Kalman filter demystified: from intuition to probabilistic graphical model to real case in financial markets

Eric Benhamou

Kalman filter demystified: from intuition to probabilistic graphical model to real case in financial markets

Eric Benhamou*

A.I. SQUARE CONNECT,
35 Boulevard d'Inkermann 92200 Neuilly sur Seine, France

Affiliated researcher to LAMSADE (UMR CNRS 7243)
and QMI (Quantitative Management Initiative) chair,
Université Paris Dauphine,
Place du Maréchal de Lattre de Tassigny,75016 Paris, France

Abstract: In this paper, we revisit the Kalman filter theory. After giving the intuition on a simplified financial markets example, we revisit the maths underlying it. We then show that Kalman filter can be presented in a very different fashion using graphical models. This enables us to establish the connection between Kalman filter and Hidden Markov Models. We then look at their application in financial markets and provide various intuitions in terms of their applicability for complex systems such as financial markets. Although this paper has been written more like a self contained work connecting Kalman filter to Hidden Markov Models and hence revisiting well known and establish results, it contains new results and brings additional contributions to the field. First, leveraging on the link between Kalman filter and HMM, it gives new algorithms for inference for extended Kalman filters. Second, it presents an alternative to the traditional estimation of parameters using EM algorithm thanks to the usage of CMA-ES optimization. Third, it examines the application of Kalman filter and its Hidden Markov models version to financial markets, providing various dynamics assumptions and tests. We conclude by connecting Kalman filter approach to trend following technical analysis system and showing their superior performances for trend following detection.

Keywords and phrases: kalman filter, hidden markov models, graphical model, CMA ES, trend detection, systematic trading.

1. Introduction

One of the most challenging question in finance is to be able from past observation to make some meaningful guests. Obviously, as no one is God, no one would ultimately be able to predict with one hundred percent accuracy where for instance the price of a stock will end. However, between a pure blind quest and a perfect accuracy forecast, there is room for improvement and work. If in addition, we are able somehow to model the dynamics of the stock price and factor in some noise due to unpredictable human behavior, we can leverage this model information to make an informed guess. It will not be 100 percent accurate but it is much better than a pure random guess. Indeed, this scientific question of using a model and filtering noise has been extensively examined in various fields: control theory leading to Kalman filter, markov processes leading to hidden markov models and lately machine learning using bayesian probabilistic graphical models. In this work, we revisit these three fields to give a didactic presentation of this three approaches and

*eric.benhamou@aisquareconnect.com, eric.benhamou@dauphine.eu

emphasizing the profound connection between Kalman filter and Hidden Markov Models (HMM) thanks to Bayesian Probabilistic Graphical models. We show in particular that derivation of Kalman filter equations are much easier once we use the more general framework of Bayesian Probabilistic Graphical models. In particular, we show that Kalman filter equations are just a rewriting of the sum product algorithm (also referred to as the Viterbi algorithm for HMM). We then provide various dynamics assumptions for financial markets and comment their overall performance. We compare these trading system with simpler moving average trend detection trading systems and show that they provide better performances.

2. Intuition

In a nutshell, a Kalman filter is a method for predicting the future state of a system based on previous ones.
It was discovered in the early 1960's when Kalman introduced the method as a different approach to statistical prediction and filtering (see Kalman (1960) and Kalman and Bucy (1961)). The idea is to estimate the state of a noisy system. Historically, it was build to monitor the position and the speed of an orbiting vehicle. However, this can be applied to non physical system like economic system. In this particular settings, the state will be described by estimated equations rather than exact physical laws. We will emphasize this point in the section 5 dealing with practical application in finance.

To build an intuition, let us walk through a simple example - if you are given the data with green dots (that represent, by the way, the predicted Kalman filtered predicted price of the S&P500 index on October 17, 2018), it seems reasonable to predict that the orange dot should follow, by simply extrapolating the trend from previous samples and inferring some periodicity of the signal. However, how assured would you be anticipating the dark red point on the right (that is just 10 minutes later)? Moreover, how certain would you be about predicting the orange point, if you were given the black series (that represent the real prices) instead of the green one?

From this simple example, we can learn three important rules:

- Predicting far ahead in the future is less reliable than near ahead.
- The reliability of your data (the noise) influences the reliability of your forecast.
- It's not good enough to give a prediction - you also want to provide a confidence interval.

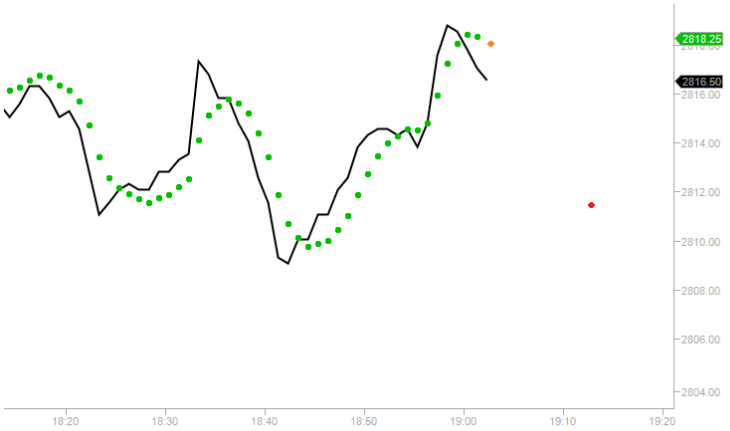


Fig 1: A simple example from financial markets to build an intuition of Kalman filter

From this simple presentation, we can conceptualize the approach and explain what a Kalman filter is. First, we need a **state**. A state is a representation of all the parameters needed to describe the current system and perform its prediction. In our previous example, we can define the state by two numbers: the price of the S&P 500 at time t, denoted by $p(t)$ and the slope of the price at time t, $s(t)$. In general, the state is a **vector**, commonly denoted $\mathbf{x}$. Of course, you can include many more parameters if you wish to accommodate more complex systems.

Second, we need a **model**. The model describes how the system behaves. It provides the underlying equations that rules our system. It may be an ideal representation or a simplified version of our system. It may be justified by some physical laws (in the case for instance of a Kalman filter for a GPS system), or by some empirical analysis (in the case of finance). In the standard Kalman filter, the model is always a linear function of the state. In extended Kalman filter, it is a non linear function of the state. In our previous example, our model could be:

- the price at time t, $p(t)$, is obtained as the previous price $p(t-1)$ plus the slope at time $t-1$: $s(t-1)$
$$p(t) = p(t-1) + s(t-1) \tag{2.1}$$
- the slope is evolving over time with a periodic sinusoidal function $\psi(t) = a\sin(bt+c)$
$$s(t) = s(t-1) + \psi(t) \tag{2.2}$$

Usually, one expresses this model in matrix form to make it simple. We have

$$\underbrace{\begin{pmatrix} p(t) \\ s(t) \end{pmatrix}}_{\mathbf{x}_t} = \underbrace{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}}_{\mathbf{F}} \cdot \underbrace{\begin{pmatrix} p(t-1) \\ s(t-1) \end{pmatrix}}_{\mathbf{x}_{t-1}} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \psi(t) \end{pmatrix}}_{\mathbf{B_t}\cdot\mathbf{u_t}} \tag{2.3}$$

In the above equation, following standard practice, we have denoted by $\mathbf{x}_t = (p(t), s(t))^T$, the **state vector** that combines the price $p(t)$ and the slope $s(t)$. We have denoted by $\mathbf{u}_t = (0, \psi(t))^T$, the **control vector**. The idea underlying the control vector is that it can control somehow the dynamic of the state vector. The previous equation (2.3) is called the **state equation** and writes

$$\mathbf{x}_t = \mathbf{F} \cdot \mathbf{x}_{t-1} + \mathbf{B}_t \cdot \mathbf{u}_t \tag{2.4}$$

Of course, our model is too simple, else we wouldn't need a Kalman Filter! To make it more realistic, we add an additional term - the **noise process**, $\mathbf{w_t}$ that represents anything else that we do not measure in our model. Although we don't know the actual value of the noise, we assume we can estimate how "noisy" the noise is and can make assumptions about the noise distribution. In standard Kalman modeling, we keep things simple and assumed this noise to be normally distributed. To measure how "noisy" the noise is, we refer to the variance of the normal distribution underpinning the noise $\mathbf{w_{t-1}}$. The **state equation** is modified into:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}_t \cdot \mathbf{u}_t + \mathbf{w}_{t-1} \tag{2.5}$$

Third, we need to use **measurement** to improve our model. When new data arrived, we would like to change the value of our model parameters to reflect our improved understanding of the state dynamic. We would proceed in two steps: make a prediction and then a correction based on what has happened in time t. There is a subtleties here. What we measure does not have to be exactly the states. It has to be related but not the same. For instance, in a Kalman filter system in a GPS, the state could be the GPS's acceleration, speed and position, while what we measure may be the car's position and the wheel velocity. In our financial example, we may only be able to observe the price but not the slope. Our measure would therefore be limited to the price in this particular setting. Hence, the measurement would be represented as follows:

$$\text{measurement} = \begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p(t) \\ s(t) \end{pmatrix} \tag{2.6}$$

In general, the measurement should be a vector, denoted by $\mathbf{z}$, as we may have more than one number for our measurement. Measurements should also be noisy to reflect that we do not measure perfectly. Hence, the equation specify measurement would write:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \tag{2.7}$$

Where $\mathbf{v}_t$ is the **measurement noise**, and H is a matrix with rows equal to the number of measurement variables and columns equal to the number of state variables.

Fourth, we need to work on **prediction** as we have set up the scene in terms of modeling. This is the key part in Kalman filter. To build intuition, let us assume, for the time being, noise is equal to zero. This implies our model is perfect. How would you predict the state at time t, $\mathbf{x}_t$ knowing the previous state (at time $t-1$)? This is dammed simple. Just use the state equation and compute our prediction of the state, denoted by $\hat{\mathbf{x}}_t$ as:

$$\hat{\mathbf{x}}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_{t-1} \tag{2.8}$$

Hold on! In the mean time, we do some measurements reflected by our measurement equation:

$$\hat{\mathbf{z}}_t = \mathbf{H}\hat{\mathbf{x}}_t \tag{2.9}$$

These are measurements so what we measure might be slightly different. Intuitively, the measurement error computed as $\mathbf{z}_t - \hat{\mathbf{z}}_t$ would be slightly different from 0. We would call this difference $\mathbf{y} = \mathbf{z}_t - \hat{\mathbf{z}}_t$, the **innovation**. It represents the bias of our measurement estimation. Obviously, if everything were perfect, the innovation should be zero, as we do not have any bias! To incorporate the innovation in our model, we would add it to our state. But we would not add blindly in the state

equation. We should multiply this innovation by a matrix factor that reflects somehow the correlation between our measurement bias and our state bias. Hence, we would compute a **correction** to our state equation as follows:

$$\hat{\mathbf{x}}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t + \mathbf{K}_t\mathbf{y} \tag{2.10}$$

The matrix $\mathbf{K}_t$ is called the **Kalman gain**. We would see in section 3 how to determine it but for the sake of intuition of Kalman filter, we skip this details. What really matters is to build an intuition. We can easily understand the following rules of thumb:

1. The more noisy our measurement is, the less precise it is. Hence, the innovation that represents the bias in our measurement may not be real innovation but rather an artifact of the measurement noise. Noise or uncertainty is directly captured by variance. Thus the larger the measurement variance noise, the lower the Kalman gain should be.
2. The more noisy our process state is, the more important the innovation should be taken into account. Hence, the larger the process state variance, the larger the Kalman gain should be.

Summarizing these two intuitions, we would expect the Kalman gain to be:

$$\mathbf{K}_t \sim \frac{\text{Process Noise}}{\text{Measurement Noise}} \tag{2.11}$$

Fifth, noise itself should be modeled as we have no clue about real noise. Noise in our model is represented by variance, or more precisely by the covariance of our state. Traditionally, we denote by $\mathbf{P}_t$ the covariance matrix of the state:

$$\mathbf{P}_t = \text{Cov}(\hat{\mathbf{x}}_t) \tag{2.12}$$

Using the state equation, and using the fact that for any matrix $\mathbf{A}$, $\text{Cov}(\mathbf{A}.\mathbf{X}) = \mathbf{A}\,\text{Cov}(\mathbf{X})\mathbf{A}^\top$, we can derive $\mathbf{P}_t$ from its previous state:

$$\mathbf{P}_t = \text{Cov}(\hat{\mathbf{x}}_t) = \text{Cov}(\mathbf{F}\mathbf{x}_{t-1}) = \mathbf{F}\,\text{Cov}(\mathbf{x}_{t-1})\mathbf{F}^\top = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^\top \tag{2.13}$$

At this stage, we can make the model even more realistic. Previous equation (2.13) assumes that our process model is perfect. But keep in mind that we temporarily assumed no noise. However, real world is more complex and we should now use our real state equation (2.5). In particular, if we assume that the state noise $\mathbf{w_t}$ is independent from the state $\mathbf{x_t}$ and if the state noise $\mathbf{w_t}$ is assumed to be distributed as a normal distribution with covariance matrix $\mathbf{Q}_t$, the prediction for our state covariance matrix becomes:

$$\mathbf{P}_t = \mathbf{F}\mathbf{P}_{t-1}\mathrm{F}^\top + \mathbf{Q}_{t-1} \tag{2.14}$$

Likewise, we could compute the covariance matrix for the measurement and model the evolution of noise that we will denote by $\mathbf{S}_t$. The last source of noise in our system is the measurement. Following the same logic we obtain a covariance matrix for $\hat{\mathbf{z}}_t$ and denoting by $\mathbf{v}_t$ the normally distributed independent noise for our measurements, with covariance matrix given by $\mathbf{R}_t$, we get

$$\mathbf{S}_t = \text{Cov}(\hat{\mathbf{z}}_t) = \text{Cov}(\mathbf{H}\hat{\mathbf{x}}_{t-1} + \mathbf{v}_{t-1}) = \mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^\top + \mathbf{R}_{t-1} \tag{2.15}$$

Let us come back to our Kalman gain computation. Intuitively, we found that it should be larger for larger process noise and lower for large measurement noise, leading to an equation of the type $\mathbf{K}_t \sim \frac{\text{Process Noise}}{\text{Measurement Noise}}$. As process noise is measured $\mathbf{P}_t$ and measurement noise by $\mathbf{S}_t$, we should

get something like $\mathbf{K}_t = \mathbf{P}_t\mathbf{S}_t^{-1}$. We shall see in the next section that the real equation for the Kalman gain is closely related to our intuition and given by

$$\mathbf{K}_t = \mathbf{P}_t\mathbf{H}^\top\mathbf{S}_t^{-1} \tag{2.16}$$

3. The maths

In this section, we will prove rigorously all the equations provided in section 2. To make notations more robust and make the difference for a time dependent vector or matrix $\mathbf{A}$ between its value at time t, knowing information up to time $t-1$, and its value at time t, knowing information up to time t, we will denote these two different value $\mathbf{A}_{t|t-1}$ and $\mathbf{A}_{t|t}$.

3.1. Kalman filter modeling assumption

The Kalman filter model assumes the true state at time t is obtained from the state at the previous time $t-1$ by the following state equation

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t + \mathbf{w}_t \tag{3.1}$$

The noise process $\mathbf{w}_t$ is assumed to follow a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{Q}_t$: $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$.

