Bayesian Online Spectral Tracking

René Duijkers
Technical University Eindhoven
Faculty of Electrical Engineering
Email: r.duijkers@student.tue.nl

Internship supervised by:
Prof.dr.ir. B. de Vries

***Abstract*—We consider the estimation of the dynamic state and the time-varying noise parameters of a linear Gaussian process. This estimation can be used for a wide variety of purposes, including noise reduction and spectral tracking. We derive two recursive algorithms based on a Bayesian Kalman Filter using different assumptions for each one of them. Both algorithms are able to give a good estimate of the dynamic state.**

I. Introduction

A common goal in the design of signal processing systems is noise reduction. You have noise corrupted observations and want to recover the original signal. In audio processing for example, you might have a speech recording with background noises. In order to make good use of the speech signal you want to remove the background noises and only keep the speech signal.
Noise can be caused by various causes. You often think of noise induced by hardware (e.g. a bad microphone) or caused by measuring mixed or corrupted signals, but also software algorithms can cause noisy observations. Simplifications and bad assumptions can change the original signal without the user being aware of it.

An example is the use of a windowed *Discrete Fourier Transformation* (DFT) for spectral estimation. This algorithm computes the discrete Fourier transform which converts a signal from time to frequency domain. Although this transformation is often considered to be exact, it is subject to errors caused by spectral leakage.

The amount of leakage depends on the choice of the window, hence choosing a different window will result in a different spectrum. So if we calculate the power over a given frequency band we actually getting noisy observations.

Figure 1 shows an example of a time-varying power sequence of a speech signal estimated using the windowed DFT. Intuition tells us this power sequence contains high frequency noise. So we could use a *Low Pass Filter* (LPF) to smoothen the signal and obtain a good estimation of the original signal. When using a LPF, the challenge lies in setting the right cutoff frequency. Setting this frequency too high will not remove all the noise, but setting it too low will filter out dynamics of the original signal. Without knowledge of the noise statistics it is difficult to know which value to choose.

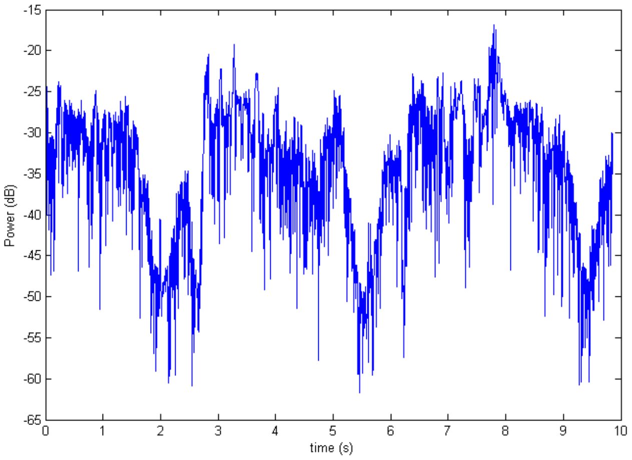

Fig. 1. Spectral power of a speech signal estimated using the windowed DFT

In this paper we want to solve this filter problem by learning from data. We assume we have a description of the data-generating system and together with observations we estimate the most likely value of the original signal. We aim to do this using a *Kalman filter* (KF) [1], [2] in section III. Because we have no a-priori knowledge of the noise statistics we will extent the standard Kalman filter so it will also estimate noise variances. We describe two different methods in section IV and evaluate them in section V

II. Problem statement

We have a signal that contains noise and we want to remove this noise. When we know the underlying data-generating structure we can use knowledge of the past to estimate the original signal. However, since the noise is stochastic we will always have uncertainty about the state of the system, and therefore uncertainty about the signal. Hence, we need observations to adjust our predictions.
We do this by first assuming the signal can be described by the following linear process

$$x_n = Ax_{n-1} + q_n \tag{1a}$$
$$y_n = Cx_n + r_n \tag{1b}$$

where $x_n \sim \mathcal{N}(\hat{x}_n, P_n)$ is the original m-dimensional signal vector (also know as the hidden state vector), y_n is a d-dimensional observation vector, $q_n \sim \mathcal{N}(0, Q)$ is the process noise and $r_n \sim \mathcal{N}(0, R)$ is the observation noise.

Using this model our problem becomes the search for $\hat{x}_n$, the mean value of the probability of the hidden state x_n. The variance P_n is also of interest, because it indicates how certain we are in the value of $\hat{x}_n$.

III. State estimation

In this section we cover the problem of estimating the hidden states. We will do this by calculating the posterior density function of the states, resulting in a Kalman filter. At the end of the section we will see possible extensions to this Kalman filter.

