Project Stochastic Processes

Clémence Cardon de Lichtbuer - Henry Grafé

23 April 2021

1 Theoretical Questions

1.1 Alternative derivation of Kalman Filter (KF)

1.1.1 Forecast step

Using the state-space model :

$$\tilde{\mu}_k = E\left[x_k | y_{1:k-1}\right] = E\left[A(x_{k-1}|y_{1:k-1}) + Bu_k + w_k\right]$$

since the noise w_k is zero centered, u_k is deterministic and $E\left[x_{k-1}|y_{1:k-1}\right] = \mu_{k-1}$, we obtain :

$$\tilde{\mu}_k = A\mu_{k-1} + Bu_k$$

using this result we can also compute the variance P_k :

$$\tilde{P}_k = E\left[(x_k|y_{1:k-1} - \tilde{\mu}_k)(x_k|y_{1:k-1} - \tilde{\mu}_k)^T\right]$$

$$= E\left[((A(x_{k-1}|y_{1:k-1}) + Bu_k + w_k) - (A\mu_{k-1} + Bu_k))((A(x_{k-1}|y_{1:k-1}) + Bu_k + w_k) - (A\mu_{k-1} + Bu_k))^T\right]$$

$$E\left[(A(x_{k-1}|y_{1:k-1} - \mu_{k-1}) + w_k)(A(x_{k-1}|y_{1:k-1} - \mu_{k-1}) + w_k)^T\right]$$

Since w_k is zeros mean, we can see this as the variance of two zero-centered RV, and the variance is the sum of each variances :

$$\tilde{P}_k = AP_{k-1}A^T + Q$$

1.1.2 Update step

we use the formula for the conditional distribution for jointly gaussian random vectors, which state that if X and Y are gaussian RVs, the distribution of X knowing $Y = y$ is also gaussian with mean and variance :

$$\bar{\mu} = \mu_X + C_{XY}C_{YY}^{-1}(y - \mu_Y)$$

$$\bar{V} = C_{XX} - C_{XY}C_{YY}^{-1}C_{YX}$$

we consider in our case that X is $x_k|y_{1:k-1}$ of mean $\tilde{\mu}_k$ and variance $\tilde{P}_k$ and Y is the measurement done at time k, wich has a mean (from state-space equation) of $C\tilde{\mu}_k$ and a variance we can compute :

$$Var(Y) = C_{YY} = E\left[(Cx_k|y_{1:k-1} + v_k - C\tilde{\mu}_k)(Cx_k|y_{1:k-1} + v_k - C\tilde{\mu}_k)^T\right]$$

$$= E\left[(C(x_k|y_{1:k-1} - \tilde{\mu}_k) + v_k)(C(x_k|y_{1:k-1} - \tilde{\mu}_k) + v_k)^T\right]$$

$$= C\tilde{P}_k C^T + R$$

The covariance between X and Y is :

$$C_{XY} = E\left[(x_k|y_{1:k-1} - \tilde{\mu}_k)(Cx_k|y_{1:k-1} + v_k - C\tilde{\mu}_k)^T\right]$$

$$= E\left[(x_k|y_{1:k-1} - \tilde{\mu}_k)(C(x_k|y_{1:k-1} - \tilde{\mu}_k) + v_k)^T\right]$$

$$= E\left[(x_k|y_{1:k-1} - \tilde{\mu}_k)(x_k|y_{1:k-1} - \tilde{\mu}_k)^T\right] C^T + E\left[(x_k|y_{1:k-1} - \tilde{\mu}_k)(v_k)^T\right]$$

since v_k is decorrelated from $x_k|y_{1:k-1}$ the second part is equal to 0 and we obtain :

$$= \tilde{P}_k C^T$$

With all this computed we can obtain the mean μ_k of $x_k|y1:k$:

$$\mu_k = \tilde{\mu}_k + \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + R}(y_k - C\tilde{\mu}_k)$$

and if we define $K_k = \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + R}$ we can write it :

$$\mu_k = \tilde{\mu}_k + K_k(y_k - C\tilde{\mu}_k)$$

To compute the variance $\tilde{P}_k$ of $x_k|y1:k$ we still need to compute the other covariance (C_{YX}), which is easily done since we did its complementary just above :

$$C_{YX} = C\tilde{P}_k$$

we can now use the formula for conditionnal distruibution of jointly gaussian random vectors :

$$P_k = \tilde{P}_k - \frac{\tilde{P}_k C^T C\tilde{P}_k}{C\tilde{P}_k C^T + R} = (I_{n_x} - K_k C)\tilde{P}_k$$

