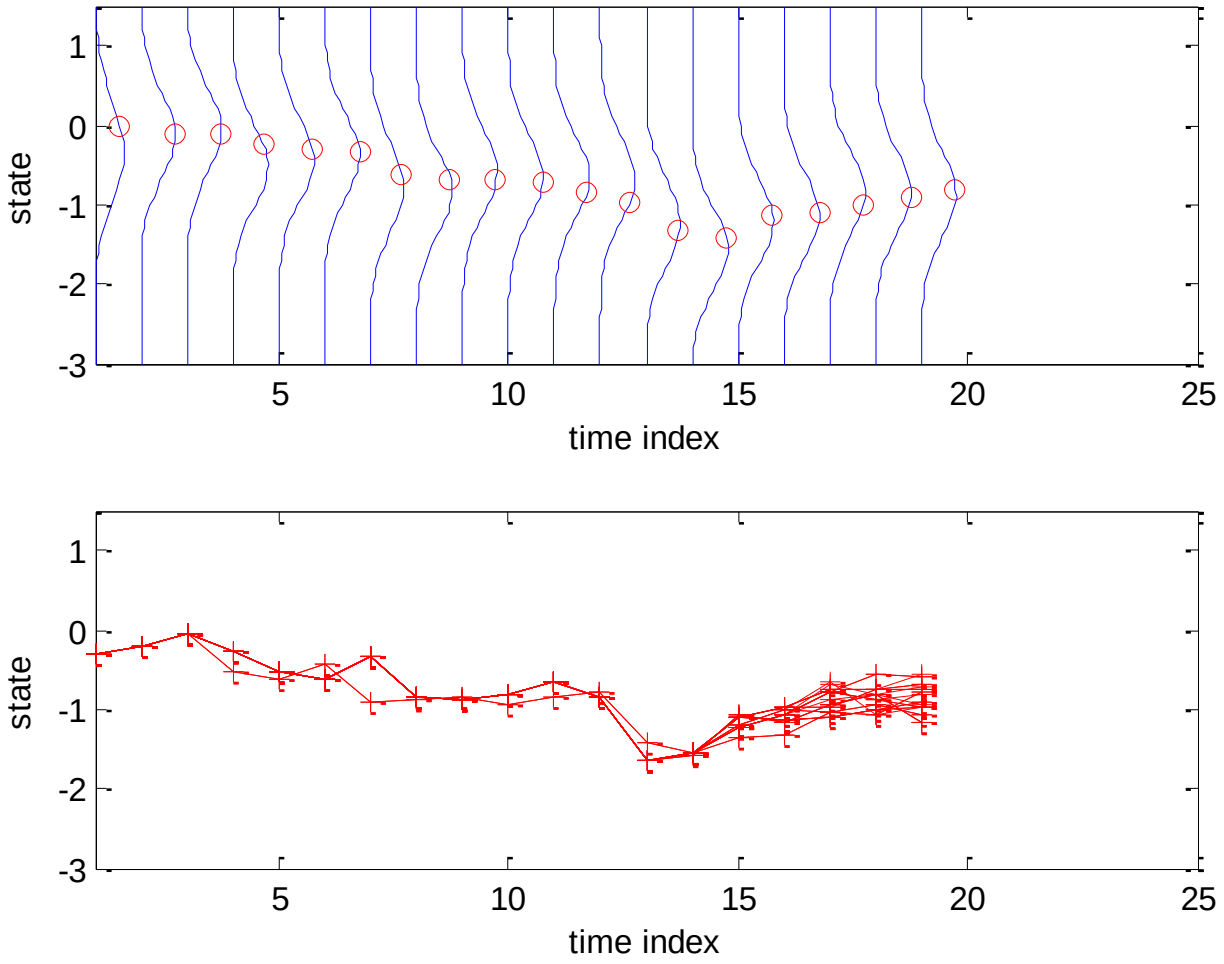
Linear Gaussian Example



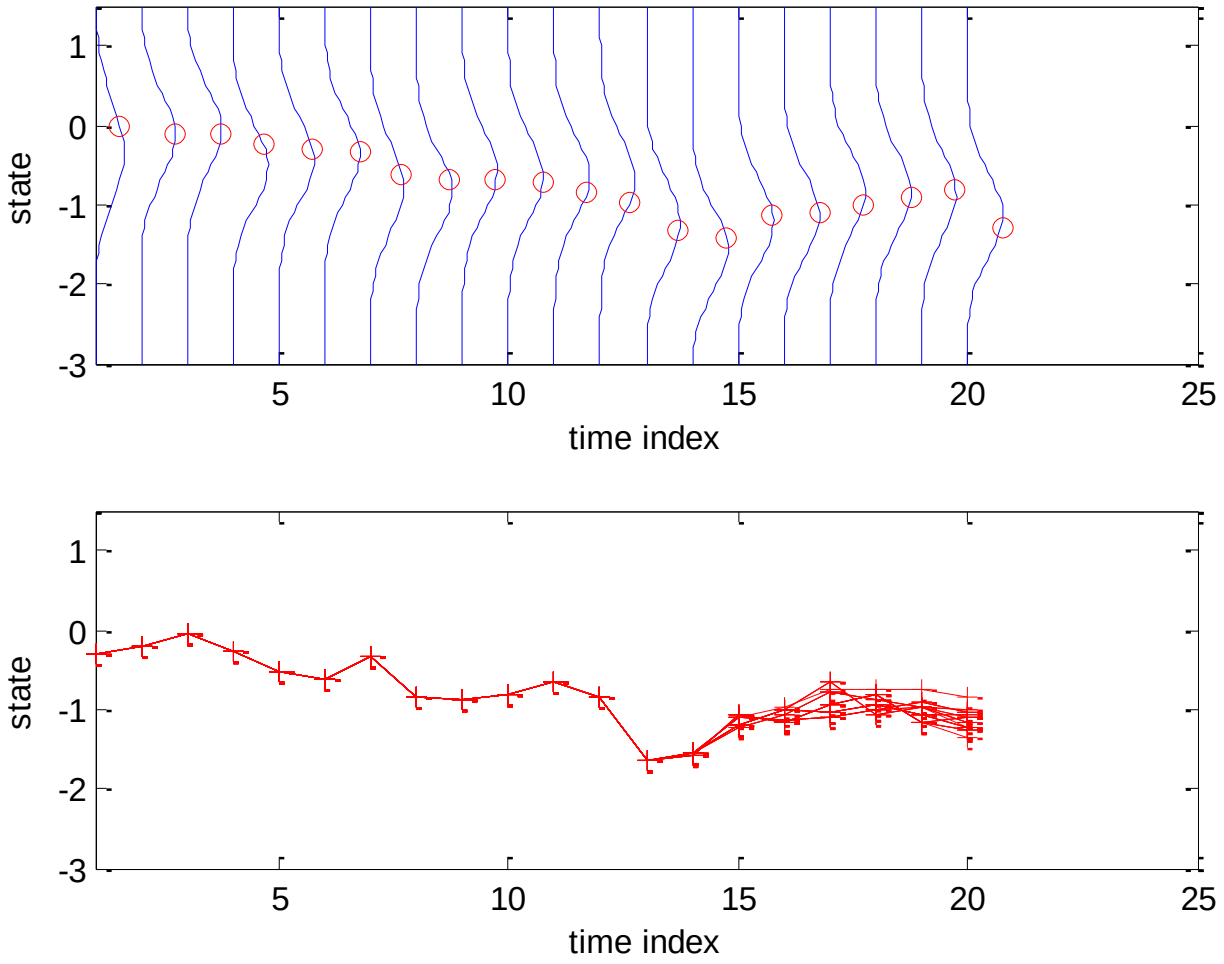
Linear Gaussian Example



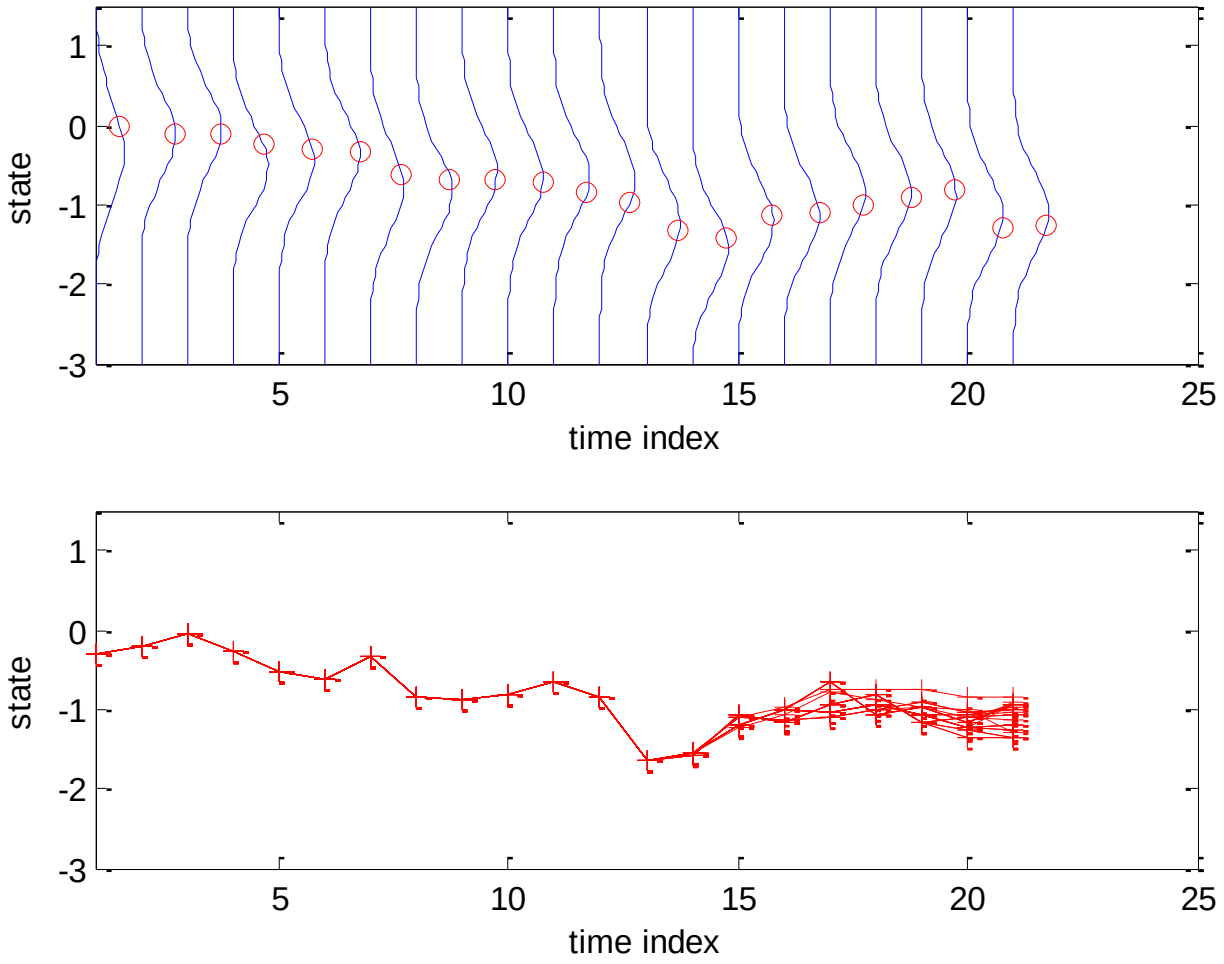
Linear Gaussian Example



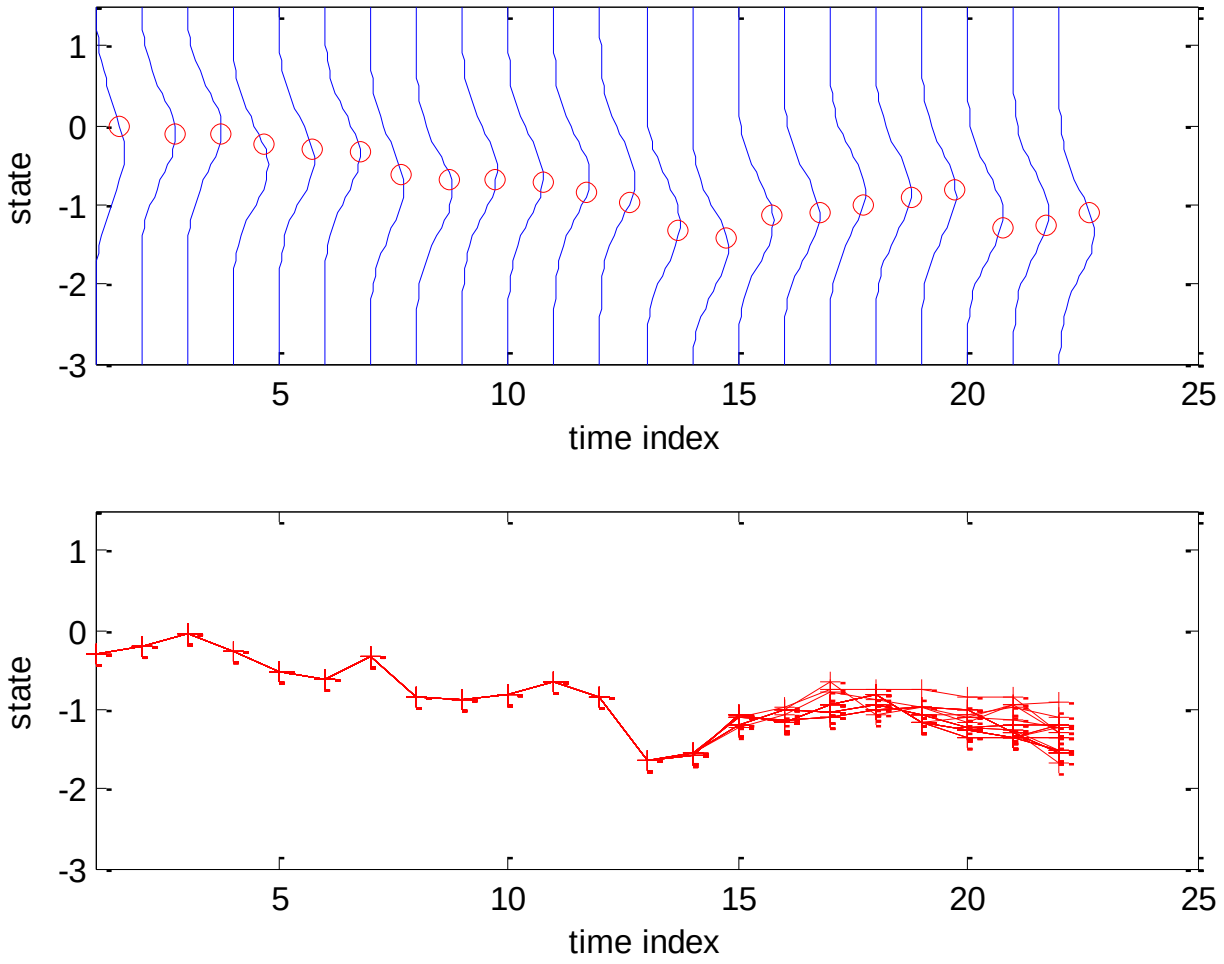
Linear Gaussian Example



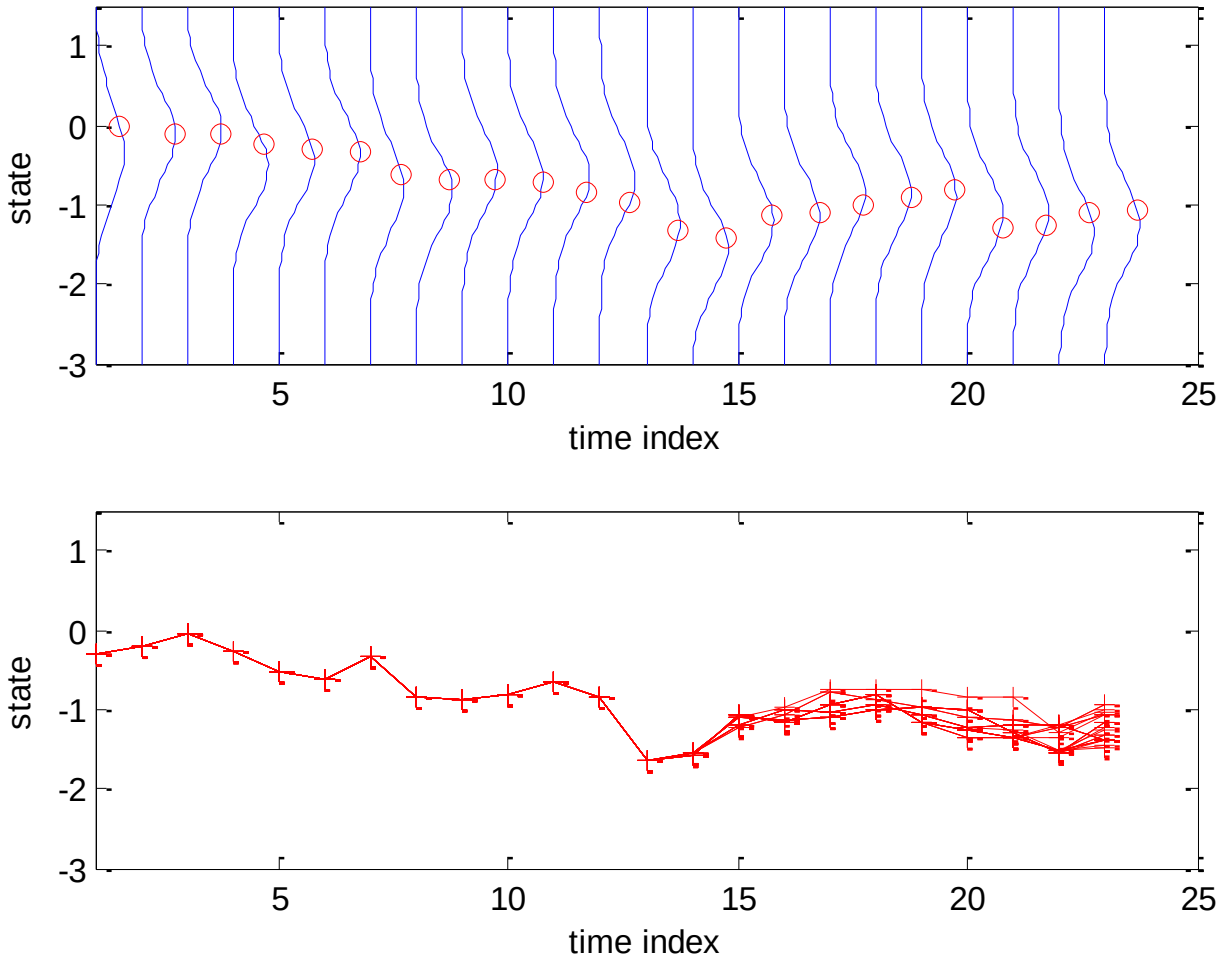
Linear Gaussian Example



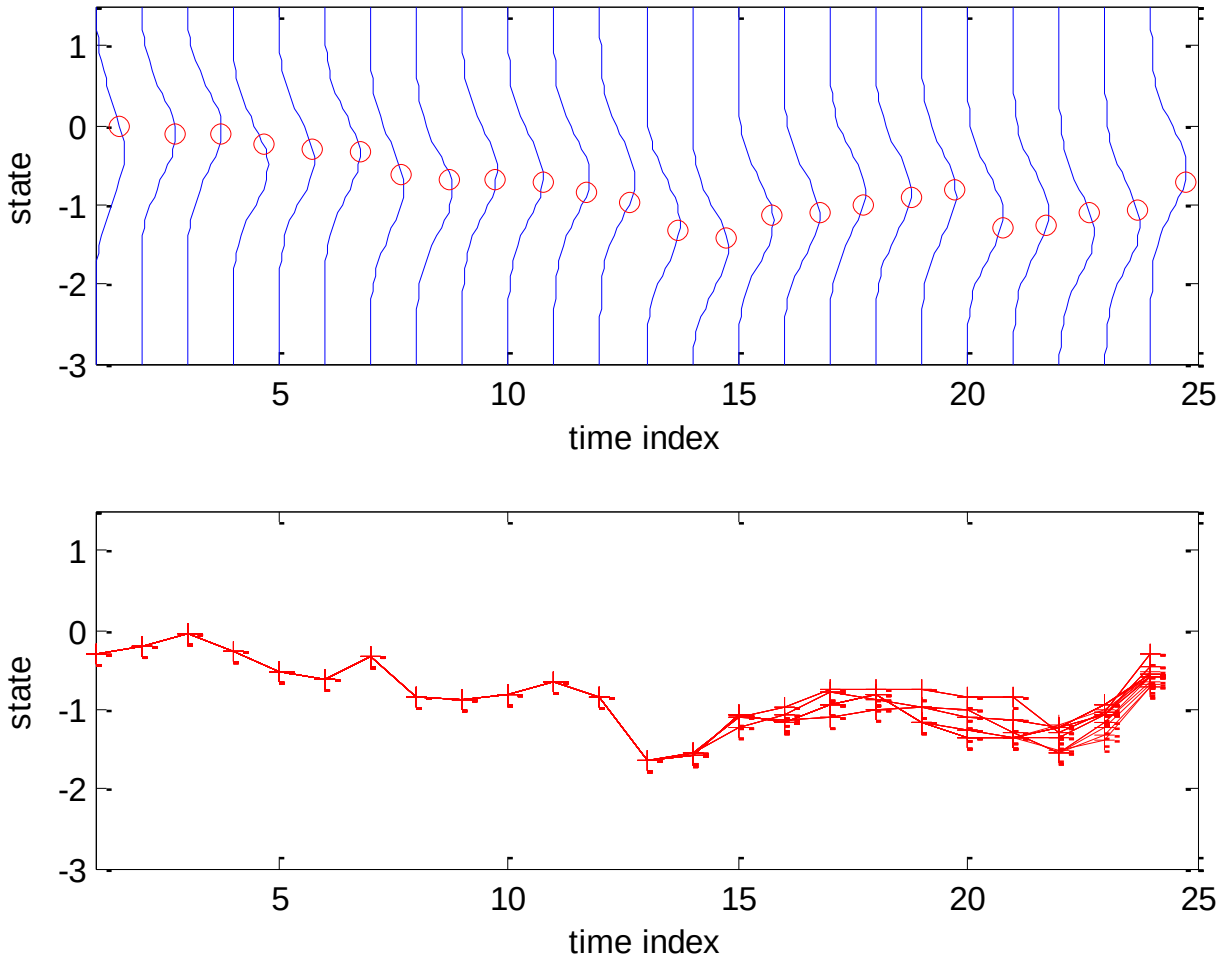
Linear Gaussian Example



Linear Gaussian Example



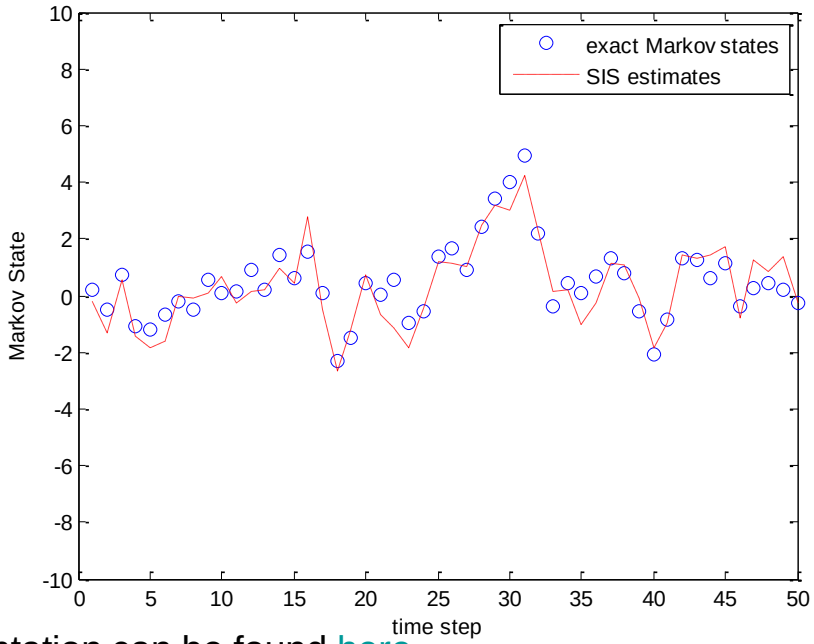
Linear Gaussian Example



Linear Gaussian Model with Locally Optimal Importance Distrib.

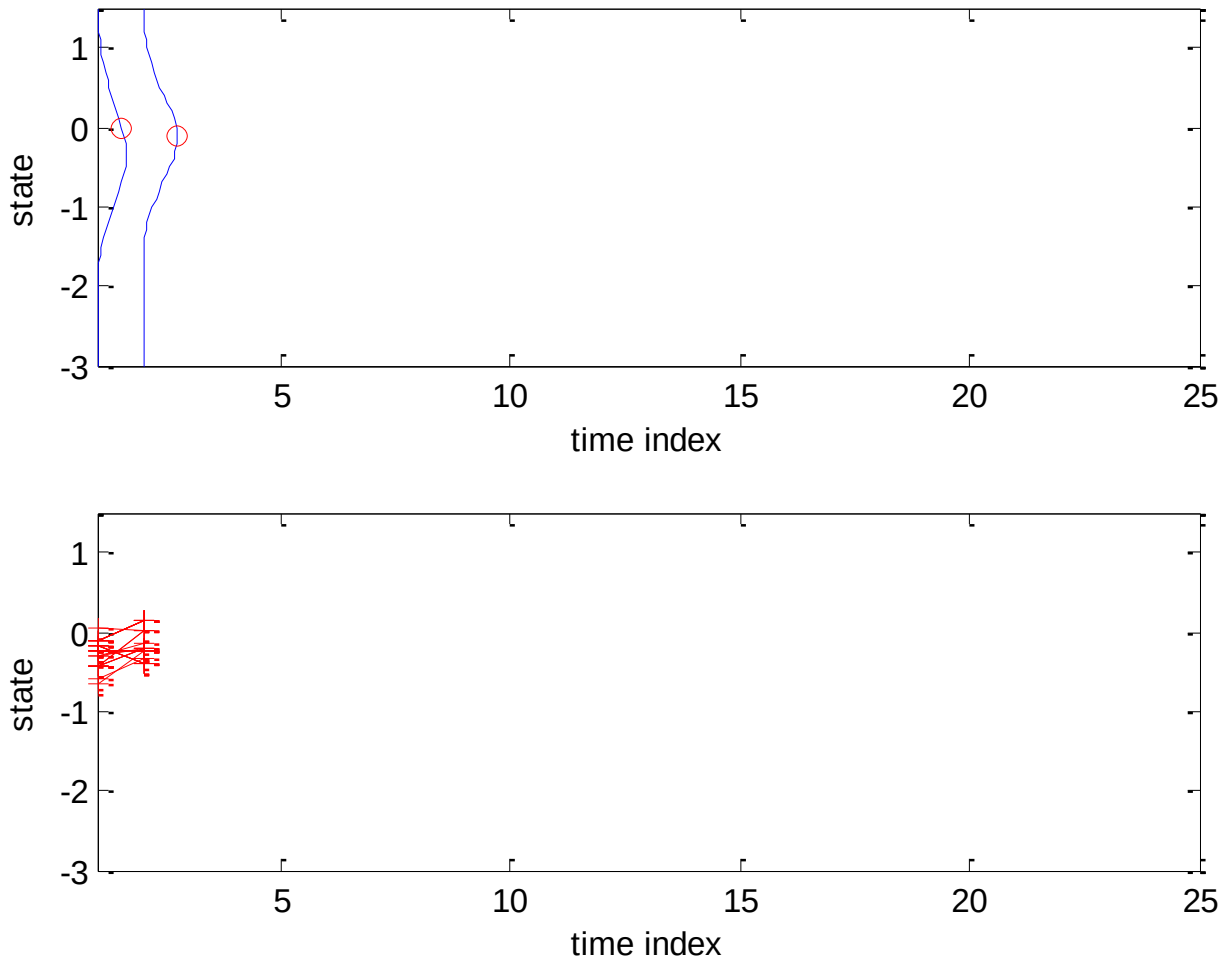
- It is easy to take samples from such Gaussian proposal distribution and compute the importance weight.

- Using the same SIS/Resampling algorithm (except different expressions for the proposal distribution and importance weight), the estimated random states (posterior mean) are plotted below:

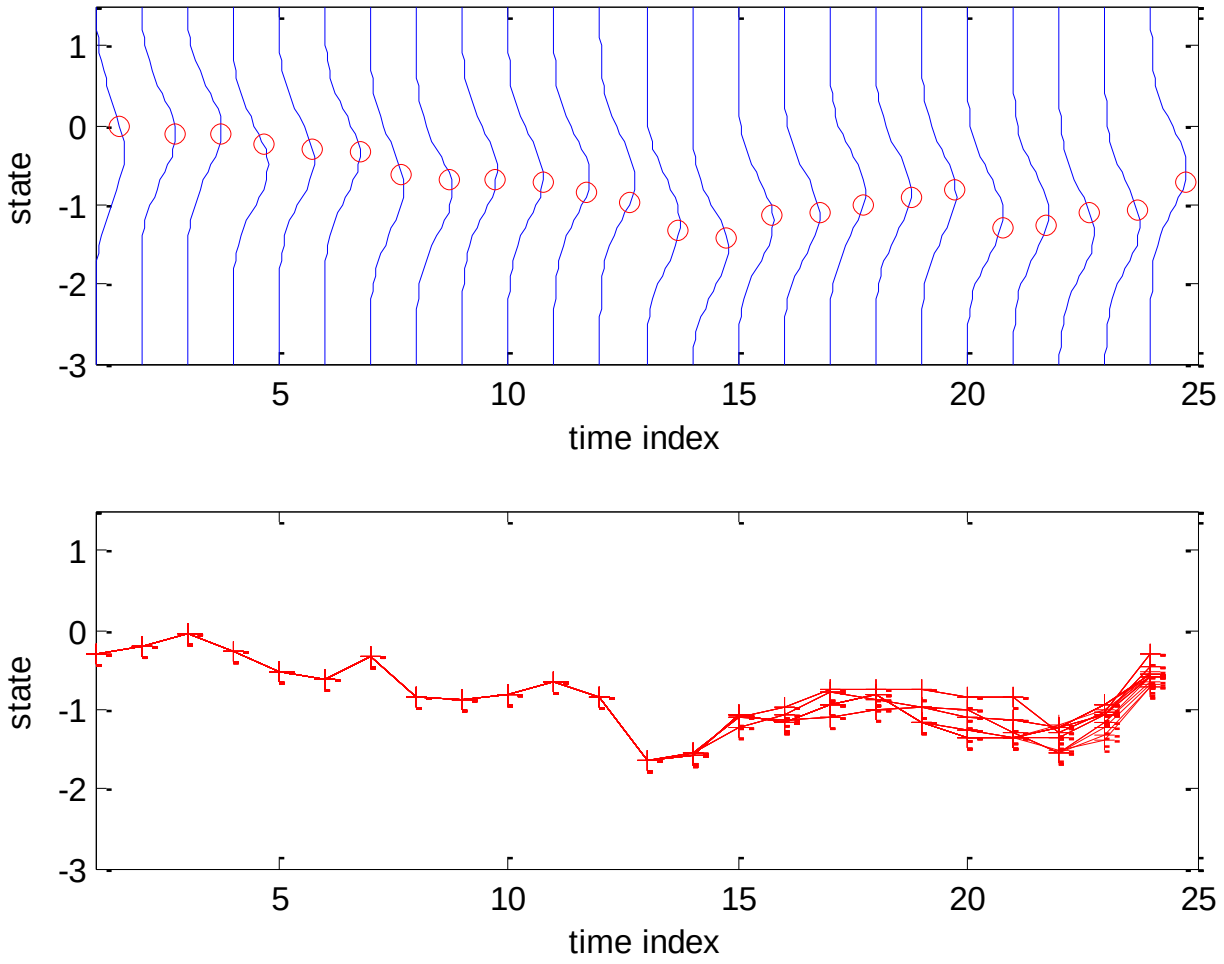


A MatLab implementation can be found here

Step 2


1
0
-1
-2
-3
state
5
10
15
20
25
time index
1
0
-1
-2
-3
state
5
10
15
20
25
time index

Step 24



Linear Gaussian Dynamic Model

- In the resampling procedure, e.g. from $\left\{\boldsymbol{X}_{1:24}^{(i)}, W_{24}^{(i)}\right\}$ to $\left\{\widetilde{\boldsymbol{X}}_{1:24}^{(i)}, 1/N\right\}$, particles with large weights are probable to be selected, and those with small weights could be eliminated.

- As a result, some particles may coalesce. In this example, at time step 24, all the 25 samples $\left\{\widetilde{\boldsymbol{X}}_{1:25}^{(i)}\right\}$ coalesce within steps 1-13. In other words, $\widetilde{\boldsymbol{X}}_{1:13}^{(i)}$ are identical among different samples.

- Obviously, the "diversity" of the samples is reduced due to resampling. To relieve this problem, we could

  - reduce the frequency of resampling by setting a proper criterion, i.e. make resampling only when necessary

  - increase the number of particles

Stochastic Volatility Example with Resampling

- Let us return to the Stochastic Volatility Model with the data defined earlier in this lecture

- Example : Stochastic Volatility

$$x_0 \sim \mathcal{N}\left(0, \sigma^2\right)$$

