Advanced simulation on State space models using MCMC methods

xy921

April 2022

Abstract

This is part of a coursework from my MSc Statistics degree at Imperial College, on the module *MATH70013 Advanced Simulation methods.* The lecturer is Nicolas Kantas, who has given the idea of the whole concept and part of the code written in MATLAB.

Contents

1 Stochastic Volatility Model

In this section we are running a stochastic volatility model

$$X_n = \rho X_{n-1} + \sigma V_n, \quad Y_n = \beta \exp\left(\frac{X_n}{2}\right) W_n$$

where $W_n, V_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1), X_0 \sim \mathcal{N}(0,1)$.
After running the model for $T = 100$, we obtained observation $y_{0:T}$ and store the real trajectory $x^*_{0:T}$ with parameter values $\rho = 0.91, \sigma = 1, \beta = 0.5$.

1.1 Particle Smoothing

1.1.1 Methods and details

In order to approximate the smoothing density we are considering $p(x_n|y_{0:T})$ with (a) a single forward run of a SIR particle filter and (b) a dedicated particle smoothing algorithm, here we use Forward Filtering Backward Sampling (FFBSa).

For smoothing method (b) we use number of sampled paths same as the particle number N in SIR particle filter.

1.1.2 N=50, 500

In this section, We ran two SIR particle filter for $N = 50, 500$ respectively. Then their smoothing and filtering particle approximations for densities at specific times (n=10, 50, 90) will be compared through plots.

Figure 1 shows the histogram for $n = 10, 50$ and smooth kernel density plot for $n = 90$; Figure 2 shows the smooth kernel density plots for all three ns. Due to path degeneracy for SIR particle filters with small particle size ($N = 50$ for example), the particles stuck at one points for early times ($n = 10, 50$) and would be impossible to plot the density. Therefore in Figure 1 histograms are plotted instead.

1.1.3 N=100

In this section, We ran two SIR particle filter for $N = 100$, and compare the particle approximations for filtering/smoothing mean and variance.

Figure 3 shows the filtering mean from SIR, smoothing mean from both SIR and FFBSa, compared with real trajectory x^*_n. While Figure 4 shows the estimated variances for different approaches.

1.1.4 Comment on the results

Comparing filtering with smoothing in general:

- (a) smoothing and (b) smoothing behaved quite differently. (b) smoothing have a similar property on densities, mean and variance to (a) filtering while (a) smoothing have a large difference.
- For later time n smoothing and filtering have similar result for the density. However for earlier n both results for $N = 50, 500$ quite differs. For smoothing (a) due to path degeneracy it couldn't show a full density, especially for a smaller $N = 50$; and the position for smoothing mean quite biased from whatever true value or filter mean.
- The smoothing variance for both methods are in general more fluctuated, especially the (a) smoother. Although in Figure 4 the variance for (a) smoothing and (b) filter quite overlapped, we can still see the green line was slightly above blue line, and the orange line was obviously jumped to 7 at time 80.
- The above could also be found in 3 such that smoother (a) have a much higher variance compared to other two estimations.

