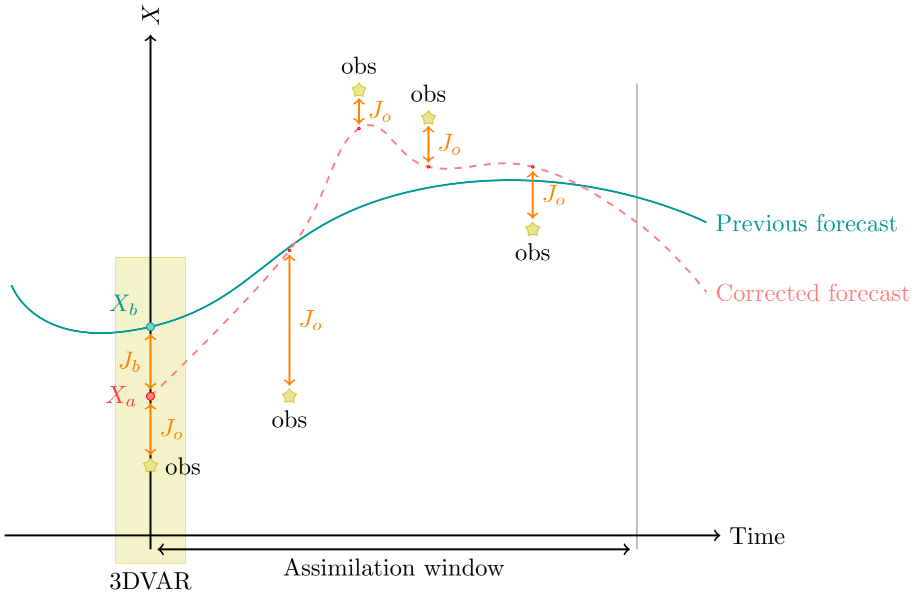
Statistical Data Assimilation: Nonlinear Filters

Mark Asch - CSU/IMU/2023



Outline of the Basic course

Adjoint methods and variational data assimilation (4h)

1. Introduction to data assimilation: setting, history, overview, definitions.
2. Optimization methods.
3. Adjoint method.
4. Variational data assimilation methods:
   (a) 3D-Var,
   (b) 4D-Var.

Statistical estimation, Kalman filters and sequential data assimilation (4h)

1. Introduction to statistical DA.

2. Statistical estimation.

3. The Kalman filter.

Outline of the Advanced course

Statistical estimation, nonlinear Kalman filters and sequential data assimilation (4h)

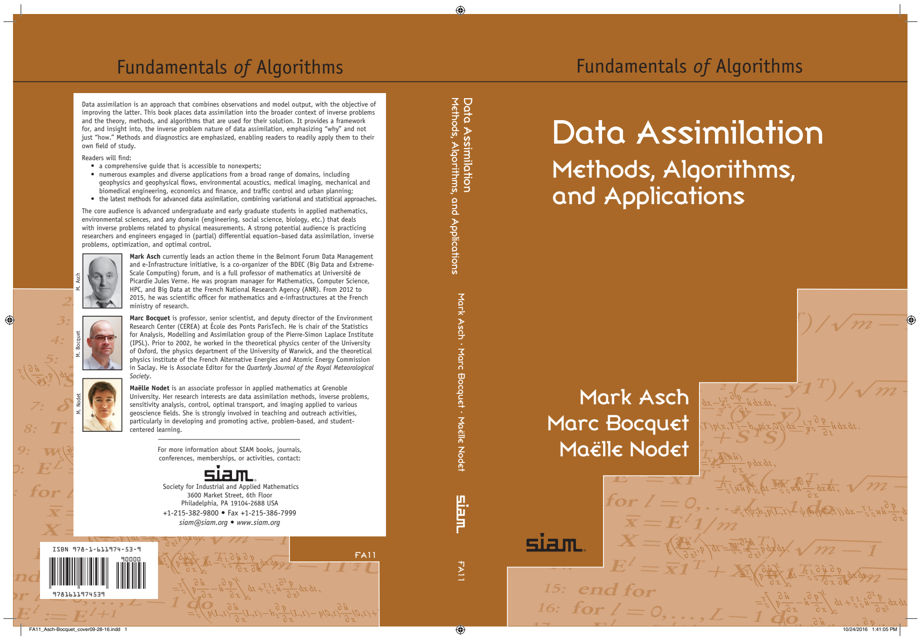
1. Introduction to statistical DA.

2. Statistical estimation.

3. The Kalman filter.

4. Nonlinear extensions and ensemble filters.

Reference Textbooks



Fundamentals *of* Algorithms

Data assimilation is an approach that combines observations and model output, with the objective of improving the latter. This book places data assimilation into the broader context of inverse problems and the theory, methods, and algorithms that are used for their solution. It provides a framework for, and insight into, the inverse problem nature of data assimilation, emphasizing “why” and not just “how.” Methods and diagnostics are emphasized, enabling readers to readily apply them to their own field of study.

Readers will find:

- a comprehensive guide that is accessible to nonexperts;
- numerous examples and diverse applications from a broad range of domains, including geophysics and geophysical flows, environmental acoustics, medical imaging, mechanical and biomedical engineering, economics and finance, and traffic control and urban planning;
- the latest methods for advanced data assimilation, combining variational and statistical approaches.

The core audience is advanced undergraduate and early graduate students in applied mathematics, environmental sciences, and any domain (engineering, social science, biology, etc.) that deals with inverse problems related to physical measurements. A strong potential audience is practicing researchers and engineers engaged in (partial) differential equation–based data assimilation, inverse problems, optimization, and optimal control.

M. Asch

Mark Asch currently leads an action theme in the Belmont Forum Data Management and e-Infrastructure initiative, is a co-organizer of the BDEC (Big Data and Extreme-Scale Computing) forum, and is a full professor of mathematics at Université de Picardie Jules Verne. He was program manager for Mathematics, Computer Science, HPC, and Big Data at the French National Research Agency (ANR). From 2012 to 2015, he was scientific officer for mathematics and e-infrastructures at the French ministry of research.

M. Bocquet

Marc Bocquet is professor, senior scientist, and deputy director of the Environment Research Center (CEREA) at École des Ponts ParisTech. He is chair of the Statistics for Analysis, Modelling and Assimilation group of the Pierre-Simon Laplace Institute (IPSL). Prior to 2002, he worked in the theoretical physics center of the University of Oxford, the physics department of the University of Warwick, and the theoretical physics institute of the French Alternative Energies and Atomic Energy Commission in Saclay. He is Associate Editor for the *Quarterly Journal of the Royal Meteorological Society*.

M. Nodet

Maëlle Nodet is an associate professor in applied mathematics at Grenoble University. Her research interests are data assimilation methods, inverse problems, sensitivity analysis, control, optimal transport, and imaging applied to various geoscience fields. She is strongly involved in teaching and outreach activities, particularly in developing and promoting active, problem-based, and student-centered learning.

For more information about SIAM books, journals,
conferences, memberships, or activities, contact:

siam
Society for Industrial and Applied Mathematics
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA
+1-215-382-9800 • Fax +1-215-386-7999
siam@siam.org • *www.siam.org*

ISBN 978-1-611974-53-9
90000
9781611974539

FA11

Data Assimilation
Methods, Algorithms, and Applications

Mark Asch · Marc Bocquet · Maëlle Nodet

siam

FA11

Fundamentals *of* Algorithms

Data Assimilation

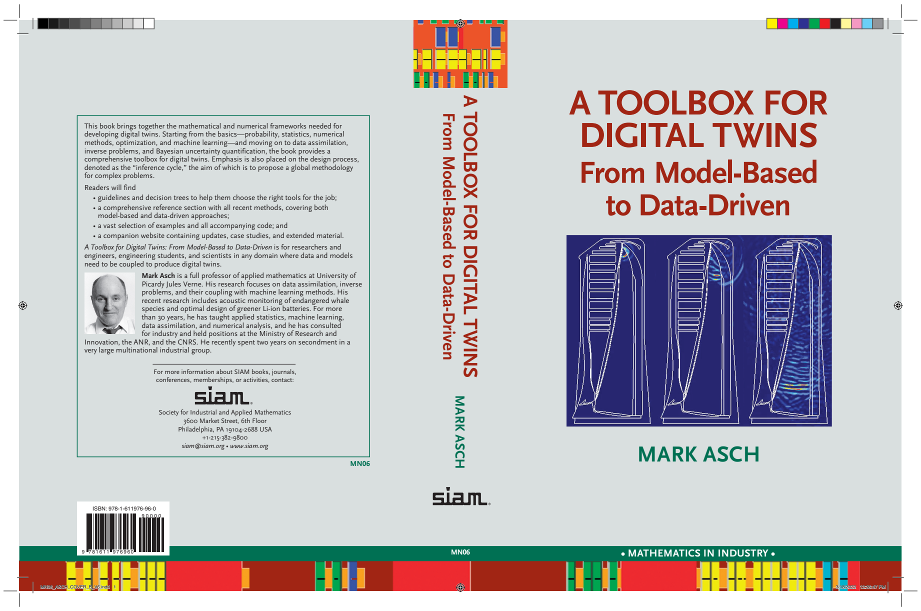
Methods, Algorithms,
and Applications

Mark Asch
Marc Bocquet
Maëlle Nodet

siam

FA11_Asch-Bocquet_cover09-28-16.indd 1

10/24/2016 1:41:05 PM



This book brings together the mathematical and numerical frameworks needed for developing digital twins. Starting from the basics—probability, statistics, numerical methods, optimization, and machine learning—and moving on to data assimilation, inverse problems, and Bayesian uncertainty quantification, the book provides a comprehensive toolbox for digital twins. Emphasis is also placed on the design process, denoted as the “inference cycle,” the aim of which is to propose a global methodology for complex problems.

Readers will find

- guidelines and decision trees to help them choose the right tools for the job;
- a comprehensive reference section with all recent methods, covering both model-based and data-driven approaches;
- a vast selection of examples and all accompanying code; and
- a companion website containing updates, case studies, and extended material.

A Toolbox for Digital Twins: From Model-Based to Data-Driven is for researchers and engineers, engineering students, and scientists in any domain where data and models need to be coupled to produce digital twins.

Mark Asch is a full professor of applied mathematics at University of Picardy Jules Verne. His research focuses on data assimilation, inverse problems, and their coupling with machine learning methods. His recent research includes acoustic monitoring of endangered whale species and optimal design of greener Li-ion batteries. For more than 30 years, he has taught applied statistics, machine learning, data assimilation, and numerical analysis, and he has consulted for industry and held positions at the Ministry of Research and Innovation, the ANR, and the CNRS. He recently spent two years on secondment in a very large multinational industrial group.

For more information about SIAM books, journals,
conferences, memberships, or activities, contact:

siam
Society for Industrial and Applied Mathematics
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA
+1-215-382-9800
siam@siam.org • *www.siam.org*

MN06

ISBN: 978-1-611976-96-0
90000
9 781611 976960

A TOOLBOX FOR DIGITAL TWINS
From Model-Based to Data-Driven

MARK ASCH

siam

MN06

A TOOLBOX FOR DIGITAL TWINS

From Model-Based
to Data-Driven

MARK ASCH

• MATHEMATICS IN INDUSTRY •

Recall: Statistical DA - introduction

- In statistical DA we generalize the variational approach to deal with errors and noise in

  ⇒ the models,
  ⇒ the observations and
  ⇒ the initial conditions.

