Solution to analysis in Home Assignment 4

Vivien Lacorre (lacorre)

May 13, 2024

Analysis

In this report I will present my independent analysis of the questions related to home assignment 4. I swear that the analysis written here are my own.

1 Smoothing

(a) Figure 1.1 shows the smoothed filtered trajectory with covariance plus measurements. As expected the smoothed trajectory is less susceptible to bad measurements and has lower uncertainty.

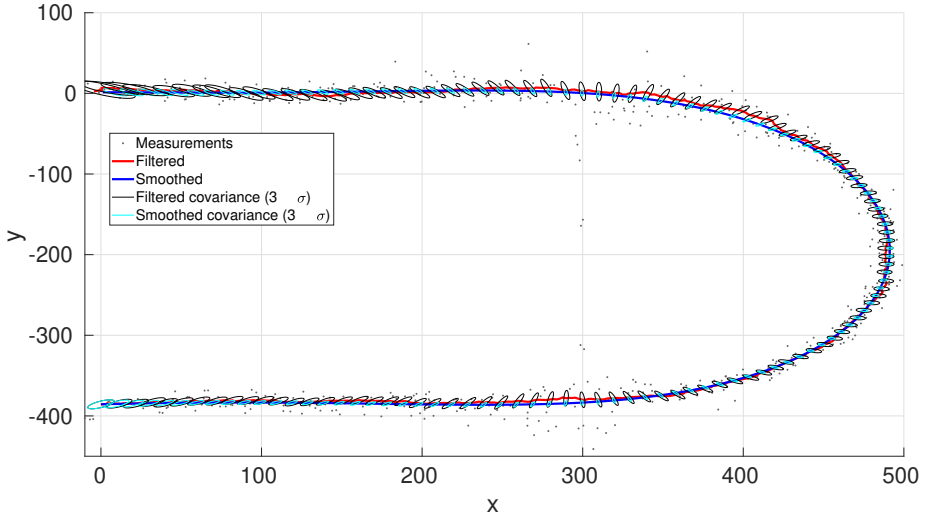

Figure 1.1: Filtered and smoothed trajectory

(b) Figure 1.2 shows that the smoothed trajectory is robust to outliers, as opposed to the filtered one. The circled measurements causes a noticeable drift, which can be fixed by smoothing.