A. Kalman filter

Consider the following linear process as described in (1) and assume we have knowledge of the noise variances Q and R. When new data arrives, it is possible to estimate the posterior density function of the state using Bayes' rule. That is:

$$Posterior = \frac{Likelihood}{Evidence} \times Prior \tag{2}$$

Because the system is linear and we assume that the prior and likelihood are Gaussian, the posterior will also be Gaussian. This implies that, in order to update the model parameters estimate we only need to know the mean and variance of the conditional density function. Under the assumption that the noise processes are zero mean and uncorrelated we can derive the likelihood, evidence and prior in terms of first and second order statistics (See Appendix A):

$$Prior = p(x_n|Y_n, M) = \mathcal{N}(\hat{x}_{n-1}, AP_{n-1}A^T + Q) \tag{3}$$

$$\begin{aligned} Evidence &= p(y_n|Y_{n-1}, M) \\ &= \mathcal{N}(C\hat{x}_n, C(AP_{n-1}A^T + Q)C^T + R) \end{aligned} \tag{4}$$

$$Likelihood = p(y_n|x_n, M) = \mathcal{N}(Cx_n, R) \tag{5}$$

here Y_n represents the observations up to time step n and M represents the model. Substituting equations (3), (4) and (5) into (2) gives the posterior:

$$p(x_n|Y_n, M) = \alpha_n \exp\left(-1/2(x_n - \hat{x}_n)P_n^{-1}(x_n - \hat{x}_n)\right) \tag{6}$$

where the coefficient α_n is given by

$$\alpha_n = \frac{|C(AP_{n-1}A^T + Q)C^T + R|^{1/2}}{(2\pi)^{m/2}|R|^{1/2}|AP_{n-1}A^T + Q|^{1/2}} \tag{7}$$

As previously stated this is a Gaussian distribution. The values of interest are the mean $\hat{x}_n$ and variance P_n:

$$\hat{x}_n = \hat{x}_{n-1} + K_n(y_n - C\hat{x}_{n-1}) \tag{8}$$

$$P_n = AP_{n-1}A^T + Q - K_nC(AP_{n-1}A^T + Q) \tag{9}$$

where, K_n represents the *Kalman gain* and is given by

$$K_n = (AP_{n-1}A^T + Q)C^T[R + C(AP_{n-1}A^T + Q)C^T]^{-1} \tag{10}$$

Equations (8), (9) and (10) only depend on the values at the previous time step. So the Kalman filter algorithm can be implemented recursively.

When combining equations (8), (9) and (10) the Kalman filter can be written as a single equation. However, it is most often divided in two distinct steps: ”Predict” and ”Update”. The predict step uses the state estimate from the previous time step to produce a prediction of the state, without using observation information from the current time step. In the update step, this prediction is combined with current observation information to refine the state estimate. When necessary, we use $\tilde{x}_n$ and $\tilde{P}_n$ to indicate the estimates of the predict step.

B. Extended Kalman filter

The Kalman filter is limited to a linear process. If the observations are based on the nonlinear system $y_n = g(x) + r_n$ we can use the *Extended Kalman filter* (EKF). This estimator is based on a Taylor series expansion of the nonlinear function $g(x)$ around the previous estimate [3]. That is,

$$g(x) = g(\hat{x}) + \left.\frac{\partial g}{\partial x}\right|_{(x=\hat{x})}(x - \hat{x}) + \cdots \tag{11}$$

Following the same derivations as with the standard Kalman filter, we get the following update equations:

$$\hat{x}_n = \hat{x}_{n-1} + K_n(y_n - G_n^T\hat{x}_{n-1}) \tag{12}$$

$$P_n = AP_{n-1}A^T + Q - K_nG_n^T(AP_{n-1}A^T + Q) \tag{13}$$

$$K_n = (AP_{n-1}A^T + Q)G_n[R + G_n^T(AP_{n-1}A^T + Q)G_n]^{-1} \tag{14}$$

Here G represents the Jacobian matrix:

$$G = \left.\frac{\partial g}{\partial x}\right|_{(x=\hat{x})} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_2}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_1} \\ \frac{\partial g_1}{\partial x_2} & & & \\ \vdots & & & \vdots \\ \frac{\partial g_1}{\partial x_m} & \cdots & & \frac{\partial g_n}{\partial x_m} \end{bmatrix}^T_{(x=\hat{x})}$$

The EKF is only an approximation to the expected value because the linearizion of the nonlinear mapping is suboptimal. Because of these approximations the EKF may diverge and give poor results when having highly nonlinear functions.

IV. Noise parameter estimation

A limitation of the Kalman filter is the assumption that the noise variances R and Q must be known. However, in most practical cases it is difficult to have prior knowledge about these statistics. In addition, when their values are non-stationary the Kalman filter can give poor results. A solution to this, is to estimate these variances from data. In this section we will study two different methods of noise estimation. The first method uses *evidence maximization* [3]. The second method uses *Variational Bayes* (VB) assumptions to derive a noise estimate [4].

