ECE 594E: Homework 5

Steven Munn

Wednesday, May 27, 2015

1 Particle Filter for non-linear system

1.1

We plot two sample trajectories in figures 1 and 2.

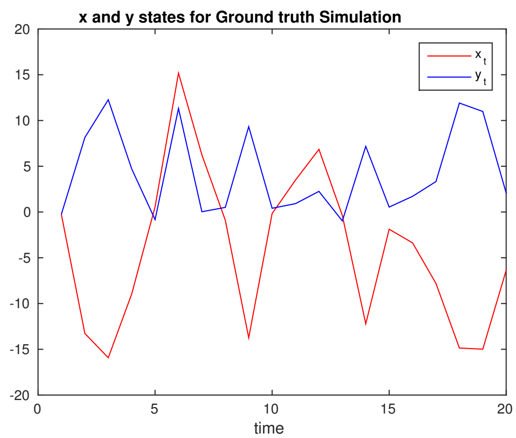

Figure 1: First sample trajectory

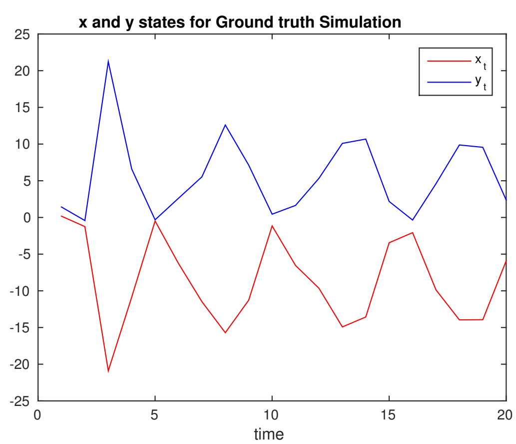

Figure 2: Second sample trajectory

1.2

The full implementation code can be found at `https://github.com/stevenjlm/ML-code/tree/master/bootstrap`

1.3

For $N = 100$ particles, using measurements y_t, $t = 1$ to $t = T = 100$ we plot the conditional mean and the actual state in figure 3. The mean square error between estimate and actual state is in figure 4.

In figures 5, 6, and 7 we plot $p(x_t|y_{1:t})$ at times 10, 50, and 100. Only the plot at time 100 is bimodal, this mostly coincidence though. You can see in figure 6 that the peak is very close to zero. The distribution is a mixture of two Gaussians, it's just that their means are too close to produce distinguishable peaks. And, for figure 5 the value is so extremely low that the state must have been in a downward slope already and the step function didn't produce particles at the other mode peak.