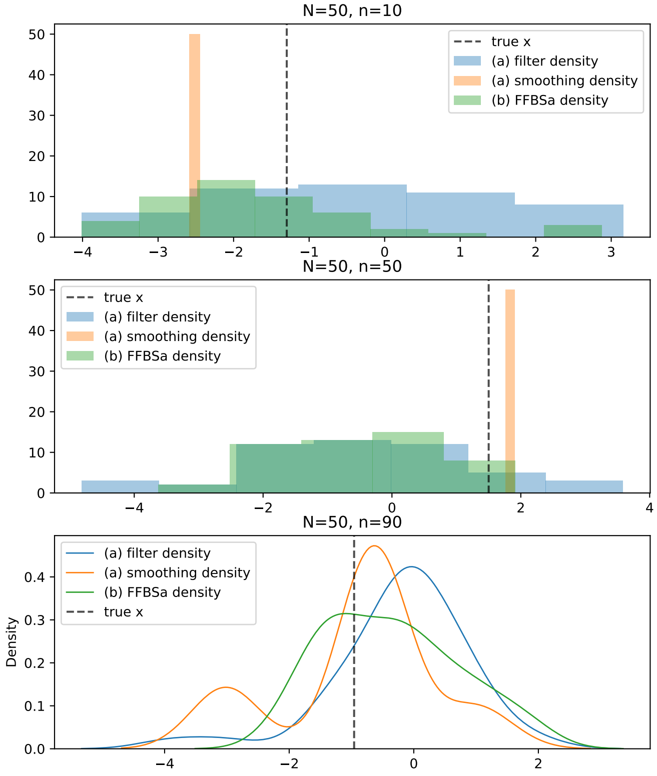

Figure 1: Densities for N=50

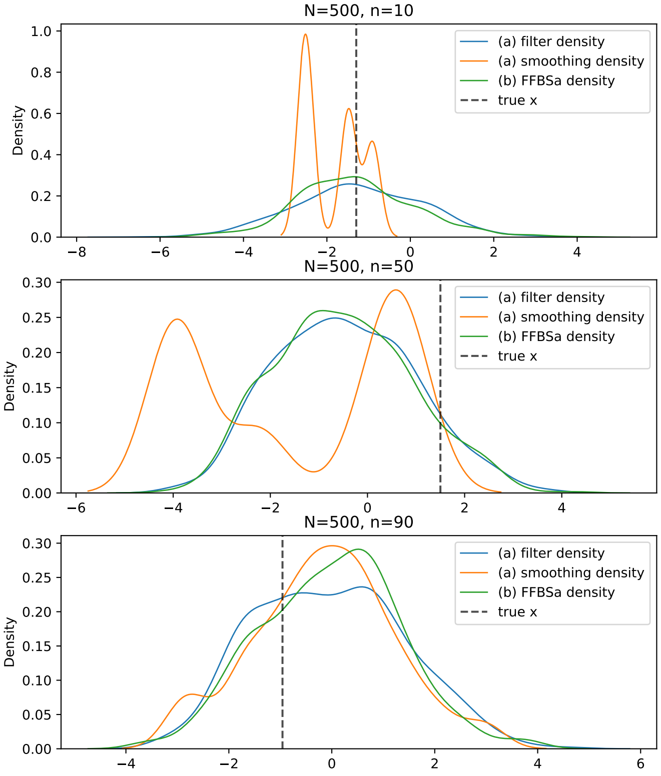

Figure 2: Densities for N=500

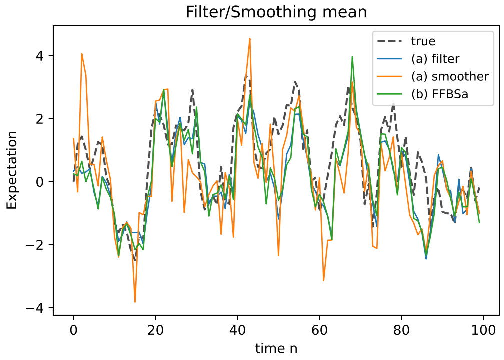

Figure 3: Comparing 3 expectations

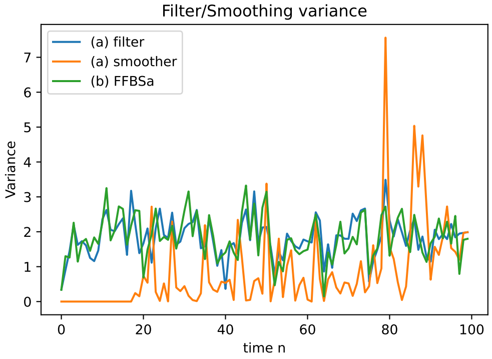

Figure 4: Comparing 3 estimated variances

Comparing smoothing using method (a)SIR particle filtering and (b)FFBSa:

- Particle smoothing using (a) suffers from path degeneracy, especially for small particle size ($N = 50$) or when predicting early stages ($n = 10, 50$). In Figure 1, although all three methods are not having a good prediction, particles at $n = 10, 50$ stayed at one single position while the distribution of particle smoothing using (b) is more spread. Moreover, in Figure 2 the (a) smoothing density have multiple peaks at $n = 10, 50$. This would not be a good method if we want to predict the position of particles at early stages.

- The smoother mean obtained using method (a) is more fluctuated. From Figure 3 we could see that the smoothing mean $E[X_n|X_{0:T}]$ from (b) is close to true value, while the orange line indicating mean from (a) is quite variated. Sometimes it has a higher bias from the true trajectory.