- The variational results could of course be derived as a special case of statistical DA, in the limit where the noise disappears.

- Even the statistical results can be derived in a very general way, using SDEs and/or Bayesian analysis, and then specialized to the various Kalman-type filters that we will study here.

- Practical inverse problems and data assimilation problems involve measured data.

$\Rightarrow$ These data are inexact and are mixed with random noise.

$\Rightarrow$ Only statistical models can provide rigorous, effective means for dealing with this measurement error.

KALMAN FILTERS

Recall: Kalman Filters - stochastic model

- We seek to estimate the state $\mathbf{x} \in \mathbb{R}^n$ of a discrete-time dynamic process that is governed by the linear stochastic difference equation

$$\mathbf{x}_{k+1} = \mathbf{M}_{k+1}\mathbf{x}_k + \mathbf{w}_k \tag{1}$$

- with a measurement/observation $\mathbf{y} \in \mathbb{R}^m$,

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k. \tag{2}$$

- Note:

⇒ $\mathbf{M}_{k+1}$ and $\mathbf{H}_k$ are considered linear, here.

⇒ The random vectors, $\mathbf{w}_k$ and $\mathbf{v}_k$, represent the process/modeling and measurement/observation errors respectively.

$\Rightarrow$ They are assumed to be independent, white noise processes with Gaussian/normal probability distributions,

$$\begin{aligned}\mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k),\\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}_k),\end{aligned}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices (supposed known) of the modeling and observation errors respectively.

- All these assumptions about unbiased and uncorrelated errors (in time and between each other) are not limiting, since extensions of the standard Kalman filter can be developed should any of these not be valid—see Advanced Course.

- We note that, for a broader mathematical view on the above system, we could formulate all of statistical DA in terms of stochastic differential equations (SDEs).

$\Rightarrow$ Then the theory of Itô provides a detailed solution of the problem of optimal filtering as well as rigorous existence and uniqueness results... see [Law, Sarkka].

Kalman Filters - sequential assimilation scheme

The typical assimilation scheme is made up of two major steps:

1. a prediction/forecast step, and

2. a correction/analysis step.

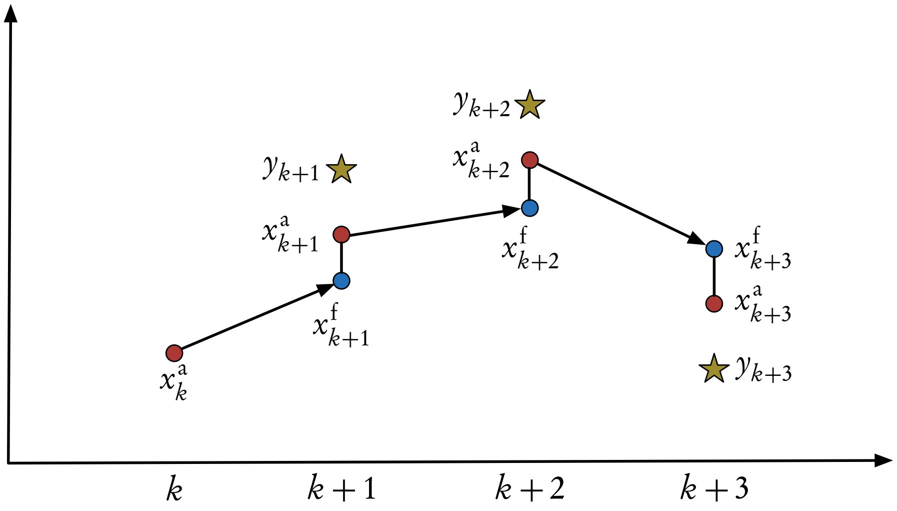


Figure 1: Sequential assimilation scheme for the Kalman filter. The x-axis denotes time, the y-axis denotes the values of the state and observations vectors.

- At time t_k we have the result of a previous forecast, $\mathbf{x}_k^{\mathrm{f}}$, (the analogue of the background state $\mathbf{x}_k^{\mathrm{b}}$) and the result of an ensemble of observations in $\mathbf{y}_k$.

- Based on these two vectors, we perform an analysis that produces $\mathbf{x}_k^{\mathrm{a}}$.

- We then use the evolution model to obtain a prediction of the state at time t_{k+1}.

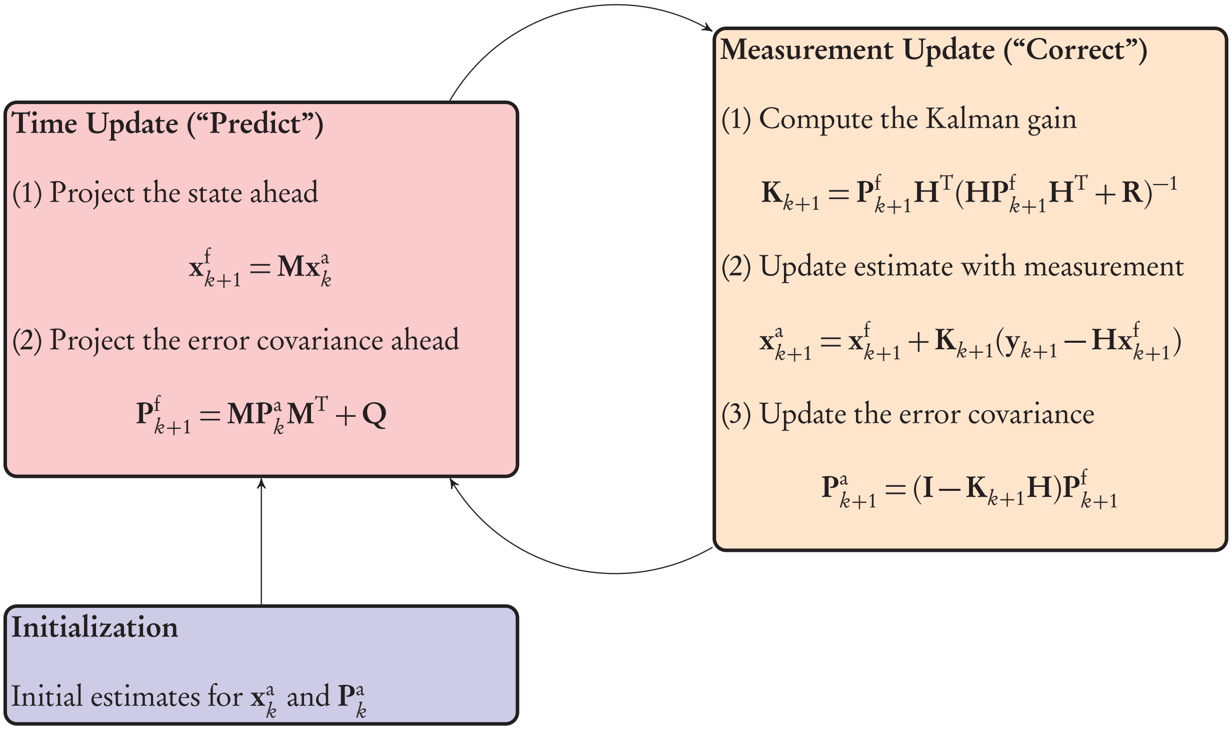
- The result of the forecast is denoted $\mathbf{x}_{k+1}^{\mathrm{f}}$, and becomes the background, or initial guess, for the next time-step—see Figure 1.

- The Kalman filter problem can be resumed as follows:

  ⇒ given a prior/background estimate $\mathbf{x}^{\mathrm{f}}$ of the system state at time t_k,

  ⇒ what is the best update/analysis $\mathbf{x}_k^{\mathrm{a}}$ based on the currently available measurements $\mathbf{y}_k$?

Kalman Filters - 2-step procedure



Time Update (“Predict”)

(1) Project the state ahead

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathrm{a}}_{k}$$

(2) Project the error covariance ahead

$$\mathbf{P}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{\mathrm{T}} + \mathbf{Q}$$

Measurement Update (“Correct”)

(1) Compute the Kalman gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

(2) Update estimate with measurement

$$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1})$$

(3) Update the error covariance

$$\mathbf{P}^{\mathrm{a}}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}^{\mathrm{f}}_{k+1}$$

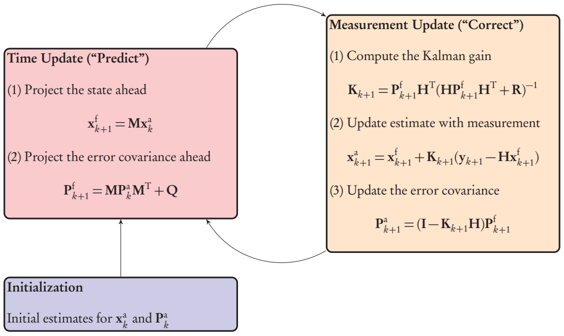
Initialization

Initial estimates for $\mathbf{x}^{\mathrm{a}}_{k}$ and $\mathbf{P}^{\mathrm{a}}_{k}$

Figure 2: Kalman filter loop, showing the two phases, predict and correct, preceded by an initialization step.

The predictor-corrector loop is illustrated in the Figure and can be transposed, as is, into an operational algorithm.

KF - predictor/forecast step



Time Update ("Predict")

(1) Project the state ahead

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathrm{a}}_{k}$$

(2) Project the error covariance ahead

$$\mathbf{P}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{\mathrm{T}} + \mathbf{Q}$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

(2) Update estimate with measurement

$$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1})$$

(3) Update the error covariance

$$\mathbf{P}^{\mathrm{a}}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}^{\mathrm{f}}_{k+1}$$

Initialization

Initial estimates for $\mathbf{x}^{\mathrm{a}}_{k}$ and $\mathbf{P}^{\mathrm{a}}_{k}$

- Start from a previous analyzed state, $\mathbf{x}^{\mathrm{a}}_{k}$, or from the initial state if $k = 0$, characterized by the Gaussian pdf $p(\mathbf{x}^{\mathrm{a}}_{k} \mid \mathbf{y}^{\mathrm{o}}_{1:k})$ of mean $\mathbf{x}^{\mathrm{a}}_{k}$ and covariance matrix $\mathbf{P}^{a}_{k}$.[1]

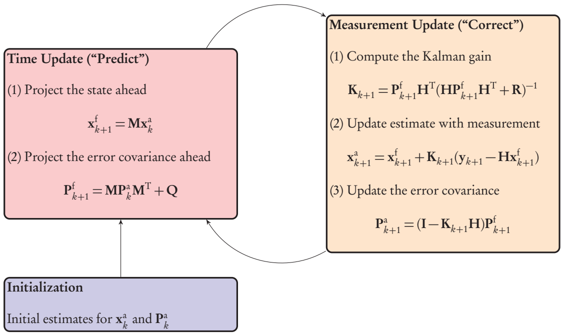
- An estimate of $\mathbf{x}^{\mathrm{t}}_{k+1}$ is given by the dynamical model which defines the forecast as

$$\begin{aligned} \mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{a}}_{k}, & (3) \\ \mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{\mathrm{T}}_{k+1} + \mathbf{Q}_{k+1}, & (4) \end{aligned}$$

[1] We use here the classical notation $\mathbf{y}_{i:j} = (\mathbf{y}_i, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_j)$ for $i \leq j$ that denotes conditioning on all the observations in the interval.

where the expression for f_{k+1} is obtained from the dynamics equation and the definition of the model noise covariance, $\mathbf{Q}$.

