ANIRUDH SETH
aniset@kth.se

# Tutorial 7
MCMC

2020-02-12

## 2.3

1. **Rejection Sampling**

   a) To implement rejection sampling we sample from another easy-to-sample distribution $q(x)$ such that

   $$\frac{p(x)}{q(x)} \leq M.$$

   If we know $p(x)$ up to a normalizing constant i.e. $p(x) = \frac{1}{Z}\tilde{p}(x)$.

   $$\begin{aligned} \frac{p(x)}{Zq(x)} &\leq \frac{M}{Z} \\ \frac{\tilde{p}(x)}{q(x)} &\leq \frac{M}{Z} \\ \frac{\tilde{p}(x)}{q(x)} &\leq M'. \\ \textit{where, } M' &= \frac{M}{Z} \end{aligned}$$

   The algorithm accepts when $p(x)/Mq(x)$ is less than a realization from a uniform random variable, which is the same as $\tilde{p}(x)/M'q(x)$ being less than the realization. Thus, the algorithm can be implemented.

   b)

   $$\begin{aligned} \mathbb{P}(X \text{ accepted}) &= \mathbb{P}\left(U \leq \frac{\tilde{p}(X)}{M'\, q(X)}\right) \\ &= \int \mathbb{P}\left(U \leq \frac{\tilde{p}x)}{M'\, q(x)}\middle| X = x\right) q(x)\, dx \\ &= \int \frac{p(x)}{M'\, q(x)} q(x)\, dx \\ &= \frac{1}{M'} \\ &= \frac{Z}{M} \end{aligned}$$

2. **Importance Sampling**

   a)

   $$\begin{aligned} \mathbb{E}_{p(x)}[f(\mathbf{x})] &= \int_{-\infty}^{\infty} f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \int_{-\infty}^{\infty} f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &\approx \frac{1}{LZ}\sum_{l=1}^{L} f\left(\boldsymbol{x}_i\right) w\left(\boldsymbol{x}_i\right) \end{aligned}$$

   where $x_i$,$i = 1, 2, \ldots, L$, are samples drawn from $q(x)$ and $w(\boldsymbol{x_i}) := \frac{p(\boldsymbol{x_i})}{q(\boldsymbol{x_i})}$

   b) The importance weights,$w(\boldsymbol{x_i}) := \frac{p(\boldsymbol{x_i})}{q(\boldsymbol{x_i})}$ compensate for the difference between the real distribution $p(x)$ and the sampling distribution $q(x)$. For the standard empirical average

   $$\mathrm{E}_{\mathrm{Pr}}[h(\boldsymbol{X})] \approx \frac{1}{n}\sum_{t=1}^{n} h\left(\boldsymbol{x}^{(t)}\right)$$

all samples drawn from $p(x)$ have the same weight in the approximation. For importance sampling, any sample from $q(x)$ is, compared to $p(x)$, either “over-sampled” (too frequently sampled) and receives a weight less than 1, or “under-sampled” (too infrequently sampled) and receives a weight larger than 1. Figure 1 illustrates the principle behind importance sampling. The dotted line is the sampling distribution, and the solid line is from the target distribution, from which we can’t sample. Samples drawn at locations where the dotted line lies above the solid plot, will be drawn more often than necessary, and vice versa. To correct for that mismatch, the importance weights are required.

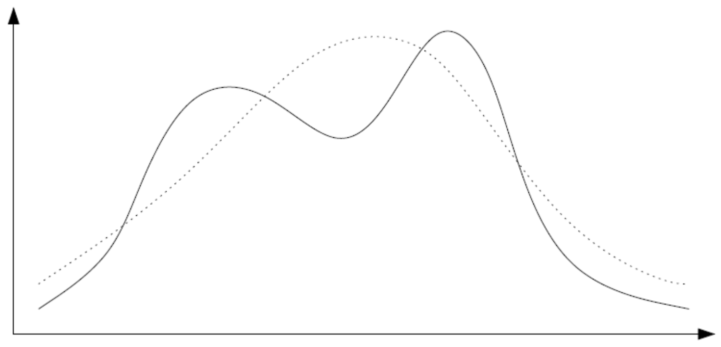

Figure 1: The dotted line is $q(x)$, and the solid line is $p(x)$.