- The particle variances for $p(x_n|y_{0:T})$ is also different. For method (b) the variance is fluctuating around 2 while for method (a) the variance was initially small and increases as time n increases.

1.1.5 Comment on the computational cost

The above comparison would be unfair in terms of computational cost. The FFBSa algorithm would have a much higher computational cost of $\mathcal{O}(N^2T)$ while the SIR particle filter with N particles only have $\mathcal{O}(NT)$. The computational cost would be similar if we implement SIR with N^2 particles. With the same computational cost we could discuss about bias/variance trade-off.

1.2 Likelihood inference

1.2.1 Deriving $\nabla_\theta \log p_\theta\left(y_{::T}\right)$

By Fisher's identity we have

$$\nabla_\theta \log p_\theta\left(y_{0:T}\right) = \int \nabla_\theta \log p_\theta\left(x_{0:T}, y_{0:T}\right) p_\theta\left(x_{0:T} \mid y_{0:T}\right) dx_{0:T}$$

And we could write

$$\begin{aligned}
\nabla_\theta \log p_\theta\left(x_{0:T}, Y_{0:T}\right) &= \nabla_\theta \log \prod_{p=0}^{T} f_\theta\left(x_p \mid x_{p-1}\right) g_\theta\left(y_p \mid x_p\right) \\
&= \sum_{p=0}^{T} \left[\nabla_p \log f_\theta\left(x_p \mid x_{p-1}\right) + \nabla_\theta \log g_\theta\left(y_p \mid x_p\right)\right]
\end{aligned}$$

$$\begin{aligned}
&\text{Let } s_p\left(x_{p-1}, x_p\right) = \nabla_\theta \log f_\theta\left(x_p \mid x_{p-1}\right) + \nabla_\theta \log g_\theta\left(y_p \mid x_p\right) \\
\Rightarrow \nabla_\theta \log p_\theta\left(y_{n:T}\right) &= \int \sum_{p=0}^{T} s_p\left(x_{p-1}, x_p\right) p_\theta\left(x_{0:T} \mid Y_{0:T}\right) dx_{x_{0:T}} \\
&= \mathbb{E}\left[\sum_{p=0}^{T} s_p\left(x_{p-1}, x_p\right)\right]
\end{aligned}$$

1.2.2 Gradient ascent on ρ

Now we assume only ρ is unknown and we want to inference ρ. By equation (100) in the lecture note we could have

$$\rho_{k+1} = \rho_k + \gamma \nabla_\rho l_T(\rho)\big|_{\rho=\rho_k}$$

From the expression in last part, we could derive an expression for the gradient ascent as follows:

$$\begin{aligned}
s_p\left(x_{p-1}, x_p\right) &= \nabla_\rho \log f_\theta\left(x_p \mid x_{p-1}\right) + \nabla_\rho \log g_\theta\left(y_p \mid x_p\right) \\
&= \frac{\partial}{\partial \rho}\left[-\frac{1}{2}\log(2\pi) - \log\sigma - \frac{1}{2\sigma^2}\left(x_p - \rho x_{p-1}\right)^2\right] \\
&= \frac{x_{p-1}}{\sigma^2}\left(x_p - \rho x_{p-1}\right) + 0
\end{aligned}$$

$$\Rightarrow \nabla_\theta l_T(\theta)\big|_{\theta=\theta_k} = \nabla_\rho l_T(\rho)\big|_{\rho=\rho_k} = \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{T}\frac{x_{p-1}^i}{\sigma^2}\left(x_p^i - \rho_k x_{p-1}^i\right)$$

$$\Rightarrow \rho_{k+1} = \rho_k + \gamma \cdot \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{T}\frac{x_{p-1}^i}{\sigma^2}\left(x_p^i - \rho_k x_{p-1}^i\right)$$

By implementing above updating rule for the gradient ascent, we got the plot of parameter value against iteration number k in Figure 5. We initialized ρ from 3 different values: $0.5, -1.1.5$ and run for 20 iterations.

The constant step size was set to be small difference $\gamma = 0.001$. This is because for each iteration the gradient would be large as the average of recursive sum for total running time $T = 100$. Therefore we should choose step size proportional to $\frac{1}{T} = 0.01$ to avoid the step being too large. Here using this step size we have a good convergence speed to the true value, and all 3 initialization converges after 3 iterations.