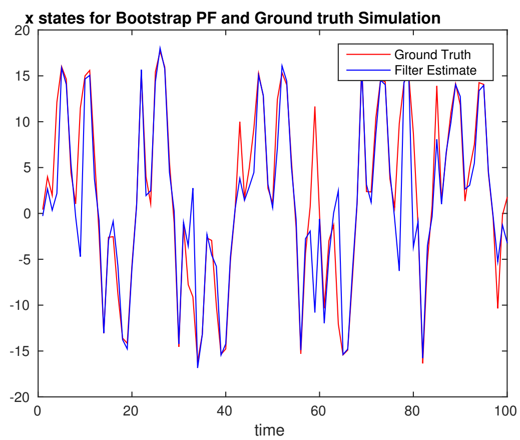

Figure 3: Particle Filter estimate and actual state

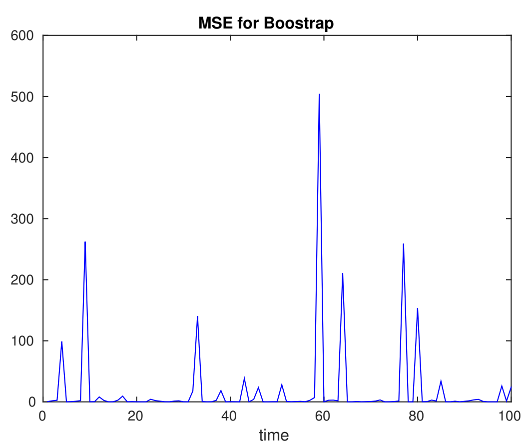

Figure 4: Mean square error of estimate

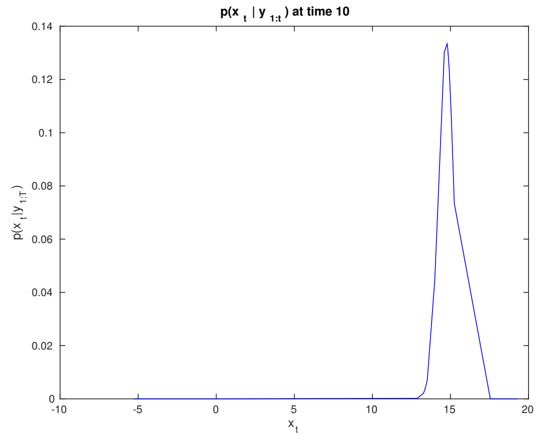

Figure 5: Filtering density at $t = 10$ and $N = 100$

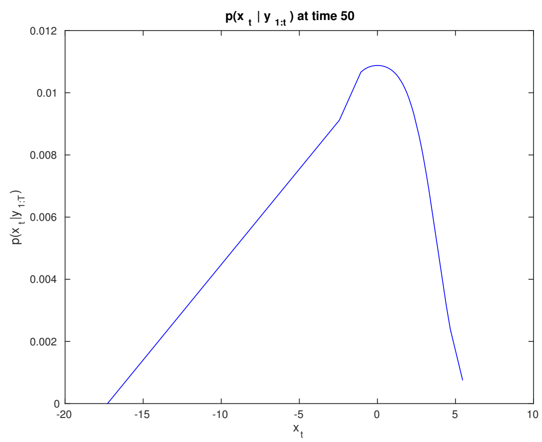

Figure 6: Filtering density at $t = 50$ and $N = 100$

1.4

For $N = 10$ we have the same plots as above in figures 8, 8, 9, 10, 11, and 12.

For $N = 1000$ we have the same plots as above in figures 13, 13, 14, 15, 16, and 17.

1.5

Backward filter is implemented in the same runme.m file as the bootstrap filter. Figure 19 shows the actual state and some sample backward trajectories.

1.6

The mean square error over the average of forward filtered particles is 0.3012 and the backward filter gets 0.2561.

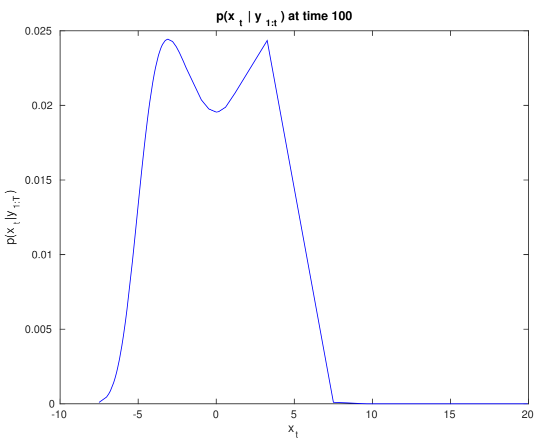

Figure 7: Filtering density at $t = 100$ and $N = 100$

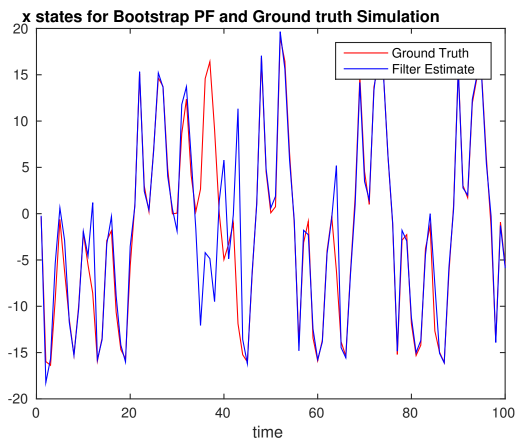

Figure 8: Particle Filter estimate and actual state

1.7

With $N = 10$ backward particles, the MSE is hardly any better after backward filtering. With $N = 100$ particles however, the MSE is significantly better since it goes down nearly one third.