At time t, we make a measurement (or an observation) $\mathbf{z}_t$ of the true state $\mathbf{x}_t$ according to our measurement equation:

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t \tag{3.2}$$

Like for the state equation, we assume that the observation noise $\mathbf{v}_t$ follows a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{R}_t$: $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$. In addition, the initial state, and noise vectors at each step $\mathbf{x}_0, \mathbf{w}_1, \ldots, \mathbf{w}_t, \mathbf{v}_1, \ldots, \mathbf{v}_t$ are assumed to be all mutually independent.

3.2. Properties

It is immediate to derive the prediction phase as the estimated value of the state $\hat{\mathbf{x}}_{t|t-1}$ is simply the expectation of the state equation (2.5). Similarly, it is trivial to derive the estimated value for the error covariance. This provides the prediction phase that is summarized below

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t \qquad \text{(Predicted state estimate)} \tag{3.3}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t \qquad \text{(Predicted error covariance)} \tag{3.4}$$

The correction phase that consists in incorporating the measurements to correct our prediction

is slightly more tricky. It consists in the following equations:

$$\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1} \quad \text{(Measurement pre-fit residua)} \tag{3.5}$$
$$\mathbf{S}_t = \mathbf{R}_t + \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}} \quad \text{(Innovation covariance)} \tag{3.6}$$
$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{S}_t^{-1} \quad \text{(Optimal Kalman gain)} \tag{3.7}$$
$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t \quad \text{(Updated state estimate)} \tag{3.8}$$
$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1} \quad \text{(Updated estimate covariance)} \tag{3.9}$$
$$\tilde{\mathbf{y}}_{t|t} = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t} \quad \text{(Measurement post-fit residual)} \tag{3.10}$$

Proposition 1. Under the assumptions stated in subsection 3.1, the Kalman gain that minimizes the expected squared error defined as the square of the Euclidean (or L_2) norm of the error vector, representing the error between the true state and our estimated state : $\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}$ is given by

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{S}_t^{-1} \tag{3.11}$$

It is referred to as the optimal Kalman gain. For any Kalman gain (and not necessarily the optimal one), the estimate covariance updates as follows:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)^{\mathrm{T}} + \mathbf{K}_t\mathbf{R}_t\mathbf{K}_t^{\mathrm{T}} \tag{3.12}$$

For the optimal Kalman filter, this reduces to the usual Kalman filter Updated estimate covariance as follows:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1} \tag{3.13}$$

Proof. The derivation of all these formulae consists in three steps. First, we derive the posteriori estimate of the covariance matrix. Then we compute the Kalman gain as the minimum square error estimator. Third, we show that in the particular case of the optimal gain the equation for the updated estimate, covariance reduces to the formula provided in equation (3.9). All these steps are provided in appendix A to make the reading of this article smooth. □

4. Kalman as Graphical model

Historically, Hidden Markov Model (HMM) and Kalman filter were developed in distinct and unconnected research communities. Hence, their close relationship has not always been widely emphasized and appreciated. Part of the explanation lies also to the fact that the general framework for unifying these two approaches, namely graphical models came much later than HMM and Kalman filter. Without Bayesian graphical framework, the two algorithms underlying the inference calculation look rather different and unrelated. However, their difference is simply a consequence of the differences between discrete and continuous hidden variables and more specifically between multinomial and normal distribution. These details, important as they may be in practice, should not obscure us from the fundamental similarity between these two models. As we shall see in this section, the inference procedure for the state space model (SSM) shall prove us shortly that HMM and Kalman filter's model are cousin and share the same underlying graphical model structure, namely a hiddent state space variable and an observable variable. The interest of using Bayesian Probabilistic Graphical model is multiple. First, it emphasizes the general graphical model architecture underpinning both HMM and Kalman filter. Second, it provides modern computational tools used commonly in

machine learning to do the inference calculation. It shows how to generalize Kalman filter in the case of non Gaussian assumptions. It is interesting to realize that graphical models have been the marriage between probability theory and graph theory.

They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity – and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms.

From a literature point of view, Hidden Markov models were discussed as early as Rabiner and Juang (1986), and expanded on in Rabiner (1989). The first temporal extension of probabilistic graphical models is due to Dean and Kanazawa (1989), who also coined the term dynamic Bayesian network. Much work has been done on defining various representations that are based on hidden Markov models or on dynamic Bayesian networks; these include generalizations of the basic framework, or special cases that allow more tractable inference. Examples include mixed memory Markov models (see Saul and Jordan (1999)); variable-duration HMMs (Rabiner (1989)) and their extension segment models (Ostendorf et al. (1996)); factorial HMMs (Ghahramani and Jordan (1994)); and hierarchical HMMs (Fine et al. (1998) and Bui et al. (2002)). Smyth et al. (1997) is a review paper that was influential in providing a clear exposition of the connections between HMMs and DBNs. Murphy and Paskin (2001) show how hierarchical HMMs can be reduced to DBNs, a connection that provided a much faster inference algorithm than previously proposed for this representation. Murphy (2002) (and lately the book Murphy (2013)) provides an excellent tutorial on the topics of dynamic Bayesian networks and related representations as well as the non published book of Jordan (2016)

4.1. State space model

The state space model as emphasized for instance in Murphy (2013) is described as follows:

- there is a continuous chain of states denoted by $(\mathbf{x}_t)_{t=1,\ldots,n}$ that are non observable and influenced by past states only through the last realization. In other words, $\mathbf{x}_t$ is a Markov process, meaning $\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1}) = \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ Using graphical model, this can also be stated as given a state at one point in time, the states in the future are conditionally independent of those in the past.
- for each state, we can observe a space variable denoted by $\mathbf{z}_t$ that depends on the non observable space $\mathbf{x}_t$

Compared to the original presentation of the Kalman filter model, this is quite different. We now assume that there is an hidden variable (our state) and we can only measure a space variable. Since at each step, the space variable only depends on the non observable, there is only one arrow or edge between two latent variables horizontal nodes. This model is represented as a graphical model in figure 2.

Obviously, the graphical model provided in figure 2 can host both HMM and Kalman filter model. To make our model tractable, we will need to make additional assumptions. We will emphasize the ones that are common to HMM and Kalman filter models and the ones that differ. We impose the following conditions:

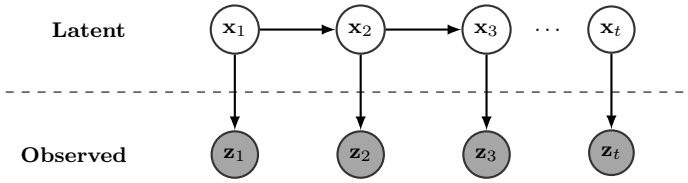


Fig 2: State Space model as a Bayesian Probabilistic Graphical model. Each vertical slice represents a time step. Nodes in white represent unobservable or latent variables called the states and denoted by $\mathbf{x}_t$ while nodes in gray observable ones and are called the spaces and denoted by $\mathbf{z}_t$. Each arrow indicates that there is a relationship between the arrow originating node and the arrow targeting node. Dots indicate that there is many time steps. The central dot line is to emphasize the fundamental difference between latent and observed variables

- The relationship between the state and the space is linear (this is our measurement equation in the Kalman filter and this is common to Kalman filter and HMM models):

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \tag{4.1}$$

  where the noise term $\mathbf{v}_t$ is assumed to follow a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{R}_t$.
- We will make the simplest choice of dependency between the state at time $t-1$ and t and assume that this is linear (this is our state equation and this is common to Kalman filter and HMM models)

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t \tag{4.2}$$

  where the noise term $\mathbf{w}_t$ is assumed to follow a multi dimensional normal distribution with zero mean and covariance matrix given by $\mathbf{Q}_t$ and where $\mathbf{B}_t \mathbf{u}_t$ is an additional trend term (that represents our control in Kalman filter). This control term is not common in HMM model but can be added without any difficulty. This results in slightly extended formula that we will signal. These formula are slight improvement of the common one found in the literature. Although, from a theoretical point of view, this control term may seem a futility, it is very important in practice and makes a big difference in numerical applications.
- We will assume as in the Kalman filter section that the initial state, and noise vectors at each step $\mathbf{x}_0, \mathbf{w}_1, \ldots, \mathbf{w}_t, \mathbf{v}_1, \ldots, \mathbf{v}_t$ are all mutually independent (this is common to HMM and Kalman filter models).
- Last but not least, we assume that the distribution of the state variable $\mathbf{x}_t$ follows a multi dimensional normal distribution. This is **Kalman filter specific**. For HMM, the state is assumed to follow a multinomial distribution.

The above assumptions restrict our initial state space model to a narrower class called the Linear-Gaussian SSM (LG-SSM). This model has been extensively studied and more can be found in Durbin and Koopman (2012) for instance.

Before embarking into the inference problem for the SSM, it is interesting to examine the unconditional distribution of the states $\mathbf{x}_t$. Using equation (4.2)

The unconditional mean of $\mathbf{x}_t$ is computed recursively

$$\prod_{k=2}^{t} \mathbf{F}_k \mathbf{x}_1 + \sum_{k=2}^{t} \prod_{l=k+1}^{t} \mathbf{F}_l \mathbf{B}_k \tag{4.3}$$

with the implicit assumption that empty product equals 1:

$$\prod_{l=t+1}^{t} \mathbf{F}_l = 1.$$

In the specific case of a null control term ($\mathbf{B}_t = 0$), the latter equation simplifies into

$$\prod_{k=2}^{t} \mathbf{F}_k \mathbf{x}_1 \tag{4.4}$$

The unconditional co-variance is computed as follows $\mathbf{P}_t = \mathbb{E}[\mathbf{x}_t \mathbf{x}_t^{\mathrm{T}}]$. Using our assumptions on independence as well as the state equation (4.2), we can compute it easily as:

$$\mathbf{P}_t = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t \tag{4.5}$$

This last equation remains unchanged in case of a non zero control term as the control term is deterministic. This last equation provides a dynamic equation for the unconditional variance and is referred to as the *Lyapunov equation*. It is also easy to checked that the unconditional covariance between neighboring states $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ is given by $\mathbf{F}_{t+1} \mathbf{P}_t \mathbf{F}_{t+1}^{\mathrm{T}}$.

4.2. Inference

The inference problem consists in calculating the posterior probability of the states given an output sequence. This calculation can be done both forward and backward. By forward, we mean that the inference information (called the evidence) at time t consists of the partial sequence of outputs up to time t. The backward problem is similar except that the evidence consists of the partial sequence of outputs after time t. Using standard graphical model terminology, we distinguish between filtering and smoothing problem.

In filtering, the problem is to calculate an estimate of the state $\mathbf{x}_t$ based on a partial output sequence $\mathbf{z}_0, \ldots, \mathbf{z}_t$. That is, we want to calculate $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_0, \ldots, \mathbf{z}_t)$. This is often referred to as the alpha recursion in HMM models (see for instance Rabiner and Juang (1986) and Rabiner (1989)).

Using standard Kalman filter notations, we shall denote by

$$\hat{\mathbf{x}}_{t|t} \triangleq \mathbb{E}[\mathbf{x}_t \mid \mathbf{z}_0, \ldots, \mathbf{z}_t] \tag{4.6}$$

$$\mathbf{P}_{t|t} \triangleq \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^{\mathrm{T}} \mid \mathbf{z}_0, \ldots, \mathbf{z}_t] \tag{4.7}$$

Under this settings, it is fairly easy to derive the following property that provides the conditional posterior distribution in the forward recursion. And to recover traditional results of Kalman filter, we shall decompose our time propagation into two steps:

- time update: $\mathbb{P}[\mathbf{x}_t \mid \mathbf{z}_0, \ldots, \mathbf{z}_t] \rightarrow \mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \ldots, \mathbf{z}_t]$

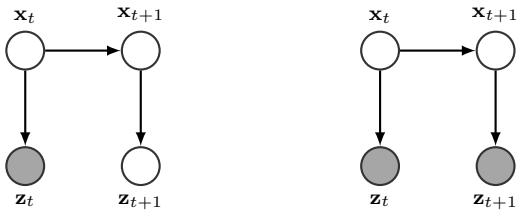


Fig 3: (a) A portion of the State Space Model before a measurement and (b) after a measurement update. White nodes are non observable variables while gray nodes are observed nodes.

- measurement update: $\mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \ldots, \mathbf{z}_t] \rightarrow \mathbb{P}[\mathbf{x}_{t+1} \mid \mathbf{z}_0, \ldots, \mathbf{z}_{t+1}]$

This can be represented nicely in terms of graphical models by the figure 3 below.

Proposition 2. Conditioned on past outputs $\mathbf{z}_0, \ldots, \mathbf{z}_t$, the variables $\mathbf{x}_{t+1}$ and $\mathbf{z}_{t+1}$ have a joint Gaussian distribution with mean and covariance matrix given by:

$$\begin{bmatrix} \hat{\mathbf{x}}_{t+1|t} \\ \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{P}_{t+1|t} & \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \\ \mathbf{H}_{t+1}\mathbf{P}_{t+1|t} & \mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1} \end{bmatrix} \tag{4.8}$$

Proof. This is trivial as the considered state space model is the Linear Gauss State Space Model (LGSSM). □

As simple as it may seem, the previous proposition makes our graphical model a full powerhouse as it provides the building block to start the inference. Indeed, knowing the conditional posterior distribution at step (a) of figure 3 gives us the first bullet to conclude for the step (b). Moreover, using results from simpler graphical models like the factor analysis graphical model, we can immediately conclude that the second step is given as follows

Proposition 3. Conditioned on past outputs $\mathbf{z}_0, \ldots, \mathbf{z}_{t+1}$, we have the following relationship

$$\begin{aligned} \hat{\mathbf{x}}_{t+1|t+1} &= \hat{\mathbf{x}}_{t+1|t} + \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \\ &\qquad (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t}) \qquad (4.9) \\ \mathbf{P}_{t+1|t+1} &= \mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1}\mathbf{H}_{t+1}\mathbf{P}_{t+1|t} \qquad (4.10) \end{aligned}$$

Proof. This is a direct consequence of the spatial model and can be found for instance in Murphy (2013) or Jordan (2016). We provide a self contained proof in appendix B □

Summarizing all these results leads to the seminal recursions of the Kalman filter as provided in the section 3 and given by the following proposition.

Proposition 4. Kalman filter consists in the following recursive equations:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \tag{4.11}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t \tag{4.12}$$

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t}) \tag{4.13}$$

$$\mathbf{P}_{t+1|t+1} = \mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \mathbf{H}_{t+1}\mathbf{P}_{t+1|t} \tag{4.14}$$

Proof. This is a trivial consequence of proposition 3. Equation (4.11) (resp. (4.12)) is the same as the one provided in (3.3) (resp. (3.4)) but using graphical models induction. □

Remark 4.1. If we introduce the following intermediate variables already provided in section 3:

$$\tilde{\mathbf{y}}_{t+1} = \mathbf{z}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t} \tag{4.15}$$

$$\mathbf{S}_{t+1} = \mathbf{R}_{t+1} + \mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \tag{4.16}$$

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{S}_{t+1}^{-1} \tag{4.17}$$

The equations (4.13) and (4.14) transform into equations

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}\tilde{\mathbf{y}}_{t+1} \tag{4.18}$$

$$\mathbf{P}_{t+1|t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\,\mathbf{P}_{t+1|t} \tag{4.19}$$

which are equations (3.8) and (3.9). This proves that the derivation using graphical models and control theory are mathematically equivalent!