KF - corrector/analysis step



Time Update (“Predict”)

(1) Project the state ahead

$$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathrm{a}}_{k}$$

(2) Project the error covariance ahead

$$\mathbf{P}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{\mathrm{T}} + \mathbf{Q}$$

Measurement Update (“Correct”)

(1) Compute the Kalman gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

(2) Update estimate with measurement

$$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1})$$

(3) Update the error covariance

$$\mathbf{P}^{\mathrm{a}}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}^{\mathrm{f}}_{k+1}$$

Initialization

Initial estimates for $\mathbf{x}^{\mathrm{a}}_{k}$ and $\mathbf{P}^{\mathrm{a}}_{k}$

- At time t_{k+1}, the pdf $p(\mathbf{x}^{\mathrm{f}}_{k+1} \mid \mathbf{y}^{\mathrm{o}}_{1:k})$ is known, thanks to the mean $\mathbf{x}^{\mathrm{f}}_{k+1}$ and covariance matrix $\mathbf{P}^{\mathrm{f}}_{k+1}$ just calculated, as well as the assumption of a Gaussian distribution.

- The analysis step then consists of correcting this pdf using the observation available at time t_{k+1} in order to compute $p(\mathbf{x}^{\mathrm{a}}_{k+1} \mid \mathbf{y}^{\mathrm{o}}_{1:k+1})$. This comes from the BLUE in the dynamical context and gives

$$\begin{aligned}
\mathbf{K}_{k+1} &= \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\mathrm{T}}+\mathbf{R}_{k+1}\right)^{-1}, && (5)\\
\mathbf{x}^{\mathrm{a}}_{k+1} &= \mathbf{x}^{\mathrm{f}}_{k+1}+\mathbf{K}_{k+1}\left(\mathbf{y}_{k+1}-\mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1}\right), && (6)\\
\mathbf{P}^{\mathrm{a}}_{k+1} &= \left(\mathbf{I}-\mathbf{K}_{k+1}\mathbf{H}\right)\mathbf{P}^{\mathrm{f}}_{k+1}. && (7)
\end{aligned}$$

ENSEMBLE KALMAN FILTERS

Ensemble Kalman Filter - EnKF

- The ensemble Kalman filter (EnKF) is an elegant approach that avoids
  - ⇒ the steps of linearization in the classical Kalman Filter,
  - ⇒ and the need for adjoints in the variational approach.

- It is still based on a Kalman filter, but an ensemble of realizations is used to compute an estimate of the population mean and variance, thus avoiding the need to compute inverses of potentially large matrices to obtain the posterior covariance, as was the case above in equations (5) and (7).

- The EnKF and its variants have been successfully developed and implemented in meteorology and oceanography, including in operational weather forecasting systems. Because the method is simple to implement, it has been widely used in these fields.

- But it has spread out to other geoscience disciplines and beyond. For instance, to name a few domains, it has been applied in greenhouse gas inverse modeling, air quality forecasting, extra-terrestrial atmosphere forecasting , detection and attribution in climate sciences, geomagnetism re-analysis , and ice-sheet parameter estimation and forecasting. It has also been used in petroleum reservoir estimation, in adaptive optics for extra large telescopes, and highway traffic estimation.

- More recently, the idea was proposed to exploit the EnKF as a universal approach for all inverse problems. The term EKI, Ensemble Kalman Inversion [Stuart], is used to describe this approach.

Principle of the EnKF

- The EnKF was originally proposed by G. Evensen in 1994 and amended in [Evenson2009].

Definition 1. The ensemble Kalman filter (EnKF) is a Kalman filter that uses an ensemble of realizations to compute estimates of the population mean and covariance.

- Since it is based on Gaussian statistics (mean and covariance) it does not solve the Bayesian filtering problem in the limit of a large number of particles, as opposed to the more general *particle filter*—see: Advanced Course. Nonetheless, it turns out to be an excellent approximate algorithm for the filtering problem.

- As in the particle filter, the EnKF is based on the concept of particles, a collection of state vectors, which are called the members of the ensemble.

⇒ Rather than propagating huge covariance matrices, the errors are emulated by scattered particles, a collection of state vectors whose variability is meant to be representative of the uncertainty of the system's state resulting from the forecaster's ignorance.

⇒ Just like the particle filter, the members are propagated by the nonlinear model, without any linearization. Not only does this avoid the derivation of the tangent linear model, but it also circumvents the approximate linearization.

⇒ Finally, as opposed to the particle filter, the EnKF does not irremediably suffer from the curse of dimensionality.

- To sum up, here are the important remarks:

⇒ the EnKF avoids the linearization step of the KF;

⇒ the EnKF avoids the inversion of potentially large matrices;

⇒ the EnKF does not require any adjoint, as in variational assimilation;

$\Rightarrow$ the EnKF has been applied to a vast number of real-world problems.

EnKF - the Three Steps

1. **Initialization:** generate an ensemble of m random states $\left\{\mathbf{x}_{i,0}^{\mathrm{f}}\right\}_{i=1,\ldots,m}$ at time $t=0$.

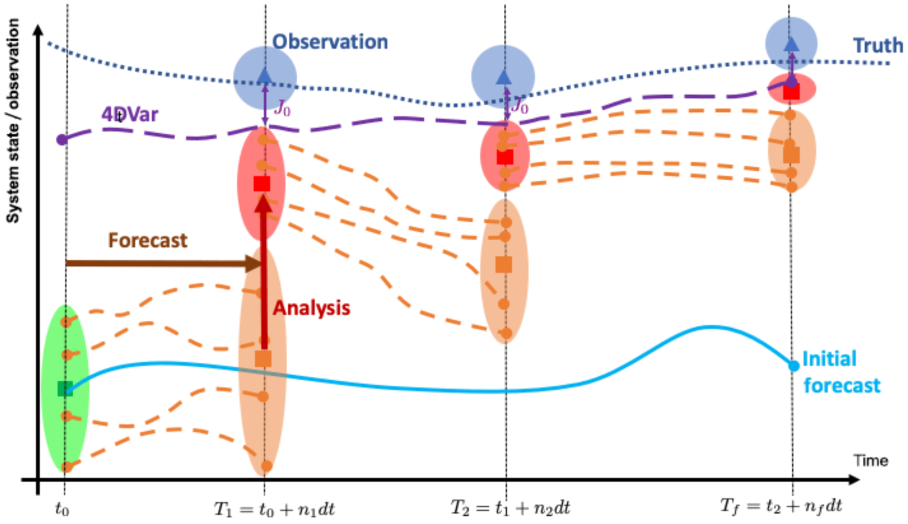
2. **Forecast:** compute the prediction for each member of the ensemble.

3. **Analysis:** correct the prediction in light of the observations.

- **Notes**:
  1. Propagation can equivalently be performed either at the end of the analysis step or at the beginning of the forecast step.
  2. The Kalman gain is not computed directly, but estimated from the ensemble statistics.
  3. With the important exception of the Kalman gain computation, all operations on the ensemble members are independent. As a result, parallelization is straightforward.

4. This is one of the main reasons for the success/popularity of the EnKF.
5. Full details can be found in the Basic Course Lectures, and in the references.

Comparison: EnKf and 4D-Var



- Principle of data assimilation: Having a physical model able to forecast the evolution of a system from time $t = t_0$ to time $t = T_f$ (cyan curve), the aim of DA is to use available observations (blue triangles) to correct the model projections and get closer to the (unknown) truth (dotted line).

- In EnKFs, the initial system state and its uncertainty (green square and ellipsoid) are represented by N_e members.

$\Rightarrow$ The members are propagated forward in time during n_1 model time steps $\mathrm{d}t$ to $t = T_1$ where observations are available (forecast phase, orange dashed lines).

$\Rightarrow$ At $t = T_1$ the analysis uses the observations and their uncertainty (blue triangle and ellipsoid) to produce a new system state that is closer to the observations and with a lower uncertainty (red square and ellipsoid).

$\Rightarrow$ A new forecast is issued from the analysed state and this procedure is repeated until the end of the assimilation window at $t = T_f$.

$\Rightarrow$ The model state should get closer to the truth and with lower uncertainty as more observations are assimilated.

- Time-dependent variational methods (4D-Var) iterate over the assimilation window to find the trajectory that minimises the misfit (J_0) between the model and all observations available from t_0 to T_f (violet curve).

- For linear dynamics, Gaussian errors and infinite ensemble sizes, the states produced at the end of the assimilation window by the two methods should be equivalent.

Hybrid - EnsVar - DA

- The term hybrid DA refers to a system where two DA methods run concurrently, exchanging information about errors and estimated model states, to obtain improved estimations of these.

- For challenging DA problems, neither pure EnKF nor pure 4D-Var can do the job.

- These methods have cross-fertilized to combine the benefits of variational and ensemble Kalman methods, and at the same time, overcome the limitations of the individual methods.

- All the details of the different hybridizations are beyond the scope of this lecture, and [Asch2016] should be consulted for full explanations of the various options as well as references.

EnsVar - Motivation

- Here, we look into the many ways to combine the benefits of variational methods and of the ensemble Kalman approaches.

- The focus has mainly been in the domain of numerical weather prediction (NWP).

- Combining the advantages of these methods, one also wishes to avoid some of the drawbacks of both classes of methods.

  ⇒ From a theoretical standpoint, the EnKF propagates the errors and has a dynamical, flow-dependent, representation of those errors. It also does not require the tangent linear and adjoint model of the observation operator, as seen in the Basic Course.

  ⇒ On the other hand, 4D-Var by definition operates on a time data assimilation window over which asynchronous observations can consistently, i.e. model-wise, be assimilated. Moreover, 4D-Var can perform a full nonlinear analysis within its data assimilation window thanks to numerical optimization techniques.

- On the downsides,

⇒ the EnKF requires the use of regularization techniques, inflation and localization specifically, to filter out sampling errors and address the rank-deficiency issue of the ensemble (collapse...).

⇒ The 4D-Var requires the use of the tangent linear and adjoint models (evolution and observation) which are very time-consuming to derive and maintain.

Hybrid Methods

- Hybrid EnKF with 3D-Var (EnsVAR)
- Ensemble of variational DAs (EDA)
- Hybrid EnKF with 4D-Var, known as 4DEnVar
- Iterative ensemble Kalman smoother (IEnKS), derived from Bayes' Law.

Hybrid EnKF with 3D-Var

- The 3D-Var system relies on a full rank static (and/or predetermined) error covariance matrix $\mathbf{C}$.

- The EnKF estimates the flow-dependent errors through the perturbations $f = \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}} = \mathbf{X}_{\mathrm{f}}\mathbf{X}_{\mathrm{f}}^{\mathrm{T}}$, where $\mathbf{X}_{\mathrm{f}}$ is the normalized perturbation matrix (see Ensemble Kalman Filter above).

- Whatever the type of EnKF, the simplest idea is to perform a state analysis using a linear combination of these error covariances,

$$\mathbf{B} = \gamma\mathbf{C} + (1-\gamma)f,$$

where $\gamma \in [0, 1]$ is a scalar parameter that controls the blending of the covariances:

- ⇒ using $\mathbf{B}$ with $\gamma = 1$ corresponds to a 3D-Var state analysis while
- ⇒ using $\mathbf{B}$ with $\gamma = 0$ corresponds to the pure EnKF state analysis.