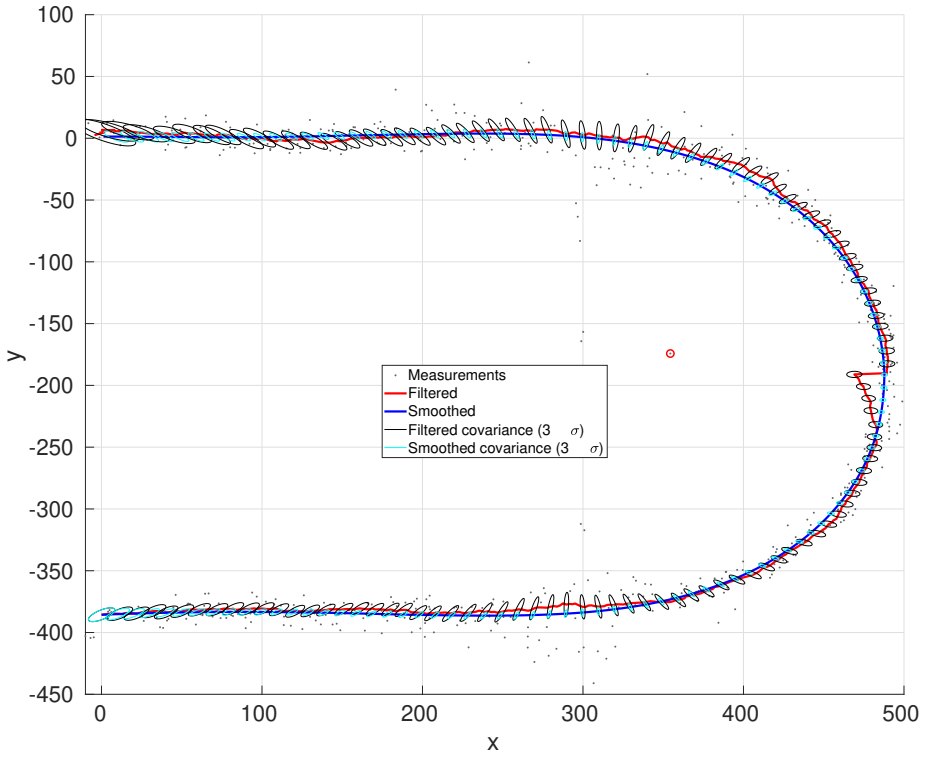

Figure 1.2: Impact of an outlier on the filtered and smoothed trajectories

2 Particle filters for linear/Gaussian systems

(a) Table 2 shows that 1000 particles is enough to reach a good MSE with resampling. 100 on the other hand is not good.

Table 1: Mean Square Error

N	Method	MSE
100	KF	1.4548
100	PF without resampling	2.0148
100	PF with resampling	1.6484
1000	KF	1.4548
1000	PF without resampling	2.0030
1000	PF with resampling	1.4847

Resampling avoids posteriors with really wrong values, as seen on Figure 2.1. One can also see that the PF yields good results, really close to the optimal KF, both in terms of mean and covariance.

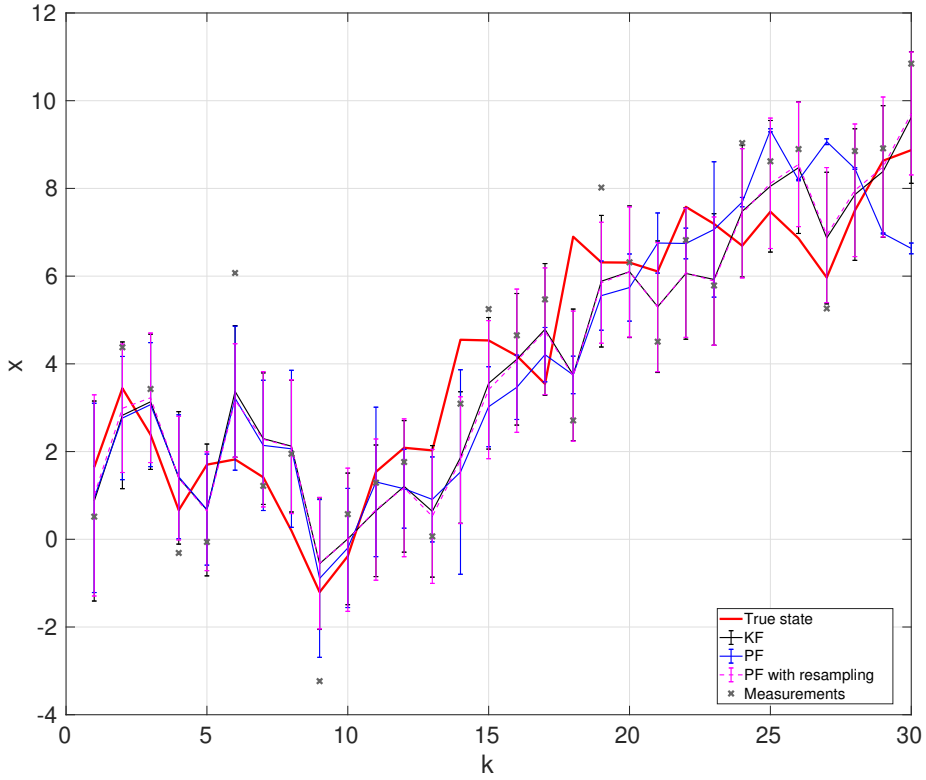

Figure 2.1: PF with resampling

Let us have a look at the evolution of the posterior distribution. Figures 2.2, 2.3 and 2.4 uses a superposition of Gaussian kernels to approximate a Gaussian distribution of the posterior. As one can see, as time goes by, the system becomes more and more confident about the position.