The value of convergence for 3 initialization is close. However they all have bias higher that 0.05 above the true value.

1.2.3 Grid Method

It is also suggested to use grid method to estimate the gradient $\nabla_\theta \log p_\theta\left(y_{0:T}\right)$. It could be done by simply first calculate the log-likelihood value at each grid, and then calculating their differences as an estimation of the gradient.

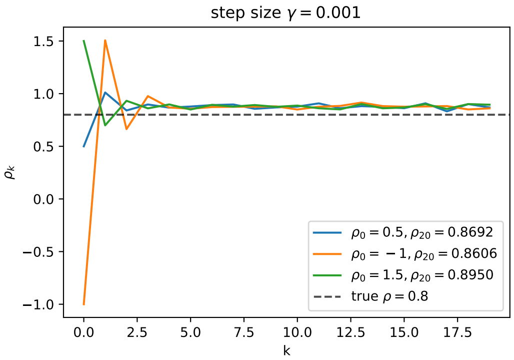

Figure 5: Values of ρ from 3 different initial points converges around true value 0.8

However since our aim was to find the MLE of ρ, it would be unnecessary to calculate the gradient. Simply comparing the log-likelihood using grid methods will do. Even though to have a good approximation to the parameter values, here the parameter converges very quickly. It may not be efficient as gradient ascent in this problem.

2 Parameter inference for the Linear Gaussian Model

The linear Gaussian HMM is

$$X_n = \rho X_{n-1} + \tau V_n, \quad Y_n = X_n + \sigma W_n$$

where $W_n, V_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1), X_0 \sim \mathcal{N}(0,1)$.
After running the model for $T = 100$, we first obtained the observation $y_{0:T}$ and real trajectory $x^*_{0:T}$ with parameter $\theta = (\rho, \tau, \sigma) = (0.8, 1, 0.5)$ and we want to inference for $p\left(\rho, \tau^2 \mid y_{0:T}\right)$ assuming $\sigma = 0.5$ is known. We suppose the priors are $\rho \sim \mathcal{U}[-1,1]$ and $\tau^2 \sim \mathcal{IG}(1,1)$.

2.1 Ideal Marginal Metropolis Hastings targeting $p(\rho, \tau^2 | y_{0:T})$

2.1.1 Algorithm

For the details of algorithm we have

- Total iteration number 20000;
- Initialize the parameters $(\rho_0, \tau_0^2) = (0.2, 0.1)$;
- Normal random walk proposal q for ρ, τ^2 with step size $(0.2, 0.1)$ for each parameter respectively;
- Kalman filter used to estimate recursive likelihood $\hat{p}_\theta(y_{0:T})$;

Algorithm 1 Ideal Marginal Metropolis Hastings

```
At iteration $k = 0$:
  Initialize the variable $\theta_0 = (\rho_0, \tau_0^2, \sigma)$
  Run the Kalman Filter to compute $\hat{p}_{\theta_0}(y_{0:T})$ using parameters $\theta_0$
At iteration $k \geq 1$:
  Sample a proposal $\theta' \sim q(\theta \mid \theta_{k-1})$
  Run the Kalman Filter to compute $\hat{p}_{\theta'}\left(y_{0:T}\right)$
  Set $\theta_k = \theta', X_{0:T} = X'_{0:T}$, and $\hat{p}_{\theta_k}\left(y_{0:T}\right) = \hat{p}_{\theta'}\left(y_{0:T}\right)$ with probability
      $$\alpha = 1 \wedge \frac{\hat{p}_{\theta'}\left(y_{0:T}\right) p\left(\theta'\right) q\left(\theta_{k-1} \mid \theta'\right)}{\hat{p}_{\theta_{k-1}}\left(y_{0:T}\right) p(\theta_{k-1}) q\left(\theta' \mid \theta_{k-1}\right)},$$
  otherwise set $\theta_k = \theta_{k-1}, X_{0:T} = X_{0:T}$, and $\hat{p}_{\theta_k}\left(y_{0:T}\right) = \hat{p}_{\theta_{k-1}}\left(y_{0:T}\right)$.
```

2.1.2 Results

Histograms plotting the density of $p(\rho|y_{0:T})$ and $p(\tau^2|y_{0:T})$ could be found at Figure 6. Both histograms have a stable mean to original values but small bias still appears, especially for τ^2 (0.858 vs 1.0).