A. Evidence maximizing Kalman filter (EM-KF)

In this section we assume $q_n \sim \mathcal{N}(0, Q_n)$ is the process noise with a varying variance Q_n. For simplicity, we have restricted our analysis in this section to a one-dimensional output. Our goal is to get a good estimation of this process noise by maximizing the likelihood of its variance:

$$p(y_n|Y_{n-1}, Q_n, M) \tag{15}$$

Note that this likelihood is the same as the evidence at the parameter level (4). So if we want to maximize the likelihood of the noise variance we can do this by maximizing the evidence (or marginal likelihood) at the parameter estimation level. If we write the evidence as a Gaussian function we get:

$$\begin{aligned} p(y_n|Y_{n-1}, Q_n, M) = & \frac{1}{(2\pi)^{\frac{1}{2}}(C\tilde{P}_n C^T + R)^{-\frac{1}{2}}} \\ & \times \exp\left(\frac{1}{2}\frac{(y_n - C\hat{x}_n)^2}{C\tilde{P}_n C^T + R}\right) \end{aligned} \tag{16}$$

where

$$\tilde{P}_n = AP_{n-1}A^T + Q_n$$

If we assume that the process noise variance can be described by a single parameter q_n (more specifically $Q_n = q_n I_q$), we can find the maximum of the evidence function by differentiating with respect to q_n. Additionally, to make the calculation easier we introduce the residual $r_n = y_n - C\hat{x}_n$ which transforms the evidence into a zero mean Gaussian. Now the derivation becomes:

$$\begin{aligned} \frac{d}{dq_n}p(r_n) = & \frac{1}{(2\pi)^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\frac{r_n^2}{C\tilde{P}_n C^T + R}\right) \times \\ & \left[\frac{-1}{2}CC^T(C\tilde{P}_n C^T + R)^{\frac{-3}{2}} + \frac{1}{2}r_n^2 CC^T(C\tilde{P}_n C^T + R)^{\frac{-5}{2}}\right] \end{aligned} \tag{17}$$

Since $\mathbb{E}[r_n^2] = C\tilde{P}_n C^T + R$ (see (4)) we can equate the derivative to zero which gives us:

$$r_n^2 = \mathbb{E}[r_n^2] \tag{18}$$

This shows that in order to maximize the evidence, we need to match the residual with its variance. To find an algorithm to update q_n we use

$$r_n^2 = CAP_{n-1}A^T C^T + q_n CC^T + R \tag{19}$$

So it follows that q_n can be computed by

$$q_n = \max\left(\frac{r_n^2 - CAP_{n-1}A^T C^T - R}{CC^T}, 0\right) \tag{20}$$

The estimator increases q_n when the residuals are greater then predicted. Each time this happens, the Kalman gain increases and the model parameters also increase. So, as a consequence the algorithm places more emphasis on the observations. As long as the residuals remain smaller than predicted, the process noise is set to zero.

Strictly speaking, this is not a full Bayesian solution since we only compute the likelihood of the noise variances. We assume no knowledge of the prior at the noise estimation level and the estimate is based on a single residual. We can overcome this problem by using a moving window on the residuals. If we use the following sample mean:

$$m_r = \frac{1}{N}\sum_{l=0}^{N-1}\frac{r_{n-l}}{R^{1/2}} \tag{21}$$

we can follow the same procedure and find the windowed noise estimation:

$$q_n = \max\left(\frac{m_r^2 - S_N P_{n-1} S_N^T + \frac{1}{N}}{S}, 0\right). \tag{22}$$

where

$$S_k = \frac{1}{N}\sum_{l=N-k}^{N}\frac{C}{R^{1/2}}$$

$$S = S_N S_N^T + S_{N-1}S_{N-1}^T + \cdots + S_1 S_1^T$$

This estimator uses window length N to change the effect of the noise adaptation on the Kalman filter. Choosing the right window length is a dilemma because without a-priori knowledge we don't know the smoothness of the data.

B. Variational bayes Kalman Filter (VB-KF)

We have seen that evidence maximizing method is not a full Bayesian solution. Using Variational Bayes assumptions we can derive a solution that not only computes the joint likelihood of the state and noise variances, but also uses the priors.

Variational Bayes literature mainly focuses on the estimation of the varying variance R_n. So now we assume Q_n to be known. Like in [4] we assume that this variance has d independent parameters $\sigma_{n,i}^2, i = 1, \ldots, d$ such that $R_n = \text{diag}(\sigma_{n,1}^2, \cdots, \sigma_{n,d}^2)$. We also assume that we can approximate the conditional distribution $p(x_n, R_n|Y_n)$ as a product of Gaussian and independent Inverse-Gamma distributions:

$$p(x_n, R_n|Y_n) = \mathcal{N}(\hat{x}_n, P_n)\prod_{i=1}^{d} IG(\alpha_{n,i}, \beta_{n,i}) \tag{23}$$

This approximation is chosen because the Inverse-Gamma distribution is the conjugate prior distribution for the variance of a Gaussian distribution.

Because the dynamics of the state and observation noise variances are assumed to be independent, the prediction step of the state will stay the same as with the standard Kalman filter. This step will result in a Gaussian distribution with mean $\tilde{x}_n$ and variance $\tilde{P}_n$.