Remark 4.2. There is nothing new to fancy at this stage except that we have shown with graphical models the Kalman filter recursion. And we can check this is way faster, easier and more intuitive. It is worth noticing that we have also multiple ways to write the gain matrix. Using the Sherman–Morrison–Woodbury formula, we can also derive various forms for the gain matrix as follows:

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} \left(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1}\right)^{-1} \tag{4.20}$$

$$= \left(\mathbf{P}_{t+1|t}^{-1} + \mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}\mathbf{H}_{t+1}\right)^{-1} \mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.21}$$

$$= \left(\mathbf{P}_{t+1|t} - \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}(\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}} + \mathbf{R}_{t+1})^{-1}\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\right) \mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.22}$$

$$= \mathbf{P}_{t+1|t+1}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1} \tag{4.23}$$

These forms may be useful whenever the reduced form (which is the last equation) is numerically unstable. Equation (4.23) is useful as it relates $\mathbf{K}_{t+1}$ to $\mathbf{P}_{t+1|t+1}$. It is interesting to notice the two form of the Kalman gain

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{S}_{t+1}^{-1} = \mathbf{P}_{t+1|t+1}\mathbf{H}_{t+1}^{\mathrm{T}}\mathbf{R}_{t+1}^{-1}$$

Remark 4.3. The Kalman filter appeals some remarks. We can first note that Kalman filtering equations can be interpreted differently. Combining equations (3.8) and (3.3), we retrieve an error correcting algorithm as follows:

$$\hat{\mathbf{x}}_{t|t} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{K}_t \left(z_t - \mathbf{H}_t (\mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t) \right)$$

or equivalently, regrouping the term $\hat{\mathbf{x}}_{t-1|t-1}$

$$\hat{\mathbf{x}}_{t|t} = \left(\mathbf{F}_t - \mathbf{K}_t \mathbf{H}_t \mathbf{F}_t \right) \hat{\mathbf{x}}_{t-1|t-1} + \left(\mathbf{B}_t \mathbf{u}_t + \mathbf{K}_t \left(z_t - \mathbf{H}_t \mathbf{B}_t \mathbf{u}_t \right) \right)$$

This shows us two things:

- Kalman filter can be seen as the discrete version of an Ornstein Uhlenbeck process
- Kalman filter can be seen as Auto Regressive process in discrete times

Since, recursive equation do appear similarly in Recursive Least Square (RLS) estimates, we also see here a connection with RLS. It is striking that these connections are not often made, mostly because Kalman filter was originally a control problem and not a statistician one.

As nice as the Kalman filter equation may look like, they have one major problem. It is the numerical stability of the filter. If the process noise covariance $\mathbf{Q}_t$ is small, round-off error would lead to obtain numerically a negative number for small positive eigenvalues of this matrix. As the scheme will propagate round off errors, the state covariance matrix $\mathbf{P}_t$ will progressively become only positive semi-definite (and hence indefinite) while it is theoretically a true positive definite matrix fully invertible.

In order to keep the positive definite property, we can slightly modify the recursive equation to find recursive equations that preserve the positive definite feature of the two covariance matrices. Since any positive definite matrix $\mathbf{S}^d$ can be represented and also reconstructed by its upper triangular square root matrix $\mathbf{R}$ with $\mathbf{S}^d = \mathbf{R}^t \mathbf{R}^{\mathrm{T}}$, with the strong property that representing in this form will guarantee that the resulting matrix will never get a negative eigen value, it is worth using a square root scheme with square root representation. Alternatively, we can also represent our covariance matrix using the so called unit diagonal (U-D) decomposition form, with $\mathbf{S}^d = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$ where $\mathbf{U}$' is a unit triangular matrix (with unit diagonal), and $\mathbf{D}$' is a diagonal matrix. This form avoids in particular many of the square root operations required by the matrix square root representation. Moreover, when comparing the two approaches, the U-D form has the elegant property to need same amount of storage, and somewhat less computation. This explains while the U-D factorization is often preferred Thornton (1976). A slight variation is the *LDL* decomposition of the innovation covariance matrix. The LDL decomposition relies on decomposing the covariance matrix $\mathbf{S}^d$ with two matrices: a lower unit triangular (unitriangular) matrix $\mathbf{L}$, and $\mathbf{D}$ a diagonal matrix. The algorithm starts with the LU decomposition as implemented in the Linear Algebra PACKage (LAPACK) and further it factors into the LDL form. Any singular covariance matrix is pivoted so that the first diagonal partition is always non-singular and well-conditioned (for further details see Bar-Shalom et al. (2002)).

4.3. Connection to information filter

It is worth showing the close connection with particle and information filter. This is a direct consequence of the fact that a multivariate Gaussian belongs to the exponential family and as such admits canonical parameters. Hence, we can rewrite the filter in terms of the later instead of the

initial usage of moment parameters. This is an interesting rewriting of the Kalman filter as it makes it numerically more stable. The canonical parameters of a multi variate Gaussian distribution, denoted by $\mathbf{\Lambda}$ and η, are obtained from the moment parameters Σ and μ as follows: $\mathbf{\Lambda} = \Sigma^{-1}$ and $\eta = \Sigma^{-1}\mu$. We are interested in deriving the canonical parameters of $\mathbf{x}_t$ first, at the prediction phase, conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$.

In the Kalman filter settings, the covariance Σ is denoted by $\mathbf{P}$ while the moment is given by $\mathbf{x}$ with the relationship $\eta = \mathbf{P}^{-1}\mathbf{x}$. Hence, we will write the precision matrix as $\mathbf{\Lambda}$ with the relationship with the covariance matrix given by $\mathbf{\Lambda} = \mathbf{P}^{-1}$. We shall write our new variables to be consistent with previous development as $\mathbf{\Lambda}_{t|t-1}$ and $\eta_{t|t-1}$. At the correction or measurement phase, we are interested in the same parameters but now conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_t$. We shall write them $\mathbf{\Lambda}_{t|t}$ and $\eta_{t|t}$. We can easily derive the following recursive scheme that is given by the proposition below:

Proposition 5. The filter equations are given by the following recursive scheme:

$$\begin{aligned}
\hat{\eta}_{t|t-1} &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1} \\
&\quad +(\mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t \qquad (4.24)\\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \qquad (4.25)\\
\mathbf{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} \qquad (4.26)\\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \qquad (4.27)
\end{aligned}$$

Proof. The derivation is easy and given in section B.3 □

Remark 4.4. The recursive equation for the information filter are very general. They include the control term $\mathbf{B}_t\mathbf{u}_t$ that is often neglected in literature introduced as early as 1979 in Anderson and Moore (1979). It is worth noticing that we can simplify computation by pre-computing a term $\mathbf{M}_t$ as follows:

$$\begin{aligned}
\mathbf{M}_t &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1} \qquad (4.28)\\
\mathbf{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1} - \mathbf{M}_t\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} \qquad (4.29)\\
\hat{\eta}_{t|t-1} &= \mathbf{M}_t\hat{\eta}_{t-1|t-1} + \mathbf{\Lambda}_{t|t-1}\mathbf{B}_t\mathbf{u}_t \qquad (4.30)\\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \qquad (4.31)\\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \qquad (4.32)
\end{aligned}$$

These equations are more efficient than the ones provided in proposition 5. As for the initialization of this recursion, we define the initial value as follows $\hat{\eta}_{1|0} = \hat{\eta}_1$ and $\mathbf{\Lambda}_{1|0} = \mathbf{\Lambda}_1$. It is interesting to note that the Kalman filter and the information filter are mathematically equivalent. They both share the same assumptions. However, they do not use the same parameters. Kalman filter (KF) uses moment parameters while particle or information filter (IF) relies on canonical parameters, which makes the later numerically more stable in case of poor conditioning of the covariance matrix. This is easy to understand as a small eigen value of the covariance translates into a large eigen value of the precision matrix as the precision matrix is the inverse of the covariance matrix. Reciprocally, a poor conditioning of the information filter should convert to a more stable scheme for the Kalman filter as the he condition number of a matrix is the reciprocal of the condition number of its inverse. Likewise, for initial condition as they are inverse, a small initial state in KF should translate to a large state in IF.

4.4. Smoothing

Another task we can do on dynamic Bayesian network is smoothing. It consists in obtaining estimates of the state at time t based on information from t on-wards (we are using future information in a sense). Like for HMM, the computation of this state estimate requires combining forward and backward recursion, either by starting by a backward-filtered estimates and then a forward-filtered estimates (an 'alpha-beta algorithm"), or by doing an algorithm that iterates directly on the filtered-and-smoothed estimates (an "alpha-gamma algorithm"). Both kinds of algorithm are available in the literature on state-space models (see for instance Koller and Friedman (2009) and Cappe et al. (2010) for more details), but the latter approach appears to dominate (as opposed to the HMM literature, where the former approach dominates) . The "alpha-gamma" approach is referred to as the "Rauch-Tung-Striebel (RTS) smoothing algorithm" (although it was developed using very different tool, namely control theory as early as 1965: see Rauch et al. (1965)) while the other approach is just the "alpha-beta" recursion.

4.4.1. Rauch-Tung-Striebel (RTS) smoother

The Rauch-Tung-Striebel (RTS) smoother developed in Rauch et al. (1965) relies precisely on the idea of of doing first a backward estimate and then a forward filter. Smoothing should not be confused with smoothing in times series analysis that is more or less a convolution. Smoothing for a Bayesian network means infering the distribution of a node conditioned on future information. It is the reciprocal of filtering that has also two meanings. Filtering for time series means doing a convolution to filter some noise. But filtering for Bayesian network means infering the distribution of a node conditioned on past information.

We can work out the RTS smoother and find the recursive equations provided by the following proposition

Proposition 6. The RTS smoothing algorithm works as follows:

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{4.33}$$

$$\hat{\mathbf{P}}_{t|T} = \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} \tag{4.34}$$

$$\text{where } \mathbf{L}_t = \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1} \tag{4.35}$$

with an initial condition given by

$$\hat{\mathbf{x}}_{T|T} = \hat{\mathbf{x}}_T \tag{4.36}$$

$$\hat{\mathbf{P}}_{T|T} = \hat{\mathbf{P}}_T \tag{4.37}$$

Proof. The proof consists in writing rigorously the various equations and is given in section B.4 □

4.4.2. Alternative to RTS filter

It is worth noting that we can develop an alternative approach to the RTS filter that relies on the alpha beta approach without any observation. This approach is quite standard for HMM models (see for instance Rabiner and Juang (1986) or Russell and Norvig (2009)). Following Kitagawa (1987),

this approach has been called in the literature the *two-filter algorithm*. It consists in combining the *forward* conditional probability $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_t)$ with the *backward* conditional probability $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_{t+1}, \ldots, \mathbf{z}_T)$. The intuition is illustrated by the figure 4.

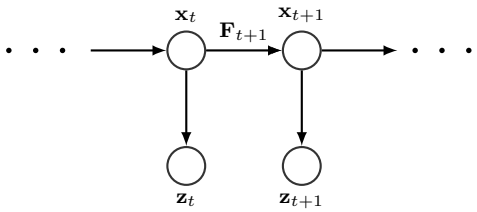


Fig 4: SSM with no observations

The graphical model provided by figure 4 can be easily characterized. The joint probability distribution of $(\mathbf{x}_t, \mathbf{x}_{t+1})$ is a multi variate normal whose Lyapunov equation (equation of the covariance matrix) is given by

$$\mathbf{P}_{t+1} = \mathbf{F}_{t+1}\mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} + \mathbf{Q}_{t+1} \tag{4.38}$$

Hence the covariance matrix of $(\mathbf{x}_t, \mathbf{x}_{t+1})$ is given by

$$\begin{bmatrix} \mathbf{P}_t & \mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} \\ \mathbf{F}_{t+1}\mathbf{P}_t & \mathbf{F}_{t+1}\mathbf{P}_t\mathbf{F}_{t+1}^{\mathrm{T}} + \mathbf{Q}_{t+1} \end{bmatrix} \tag{4.39}$$

The underlying idea in this approach is to invert the arrows in the graphical model leading to a new graphical model given by

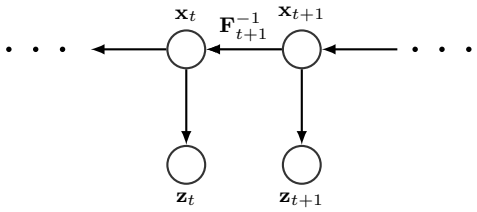


Fig 5: Inverted arrows for the SSM with no observations

Mathematically, this imply to change the relationship and now solve for $\mathbf{P}_t$ in terms of $\mathbf{P}_{t+1}$ using equation (4.38). We get:

$$\mathbf{P}_t = \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1} - \mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}} \tag{4.40}$$

where we have assumed that $\mathbf{F}_{t+1}$ is invertible. As a matter of fact, if it is not the case, we can rewrite everything with an approaching matrix to $\mathbf{F}_{t+1}$ (in the sense of the Frobenius norm) that is invertible, whose distance is less than ε, derive all the relationship below and then take the limit

as ϵ tends to zero. We will therefore assume in the following that $\mathbf{F}_{t+1}$ is invertible as this is not a strict condition for our derivation. Hence we can rewrite the covariance matrix now as follows:

$$\begin{bmatrix} \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1}-\mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}} & \mathbf{F}_{t+1}^{-1}(\mathbf{P}_{t+1}-\mathbf{Q}_{t+1}) \\ (\mathbf{P}_{t+1}-\mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-T} & \mathbf{P}_{t+1} \end{bmatrix} \tag{4.41}$$

If we define the new matrix $\widetilde{\mathbf{F}}_{t+1}$ as:

$$\widetilde{\mathbf{F}}_{t+1} = \mathbf{F}_{t+1}^{-1}(\mathbf{I}-\mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}) \tag{4.42}$$

we can simplify the covariance matrix as follows:

$$\begin{bmatrix} \widetilde{\mathbf{F}}_{t+1}\mathbf{P}_{t+1}\mathbf{F}_{t+1}^{-T} & \widetilde{\mathbf{F}}_{t+1}\mathbf{P}_{t+1} \\ \mathbf{P}_{t+1}\widetilde{\mathbf{F}}_{t+1}^{\mathrm{T}} & \mathbf{P}_{t+1} \end{bmatrix} \tag{4.43}$$

As this looks like a forward covariance matrix. Recall that the forward dynamics is defined by:

$$\mathbf{x}_{t+1} = \mathbf{F}_{t+1}\mathbf{x}_t + \mathbf{B}_{t+1}\mathbf{u}_{t+1} + w_{t+1} \tag{4.44}$$

The following proposition gives the inverse dynamics:

Proposition 7. The inverse dynamics is given by:

$$\mathbf{x}_t = \widetilde{\mathbf{F}}_{t+1}\mathbf{x}_{t+1} + \widetilde{\mathbf{B}}_{t+1}\mathbf{u}_{t+1} + \widetilde{w}_{t+1} \tag{4.45}$$

$$\text{with} \quad \widetilde{w}_{t+1} = -\mathbf{F}_{t+1}^{-1}(w_{t+1}-\mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}\mathbf{x}_{t+1}) \tag{4.46}$$

$$\widetilde{\mathbf{B}}_{t+1} = -\mathbf{F}_{t+1}^{-1}\mathbf{B}_{t+1} \tag{4.47}$$

and with $\widetilde{w}_{t+1}$ independent of the *past* information $\mathbf{x}_{t+1}, \ldots, \mathbf{x}_T$ and with the covariance matrix of $\widetilde{w}_{t+1}$ given by

$$\widetilde{\mathbf{Q}}_{t+1} \triangleq \mathbb{E}[\widetilde{w}_{t+1}\widetilde{w}_{t+1}^{\mathrm{T}}] = \mathbf{F}_{t+1}^{-1}\mathbf{Q}_{t+1}(\mathbf{I}-\mathbf{P}_{t+1}^{-1}\mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}}, \tag{4.48}$$

and with the forward Lyapunov equation given by:

$$\mathbf{P}_t = \widetilde{\mathbf{F}}_t\mathbf{P}_{t+1}\widetilde{\mathbf{F}}_t^{\mathrm{T}} + \widetilde{\mathbf{Q}}_{t+1} \tag{4.49}$$