1.1.3 Reformulation of precedent results

We have to prove that

$$\mu_k = P_k(\tilde{P}_k^{-1}\tilde{\mu}_k + C^T R^{-1} y_k)$$
$$P_k^{-1} = \tilde{P}_k^{-1} + C^T R^{-1} C$$

are reformulation of the results we just obtained. We begin with the mean. We replace the expression of P_k inside the equation :

$$\begin{aligned}
\mu_k &= P_k(\tilde{P}_k^{-1}\tilde{\mu}_k + C^T R^{-1} y_k) \\
&= (\tilde{P}_k - K_k C\tilde{P}_k)(\tilde{P}_k^{-1}\tilde{\mu}_k + C^T R^{-1} y_k) \\
&= \tilde{P}_k\tilde{P}_k^{-1}\tilde{\mu}_k + \tilde{P}_k C^T R^{-1} y_k - K_k C\tilde{P}_k\tilde{P}_k^{-1}\tilde{\mu}_k - K_k C\tilde{P}_k C^T R^{-1} y_k \\
&= \tilde{\mu}_k + \tilde{P}_k C^T R^{-1} y_k - K_k C\tilde{\mu}_k - K_k C\tilde{P}_k C^T R^{-1} y_k \\
&= \tilde{\mu}_k + ((\tilde{P}_k C^T R^{-1} - K_k C\tilde{P}_k C^T R^{-1}) y_k - K_k C\tilde{\mu}_k)
\end{aligned}$$

For this expression to be equal to the one of 1.1.2, we have to have that :

$$\begin{aligned}
\tilde{P}_k C^T R^{-1} - K_k C\tilde{P}_k C^T R^{-1} &= K_k \\
\tilde{P}_k C^T R^{-1} &= K_k(C\tilde{P}_k C^T R^{-1} + 1) \\
\tilde{P}_k C^T R^{-1} R &= K_k(C\tilde{P}_k C^T R^{-1} + 1)R \\
\tilde{P}_k C^T &= K_k(C\tilde{P}_k C^T + R) \\
K_k &= \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + R}
\end{aligned}$$

Which is effectively the expression of the Kalman gain. Thus we proved the equality of the two expression of μ_k. Next we do the second equality. To prove it is true, we will show that $P_k^{-1} P_k = I_{n_x}$ using P_k^{-1} as defined here and P_k ad defined above. If we manage to prove this, it will mean the two equation (the one here and the one at 1.1.2) are equivalents.

$$\begin{aligned}
P_k P_k^{-1} &= (I_{n_x} - K_k C)\tilde{P}_k(\tilde{P}_k^{-1} + C^T R^{-1} C) \\
&= (\tilde{P}_k - K_k C\tilde{P}_k)(\tilde{P}_k^{-1} + C^T R^{-1} C) \\
&= [\tilde{P}_k - \tilde{P}_k C^T (C\tilde{P}_k C^T + R)^{-1} C\tilde{P}_k][\tilde{P}_k^{-1} + C^T R^{-1} C] \\
&= I_{n_x} - \tilde{P}_k C^T (C\tilde{P}_k C^T + R)^{-1} C + \tilde{P}_k C^T R^{-1} C - \tilde{P}_k C^T (C\tilde{P}_k C^T + R)^{-1} C\tilde{P}_k C^T R^{-1} C \\
&= I_{n_x} - \tilde{P}_k C^T [(C\tilde{P}_k C^T + R)^{-1} - R^{-1} + (C\tilde{P}_k C^T + R)^{-1} C\tilde{P}_k C^T R^{-1}] C \\
&= I_{n_x} - \tilde{P}_k C^T [(C\tilde{P}_k C^T + R)^{-1}(I_{n_x} + C\tilde{P}_k C^T R^{-1}) redRR^{-1} - R^{-1}] C \\
&= I_{n_x} - \tilde{P}_k C^T [(C\tilde{P}_k C^T + R)^{-1}(R + C\tilde{P}_k C^T) R^{-1} - R^{-1}] C \\
&= I_{n_x} - \tilde{P}_k C^T [R^{-1} - R^{-1}] C \\
&= I_{n_x}
\end{aligned}$$