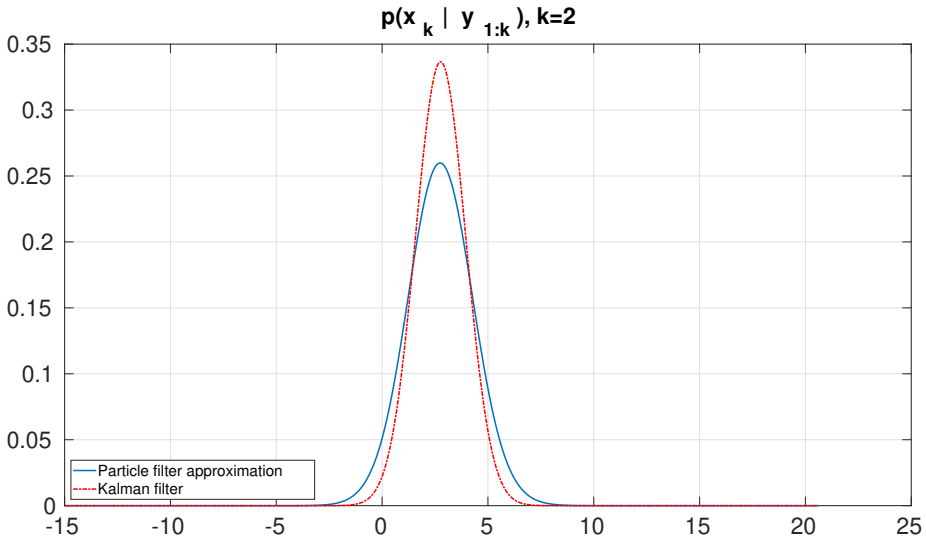

Figure 2.2: Posterior density at k=2

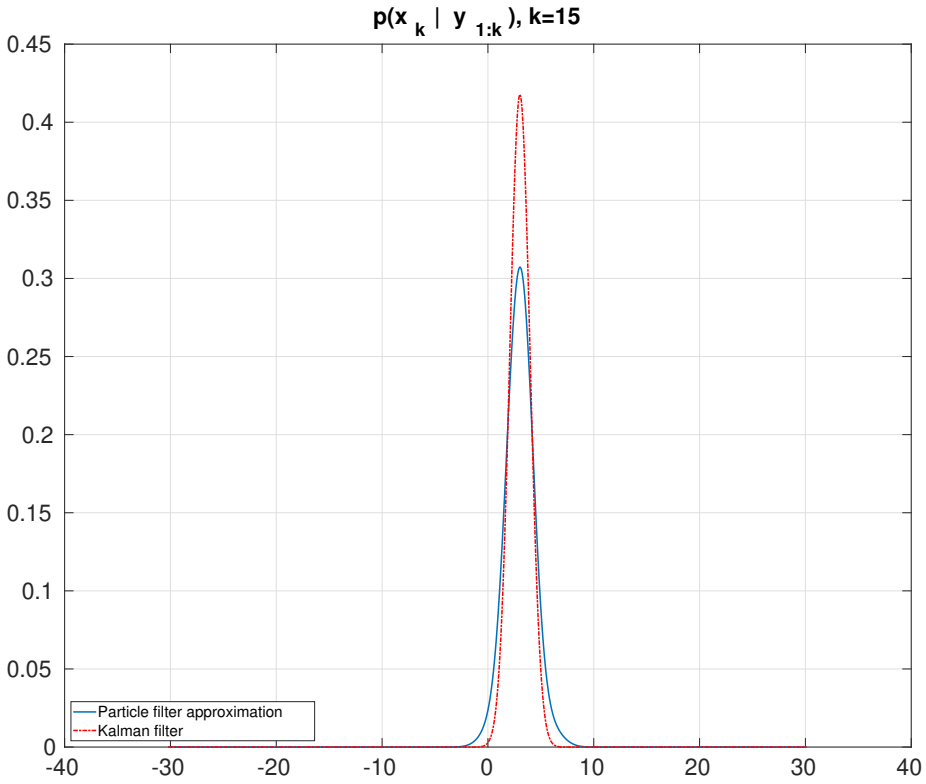

Figure 2.3: Posterior density at k=15

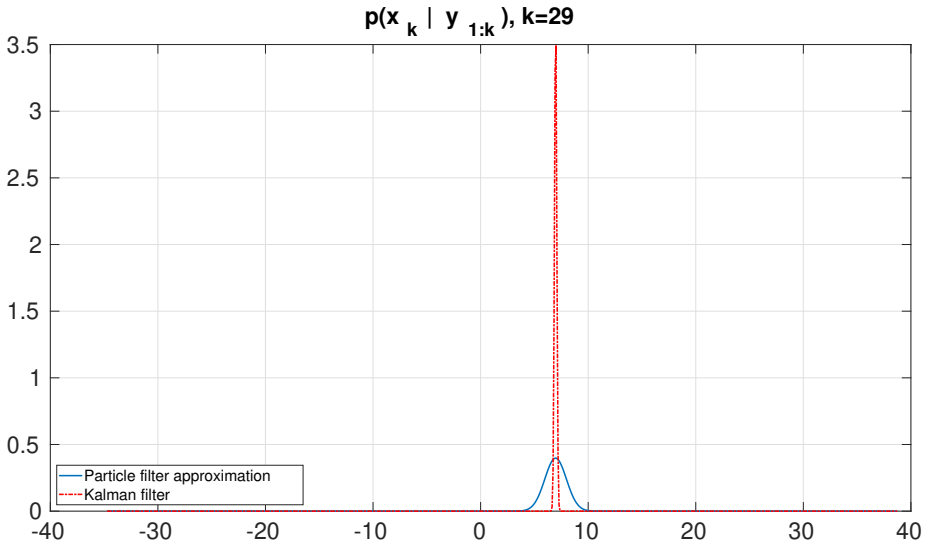

Figure 2.4: Posterior density at k=29

(b) Figure 2.5 shows how the PF recovers from a poor choice of prior. Indeed, the resampling version