Proof. The proof is straightforward and given in B.5. □

The reasoning followed here gives us a new way to generate an information filter but now backward. We should note that the conversion between the state space and the observable variable is unchanged and given by

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

We can now apply the information filter developed previously to get the new filtering equations using canonical instead of moment parameterization. We get the following new information filter :

Proposition 8. The modified Bryson–Frazier as presented in Bierman (2006) is given by:

$$\begin{aligned}
\widetilde{\mathbf{M}}_t &= \mathbf{F}_t^T \mathbf{Q}_t^{-1} (\mathbf{\Lambda}_{t+1|t+1} + \mathbf{Q}_t^{-1} - \mathbf{P}_t^{-1})^{-1} && (4.50)\\
\mathbf{\Lambda}_{t|t+1} &= \mathbf{F}_t^T (\mathbf{Q}_t - \mathbf{Q}_t \mathbf{P}_t^{-1} \mathbf{Q}_t)^{-1} \mathbf{F}_t - \widetilde{\mathbf{M}}_t \mathbf{Q}_t^{-1} \mathbf{F}_t && (4.51)\\
\hat{\eta}_{t|t+1} &= \widetilde{\mathbf{M}}_t \hat{\eta}_{t+1|t+1} - \mathbf{\Lambda}_{t|t+1} \mathbf{F}_t \mathbf{B}_t \mathbf{u}_t && (4.52)\\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}} \mathbf{R}_t^{-1} \mathbf{z}_t && (4.53)\\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}} \mathbf{R}_t^{-1} \mathbf{H}_t && (4.54)
\end{aligned}$$

Proof. The proof consists in combining all previous results and is given in B.6. □

Remark 4.5. If we want to convert to to the moment representation, we can use the inverse transform given by $\hat{\mathbf{x}}_{t|t+1} = \mathbf{S}_{t|t+1}^{-1} \hat{\eta}_{t|t+1}$ and $\mathbf{P}_{t|t+1} = \mathbf{S}_{t|t+1}^{-1}$

4.5. Inferring final posterior distribution

This problem consists in inferring the final posterior distribution $\mathbb{P}(\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_T)$. We can easily derive

$$\hat{\mathbf{x}}_{t|T} = \mathbf{P}_{t|T} (\mathbf{P}_{t|t}^{-1} \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t+1}^{-1} \hat{\mathbf{x}}_{t|t+1}) \tag{4.55}$$

and

$$\mathbf{P}_{t|T} = \left(\mathbf{P}_{t|t}^{-1} + \mathbf{P}_{t|t+1}^{-1} - \Sigma_t^{-1} \right)^{-1} \tag{4.56}$$

4.6. Parameter estimation

4.6.1. Intuition

Since the connection between HMM and Kalman filter, it has been established that the best way to estimate the initial parameters of a Kalman filter and any of its extension is to rely on EM estimation.

The expectation–maximization (EM) algorithm is an iterative method to find the maximum likelihood or maximum a posteriori (MAP) estimates of the parameters of a statistical model. This approach relies on the fact that the model depends on non observable (also called latent) variables. The EM algorithm alternates between performing an expectation (E) step, which provides the expectation of the conditional log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

The EM algorithm was explained and given its name in a classic 1977 paper by Dempster et al. (1977). The intuition is to find a way to solve the maximum likelihood solution for a model with latent variables. Since variables are hidden, solving the maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values and then solving the resulting equations. Because of latent variables, this function is unknown and the problem can not be solved. Instead, we compute the expected maximum likelihood with respect to the latent variables. We can then solve this explicit maximum likelihood function with respect to the observable variables.

4.6.2. EM Algorithm

We assume that our statistical model has a set $\mathbf{X}$ of observed data and a set of unobserved latent $\mathbf{Z}$ that depend on unknown parameters $\boldsymbol{\theta}$ that we want to determine thanks to maximum likelihood. We obviously know the likelihood function $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ given by

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) d\mathbf{Z}$$

The later is often intractable as the number of values for $\mathbf{Z}$ is vary large making the computation of the expectation (the integral) extremely difficult. In order to avoid the exact computation, we can work as follows:

First compute the expectation explicitly (called the *E step*) by taking the expected value of the log likelihood function of $\boldsymbol{\theta}$ with respect to latent variables $\mathbf{Z}$, denoted by $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}} \left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right]$$

We then maximize the resulted function with respect to the parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

The EM method works both for discrete or continuous random latent variables. It takes advantage of algoirthm like the Viterbi algorithm for hidden Markov models or recursive equation fo Kalman filter. The various steps are the following:

Algorithm 1 EM algorithm:

init: Set $\boldsymbol{\theta}$ to some random values.
while Not Converged **do**
  Compute the probability of each possible value of $\mathbf{Z}$, given $\boldsymbol{\theta}$ (E-step)
  Use the computed values of $\mathbf{Z}$ to find the argmax values for $\boldsymbol{\theta}$ (M-Step).
end while

The convergence of this algorithm is given by the following proposition.

Proposition 9. The EM algorithm converges monotonically to a local minimum for the marginal log likelihood.

Proof. There are at least two ways of proving this results that are provided in D.1 and D.2. □

4.7. Application to Kalman filter

The EM algorithm was initially developed for mixture models in particular Gaussian mixtures but also other natural laws from the exponential family such as Poisson, binomial, multinomial and exponential distributions as early as in Hartley (958). It was only once the link between latent variable and Kalman filter models was made that it became obvious that this could also be applied to Kalman and extended Kalman filter (see Cappe et al. (2010) or Einicke et al. (2010)). The EM method works as folllows:

Algorithm 2 EM algorithm for Kalman filter

init: Set the Kalman filter parameters to some random values.
while Not Converged **do**
Use Kalman filter recursive equations to obtained updated state estimates (E-step)
Use the filtered or smoothed state estimates within maximum-likelihood calculations to obtain updated parameter estimates. For instance for the standard Kalman filter, the distributions are assumed to be normal. The corresponding maximum likehood estimate for the updated measurement noise variance estimate leads to a variance given by

$$\hat{\sigma}_v^2 = \frac{1}{N}\sum_{k=1}^{N}(z_k - \hat{x}_k)^2$$

where $\hat{x}_k$ are the output estimates calculated by the Kalman filter for the measurements z_k. Likewise, the the updated process noise variance estimate is calculated as

$$\hat{\sigma}_w^2 = \frac{1}{N}\sum_{k=1}^{N}(\hat{x}_{k+1} - \hat{F}\hat{x}_k)^2$$

where $\hat{x}_k$ and $\hat{x}_{k+1}$ are scalar state estimates calculated by a filter or a smoother. The updated model coefficient estimate is obtained via

$$\hat{F} = \frac{\sum_{k=1}^{N}(\hat{x}_{k+1} - \hat{F}\hat{x}_k)}{\sum_{k=1}^{N}\hat{x}_k^2}$$

.
end while

4.7.1. CMA-ES estimation

Another radically difference approach is to minimize some cost function depending on the Kalman filter parameters. As opposed to the maximum likelihood approach that tries to find the best suitable distribution that fits the data, this approach can somehow factor in some noise and directly target a cost function that is our final result. Because our model is an approximation of the reality, this noise introduction may leads to a better overall cost function but a worse distribution in terms of fit to the data.

Let us first introduce the CMA-ES algorithm. Its name stands for covariance matrix adaptation evolution strategy. As it points out, it is an evolution strategy optimization method, meaning that it is a derivative free method that can accomodate non convex optimization problem. The terminology covariance matrix alludes to the fact that the exploration of new points is based on a multinomial distribution whose covariance matrix is progressively determined at each iteration. Hence the covariance matrix adapts in a sense to the sampling space, contracts in dimension that are useless and expands in dimension where natural gradient is steep. This algorithm has led to a large number of papers and articles and we refer to Varelas et al. (2018), Ollivier et al. (2017), Akimoto et al. (2016), Akimoto et al. (2015), Hansen and Auger (2014), Auger and Hansen (2012), Hansen and Auger (2011), Auger and Hansen (2009), Igel et al. (2007), Auger et al. (2004) to cite a few of the numerous articles around CMA-ES. We also refer the reader to the excellent wikipedia page Wikipedia (2018).

CMA-ES relies on two main principles in the exploration of admissible solution for our optimization problem. First, it relies on a multi variate normal distribution as this is the maximum entropy distribution given the first two moments. The mean of the multi variate distribution is updated at each step in order to maximize the likelihood of finding a successful candidate. The second moment, the covariance matrix of the distribution is also updated at each step to increase the likelihood of successful search steps. These updates can be interpreted as a natural gradient descent. Intuitively,

the CMA ES algorithim conducts an iterated principal components analysis of successful search steps while retaining all principal axes.

Second, we retain two paths of the successive distribution mean, called search or evolution paths. The underlying idea is keep significant information about the correlation between consecutive steps. If consecutive steps are taken in a similar direction, the evolution paths become long. The evolution paths are exploited in two ways. We use the first path is to compute the covariance matrix to increase variance in favorable directions and hence increase convergence speed. The second path is used to control step size and to make consecutive movements of the distribution mean orthogonal in expectation. The goal of this step-size control is to prevent premature convergence yet obtaining fast convergence to a local optimum.

In order to make it practical, we assume that we have a general cost function that depends on our Bayesian graphical model denoted by $\Phi(\theta)$ where θ are the parameters of our Kalman filter. Our cost function is for instance the Sharpe ratio corresponding to a generic trend detection strategy whose signal is generated by our Bayesian graphical model that is underneath a Kalman filter. This approach is more developed in a companion paper Benhamou (2018) but we will give here the general idea. Instead of computing the parameter of our Bayesian graphical model using the EM approach, we would like to find the parameters $\theta_{\max}$ that maximize our cost function $\Phi(\theta)$. Because our cost function is to enter a long trade with a predetermined target level and a given stop loss whenever our Bayesian graphical model anticipates a price risen and similarly to enter a short trade whenever our prediction based on Bayesian graphical model is a downside movement, our trading strategy is not convex neither smooth. It is a full binary function and generates spike whenever there is a trade. Moreover, our final criterium is to use the Sharpe ratio of the resulting trading strategy to compare the efficiency of our parameters. This is way too complicated for traditional optimization method, and we need to rely on Balck box optimization techniques like CMA-ES. Before presenting results, we will also discuss in the following section computational issues and the choice of the State space model dynamics.

We will describe below the most commonly used CMA -ES algorithm referred to as the $\mu/\mu_w, \lambda$ version. It is the version in which at each iteration step, we take a weighted combination of the μ best out of λ new candidate solutions to update the distribution parameters. The algorith has three main loops: first, it samples new solutions, second, it reorders the sampled solutions based on their fitness and third it updates the state variables. A pseudocode of the algorithm is provided just below:

Algorithm 3 CMA ES algorithm

Set λ ▷ number of samples / iteration
Initialize $m, \sigma, C = \mathbf{I}_n, p_\sigma = 0, p_c = 0$ ▷ initialize state variables

while (not terminated) **do**
  for $i = 1$ to λ **do** ▷ samples λ new solutions and evaluate them
    $x_i \sim \mathcal{N}(m, \sigma^2 C)$ ▷ samples multi variate normal
    $f_i = f(x_i)$ ▷ evaluates
  end for

  $x_{1\ldots\lambda} = x_{s(1)\ldots s(\lambda)}$ with $s(i) = \text{argsort}(f_{1\ldots\lambda}, i)$ ▷ reorders samples
  $m' = m$ ▷ stores current value of m

  $m = \text{update mean}(x_1, ..., x_\lambda)$ ▷ udpates mean to better solutions
  $p_\sigma = \text{update ps}(p_\sigma, \sigma^{-1} C^{-1/2}(m - m'))$ ▷ updates isotropic evolution path
  $p_c = \text{update pc}(p_c, \sigma^{-1}(m - m'), ||p_\sigma||)$ ▷ updates anisotropic evolution path
  $C = \text{update C}(C, p_c, (x_1 - m')/\sigma, ..., (x_\lambda - m')/\sigma)$ ▷ updates covariance matrix
  $\sigma = \text{update sigma}(\sigma, ||p_\sigma||)$ ▷ updates step-size using isotropic path length

  not terminated = iteration $\leq$ iteration max and $||m - m'|| \geq \varepsilon$ ▷ stop condition
end while

return m or x_1 ▷ returns solution

In this algorithm, for an n dimensional optimization program, the five state variables at iteration step k are:

1. $m_k \in \mathbb{R}^n$, the distribution mean and current best solution,
2. $\sigma_k > 0$, the variance step-size,
3. C_k, a symmetric positive-definite $n \times n$ covariance matrix initialized to $\mathbf{I}_n$,
4. $p_\sigma \in \mathbb{R}^n$, the isotropic evolution path, initially set to null,
5. $p_c \in \mathbb{R}^n$, the anisotropic evolution path, initially set to null.

It is worth noting that the order of the five update is important. The iteration starts with sampling $\lambda > 1$ candidate solutions $x_i \in \mathbb{R}^n$ from a multivariate normal distribution $\mathcal{N}(m_k, \sigma_k^2 C_k)$, that is $x_i \sim \ m_k + \sigma_k \mathcal{N}(0, C_k)$ for $i = 1, ..., \lambda$.

We then evaluate candidate solutions x_i for the objective function $f : \mathbb{R}^n \to \mathbb{R}$ of our optimization.

We then sort candidate solution according to their objective function value: $\{x_{i:\lambda} \mid i = 1 \ldots \lambda\}$ with $f(x_{1:\lambda}) \leq \ldots \leq f(x_{\mu:\lambda}) \leq \ldots \leq f(x_{\lambda:\lambda})$. It is worth noticing that we do not even need to know the value. Only the ranking is important for this algorithm.

The new mean value is updated as a weighted mean of our new candidate solutions

$$m_{k+1} = \sum_{i=1}^{\mu} w_i \, x_{i:\lambda} = m_k + \sum_{i=1}^{\mu} w_i \, (x_{i:\lambda} - m_k) \tag{4.57}$$

where the positive (recombination) weights $w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$ sum to one. Typically, $\mu \leq \lambda/2$ and the weights are chosen such that $\mu_w := 1/\sum_{i=1}^{\mu} w_i^2 \approx \lambda/4$.