we have just proved that the multiplication of the two equations was equal to the identity. This proves that the equality $P_k^{-1} = \tilde{P}_k^{-1} + C^T R^{-1} C$ is correct.

1.1.4 Why does it become expensive at large n_x and n_y ?

One of the heavy computation is for the state estimate covariance, for which we need to compute $AP_{k-1}A^T$. This implies matrix multiplication with matrices of dimensions n_x x n_x (naive matrix multiplication is $O(n_x^3)$), which would be very expensive for large n_x. The other heavy operation is at the update step, where we have to make the inversion of a matrix of shape n_y x n_y (also $O(n_y^3)$).

1.2 Ensemble Kalman Filter (EnKF)

1.2.1 Forecast step

we define $\tilde{x}_n = x_n^i | y_{1:n-1}$ and $\hat{x}_n = x_n^i | y_{1:n}$ The forecast distribution of $\tilde{x}_k^i$ is with mean and variance described using the state-space equation from the parameters of the update distribution at the previous step. The mean is as follow :

$$\tilde{\mu}_k^i = E\left[A\hat{x}_{k-1}^i + Bu_k + w_k^i\right] = A\mu_k^i + Bu_k$$

and the variance :

$$\tilde{P}_k^i = E\left[(A\hat{x}_{k-1}^i + Bu_k + w_k^i - \tilde{\mu}_k^i)(A\hat{x}_{k-1}^i + Bu_k + w_k^i - \tilde{\mu}_k^i)^T\right]$$

$$= E\left[(A\hat{x}_{k-1}^i + Bu_k + w_k^i - A\hat{\mu}_k^i - Bu_k)(A\hat{x}_{k-1}^i + Bu_k + w_k^i - A\hat{\mu}_k^i - Bu_k)^T\right]$$

$$\tilde{P}_k^i = A\hat{P}_{k-1}^i A^T + Q$$

1.2.2 Update step

We see that the updated state $\hat{x}_k^i = \tilde{x}_k^i + K_k(y_k - \tilde{y}_k^i)$, is a linear combination of multiple gaussian RVs, so it is itself a gaussian RVs.The mean and variances are described as (using the definition of $\tilde{y}_k^i$) :

$$\hat{\mu}_k^i = E\left[\hat{x}_k^i\right] = E\left[\tilde{x}_k^i + K_k(y_k - \tilde{y}_k^i)\right]$$

$$= \tilde{\mu}_k^i + K_k(y_k - C\tilde{\mu}_k^i)$$

For the variance, we just have to make the demonstration of 1.1.2, with no difference whatsoever, except the indice i over each variable. We obtain the variance :

$$\hat{P}_k^i = (I_{n_x} - K_k C)\tilde{P}_k^i$$

with $K_k = \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + R}$.

1.2.3 Advantages of EnKF

The Kalman gain is replaced by $\hat{K}_k = \hat{\hat{P}}_k C^T (C\hat{\hat{P}}_k C^T + R)^{-1}$ with $\hat{\hat{P}}$ the covariance of the forecast ensemble. Now this covariance is not anymore updated at each steps, but computed from the N samples at each steps. This avoid the multiplication we did in 1.1.1 to update $P_k = AP_{k-1}A^T$, of n_x x n_x matrices ($O(n_x^3)$). The inversion operation we did in 1.1.2 is still necessary to get the Kalman coefficient, but it is with matrices n_y x n_y ($O(n_y^3)$). This means that if our models has lots of parameters (n_x very big) this EnKF method spares us lots of computation. and just as for the usual Kalman filter, a low n_y lowers computation times, but in return offers less information for state estimate adjustments at each steps.