## 3.2

### 3.2.1

1. If $T\left(x^{(i+1)}|x^{(i)}\right)$ is the transition kernel, then we need to show that

$$p\left(x^{(i+1)}\right)T\left(x^{(i)}|x^{(i+1)}\right) = p\left(x^{(i)}\right)T\left(x^{(i+1)}|x^{(i)}\right)$$

The transition kernel for MH is

$$T\left(x^{(i+1)}|x^{(i)}\right) = q\left(x^{(i+1)}|x^{(i)}\right)A\left(x^{(i)},x^{(i+1)}\right) + \delta_{x^{(i)}}\left(x^{(i+1)}\right)r\left(x^{(i)}\right) \tag{1}$$

The function $T\left(x^{(i+1)}|x^{(i)}\right)$ can be decomposed into two parts and we can treat each separately. First we can show that

$$\begin{aligned} T(x^{(i+1)} \mid x^{(i)})p(x^{(i)}) &= T(x^{(i)} \mid x^{(i+1)})p(x^{(i+1)}) \\ &\textbf{PART 1} \\ A(x^{(i+1)},x^{(i)})q(x^{(i+1)} \mid x^{(i)})p(x^{(i)}) &= A(x^{(i)},x^{(i+1)})q(x^{(i)} \mid x^{(i+1)})p(x^{(i+1)}) \\ \min\left\{\frac{p(x^{(i+1)})}{p(x^{(i)})}\frac{q(x^{(i)} \mid x^{(i+1)})}{q(x^{(i+1)} \mid x^{(i)})},1\right\}q(x^{(i+1)} \mid x^{(i)})p(x^{(i)}) &= \min\left\{\frac{p(x^{(i)})}{p(x^{(i+1)})}\frac{q(x^{(i+1)} \mid x^{(i)})}{q(x^{(i)} \mid x^{(i+1)})},1\right\}q(x^{(i)} \mid x^{(i+1)})p(x^{(i+1)} \\ \min(p(x^{(i+1)})q(x^{(i)} \mid x^{(i+1)}),\, q(x^{(i+1)} \mid x^{(i)})p(x^{(i)})) &= \min(p(x^{(i)})q(x^{(i+1)} \mid x^{(i)}),\, q(x^{(i)} \mid x^{(i+1)})p(x^{(i+1)})) \end{aligned}$$

The last line is trivially true because on both sides of the equation we are taking the minimum of the same quantities.

**PART 2**

$$\delta_{x^{(i)}}\left(x^{(i+1)}\right)r\left(x^{(i)}\right) = \delta_{x^{(i+1)}}\left(x^{(i)}\right)r\left(x^{(i+1)}\right) \tag{2}$$

Here $r\left(x^{(i)}\right) = \int q\left(x^*|x^{(i)}\right)\left(1 - A\left(x^{(i)},x^*\right)\right)dx^*$ and $\delta$ is the Dirac delta function. The equation 2 becomes

$$\mathbf{1}\left\{x^{(i+1)} = x^{(i)}\right\}r(x^{(i)})p(x^{(i)}) = \mathbf{1}\left\{x^{(i)} = x^{(i+1)}\right\}r(x^{(i+1)})p(x^{(i+1)})$$

1 is the indicator function. This is trivially true because iff $(x_i)$=$(x_{(i+1)})$ , every other quantity in the above equation is the same regardless.

2. The MH algorithm starts with simulating a "candidate" sample $x_{cand}$ from the proposal distribution $q(x)$.These candidate samples are accepted based on the acceptance probability A. There are mainly two kinds of proposal distributions, symmetric and asymmetric. A proposal distribution is a symmetric distribution if $q(x_{(i+1)}|x_{(i)}) = q(x_{(i)}|x_{(i+1)}$ We may choose a proposal distribution that is inherently asymmetric, such as the log-normal density, which is skewed towards larger values. In other cases, we may need to work with asymmetric proposal distributions to accommodate for particular constraints in our models. For example, if we wish to estimate the posterior distribution for a variance parameter, we require that our proposal does not generate values smaller than 0.A symmetric proposal distribution may be ill-fit for many problems, like when we want to sample from distributions that are bounded on semi infinite intervals (e.g.$[0,\infty)$ ).