The step-size σ_k is updated using cumulative step-size adaptation (CSA), sometimes also denoted as path length control. The evolution path p_σ is updated first as it is used in the update of the step-size σ_k:

$$p_\sigma = \underbrace{(1-c_\sigma)}_{\text{discount factor}}\ p_\sigma + \overbrace{\sqrt{1-(1-c_\sigma)^2}}^{\text{complements for discounted variance}} \underbrace{\sqrt{\mu_w}\, C_k^{-1/2}\, \overbrace{\frac{m_{k+1}-m_k}{\sigma_k}}^{\text{displacement of } m}}_{\text{distributed as } \mathcal{N}(0,I) \text{ under neutral selection}} \tag{4.58}$$

$$\sigma_{k+1} = \sigma_k \times \exp\left(\frac{c_\sigma}{d_\sigma}\underbrace{\left(\frac{\|p_\sigma\|}{E\|\mathcal{N}(0,I)\|}-1\right)}_{\text{unbiased about 0 under neutral selection}}\right) \tag{4.59}$$

where

- $c_\sigma^{-1} \approx n/3$ is the backward time horizon for the evolution path p_σ and larger than one ($c_\sigma \ll 1$ is reminiscent of an exponential decay constant as $(1-c_\sigma)^k \approx \exp(-c_\sigma k)$ where c_σ^{-1} is the associated lifetime and $c_\sigma^{-1}\ln(2) \approx 0.7 c_\sigma^{-1}$ the half-life),
- $\mu_w = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1}$ is the variance effective selection mass and $1 \le \mu_w \le \mu$ by definition of w_i,
- $C_k^{-1/2} = \sqrt{C_k}^{\,-1} = \sqrt{C_k^{-1}}$ is the unique symmetric square root of the inverse of C_k, and
- d_σ is the damping parameter usually close to one. For $d_\sigma = \infty$ or $c_\sigma = 0$ the step-size remains unchanged.

The step-size σ_k is increased if and only if $\|p_\sigma\|$ is larger than $\sqrt{n}\,(1-1/(4\,n)+1/(21\,n^2))$ and decreased if it is smaller (see Hansen (2006) for more details).

Finally, the covariance matrix is updated, where again the respective evolution path is updated first.

$$p_c \leftarrow \underbrace{(1-c_c)}_{\text{discount factor}}\ p_c + \underbrace{\mathbf{1}_{[0,\alpha\sqrt{n}]}(\|p_\sigma\|)}_{\text{indicator function}} \overbrace{\sqrt{1-(1-c_c)^2}}^{\text{complements for discounted variance}} \underbrace{\sqrt{\mu_w}\,\frac{m_{k+1}-m_k}{\sigma_k}}_{\text{distributed as } \mathcal{N}(0,C_k) \text{ under neutral selection}} \tag{4.60}$$

$$C_{k+1} = \underbrace{(1-c_1-c_\mu+c_s)}_{\text{discount factor}}\ C_k + c_1 \underbrace{p_c p_c^T}_{\text{rank one matrix}} + c_\mu \underbrace{\sum_{i=1}^{\mu} w_i \frac{x_{i:\lambda}-m_k}{\sigma_k}\left(\frac{x_{i:\lambda}-m_k}{\sigma_k}\right)^T}_{\text{rank } \min(\mu,n) \text{ matrix}} \tag{4.61}$$

where T denotes the transpose and

$c_c^{-1} \approx n/4$ is the backward time horizon for the evolution path p_c and larger than one,

$\alpha \approx 1.5$ and the indicator function $\mathbf{1}_{[0,\alpha\sqrt{n}]}(\|p_\sigma\|)$ evaluates to one if and only if $\|p_\sigma\| \in [0, \alpha\sqrt{n}]$ or, in other words, $\|p_\sigma\| \le \alpha\sqrt{n}$, which is usually the case,

$c_s = (1-\mathbf{1}_{[0,\alpha\sqrt{n}]}(\|p_\sigma\|)^2)\, c_1 c_c (2-c_c)$ makes partly up for the small variance loss in case the indicator is zero,

$c_1 \approx 2/n^2$ is the learning rate for the rank-one update of the covariance matrix and

$c_\mu \approx \mu_w/n^2$ is the learning rate for the rank-μ update of the covariance matrix and must not exceed $1-c_1$.

The covariance matrix update tends to increase the likelihood function for p_c and for $(x_{i:\lambda}-m_k)/\sigma_k$ to be sampled from $\mathcal{N}(0, C_{k+1})$. This completes the iteration step. The stop condition is quite standard and makes sure we stop if the best solution does not move or if we reached the maximum iterations number.

Remark 4.6. The number of candidate samples per iteration, λ, is an important and influential parameter. It highly depends on the objective function and can be tricky to be determined. Smaller values, like 10, tends to do more local search. Larger values, like 10 times the dimension n makes the search more global. A potential solution to the determination of the number of candidate samples per iteration is to repeatedly restart the algorithm with increasing λ by a factor of two at each restart (see Auger and Hansen (2005))

Besides of setting λ (or possibly μ instead, if for example λ is predetermined by the number of available processors), the above introduced parameters are not specific to the given objective function. This makes this algorithm quite powerful as the end user has relatively few parameters to set to do an efficient optimization search.

4.8. Computational issues

Although the computational cost of Kalman filter is very small and its complexity dammed simple, there are room for improvements. The two dominant costs in the Kalman filter are the matrix inversion to compute the Kalman gain matrix, $\mathbf{K}_t$, which takes $O(|\mathbf{x}_t|^3)$ time where $|\mathbf{x}_t|$ stands for the dimension of $\mathbf{x}_t$; and the matrix-matrix multiplication to compute the estimated covariance matrix which takes $O(|\mathbf{z}_t|^2)$ time.

In some applications (e.g., robotic mapping), we have $|\mathbf{z}_t \gg |\mathbf{x}_t|$, so the latter cost dominates. However, in such cases, we can sometimes use sparse approximations (see for instance Thrun et al. (2005)). In the opposite cases where $|\mathbf{x}_t| \gg |\mathbf{z}_t$, we can precompute the Kalman filter gain $\mathbf{K}_t$, since, it does not depend on the actual observations $(\mathbf{z}_1, \ldots, \mathbf{z}_t)$. This is a an unusual property that is specific to linear Gaussian systems.

The iterative equations for updating the estimate covariance in equation (3.9) or in (A.6) are Riccati equations as they are first order ordinary differential recursive equations quadratic in Kalman gain. Standard theory state that for time invariant systems, they converge to a fixed point. Hence, it may be wised to use this steady state solution instead of using a time-specific gain matrix in our filtering. In practice however, this does not provide better results. More sophisticated implementations of the Kalman filter should be used, for reasons of numerical stability. One approach is the information filter also called particle filter, which recursively updates the canonical parameters of the underlying Gaussian distribution, namely the precision matrix defined as the inverse of the covariance matrix $\Lambda_{t|t} = \mathbf{P}_{t|t}$ and the first order canonical parameter given by $\eta_{t|t} = \Lambda_{t|t}\mathbf{x}_{t|t}$. These parameters instead of the traditional moment parameters makes the inference stronger.

Another approach is the square root filter, which works with the Cholesky decomposition or the UDU factorization of the estimated error covariance: $\mathbf{P}_{t|t}$. This is much more numerically stable than directly updating the estimated error covariance $\mathbf{P}_{t|t}$. Further details can be found at the reference site http://www.cs.unc.edu/ welch/kalman/ and the reference book for Kalman filter Simon (2006).

5. In practice

Financial markets are complex dynamics models. Notoriously, one does not know the exact dynamics as opposed to a noise filtering problem for a GPS. We are doomed to make assumptions about the

dynamics of financial systems. This can seriously degrade the filter performance as we are left with some un-modeled dynamics. As opposed to noise that the filter can accommodate (it has been designed for it), non modeled dynamics depends on the input. They can bring the estimation algorithm to instability and divergence. It is a tricky and more experimental issue to distinguish between measurement noise and non modeled dynamics.

Kalman filter has been applied by many authors (see Javaheri et al. (2003), Lautier (2004), Bierman (2006), Bruder et al. (2011), Dao. (2011) Chan (2013) and Benhamou (2017)).

The following linear dynamics have been suggested:

$$\mathbf{x}_{t+1} = \Phi \mathbf{x}_t + c_t + w_t \tag{5.1}$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + d_t + v_t \tag{5.2}$$

with the following choice for the various parameters:

Model	Φ	$\mathbf{H}$	$\mathbf{Q}$	$\mathbf{R}$	$\mathbf{P}_{t=0}$	c_t
1	$\begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} p_1^2 & p_1 p_2 \\ p_1 p_2 & p_3^2 \end{bmatrix}$	$\begin{bmatrix} p_4 \end{bmatrix}$	$\begin{bmatrix} p_5 & 0 \\ 0 & p_5 \end{bmatrix}$	0
2	$\begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} p_1^2 & p_1 p_2 \\ p_1 p_2 & p_3^2 \end{bmatrix}$	$\begin{bmatrix} p_4 \end{bmatrix}$	$\begin{bmatrix} p_5 & 0 \\ 0 & p_6 \end{bmatrix}$	0
3	$\begin{bmatrix} p_1 & p_2 \\ 0 & p_3 \end{bmatrix}$	$\begin{bmatrix} p_4 \\ p_5 \end{bmatrix}$	$\begin{bmatrix} p_6^2 & p_6 p_7 \\ p_7 p_6 & p_8^2 \end{bmatrix}$	$\begin{bmatrix} p_9 \end{bmatrix}$	$\begin{bmatrix} p_{10} & 0 \\ 0 & p_{11} \end{bmatrix}$	0
4	$\begin{bmatrix} p_1 & p_2 \\ 0 & p_3 \end{bmatrix}$	$\begin{bmatrix} p_4 \\ p_5 \end{bmatrix}$	$\begin{bmatrix} p_6^2 & p_6 p_7 \\ p_7 p_6 & p_8^2 \end{bmatrix}$	$\begin{bmatrix} p_9 \end{bmatrix}$	$\begin{bmatrix} p_{10} & 0 \\ 0 & p_{11} \end{bmatrix}$	$\begin{bmatrix} p_{12}(p_{13} - K_t) \\ p_{14}(p_{15} - K_t) \end{bmatrix}$

TABLE 1
Various Kalman filter model specifications

6. Numerical experiments

In order to test our results, we look at the following trend following algorithm based on our Kalman filter algorithm where we enter a long trade if the prediction of our kalman filter is above the close of the previous day and a short trade if the same prediction is below the close of the previous day. For each comparison, we add an offset μ to avoid triggering false alarm signals. We set for each trade a pre-determined profit and stop loss target in ticks. These parameters are optimized in order to provide the best sharpe ratio over the train period together with the Kalman filter parameters. The pseudo code of our algorithm is listed below

Algorithm 4 Kalman filter Trend following algorithm

```
Initialize common trade details
SetProfitTarget( target)                                   ▷ fixed profit target in ticks
SetStopLoss( stop_loss )                                   ▷ fixed stop loss in ticks

while Not In Position do                                   ▷ look for new trade
    if KF( p_1, ..., p_n).Predict[0] ≥ Close[0] + μ then   ▷ up trend signal
        EnterLong()                                        ▷ market order for the open
    else if KF( p_1, ..., p_n).Predict[0] ≤ Close[0] + μ then   ▷ down trend signal
        EnterShort()                                       ▷ market order for the open
    end if
end while
```

Our resulting algorithm depends on the following parameters $p_1, \ldots, p_n$ the Kalman filter algorithm, the profit target, the stop loss and the signal offset μ. We could estimate the Kalman filter parameters with the EM procedure, then optimize the profit target, the stop loss and the signal offset μ. However, if by any chance the dynamics of the Kalman filter is incorrectly specified, the noise generated by this wrong specification will only be factored in the three parameters: the profit target, the stop loss and the signal offset μ. We prefer to do a combined optimization of all the parameters. We use daily data of the S&P 500 index futures (whose CQG code is *EP*) from 01Jan2017 to 01Jan2018. We train our model on the first 6 months and test it on the next six months. Deliberately, our algorithm is unsophisticated to keep thing simple and concentrate on the parameter estimation method. The overall idea is for a given set of parameter to compute the resulting sharpe ratio over the train period and find the optimal parameters combination. For a model like the fourth one in the table 1, the optimization encompasses 18 parameters: $p_1, \ldots, p_{15}$, the profit target, the stop loss and the signal offset μ, making it non trivial. We use the CMA-ES algorithm to find the optimal solution. In our optimization, we add some penalty condition to force non meaningful Kalman filter parameters to be zero, namely, we add a L1 penalty on this parameters.

Results are given below

TABLE 2
Optimal parameters

Parameters	p_{01}	p_{02}	p_{03}	p_{04}	p_{05}	p_{06}	p_{07}	p_{08}	p_{09}	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	offset	stop	target
Value	24.8	0	11.8	46.2	77.5	67	100	0	0	0	0	100	0	0	0	5	80	150

TABLE 3
Train test statistics 1/4

Performance	Net Profit	Gross Profit	Gross Loss	# of Trades	# of Contracts	Avg. Trade	Tot. Net Profit (%)	Ann. Net Profit (%)
Train	5,086 €	11,845 €	-6,759 €	15	15	339.05 €	5.09%	10.59%
Test	4,266 €	11,122 €	-6,857 €	15	15	284.38 €	4.27%	8.69%

TABLE 4
Train test statistics 2/4

Performance	Vol	Sharpe Ratio	Trades per Day	Avg. Time in Market	Max. Drawdown	Recovery Factor	Daily Ann. Vol	Monthly Ann. Vol
Train	6.54%	1.62	0.10	8d14h	-2,941 €	3.510	6.54%	5.72%
Test	6.20%	1.40	0.10	8d19h	-1,721 €	4.948	6.20%	5.32%

TABLE 5
Train test statistics 3/4

Performance	Daily Sharpe Ratio	Daily Sortino Ratio	Commission	Percent Profitable	Profit Factor	# of Winning Trades	Avg. Winning Trade	Max. conseq. Winners
Train	1.62	2.35	49 €	46.67%	1.75 €	7	1,692.09 €	3
Test	1.40	2.05	46 €	46.67%	1.62 €	7	1,588.92 €	2

TABLE 6
Train test statistics 4/4

Performance	Largest Winning Trade	# of Losing Trades	Avg. Losing Trade	Max. conseq. Losers	Largest Losing Trade	Avg. Win/Avg. Loss	Avg. Bars in Trade	Time to Recover
Train	1,776.11 €	8	-844.85 €	3	-1,011.82 €	2.00	6.1	77.00 days
Test	1,609.32 €	8	-857.1 €	2	-860.26 €	1.85	6.2	70.00 days