2 Pratical Models Implementation

In this section, we present the results we obtained when implementing the filters we described in the last section on a double-tank system. In addition, we implement an EnKF filter for a nonlinear system, where we replace the state-transition matrix A by a RK4-like nonlinear transition step, implemented following the instructions. We show that the classical KF performs correctly, and that the linear and nonlinear EnKF perform correctly when N = 100 sample states are used.

2.1 KF and EnKF for linear Double-tank system

2.1.1 the real state trajectories, the filtered states and the 95% confidence intervals ($\pm 1.96\sigma$) around the filtered states

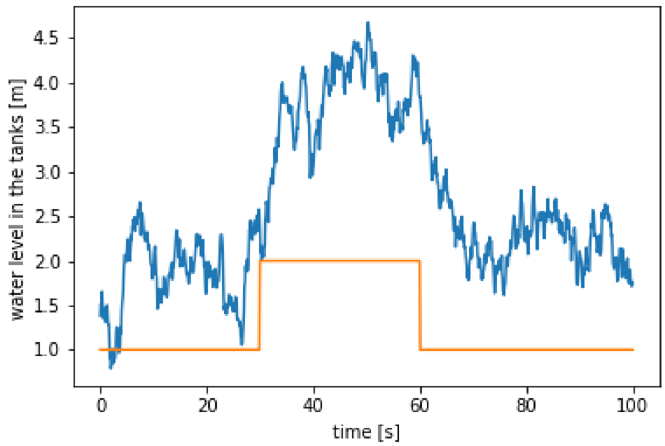

Figure 1: The output of the system y and the control signal u

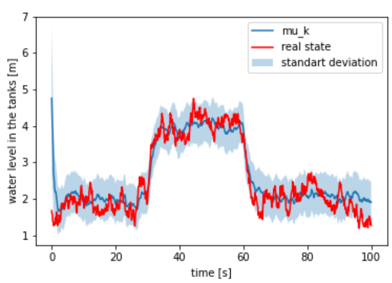

Figure 2: the 95% confidence interval around the filtered state x1 with KF

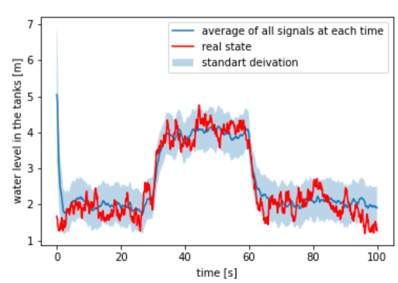

Figure 3: the 95% confidence interval around the filtered state x1 with EnKF with N = 100

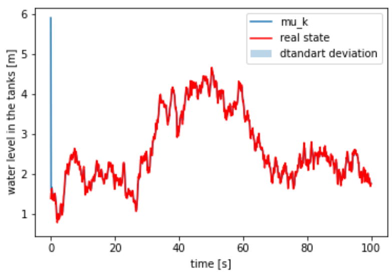

Figure 4: the 95% confidence interval around the filtered state x2 with KF

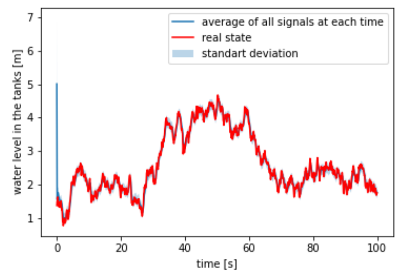

Figure 5: the 95% confidence interval around the filtered state x2 with EnKF with N = 100

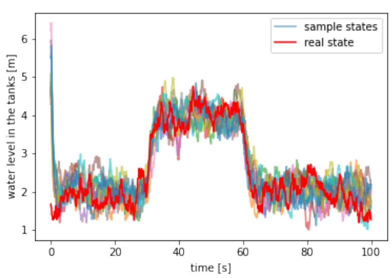

Figure 6: the samples states x1 of the EnKF (with N = 100) compared to the real state x1