The noise variance need to be predicted such that the distribution of each σ_i^2 will result in an Inverse-Gamma distribution. Usually, the dynamical model of the noise variances is not known in detail, and so we take a predict step which simply "spreads" the previous approximate posteriors. We choose to multiply the hyper-parameters α and β by a factor of $\rho \in (0, 1]$, keeping the noise variances σ_i^2 constant. So:

$$\begin{aligned} \tilde{\alpha}_{n,i} &= \rho_i \, \alpha_{n-1,i} \\ \tilde{\beta}_{n,i} &= \rho_i \, \beta_{n-1,i} \end{aligned} \tag{24}$$

Keeping the variances constant ensures that the algorithm starts with the same values as the one estimated at the previous iteration. So, if the variances already have the correct values they remain unaffected. The value of ρ represents the assumed time-fluctuations. $\rho = 1$ corresponds to stationary hyper-parameters.

We now have a joint prediction distribution:

$$\begin{aligned} p(x_n, R_n|Y_{n-1}) &= p(x_n|Y_{n-1})p(R_n|Y_{n-1}) \\ &= \mathcal{N}(\tilde{x}_n, \tilde{P}_n) \prod_{i=1}^{d} IG(\tilde{\alpha}_{n,i}, \tilde{\beta}_{n,i}) \end{aligned} \tag{25}$$

In the update step, we want to compute the posterior distribution by Bayes' rule:

$$p(x_n, R_n|Y_n, M) = \frac{p(y_n|x_n, R_n, M)p(x_n, R_n|Y_{n-1}, M)}{p(y_n|Y_{n-1}, M)} \tag{26}$$

However, the exact posterior will not be tractable. In order to make it possible to compute a posterior distribution we will now form a variational approximation. We use the VB-approach used in [4] and search for a distribution that is simpler but gives a good approximation for the posterior $p(x_n, R_n|Y_n)$ as follows:

$$p(x_n, R_n|Y_n, M) \approx Q_x(x_n)Q_R(R_n) \tag{27}$$

The VB-approximation can now be formed by minimizing the *Kullback-Leibler* divergence (KL) between the separable approximation and the true posterior:

$$\begin{aligned} &KL[Q_x(x_n)Q_R(R_n)||p(x_n, R_n|Y_n)] \\ &= \int Q_x(x_n)Q_R(R_n) \log\left(\frac{Q_x(x_n)Q_R(R_n)}{p(x_n, R_n|Y_n)}\right) dx_n dR_n \end{aligned} \tag{28}$$

The KL-divergence is a measure of discrepancy from one distribution to another. It is always a non-negative quantity, so the distribution which gives the lowest value gives the best approximation ($KL[P||Q] = 0$ if and only if $P = Q$). As shown in [4], minimizing the KL-divergence with respect to the distributions $Q_x(x_n)$ and $Q_R(R_n)$ gives:

$$Q_x(x_n) \propto \exp\left(\int \log p(y_n, x_n, R_n|Y_{n-1})Q_R(R_n)dR_n\right) \tag{29}$$

$$Q_R(R_n) \propto \exp\left(\int \log p(y_n, x_n, R_n|Y_{n-1})Q_x(x_n)dx_n\right) \tag{30}$$

These distributions are coupled, so they cannot be solved directly. However, when we calculate the integral in the first equation we find:

$$\begin{aligned} &\int \log p(y_n, x_n, R_n|Y_{n-1})Q_R(R_n)dR_n \\ &= -\frac{1}{2}(y_n - C\hat{x}_n)^T \mathbb{E}[R_n^{-1}]_R(y_n - C\hat{x}_n) \\ &- \frac{1}{2}(\hat{x}_n - \tilde{x}_n)^T \tilde{P}_n^{-1}(\hat{x}_n - \tilde{x}_n) \\ &+ C_1 \end{aligned} \tag{31}$$

where $\mathbb{E}[.]_R = \int(.)Q_R(R_n)dR_n$ is the expected value with respect to the distribution $Q_R(R_n)$, and C_1 denotes terms that are independent of x_n. The result is quadratic in x_n, so we see that $Q_x(x_n)$ is a Gaussian distribution.

In a similar way we can calculate the integral in the second equation:

$$\begin{aligned} &\int \log p(y_n, x_n, R_n|Y_{n-1})Q_x(x_n)dx_n \\ &= -\sum_{i=1}^{d}\left(\frac{3}{2} + \alpha_{n,i}^2\right)\ln(\sigma_{n,i}^2) - \sum_{i=1}^{d}\frac{\beta_{n,i}}{\sigma_{n,i}^2} \\ &- \frac{1}{2}\sum \frac{1}{\sigma_{n,i}^2}\mathbb{E}\left[(y_n - C\hat{x}_n)_i^2\right]_x + C_2 \end{aligned} \tag{32}$$

where $\mathbb{E}[.]_x = \int(.)Q_x(x_n)dx_n$ is the expected value with respect to the distribution $Q_x(x_n)$. Here the result is a product of Inverse-Gamma distributions. By evaluating the expectations in Equations (31) and (32), and matching the parameters of the distributions on left and right hand sides we get the following densities:

$$Q_x(x_n) = \mathcal{N}(\hat{x}_n, P_n) \tag{33}$$

$$Q_R(R_n) = \prod_{i=1}^{d} IG(\alpha_{n,i}, \beta_{n,i}) \tag{34}$$

where the parameters $x_n, P_n, \alpha_{n,i}$ and $\beta_{n,i}$ can be calculated by the following (coupled) set of equations:

$$\begin{aligned} \hat{x}_n &= \tilde{x}_n + \tilde{P}_n C^T (C\tilde{P}_n C^T + R_n)^{-1}(y_n - C\tilde{x}_n) \\ P_n &= \tilde{P}_n - \tilde{P}_n C^T (C\tilde{P}_n C^T + R_n)^{-1} C\tilde{P}_n \\ \alpha_{n,i} &= \frac{1}{2} + \tilde{\alpha}_{n,i} \\ \beta_{n,i} &= \frac{1}{2}\left[(y_n) - C\hat{x}_n)_i^2 + (CP_nC^T)_{ii}\right] \end{aligned} \tag{35}$$

where $i = 1, \cdots, d$ and $R_n = \text{diag}\left(\frac{\beta_{n,1}}{\alpha_{n,1}}, \cdots, \frac{\beta_{n,d}}{\alpha_{n,d}}\right)$.

Because the equations (35) are coupled we have to compute a couple of iterations to find a good estimate of the posterior distribution. This distribution has the same form as the one we began with, so we can implement this algorithm recursively.

V. Experimental Evaluation

In this section we evaluate both methods by means of two simulation examples. In the first simulation we generate a signal using a known model and random noise generated with known and constant variances. This gives us the opportunity to evaluate the methods when the statistical properties of the underlying system are known. In the second simulation we

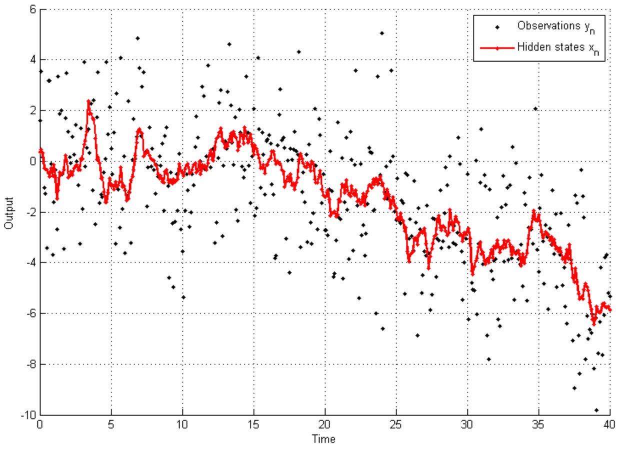

Fig. 2. Generated Random Walk

use a time-varying spectral power of a speech signal of which we do not know the statistical properties. This simulation will show how the methods perform in a more realistic situation.

A. *Gaussian random walk*

A random walk is a mathematical formalization of a path that consists of a succession of random steps. When we want to use a state space system to simulate a random walk we can use the following 1-dimensional linear process:

$$x_n = x_{n-1} + q_n \tag{36a}$$

$$y_n = x_n + r_n \tag{36b}$$

where $q_n \sim \mathcal{N}(0, Q)$ is the process noise and $r_n \sim \mathcal{N}(0, R)$ is the observation noise.

In our simulation we have generated 400 data points taking $Q = 0.1$ and $R = 5$. This gives us a slowly drifting sequence with a relative large observation noise. The state sequence and corresponding observations are shown in Figure 2.

Figure 3 shows the estimated hidden states using both the EM-KF and VB-KF methods. Here the EM-KF method uses a window of $N = 5$. The colored areas visualize the 95%-confidence bounds for which we use two standard deviations ($= 2\sigma$). The figure shows that both methods estimate similar hidden states. The biggest differences lie in the confidence bounds. Because of its design the confidence-bounds of the EM-KF method have more extreme values in comparison to more constant VB-KF method.

B. *Speech signal*

As mentioned in section IV, the noise variances are usually not known beforehand. So in this simulation, we consider the power sequence of a speech signal with unknown statistical properties. This gives a more practical situation and directly show us the consequences of not knowing the variances.

The signal we consider is the power sequence of a speech signal shown in Figure 1. This signal is sampled with a sample rate of 8 kHz. Because we don't know the data-generating structure we model the process with a Gaussian random walk. After some trials, the noise statistics are guessed to be $R = 200$ and $Q = 5 \cdot 10^{-4}$ for EM-KF and VB-KF, respectively.

Figure 4 shows the estimates calculated by using the EM-KF and VB-KF methods in the time interval between 2.5 and 4 seconds. Here, both methods give a smooth estimate of the spectral power.

The variances (and thus the confidence-bounds) appear to be very small. This is because the sample rate of 8 kHz which makes the state differences at each time step relative small. Nevertheless the figure shows the peaks in the variance when using the EM-KF method. Apparently the model is not able to predict some of the rapid changes, causing the variances to increase enormously.