Fig 6: Kalman filter algorithm on train data set

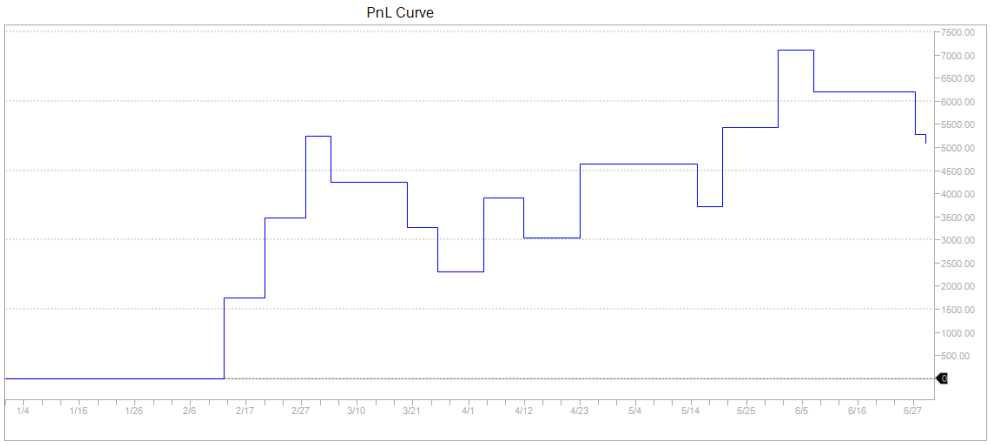
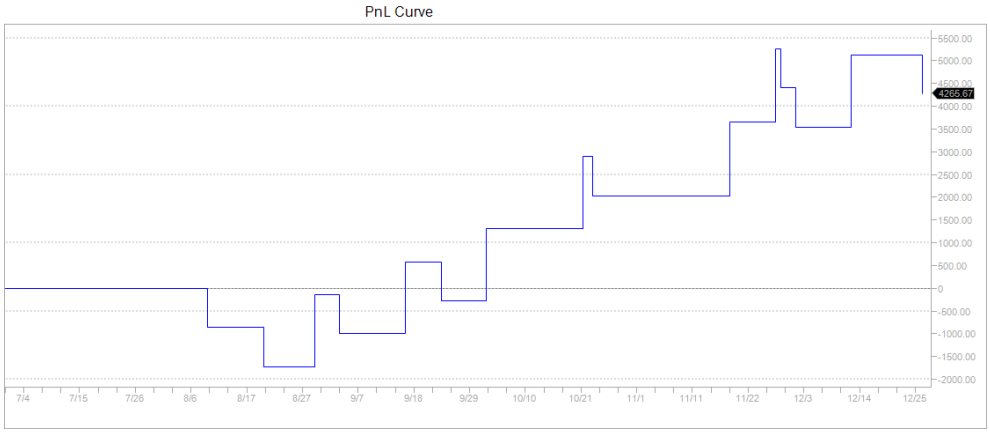


Fig 7: Kalman filter algorithm on test data set



We compare our algorithm with a traditional moving average crossover algorithm to test the efficiency of Kalman filter for trend detection. The moving average cross over algorithm generates a buy signal when the fast moving average crosses over the long moving average and a sell signal when the former crosses below the latter. A d period moving average is defined as the arithmetic average of the daily close over a d period, denoted by SMA(d). Our algorithm is given by the following pseudo code

Algorithm 5 Moving Average Trend following algorithm

```
Initialize common trade details
SetProfitTarget( target)                          ▷ fixed profit target in ticks
SetStopLoss( stop_loss )                          ▷ fixed stop loss in ticks

while Not In Position do                          ▷ look for new trade
    if SMA(Short)[0] > SMA(Long)[0] + offset then ▷ up trend signal
        EnterLong()                               ▷ market order for the open
    else if SMA(Short)[0] < SMA(Long)[0] + offset then ▷ down trend signal
        EnterShort()                              ▷ market order for the open
    end if
end while
```

We can now compare moving average cross over versus Kalman filter algorith. The table 7 compares the two algorithms. We can see that on the train period, the two algorithms have similar performances : $5,260$ vs $5,086$. However on the test period, moving average performs very badly with a net profit of 935 versus $4,266$ for the bayesian graphical model (the kalman filter) algorithm.

TABLE 7
Moving average cross over versus Kalman filter

Algo	Total Net Profit	Recovery Factor	Profit Factor	Max. Drawdown	Sharpe Ratio	Total # of Trades	Percent Profitable	Train: Total Net Profit
MA Cross over	935 €	0.32	1.13	-€2,889	0.41	26	0.54	5,260.00
Kalman filter	4,266 €	2.48	1.62	-€1,721	1.40	30	0.47	5,085.79

Fig 8: Moving Average Crossover algorithm on train data set

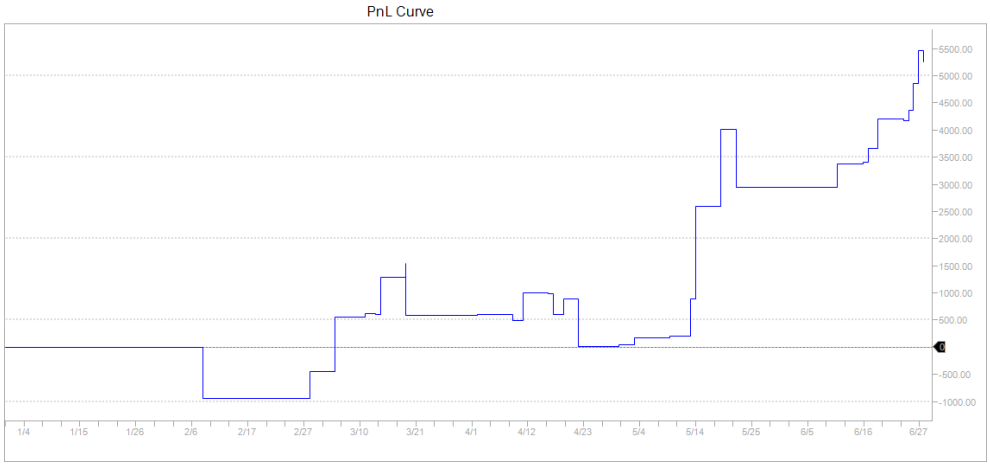
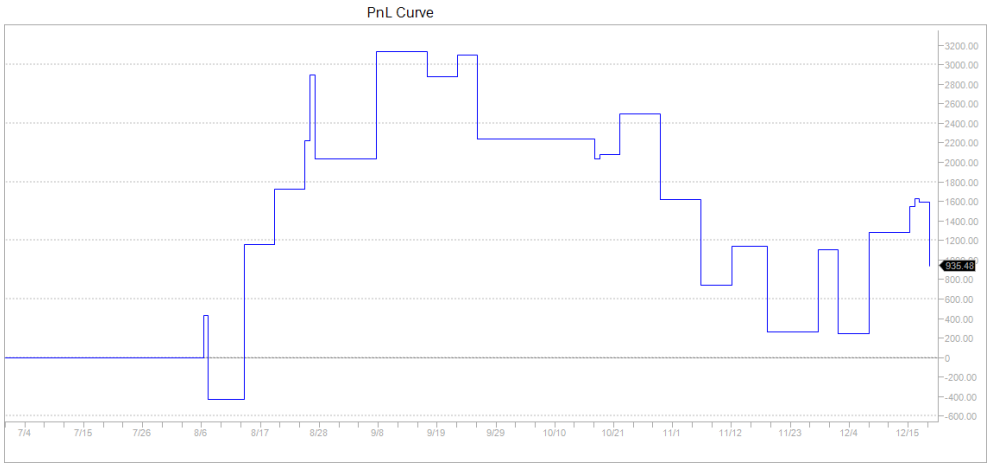


Fig 9: Moving Average Crossover algorithm on test data set



7. Conclusion

In this paper, we have revisited the Kalman filter theory and showed the strong connection between Kalman filter and Hidden Markov Models. After discussing various recursive scheme for the underlying Bayesian graphical model, we provided various insight on the parameter estimation. The traditional method is to use the expectation maximization (EM) method. This method aims to find the best fit distribution according to our modeling assumptions. This may work well when the state space model is ruled by physical law. However, in finance, there is no such a thing as the physical law of markets. One makes assumptions about the price dynamics that may be completely wrong. Hence, we present a new method based on CMA-ES optimization that does not decouple the Bayesian graphical model parameters from the other parameters of the financial strategy. We show on a numerical example on the S&P 500 index futures over one year of data (from Jan 1st 2017 to Jan 1st 2018) that the combined estimation of the Bayesian graphical model works well and that there is no much performance deterioration between train and test. This suggests little over fitting. We also show that the traditional moving average cross over method does not perform that well compared to our method and show strong over fitting bias as the net performance over the test data set is dramatically lower than the one on the train, suggesting over fitting.

Appendix A: Kalman filter proof

The proof is widely known and presented in various books (see for instance Gelb (1974) or Brown and Hwang (1997)). We present it here for the sake of completeness. It consists in three steps:

- Derive the posteriori estimate covariance matrix
- Compute the Kalman gain
- Simplify the posteriori error covariance formula

A.1. Step 1: A posteriori estimate covariance matrix

Proof. The error covariance $\mathbf{P}_{t|t}$ is defined as the covariance between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$: $\mathbf{P}_{t|t} = \text{Cov}\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\right)$. We can substitute in this equation the definition of $\hat{\mathbf{x}}_{t|t}$ given by equation (3.8) to get:

$$\mathbf{P}_{t|t} = \text{Cov}\left[\mathbf{x}_t - \left(\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t\right)\right] \tag{A.1}$$

Using the definition of $\tilde{\mathbf{y}}_t$ from equation (3.5), we get:

$$\mathbf{P}_{t|t} = \text{Cov}\left(\mathbf{x}_t - \left[\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}\right)\right]\right) \tag{A.2}$$

Using the measurement equation for $\mathbf{z}_t$ (equation (3.2)), this transforms into:

$$\mathbf{P}_{t|t} = \text{Cov}\left(\mathbf{x}_t - \left[\hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}\right)\right]\right) \tag{A.3}$$

By collecting the error vectors, this results in:

$$\mathbf{P}_{t|t} = \text{Cov}\left[\left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t\right)\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}\right) - \mathbf{K}_t\mathbf{v}_t\right] \tag{A.4}$$

Since the measurement error v_k is not correlated with the other terms, this simplifies into:

$$\mathbf{P}_{t|t} = \text{Cov}\left[\left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_k\right)\left(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}\right)\right] + \text{Cov}\left[\mathbf{K}_t\mathbf{v}_t\right] \tag{A.5}$$

By property of covariance for matrix: $\text{Cov}(A\mathbf{x}_t) = A\,\text{Cov}(\mathbf{x}_t)A^{\text{T}}$ and using the definition for $\mathbf{P}_{t|t-1}$ and $\mathbf{R}_t = \text{Cov}\left(\mathbf{v}_t\right)$, we get the final result:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)^{\text{T}} + \mathbf{K}_t\mathbf{R}_t\mathbf{K}_t^{\text{T}} \tag{A.6}$$

This last equation is referred to as the *Joseph form* of the covariance update equation. It is true for any value of Kalman gain $\mathbf{K}_t$ no matter if it is optimal or not. In the special case of optimal Kalman gain, this further reduces to equation (3.9) which we will prove in A.3 □

A.2. Step 2: Kalman gain

Proof. The goal of the Kalman filter is to find the minimum mean-square error estimator. It is obtained by minimizing the error of the *a posteriori* state given by $\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}$. This error is stochastic, hence we are left with minimizing the expected value of the square of the L_2 norm of this vector given by $\text{E}\left[\left\|\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\right\|^2\right]$. Since the error of the *a posteriori* state follows a normal distribution with zero mean and covariance given by $\mathbf{P}_{t|t}$, the optimization program is equivalent to minimizing

the matrix trace of the a posteriori estimate covariance matrix $\mathbf{P}_{t|t}$.

Expanding out and collecting the terms in equation (A.6), we find:

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{K}_t^{\mathrm{T}} + \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} \tag{A.7}$$

This minimization program is very simple. It is quadratic in the Kalman gain, $\mathbf{K}_t$. The optimum is obtained when the derivative with respect to $\mathbf{K}_t$ is null. The equation for the critical point writes as:

$$\frac{\partial \operatorname{tr}(\mathbf{P}_{t|t})}{\partial \mathbf{K}_t} = -2(\mathbf{H}_t\mathbf{P}_{t|t-1})^{\mathrm{T}} + 2\mathbf{K}_t\mathbf{S}_t = 0. \tag{A.8}$$

We can solving for $\mathbf{K}_t$ to find the *optimal* Kalman gain as follows:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{S}_t^{-1} \tag{A.9}$$

□

A.3. Step 3: Simplification of the posteriori error covariance formula

Proof. We can work on our formula further to simplify equation (A.6) as follows. A trick is to remark that in equation (A.9), multiplying on the right both sides by $\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}}$, we find that:

$$\mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{K}_t^{\mathrm{T}} \tag{A.10}$$

Injecting this equality in equation (A.7) and noticing that the last two terms cancel out, we get:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}. \tag{A.11}$$

This last formula appeals some remarks. It makes Kalman filter dammed simple. It is computationally cheap and much cheaper than the equation (A.7). Thus it is nearly always used in practice. However, this equation is only correct for the optimal gain. If by back luck, arithmetic precision is unusually low due to numerical instability, or if the Kalman filter gain is non-optimal (intentionally or not), equation A.11 does not hold any more. Using it would lead to inaccurate results and can be troublesome. In this particular case, the complete formula given by equation (A.7) must be used. □

Appendix B: Factor Analysis model

B.1. Model Presentation and proof

We are interested in the following very simple graphical model called the Gaussian Factor analysis model. It is represented by a two nodes graphical models in figure 10

We assume that X follows a multi dimensional normal distribution $X \sim \mathcal{N}(0, \Sigma)$. We assume that X is a latent variable and that only Z is observed. We also assume that there is a direct linear relationship between X and Z given by

Fig 10: Factor analysis model as a graphical model

$$Z = \mu + \Lambda X + W$$

where W is also distributed as a a multi dimensional normal distribution $W \sim \mathcal{N}(0, \Psi)$ independent of X. This is obviously a simplified version of our Linear Gaussian State Space Model and can provide building blocks for deriving the Kalman filter. Trivially, we can compute the unconditional distribution of Z as a normal distribution whose mean is $\mathbb{E}[Z] = \mathbb{E}[\mu + \Lambda X + W] = \mu$. Likewise, it is immediate to compute the unconditional covariance matrix of Z as

$$\begin{aligned} \text{Var}(Z) &= \mathbb{E}[(\mu + \Lambda X + W)(\mu + \Lambda X + W)^{\text{T}}] && \text{(B.1)} \\ &= \Lambda\Sigma\Lambda^{\text{T}} + \Psi && \text{(B.2)} \end{aligned}$$

The covariance between X and Z is also easy to compute and given by

$$\begin{aligned} \text{Cov}(X, Z) &= \mathbb{E}[X(\mu + \Lambda X + W)^{\text{T}}] && \text{(B.3)} \\ &= \Sigma\Lambda^{\text{T}} && \text{(B.4)} \end{aligned}$$

Hence, collecting all our results, we have shown that the joint distribution of X and Z is a Gaussian given by

$$\mathcal{N}\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma\Lambda^{\text{T}} \\ \Lambda\Sigma & \Lambda\Sigma\Lambda^{\text{T}} + \Psi \end{bmatrix}\right).$$

We shall be interested in computing the conditional distribution of X given Z. Using lemma provided in B.2, we obtained that $X \mid Z$ is a multi dimensional Gaussian whose mean is:

$$\mathbb{E}(X \mid z) = \Sigma\Lambda^{\text{T}}(\Lambda\Sigma\Lambda^{\text{T}} + \Psi)^{-1}(z - \mu) \quad \text{(B.5)}$$

Using the Woodbury matrix inversion formula (see formula (158) in Petersen and Pedersen (2012)), this is also

$$\mathbb{E}(X \mid z) = (\Sigma^{-1} + \Lambda^{\text{T}}\Psi^{-1}\Lambda^{-1})\Lambda^{\text{T}}\Psi^{-1}(z - \mu) \quad \text{(B.6)}$$

The two formula are equivalent and one should use the one that is easier to compute depending on the dimension of X and Z. Our partitioned Gaussian lemma B.2 gives us also the conditional variance of X:

$$\begin{aligned} \text{Var}(X \mid z) &= \Sigma - \Sigma\Lambda^{\text{T}}(\Lambda\Sigma\Lambda^{\text{T}} + \Psi)^{-1}\Lambda\Sigma && \text{(B.7)} \\ &= (\Sigma^{-1} + \Lambda^{\text{T}}\Psi^{-1}\Lambda)^{-1} && \text{(B.8)} \end{aligned}$$

where in the last equation, we have used again the Woodbury matrix inversion formula (Woodbury variant formula (157) in Petersen and Pedersen (2012)).