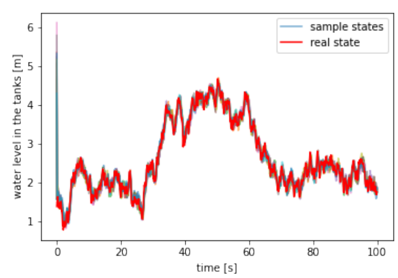

Figure 7: the samples states x2 of the EnKF (with N = 100) compared to the real state x2

The filtered states show quite clearly that the filter state for x_1 is more uncertain than x_2. This was to be expected since the output of the system is only x_2 with added noise. So knowing the state of x_2 is easy for the filter. x_1 is much harder since the filter have to rely on the knowledge of the system's behavior and of x_2 to deduce it. For the EnKF, the same results as for the KF are found. The x_1 state is much more uncertain than x_2. The levels of uncertainty is similar to the KF method, which prove that the method of computing the covariance matrix from the N sample states is valid. The plot for EnKF of the N filtered states reflect the bigger uncertainty for the estimate of x_1.

2.1.2 Impact of the number of sample states N on the precision of EnKF method

Here is showed the EnKF results for values N of 2, 5 and 20 (the one done just above was with N = 100). At N = 100 a performance similar to the classical KF is achieved, even tough N = 50 would probably suffice.

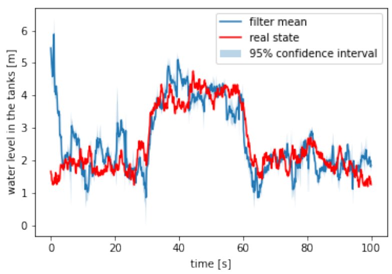

Figure 8: the 95% confidence interval around the filtered state x1 with EnKF with N = 2

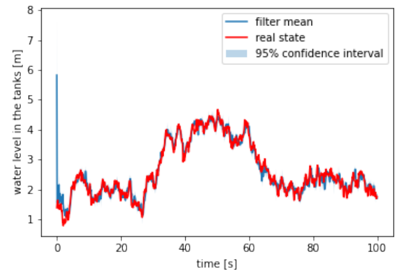

Figure 9: the 95% confidence interval around the filtered state x2 with EnKF with N = 2

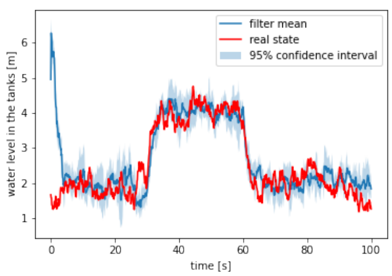

Figure 10: the 95% confidence interval around the filtered state x1 with EnKF with N = 5

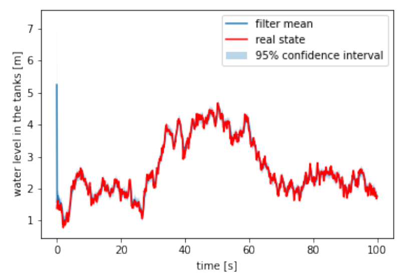

Figure 11: the 95% confidence interval around the filtered state x2 with EnKF with N = 5

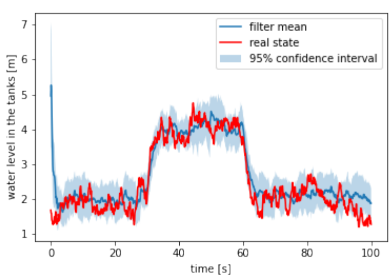

Figure 12: the 95% confidence interval around the filtered state x1 with EnKF with N = 20

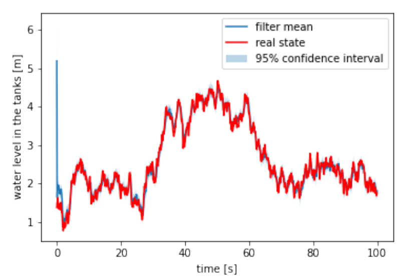

Figure 13: the 95% confidence interval around the filtered state x2 with EnKF with N = 20