$$x_n = v + \phi x_{n-1} + \sigma v_n, \;\; v_n \sim \text{i.i.d.}\, \mathcal{N}\left(0,1\right)$$

$$y_n = \exp\left(\frac{x_n}{2}\right) w_n, \;\; w_n \sim \text{i.i.d.}\, \mathcal{N}\left(0,1\right)$$

where x_n denotes the unobserved Markov states, and y_n denotes the observations.

- We still use the same proposal distribution for this model as before (sub-optimal), except adding the resampling procedure.

Bayesian Monte Carlo Filtering for Stochastic Volatility Models, R. Casarin .

SIS Algorithm with Resampling

- The procedure of the SIS algorithm with re-sampling is as follows:
  - generate *N* sequence of samples denoted by upper index (i)
  - sample from $\mathcal{N}\left(0,\sigma^2\right)$ to get the initial states $x_0^{(i)}$
  - at time step *n=1,2,....*
    - sample $q_n\left(x_n^{(i)} \mid \boldsymbol{x}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n}\right) = f\left(x_n^{(i)} \mid x_{n-1}^{(i)}\right) \sim \mathcal{N}\left(v+\phi x_{n-1}^{(i)}, \sigma^2\right)$ and set $\boldsymbol{X}_{1:n}^{(i)} = \left(\boldsymbol{X}_{1:n-1}^{(i)}, X_n^{(i)}\right)$
    - evaluate the importance weights up to a normalizing constant

$$w_n = w_{n-1} \times g\left(y_n \mid x_n\right), \quad \text{where } f\left(y_n \mid x_n\right) = \mathcal{N}\left(0, \exp\left(x_n\right)\right)$$

    - normalize the importance weights

$$W_n^{(i)} = w_n^{(i)} \Big/ \sum_{i=1}^{N} w_n^{(i)}$$

  - If *ESS* < *C*, we start resampling according to the normalized importance weights. Here *C* denotes the minimum set effective sample size.

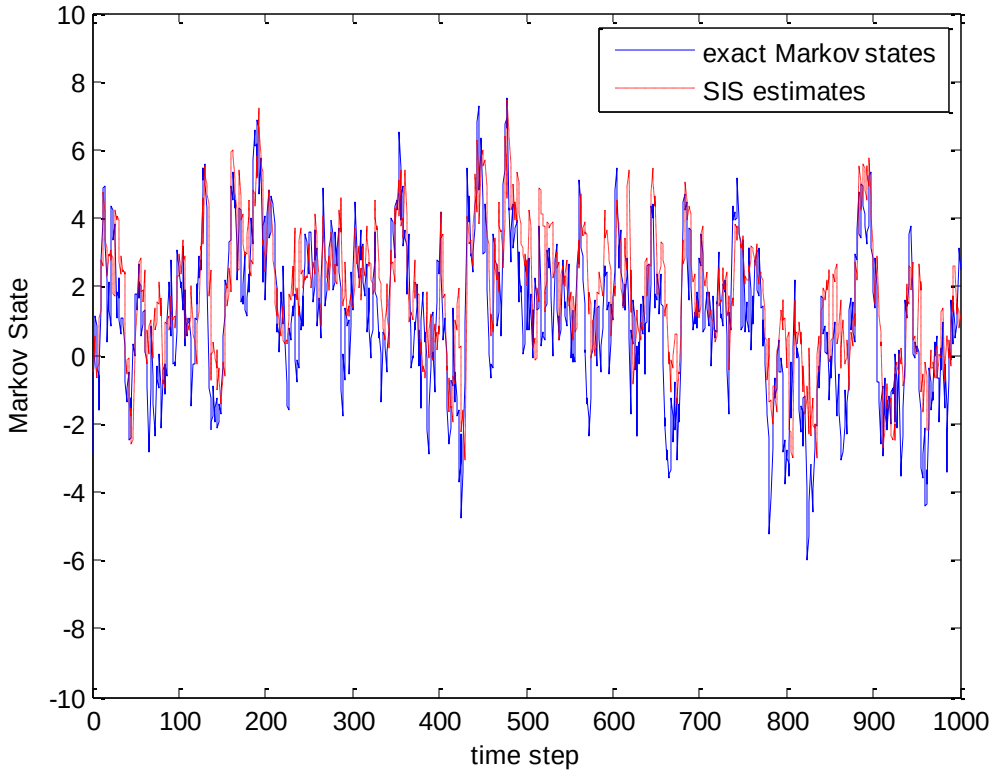
Stochastic Volatility Example with Resampling

- In this example, we use 500 samples.
- We want to guarantee that the number of samples with importance weights close to ZERO is no more than 70% of the total samples.
- This means the ESS should be larger than 30% * total sample size. Then we set in our code that

```
if ESS < 30% * total sample size
        resample;
end
```

Stochastic Volatility: Estimation with resamping

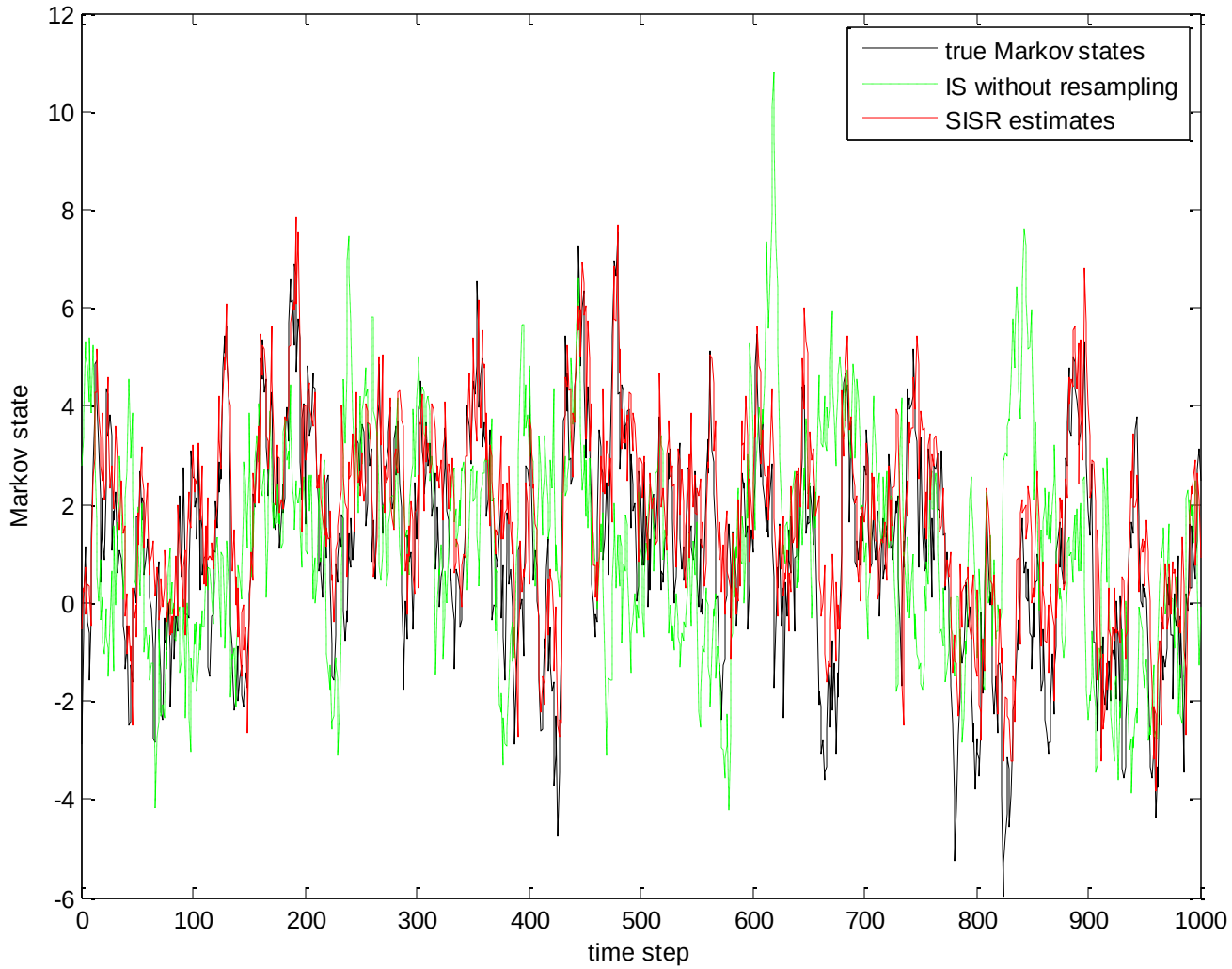
- Blue line represents the true Markov states
- Red line represents estimation with resampling
- The resampling criterion is : ESS < 30% * total sample size