This behavior can be explained by non-stationarity to which the filter wants to adapt. The filter could also be oversensitive to rapid changes. Since the values of the fixed noise variances are unknown it is difficult to say which explanation is the right one.

VI. Conclusions and Future Work

As shown in section V, both the EM-KF and VB-KF are able to get a good data based estimate of the hidden states when only having noisy observations. Although this is the primary goal of this paper, the results are not fully satisfactory.

Looking back at section I, we introduced the problem of selecting the right cutoff frequency to remove noise using a LPF. We wanted to solve this parameter selection problem by using a data driven approach. When we use the EM-KF or VB-KF method to solve this example we indeed have a data driven result, but nevertheless the parameter selection problem remains. Since we only estimate one noise variance, the other ”fixed” variance will act as a smoothening parameter, essentially giving the same problem as with the LPF.

An obvious solution would be to estimate both variances at the same time. Unfortunately literature does not mention many methods able to do this. In almost every method one variance is assumed to be known.

Next to the selection problem of the ”fixed” variance there are other practical downsides worth mentioning:

- The derived algorithms are not flexible. When due to new knowledge some key assumptions change, the whole derivation needs to be re-done. You can already see this by looking at the EM-KF and VB-KF methods. Although the assumptions and outcomes are very similar, the derivations are completely different.
- It is difficult to compare the quality of the different methods. Both methods give different estimates and without the noiseless signal it is often difficult to say which one is better. The possibility of poorly chosen variances only increase this difficulty.

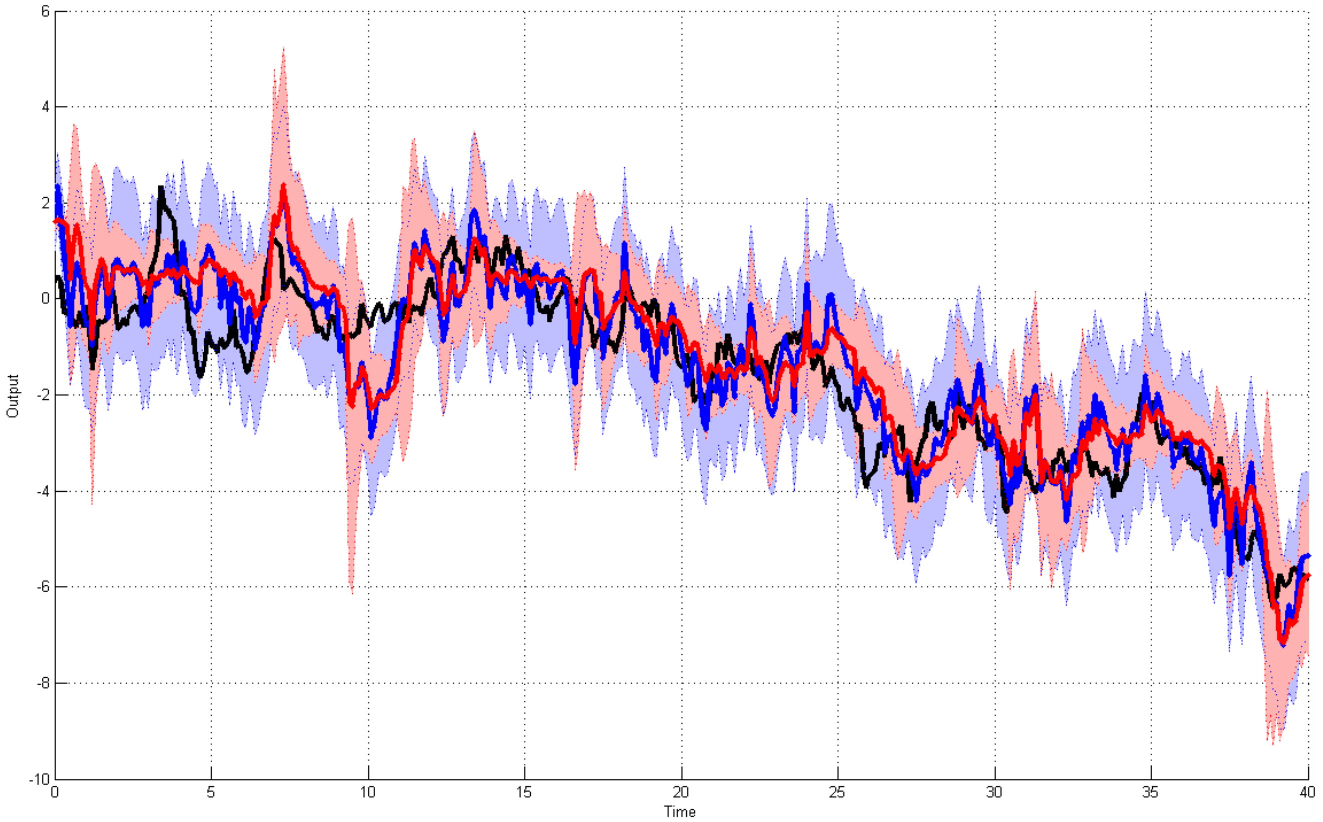

Fig. 3. Estimates of a random walk using VB-KF (blue) and EM-KF (red). The true state values are shown in black