3. We can choose the prior as one of the proposal distribution in this case. Suppose that the proposal distribution for the Metropolis-Hastings algorithm is chosen such that for some fixed density $g$

$$g\left(\mathbf{x}^{*}|\mathbf{x}^{(t)}\right) = g\left(\mathbf{x}^{*}\right)$$

This yields an independence chain, where each candidate value is drawn independently of the past. In this case, the Metropolis-Hastings ratio is

$$R\left(\mathrm{x}^{(t)}, \mathrm{X}^{*}\right) = \frac{f\left(\mathrm{X}^{*}\right) g\left(\mathrm{x}^{(t)}\right)}{f\left(\mathrm{x}^{(t)}\right) g\left(\mathrm{X}^{*}\right)}$$

In above notation Metropolis-Hastings notation, $f(\boldsymbol{\theta}) = p(\boldsymbol{x}|\mathbf{y})$ and if we choose the prior as the proposal distribution $g\left(\boldsymbol{x}^{*}\right) = p\left(\boldsymbol{x}^{*}\right)$ Then the Metropolis-Hastings ratio is given by(by applying bayes rule):

$$R\left(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{*}\right) = \frac{p\left(\mathbf{y}|\boldsymbol{x}^{*}\right)}{p\left(\mathbf{y}|\boldsymbol{x}^{(t)}\right)}$$

• If proposal distribution = prior distribution, the Metropolis-Hastings ratio equals the likelihood ratio.
• By definition, the support of the prior covers the support of the target posterior, so the stationary distribution of this chain is the desired posterior.

### 3.2.2

1. The Metropolis-Hastings algorithm for the general case, when the normalizing constant of the target distribution is unknown has been implemented in python.

2. The results of the algorithm for the distribution

$$p(x) \propto a_1\mathcal{N}\left(x|\mu_1, \sigma_1^2\right) + a_2\mathcal{N}\left(x|\mu_2, \sigma_2^2\right)$$

with the following parameters: $a_1 = a_2 = 0.5, \mu_1 = 0, \sigma_1 = 1, \mu_2 = 3, \sigma_2 = 0.5$ can be seen in below figures.A total of 10000 points were sampled for $\sigma_p = [0.1, 1.0, 10.0, 100.0]$.
Lastly, the success or failure of the algorithm often hinges on the choice of proposal distribution. Different choices of the proposal standard deviation $\sigma_p$ also known as proposal width lead to very different results,and will determine how far you propose jumps. If the proposal is too narrow, only one mode of p(x) might be visited(figure 2). On the other hand, if it is too wide, the rejection rate can be very high, resulting in high correlations (figure 5). If all the modes are visited while the acceptance probability is high, the chain is said to "mix" well(figure 3,4)

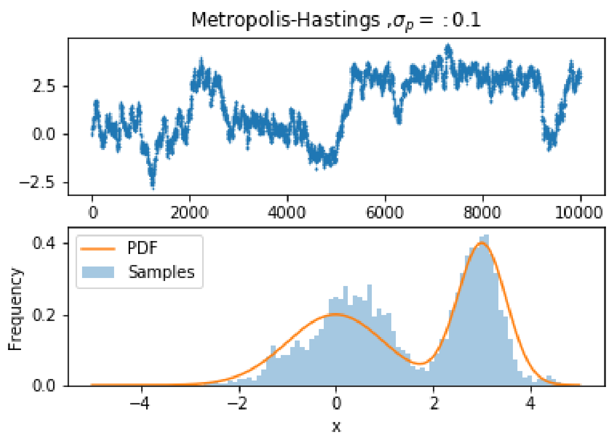

Figure 2: Target distribution and histogram of the MCMC samples with varying $\sigma_p = 0.1$