2.1.3 Root-Mean-Square Deviation (RMSD) as a function of k

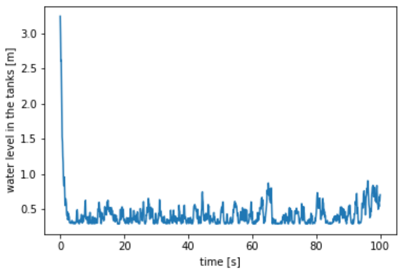

Figure 14: RMSD of x1 with KF

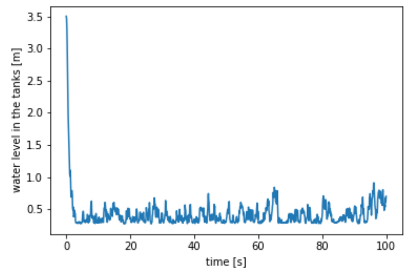

Figure 15: RMSD of x1 with EnKF

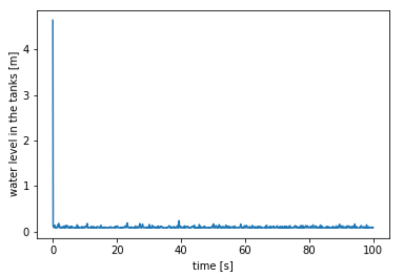

Figure 16: RMSD of x2 with KF

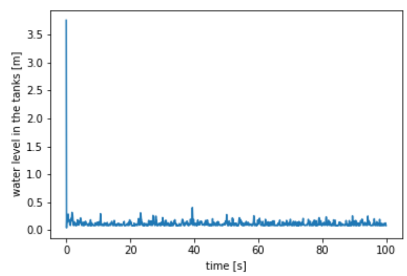

Figure 17: RMSD of x2 with EnKF

The RMSE is composed of two things : the difference between the mean of the filter and the actual state, and the variance of the filter result itself. Juste like before, the RMSE for KF and EnKF is bigger on average for x_1 than for x_2, and tends to somehow stabilize after a short time.

2.2 EnKF for nonlinear Double-tank system

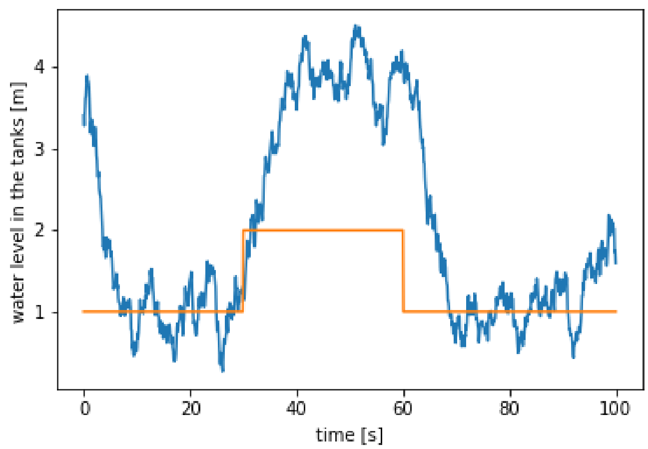

Figure 18: The output of the system y and the control signal u

2.2.1 the real state trajectories, filtered states, and 95% confidence intervals ($\pm 1.96\sigma$) around the filtered states

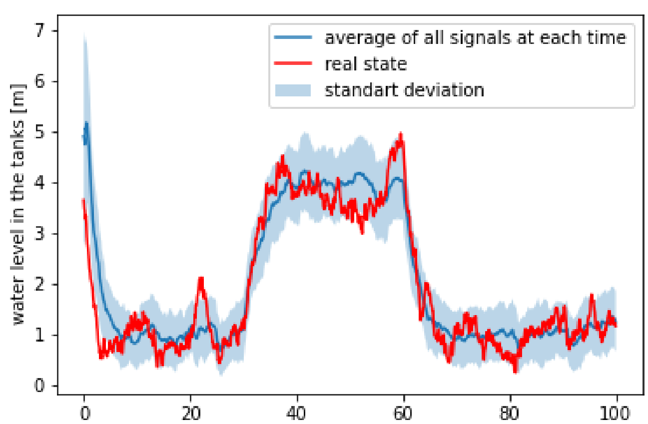

Figure 19: the 95% confidence interval around the filtered state x1 with nonlinear EnKF (N = 100)