- If the EnKF is stochastic, then one can update each member $i = 1, \ldots, m$ of the ensemble using a state analysis with $\mathbf{B}$, which essentially solves the following variational problem,

$$\mathcal{L}_i(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} + \boldsymbol{\epsilon}_i - \mathcal{H}(\mathbf{x})\|^2_{\mathbf{R}} + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_i\|^2_{\mathbf{B}} \,,$$

  where $\mathbf{x}_i$ is the first-guess of member i.

- We recall that $\|\mathbf{z}\|^2_{\mathbf{B}} = \mathbf{z}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{z}.$

- The elegance of the scheme lies in the fact that it yields a statistically consistent update of the ensemble thanks to the stochastic representation of errors.

- This EnKF-3D-Var scheme was shown to improve the performance of data assimilation

⇒ over the EnKF when the ensemble is small, and

⇒ over 3D-Var when the observation network is not dense enough. Of course, this requires a proper tuning of γ.

- If the EnKF is deterministic, the construction of the hybrid is not as straightforward, especially concerning the update of the perturbation ensemble—please consult [Asch2016] and references therein.

Ensemble of variational DAs

- Idea: Process an *ensemble of data assimilations* or *EDA*.

  ⇒ More generally an EDA system denotes a data assimilation system that processes several variational analyses in parallel.

  ⇒ The goal is to introduce some flow-dependence in the 4D-Var operational schemes that were initially only based on the static background error covariance matrix.

  ⇒ This scheme actually mimics closely the stochastic EnKF and, even more to the point, the hybrid EnKF-3D-Var scheme described above.

- The main idea is to maintain an ensemble of Var, which will be assumed to be a 4D-Var in the following.

  ⇒ This is usually numerically costly for 4D-Var's and may require the degrading of the model resolution.

- ⇒ Each analysis uses the same background error covariance matrix $\mathbf{B}$, which may have been obtained from the sampled covariances whose variances have been properly filtered and whose correlations have been regularized, as well as the static background covariances.
- ⇒ Just as in the stochastic EnKF, it is necessary for each analysis to have perturbed observations so as to maintain statistical consistency.
- ⇒ The strong-constraint 4D-Var cost function for each analysis $i = 1, \ldots, m$ has the form

$$\mathcal{L}_i(\mathbf{x}_0) = \frac{1}{2}\sum_{k=1}^{K} \left\| \mathbf{y}_k + \boldsymbol{\epsilon}_k^i - \mathcal{H}_k \circ \mathcal{M}_{k:0}(\mathbf{x}_0) \right\|_{\mathbf{R}_k}^2 + \frac{1}{2}\left\| \mathbf{x}_0 - \mathbf{x}_0^i \right\|_{\mathbf{B}}^2,$$

where $\mathcal{M}_{k:0}$ is the resolvent of the forecast model from t_0 to t_k, $\mathcal{H}_k$ is the observation operator at t_k, $\boldsymbol{\epsilon}_k^i$ is the random noise added to observation $\mathbf{y}_k$ and is related to the i-th member analysis. The symbol $\circ$ stands for the composition operator.

⇒ This generates an ensemble of updates for $\mathbf{x}_0^i$ similarly to the stochastic EnKF.

⇒ It is also possible to perturb each member of the ensemble in the forecast step so as to account for (parametric or not) model error.

⇒ Hence, the ensemble will be instrumental in accounting for flow-dependence and model error.

Hybrid EnKF with 4D-Var (4DEnVar)

- The future of 4DVar as an operational tool is uncertain because of its

  ⇒ poor scalability and of the

  ⇒ cost of the adjoint model maintenance.

- The 4DEnVar method has emerged as a way to circumvent the development of the adjoint of the dynamical model.

  ⇒ The key idea is based on the ability of the EnKF to estimate the sensitivities of the observation to the state variables using the full observation model in place of the tangent linear one and of its adjoint in the computation of the Kalman gain.

- Many variants of the 4DEnVar are possible depending on the way the perturbations are generated, or if the adjoint model is available or not.

- Full 4DEnVar operational systems are now implemented or are in the course of being so.

- For implementation details, please consult [Asch2016].

Iterative ensemble Kalman smoother (IEnKS)

- Most of these EnVar methods, with the noticeable exception of the more firmly grounded EDA ones (see above), have been designed heuristically blending theoretical insights and operational constraints. This led to many variants of the schemes, even when this is not mathematically justified. Most of these ideas stemmed from the variational DA community.

- By contrast, the iterative ensemble Kalman smoother (IEnKS, Bocquet and Sakov, 2014), is a four-dimensional EnVar method that is derived from Bayes' rule and where all approximations are understood at a theoretical level.

- It comes from ensemble-based DA and, specifically, extends the iterative ensemble Kalman filter (Sakov et al., 2012b) to a full data assimilation window (DAW) as in 4DVar. The name smoother reminds us that the method smooths trajectories like 4DVar. However, it can equally be used for smoothing and filtering.

- Basically, the IEnKS can be seen as an EnKF, for which each analysis corresponds to a nonlinear 4DVar analysis but within the reduced subspace defined by the ensemble.

- Hence, the associated cost function is of the form

$$\mathcal{J}(\mathbf{w}) = \sum_{k=L-S+1}^{L} \frac{1}{2} \left\| \mathbf{y}_k - \mathcal{F}_{k:0}\overline{\mathbf{x}}_0 + \mathbf{X}_0\mathbf{w} \right\|^2_{\mathbf{R}_k} + \frac{1}{2} \left\| \mathbf{w} \right\|^2 .$$

where

⇒ $\mathcal{F}_{k:0}$ stands for the composition of $\mathcal{H}_k$ and the resolvent $\mathcal{M}_{k:0}$.

⇒ L is the length of the DAW

⇒ S is he length of the forecast between cycles, in units of $t_{k+1} - t_k$

- Because the IEnKS catches the best of 4DVar (nonlinear analysis) and EnKF (flow-dependence of the error statistics), both these parameters could be critical.

- The minimization of $\mathcal{J}$ can be performed in the ensemble subspace using any nonlinear optimization method, such as Gauss-Newton, Levenberg-Marquardt or trust-region methods

- In chaotic systems, the IEnKS outperforms any reasonably scalable DA method in terms of accuracy.

  $\Rightarrow$ By construction, it outperforms the EnKF, the EnKS and 4DVar for smoothing but also filtering.

EnVar Techniques - table

Credit: Carrassi, Bertino, Bocquet, Evensen. 2018

Table 1: Comparison of EnVar data assimilation techniques. This table answers the following questions: (i) Is the analysis based on a linear or nonlinear scheme? (ii) Is the adjoint of the evolution model required? (iii) Is the adjoint of the observation operator required? (iv) Is the background flow-dependent? (v) Are the updated perturbations stochastic or deterministic? (vi) Are the updated perturbations fully consistent with the analysis, i.e., are they a faithful representation of the analysis uncertainty? (vii) Is localization of the ensemble analysis required? (viii) Is a static background used? To some extent, all algorithms can accommodate a static background; the answer tells whether the published algorithm has a static background. Blank answers correspond to irrelevant questions.

algorithm	analysis type	evol. model adjoint required?	obs. operator adjoint required?	background flow-dependence?	sto. or det. perturbations?	consistent perturbations?	localization required?	static background
EnKF	linear		no	yes	both	yes	yes	no[4]
3DVar	nonlinear		yes	no				yes
4DVar	nonlinear	yes	yes	no				yes
EDA with 4DVar	nonlinear	yes[1]	yes[1]	yes	sto.	yes	part.	yes[3]
4DEnVar	linear	no	no	yes	sto.	no[2]	yes	yes[3]
IEnKS	nonlinear	no	no	yes	det.	yes	yes	no[4]
MLEF	nonlinear		no	yes	det.	yes	yes	no[4]
4D-ETKF	linear	no	no	yes	det.	yes	yes	no[4]

[1] The adjoint models could be avoided considering an EDA of 4DEnVar.

[2] It depends on the implementation of 4DEnVar; the perturbation are often generated by a concomitant EnKF.

[3] With an hybridization of the covariances.

[4] But possible with an hybridization of the covariances.

NONLINEAR FILTERS

Nonlinear Filters

- General, Bayesian filters
- Extended Kalman Filter
- Unscented Kalman Filter
- Particle Filter

Bayesian filters

- Numerous variants of the Kalman filter have been developed and are used in various applications.

- All of these attempt to generalize the KF to deal with either

  ⇒ a nonlinear process and/or
  ⇒ a nonlinear measurement operator.

- Our dynamic system now takes the more general form

$$
\begin{aligned}
\mathbf{x}_{k+1} &= M_{k+1}(\mathbf{x}_k) + \mathbf{w}_k, \\
\mathbf{y}_k &= H_k(\mathbf{x}_k) + \mathbf{v}_k,
\end{aligned}
$$

  where

  ⇒ M_k now represents a nonlinear function of the state at time step k and

⇒ H_k represents the nonlinear observation operator.

- We can reformulate the general filtering problem in a Bayesian setting, in which both the linear (Kalman) and nonlinear variants are easily deduced as special cases.

Recall: Bayes' Law

- **Independence and Conditioning**:

Definition 2 (Independence)**.** Two events, A and B, are independent if

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B).$$

Definition 3 (Conditional Probability)**.** The conditional probability of A on B is the probability that A occurs provided (or knowing) that B has occurred, and is given by

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)}.$$

- It follows that if A and B are mutually independent, then

$$\mathrm{P}(A \mid B) = \mathrm{P}(A), \quad \mathrm{P}(B \mid A) = \mathrm{P}(B).$$

- We can now state **Bayes' Theorem**

  $\Rightarrow$ in discrete form, for events

  $\Rightarrow$ in continuous form, for probability distributions

Theorem 1 (Bayes' Theorem for Two Events). *Let A and B be two events in Ω, then*

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A)\mathrm{P}(A)}{\mathrm{P}(B)}$$

or

$$\mathrm{P}(B \mid A) = \frac{\mathrm{P}(A \mid B)\mathrm{P}(B)}{\mathrm{P}(A)}.$$

- A more general situation is where Ω is subdivided into a partition of events, such as blood types in a human population.

Theorem 2 (Bayes' Theorem for a Partition). *Let $A_1, A_2, \ldots, A_k$ be a partition of Ω with $\mathrm{P}(A_i) > 0$ for each i. If $\mathrm{P}(B) > 0$, then, for each $i = 1, \ldots, k$,*

$$\mathrm{P}(A_i \mid B) = \frac{\mathrm{P}(B \mid A_i)\mathrm{P}(A_i)}{\mathrm{P}(B)},$$

where the total probability

$$\mathrm{P}(B) = \sum_{j=1}^{k} \mathrm{P}(B \mid A_j)\mathrm{P}(A_j).$$

Bayesian Inference

In the most general case, where we want to perform Bayesian inference for the estimation of parameters (an inverse problem!), we simply replace the probabilities by the corresponding density functions.

- Then Bayesian inference is performed in three steps:
  1. Choose a probability density $f(\theta)$, called the prior distribution, that expresses our beliefs, or prior experimental or historical knowledge, about a parameter θ before we see any data.
  2. Choose a statistical model $f(x \mid \theta)$ that reflects our beliefs about x given θ. Notice that this is expressed as a conditional probability, called the likelihood function, and not as a joint probability function.
  3. After observing data $x_1, \ldots, x_n$, update our beliefs and calculate the posterior distribution $f(\theta \mid x_1, \ldots, x_n)$.