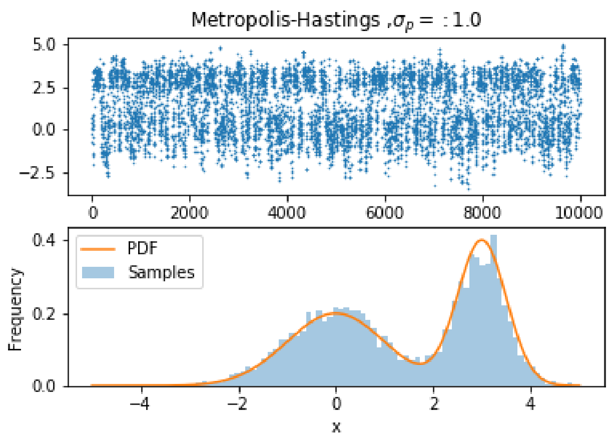

Figure 3: Target distribution and histogram of the MCMC samples with varying $\sigma_p = 1.0$

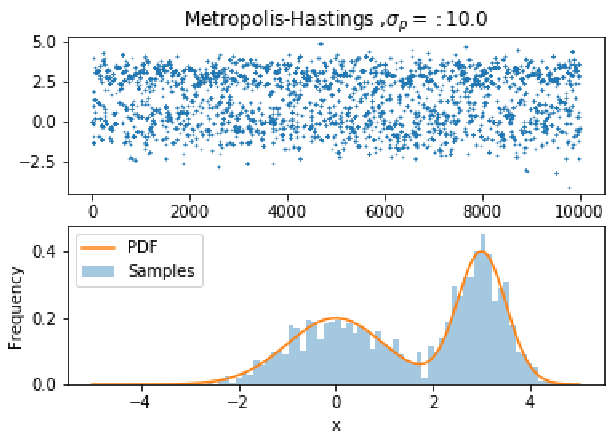

Figure 4: Target distribution and histogram of the MCMC samples with varying $\sigma_p = 10.0$

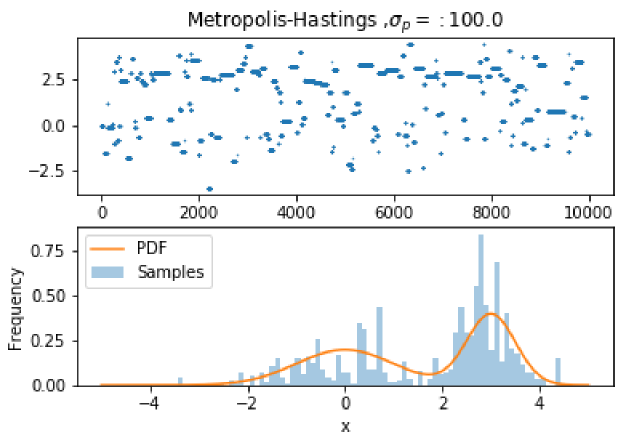

Figure 5: Target distribution and histogram of the MCMC samples with varying $\sigma_p = 100.0$

## 4.4

### 4.4.1

$$\pi\left(x_{0:t}|y_{1:t}\right) = \pi\left(x_{0:t-1}|y_{1:t-1}\right)\pi\left(x_t|x_{0:t-1}, y_{1:t}\right) \tag{3}$$

$$w\left(x_{0:t}\right) = \frac{p\left(x_{0:t}|y_{1:t}\right)}{\pi\left(x_{0:t}|y_{1:t}\right)} \;:\; \text{importance weights} \tag{4}$$

$$\tilde{w}_t^{(i)} = \frac{w\left(x_{0:t}^{(i)}\right)}{\sum_{j=1}^{N} w\left(x_{0:t}^{(j)}\right)} \;:\; \text{normalized importance weights} \tag{5}$$