2 Kalman Filtering

2.1

The full implementation code can be found at `https://github.com/stevenjlm/ML-code/tree/master/kalman`

2.2

Figures 19, 20, 21, and 22 show the various mean square errors for different values of N. $N = 5000$ seems to be the point at which bootstrap becomes more acceptable; however, the Kalman filter is always the best.

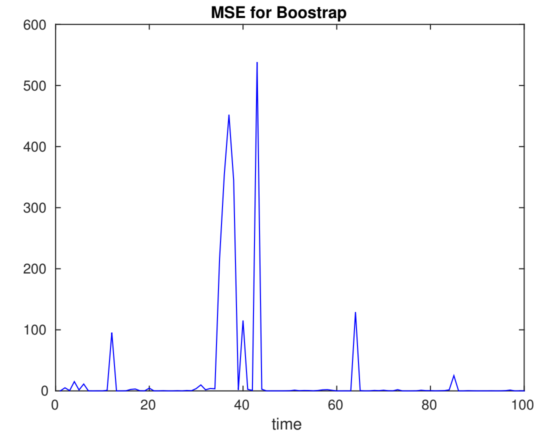

Figure 9: Mean square error of estimate

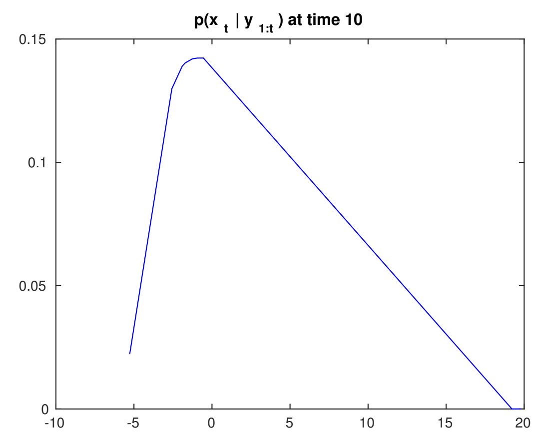

Figure 10: Filtering density at $t = 10$ and $N = 10$

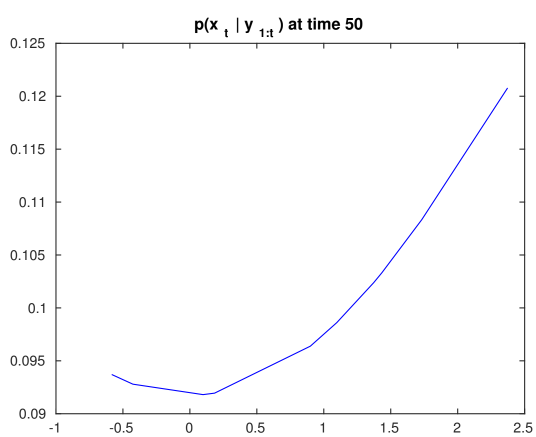