- Let us look more closely at the three components of Bayes' Law.

Definition 4 (Prior Distribution)**.** For a given statistical model that depends on a parameter θ, considered as random, the distribution assigned to θ before observing the other random variables of interest is called the *prior distribution*. This is just the marginal distribution of the parameter.

Definition 5. [Posterior Distribution] For a statistical inference problem, with parameter θ and random sample $X_1, \ldots, X_n$, the conditional distribution of θ given $X_1 = x_1$, $\ldots$, $x_n = X_n$ is called the *posterior distribution* of θ.

Definition 6 (Likelihood Function)**.** Suppose that $X_1, X_2, \ldots, X_n$ have a joint density function

$$f(X_1, X_2, \ldots, X_n \mid \theta).$$

Given the observations $X_1 = x_1$, $X_2 = x_2$, $\ldots$,

$X_n = x_n$, the likelihood function of θ is

$L(\theta) = L(\theta \mid x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n \mid \theta).$

If the X_i are i.i.d. with density $f(X_i \mid \theta)$, , then the joint density is a product and

$$L(\theta \mid x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i \mid \theta).$$

- We point out the following properties of the likelihood:

  ⇒ The likelihood is not a probability density function and can take values outside the interval $[0, 1]$.

  ⇒ Likelihood is an important concept in both frequentist and Bayesian statistics.

  ⇒ Likelihood is a measure of the extent to which a sample provides support for particular values of a parameter in a parametric model—this will be very important when we will deal with parameter estimation, and inverse problems in general.

- ⇒ The likelihood measures the support (evidence) provided by the data for each possible value of the parameter. This means that if we compute the likelihood function at two points, $\theta = \theta_1$, $\theta = \theta_2$, and find that $L(\theta_1 \mid x) > L(\theta_2 \mid x)$, then the sample observed is more likely to have occurred if $\theta = \theta_1$. We say that θ_1 is a more plausible value for θ than θ_2.
- ⇒ For i.i.d. random variables, the log-likelihood is usually used, since it reduces the product to a sum.

Bayes' Theorem

- We now formulate the general version of **Bayes' Theorem**.

Theorem 3. *Suppose that n random variables, $X_1, \ldots, X_n$, form a random sample from a distribution with density, or probability function in the case of a discrete distribution, $f(x \mid \theta)$. Suppose also that the unknown parameter, θ, has a prior pdf $f(\theta)$. Then the posterior pdf of θ is*

$$f(\theta \mid x) = \frac{f(x_1 \mid \theta) \cdots f(x_n \mid \theta) f(\theta)}{f_n(x)}, \qquad (8)$$

where $f_n(x)$ is the marginal joint pdf of $X_1, \ldots, X_n$.

- In this theorem,

  ⇒ the *prior*, $f(\theta)$, represents the credibility of, or belief in the values of the parameters we seek, without any consideration of the data/observations;

$\Rightarrow$ the *posterior*, $f(\theta \mid x)$, is the credibility of the parameters with the data taken into account;

$\Rightarrow$ $f(x \mid \theta)$, considered as a function of θ, is the *likelihood* function, which is the probability that the data/observation could be generated by the model with a given value of the parameter;

$\Rightarrow$ the denominator, called the *evidence*, $f_n(x)$, is the total probability of the data taken over all the possible parameter values, also called the *marginal likelihood*, or the marginal, and can be considered as a normalization factor;

$\Rightarrow$ the posterior distribution is thus proportional to the product of the likelihood and the prior distribution, or, in applied terms,

$$f(\text{parameter} \mid \text{data}) \propto f(\text{data} \mid \text{parameter})\, f(\text{parameter}).$$

- What can one do with the posterior distribution thus obtained? The answer is a lot of things, in fact a complete quantification of the incertitude of the parameter's estimation is possible. We can compute:

- ⇒ Point estimates by summarizing the center of the posterior. Typically, these are the posterior mean or the posterior mode.
- ⇒ Interval estimates for a given level α—see below.
- ⇒ Estimates of the probability of an event, such as $\mathrm{P}(a < \theta < b)$ or $P(\theta > b)$.
- ⇒ Posterior quantiles.

Bayesian filters (II)

- We begin by defining a probabilistic state-space, or nonlinear filtering model, of the form

$$\mathbf{x}_k \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}), \tag{9}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k \mid \mathbf{x}_k), \quad k = 0, 1, 2, \ldots, \tag{10}$$

where

⇒ $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time k,

⇒ $\mathbf{y}_k \in \mathbb{R}^m$ is the observation vector at time k,

⇒ the conditional probability, $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$, represents the stochastic dynamics model, and can be a probability density or a discrete probability function, or a mixture of both,

⇒ the conditional probability, $p(\mathbf{y}_k \mid \mathbf{x}_k)$, represents the measurement model and its inherent noise.

- In addition, we assume that the model is Markovian, such that

$$p(\mathbf{x}_k \mid \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}),$$

  and that the observations are conditionally independent of state and measurement histories,

$$p(\mathbf{y}_k \mid \mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k \mid \mathbf{x}_k).$$

Example of Gaussian Random Walk

- To fix ideas and notation, we begin with a very simple, scalar case, the Gaussian random walk model. This model can then easily be generalized.

- Consider the scalar system

$$x_k = x_{k-1} + w_{k-1}, \quad w_{k-1} \sim \mathcal{N}(0, Q), \tag{11}$$

$$y_k = x_k + v_k, \quad v_k \sim \mathcal{N}(0, R), \tag{12}$$

  where x_k is the (hidden) state and y_k is the (known) measurement.

- Noting that $x_k - x_{k-1} = w_{k-1}$ and that $y_k - x_k = v_k$, we can immediately rewrite this system in terms of the conditional probability densities,

$$
\begin{aligned}
p(x_k \mid x_{k-1}) = \mathcal{N}\left(x_k \mid x_{k-1}, Q\right) \\
&= \frac{1}{\sqrt{2\pi Q}} \exp\left[-\frac{1}{2Q}\left(x_k - x_{k-1}\right)^2\right]
\end{aligned}
$$

and

$$
\begin{aligned}
p(y_k \mid x_k) = \mathcal{N}\left(y_k \mid x_k, R\right) \\
&= \frac{1}{\sqrt{2\pi R}} \exp\left[-\frac{1}{2R}\left(y_k - x_k\right)^2\right].
\end{aligned}
$$

- A realization of the model is shown in Figure 3.

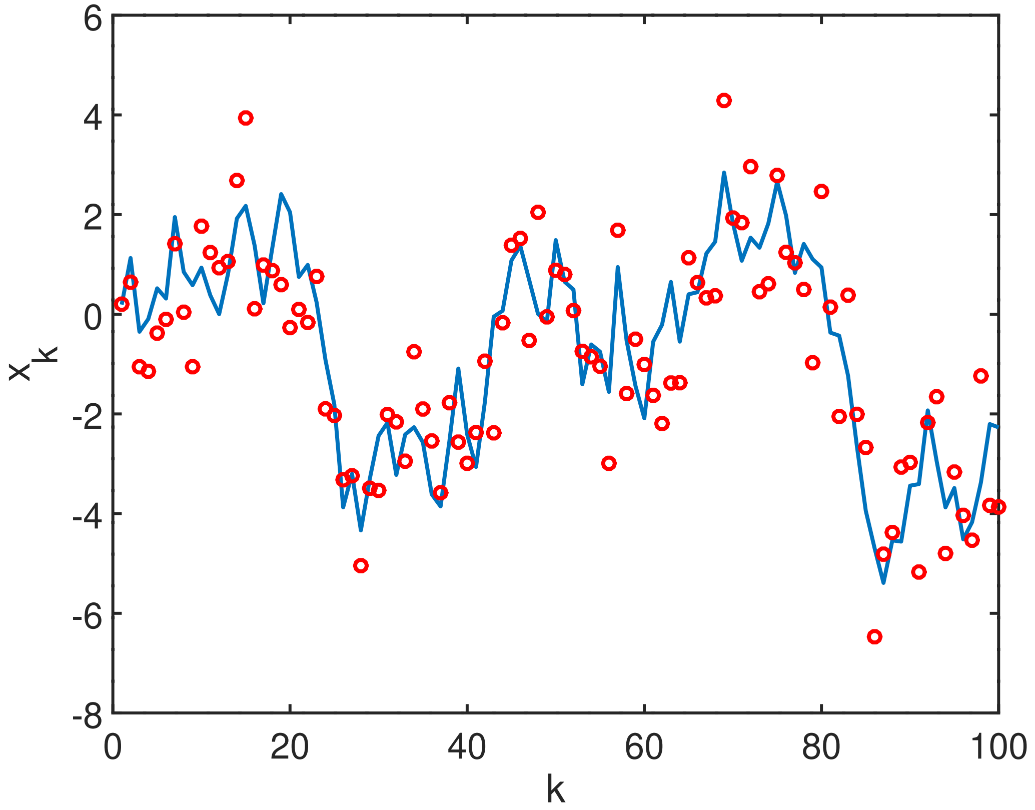


Figure 3: Gaussian random walk state space model (11)-(12). State, x_k, is solid blue curve, measurements, y_k, are red circles. Fixed values of noise variance are $Q = 1$ and $R = 1$.

Code

```
% Simulate a Guassian random walk.
% initialize
randn(’state’,123)
R=1; Q=1; K=100;
% simulate
X_init = sqrt(Q)*randn(K,1);
X = cumsum(X_init);
W = sqrt(R)*randn(K,1);
Y = X + W;
% plot
plot(1:K,X,1:K,Y(1:K,1),’ro’)
xlabel(’k’), ylabel(’x_k’)
```

Nonlinear Filter Model

- Using the nonlinear filtering model (9)-(10) and the Markov property, we can express the joint prior of the states, $\mathbf{x}_{0:T} = \{\mathbf{x}_0, \ldots, \mathbf{x}_T\}$, and the joint likelihood of the measurements, $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, as the products

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_0) \prod_{k=1}^{T} p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

  and

$$p(\mathbf{y}_{1:T} \mid \mathbf{x}_{0:T}) = \prod_{k=1}^{T} p(\mathbf{y}_k \mid \mathbf{x}_k)$$

  respectively.

- Then, applying Bayes' law, we can compute the complete posterior distribution of the states as

$$p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}) = \frac{p(\mathbf{y}_{1:T} \mid \mathbf{x}_{0:T}) p(\mathbf{x}_{0:T})}{p(\mathbf{y}_{1:T})}. \tag{13}$$

- But this type of complete characterization is not feasible to compute in real-time, or near real-time, since the number of computations per time-step increases as measurements arrive.

- What we need is a fixed number of computations per time-step.

  ⇒ This can be achieved by a recursive estimation that, step by step, produces the filtering distribution defined above.

  ⇒ In this light, we can now define the general Bayesian filtering problem, of which Kalman filters will be a special case.

Definition 7 (Bayesian Filtering)**.** Bayesian filtering is the recursive computation of the marginal posterior distribution,

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$$

known as the filtering distribution, of the state $\mathbf{x}_k$ at each time step k, given the measurements up to time k.

- Now, based on Bayes' rule, we can formulate the Bayesian filtering theorem [Sarkka2013] .