Returning to the sequential case, at each iteration, one could have samples constituting an approximation to $= p\left(\mathbf{x}_{0k-1}|\mathbf{z}_{1:k-1}\right)$ and want to approximate $p\left(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}\right)$ Twith a new set of samples. If the importance density is chosen mo factorize such that

$$\pi\left(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}\right) = \pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}\right)\pi\left(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}\right)$$

$$\begin{aligned} p\left(x_{0:t}|y_{1:t}\right) &= \frac{p\left(x_{0:t}, y_{1:t}\right)}{p\left(y_{1:t}\right)} \\ &\propto p\left(y_{1:t-1}, y_t|x_{0:t}\right)p\left(x_{0:t}\right) \\ &= p\left(y_t|x_{0:t}, y_{1:t-1}\right)p\left(x_{0:t}|y_{1:t-1}\right) \\ &= p\left(y_t|x_t\right)p\left(x_{0:t}|y_{1:t-1}\right) \\ &= p\left(y_t|x_t\right)p\left(x_t|x_{0:t-1}, y_{1:t-1}\right)p\left(x_{0:t-1}|y_{1:t-1}\right) \\ &= p\left(y_t|x_t\right)p\left(x_t|x_{t-1}\right)p\left(x_{0:t-1}|y_{1:t-1}\right) \end{aligned} \tag{6}$$

By substituting (3) and (6) into (4) the weight update equation can be shown to be

$$\begin{aligned} w_t^i &\propto \frac{p\left(\mathbf{y}_t|\mathbf{x}_t^i\right)p\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right)p\left(\mathbf{x}_{0:t-1}^i|\mathbf{y}_{1:t-1}\right)}{\pi\left(\mathbf{x}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{y}_{1:t}\right)\pi\left(\mathbf{x}_{0:t-1}^i|\mathbf{y}_{1:t-1}\right)} \\ &= w_{t-1}^i \frac{p\left(\mathbf{y}_t|\mathbf{x}_t\right)p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)}{\pi\left(\mathbf{x}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{y}_{1:t}\right)} \end{aligned}$$

### 4.4.2

The Bootstrap filter algorithm has been implemented.The plot for the estimated filtering distribution $p\left(x_t|y_{1:t}\right)$ on a 2D grid of $x_t \in [-25, 25]$ , $t \in [0, 200]$ and 1000 particles can be seen below.