A C++ implementation can be found here

Stochastic Volatility Example

- **Comparison**: resampling (red line) vs. no resampling (green line)

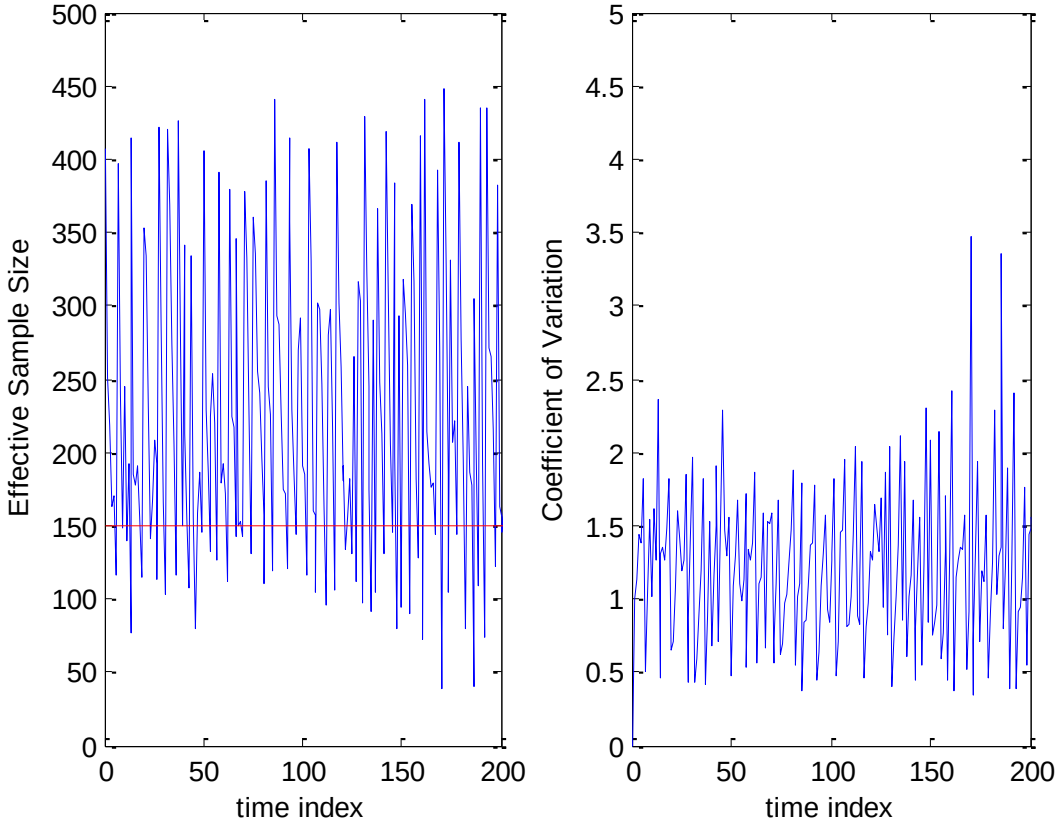


Stochastic Volatility

- From the last plot, we can see that the estimated Markov states are very close to the true values using SIS with resampling; while without resampling, there is great discrepancy between estimated states and the true states.

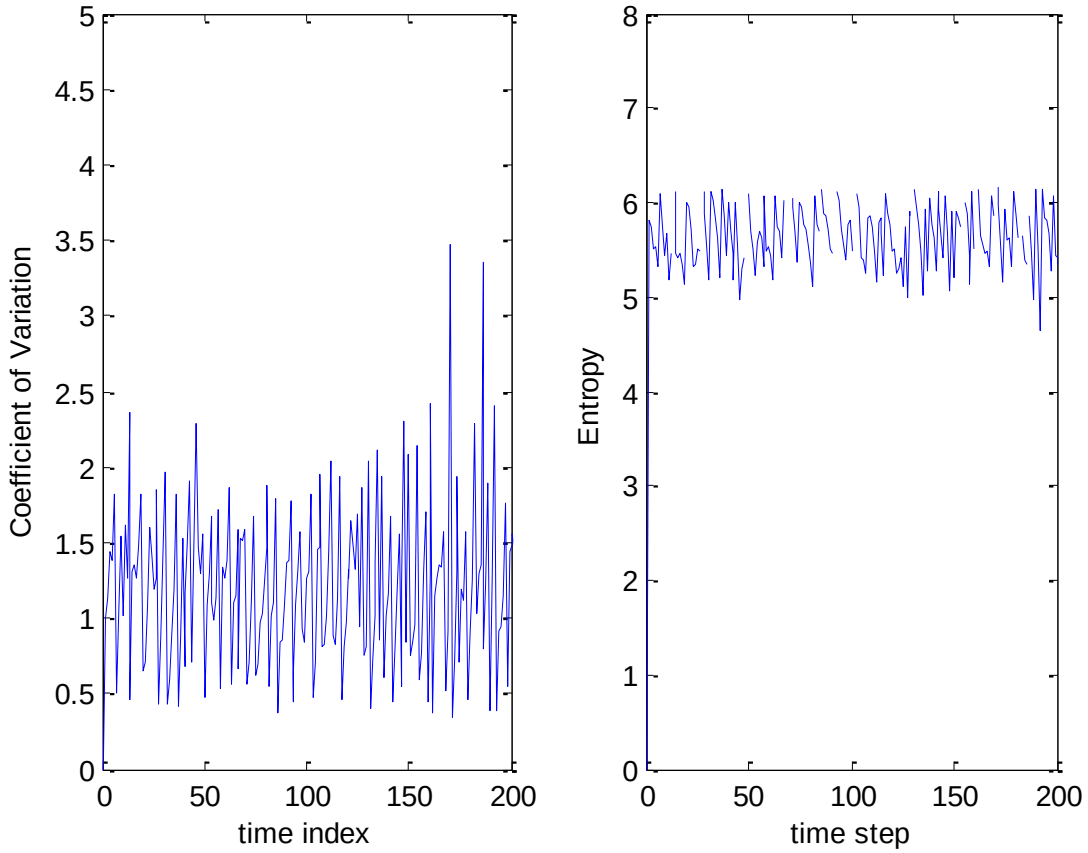
Stochastic Volatility Example

- Effective sample size and coefficient of variance (with resampling)
  - the red line in the left subplot denotes the critical ESS below which resampling will be taken



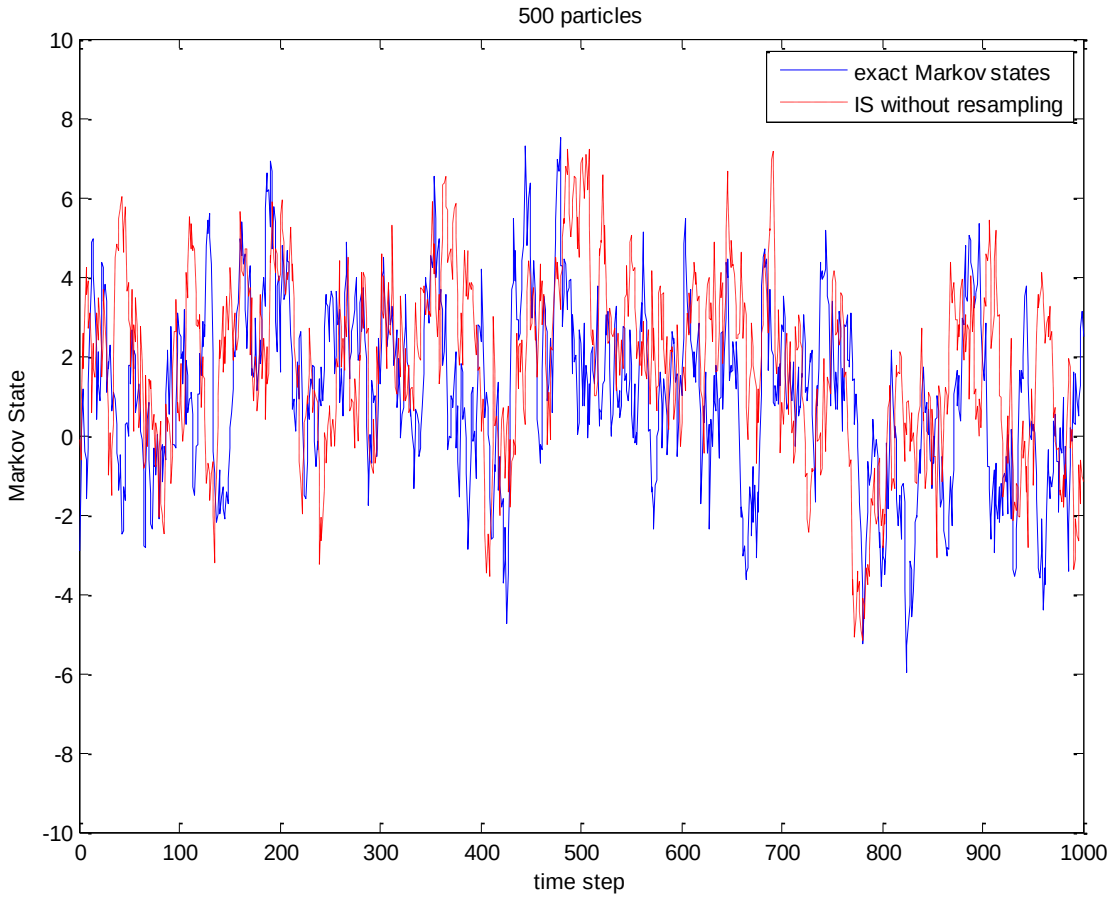
Stochastic Volatility Example

- Coefficient of variation and Entropy (with resampling)



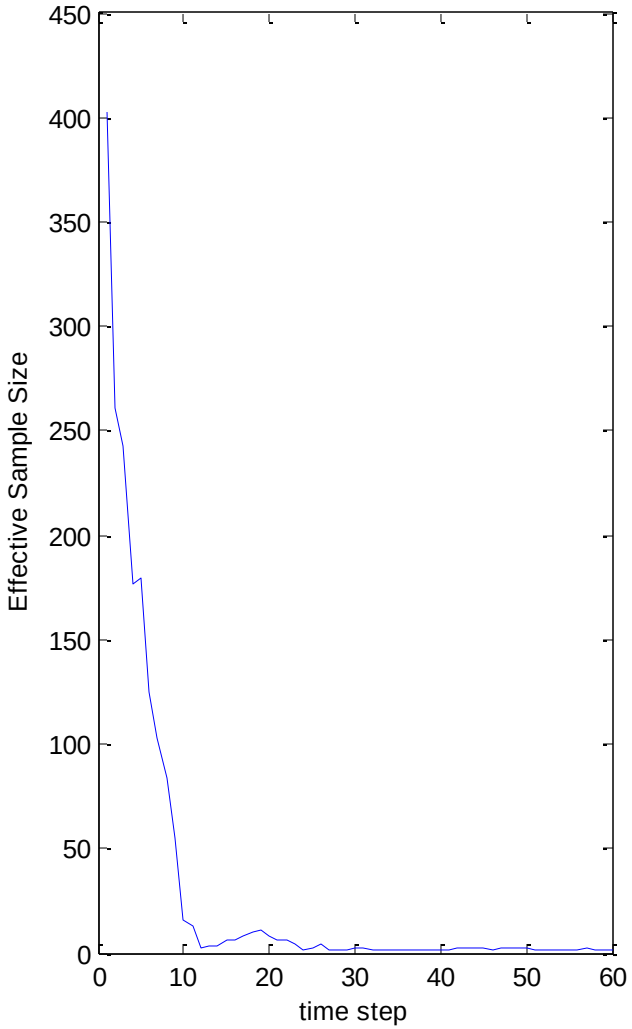
Stochastic Volatility Example

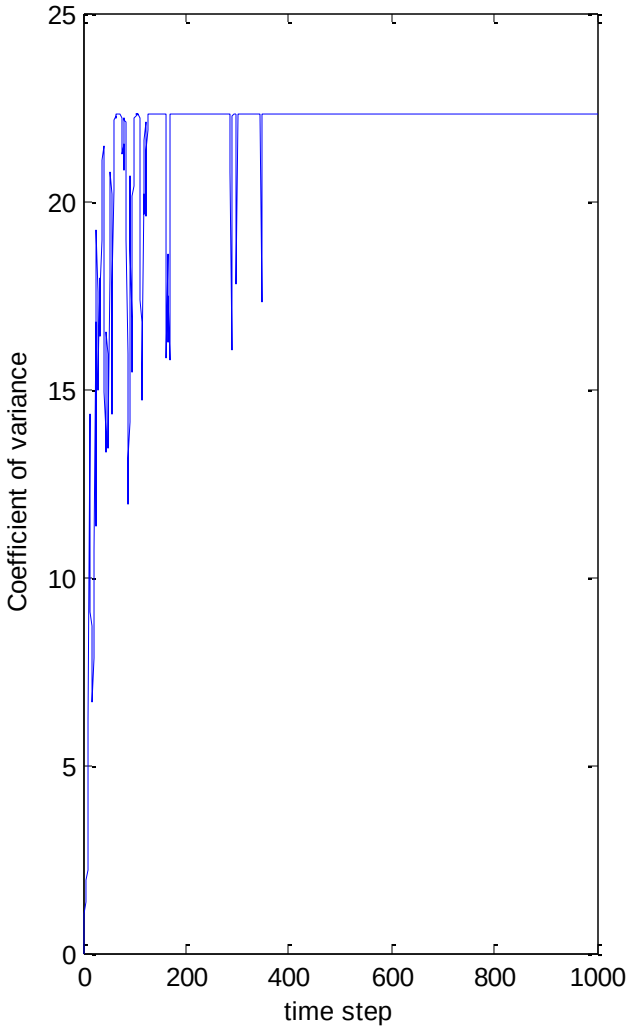
- Estimation without resampling
  - blue line represents the true (unobserved) Markov states
  - red line represents estimations (posterior mean) without resampling



Stochastic Volatility Example

- Effective sample size and coefficient of variation (without resampling)



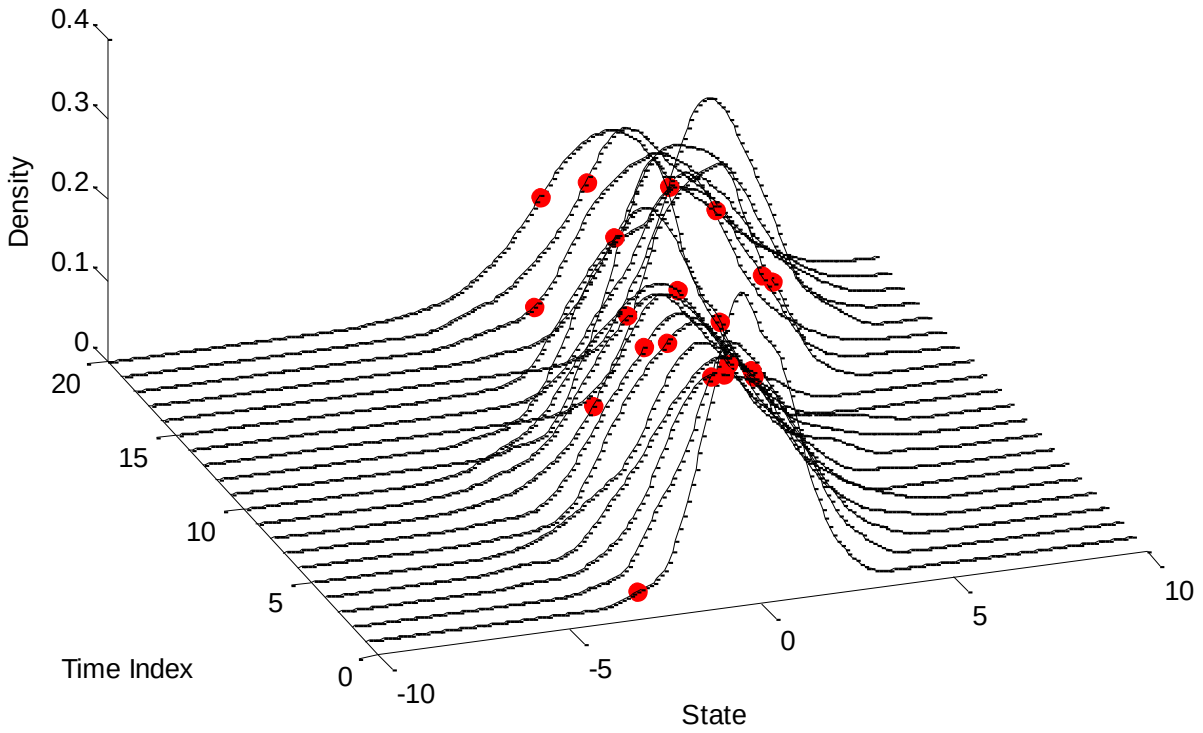


Stochastic Volatility Example

- From the histogram of effective sample size, we can see that by resampling procedure, we ensure that the importance weight of the samples is reasonably high.
- We can compare the estimated result by SIS without resampling.
- Obviously, the effective sample size decreases rapidly with time steps. There are only a few samples with significant importance weights left.

Stochastic Volatility Example: Marginal $p(x_n \mid \boldsymbol{y}_{1:n})$

- The SMC (with resampling) estimates of the marginal distributions $p(x_n \mid \boldsymbol{y}_{1:n})$ and true values of $\{X_n\}$ is

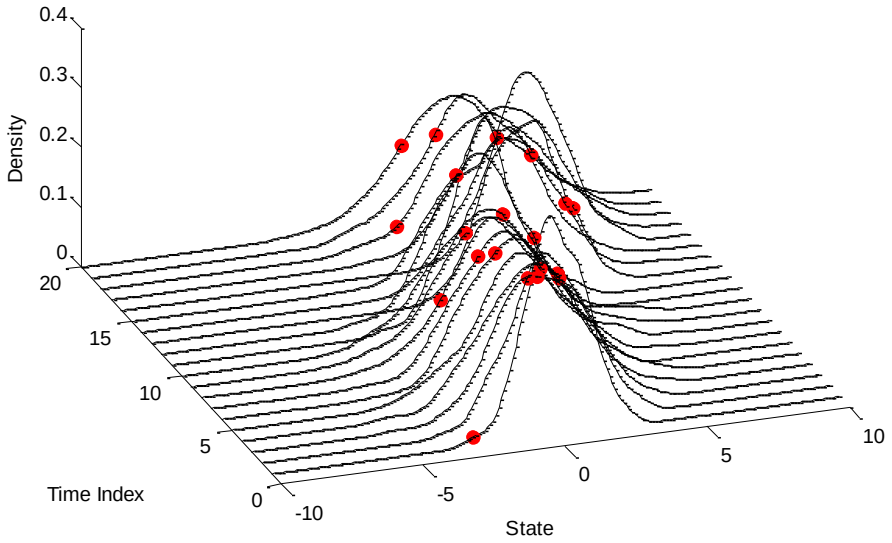


red dots --- true Markov states X_n at each time step

Stochastic Volatility Example: Marginal $p(x_n \mid \mathbf{y}_{1:n})$

- Generally, we don't have the analytic expressions for $p(x_n \mid \mathbf{y}_{1:n})$ except for some special cases like linear Gaussian model in which this distribution is subject to Gaussian distribution.

- The curve for $p(x_n \mid \mathbf{y}_{1:n})$ is estimated from samples, i.e. at time step *n*, using the generated samples $\{x_n^{(i)}\}$ to estimate the probability density (Matlab and other statistics libraries provide commands for density estimation). The curves at different *n*'s are put together as shown below.



Another Stochastic Volatility Example

- Example : Stochastic Volatility

$$x_0 \sim \mathcal{N}(0, \sigma^2)$$

$$x_n = \phi x_{n-1} + \sigma v_n \quad , \quad v_n \sim \mathcal{N}(0,1)$$

$$y_n = \beta \exp\left(\frac{x_n}{2}\right) w_n \, , \quad w_n \sim \mathcal{N}(0,1)$$

The parameters here are different from the earlier examined problem. They are set to be

$$\sigma = 0.14, \;\; \phi = 0.98, \;\; \beta = 0.66$$

- In this example, we will use an approximation of the locally optimal proposal distribution.