2.1.3 MCMC diagnostics

Here we introduced the average acceptance ratio (Figure 7) and the trace plot (Figure 8) as MCMC diagnostics. It could be found that the average acceptance ratio is stable around 0.4 after 2500 iteration. The trace of ρ fluctuate around true value with MCMC variance less than 0.5, while traces of τ^2 was fluctuating slightly below the true value, and the variance seems higher through out the 20000 iterations.

2.2 PMMH targeting $p(\rho, \tau^2 | y_{0:T})$

We are using the same MCMC algorithm design as written in section 3.1.1 except for estimate recursive likelihood $\hat{p}_\theta(y_{0:T})$, where we are using Sequential Monte Carlo methods. To be more specific, we used SIR particle filter to sample $X_{0:T}$.

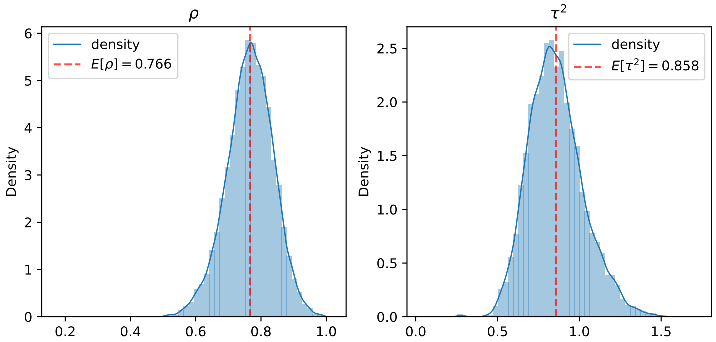

Figure 6: Histograms for ρ, τ^2 after 20000 iterations using ideal MMH

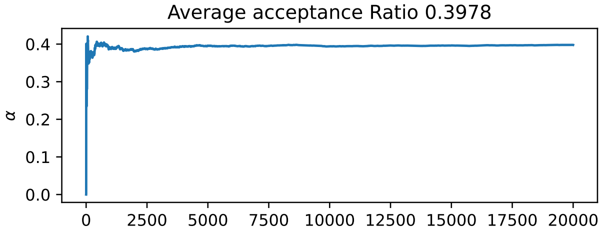

Figure 7: Acceptance ratio converges to around 0.4 after 20000 iterations

(a) MCMC results and diagnostics

The resulting histogram for ρ, τ^2 could be found in Figure 9 as well as the resulting diagnostics, average acceptance ratios (Figure 10) and trace plots (Figure 11).

For ρ the PMMH algorithm produces a density close to the one from ideal MMH, and the expectation is quite close. For τ^2 the peak has small bias but their mean is still close. Comparing the histogram themselves, ideal MMH seems to be more smooth.

Although the histogram produces similar posterior densities, the acceptance ratio with same MCMC design was quite low. The final acceptance rate decreases before first 2500 iterations and was below 0.2 at the end. The trace plot also indicates a worse MCMC performance than the MCMC chain from ideal MMH.

The low acceptance rate could be due to a wide step size inherited from last section (0.2 for ρ and 0.1 for τ^2). In fact another run with step size of (0.01, 0.01) was used and the acceptance rate was improved to around 0.2.

(b) N selection

In the particle filter we mentioned in the 2, we have chosen $N = 150$ as the number of particles.

As written in the section 7.1.3 of Lecture Notes, it is suggested that ”the variance of acceptance rate is proportional to T/N” and we hope to choose N such that Monte Carlo variance for $\log p(\theta|y_{0:T})$ is near 1. Moreover, the lecturer mentioned in lecture note section 7.1.4 that for a simulation with $T = 500$ we reached optimal average acceptance rates at $N = 500, 1000$.

Therefore with the above results from other experiments, we could use $N = 100 = T$ as an appropriate particle number. However from the simulation the acceptance rate is quite low (around 0.11). It is also mentioned in lecture note section 7.1.4 that from $N = 500$ to $N = 1000$ there is an improvement for the average acceptance rate. Therefore in sense of a trade-off between computational cost and the MCMC performance we have chosen $N = 150$ to reached a slightly higher $\alpha = 0.16$.