B.2. A quick lemma about partitioned Gaussian

Lemma B.1. Let Y be a multivariate normal vector $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Consider partitioning $\boldsymbol{Y}$, $\boldsymbol{\mu}$, Σ into

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Then, $(\boldsymbol{y}_1 | \boldsymbol{y}_2 = \boldsymbol{a})$, the conditional distribution of the first partition given the second, is $\mathcal{N}(\overline{\boldsymbol{\mu}}, \overline{\Sigma})$, with mean

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \Sigma_{12} {\Sigma_{22}}^{-1} (\boldsymbol{a} - \boldsymbol{\mu}_2)$$

and covariance matrix

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12} {\Sigma_{22}}^{-1} \Sigma_{21}$$

Proof. Let $\mathbf{y}_1$ be the first partition and $\mathbf{y}_2$ the second. Now define $\mathbf{z} = \mathbf{y}_1 + \mathbf{A}\mathbf{y}_2$ where $\mathbf{A} = -\Sigma_{12}\Sigma_{22}^{-1}$. We can easily compute the covariance between z and y as follows:

$$\begin{aligned} \mathrm{cov}(\mathbf{z}, \mathbf{y}_2) &= \mathrm{cov}(\mathbf{y}_1, \mathbf{y}_2) + \mathrm{cov}(\mathbf{A}\mathbf{y}_2, \mathbf{y}_2) \\ &= \Sigma_{12} + \mathbf{A}\,\mathrm{Var}(\mathbf{y}_2) \\ &= 0 \end{aligned}$$

Since $\mathbf{z}$ and $\mathbf{y}_2$ are jointly normal and uncorrelated, they are independent. The expectation of $\mathbf{z}$ is trivially computed as $E(\mathbf{z}) = \boldsymbol{\mu}_1 + \mathbf{A}\boldsymbol{\mu}_2$. Using this, we compute the conditional expectation of $\mathbf{y}_1$ given $\mathbf{y}_2$ as follows:

$$\begin{aligned} E(\mathbf{y}_1 | \mathbf{y}_2) &= E(\mathbf{z} - \mathbf{A}\mathbf{y}_2 | \mathbf{y}_2) \\ &= E(\mathbf{z} | \mathbf{y}_2) - E(\mathbf{A}\mathbf{y}_2 | \mathbf{y}_2) \\ &= E(\mathbf{z}) - \mathbf{A}\mathbf{y}_2 \\ &= \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \end{aligned}$$

which proves the first part. The second part is as simple. Note that

$$\begin{aligned} \mathrm{Var}(\mathbf{y}_1 | \mathbf{y}_2) &= \mathrm{Var}(\mathbf{z} - \mathbf{A}\mathbf{y}_2 | \mathbf{y}_2) \\ &= \mathrm{Var}(\mathbf{z} | \mathbf{y}_2) \\ &= \mathrm{Var}(\mathbf{z}) \end{aligned}$$

since $\mathbf{z}$ and $\mathbf{y}_2$ are independent. We can trivially conclude using the following last computation:

$$\begin{aligned} \mathrm{Var}(\mathbf{y}_1 | \mathbf{y}_2) = \mathrm{Var}(\mathbf{z}) &= \mathrm{Var}(\mathbf{y}_1 + \mathbf{A}\mathbf{y}_2) \\ &= \mathrm{Var}(\mathbf{y}_1) + \mathbf{A}\,\mathrm{Var}(\mathbf{y}_2)\mathbf{A}^{\mathrm{T}} + \mathbf{A}\mathrm{cov}(\mathbf{y}_1, \mathbf{y}_2) + \mathrm{cov}(\mathbf{y}_2, \mathbf{y}_1)\mathbf{A}^{\mathrm{T}} \\ &= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned}$$

□

B.3. Derivation of the filter equations

Proof. It is easy to start with the precision matrix (the inverse of the covariance matrix). We have

$$\begin{aligned}
\mathbf{\Lambda}_{t|t-1} &= \mathbf{P}_{t|t-1}^{-1} && \text{(B.9)}\\
&= (\mathbf{F}_t \mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t)^{-1} && \text{(B.10)}\\
&= \mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{P}_{t-1|t-1}^{-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} && \text{(B.11)}\\
&= \mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} && \text{(B.12)}
\end{aligned}$$

where in the previous equation, we have used extensively matrix inversion formula. Similarly, applying matrix inversion but for $\mathbf{\Lambda}_{t|t}$ (conditioned on t now)

$$\begin{aligned}
\mathbf{\Lambda}_{t|t} &= \mathbf{P}_{t|t}^{-1} && \text{(B.13)}\\
&= (\mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}} + \mathbf{R}_t)^{-1}\mathbf{H}_t\mathbf{P}_{t|t-1})^{-1} && \text{(B.14)}\\
&= \mathbf{P}_{t|t-1}^{-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.15)}\\
&= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.16)}
\end{aligned}$$

We can now handle the parameter η. We have

$$\begin{aligned}
\hat{\eta}_{t|t-1} &= \mathbf{P}_{t|t-1}^{-1}\hat{\mathbf{x}}_{t|t-1} && \text{(B.17)}\\
&= \mathbf{P}_{t|t-1}^{-1}\left(\mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t\right) && \text{(B.18)}\\
&= \mathbf{P}_{t|t-1}^{-1}\left(\mathbf{F}_t\mathbf{P}_{t-1|t-1}\hat{\eta}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t\right) && \text{(B.19)}\\
&= (\mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t)^{-1}\left(\mathbf{F}_t\mathbf{P}_{t-1|t-1}\hat{\eta}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t\right) &&\\
&= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{P}_{t-1|t-1}^{-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1} && \text{(B.20)}\\
&\qquad + (\mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t)^{-1}\mathbf{B}_t\mathbf{u}_t && \text{(B.21)}\\
&= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1} &&\\
&\quad + (\mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t && \text{(B.22)}
\end{aligned}$$

Likewise, we derive the same type of equations for η but conditioned on t as follows:

$$\begin{aligned}
\hat{\eta}_{t|t} &= \mathbf{P}_{t|t}^{-1}\hat{\mathbf{x}}_{t|t} && \text{(B.23)}\\
&= \mathbf{P}_{t|t}^{-1}(\hat{\mathbf{x}}_{t|t-1} + \mathbf{P}_{t|t}\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1})) && \text{(B.24)}\\
&= (\mathbf{P}_{t|t}^{-1} - \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t)\mathbf{P}_{t|t-1}\hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.25)}\\
&= (\mathbf{P}_{t|t-1}^{-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t - \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t)\mathbf{P}_{t|t-1}\hat{\eta}_{t|t-1} &&\\
&\qquad + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.26)}\\
&= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.27)}
\end{aligned}$$

Summarizing all these equation leads to the so called filter equations:

$$\begin{aligned}
\hat{\eta}_{t|t-1} &= \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\hat{\eta}_{t-1|t-1} \\
&\quad +(\mathbf{Q}_t^{-1}-\mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1}+\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1})\mathbf{B}_t\mathbf{u}_t \qquad \text{(B.28)} \\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t \qquad \text{(B.29)} \\
\mathbf{\Lambda}_{t|t-1} &= \mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{F}_t(\mathbf{\Lambda}_{t-1|t-1} + \mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1}\mathbf{F}_t)^{-1}\mathbf{F}_t^{\mathrm{T}}\mathbf{Q}_t^{-1} \qquad \text{(B.30)} \\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t \qquad \text{(B.31)}
\end{aligned}$$

which concludes the proof. □

B.4. Proof of RTS recursive equations

Proof. We start by writing down the distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ conditioned on $\mathbf{z}_1, \ldots, \mathbf{z}_t$. As $\hat{\mathbf{x}}_{t+1|t} = \mathbf{F}_{t+1}\mathbf{x}_t + \mathbf{B}_{t+1}\mathbf{u}_{t+1}$ and using the fact that the control term $\mathbf{B}_{t+1}\mathbf{u}_{t+1}$ is deterministic, we have

$$\mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t})^{\mathrm{T}} \mid \mathbf{z}_1, \ldots, \mathbf{z}_t] = \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}} \qquad \text{(B.32)}$$

Thus the vector distribution of $\begin{bmatrix} \mathbf{x}_{t|t} \\ \mathbf{x}_{t+1|t} \end{bmatrix}$ has its mean given by $\begin{bmatrix} \hat{\mathbf{x}}_{t|t} \\ \hat{\mathbf{x}}_{t+1|t} \end{bmatrix}$ and its covariance matrix given by $\begin{bmatrix} \mathbf{P}_{t|t} & \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}} \\ \mathbf{F}_{t+1}\mathbf{P}_{t|t} & \mathbf{P}_{t+1|t} \end{bmatrix}$

Let us now work on the backward recursion. We are interested in computing the distribution of $\mathbf{x}_t$, conditioned on $\mathbf{x}_{t+1}$ and $\mathbf{z}_1, \ldots, \mathbf{z}_t$. Using the Factor analysis model, it is easy to see that

$$\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] = \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \qquad \text{(B.33)}$$

Similarly, we have

$$\mathrm{Var}\left[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t\right] = \mathbf{P}_{t|t} - \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}\mathbf{F}_{t+1}\mathbf{P}_{t|t} \qquad \text{(B.34)}$$

But using the Markov property that states that

$$\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] = \mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] \qquad \text{(B.35)}$$

and

$$\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_t] = \mathrm{Var}(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \qquad \text{(B.36)}$$

We have

$$\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] = \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \qquad \text{(B.37)}$$

Similarly, we have

$$\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] = \mathbf{P}_{t|t} - \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}\mathbf{F}_{t+1}\mathbf{P}_{t|t}^{-1} \qquad \text{(B.38)}$$

Using conditional properties states in appendix section C.2, we can work the recursion as follows:

$$\begin{aligned}
\hat{\mathbf{x}}_{t|T} &\triangleq \mathbb{E}[\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.39)}\\
&= \mathbb{E}[\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.40)}\\
&= \mathbb{E}[\hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.41)}\\
&= \hat{\mathbf{x}}_{t|t} + \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1}(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) && \text{(B.42)}\\
&= \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) && \text{(B.43)}
\end{aligned}$$

where

$$\mathbf{L}_t = \mathbf{P}_{t|t}\mathbf{F}_{t+1}^{\mathrm{T}}\mathbf{P}_{t+1|t}^{-1} \quad \text{(B.44)}$$

The latter equation is the basic update equation in the RTS smoothing algorithm. This equation provides an estimate of $\mathbf{x}_t$ based on the filtered estimate $\hat{\mathbf{x}}_{t|t}$ corrected by the convolution of $\mathbf{L}_t$ with the error term $\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}$ that represents the difference between the smoothed estimate of $\mathbf{x}_T$ and the filtered estimate $\hat{\mathbf{x}}_{t+1|t}$. The matrix $\mathbf{L}_t$ can be interpreted as a gain matrix that depends only on forward information. and can be computed in the forward pass.

As for the conditional variance, we have

$$\begin{aligned}
\hat{\mathbf{P}}_{t|T} &\triangleq \mathrm{Var}[\mathbf{x}_t \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.45)}\\
&= \mathrm{Var}[\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.46)}\\
&\quad + \mathbb{E}[\mathrm{Var}[\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{z}_1, \ldots, \mathbf{z}_T] \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.47)}\\
&= \mathrm{Var}[\mathbf{x}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.48)}\\
&\quad + \mathbb{E}[\mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} \mid \mathbf{z}_1, \ldots, \mathbf{z}_T] && \text{(B.49)}\\
&= \mathbf{L}_t\,\mathrm{Var}[\mathbf{x}_{t|t} \mid \mathbf{z}_1, \ldots, \mathbf{z}_T]\mathbf{L}_t^{\mathrm{T}} + \mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} && \text{(B.50)}\\
&= \mathbf{L}_t\mathbf{P}_{t+1|T}\mathbf{L}_t^{\mathrm{T}} + \mathbf{P}_{t|t} - \mathbf{L}_t\mathbf{P}_{t+1|t}\mathbf{L}_t^{\mathrm{T}} && \text{(B.51)}\\
&= \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} && \text{(B.52)}
\end{aligned}$$

We can summarize the RTS smoothing algorithm as follows:

$$\begin{aligned}
\hat{\mathbf{x}}_{t|T} &= \hat{\mathbf{x}}_{t|t} + \mathbf{L}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) && \text{(B.53)}\\
\hat{\mathbf{P}}_{t|T} &= \mathbf{P}_{t|t} + \mathbf{L}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{L}_t^{\mathrm{T}} && \text{(B.54)}
\end{aligned}$$

with an initial condition given by

$$\begin{aligned}
\hat{\mathbf{x}}_{T|T} &= \hat{\mathbf{x}}_T && \text{(B.55)}\\
\hat{\mathbf{P}}_{T|T} &= \hat{\mathbf{P}}_T && \text{(B.56)}
\end{aligned}$$

□

B.5. Proof of the inverse dynamics

Proof. We have

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{F}_{t+1}^{-1}\left(\mathbf{x}_{t+1} - \mathbf{B}_{t+1}\mathbf{u}_{t+1} - w_{t+1}\right) && \text{(B.57)}\\
&= \widetilde{\mathbf{F}}_{t+1}\mathbf{x}_{t+1} + \mathbf{F}_{t+1}^{-1}\left((\mathbf{I} - \mathbf{F}_{t+1}\widetilde{\mathbf{F}}_{t+1})\mathbf{x}_{t+1} - \mathbf{B}_{t+1}\mathbf{u}_{t+1} - w_{t+1}\right) && \text{(B.58)}\\
&= \widetilde{\mathbf{F}}_{t+1}\mathbf{x}_{t+1} + \mathbf{F}_{t+1}^{-1}\left(\mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}\mathbf{x}_{t+1} - \mathbf{B}_{t+1}\mathbf{u}_{t+1} - w_{t+1}\right) && \text{(B.59)}\\
&= \widetilde{\mathbf{F}}_{t+1}\mathbf{x}_{t+1} - \mathbf{F}_{t+1}^{-1}\mathbf{B}_{t+1}\mathbf{u}_{t+1} - \mathbf{F}_{t+1}^{-1}\left(w_{t+1} - \mathbf{Q}_{t+1}\mathbf{P}_{t+1}^{-1}\mathbf{x}_{t+1}\right) && \text{(B.60)}
\end{aligned}$$

which provides the first result 4.45.

Using the forward dynamics (equation (4.44)) that relates $\mathbf{x}_{t+1}$ and w_{t+1}, we get that

$$\mathbb{E}[\widetilde{w}_{t+1}\widetilde{w}_{t+1}^{\mathrm{T}}] = \mathbf{F}_{t+1}^{-1}\mathbf{Q}_{t+1}(\mathbf{I} - \mathbf{P}_{t+1}^{-1}\mathbf{Q}_{t+1})\mathbf{F}_{t+1}^{-\mathrm{T}}, \quad \text{(B.61)}$$

which proves 4.48.

The independence between $\widetilde{w}_{t+1}$ and the past information $\mathbf{x}_{t+1}, \ldots, \mathbf{x}_T$ is obvious for $\mathbf{x}_{t+2}, \ldots, \mathbf{x}_T$ and inferred from the fact that the two processes are Gaussian with zero correlation: $\mathbb{E}[\widetilde{w}_{t+1}\mathbf{x}_{t+1}] = -\mathbf{F}_{t+1}^{-1}(\mathbf{Q}_{t+1} - \mathbf{Q}_{t+1}) = 0$.