Theorem 4 (Bayesian Filter)**.** *The recursive equations, known as the Bayesian filter, for computing the filtering distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ and the predicted distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ at the time step k, are given by the three-stage process:*

***Initialization**: Define the prior distribution $p(\mathbf{x}_0)$.*

***Prediction:** Compute the predictive distribution*

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \, \mathrm{d}\mathbf{x}_{k-1}.$$

***Correction**: Compute the posterior distribution by Bayes' rule,*

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})} \, \mathrm{d}\mathbf{x}_k.$$

Kalman Filter - a special case

Now, if we assume that the dynamic and measurement models are linear, with i.i.d. Gaussian noise, then we obtain the closed-form solution for the Kalman filter, already derived in the Basic Course. We recall the linear, Gaussian state-space model,

$$\mathbf{x}_k = \mathbf{M}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{14}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{15}$$

for the Kalman filter, where

- $\mathbf{x}_k \in \mathbb{R}^n$ is the state,
- $\mathbf{y}_k \in \mathbb{R}^m$ is the measurement,
- $\mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$ is the process noise,
- $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ is the measurement noise,

- $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ is the Gaussian distributed initial state, with mean $\mathbf{m}_0$ and covariance $\mathbf{P}_0$,
- $\mathbf{M}_{k-1}$ is the time-dependent transition matrix of the dynamic model at time $k-1$, and
- $\mathbf{H}_k$ is the time-dependent measurement model matrix.

This model can be very elegantly rewritten in terms of conditional probabilities as

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) &= \mathcal{N}\left(\mathbf{x}_k \mid \mathbf{M}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}\right), \\ p(\mathbf{y}_k \mid \mathbf{x}_k) &= \mathcal{N}\left(\mathbf{y}_k \mid \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k\right). \end{aligned}$$

Theorem 5 (Kalman Filter). *The Bayesian filtering equations for the linear, Gaussian model (14)-(15) can be explicitly computed and the resulting conditional probability distributions are Gaussian. The prediction distribution is*

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \mathcal{N}\left(\mathbf{x}_k \mid \hat{\mathbf{m}}_k, \hat{\mathbf{P}}_k\right),$$

the filtering distribution is

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \mathcal{N}\left(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k\right)$$

and the smoothing distribution is

$$p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) = \mathcal{N}\left(\mathbf{y}_k \mid \mathbf{H}_k\hat{\mathbf{m}}_k, \mathbf{S}_k\right).$$

The parameters of these distributions can be computed by the three-stage Kalman filter loop:

***Initialization**: Define the prior mean $\mathbf{m}_0$ and prior covariance $\mathbf{P}_0$.*

***Prediction**: Compute the predictive distribution mean and covariance,*

$$\begin{aligned}
\hat{\mathbf{m}}_k &= \mathbf{M}_{k-1}\mathbf{m}_{k-1}, \\
\hat{\mathbf{P}}_k &= \mathbf{M}_{k-1}\mathbf{P}_{k-1}\mathbf{M}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}.
\end{aligned}$$

***Correction**: Compute the filtering distribution* *mean and covariance by first defining*

$$
\begin{aligned}
\mathbf{d}_k &= \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{m}}_k, \quad \textit{the innovation},\\
\mathbf{S}_k &= \mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k, \quad \textit{the measurement covariance},\\
\mathbf{K}_k &= \hat{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}, \quad \textit{the Kalman gain,}
\end{aligned}
$$

then finally updating the filter mean and covariance,

$$
\begin{aligned}
\mathbf{m}_k &= \hat{\mathbf{m}}_k + \mathbf{K}_k \mathbf{d}_k,\\
\mathbf{P}_k &= \hat{\mathbf{P}}_k - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{T}}.
\end{aligned}
$$

Proof. The proof—see [Sarkka2013]—is a direct application of classical results for the joint, marginal, and conditional distributions of two Gaussian random variables, $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$. □

KF for Gaussian Random Walk

We now return to the Gaussian random walk model seen above in the Example, and formulate a Kalman filter for estimating its state from noisy measurements.

Example (Kalman Filter for Gaussian Random Walk)**.** Suppose that we have measurements of the scalar y_k from the Gaussian random walk model

$$x_k = x_{k-1} + w_{k-1}, \quad w_{k-1} \sim \mathcal{N}(0, Q), \tag{16}$$

$$y_k = x_k + v_k, \quad v_k \sim \mathcal{N}(0, R). \tag{17}$$

This very basic system is found in many applications where

- x_k represents a slowly varying quantity that we measure directly.
- process noise, w_k, takes into account fluctuations in the state x_k.

- measurement noise, v_k, accounts for measurement instrument errors.

We want to estimate the state x_k over time, taking into account the measurements y_k. That is, we would like to compute the filtering density,

$$p(x_k \mid y_{1:k}) = \mathcal{N}\left(x_k \mid m_k, P_k\right).$$

We proceed by simply writing down the three stages of the Kalman filter, noting that $M_k = 1$ and $H_k = 1$ for this model. We obtain:

Initialization: Define the prior mean m_0 and prior covariance P_0.

Prediction:

$$\begin{aligned}\hat{m}_k &= m_{k-1},\\ \hat{P}_k &= P_{k-1} + Q.\end{aligned}$$

Correction: Define

$$
\begin{aligned}
d_k &= y_k - \hat{m}_k, \quad \text{the innovation},\\
S_k &= \hat{P}_k + R, \quad \text{the measurement covariance},\\
K_k &= \hat{P}_k S_k^{-1}, \quad \text{the Kalman gain},
\end{aligned}
$$

then update,

$$
\begin{aligned}
m_k &= \hat{m}_k + K_k d_k,\\
P_k &= \hat{P}_k - \frac{\hat{P}_k^2}{S_k}.
\end{aligned}
$$

In Figure 4 we show simulations with system noise standard deviation of 1 and measurement noise standard deviation of 0.5. We observe that the KF tracks the random walk very efficiently.

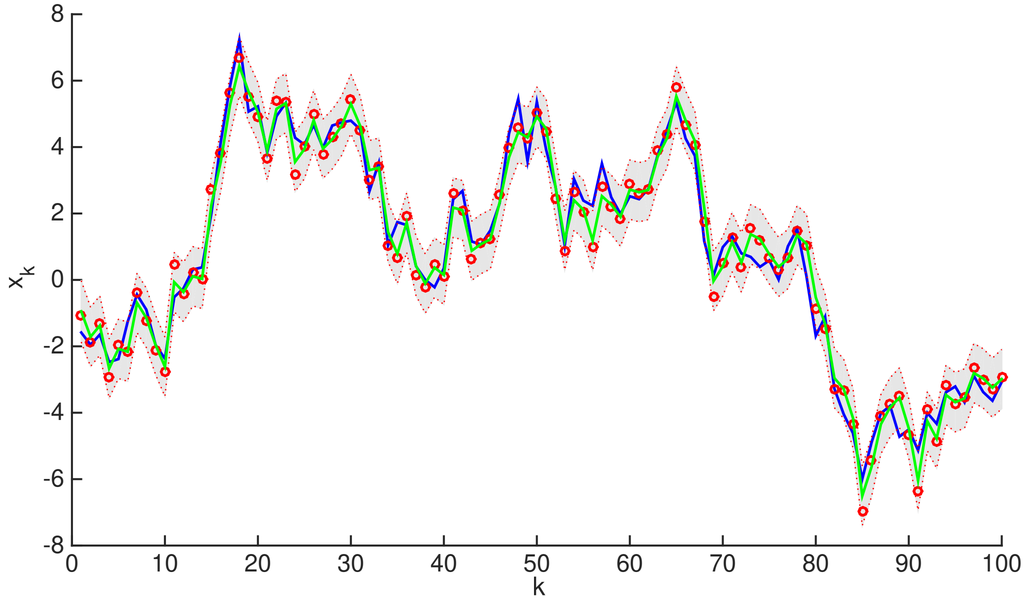


Figure 4: Kalman filter for tracking a Gaussian random walk state space model (16)-(17). State, x_k, is solid blue curve; measurements, y_k, are red circles; Kalman filter estimate is green curve and 95% quantiles are shown. Fixed values of noise variances are $Q = 1$ and $R = 0.5^2$. Results computed by `kf_gauss_state.m`.

OCTAVE code

```
% Kalman Filter for scalar Gaussian random walk
% Set parameters
sig_w = 1; sig_v = 0.5;
M = 1;
Q = sig_w^2;
H = 1;
R = sig_v^2;
% Initialize
m0 = 0;
P0 = 1;
% Simulate data
randn('state',1234);
steps = 100; T = [1:steps];
X = zeros(1,steps);
Y = zeros(1,steps);
x = m0;
for k=1:steps
  w = Q'*randn(1);
  x = M*x + w;
  y = H*x + sig_v*randn(1);
  X(k) = x;
  Y(k) = y;
end
plot(T,X,'-',T,Y,'.');
legend('Signal','Measurements');
xlabel('{k}'); ylabel('{x}_k');
% Kalman filter
m = m0;
P = P0;
for k=1:steps
  m = M*m;
  P = M*P*M' + Q;
  d = Y(:,k) - H*m;
  S = H*P*H' + R;
  K = P*H'/S;
  m = m + K*d;
  P = P - K*S*K';
  kf_m(k) = m;
  kf_P(k) = P;
end
% Plot
```

```
clf; hold on
fill([T fliplr(T)],[kf_m+1.96*sqrt(kf_P) ...
  fliplr(kf_m-1.96*sqrt(kf_P))],1, ...
  'FaceColor',[.9 .9 .9],'EdgeColor',[.9 .9 .9])
plot(T,X,'-b',T,Y,'or',T, kf_m(1,:),'-g')
plot(T,kf_m+1.96*sqrt(kf_P),':r',T,kf_m-1.96*sqrt
hold off
```

- Notes
  - ⇒ Line 3: by modifying these noise amplitudes, one can better understand how the KF operates.
  - ⇒ Lines 31-32 and 34-38: the complete filter is coded in only 7 lines, exactly as prescribed by Theorem 5. This is the reason for the excellent performance of the KF, in particular in real-time systems. In higher dimensions, when the matrices become large, more attention must be paid to the numerical linear algebra routines used. The inversion of the measurement covariance matrix, S, in line 36, is particularly challenging and requires highly tuned decomposition methods.

EXTENDED KALMAN FILTERS

Extended Kalman filters

- In real applications, we are usually confronted with nonlinearity in the model and in the measurements.

  ⇒ Moreover, the noise is not necessarily additive.

- To deal with these nonlinearities, one possible approach is to linearize about the current mean and covariance, which produces the *extended Kalman filter* (EKF).

- This filter is widely accepted as the *standard* for navigation and GPS systems, among others.

Recall the nonlinear problem,

$$\begin{aligned} \mathbf{x}_k &= \mathcal{M}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \qquad (18) \\ \mathbf{y}_k &= \mathcal{H}_k(\mathbf{x}_k) + \mathbf{v}_k, \qquad (19) \end{aligned}$$

where

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^m$, $\mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$,

- and now $\mathcal{M}_{k-1}$ and $\mathcal{H}_k$ are nonlinear functions of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ respectively.

The EKF is then based on Gaussian approximations of the filtering densities,

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathcal{N}\left(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k\right),$$

where these approximations are derived from the first-order truncation of the corresponding Taylor series in terms of the statistical moments of the underlying random variables.