Stochastic Volatility Example

- We cannot sample from the locally optimal importance distribution $p(x_n \mid x_{n-1}, y_n)$, but it is unimodal and we can approximate the optimal kernel by a simpler importance distribution.

- In this model, the state and observation equations are

$$f(x_n \mid x_{n-1}) = \mathcal{N}(\phi x_{n-1}, \sigma^2)$$
$$g(y_n \mid x_n) = \mathcal{N}(0, \beta^2 \exp(x_n))$$

  Then we get

$$p(x_n \mid x_{n-1}, y_n) = \frac{f(x_n \mid x_{n-1}) g(y_n \mid x_n)}{\int f(x_n \mid x_{n-1}) g(y_n \mid x_n) dx_n}$$

- The function $\log(f(x_n \mid x_{n-1}) g(y_n \mid x_n))$ is indeed strictly concave. Given x_{n-1}, the mode of $p(x_n \mid x_{n-1}, y_n)$ is

$$m_n(x_{n-1}) = -\arg\min_{x_n} \{\log(f(x_n \mid x_{n-1}) g(y_n \mid x_n))\}$$

Stochastic Volatility Example

$$f\left(x_n \mid x_{n-1}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(x_n - \phi x_{n-1}\right)^2}{2\sigma^2} \right]$$

$$g\left(y_n \mid x_n\right) = \frac{1}{\sqrt{2\pi\beta^2 \exp\left(x_n\right)}} \exp\left[-\frac{y_n^2}{2\beta^2 \exp\left(x_n\right)} \right]$$

$$= \frac{1}{\sqrt{2\pi\beta^2}} \exp\left[-\frac{y_n^2}{2\beta^2} \exp\left(-x_n\right) - \frac{1}{2} x_n \right]$$

- Then

$$\log\left(f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) \right) = \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{\sqrt{2\pi\beta^2}} \right) + \left[-\frac{\left(x_n - \phi x_{n-1}\right)^2}{2\sigma^2} - \frac{y_n^2}{2\beta^2} \exp\left(-x_n\right) - \frac{1}{2} x_n \right]$$

and the first order derivative with respect to x_n is:

$$\frac{d\left(\log\left(f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right) \right)\right)}{dx_n} = -\frac{\left(x_n - \phi x_{n-1}\right)}{\sigma^2} + \frac{y_n^2}{2\beta^2} \exp\left(-x_n\right) - \frac{1}{2}$$

Stochastic Volatility Example

- Thus the mode $m_n(x_{n-1})$ is the unique solution of the non-linear equation

$$-\frac{1}{\sigma^2}\left(x_n-\phi x_{n-1}\right)+\frac{y_n^2}{2\beta^2}\exp\left(-x_n\right)-\frac{1}{2}=0$$

- If we assume that $p(x_n \mid x_{n-1}, y_n)$ can be approximated by a Gaussian distribution, the variance $\sigma_n^2(x_{n-1})$ is approximately minus the inverse of the second-order derivative of $\log\left(f(x_n \mid x_{n-1})g(y_n \mid x_n)\right)$ evaluated at the mode, i.e.

$$\sigma_n^{-2}\left(x_{n-1}\right)=\frac{1}{\sigma^2}+\frac{y_n^2}{2\beta^2}\exp\left[-m_n\left(x_{n-1}\right)\right]$$

This is justified since it is the way we get the squared variance of a Gaussian distribution.

Stochastic Volatility Example

- In fact, there are two ways of selecting the importance distribution
  - **Gaussian :**
    fit the mean of the Gaussian distribution to the mode of $p(x_n \mid x_{n-1}, y_n)$ and fit the variance to minus the inverse of the second order derivative of $\log\left(p(x_n \mid x_{n-1}, y_n)\right)$ as shown in the earlier slide
  - **Student-t :**
    fit the location and scale parameters as the mean and covariance parameters in the normal case.
- The *pdf* of the t-distribution p(x) is

$$p(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left[\frac{\nu + \left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right]^{-\frac{\nu+1}{2}}$$

where μ is the location parameter, σ is the scale parameter and ν is the degree of freedom.

Stochastic Volatility Example

- In this example, we choose the t-distribution with 5-degrees of freedom. The importance weight is obtained by

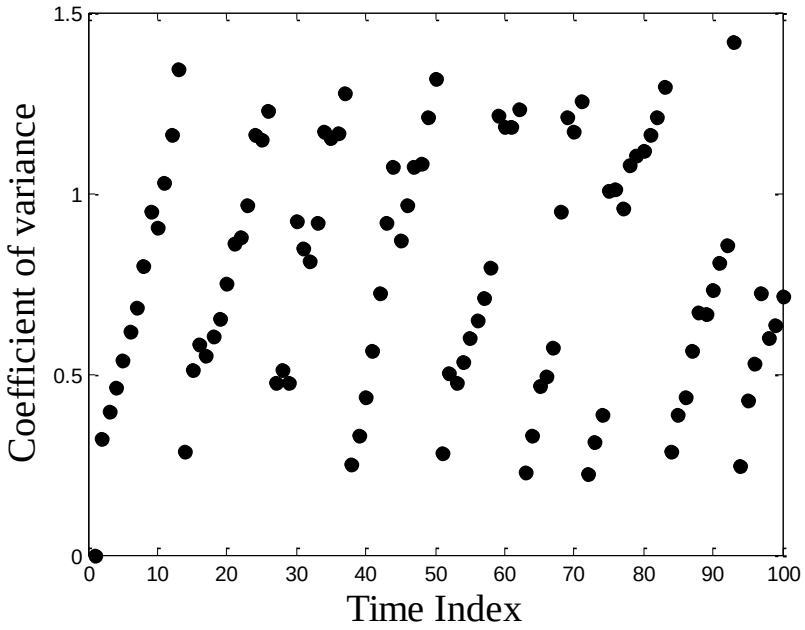
$$W_n \propto W_{n-1} \frac{f\left(x_n \mid x_{n-1}\right) g\left(y_n \mid x_n\right)}{\left\{\eta + \left[\frac{x_n - m_n\left(x_{n-1}\right)}{\sigma_n\left(x_{n-1}\right)}\right]^2\right\}^{-(\eta+1)/2}}$$

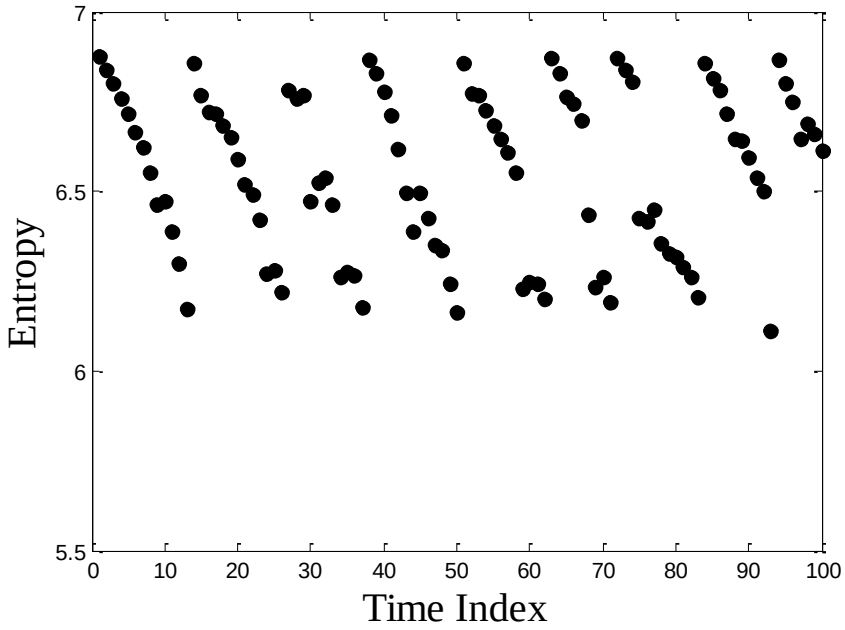
- In the implementation of the SMC algorithm, *N=1000* particles are used and resampling takes place when

$$ESS < 0.4N$$

Resampling

- When **Resampling** is performed



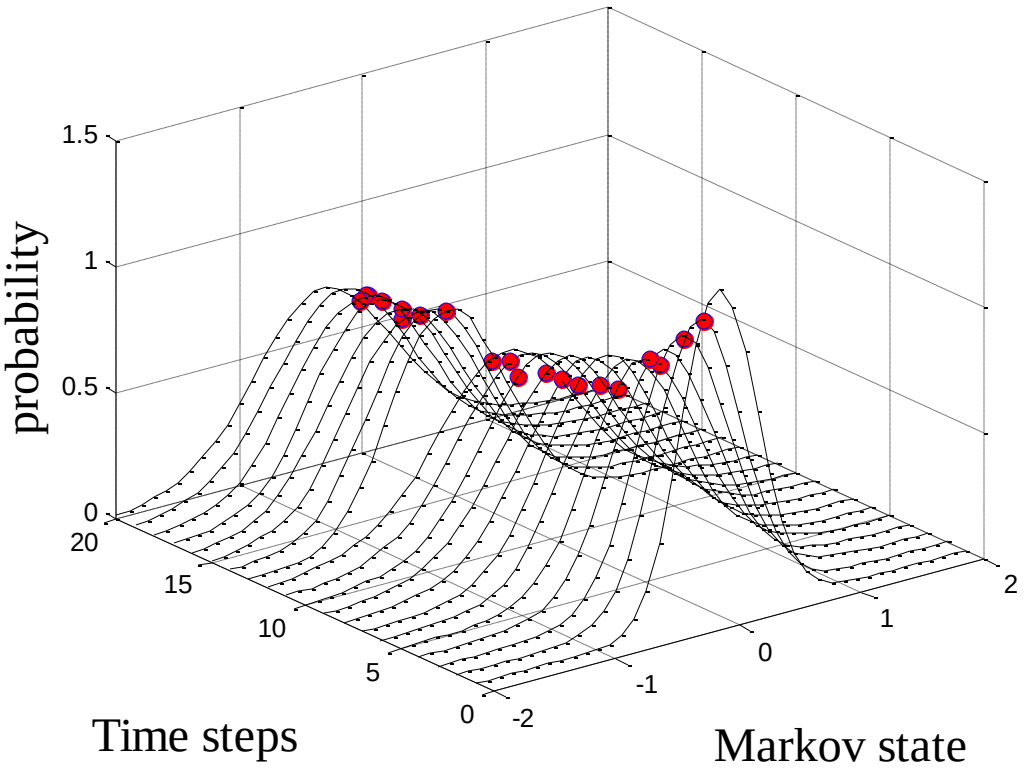


- In this example, it is easier to see that the COV/Entropy decrease/increase every time we resample with almost linear changes in between.

The data and a MatLab implementation can be found here

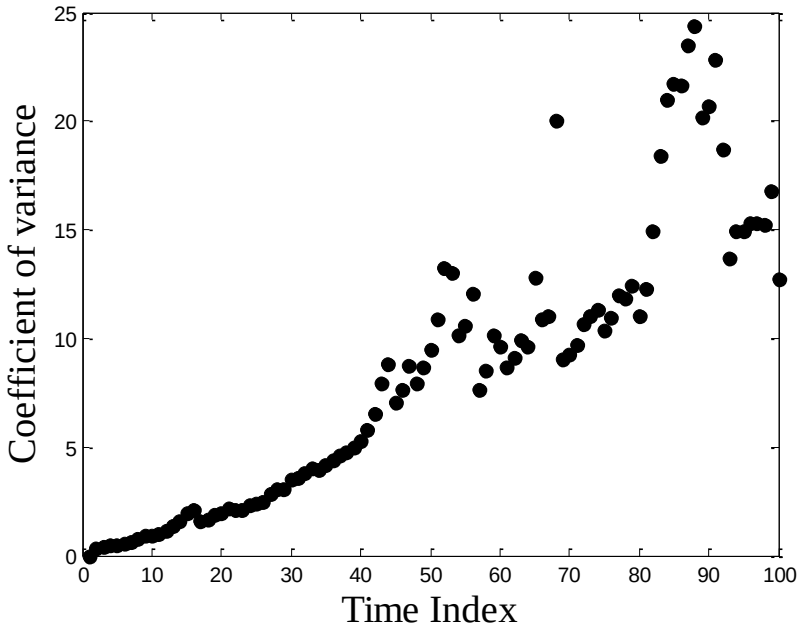
Resampling

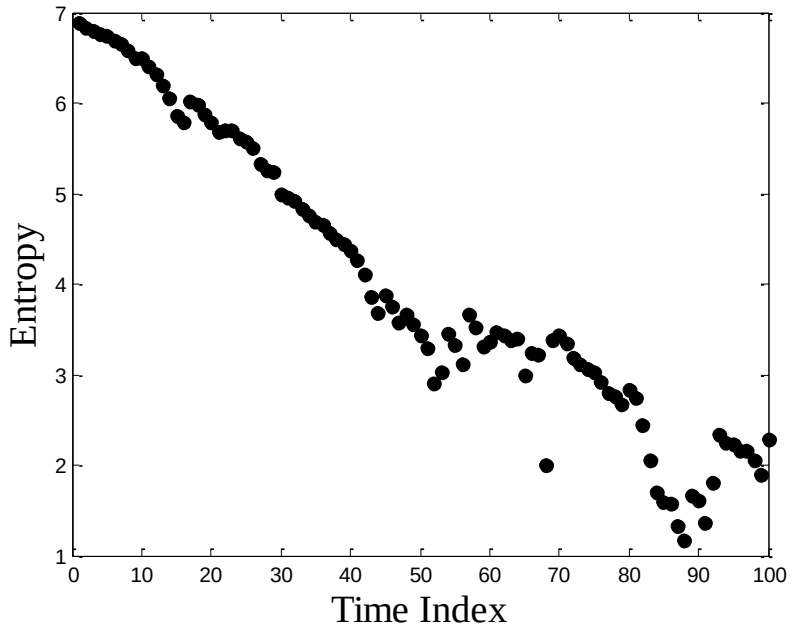
- Monte Carlo estimates of the marginal distribution $p(x_n \mid y_{1:n})$ and true values of X_n (red spots)



Resampling

- **NO Resampling** is used here.





CONVERGENCE, ERROR ESTIMATES and DEGENERACY: SIS Vs. SISR

Convergence of Sequential Monte Carlo

- The following error estimate can be shown:

$$\mathbb{E}\left[\left|\int \varphi(\boldsymbol{x}_{1:n})\left(\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) - p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})\right)d\boldsymbol{x}_{1:n}\right|^p\right]^{1/p} \leq \frac{C_n}{N}\|\varphi\|_\infty, \ \forall\, p > 1$$

- This is not very useful as the constant C_n increases (polymomially/ exponentially) with time n.
- To achieve comparable accuracy at all n, one needs to increase the sample size N – something that of course is not practical.
- We cannot expect to approximate the target distribution with the same accuracy with increasing n. We want not to accumulate errors with time. The best we can hope is for some L

$$\mathbb{E}\left[\left|\int \varphi(\boldsymbol{x}_{n-L+1:n})\left(\hat{p}_N(\boldsymbol{x}_{n-L+1:n}|\boldsymbol{y}_{1:n}) - p(\boldsymbol{x}_{n-L+1:n}|\boldsymbol{y}_{1:n})\right)d\boldsymbol{x}_{n-L+1:n}\right|^p\right]^{1/p} \leq \frac{C_L}{N}\|\varphi\|_\infty$$

for a model that has good mixing/forgetting properties, e.g.

$$\left\| p(x_n \mid \boldsymbol{y}_{2:n}, x_1) - p(x_n \mid \boldsymbol{y}_{2:n}, x_1^{'}) \right\| \leq D\lambda^{n-1}, \ \textit{for } \lambda < 1$$

P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004

Convergence of Sequential Monte Carlo

- The last error estimates indicate that we can approximate well $p(\boldsymbol{x}_{n-L+1:n} \mid \boldsymbol{y}_{1:n})$ but not $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$
- This is quite useful as in most applications we are interested in the filtering distribution $p(x_n \mid \boldsymbol{y}_{1:n})$
- For a system that has ergodic properties, we also use this to estimate smoothing distributions

$$p(x_k|\boldsymbol{y}_{1:n}) \simeq p(x_k|\boldsymbol{y}_{1:k+L})$$

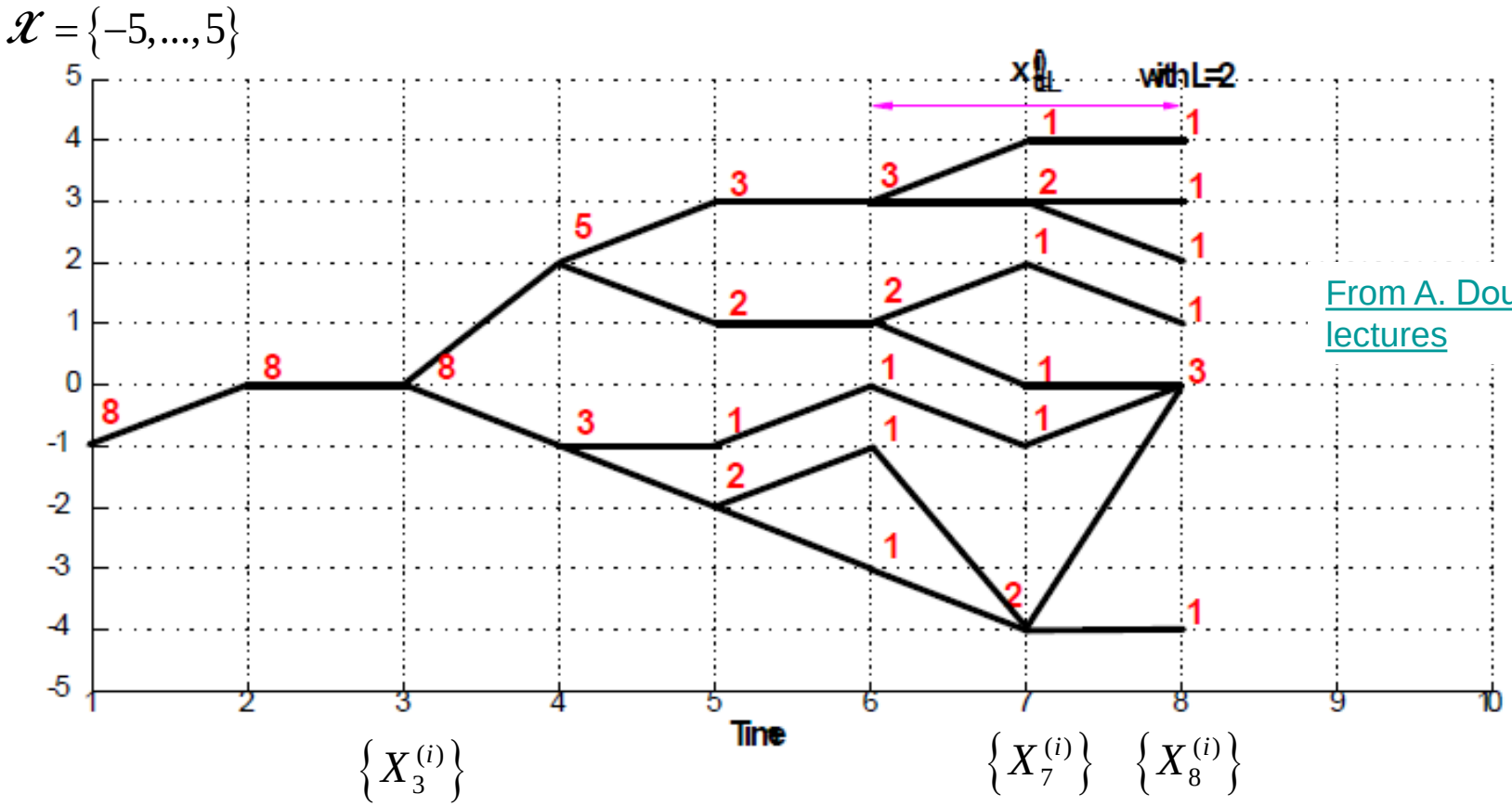
- Finally, as discussed earlier in this lecture, we can estimate the marginal distribution as follows:

$$p(\boldsymbol{y}_{1:n}) = p(y_1)\prod_{k=2}^{n} p\left(y_k \mid \boldsymbol{y}_{1:k-1}\right)$$

$$p\left(y_k \mid \boldsymbol{y}_{1:k-1}\right) = \int g\left(y_k \mid x_k\right) p\left(x_k \mid \boldsymbol{y}_{1:k-1}\right) dx_k$$

Convergence of Sequential Monte Carlo

- The paths coalesce as we go backwards on time. The SMC approximation deteriorates as we go back in time. Thus $p(x_8 \mid \boldsymbol{y}_{1:8}), p(x_7 \mid \boldsymbol{y}_{1:8})$ are well approximated, but not e.g. $p(x_i \mid \boldsymbol{y}_{1:8}), i \le 4.$



From A. Doucet's lectures

- Variables X_7, X_8 have a good coverage but not e.g. $X_i, i \le 4.$

Central Limit Theorem for SIS

- We have seen earlier for the standard SIS

$$\sqrt{N}\left(\mathbb{E}_{\hat{p}_N(x_n|\boldsymbol{y}_{1:n})}\left(\varphi(X_n)\right) - \mathbb{E}_{p(x_n|y_{1:n})}\left(\varphi(X_n)\right)\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{IS,n}(\varphi)\right)$$

$$\sigma^2_{IS,n}(\varphi) = \int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(\varphi(x_n) - \mathbb{E}_{p(x_n|y_{1:n})}\left(\varphi(X_n)\right)\right) d\boldsymbol{x}_{1:n}$$

- One can also show:

$$\sqrt{N}\left(\log\hat{p}_N(\boldsymbol{y}_{1:n}) - \log p(\boldsymbol{y}_{1:n})\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{IS,n}\right)$$

$$\sigma^2_{IS,n} = \int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})} d\boldsymbol{x}_{1:n} - 1$$

- The variance grows exponentially fast for sequential importance sampling.