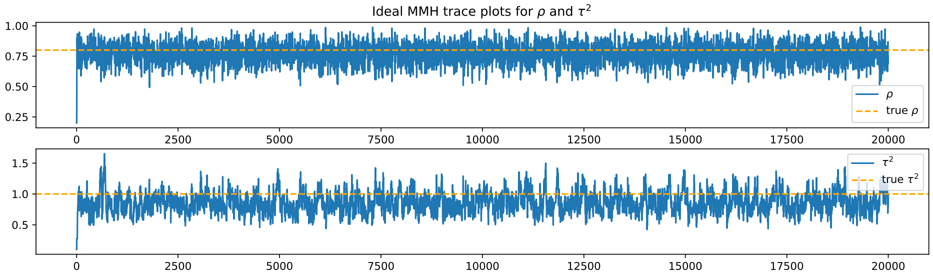

Figure 8: ρ, τ^2 converges very quick to value around their true values

Algorithm 2 Particle Marginal Metropolis Hastings

```
At iteration $k = 0$:
  1: Initialize the variable $\theta_0 = (\rho_0, \tau_0^2, \sigma)$
  2: Run the SIR algorithm $p\left(x_{0:T} \mid y_{0:T}, \theta_0\right)$, sample $X_{0:T} \sim \widehat{p}\left(\cdot \mid y_{0:T}, \theta_0\right)$, and compute estimate $\hat{p}_{\theta_0}(y_{0:T})$
At iteration $k \geq 1$:
  1: Sample a proposal $\theta' \sim q(\theta \mid \theta_{k-1})$
  2: Run the SIR algorithm targeting $p\left(x_{0:T} \mid y_{0:T}, \theta'\right)$, sample $X'_{0:T} \sim \widehat{p}\left(\cdot \mid y_{0:T}, \theta'\right)$, and compute estimate $\hat{p}_{\theta'}\left(y_{0:T}\right)$
  3: Set $\theta_k = \theta', X_{0:T} = X'_{0:T}$, and $\hat{p}_{\theta_k}\left(y_{0:T}\right) = \hat{p}_{\theta'}\left(y_{0:T}\right)$ with probability
       $\alpha = 1 \wedge \frac{\hat{p}_{\theta'}\left(y_{0:T}\right) p\left(\theta'\right) q\left(\theta_{k-1} \mid \theta'\right)}{\hat{p}_{\theta_{k-1}}\left(y_{0:T}\right) p(\theta_{k-1}) q\left(\theta' \mid \theta_{k-1}\right)}$,
  4: otherwise set $\theta_k = \theta_{k-1}, X_{0:T} = X_{0:T}$, and $\hat{p}_{\theta_k}\left(y_{0:T}\right) = \hat{p}_{\theta_{k-1}}\left(y_{0:T}\right)$.
```

(c) Using PMMH to estimate $X_{0:T}$ from $y_{0:T}$

The PMMH algorithm produced a chain of targeted parameters posterior estimations. Therefore given observations $y_{0:T}$ we could first use the PMMH to inference unknown parameters ρ, τ^2 and get the parameter estimate by taking the mean of each Markov Chain.

With the resulting estimation we could run SIR particle filter or Kalman Filter to obtain an estimation on the hidden states.

(d) Estimating $X_{0:T}$

We are now having two estimation of $X_{0:T}$ as shown in Figure 12. One from Kalman Filter with parameters $\hat{\rho} = 0.776, \hat{\tau}^2 = 0.8441$, and another from Particle Filter using same SIR algorithm implemented in PMMH algorithm (Algorithm 2), with $(N = 150)$.

It could be found that SIR Particle filter have a better estimation to the original trajectory, and obtained a thinner 95% credible intervals.

2.3 Compare the result

We have already concluded some results in above sections 3.2(a). We could have a summary here:

- The histograms shows two MCMC has similar resulting posterior densities for the two parameters;
- The acceptance rate and trace plots shows that MCMC performances for the ideal MMH is better. But this could be due to a wide random walk proposal step size for PMMH. From that we could see the optimal variance for the random walk proposal is smaller for PMMH.
- Around iteration 15000 there is a relatively long stay at one state for two parameters, which have not apperaed in ideal MMH algorithm traces.