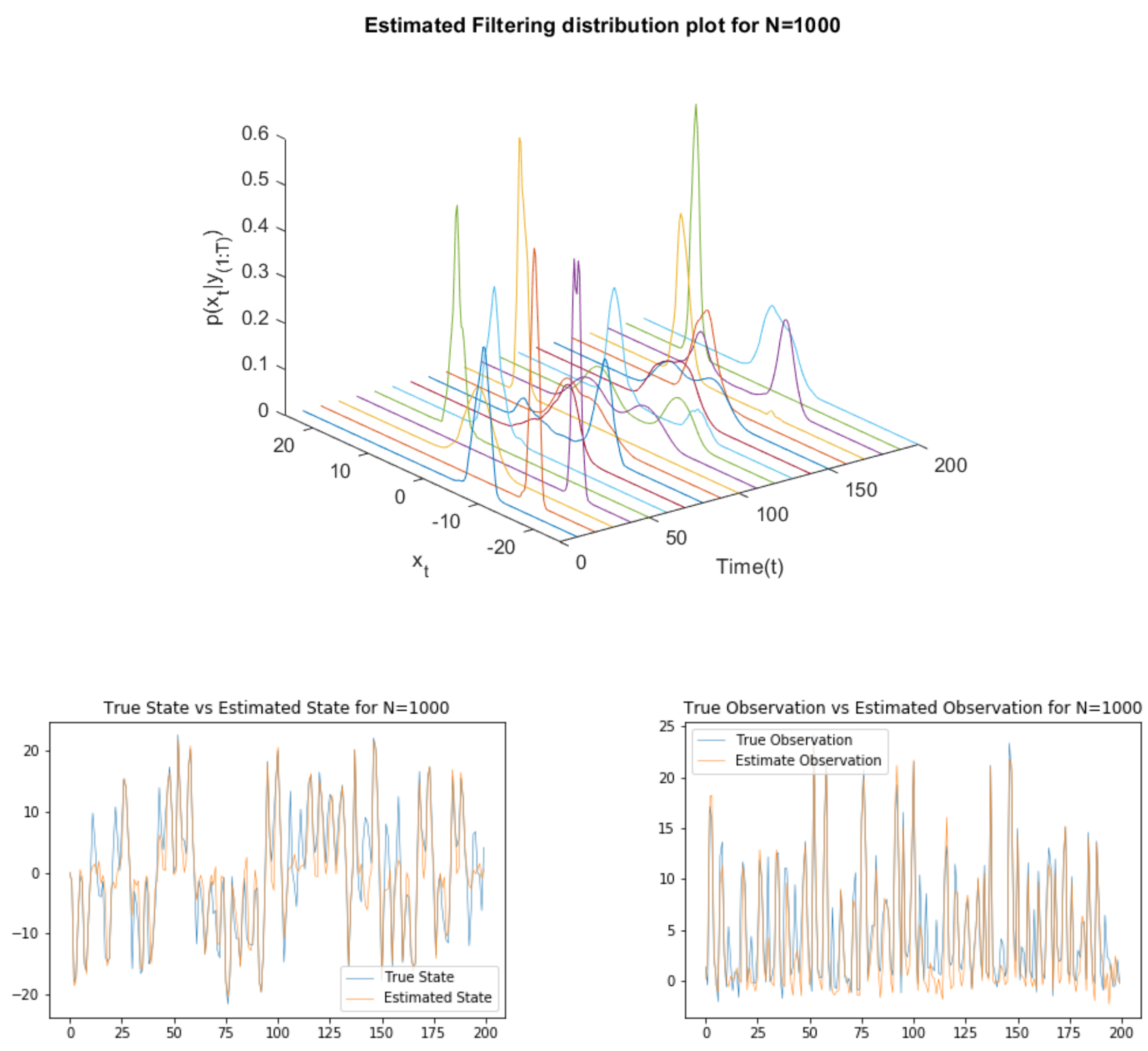

Figure 6: Number of particles = 1000

The algorithm was tested for number of particles, $N \in [100, 500, 1000, 1500, 2000, 2100]$ Multi modality started appearing for number of particles=100 which can be seen in figure 6.

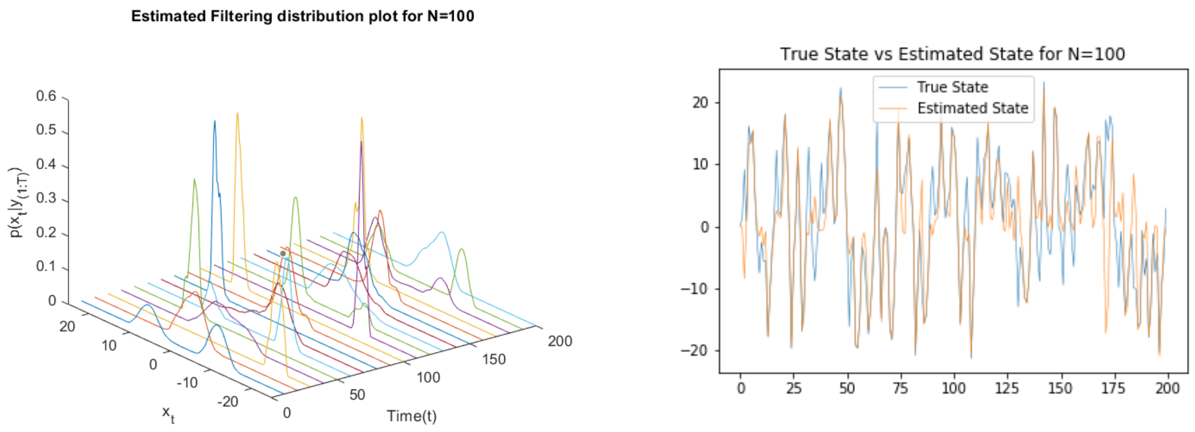

Figure 7: Number of particles = 100

As MCMC is an approximation , with increasing N, number of particles the approximation becomes closer to the true distribution. Theoretically convergence can be achieved with $N \approx \infty$. Of all the number of particles tested, the change in the filtering distribution with N=100 (figure 6) is quite significant when compared to N =1500 (figure 7). The change in the distribution as compared to N=2100 (figure 8) is not that significant.