P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004

Central Limit Theorem for SISR

- For Sequential Importance Sampling with Resampling (SISR):

$$\sqrt{N}\left(\mathbb{E}_{\hat{p}_N(x_n|\boldsymbol{y}_{1:n})}\left(\varphi(X_n)\right) - \mathbb{E}_{p(x_n|\boldsymbol{y}_{1:n})}\left(\varphi(X_n)\right)\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{SMC,n}(\varphi)\right)$$

$$\sigma^2_{SMC,n}(\varphi) = \int \frac{p^2(x_1|\boldsymbol{y}_{1:n})}{q(x_1|y_1)}\left(\int \varphi(x_n)p(x_n|y_n,x_1)dx_n - \bar{\varphi}_n\right)^2 dx_1$$

$$+ \sum_{k=2}^{n-1} \int \frac{p^2(x_{k-1},x_k|\boldsymbol{y}_{1:n})}{p(x_{k-1}|\boldsymbol{y}_{1:k-1})q(x_k|y_k,x_{k-1})}\left(\int \varphi(x_n)p(x_n|y_n,x_k)dx_n - \bar{\varphi}_n\right)^2 d\boldsymbol{x}_{k-1:k}$$

$$+ \int \frac{p^2(x_{n-1},x_n|\boldsymbol{y}_{1:n})}{p(x_{n-1}|\boldsymbol{y}_{1:n-1})q(x_n|y_n,x_{n-1})}(\varphi(x_n) - \bar{\varphi}_n)^2 d\boldsymbol{x}_{n-1:n}$$

where $\bar{\varphi}_n = \mathbb{E}_{p(x_n|\boldsymbol{y}_{1:n})}\left(\varphi(X_n)\right)$.

P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004

Central Limit Theorem for SISR

- For Sequential Importance Sampling with Resampling (SISR) one can also show:

$$\sqrt{N}\left(\log\hat{p}_N(\boldsymbol{y}_{1:n}) - \log p(\boldsymbol{y}_{1:n})\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{SMC,n}\right)$$

$$\sigma^2_{SMC,n} = \int \frac{p^2(x_1|\boldsymbol{y}_{1:n})}{q(x_1|y_1)} dx_1 - 1$$

$$+ \sum_{k=2}^{n-1} \int \frac{p^2(x_{k-1}, x_k|\boldsymbol{y}_{1:n})}{p(x_{k-1}|\boldsymbol{y}_{1:k-1})q(x_k|y_k, x_{k-1})} d\boldsymbol{x}_{k-1:k} - 1$$

$$+ \int \frac{p^2(x_{n-1}, x_n|\boldsymbol{y}_{1:n})}{p(x_{n-1}|\boldsymbol{y}_{1:n-1})q(x_n|y_n, x_{n-1})} d\boldsymbol{x}_{n-1:n} - 1$$

- The variance grows linearly with SMC under mixing assumptions.

P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004

Degeneracy Problem

- For $n-k$ large, the SISR approximation becomes:

$$\hat{p}(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:n}) = \delta_{X_{1:k}^*}(\boldsymbol{x}_{1:k})$$

  i.e. we obtain one single delta Dirac function.

- Thus $\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ is an unreliable representation of $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}), n \to \infty,$ thus SMC does not approximate well joint distributions as n→∞.

- Only marginals $\hat{p}(\boldsymbol{x}_{n-L+1:n}|\boldsymbol{y}_{1:n})$ can be accurately evaluated under the mixing/ergodicity conditions discussed earlier.

- The smoothing recursions discussed in an earlier lecture are needed to approximate well $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}), n \to \infty.$ We will discuss this in a follow up lecture.

Performance When Static Parameters Are Included

- The earlier results also demonstrate that you cannot obtain good performance when you have static parameters, e.g. in the following model:

$$X_1 \sim \mu, X_k | (X_{k-1} = x_{k-1}) \sim f_\theta(\cdot | x_{k-1}), Y_k | (X_k = x_k) \sim g_\theta(\cdot | x_k)$$

- Here with prior $\theta \sim \pi(\theta)$, our goal is to sample $p(\boldsymbol{x}_{1:n}, \theta | \boldsymbol{y}_{1:n})$.
- You can see this by noticing that the dynamic model $Z_n = (X_n, \theta)$ is not ergodic since

$$f(x', \theta' | x, \theta) = \delta_\theta(\theta') f_\theta(x' | x)$$

- Note that the N particles $\theta^{(i)}$ sampled at time n=1, are never modified later on.
- A simple (but not the best) solution consists of adding noise to a fixed parameter to transform it as a time-varying parameter

$$\theta_n = \theta_{n-1} + \varepsilon_n.$$

Summary of the Key Points of Sequential Monte Carlo Methods

Bayesian Recursion for the State Space Model

- Let us summarize our state space model where the objective is to compute $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$
- We can write the following recursion equation:

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{p(\boldsymbol{y}_{1:n})} = \frac{g(y_n \mid x_n) f(x_n \mid x_{n-1}) p(\boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1})}{p(y_n \mid \boldsymbol{y}_{1:n-1}) p(\boldsymbol{y}_{1:n-1})} =$$

$$= \frac{g(y_n \mid x_n) \overbrace{f(x_n \mid x_{n-1}) p(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1})}^{\text{Predictive: } p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1})}}{p(y_n \mid \boldsymbol{y}_{1:n-1})}$$

where

$$p(y_n \mid \boldsymbol{y}_{1:n-1}) = \int p(y_n, x_n \mid \boldsymbol{y}_{1:n-1}) dx_n = \int g(y_n \mid x_n) p(x_n \mid \boldsymbol{y}_{1:n-1}) dx_n$$

$$= \int g(y_n \mid x_n) f(x_n \mid x_{n-1}) p(x_{n-1} \mid \boldsymbol{y}_{1:n-1}) d\boldsymbol{x}_{n-1:n}$$

- We can write our update equation above in two steps:

$$\textit{Step I - Prediction}: p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1}) = f(x_n \mid x_{n-1}) p(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1})$$

$$\textit{Step II - Update}: p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{g(y_n \mid x_n) p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1})}{p(y_n \mid \boldsymbol{y}_{1:n-1})} \propto g(y_n \mid x_n) p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1})$$

Monte Carlo Implementation of the Prediction

- Assume that at step $n-1$ we have the following Monte Carlo approximation:

$$\hat{p}(\boldsymbol{x}_{1:n-1}|\boldsymbol{y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n-1}^{(i)}}(\boldsymbol{x}_{1:n-1})$$

- Considering $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1}) = f(x_n \mid x_{n-1})\, p(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1})$, by sampling

$$\tilde{X}_n^{(i)} \sim f\left(x_n, \tilde{X}_{n-1}^{(i)}\right)$$

and setting:

$$\tilde{\boldsymbol{X}}_{1:n}^{(i)} = \left(\tilde{\boldsymbol{X}}_{1:n-1}^{(i)}, \tilde{X}_n^{(i)}\right)$$

we predict:

$$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

Importance Sampling Implementation of Update Step

- Our update equation for our target distribution is:

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{g(y_n \mid x_n)\, p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1})}{p(y_n \mid \boldsymbol{y}_{1:n-1})}$$

- Substituting our Monte Carlo approximation

we obtain: $$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

$$\hat{p}(y_n|\boldsymbol{y}_{1:n-1}) = \int g(y_n|x_n)\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n-1})d\boldsymbol{x}_{1:n} = \int g(y_n|x_n)\frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n} \Rightarrow$$

$$\hat{p}(y_n|\boldsymbol{y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} g(y_n|\tilde{\boldsymbol{X}}_{1:n}^{(i)})$$

- Finally, we can approximate our target distribution as:

$$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{g(y_n|x_n)\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n-1})}{\hat{p}(y_n|\boldsymbol{y}_{1:n-1})} = \frac{g(y_n|x_n)\frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})}{\hat{p}(y_n|\boldsymbol{y}_{1:n-1})}$$

$$= \sum_{i=1}^{N} \frac{g(y_n|\tilde{X}_n^{(i)})}{\sum_{n=1}^{N} g(y_n|\tilde{\boldsymbol{X}}_{1:n}^{(i)})} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

Importance Sampling Implementation of Update Step

- Thus our update equation is:

$$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

$$W_n^{(i)} \propto g(y_n|\tilde{X}_n^{(i)}),\ with \sum_{i=1}^{N} W_n^{(i)} = 1$$

- To obtain N samples $\boldsymbol{X}_{1:n}^{(i)}$ approximately distributed according to $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$, resample N times with replacement:

$$\boldsymbol{X}_{1:n}^{(i)} \sim \hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

to obtain (at a cost $\mathcal{O}$(N)):

$$\tilde{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}) = \sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

$$\left\{N_n^{(i)}\right\} \sim \mathcal{M}ultinomial\left\{W_n^{(i)}\right\},$$

$$\mathbb{E}\left[N_n^{(i)}\right] = NW_n^{(i)},\ Var\left[N_n^{(i)}\right] = NW_n^{(i)}\left(1 - W_n^{(i)}\right)$$

The SMC Algorithm

At step n=1:

- Sample $\tilde{X}_1^{(i)} \sim \mu(x_1)$ and then approximate:

$$\hat{p}(x_1|y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{\tilde{X}_1^{(i)}}(x_1),\ W_1^{(i)} \propto g\left(y_1, \tilde{X}_1^{(i)}\right)$$

- Resample $X_1^{(i)} \sim \tilde{p}(x_1|y_1)$ to obtain $\hat{p}(x_1|y_1) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_1^{(i)}}(x_1)$

At step n≥2:

- Sample $\tilde{X}_1^{(i)} \sim f\left(x_n|X_{n-1}^{(i)}\right)$, set $\tilde{X}_{n-1}^{(i)} \sim \left(X_{n-1}^{(i)}, \tilde{X}_n^{(i)}\right)$ and then approximate:

$$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),\ W_n^{(i)} \propto g(y_n|\tilde{X}_n^{(i)}),\ with \sum_{i=1}^{N} W_n^{(i)} = 1$$

- Resample $\boldsymbol{X}_{1:n}^{(i)} \sim \hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ to finally obtain:

$$\tilde{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- N Gordon, D J Salmond, AFM Smith, Novel Approach to nonlinear non Gaussian Bayesian state estimation, IEE, 1993
- M Isard and A Blake, Contour tracking by stochastic propagation of conditional density, Int J Comp Vision 29(1), 5–28 (1998)

SMC Output

- At step n we thus have:

$$\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\tilde{\boldsymbol{X}}_{1:n}^{(i)}}(\boldsymbol{x_{1:n}}),\ W_n^{(i)} \propto g(y_n|\tilde{X}_n^{(i)}),\ with \sum_{i=1}^{N} W_n^{(i)} = 1$$

$$\tilde{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- Using these approximations, we can now compute the following marginal likelihood:

$$\hat{p}(\boldsymbol{y}_{1:n}) = \prod_{k=1}^{n} \hat{p}(y_k|\boldsymbol{y}_{1:k-1}) = \prod_{k=1}^{n} \int \hat{p}(y_k, x_k|\boldsymbol{y}_{1:k-1})\, dx_k = \prod_{k=1}^{n} \left(\frac{1}{N}\sum_{k=1}^{N} g\left(y_k|\tilde{X}_k^{(i)}\right) \right)$$

where: $\hat{p}(y_k, x_k|\boldsymbol{y}_{1:k-1}) = g(y_k|x_k)\tilde{p}(x_k|\boldsymbol{y}_{1:k-1}) = g(y_k|x_k)\frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{X}_k^{(i)}}(\boldsymbol{x}_{1:k})$

- The computational cost of SMC is $\mathcal{O}$ (N) at each step and memory requirements $\mathcal{O}$(nN). The cost remains $\mathcal{O}$(N) if our interest is on computing $p\left(x_n \mid \boldsymbol{y}_{1:n}\right)$ or $p\left(s_n\left(\boldsymbol{x}_{1:n}\right) \mid \boldsymbol{y}_{1:n}\right)$ $where: s_n\left(\boldsymbol{x}_{1:n}\right) = \Phi_n\left(x_n, s_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\right)$ where Φ is a function of the state.

Degeneracy Problem

- For any N and k, there exist $n(k,N)$ s.t. for any $n \geq n(k,N)$

$$\hat{p}(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k}) = \delta_{\boldsymbol{x}_{1:k}^*}(\boldsymbol{x}_{1:k})$$

i.e. we obtain one single delta Dirac function.

- Thus $\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ is an unreliable representation of

$p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right), n \to \infty$, thus SMC does not approximate well joint

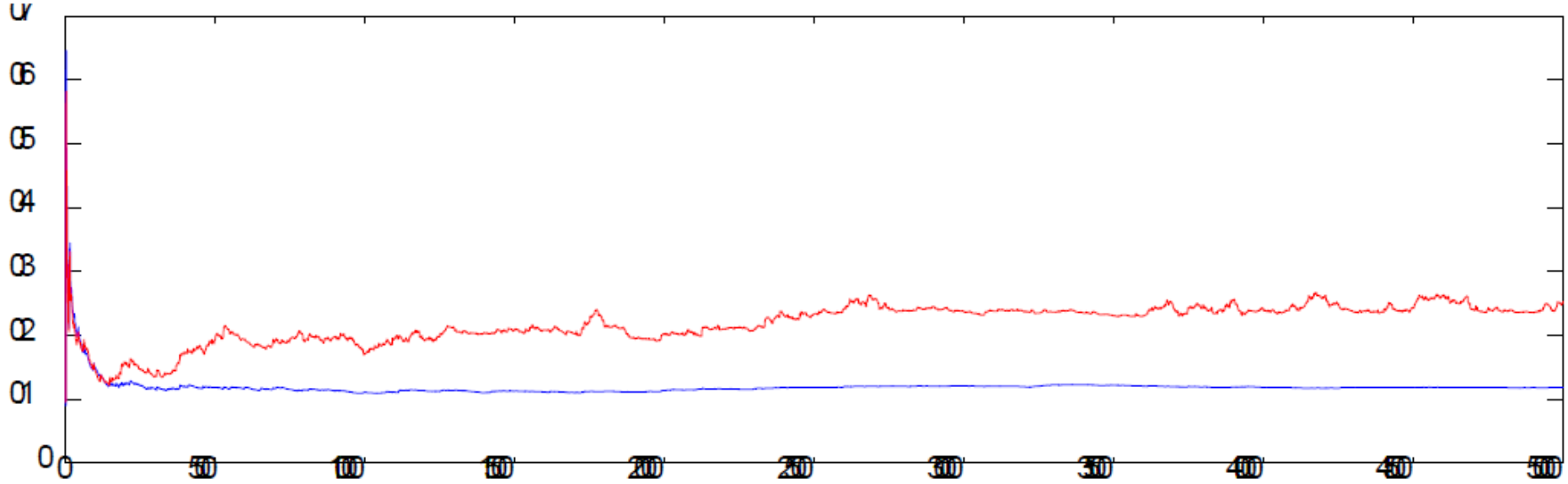
distributions as n→∝.

Degeneracy Problem

- The Figure below shows

$$\frac{1}{n}\mathbb{E}\left[\sum_{k=1}^{n} X_k \mid \boldsymbol{y}_{1:n}\right]$$

Computed with Kalman filter (blue) versus SMC (red) for N=1000.
As n increases, the SMC estimate obviously deteriorates.



From SMC lecture, A. Doucet.

Convergence Results

- Consider any function $\varphi_n : \mathcal{X}^n \to \mathbb{R}$ and consider computing its expectation with SMC:

$$\bar{\varphi}_n = \int \varphi_n(\boldsymbol{x}_{1:n}) p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n} \ (exact)$$

$$\hat{\varphi}_n = \int \varphi_n(\boldsymbol{x}_{1:n}) \tilde{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n} = \frac{1}{N}\sum_{n=1}^{N} \varphi_n\left(\boldsymbol{X}_{1:n}^{(i)}\right) \ (SMC)$$

- Assuming (e.g. bounded likelihood), one can show that for p>0

$$\mathbb{E}[|\bar{\varphi}_n - \hat{\varphi}_n|^p]^{1/p} \leq \frac{C_n}{\sqrt{N}}$$

$$\lim_{N\to\infty} \sqrt{N}(\bar{\varphi}_n - \hat{\varphi}_n) \to \mathcal{N}(0, \sigma_n^2)$$

- Unfortunately, C_n and σ_n^2 can increase with n and will for a path dependent $\varphi_n\left(\boldsymbol{x}_{1:n}\right)$ as the degeneracy problem suggests.

Convergence Results

- Assume an exponentially stability condition: For any $x_1, x_1^{'}$

$$\frac{1}{2}\int \left| p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{2:n}, X_1 = x_1\right) - p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{2:n}, X_1 = x_1^{'}\right)\right| dx_n \leq a^n, \forall |a| < 1$$

- Then one can show for any marginal function $\varphi_n\left(\boldsymbol{x}_{1:n}\right) = \varphi\left(x_n\right)$

$$\mathbb{E}[|\bar{\varphi}_n - \hat{\varphi}_n|^p]^{1/p} \leq \frac{C}{\sqrt{N}}$$

$$\lim_{N\to\infty} \sqrt{N}(\bar{\varphi}_n - \hat{\varphi}_n) \to \mathcal{N}(0, \sigma_n^2) \, with \, \sigma_n^2 \leq D$$

- *C* and *D* are exponential in $dim(X_n)$.

Convergence and Resampling

- The following convergence result can also be shown for the marginal likelihood:

$$\lim_{N\to\infty} \sqrt{N}\left(\log\tilde{p}(\boldsymbol{y}_{1:n}) - \log p(\boldsymbol{y}_{1:n})\right) \to \mathcal{N}(0, \bar{\sigma}_n^2),\ \bar{\sigma}_n^2 \leq An$$

Now the variance increases linearly with n rather than exponentially.