converges way faster towards the true state. The filter spends less computing power on bad particles.

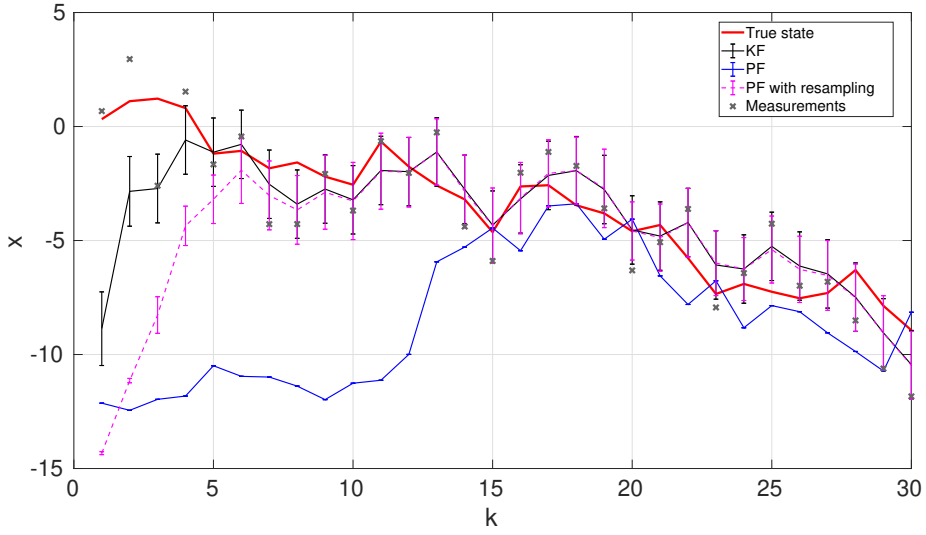

Figure 2.5: Filtered trajectories with incorrect prior

(c) Figure 2.6 shows the PF particles performances without resampling. Indeed the system has many "rogue" particles, probably with a low weight.