Linearization in the Taylor series expansions will require evaluation of the Jacobian matrices, defined as

$$\mathbf{M}_\mathbf{x} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{m}}$$

and

$$\mathbf{H}_{\mathbf{x}} = \left[\frac{\partial \mathcal{H}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{m}}.$$

Theorem 6. *The first-order extended Kalman filter with additive noise for the nonlinear system (18)-(19) can be computed by the three-stage process:*

Initialization*: Define the prior mean* $\mathbf{m}_0$ *and prior covariance* $\mathbf{P}_0$.

Prediction: *Compute the predictive distribution mean and covariance,*

$$\begin{aligned}\hat{\mathbf{m}}_k &= \mathcal{M}_{k-1}(\mathbf{m}_{k-1}),\\ \hat{\mathbf{P}}_k &= \mathbf{M}_{\mathbf{x}}(\mathbf{m}_{k-1})\mathbf{P}_{k-1}\mathbf{M}_{\mathbf{x}}^{\mathrm{T}}(\mathbf{m}_{k-1}) + \mathbf{Q}_{k-1}.\end{aligned}$$

Correction*: Compute the filtering distribution* *mean and covariance by first defining*

$$
\begin{aligned}
\mathbf{d}_k &= \mathbf{y}_k - \mathcal{H}_k(\hat{\mathbf{m}}_k), \quad \textit{the innovation,} \\
\mathbf{S}_k &= \mathbf{H}_{\mathbf{x}}(\hat{\mathbf{m}}_k)\hat{\mathbf{P}}_k\mathbf{H}_{\mathbf{x}}^{\mathrm{T}}(\hat{\mathbf{m}}_k) + \mathbf{R}_k, \ \textit{the measurement covarianc} \\
\mathbf{K}_k &= \hat{\mathbf{P}}_k\mathbf{H}_{\mathbf{x}}^{\mathrm{T}}(\hat{\mathbf{m}}_k)\mathbf{S}_k^{-1}, \quad \textit{the Kalman gain,}
\end{aligned}
$$

then finally updating the filter mean and covariance,

$$
\begin{aligned}
\mathbf{m}_k &= \hat{\mathbf{m}}_k + \mathbf{K}_k\mathbf{d}_k, \\
\mathbf{P}_k &= \hat{\mathbf{P}}_k - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathrm{T}}.
\end{aligned}
$$

Proof. The proof—see [Sarkka2013]—is once again a direct application of classical results for the joint, marginal and conditional distributions of two Gaussian random variables, $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$. In addition, use is made of the Taylor series approximations to compute the Jacobian matrices $\mathbf{M}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{x}}$ evaluated at $\mathbf{x} = \hat{\mathbf{m}}_{k-1}$ and $\mathbf{x} = \hat{\mathbf{m}}_k$ respectively. □

Extended Kalman filter - non-additivie noise

For non-additive noise, the model is now

$$\mathbf{x}_k = \mathcal{M}_{k-1}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}), \tag{20}$$

$$\mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k, \mathbf{v}_k), \tag{21}$$

where $\mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$, and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ are system and measurement Gaussian noises.

- In this case the overall three-stage scheme is the same, with necessary modifications to take into account the additional functional dependence on $\mathbf{w}$ and $\mathbf{v}$.

Initialization: Define the prior mean $\mathbf{m}_0$ and prior covariance $\mathbf{P}_0$.

Prediction: Compute the predictive distribution mean and covariance,

$$\begin{aligned}\hat{\mathbf{m}}_k &= \mathcal{M}_{k-1}(\mathbf{m}_{k-1}, \mathbf{0}),\\ \hat{\mathbf{P}}_k &= \mathbf{M}_{\mathbf{x}}(\mathbf{m}_{k-1})\mathbf{P}_{k-1}\mathbf{M}_{\mathbf{x}}^{\mathrm{T}}(\mathbf{m}_{k-1})\\ &\quad + \mathbf{M}_{\mathbf{w}}(\mathbf{m}_{k-1})\mathbf{Q}_{k-1}\mathbf{M}_{\mathbf{w}}^{\mathrm{T}}(\mathbf{m}_{k-1}) + \mathbf{Q}_{k-1}.\end{aligned}$$

Correction: Compute the filtering distribution mean and covariance by first defining

$$\begin{aligned}\mathbf{d}_k &= \mathbf{y}_k - \mathcal{H}_k(\hat{\mathbf{m}}_k, \mathbf{0}), \text{ the innovation},\\ \mathbf{S}_k &= \mathbf{H}_{\mathbf{x}}(\hat{\mathbf{m}}_k)\hat{\mathbf{P}}_k\mathbf{H}_{\mathbf{x}}^{\mathrm{T}}(\hat{\mathbf{m}}_k)\\ &\quad + \mathbf{H}_{\mathbf{v}}(\hat{\mathbf{m}}_k)\mathbf{R}_k\mathbf{H}_{\mathbf{v}}^{\mathrm{T}}(\hat{\mathbf{m}}_k), \text{ the measurement covariance},\\ \mathbf{K}_k &= \hat{\mathbf{P}}_k\mathbf{H}_{\mathbf{x}}^{\mathrm{T}}(\hat{\mathbf{m}}_k)\mathbf{S}_k^{-1}, \text{ the Kalman gain},\end{aligned}$$

then finally updating the filter mean and covariance,

$$\begin{aligned}\mathbf{m}_k &= \hat{\mathbf{m}}_k + \mathbf{K}_k\mathbf{d}_k,\\ \mathbf{P}_k &= \hat{\mathbf{P}}_k - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathrm{T}}.\end{aligned}$$

EKF - pros and cons

- Pros:
  - ⇒ Relative simplicity, based on well-known linearization methods.
  - ⇒ Maintains the simple, elegant, and computationally efficient KF update equations.
  - ⇒ Good performance for such a simple method.
  - ⇒ Ability to treat nonlinear process and observation models.
  - ⇒ Ability to treat both additive and more general nonlinear Gaussian noise.

- Cons:
  - ⇒ Performance can suffer in presence of strong nonlinearity because of the local validity of the linear approximation (valid for small perturbations around the linear term).
  - ⇒ Cannot deal with non-Gaussian noise, such as discrete-valued random variables.

⇒ Requires differentiable process and measurement operators and evaluation of Jacobian matrices, which might be problematic in very high dimensions.

In spite of this, the EKF remains a solid filter and, as mentioned earlier, remains the basis of most GPS and navigation systems.

EKF Example - nonlinear oscillator

- Consider the nonlinear ODE model for the oscillations of a noisy pendulum with unit mass and length L,

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \frac{g}{L}\sin\theta + w(t) = 0$$

where

$\Rightarrow$ θ is the angular displacement of the pendulum,
$\Rightarrow$ g is the gravitational constant,
$\Rightarrow$ L is the pendulum's length, and
$\Rightarrow$ $w(t)$ is a white noise process.

- This is rewritten in state space form,

$$\dot{\mathbf{x}} + \mathcal{M}(\mathbf{x}) + \mathbf{w} = 0,$$

where

$$\mathbf{x} = \left[\begin{array}{c} x_1 \\ x_2 \end{array}\right] = \left[\begin{array}{c} \theta \\ \dot{\theta} \end{array}\right], \quad \mathcal{M}(\mathbf{x}) = \left[\begin{array}{c} x_2 \\ -\dfrac{g}{L}\sin x_1 \end{array}\right],$$

$$\mathbf{w} = \left[\begin{array}{c} 0 \\ w(t) \end{array}\right].$$

- Suppose that we have discrete, noisy measurements of the horizontal component of the position, $\sin(\theta)$.

⇒ Then the measurement equation is scalar,

$$y_k = \sin\theta_k + v_k,$$

where v_k is a zero-mean Gaussian random variable with variance R.

- The system is thus nonlinear in both state and measurement and the state-space system is of the general form (18)-(19).

- A simple discretization, based on the simplest Euler's method, produces

$$\begin{aligned}\mathbf{x}_k &= \mathcal{M}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}\\ y_k &= \mathcal{H}_k(\mathbf{x}_k) + v_k,\end{aligned}$$

where

$$\begin{aligned}\mathbf{x}_k &= \left[\begin{array}{c} x_1 \\ x_2 \end{array}\right]_k,\\ \mathcal{M}(\mathbf{x}_{k-1}) &= \left[\begin{array}{c} x_1 + \Delta t x_2 \\ x_2 - \Delta t \dfrac{g}{L} \sin x_1 \end{array}\right]_{k-1},\\ \mathcal{H}(\mathbf{x}_k) &= [\sin x_1]_k.\end{aligned}$$

- The noise terms have distributions

$$\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, Q), \quad v_k \sim \mathcal{N}(0, R),$$

where the process covariance matrix is

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix},$$

with components (see remark below the example),

$$q_{11} = q_c \frac{\Delta t^3}{3}, \quad q_{12} = q_{21} = q_c \frac{\Delta t^2}{2}, \quad q_{22} = q_c \Delta t,$$

and q_c is the continuous process noise spectral density.

- For the first-order EKF—higher orders are possible—we will need the Jacobian matrices of $\mathcal{M}(\mathbf{x})$ and $\mathcal{H}(\mathbf{x})$ evaluated at $\mathbf{x} = \hat{\mathbf{m}}_{k-1}$ and $\mathbf{x} = \hat{\mathbf{m}}_k$. These are easily obtained here, in an explicit form,

$$\mathbf{M}_\mathbf{x} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}} \right]_{\mathbf{x}=\mathbf{m}} = \begin{bmatrix} 1 & \Delta t \\ -\Delta t \frac{g}{L} \cos x_1 & 1 \end{bmatrix}_{k-1},$$

$$\mathbf{H}_\mathbf{x} = \left[\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \right]_{\mathbf{x}=\mathbf{m}} = \begin{bmatrix} \cos x_1 & 0 \end{bmatrix}_k .$$

- For the simulations, we take:

$\Rightarrow$ 500 time steps with $\Delta t = 0.01.$

$\Rightarrow$ Noise levels $q_c = 0.01$ and $R = 0.1.$

$\Rightarrow$ Initial angle $x_1 = 1.8$ and initial angular velocity $x_2 = 0.$

$\Rightarrow$ Initial diagonal state covariance of $0.1.$

- Results are plotted in Figure 5.

$\Rightarrow$ We notice that despite the very noisy, nonlinear measurements, the EKF rapidly approaches the true state and then tracks it extremely well.

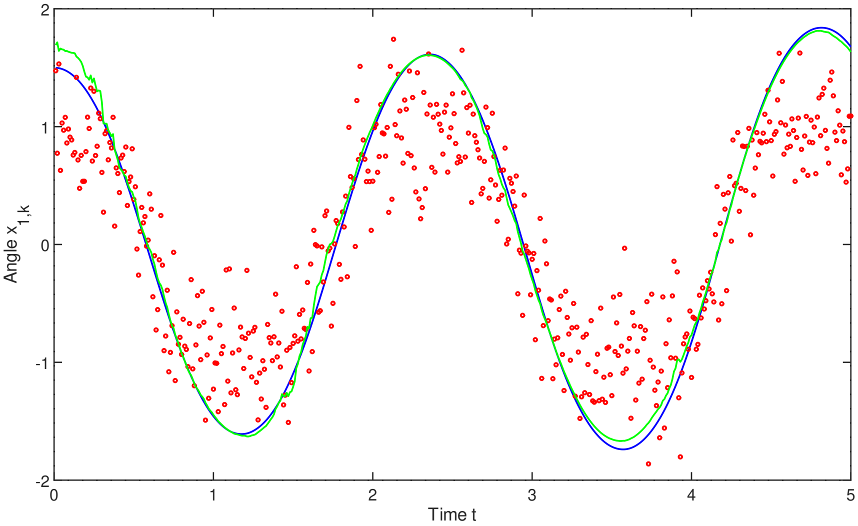


Figure 5: Extended Kalman filter for tracking a noisy pendulum model, where horizontal position is measured. State, x_k, is solid blue curve; measurements, y_k, are red circles; extended Kalman filter estimate is green curve. Results computed by `EKfPendulum.m`.