- Note that without resampling, the variance for the marginal likelihood increases exponentially (thus resampling is essential):

$$\log\hat{p}(\boldsymbol{y}_{1:n}) = \log\frac{1}{N}\sum_{n=1}^{N} g\left(y_k|\tilde{X}_k^{(i)}\right)$$

Improving the Sampling Step

- The SMC algorithm discussed earlier is very inefficient. This is particularly true for vague prior $f(x_n \mid x_{n-1})$ or picky likelihood $g(y_n \mid x_n)$
- Consider the following:

$$f(x_n \mid x_{n-1}) = \mathcal{N}(x_n; x_{n-1}, \sigma_v^2)$$

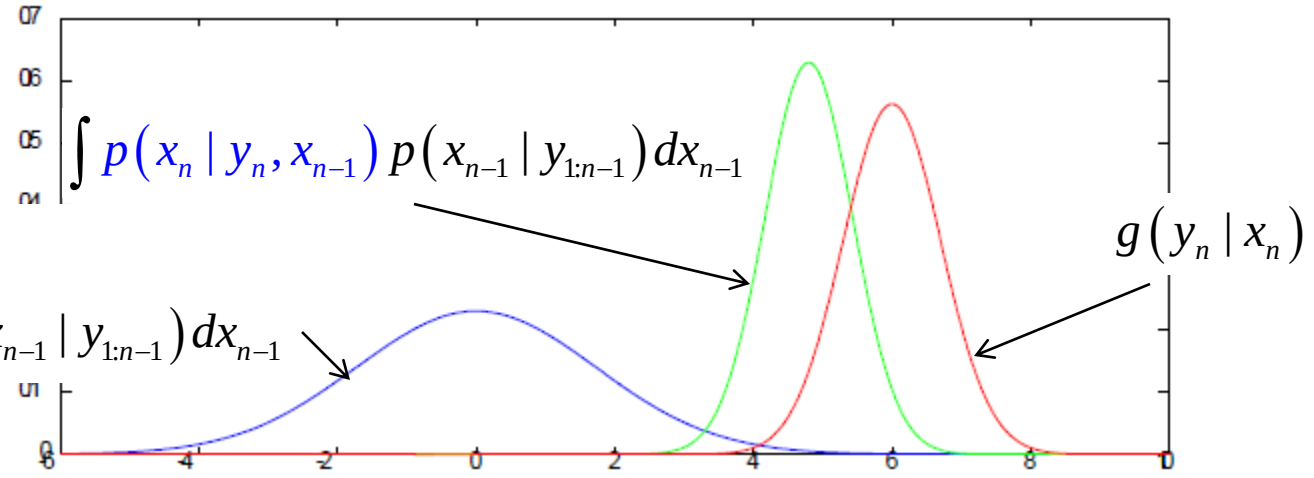
$$g(y_n \mid x_n) = \mathcal{N}(y_n; x_n, \sigma_w^2)$$

$$p(x_{n-1} \mid y_{1:n-1}) = \mathcal{N}(x_{n-1}; m, \sigma^2)$$

- Optimal Proposal/Perfect adaptation:

$$Sample: p(x_n \mid y_n, x_{n-1}) \propto g(y_n \mid x_n) f(x_n \mid x_{n-1}), Resample: W_n \propto p(y_n \mid x_{n-1})$$

$$p(x_n \mid y_{1:n-1}) = \int f(x_n \mid x_{n-1}) p(x_{n-1} \mid y_{1:n-1}) dx_{n-1}$$



Approximate Optimal Proposal

- Design analytical approximation of the optimal proposal using the EKF, UKF filters:

$$\hat{p}(x_n|y_n, x_{n-1}) \; of \; p(x_n|y_n, x_{n-1})$$

- We then sample $\hat{p}(x_n|y_n, x_{n-1})$

and set the weights as:

$$W_n \propto \frac{g(y_n|x_n)f(x_n|x_{n-1})}{\hat{p}(x_n|y_n, x_{n-1})}$$

M.K. Pitt and N. Shephard, Filtering via Simulation: Auxiliary Particle Filter, JASA, 1999

Resample Move Using MCMC

- After the resampling step, you have

$$\boldsymbol{X}_{1:n}^{(i)} = \boldsymbol{X}_{1:n}^{(j)} \quad for\ i \neq j$$

- To add diversity among particles, use an MCMC kernel

$$\boldsymbol{X}_{1:n}^{'(i)} \sim K_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{X}_{1:n}^{(i)}\right)$$

where:

$$p\left(\boldsymbol{x}_{1:n}^{'} \mid \boldsymbol{y}_{1:n}\right) = \int p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) K_n\left(\boldsymbol{x}_{1:n}^{'} \mid \boldsymbol{x}_{1:n}\right) d\boldsymbol{x}_{1:n}$$

- Note that the MCMC kernel K_n does not need to be ergodic.

W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models, JRSS B, 2001

RAO-BLACKWELLISED PARTICLE FILTER (RBPF)

RBPF

- In some models we can partition the hidden variables $\mathbf{q}_t$ and $\mathbf{x}_t$ such that we can analytically integrate out $\mathbf{x}_t$ provided we know the values of $\mathbf{q}_{1:t}$.

- Thus we only have to sample $\mathbf{q}_{1:t}$ and can represent $p(\mathbf{x}_t|\mathbf{q}_{1:t})$ parametrically. Each particle *s now* represents a value for $\boldsymbol{q}_{1:t}^{s}$ and a distribution of the form $p(x_t|\boldsymbol{y}_{1:t}, \boldsymbol{q}_{1:t}^{s})$.

- These hybrid particles are called **distributional particles** or **collapsed particles.**

- The advantage of this approach is that we reduce the dimensionality of the space in which we are sampling, which reduces the variance of our estimate.

- This technique is known as **Rao-Blackwellised particle filtering** or **RBPF.**

- Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

RBPF for Switching LG-SSMs

- As an example of the RBPF we consider an application to the switching linear dynamical system (SLDS).

- We can represent $p(x_t|\boldsymbol{y}_{1:t}, \boldsymbol{q}_{1:t}^s)$ using a mean and covariance matrix for each particle *s*, where $q_t \in \{1, \ldots, K\}$.

- If we propose from the prior $q(q_t = k|q_{t-1}^s)$ the weight update becomes

$$w_t^s \propto w_{t-1}^s p(y_t|q_t = k, \boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t-1}) = w_{t-1}^s L_{t,k}^s$$

$$L_{t,k}^s = \int p(y_t|q_t = k, x_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{q}_{1:t-1}^s) p(x_t|q_t = k, \boldsymbol{y}_{1:t-1}, \boldsymbol{q}_{1:t-1}, q_{1:t-1}^s) dx_t$$

- The $L_{t,k}^s$ is the predictive density for the new observation y_t conditioned on $q_t = k$ and $\boldsymbol{q}_{1:t-1}^s$. In SLDS models, this can be computed using the normalization constant of the Kalman filter $p(y_t|y_{1;t-1}, \boldsymbol{u}_{1:t}) = \mathcal{N}(y_t|\boldsymbol{C}_t \mu_{t|t-1}, \boldsymbol{S_t})$.

- Chen, R. and S. Liu (2000). Mixture Kalman filters. *J. Royal Stat. Soc. B*.
- Doucet, A., N. de Freitas, and N. J. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

RBPF for SLDS (Mixture of Kalman Filters)

$for\ s = 1{:}S$

$\quad k \sim p(q_t | q_{t-1}^s);$

$\quad q_t^s := k;$

$\quad \left(\boldsymbol{\mu}_t^s, \boldsymbol{\Sigma}_t^s, L_{t,k}^s\right) = KFUpdate(\boldsymbol{\mu}_{t-1}^s, \boldsymbol{\Sigma}_{t-1}^s, y_t, \boldsymbol{\theta}_k);$ // One for each particle

$\quad w_t^s = w_{t-1}^s L_{tk}^s;$

$Normalize\ weights{:}\ w_t^s = \dfrac{w_t^s}{\sum_{s'} w_t^{s'}};$

$Compute\ \hat{S}_{eff} = \dfrac{1}{\sum_{s=1}^{S} (w_t^s)^2};$

$If\ \hat{S}_{eff} < S_{min}\ then$

$\quad Resample\ S\ indices\ \boldsymbol{\pi} \sim \boldsymbol{w}_t;$

$\quad q_t^s = q_t^{\pi}, \boldsymbol{\mu}_t^s = \boldsymbol{\mu}_t^{\pi}, \boldsymbol{\Sigma}_t^s = \boldsymbol{\Sigma}_t^{\pi};$

$\quad w_t^s = {}^1/_S;$

RBPF for Switching LG-SSMs

- If K is small, we can compute the optimal proposal distribution as

$$p(q_t = k|\boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t}) = \hat{p}_{t-1}^s(q_t = k|y_t) = \frac{\hat{p}_{t-1}^s(y_t|q_t = k)\hat{p}_{t-1}^s(q_t = k)}{\hat{p}_{t-1}^s(y_t)}$$

$$= \frac{L_{tk}^s p(q_t = k|q_{t-1}^s)}{\sum_{k'} L_{tk'}^s p(q_t = k'|q_{t-1}^s)}$$

- Here we have denoted $\hat{p}_{t-1}^s(\cdot) = p(\cdot|\boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t-1})$
- We sample from $p(q_t = k|\boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t})$ and assign the resulting particle the following weight

$$w_t^s \propto w_{t-1}^s p(y_t|\boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t-1}) = w_{t-1}^s \sum_k [L_{tk}^s p(q_t = k|q_{t-1}^s)]$$

- Since the weights of the particles are independent of the new value that is actually sampled for q_t we can compute these weights first, and use them to decide which particles to propagate.

- de Freitas, N., R. Dearden, F. Hutter, R. Morales-Menendez, J. Mutch, and D. Poole (2004). Diagnosis by a waiter and a mars explorer. *Proc. IEEE 92*(3)..
- Fearnhead, P. (2004). Exact Bayesian curve fitting and signal segmentation. *IEEE Trans. Signal Processing 53*, 2160–2166.

RBPF for Switching LG-SSMs

$$w_t^s \propto w_{t-1}^s p(y_t|\boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t-1}) = w_{t-1}^s \sum_k [L_{tk}^s p(q_t = k|q_{t-1}^s)]$$

- We choose the fittest particles at $t-1$ using information y_t from time t.
- We pass each sample in the prior through all K models to get K posteriors one per sample. The normalization constants allow us to compute the optimal weights. We then resample S indices.
- Finally, for each old particle s that is chosen, we sample one new state $q_t^s = k$, and use the corresponding posterior from the K possible alternative that we have already computed. This method needs $\mathcal{O}(KS)$ storage, but has the advantage that each particle is chosen using the latest information y_t.
- A further improvement can be obtained by exploiting the fact that the state space is discrete. The resampling method of (Fearnhead 2004) avoids duplicating particles.

One Step of Look-Ahead RBPF for SLDS

$for\ s = 1{:}S$

$\quad For\ k = 1{:}K\ do;$

$\quad\quad \left[\left(\boldsymbol{\mu}_t^s, \boldsymbol{\Sigma}_t^s, L_{t,k}^s\right) = KFUpdate(\boldsymbol{\mu}_{t-1}^s, \boldsymbol{\Sigma}_{t-1}^s, y_t, \boldsymbol{\theta}_k);\right.$

$\quad w_t^s = w_{t-1}^s \sum_k [L_{tk}^s p(q_t = k | q_{t-1}^s)]$

$Normalize\ weights{:}\ w_t^s = \dfrac{w_t^s}{\sum_{s'} w_t^{s'}};$

$Resample\ S\ indices\ \boldsymbol{\pi} \sim \boldsymbol{w}_t$

$For\ s \in \pi\ do$

$\quad Compute\ optimal\ proposal\ p(k | \boldsymbol{q}_{1:t-1}^s, \boldsymbol{y}_{1:t-1}) = \dfrac{L_{tk}^s p(q_t = k | q_{t-1}^s)}{\sum_{k'} L_{tk'}^s p(q_t = k' | q_{t-1}^s)};$

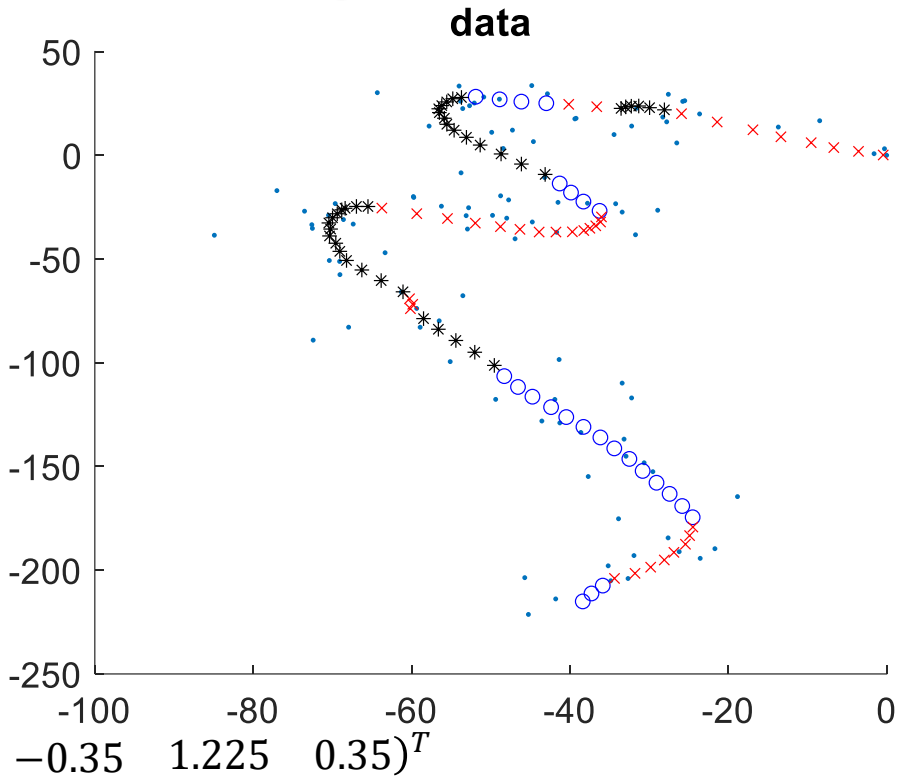
$\quad Sample\ k \sim p(k | \boldsymbol{y}_{1:t}, \boldsymbol{q}_{1:t-1})$

$\quad q_t^s = q_t^\pi, \boldsymbol{\mu}_t^s = \boldsymbol{\mu}_{tk}^s, \boldsymbol{\Sigma}_t^s = \boldsymbol{\Sigma}_{tk}^s;$

$\quad w_t^s = {}^1/_S;$

Tracking of an Object

- One application of SLDS is to track moving objects that have piecewise linear dynamics.
- q_t can specify if the object is flying normally or is taking evasive action (**maneuvering target tracking**)
- In this example we add a three-state discrete Markov chain which controls the input of the system.



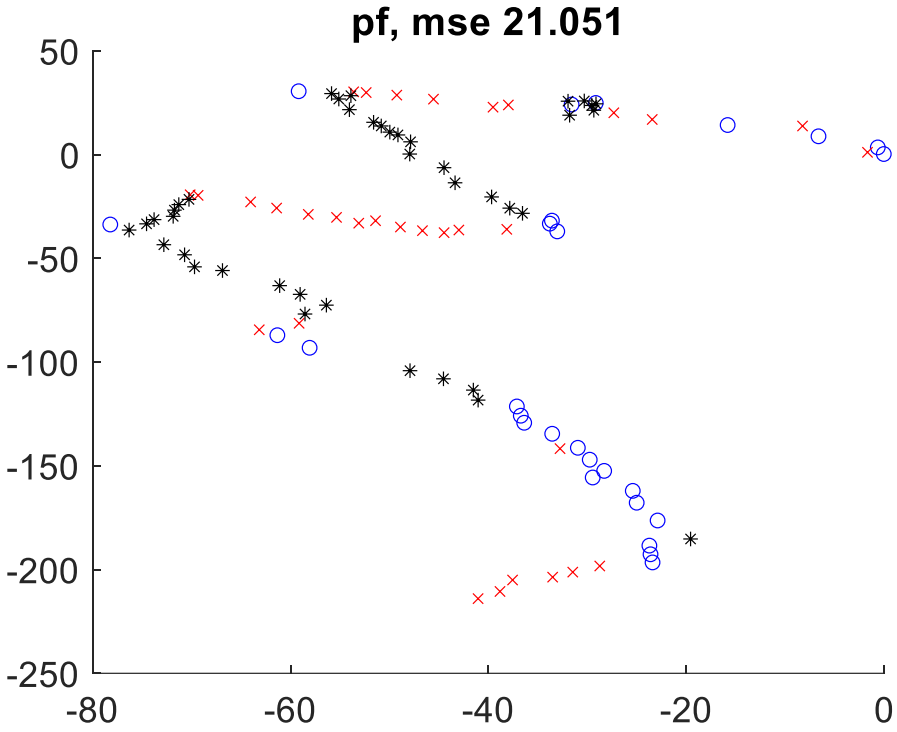
$$u_t = 1, \boldsymbol{B}_1 = (0 \quad 0 \quad 0 \quad 0)^T, \boldsymbol{B}_2 = (-1.225 \quad -0.35 \quad 1.225 \quad 0.35)^T$$
$$\boldsymbol{B}_3 = (1.225 \quad 0.35 \quad -1.225 \quad -0.35)^T$$

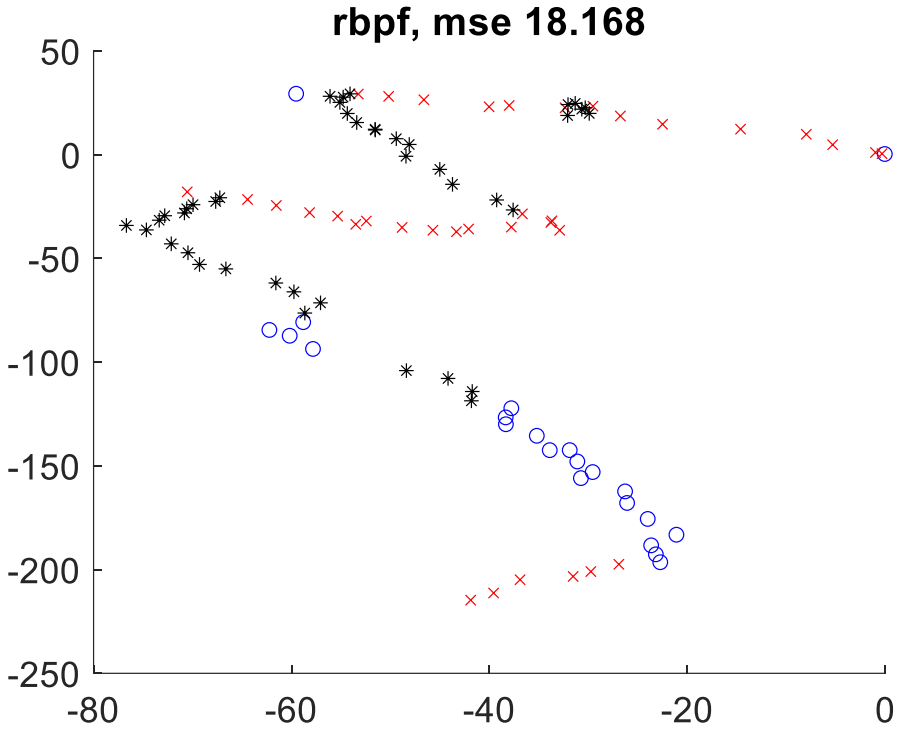
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, q_k = k, \boldsymbol{u}_t, \theta) = \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{A}_k\boldsymbol{x}_{t-1} + \boldsymbol{B}_k\boldsymbol{u}_k, \boldsymbol{Q}_k)$$

Run rbpfManueuverDemo

- The system turns in different directions depending on the discrete state.
- The Fig shows the true state from a sample run starting at (0,0). colored symbols denote the discrete state, and the location of the symbol denotes the (*x*, *y*) location. Small dots represent noisy observations