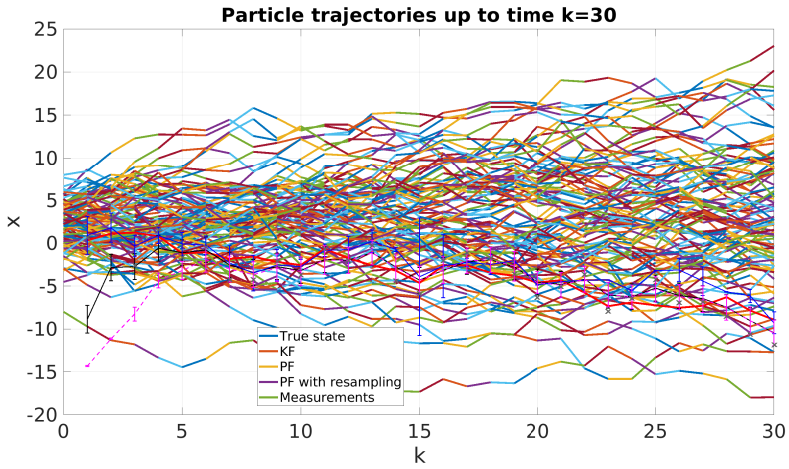

Figure 2.6: Performance of the PF without resampling (N=100)

(d) Figure 2.7 shows the performance of the PF with resampling. As one can see, the trajectory is close to the real one.

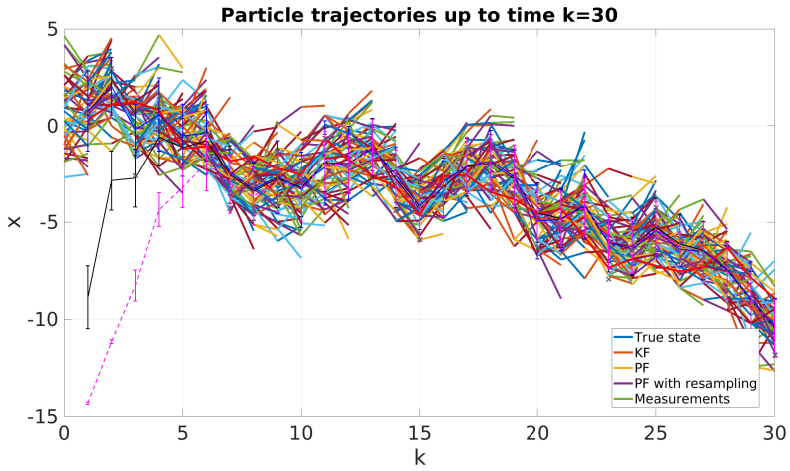

Figure 2.7: Performance of the PF with resampling (N=100)

3 Bicycle tracking in a village

(a) Here is the trajectory of the bicycle in the village on Figure 3.1

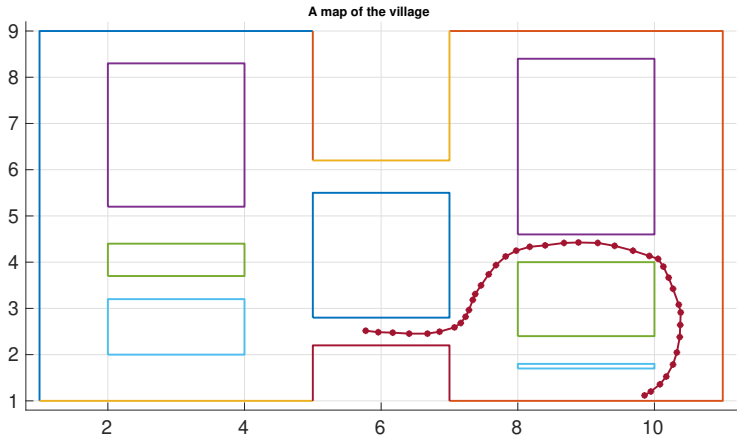

Figure 3.1: Bicycle true trajectory

(b)

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$h(x) = H * x;$$

$$\sigma_{meas\_speed} = 0.1;$$

$$R = \text{diag}([\sigma_{meas\_speed}, \sigma_{meas\_speed}].^2);$$

Figure 3.2 shows the measurement samples. Accordingly with the trajectory, we have all kinds of speed: negative or positive along x or y.