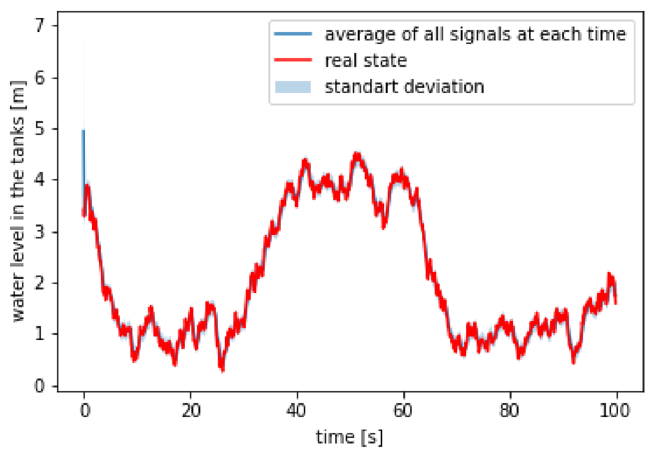

Figure 20: The 95% confidence interval around the filtered state x2 with nonlinear EnKF (N = 100)

The EnKF allows us to use a nonlinear model. The plots show estimations similar-looking to the linear KF and EnKF, with an bigger uncertainty for x_1 like before (which is normal since the output is still only x_2 plus noise). The plot of all the filtered states reflect that bigger uncertainty.

2.2.2 Impact of the number of sample states N on the precision of nonlinear EnKF method

Here is showed the nonlinear EnKF results for values N of 2, 5 and 20 (the one done just above was with N = 100). Here there is no 'classical KF' benchmark to compare the performances, but we can see that it performs sufficiently well at N = 100.

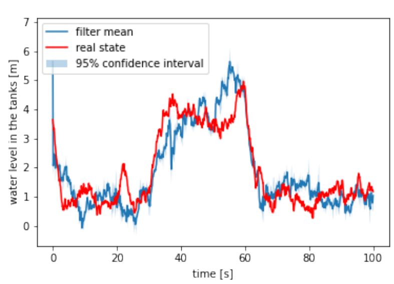

Figure 21: the 95% confidence interval around the filtered state x1 with nonlinear EnKF with N = 2

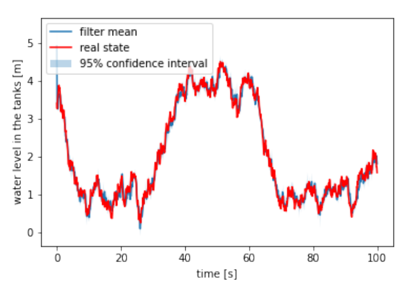

Figure 22: the 95% confidence interval around the filtered state x2 with nonlinear EnKF with N = 2

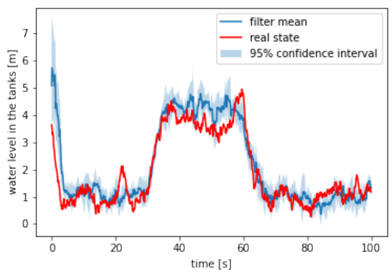

Figure 23: the 95% confidence interval around the filtered state x1 with nonlinear EnKF with N = 5

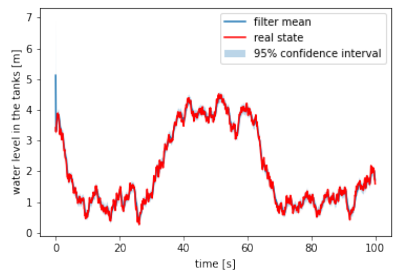

Figure 24: the 95% confidence interval around the filtered state x2 with nonlinear EnKF with N = 5