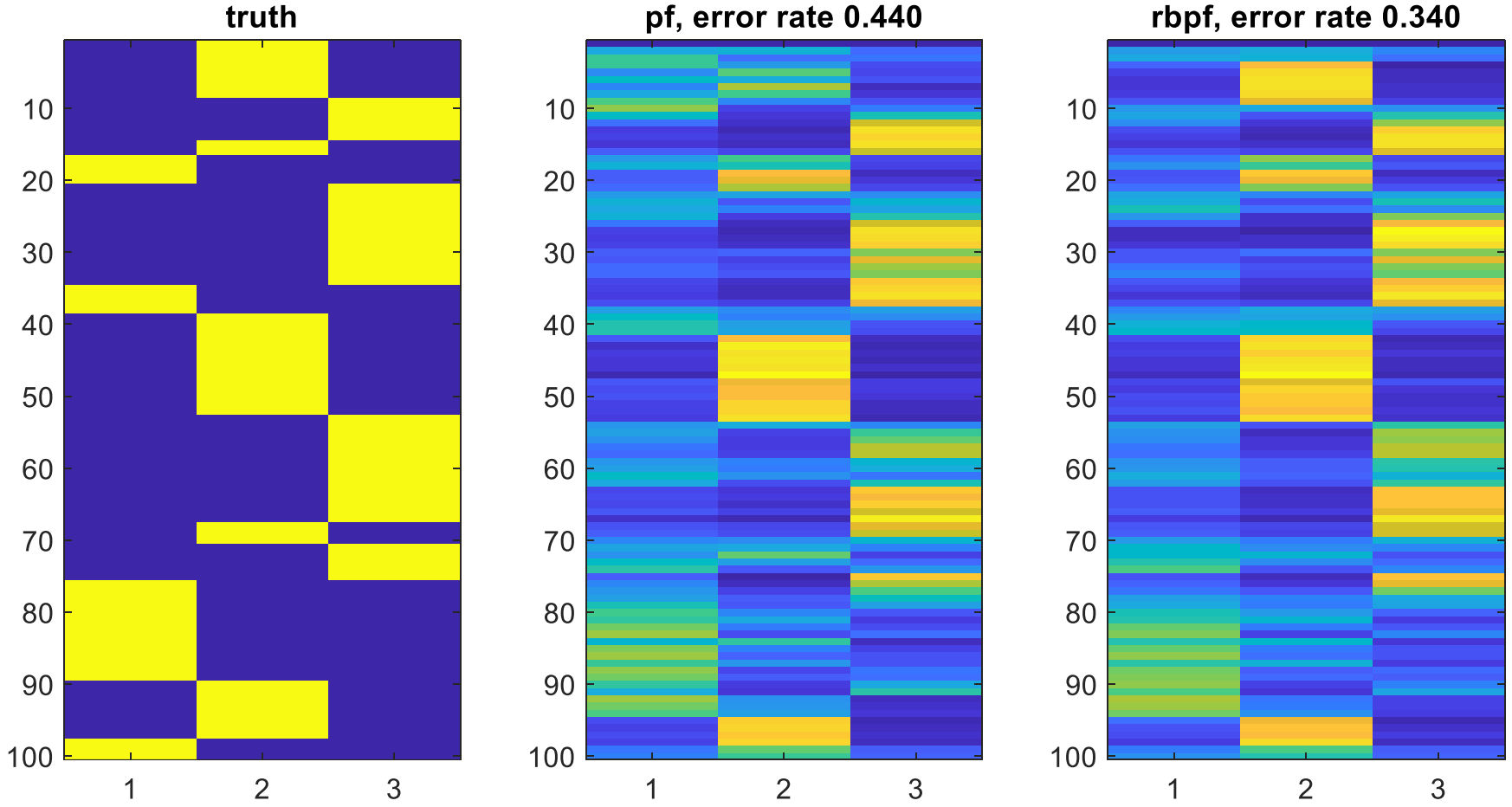
Tracking of an Object





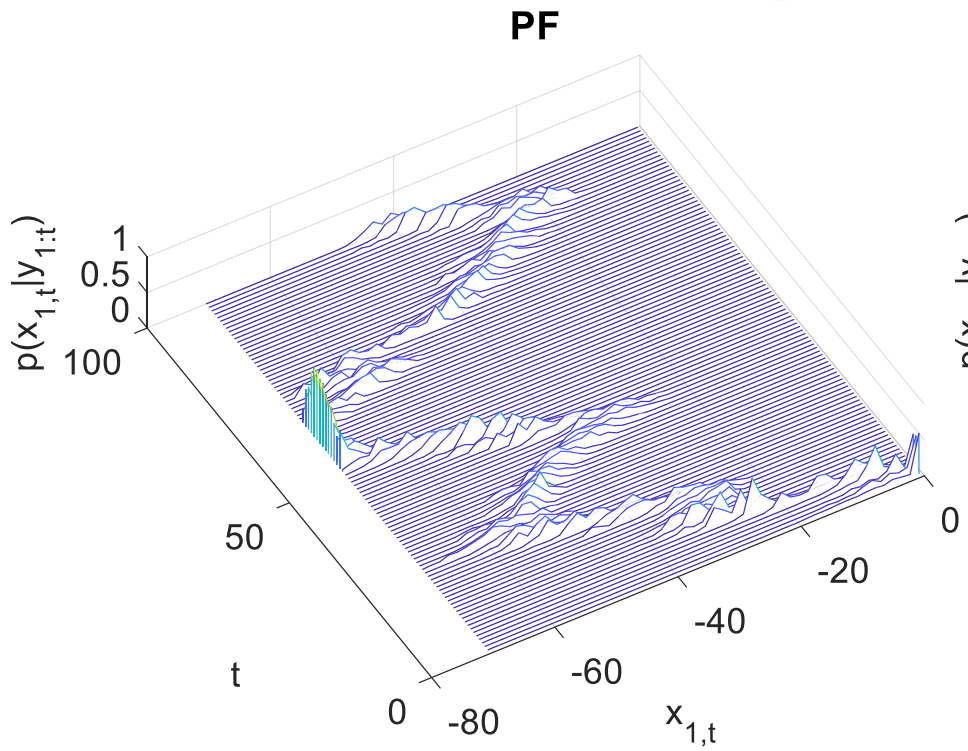
- (Left) shows the state computed using PF (500 particles) where the proposal is to sample from the prior. The colored symbols denote the MAP of the state and the location of the symbol denotes the MMSE (minimum mean square error) estimate (posterior mean).
- (Right) shows the estimate computing using RBPF with 500 particles, using the optimal proposal distribution. RBPF has slightly better performance, although it is also slightly slower.

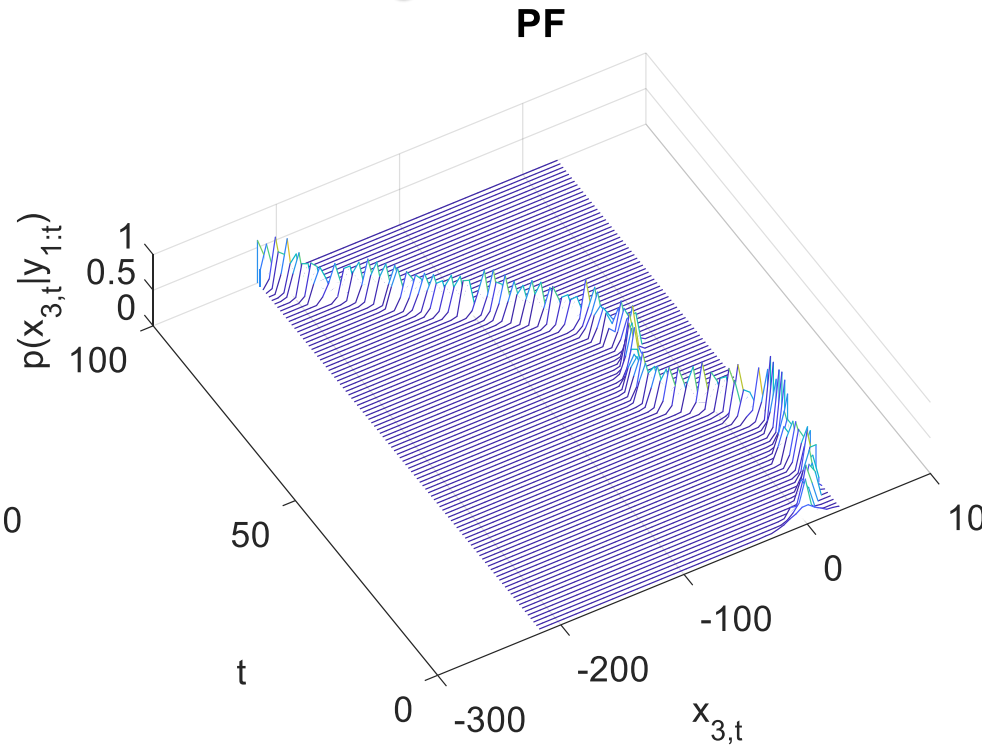
Tracking of an Object



- The distribution over the discrete states. We see that the particle filter estimate of the belief state (second column) is not as accurate as the RBPF estimate (third column) in the beginning, although after the first few observations performance is similar for both methods.

Tracking of an Object

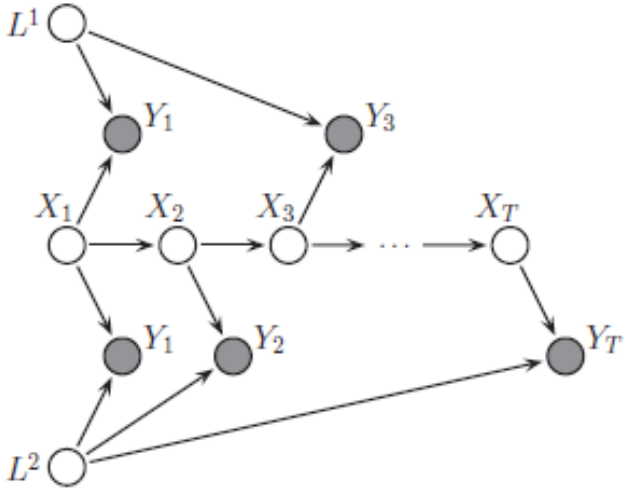




- The posterior over the *x* locations. For simplicity, we use the PF estimate, which is a set of weighted samples, but we could also have used the RBPF estimate, which is a set of weighted Gaussians.

Fast Slam

- In the problem of simultaneous localization and mapping or SLAM for mobile robotics, the main problem with the Kalman filter implementation is that it is cubic in the number of landmarks.

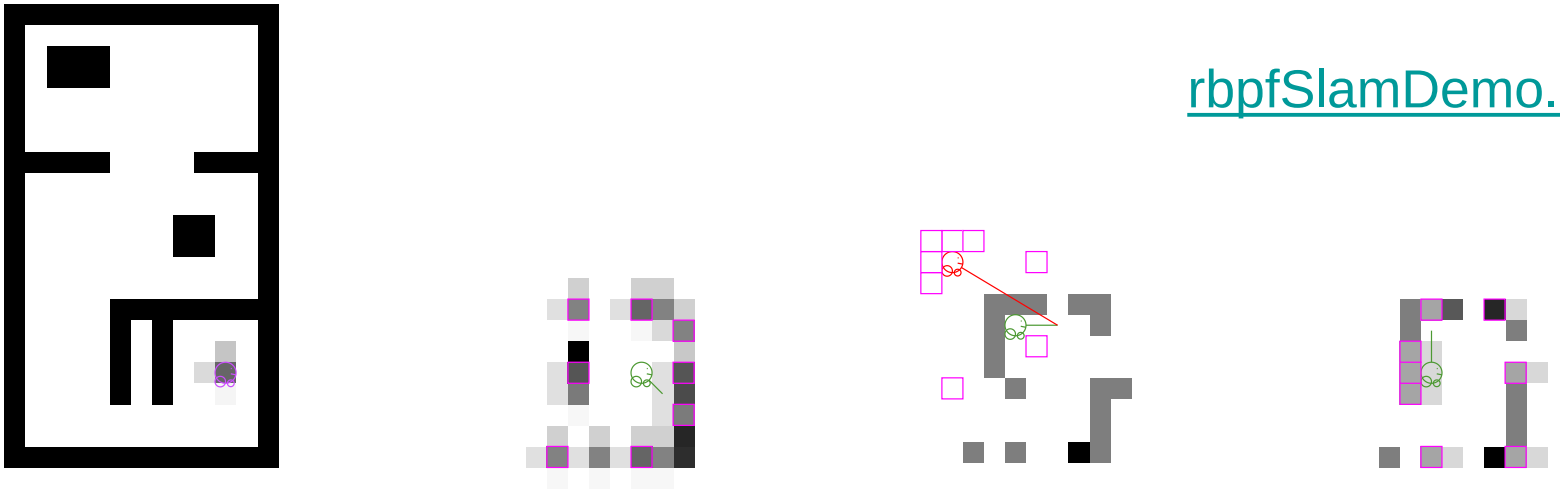


- Conditional on knowing the robot's path, $\boldsymbol{q}_{1:t}$ where $\boldsymbol{q}_t \in \mathbb{R}^2$, the landmark locations $\mathbf{z} \in \mathbb{R}^{2L}$ are independent (the landmarks don't move). That is, $p(\boldsymbol{z}|\boldsymbol{q}_{1:t}, \boldsymbol{y}_{1:t}) = \prod_{l=1}^{L}(z_l|\boldsymbol{q}_{1:t}, \boldsymbol{y}_{1:t})$
- Consequently we can use RBPF, where we sample the robot's trajectory $\boldsymbol{q}_{1:t}$ and we run L independent 2d Kalman filters inside each particle. This takes $\mathcal{O}(L)$ time per particle.
- The number of particles needed for good performance is quite small so the algorithm is essentially linear in the number of particles.

Fast Slam

- This technique has the additional advantage that it is easy to use sampling to handle the data association ambiguity, and that it allows for other representations of the map, such as occupancy grids.
- This idea first discussed in Murphy 2000 and was subsequently extended and made practical in (Thrun et al. 2004) who christened the technique **FastSLAM**.

What is out there **What the robot sees** **What the robot sees** **What the robot sees**



rbpfSlamDemo.m

- Murphy, K. (2000). Bayesian map learning in dynamic environments. In *NIPS*, Volume 12.
- Thrun, S., M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot (2004). Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *J. of Machine Learning Research 2004*.

APPLICATION: ROBOT LOCALIZATION

Robot Localization

- Consider a mobile robot wandering around an office environment. We will assume that it already has an **occupancy grid** which specifies whether each grid cell is empty space or occupied by something solid like a wall.

- The goal is for the robot to estimate its location. This can be solved optimally using an HMM filter, since we are assuming the state space is discrete.

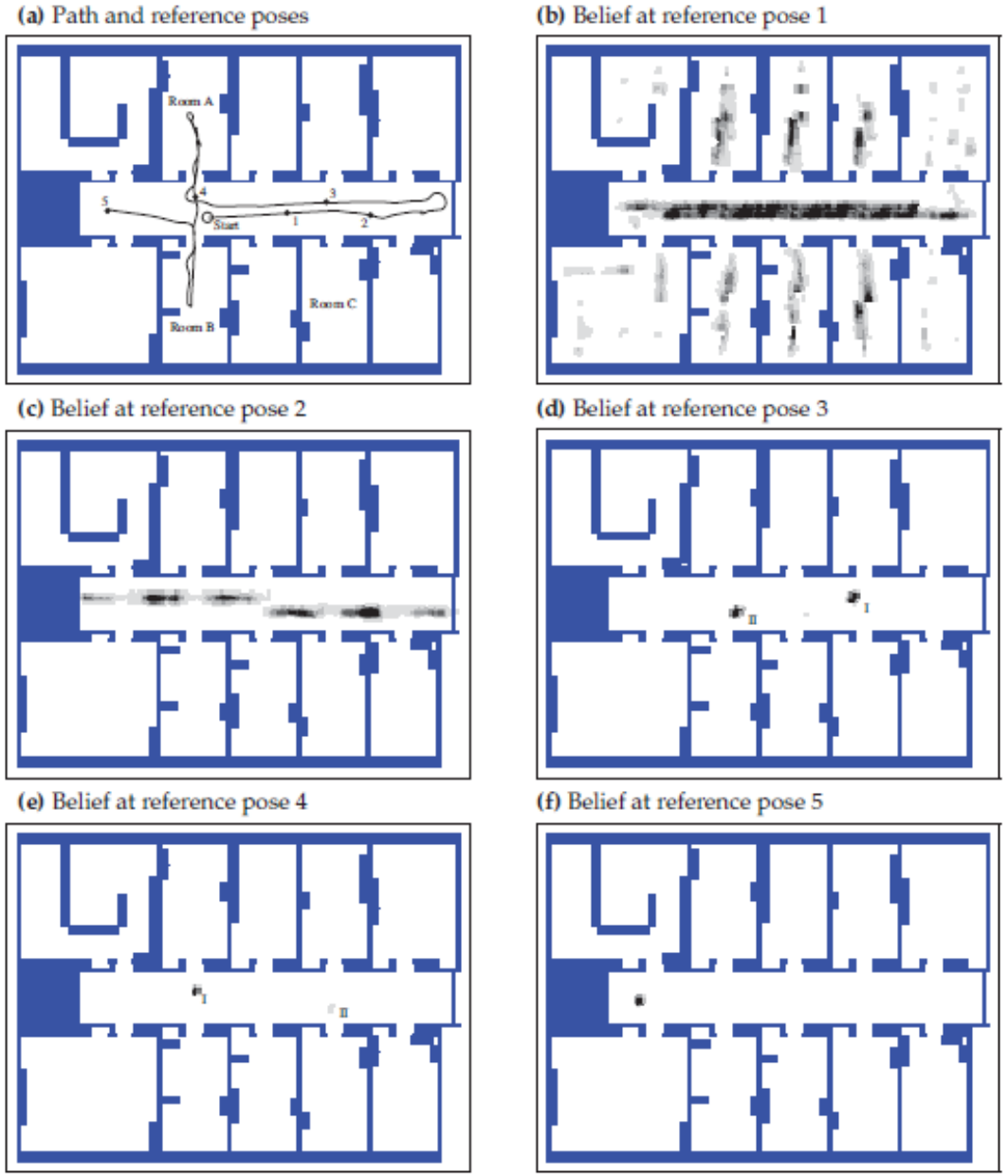
- However, since the number of states, K, is often very large, the $\mathcal{O}(K^2)$ time complexity per update is prohibitive. We can use a particle filter as a sparse approximation to the belief state.

- This is known as **Monte Carlo localization**.

- Thrun, S., W. Burgard, and D. Fox (2006). *Probabilistic Robotics*. MIT Press.

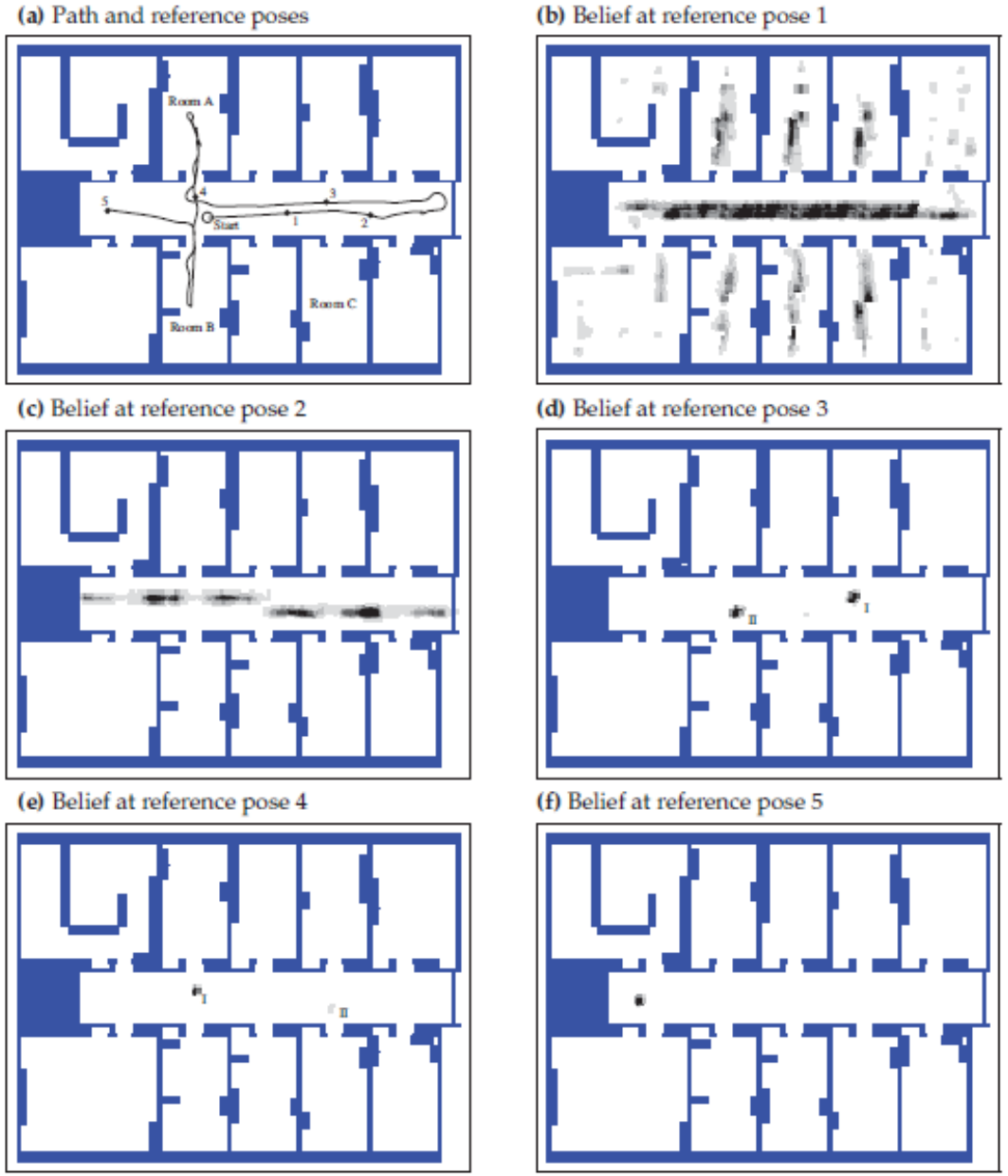
Robot Localization

- The robot uses a sonar range finder to sense distance to obstacles. It starts out with a uniform prior at an arbitrary location. Figuring out where you are starting from a uniform prior is called **global localization**.
- After the first scan, which indicates two walls on either side, the belief state is shown in (b). The posterior is broad. The robot could be in any location where the walls are close by (a corridor or a narrow room).