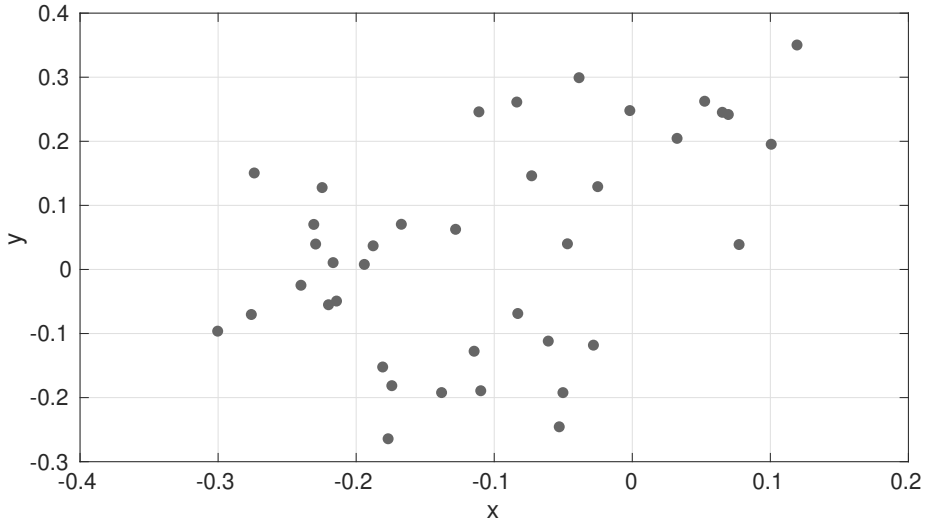

Figure 3.2: Velocity measurements

(c) By knowing the map of the village, we can rule out some particles that would collide with a wall or go out of the village altogether. To use this information in the filter, we can check that all generated particles are within the expected limits of the map. If not, we generate another sample until it's right. One more optimal approach could be to compute the nearest point close to the outliers that is plausible. This way, we don't need to wait to be "lucky" and generate a sample within the boundaries.

(d) Let us build a PF filter for the following motion model (Constant Velocity):

$$T = 1;$$

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$f(x) = A * x;$$

$$\sigma_{pos} = 0.1;$$

$$\sigma_{speed} = 1;$$

Figure 3.3 shows the performance of the filter. Some particles are shown to better illustrate its inner behavior. However, some particles are out of the map boundaries. It should be possible to apply the methods explained in part c) to avoid outliers.

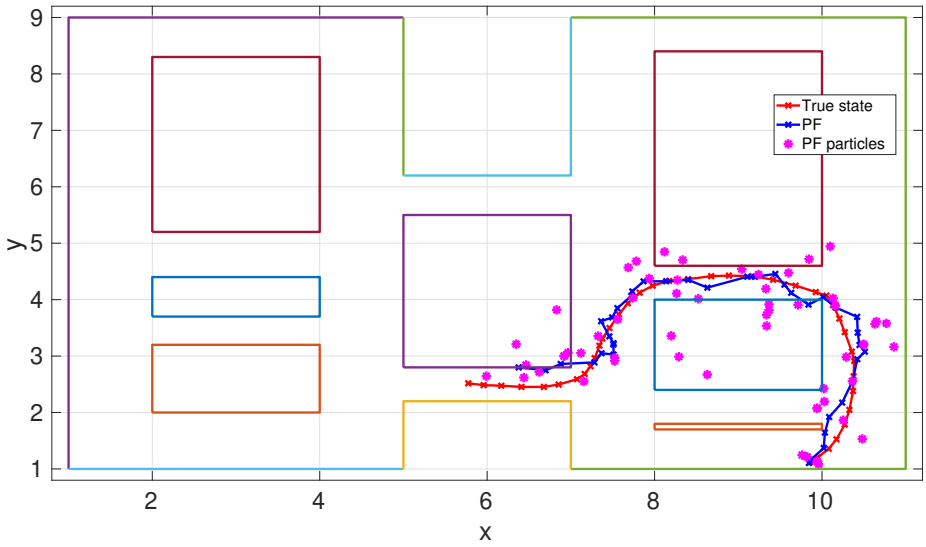

Figure 3.3: Performance of the PF with a good prior (N=1000)

(e) I did not get the time to do it, but it should be possible to compute the trajectory with the PF based on a very uncertain prior (the starting particles will be spread across the entire map), and store the trajectory of all the particles. Then, by looking at the trajectory that has the least number of particles out of the boundaries, we can pick it and get the most plausible trajectory and starting position for the bicycle.