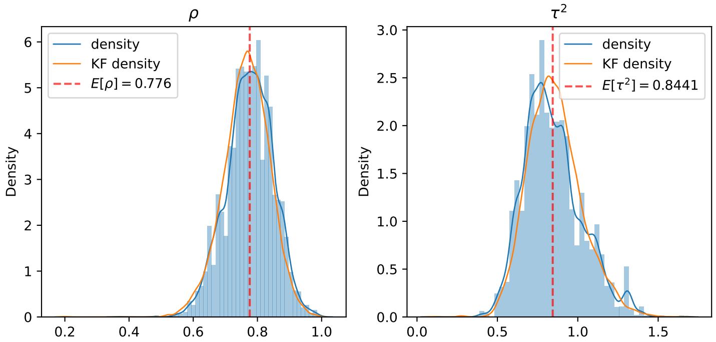

Figure 9: Histograms for ρ, τ^2 after 20000 iterations using PMMH

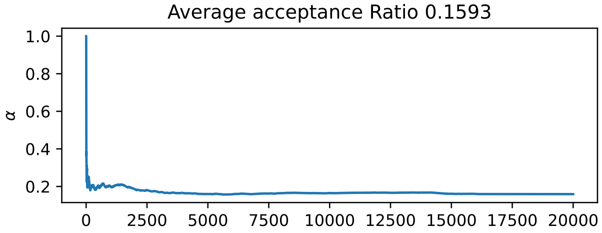

Figure 10: Acceptance ratio converges to around 0.16 after 20000 iterations

Moreover, the two credible interval estimation for $X_{0:T}$ suggest that particle filter produces a smaller variance estimation. This could be the reason why we require a smaller step size for the random walk proposal of PMMH algorithm.

This indicates that a small change on the parameter θ would cause a larger change in the estimated marginal likelihood $\hat{p}_\theta(y_{0:T})$. The marginal likelihood for ideal MMH is obtained from recursive mean/variance of $X_{0:T}$, while that for PMMH the likelihood obtained from the particle filter is a product of series of conditional probabilities such that

$$\hat{p}_\theta(y_{0:n}) = \prod_{k=0}^{n} \hat{p}_\theta(y_k|y_{0:k-1}) = \frac{1}{N}\sum_{i=1}^{N} \omega_k \left(X_{k-1:k}^i\right)$$

where $\omega_k \left(X_{k-1:k}^i\right)$ is the un-normalized weight computed at each time k. This might introduce more variability to the final result.

2.4 Particle Filter on the extended state (X_n, θ_n)

To implement the particle filtering targeting $p\left(\rho, \tau^2, x_{0:n} \mid y_{0:n}\right)$ using artifical dynamics approach $\theta_n = \theta_{n-1} + \epsilon_n$, where ϵ_n is a zero mean noise with small variance.

(a) Marginal densities

The 95% credible intervals and the median against time n could be found at Figure 13.

From particle filtering we have produced a 2-d array for each parameter with $N \times T$, that each column consists of N particles at time $n = 0, 1, ...T$. Let denote the n^{th} column as $\{\rho_n^i\}_{i=1}^N$ and $\{{\tau^2}_n^i\}_{i=1}^N$.

Then at each time n, the median for ρ at time n is computed as $\text{med}_n = \text{median}(\{\rho_n^i\}_{i=1}^N)$, and the 95% credible interval is computed as 0.025 and 0.975 quantile of each column $\{\rho_n^i\}_{i=1}^N$. The median/95% CI for τ^2 could be calculated in a similar way.

It could be found that the credible interval was quite wide in the beginning, and gradually decreases as time increases. This is because we let the initial particles to be randomly sampled from their prior (uniform