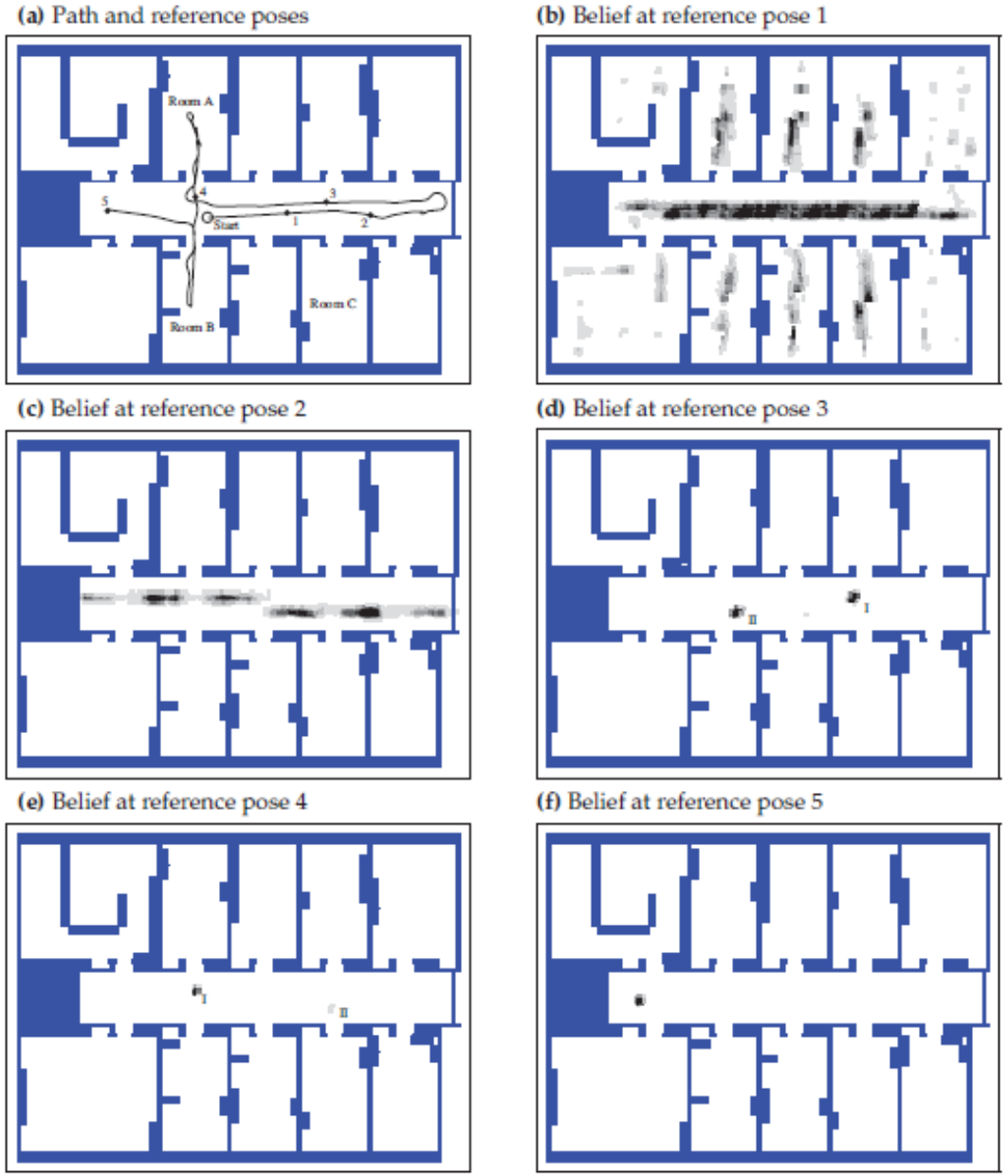
Robot Localization

- After moving to location 2, the robot is certain that it must be in the corridor as shown in (c).
- After moving to location 3, the sensor is able to detect the end of the corridor.
- Due to symmetry, it is not sure if it is in location I (the true location) or location II. (**perceptual aliasing** - different things may look the same.)



Robot Localization

- After moving to locations 4 and 5, it is finally able to figure out precisely where it is.
- The whole process is analogous to someone getting lost in an office building, and wandering the corridors until they see a sign they recognize.



Bayes Filters: Framework

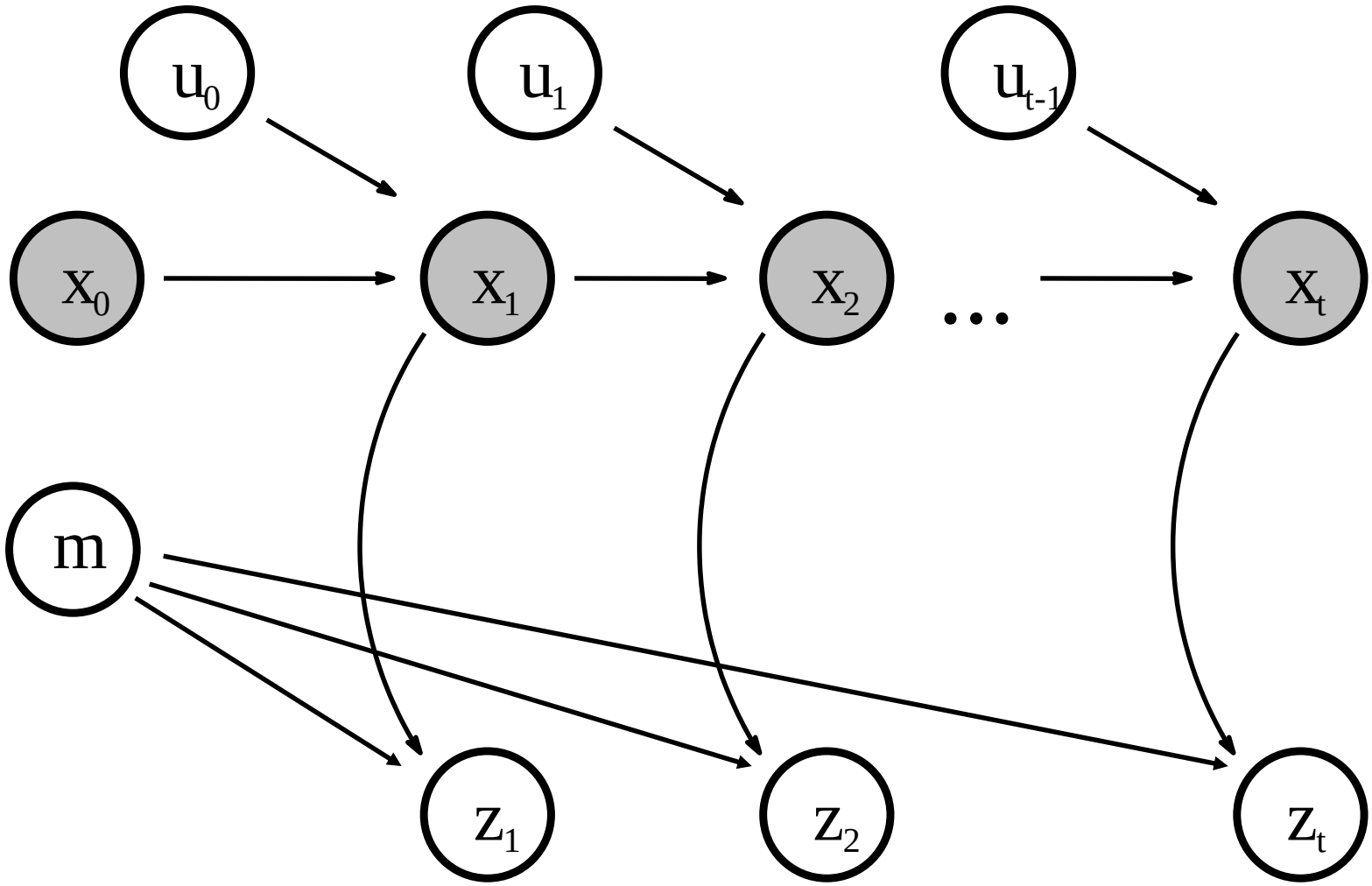
- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_2 \ldots, u_{t-1}, z_t\}$$

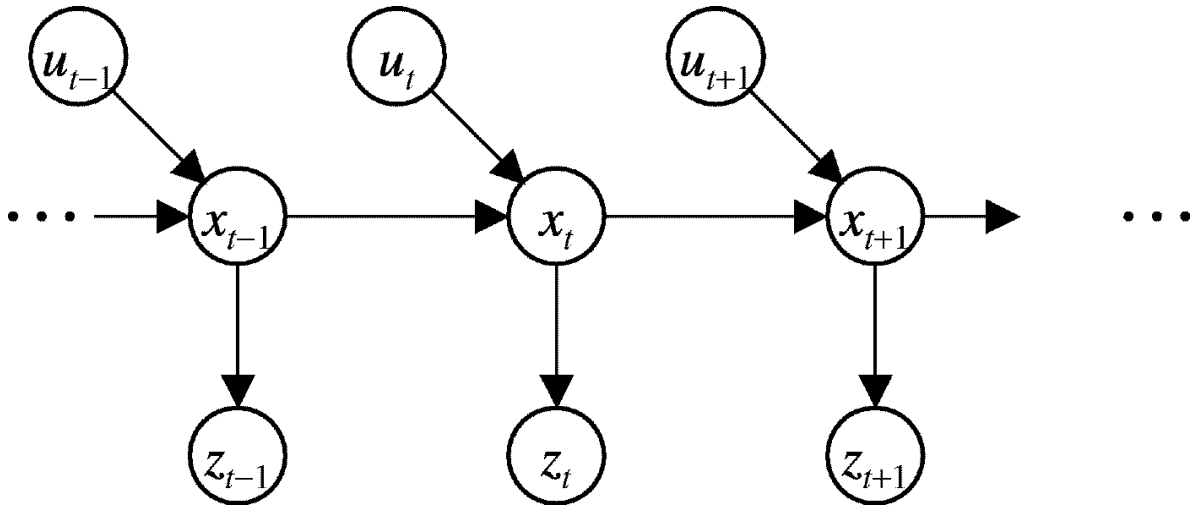
  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of the dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

Graphical Model of Localization



Markov Assumption

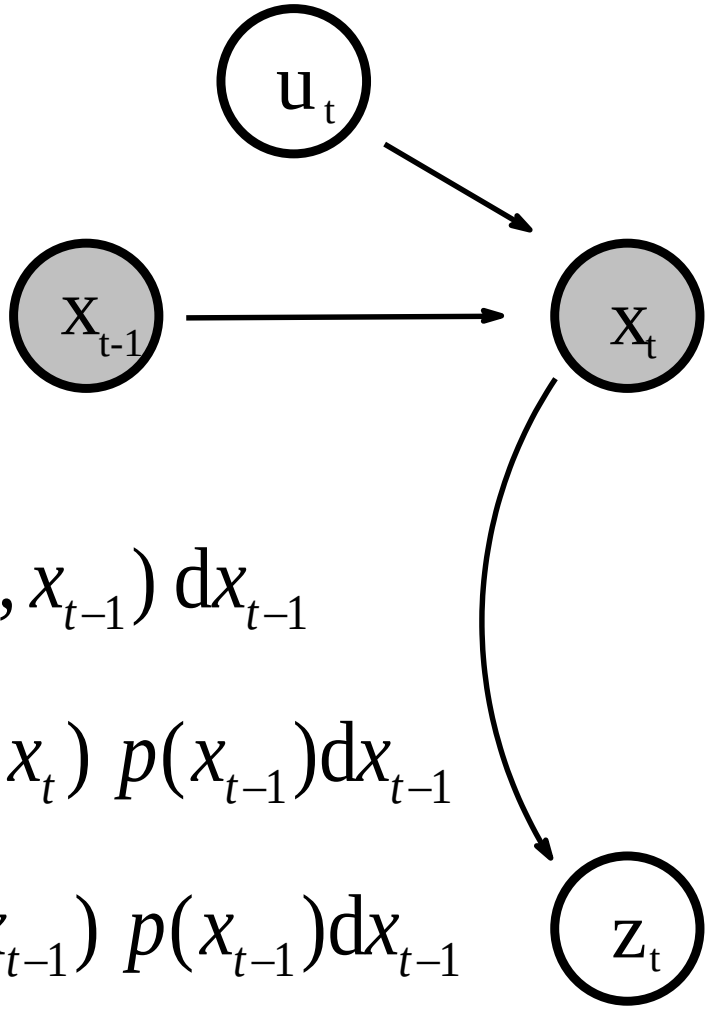


$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$
$$p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

Belief Update



$$p(x_t \mid u_t, z_t) = \alpha \int p(x_t, u_t, z_t, x_{t-1})\, \mathrm{d}x_{t-1}$$

$$= \alpha \int p(x_t \mid u_t, x_{t-1})\, p(z_t \mid x_t)\; p(x_{t-1})\, \mathrm{d}x_{t-1}$$

$$= \alpha\; p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1})\; p(x_{t-1})\, \mathrm{d}x_{t-1}$$

Bayes Filters

z = observation
u = action
x = state

$$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_2, \ldots, u_{t-1}) \; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$$

Markov $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$$

Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_2, \ldots, u_{t-1}, x_{t-1}) \; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1}) \; dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1}) \; dx_{t-1}$$

$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

VISUAL OBJECT TRACKING

Particle Filters for Object Tracking

- Consider tracking an object in a video sequence. The method uses a simple linear motion model for the centroid of the object, and a color histogram for the likelihood model, using **Bhattacharya distance** to compare histograms.

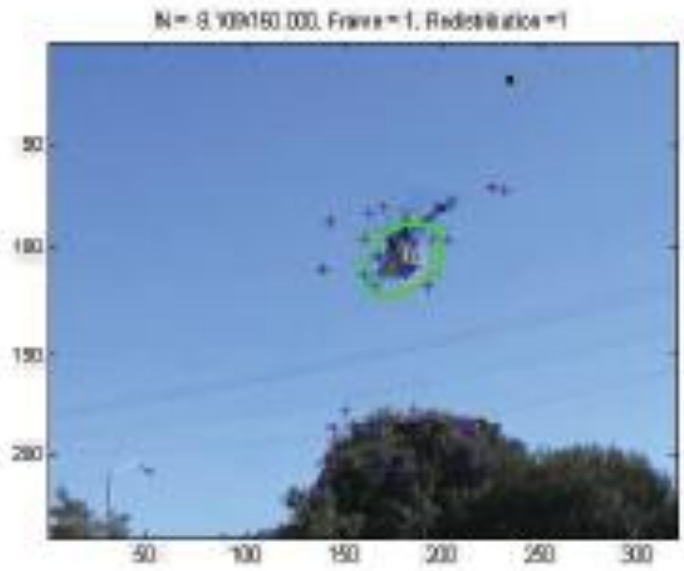
- The proposal distribution is obtained by sampling from the likelihood.

- The example system uses $N = 250$ particles, with an effective sample size of 134.

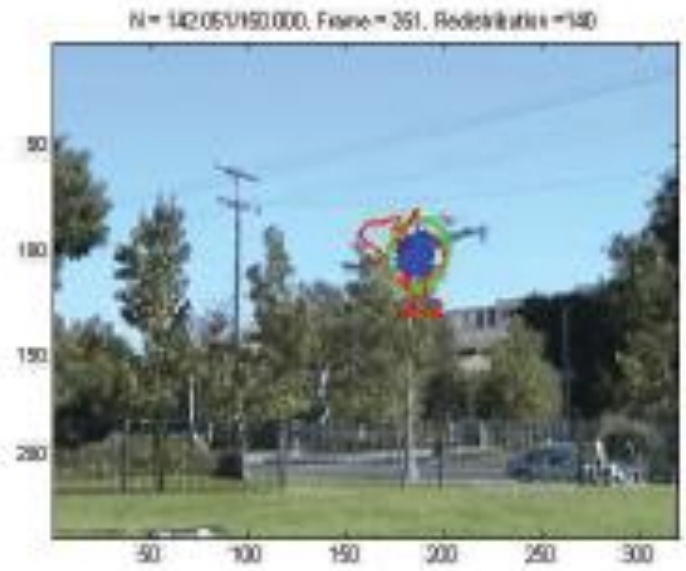
- Nummiaro, K., E. Koller-Meier, and L. V. Gool (2003). An adaptive color-based particle filter. *Image and Vision Computing 21*(1), 99–110.

Particle Filters for Object Tracking

- (a) shows the belief state at frame 1. The system has had to resample 5 times to keep the effective sample size above the threshold of 150;
- (b) shows the belief state at frame 251; the red lines show the estimated location of the center of the object over the last 250 frames.



(a)

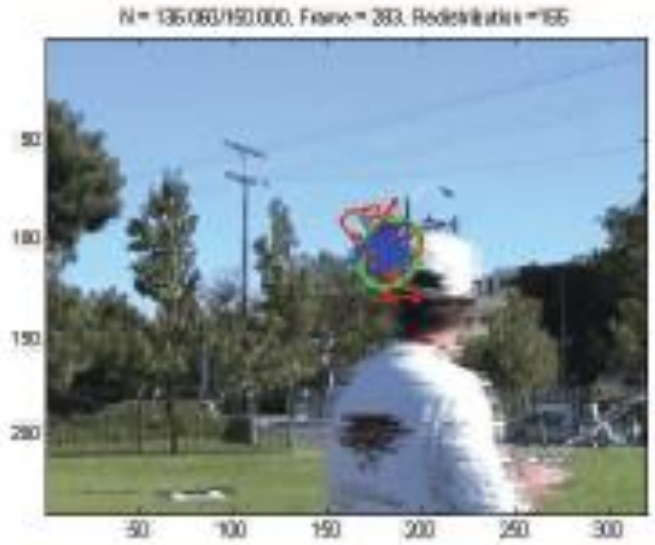


(b)

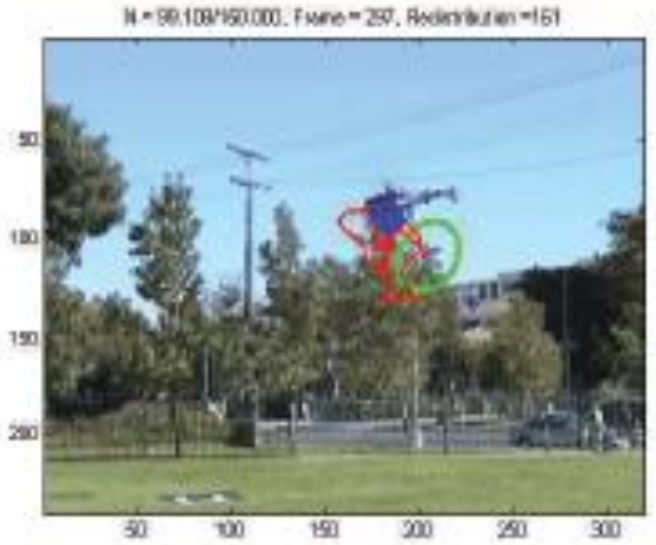
Particle Filters for Object Tracking

- (c) shows that the system can handle visual clutter, as long as it does not have the same color as the target object.

- (d) shows that the system is confused between the grey of the helicopter and of the building. The posterior is bimodal. The green ellipse representing the posterior mean and covariance is in between the two modes.



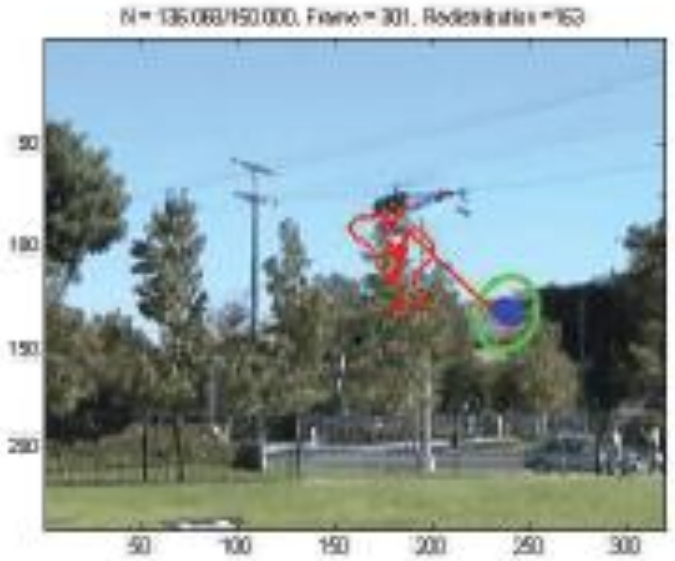
(c)



(d)

Particle Filters for Object Tracking

- (e) shows that the probability mass has shifted to the wrong mode: the system has lost track.
- (f) shows the particles spread out over the gray building; recovery of the object is very unlikely from this state using this



(e)

(f)

Particle Filters for Object Tracking

- We see that the method is able to keep track for a fairly long time, despite the presence of clutter. However, eventually it loses track of the object.
- Note that since the algorithm is stochastic, simply re-running the demo may fix the problem. But in the real world, this is not an option.
- The simplest way to improve performance is to use more particles. An alternative is to perform **tracking by detection**, by running an object detector over the image every few frames.

- Forsyth, D. and J. Ponce (2002). *Computer vision: a modern approach*. Prentice Hall.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- Prince, S. (2012). *Computer Vision: Models, Learning and Inference*. Cambridge.