Figure 11: Filtering density at $t = 50$ and $N = 10$

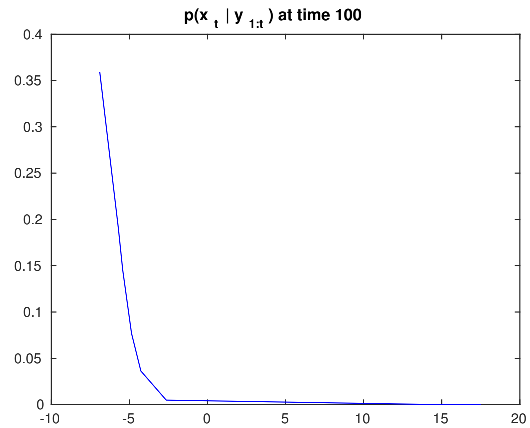

Figure 12: Filtering density at $t = 100$ and $N = 10$

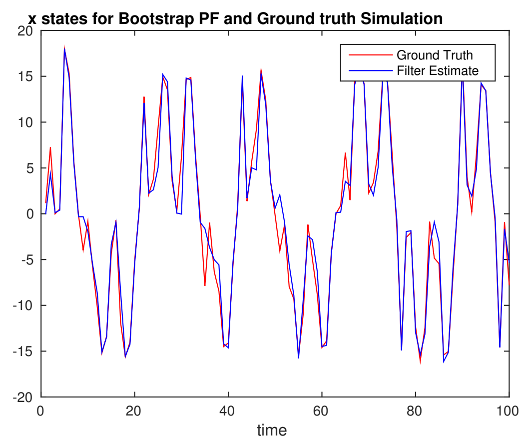

Figure 13: Particle Filter estimate and actual state

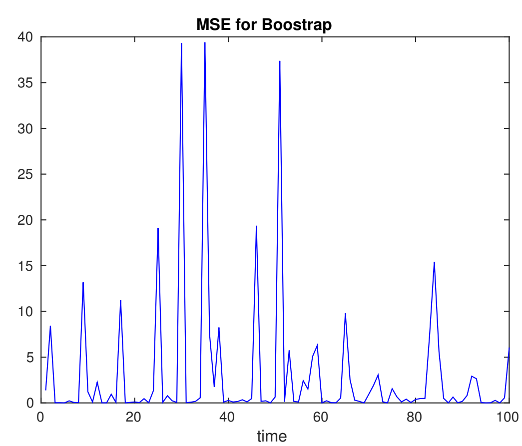

Figure 14: Mean square error of estimate

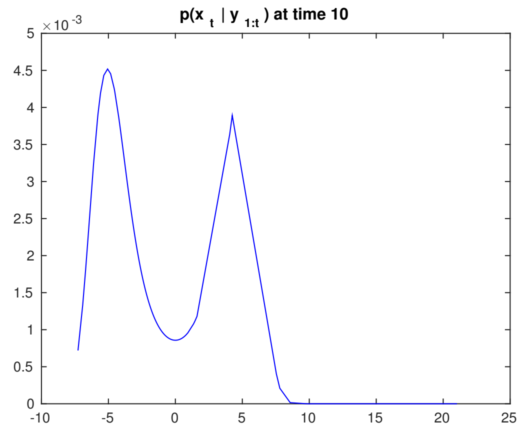

Figure 15: Filtering density at $t = 10$

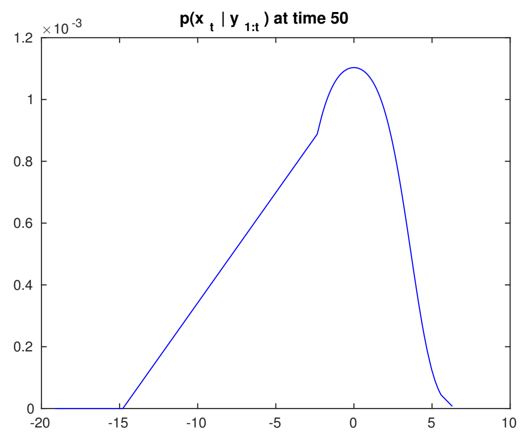

Figure 16: Filtering density at $t = 50$

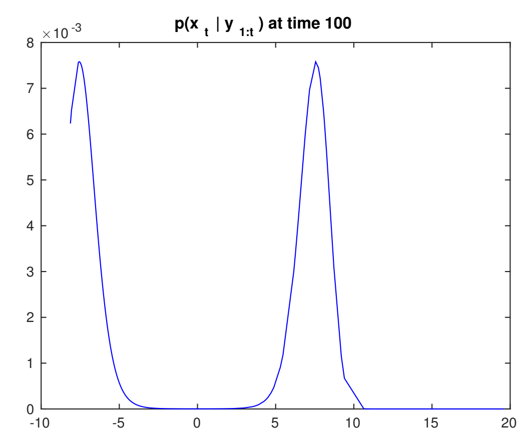