Last but not least, using the independence, the forward Lyapunov equation is trivially derived, which concludes the proof. □

B.6. Proof of modified Bryson–Frazier equations

Proof. Applying the information filter given in proposition 5 with the forward dynamics equation 4.45 gives:

$$\begin{aligned}
\widetilde{\mathbf{M}}_t &= \widetilde{\mathbf{Q}}_t^{-1}\widetilde{\mathbf{F}}_t(\mathbf{\Lambda}_{t+1|t+1} + \widetilde{\mathbf{F}}_t^{\mathrm{T}}\widetilde{\mathbf{Q}}_t^{-1}\widetilde{\mathbf{F}}_t)^{-1} && \text{(B.62)}\\
\mathbf{\Lambda}_{t|t+1} &= \widetilde{\mathbf{Q}}_t^{-1} - \widetilde{\mathbf{M}}_t\widetilde{\mathbf{F}}_t^{\mathrm{T}}\widetilde{\mathbf{Q}}_t^{-1} && \text{(B.63)}\\
\hat{\eta}_{t|t+1} &= \widetilde{\mathbf{M}}_t\hat{\eta}_{t+1|t+1} + \mathbf{\Lambda}_{t|t+1}\widetilde{\mathbf{B}}_t\mathbf{u}_t && \text{(B.64)}\\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{z}_t && \text{(B.65)}\\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{H}_t && \text{(B.66)}
\end{aligned}$$

We can notice that

$$\begin{aligned}
\widetilde{\mathbf{Q}}_t^{-1}\widetilde{\mathbf{F}}_t &= \mathbf{F}_t^T\mathbf{Q}_t^{-1} && \text{(B.67)}\\
\widetilde{\mathbf{F}}_t^{\mathrm{T}}\widetilde{\mathbf{Q}}_t^{-1}\widetilde{\mathbf{F}}_t &= \mathbf{Q}_t^{-1} - \mathbf{P}_t^{-1} && \text{(B.68)}\\
\widetilde{\mathbf{F}}_t^{\mathrm{T}}\widetilde{\mathbf{Q}}_t^{-1} &= \mathbf{Q}_t^{-1}\mathbf{F}_t && \text{(B.69)}\\
& && \text{(B.70)}
\end{aligned}$$

Hence, if we rewrite everything in terms of the initial variables, we get

$$\begin{aligned}
\widetilde{\mathbf{M}}_t &= \mathbf{F}_t^T \mathbf{Q}_t^{-1} (\mathbf{\Lambda}_{t+1|t+1} + \mathbf{Q}_t^{-1} - \mathbf{P}_t^{-1})^{-1} && \text{(B.71)} \\
\mathbf{\Lambda}_{t|t+1} &= \mathbf{F}_t^T (\mathbf{Q}_t - \mathbf{Q}_t \mathbf{P}_t^{-1} \mathbf{Q}_t)^{-1} \mathbf{F}_t - \widetilde{\mathbf{M}}_t \mathbf{Q}_t^{-1} \mathbf{F}_t && \text{(B.72)} \\
\hat{\eta}_{t|t+1} &= \widetilde{\mathbf{M}}_t \hat{\eta}_{t+1|t+1} - \mathbf{\Lambda}_{t|t+1} \mathbf{F}_t \mathbf{B}_t \mathbf{u}_t && \text{(B.73)} \\
\hat{\eta}_{t|t} &= \hat{\eta}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}} \mathbf{R}_t^{-1} \mathbf{z}_t && \text{(B.74)} \\
\mathbf{\Lambda}_{t|t} &= \mathbf{\Lambda}_{t|t+1} + \mathbf{H}_t^{\mathrm{T}} \mathbf{R}_t^{-1} \mathbf{H}_t && \text{(B.75)}
\end{aligned}$$

which concludes the proof. □

Appendix C: A few formula

C.1. Inversion matrix

We state here without proof the various variant of the Woodbury identity as its proof is ubiquous in for instance Petersen and Pedersen (2012) or in Wikipedia (2016)

$$(\mathbf{A} + \mathbf{C}\mathbf{B}\mathbf{C}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C}(\mathbf{B}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1} \tag{C.1}$$

If $\mathbf{P}, \mathbf{R}$ are positive definite, we have

$$(\mathbf{P}^{-1} + \mathbf{B}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^{\mathrm{T}}(\mathbf{B}\mathbf{P}\mathbf{B}^{\mathrm{T}} + \mathbf{R})^{-1} \tag{C.2}$$

C.2. Conditional Formula

We have the following easy formulae for conditional expectation and variance (referred to as the tower equalities)

$$\mathbb{E}[X \mid Z] = \mathbb{E}[\mathbb{E}[X \mid Y, Z] \mid Z] \tag{C.3}$$

$$\mathrm{Var}[X \mid Z] = \mathrm{Var}[\mathbb{E}[X \mid Y, Z] \mid Z] + \mathbb{E}[\mathrm{Var}[X \mid Y, Z] \mid Z] \tag{C.4}$$

Proof. The conditional expectation $\mathbb{E}[X \mid W = w, Y = y]$ is a function of w and y and can be denoted by $g(w, y)$. We can now calculate as follows:

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[X \mid W, Y] \mid Y = y] &= \mathbb{E}[g(W, Y) \mid Y = y] \\
&= \sum_{w,y'} g(w, y') \Pr[W = w, Y = y' \mid Y = y] \\
&= \sum_{w} g(w, y) \Pr[W = w \mid Y = y] \\
&= \sum_{w} \mathbb{E}[X \mid W = w, Y = y] \Pr[W = w \mid Y = y] \\
&= \sum_{w} \left[\sum_{x} x \Pr[X = x \mid W = w, Y = y] \right] \\
&\qquad \Pr[W = w \mid Y = y] \\
&= \sum_{w,x} x \Pr[X = x, W = w \mid Y = y] \\
&= \sum_{x} x \Pr[X = x \mid Y = y] \\
&= \mathbb{E}[X \mid Y = y]
\end{aligned}$$

where in our series of equation, the third equality follows from the fact that the conditional probability given by $\Pr[W = w, Y = y' \mid Y = y]$ is $\Pr[W = w \mid Y = y]$ for $y' = y$ and 0 otherwise and the sixth equality comes from Bayes' rule.

The second equality is trivial and is a consequence of the first and the law of total variance. □

Appendix D: EM Algorithm convergence proofs

D.1. First Proof

Proof. Expectation-maximization improves the expected complete log likelihood: $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ rather than the complete log likehood: $\log p(\mathbf{X}|\boldsymbol{\theta})$. We have for any $\mathbf{Z}$ with non-zero probability $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$ that the log likehood conditional to the parameters $\boldsymbol{\theta}$ can be split into two parts:

$$\log p(\mathbf{X}|\boldsymbol{\theta}) = \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) .$$

Taking the expectation over possible values for the latent variables $\mathbf{Z}$ under the current parameter estimate $\theta^{(t)}$, multiplying both sides by $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$ and summing (or integrating) over $\mathbf{Z}$, we get:

$$\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \\
&= Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) ,
\end{aligned} \tag{D.1}$$

where $H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ denotes the negated sum it is replacing. As this last equation holds for any value of $\boldsymbol{\theta}$ including $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$, we have:

$$\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) , \tag{D.2}$$

We can now subtract D.2 to D.1 to get:

$$\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) + H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}),$$

It is easy to conclude using Gibbs' inequality that tells us that $H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \geq H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$ and get:

$$\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}).$$

which states that the marginal likelihood at each step is non-decreasing, and hence concludes the proof. □

D.2. Second Proof

Proof. Another proof relies on the fact that the EM algorithm can be viewed as a coordinate ascent method, which is proved to converges monotonically to a local minimum of the function we maximize (see for instance Hastie et al. (2009) or Neal and Hinton (1999)). This works as follows. Let us consider the function:

$$F(q, \theta) := \mathrm{E}_q[\log L(\theta; x, Z)] + H(q),$$

where q is an arbitrary probability distribution over latent data $\mathbf{Z}$ and $H(q)$ is the Entropy of the distribution q. This can also be written as:

$$F(q, \theta) = -\mathrm{Div}_{\mathrm{KL}}\big(q \big\| p_{Z|X}(\cdot|x; \theta)\big) + \log L(\theta; x),$$

where $p_{Z|X}(\cdot|x; \theta)$ is the conditional distribution of the latent data given the observed data $\mathbf{X}$ and $\mathrm{Div}_{\mathrm{KL}}$ is the Kullback–Leibler divergence.

In the EM algorithm, the expectation step is to choose q to maximize F:

$$q^{(t)} = \arg\max_q \; F(q, \theta^{(t)})$$

The maximization step is to choose the parameter θ to maximize F:

$$\theta^{(t+1)} = \arg\max_\theta \; F(q^{(t)}, \theta)$$

which concludes the proof by showing that the EM algorithm is a coordinate ascent method. □

References

Y. Akimoto, A. Auger, and N. Hansen. Continuous optimization and CMA-ES. In *Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings*, pages 313–344, 2015.

Y. Akimoto, A. Auger, and N. Hansen. CMA-ES and advanced adaptation mechanisms. In *Genetic and Evolutionary Computation Conference, GECCO 2016, Denver, CO, USA, July 20-24, 2016, Companion Material Proceedings*, pages 533–562, 2016.

B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. *IEEE Congress on Evolutionary Computation*, pages 1769–1776, 2005.

A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 noisy testbed. In *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material*, pages 2467–2472, 2009.

A. Auger and N. Hansen. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012, Companion Material Proceedings*, pages 827–848, 2012.

A. Auger, M. Schoenauer, and N. Vanhaecke. LS-CMA-ES: A second-order algorithm for covariance matrix adaptation. In *Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference, Birmingham, UK, September 18-22, 2004, Proceedings*, pages 182–191, 2004.

Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002. ISBN 0471221279.

E. Benhamou. Trend without hiccups: a kalman filter approach. *IFTA Journal*, pages 38–46, 2017.

E. Benhamou. Optimal parameter inference for bayesian graphical models. *ArXiv*, November 2018.

G. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Dover Books on Mathematics Series. Dover Publications, 2006. ISBN 9780486449814.

R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied kalman filtering: with MATLAB exercises and solutions; 3rd ed.* Wiley, New York, NY, 1997.

B. Bruder, T. Dao, J. Richard, and T. Roncalli. Trend filtering methods for momentum strategies. *SSRN paper*, 2011. URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2289097.

H. H. Bui, S. Venkatesh, and G. West. Hidden markov models. chapter Tracking and Surveillance in Wide-area Spatial Environments Using the Abstract Hidden Markov Model, pages 177–196. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002. ISBN 981-02-4564-5. URL http://dl.acm.org/citation.cfm?id=505741.505750.

O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer Publishing Company, Incorporated, 2010. ISBN 1441923195, 9781441923196.

C. Chan. *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley Finance, 2013.

T. Dao. Momentum strategies: From novel estimation techniques to financial applications. *SSRN paper*, 2011. URL http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2358988.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3):142–150, Dec. 1989. ISSN 0824-7935. . URL http://dx.doi.org/10.1111/j.1467-8640.1989.tb00324.x.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1): 1–38, 1977.

J. Durbin and S. Koopman. *Time Series Analysis by State Space Methods*. Oxford Statistical

Science Series. OUP Oxford, 2012. ISBN 9780191627194. URL https://books.google.fr/books?id=lGyshsfkLrIC.

G. Einicke, G. Falco, and M. J.T. Em algorithm state matrix estimation for navigation. *IEEE SIGNAL PROCESSING LETTERS*, 17:437–440, 05/2010 2010. URL http://porto.polito.it/2363342/.

S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Mach. Learn.*, 32(1):41–62, July 1998. ISSN 0885-6125. . URL http://dx.doi.org/10.1023/A:1007469218079.

A. Gelb. *Applied Optimal Estimation*. The MIT Press, 1974. ISBN 0262570483, 9780262570480.

Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an em approach. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 120–127. Morgan-Kaufmann, 1994. URL http://papers.nips.cc/paper/767-supervised-learning-from-incomplete-data-via-an-em-approach.pdf.

N. Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, 2006.

N. Hansen and A. Auger. CMA-ES: evolution strategies and covariance matrix adaptation. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Companion Material Proceedings, Dublin, Ireland, July 12-16, 2011*, pages 991–1010, 2011.

N. Hansen and A. Auger. Evolution strategies and CMA-ES (covariance matrix adaptation). In *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014, Companion Material Proceedings*, pages 513–534, 2014.

H. . Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, (14):174–194, 958.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*. Springer series in statistics. Springer, 2009.

C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.*, 15(1):1–28, Mar. 2007. ISSN 1063-6560.

A. Javaheri, A. Galli, and D. Lautier. Filtering in finance. *Wilmott*, (5):67–83, 2003.

M. I. Jordan. *An introduction to probabilistic graphical models*. Berkeley, 2016. URL http://people.eecs.berkeley.edu/~jordan/prelims/.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, D*, 82:35–44, 1960.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, D*, 83:95–108, 1961.

G. Kitagawa. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193, 9780262013192.

D. Lautier. Simple and extended Kalman filters : an application to term structures of commodity prices. *Applied Financial Economics*, 14(13):963–974, Sept. 2004. URL https://halshs.archives-ouvertes.fr/halshs-00152998.

K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, 2002. AAI3082340.

K. P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Aug. 2013. ISBN 0262018020. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262018020.

K. P. Murphy and M. A. Paskin. Linear time inference in hierarchical hmms. In *Proceedings of the*

14th International Conference on Neural Information Processing Systems: Natural and Synthetic, NIPS'01, pages 833–840, Cambridge, MA, USA, 2001. MIT Press. URL http://dl.acm.org/citation.cfm?id=2980539.2980647.

R. M. Neal and G. E. Hinton. Learning in graphical models. chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pages 355–368. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-60032-3. URL http://dl.acm.org/citation.cfm?id=308574.308679.

Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *J. Mach. Learn. Res.*, 18(1):564–628, Jan. 2017. ISSN 1532-4435.

M. Ostendorf, V. V. Digalakis, and O. A. Kimball. From hmm's to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, Sept 1996. ISSN 1063-6676. .

K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, nov 2012. URL http://www2.imm.dtu.dk/pubdb/p.php?3274. Version 20121115.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 257–286, 1989.

L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, Aug. 1965.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009. ISBN 0136042597, 9780136042594.

L. K. Saul and M. I. Jordan. Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37(1), Oct 1999. ISSN 1573-0565. . URL https://doi.org/10.1023/A:1007649326333.

D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, New York, NY, USA, 2006. ISBN 0471708585.

P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Comput.*, 9(2):227–269, Feb. 1997. ISSN 0899-7667. . URL http://dx.doi.org/10.1162/neco.1997.9.2.227.

C. Thornton. *Triangular Covariance Factorizations for Kalman Filtering*. NASA-CR-149 147. University of California, Los Angeles–Engineering, 1976.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

K. Varelas, A. Auger, D. Brockhoff, N. Hansen, O. A. ElHara, Y. Semet, R. Kassab, and F. Barbaresco. A comparative study of large-scale variants of CMA-ES. In *Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part I*, pages 3–15, 2018.

Wikipedia. Woodbury matrix identity, 2016. URL https://en.wikipedia.org/wiki/Woodbury_matrix_identity.

Wikipedia. Cma-es, 2018. URL https://en.wikipedia.org/wiki/CMA-ES.