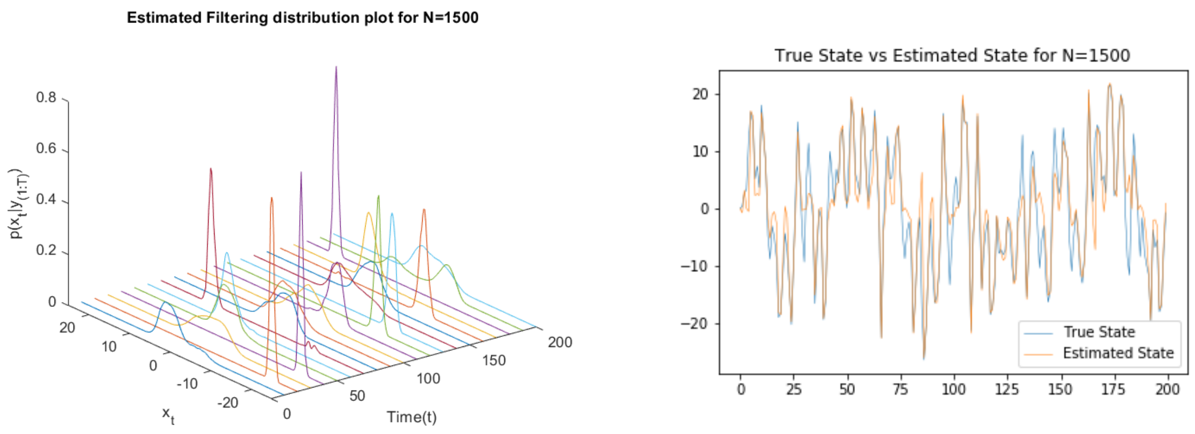

Figure 8: Number of particles = 1500

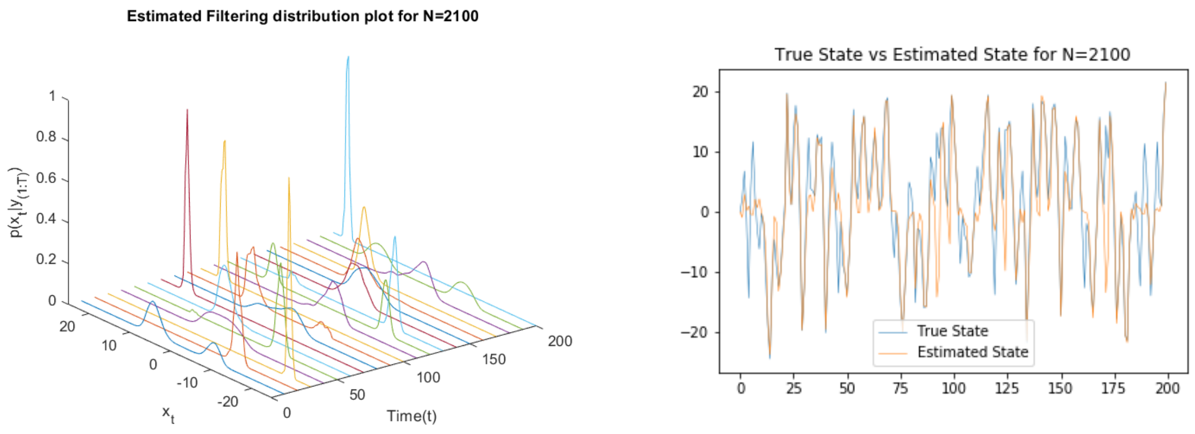

Figure 9: Number of particles = 2100

## References

[1] Andrieu, Christophe et al. "An Introduction to MCMC for Machine Learning." Machine Learning 50 (2003): 5-43.

[2] Arulampalam et. al. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing. 50 (2). p 174–188

[3] https://twiecki.io/blog/2015/11/10/mcmc-sampling/

[4] https://www.hds.utc.fr/~tdenoeux/dokuwiki/_media/en/mcmc_slides.pdf

[5] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SENEGAS/node5.html

[6] https://stats.stackexchange.com/questions/163399/metropolis-hastings-algorithm-prior-vs-proposa
rq=1