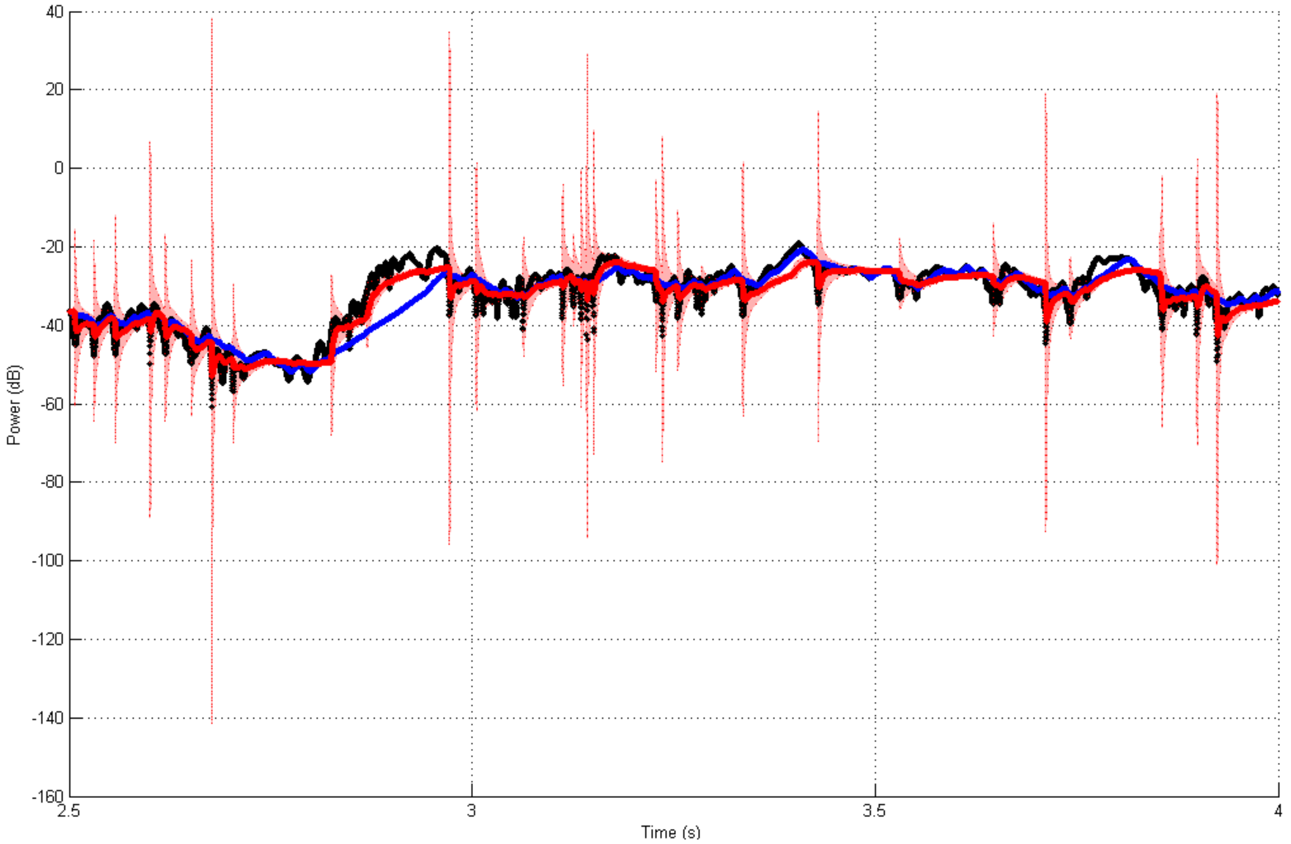

Fig. 4. Estimates of the spectral power using VB-KF (blue) and EM-KF (red). The observations are shown in black

Of course, the two proposed methods are not a summary of the whole literature about this topic. There are many variations and different approaches like sample methods (e.g. Particle filtering [5]).

This large collection of (slightly) different methods may make it difficult to select the right one. A possible solution to his might be using *Factor Graphs* with *message passing* algorithms [6]. These graphical models claim to provide a framework for the systematic and consistent derivation of classic model-based algorithms [7]. This will make the solutions more flexible more easy to understand. Algorithms like *Expectation Maximization* (EM) [2] are already solved using factor graphs [8] making it a promising direction for future work.