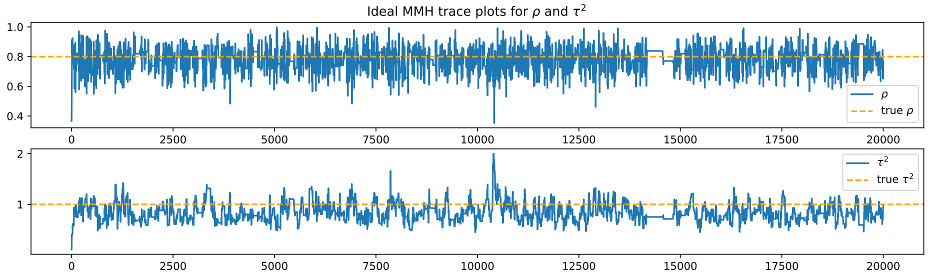

Figure 11: ρ, τ^2 converges to value around their true values

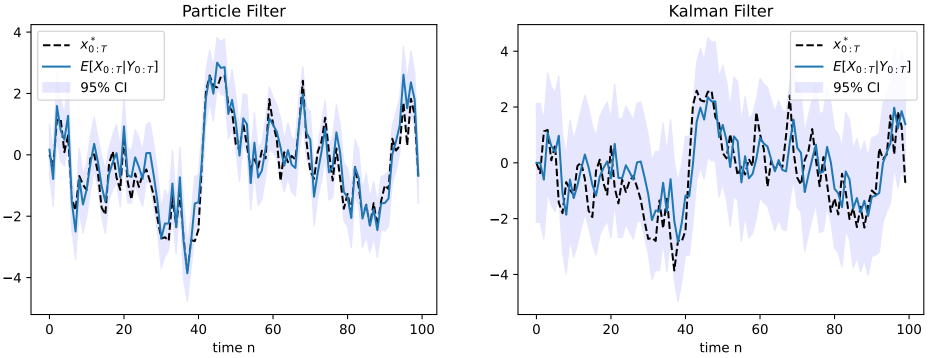

Figure 12: Estimation of $X_{0:T}$ from two filters

and inverse gamma respectively) and the variance would be high. As the filtering goes with time, resample of parameter particles makes them converge to true values.

(b) Estimation of $X_{0:T}$ against time

We could see the plot of $X_{0:T}$ against time in 14. Here the plots are quite close as we are using parameter estimations that both close to their true values. Therefore the credible interval are also close.

2.5 Comparing 3.2 and 3.4

I would have more faith in PMMH methods because in practice the method in 3.4 is not stable. It happens sometimes that parameters converge to value differ from their true values, or introduce a really wide confidence interval even till the end.

In the lecture notes it was also mentioned that when using artificial dynamics to update θ they ”have a bias that is hard to quantify in general”, but easy to implement and produces reasonable results. In this experiment we are having a good result with small bias but it was not always the case. Even through it have a much cheaper cost (obviously because only $\mathcal{O}(NT)$ for single SIR particle filter was required) compared with PMMH (which requires multiple runs of SIR), we could still endure the time consumption of getting a better approximation when the total time $T = 100$ is low.

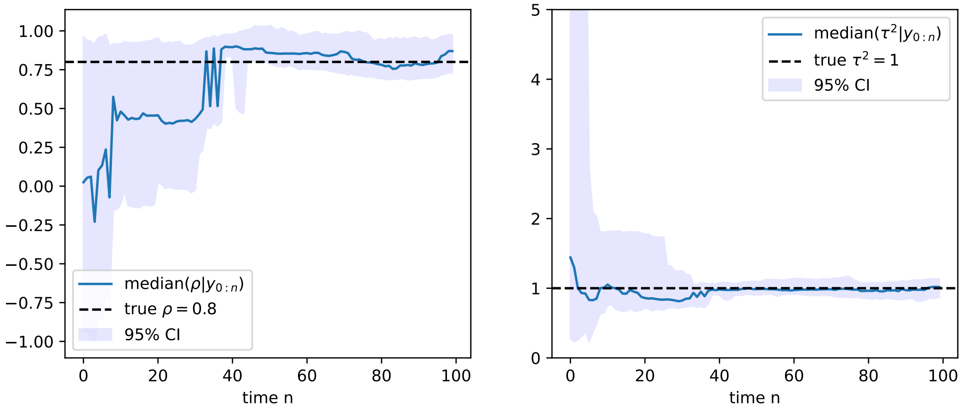

Figure 13: Estimation of $X_{0:T}$ from two filters

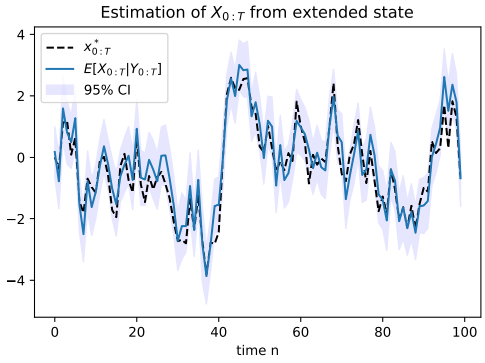

Figure 14: Estimation of $X_{0:T}$ from PF on the extended state