Figure 17: Filtering density at $t = 100$

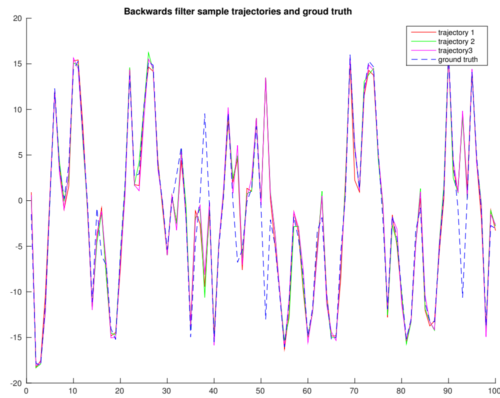

Figure 18: Backwards Filtering sample trajectories and actual trajectory.

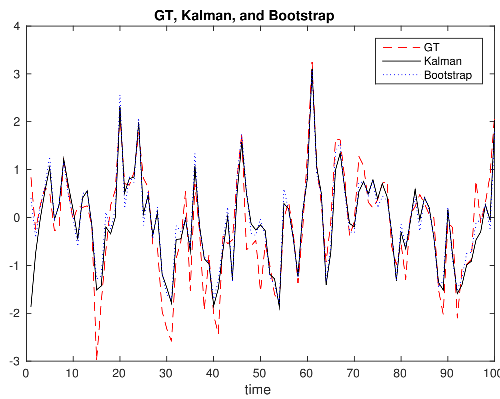

Figure 19: Sample trajectory ground truth, Kalman filter trajectory, and Bootstrap particle filter mean trajectories

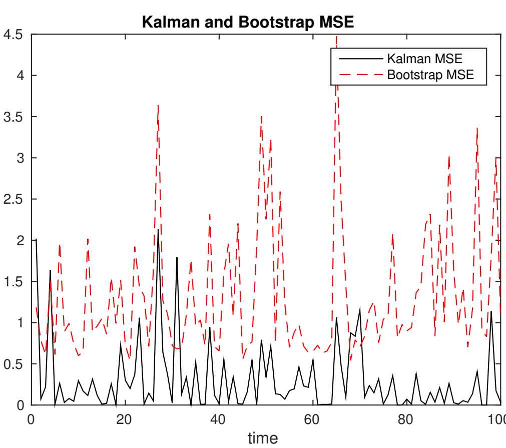

Figure 20: Mean square error for $N = 100$

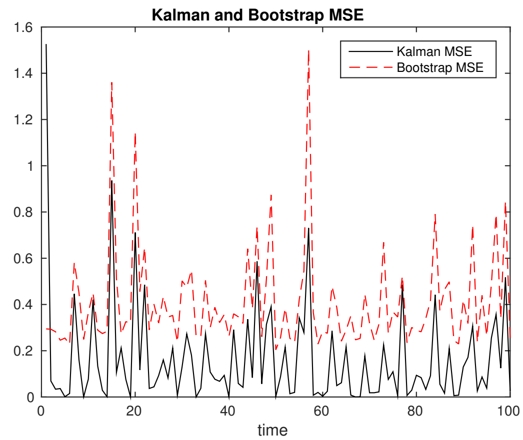

Figure 21: Mean square error for $N = 1000$

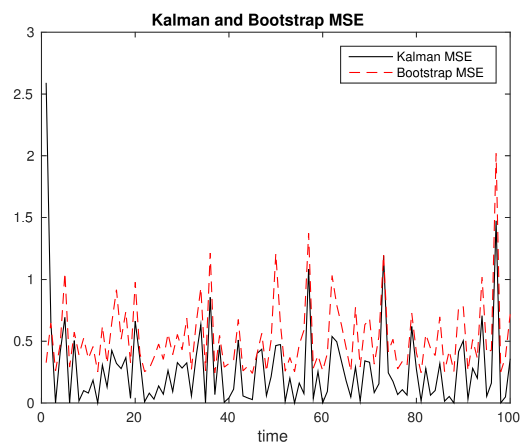

Figure 22: Mean square error for $N = 5000$