Appendix A
Bayesian derivation of the Kalman Filter

A. Prior density function

The mean of the prior is given by

$$\begin{aligned}\mathbb{E}[x_{n+1}, Y_n] &= \mathbb{E}[Ax_n + q_n|Y_n]\\ &= \mathbb{E}[Ax_n|Y_n]\\ &= A\hat{x}_n\end{aligned}$$

and the variance is given by

$$\begin{aligned}Var(x_{n+1}|Y_n) &= \mathbb{E}\left[(Ax_n + q_n - A\hat{x}_n)(Ax_n + q_n - A\hat{x}_n)^T|Y_n\right]\\ &= AP_nA^T + Q_n + 2\mathbb{E}[A(x_n - \hat{x}_n)q_n|Y_n]\\ &= AP_nA^T + Q_n\end{aligned}$$

When we combine the mean and variance we get the prior density function

$$Prior = p(x_{n+1}|Y_n) \sim \mathcal{N}(\hat{x}_n, AP_nA^T + Q_n) \tag{37}$$

B. Evidence density function

The mean of the evidence is given by

$$\begin{aligned}\mathbb{E}[y_n|Y_{n-1}] &= \mathbb{E}[Cx_n + r_{n-1}|Y_{n-1}]\\ &= C\mathbb{E}[x_n|Y_{n-1}]\\ &= C\hat{x}_n\end{aligned}$$

and the variance is given by

$$\begin{aligned}&Cob(y_n|Y_{n-1}) =\\ &= \mathbb{E}\left[(Cx_n + r_n - C\hat{x}_n)(Cx_n + r_n - C\hat{x}_n)^T|Y_{n-1}\right]\\ &= \mathbb{E}\left[(C(x_n - \hat{x}_n) + r_n)(C(x_n - \hat{x}_n) + r_n)^T|Y_{n-1}\right]\\ &= C(AP_{n-1}A^T + Q_{n-1})C^T + R_n\end{aligned}$$

When we combine the mean and variance we get the evidence density function

$$\begin{aligned}&Evidence =\\ &p(y_n|Y_{n-1}) \sim \mathcal{N}(C\hat{x}_n, C(AP_{n-1}A^T + Q_{n-1})C^T + R_n)\end{aligned} \tag{38}$$

C. Likelihood density function

The mean of the likelihood density function is given by

$$\begin{aligned}\mathbb{E}(y_n|x_n) &= \mathbb{E}\left[Cx_n + r_{n-1}|x_n\right]\\ &= C\mathbb{E}\left[x_n|x_n\right]\\ &= Cx_n\end{aligned}$$

and the variance is given by

$$\begin{aligned}&Cob(y_n|Y_{n-1}) =\\ &= \mathbb{E}\left[(Cx_n + r_n - Cx_n)(Cx_n + r_n - Cx_n)^T|Y_{n-1}\right]\\ &= R_n\end{aligned}$$

When we combine the mean and variance we get the likelihood density function

$$Likelihood = p(y_n|x_n) \sim \mathcal{N}(Cx_n, R_n) \tag{39}$$

D. Posterior density function

Substituting equations (37), (38) and (39) into Bayes' rule gives

$$p(x_n|Y_n, M) = \frac{p(y_n|x_n, M)}{p(y_n|Y_{n-1}, M)}p(x_n|Y_{n-1}, M) \tag{40}$$

$$= A_n \exp\left(-1/2(x_n - \hat{x}_n)P_n^{-1}(x_n - \hat{x}_n)\right) \tag{41}$$

where the coefficient A_n is given by

$$A_n = \frac{|C(AP_{n-1}A^T + Q_{n-1})C^T + R_{n-1}|^{1/2}}{(2\pi)^{q/2}|R_n|^{1/2}|AP_{n-1}A^T + Q_{n-1}|^{1/2}}$$

and $\hat{x}_n$ and P_n are given by

$$\begin{aligned}\hat{x}_n &= \hat{x}_{n-1} + K_n(y_n - C\hat{x}_{n-1})\\ P_n &= AP_{n-1}A^T + Q_{n-1} - K_nC(AP_{n-1}A^T + Q_{n-1})\end{aligned}$$

and the Kalman gain K_n is given by

$$K_n = \frac{(AP_{n-1}A^T + Q_{n-1})C^T}{R_n + C(AP_{n-1}A^T + Q_{n-1})C^T}$$

References

[1] W. D. Penny, "Kalman Filters," April 2000, signal Processing Course (lecture notes ch11). [Online]. Available: http://www.fil.ion.ucl.ac.uk/ wpenny/course/kalman.ps

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] N. de Freitas, M. Niranjan, and A. Gee, "Hierarchical bayesian-kalman models for regularisation and ard in sequential learning," DEPARTMENT OF ENGINEERING, CAMBRIDGE UNIVERSITY, Tech. Rep., 1998.

[4] S. Sarkka and A. Nummenmaa, "Preprint 1 recursive noise adaptive kalman filtering by variational bayesian approximations."

[5] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1008935410038

[6] H.-A. Loeliger, "An introduction to factor graphs," 2004.

[7] S. Korl, "A factor graph approach to signal modelling, system identification and filtering." Ph.D. dissertation, ETH Zurich, 2005, http://d-nb.info/976583933.

[8] A. Isler, *Parameter Identification of Bilinear Dynamical Systems: Expectation Maximization Using Factor Graphs*. ETH Zurich, Automatic Control Laboratory (IFA), 2011. [Online]. Available: http://books.google.nl/books?id=Q2pYMwEACAAJ