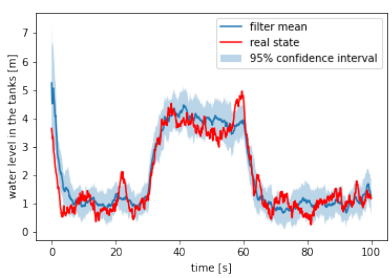

Figure 25: the 95% confidence interval around the filtered state x1 with nonlinear EnKF with N = 20

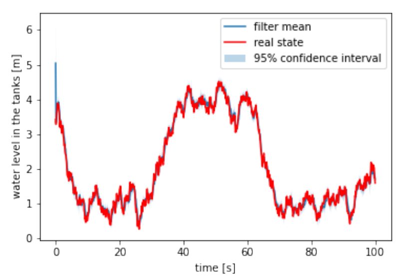

Figure 26: the 95% confidence interval around the filtered state x2 with nonlinear EnKF with N = 20

2.2.3 Root-Mean-Square Deviation (RMSD) as a function of k

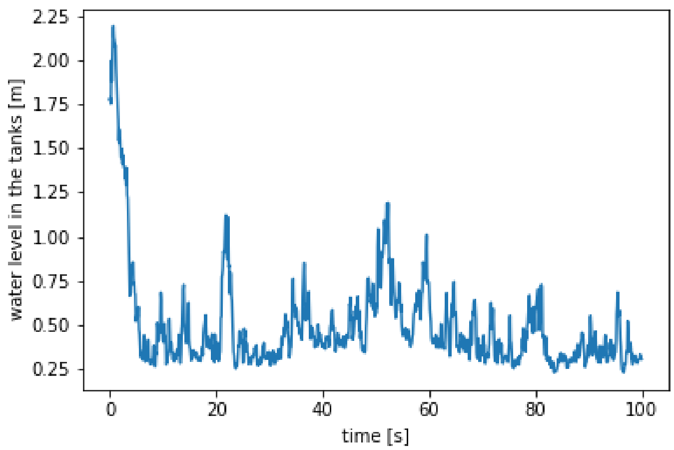

Figure 27: RMSD of x1 with EnKF non-linear (N = 100)

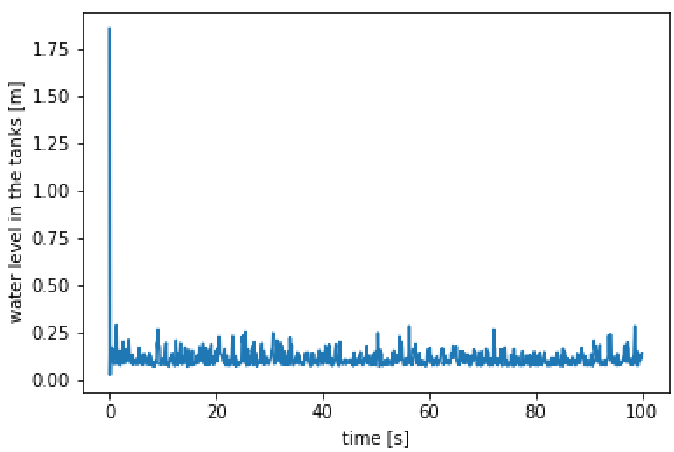

Figure 28: RMSD of x2 with EnKF non-linear (N = 100)

Just like the KF and linear EnKF, the nonlinear EnKF's RMSD for x_1 is on average bigger that for x_2. Unlike before, it doesn't really stabilize after a some times in the case of x_1, and is much more variable trough time. This could be caused by the fact that the model is nonlinear. At around 20s and 60s (among others) there are peaks on the real state x_1, high and large enough so that it is unlikely that it could be caused by the process noise. Maybe it is some kind of external perturbation. In all cases the EnKF shows robustness to such perturbation, and the system keeps doing its job correctly after that, even tough it causes peaks in the RMSD.