Remark. In the above example, we have used a rather special form for the process noise covariance, Q. It cannot be computed exactly for nonlinear systems and some kind of approximations are needed.[2] One way is to use an Euler-Maruyama method from SDEs,

[2] Thanks to Simo Särkkä (private communication) for suggesting this explanation.

but this leads to singular dynamics where the particle smoothers will not work. Another approach, which was used here, is to first construct an approximate model and then compute the covariance using that model. In this case the approximate model was taken as

$$\ddot{x} = w(t),$$

which is maybe overly simple, but works. Then the matrix Q is propagated through this simplified dynamics using an integration factor (exponential) solution and the corresponding power series expression of the transition matrix. Details of this can be found in [Grewal, Andrews 2008].

Unscented Kalman filters

- The *unscented Kalman filter* (UKF) was developed to overcome two shortcomings of the EKF:

  1. its difficulty to treat strong nonlinearities and
  2. its reliance on the computation of Jacobians.

- The UKF is based on the unscented transform (UT), a method for approximating the distribution of a transformed variable,

$$\mathbf{y} = g(\mathbf{x}),$$

  where $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P})$, without linearizing the function g.

- The UT is computed as follows:

  1. Choose a collection of so-called σ-points that reproduce the mean and covariance of the distribution of $\mathbf{x}$.

2. Apply the nonlinear function to the σ-points.
3. Estimate the mean and variance of the transformed random variable.

- This is a deterministic sampling approach, as opposed to Monte Carlo, particle filters, and ensemble filters that all use randomly sampled points. Note that the first two usually require orders of magnitude more points than the UKF.

UKF - theory

Suppose that the random variable $\mathbf{x} \in \mathbb{R}^n$ with mean $\mathbf{m}$ and covariance $\mathbf{P}$. Compute $N = 2n + 1$ σ-points and their corresponding weights

$$\{\mathbf{x}^{(\pm i),}, w^{(\pm i)}\}, \quad i = 0, 1, \ldots, N$$

by the formulas

$$\begin{aligned} \mathbf{x}^{(0)} &= \mathbf{m}, \\ \mathbf{x}^{(\pm i)} &= \mathbf{m} \pm \sqrt{n + \lambda}\, \mathbf{p}^{(i)}, \quad i = 1, 2, \ldots, n, \\ w^{(0)} &= \frac{\lambda}{n + \lambda}, \\ w^{(\pm i)} &= \frac{\lambda}{2\,(n + \lambda)}, \quad i = 1, 2, \ldots, n, \end{aligned}$$

where

- $\mathbf{p}_i$ is the i-th column of the square root of $\mathbf{P}$, which is the matrix S such that $SS^{\mathrm{T}} = \mathbf{P}$, sometimes denoted as $\mathbf{P}^{1/2}$,

- λ is a scaling parameter, defined as

$$\lambda = \alpha^2 (n + \kappa) - n, \quad 0 < \alpha < 1,$$

- α and κ describe the spread of the σ-points around the mean, with $\kappa = 3 - n$ usually,

- β is used to include prior information on non-Gaussian distributions of $\mathbf{x}$.

For the covariance matrix, the weight $w^{(0)}$ is modified to

$$w^{(0)} = \frac{\lambda}{n + \lambda} + \left(1 - \alpha^2 + \beta\right).$$

These points and weights ensure that the means and covariances are consistently captured by the UT.

UKF - algorithm

Theorem 7. *The UKF for the nonlinear system (18)-(19) computes a Gaussian approximation of the filtering distribution*

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathcal{N}\left(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k\right),$$

based on the UT, following the three-stage process:

***Initialization**: Define the prior mean $\mathbf{m}_0$, prior covariance $\mathbf{P}_0$ and the parameters α, β, κ.*

***Prediction**:*

Compute the σ-points and weights,

$$
\begin{aligned}
\mathbf{x}_{k-1}^{(0)} &= \mathbf{m}_{k-1}, \\
\mathbf{x}_{k-1}^{(\pm i)} &= \mathbf{m}_{k-1} \pm \sqrt{n+\lambda}\,\mathbf{p}_{k-1}^{(i)}, \quad i = 1, 2, \ldots, n, \\
w^{(0)} &= \frac{\lambda}{n+\lambda}, \\
w^{(\pm i)} &= \frac{\lambda}{2\,(n+\lambda)}, \quad i = 1, 2, \ldots, n.
\end{aligned}
$$

Propagate the σ-points through the dynamic model

$$
\tilde{\mathbf{x}}_k^{(i)} = \mathcal{M}_{k-1}\left(\mathbf{x}_{k-1}^{(i)}\right).
$$

Compute the predictive distribution mean and covariance,

$$
\begin{aligned}
\mathbf{m}_k^- &= \sum_{i=0}^{\pm n} w^{(i)} \tilde{\mathbf{x}}_k^{(i)} \\
\mathbf{P}_k^- &= \sum_{i=0}^{\pm n} w^{(i)} \left(\tilde{\mathbf{x}}_k^{(i)} - \mathbf{m}_k^-\right)\left(\tilde{\mathbf{x}}_k^{(i)} - \mathbf{m}_k^-\right)^{\mathrm{T}} + \mathbf{Q}_{k-1}.
\end{aligned}
$$

***Correction**:*

Compute the updated σ-points and weights,

$$\begin{aligned}
\mathbf{x}_k^{(0)} &= \mathbf{m}_k^-, \\
\mathbf{x}_k^{(\pm i)} &= \mathbf{m}_k^- \pm \sqrt{n+\lambda}\,\mathbf{p}_k^{(i)-}, \quad i = 1, 2, \ldots, n, \\
w^{(0)} &= \frac{\lambda}{n+\lambda} + \left(1 - \alpha^2 + \beta\right), \\
w^{(\pm i)} &= \frac{\lambda}{2\,(n+\lambda)}, \quad i = 1, 2, \ldots, n.
\end{aligned}$$

Propagate the updated σ-points through the measurement model

$$\tilde{\mathbf{y}}_k^{(i)} = \mathcal{H}_k\left(\mathbf{x}_k^{(i)}\right).$$

Compute the predicted mean and innovation, predicted measurement covariance, state-

measurement cross-covariance, and filter gain,

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \sum_{i=0}^{\pm n} w^{(i)} \tilde{\mathbf{y}}_k^{(i)}, \quad \textit{the mean,} \\
\mathbf{d}_k &= \mathbf{y}_k - \boldsymbol{\mu}_k, \quad \textit{the innovation,} \\
\mathbf{S}_k &= \sum_{i=0}^{\pm n} w^{(i)} \left(\tilde{\mathbf{y}}_k^{(i)} - \boldsymbol{\mu}_k \right) \left(\tilde{\mathbf{y}}_k^{(i)} - \boldsymbol{\mu}_k \right)^{\mathrm{T}} + \mathbf{R}_k, \ \textit{measur cov} \\
\mathbf{C}_k &= \sum_{i=0}^{\pm n} w^{(i)} \left(\mathbf{x}_k^{(i)} - \mathbf{m}_k^{-} \right) \left(\tilde{\mathbf{y}}_k^{(i)} - \boldsymbol{\mu}_k \right)^{\mathrm{T}}, \ \textit{s-m cross-cov,} \\
\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1}, \quad \textit{the Kalman gain.}
\end{aligned}
$$

Finally, ***update*** *the filter mean and covariance,*

$$
\begin{aligned}
\mathbf{m}_k &= \hat{\mathbf{m}}_k + \mathbf{K}_k \mathbf{d}_k, \\
\mathbf{P}_k &= \hat{\mathbf{P}}_k - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{T}}.
\end{aligned}
$$

- Just as was the case with the EKF, the UKF can also be applied to the non-additive noise model (20)-(21).

⇒ This is achieved by applying a non-additive version of the UT. Details can be found in [Sarkka2013].

Particle filters

- What happens if both the models are nonlinear and the pdfs are non Gaussian?

- The Kalman filter and its extensions are no longer optimal and, more importantly, can easily fail the estimation process. Another approach must be used.

- A promising candidate is the *particle filter* (PF)

- The particle filter [Doucet2011] (and references therein) works sequentially in the spirit of the Kalman filter, but unlike the latter, it handles an ensemble of states (the particles) whose distribution approximates the pdf of the true state.

- Bayes' rule (8) and the marginalization formula,

$$p(x) = \int p(x \mid y) p(y) \, \mathrm{d}y,$$

are explicitly used in the estimation process.

- The linear and Gaussian hypotheses can then be ruled out, in theory.

- In practice though, the particle filter cannot yet be applied to very high dimensional systems (this is often referred to as "the curse of dimensionality"). Though recent work by [Friedemann, Raffin2023] has improved this by sophisticated parallel computing.

- Particle filters are methods for obtaining *Monte Carlo approximations* of the solutions of the Bayesian filtering equations.

  ⇒ Rather than trying to compute the exact solution of the Bayesian filtering equations, the transformations of such filtering (Bayes' rule for the analysis, model propagation for the forecast) are applied to the members of the sample.

  ⇒ The statistical moments are meant to be those of the targeted pdf.

$\Rightarrow$ Obviously this sampling strategy can only be exact in the asymptotic limit; that is, in the limit where the number of members (or particles) goes to infinity.

- The most popular and simple algorithm of Monte Carlo type that solves the Bayesian filtering equations is called the *bootstrap particle filter*. It is computed by a three-stage process.

Sampling We consider a sample of particles $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$. The related probability density function at time t_k is $p_k(\mathbf{x})$, where

$$p_k(\mathbf{x}) \simeq \sum_{i=1}^{M} \omega_i^k \delta(\mathbf{x} - \mathbf{x}_k^i)$$

and δ is the Dirac mass and the sum is meant to be an approximation of the exact density that the samples emulate. A positive scalar, ω_k^i, weights the importance of particle i within the ensemble.

At this stage, we assume that the weights ω_i^k are uniform and $\omega_i^k = 1/M$

Forecast At the forecast step, the particles are propagated by the model without approximation,

$$p_{k+1}(\mathbf{x}) \simeq \sum_{i=1}^{M} \omega_k^i \delta(\mathbf{x} - \mathbf{x}_{k+1}^i),$$

with $\mathbf{x}_{k+1}^i = \mathcal{M}_{k+1}(\mathbf{x}_k)$. A stochastic noise can optionally be added to the dynamics of each particle.

Analysis The analysis step of the particle filter is extremely simple and elegant. The rigorous implementation of Bayes' rule ascribes to each particle a statistical weight that corresponds to the likelihood of the particle given the data. The weight of each particle is updated according to

$$\omega_{k+1}^{\mathrm{a},i} \propto \omega_{k+1}^{\mathrm{f},i} p(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}^i) \,.$$

It is remarkable that the analysis is carried out with only a few multiplications. It does not involve inverting any system or matrix, as opposed for instance to the Kalman filter.

Choosing a Filter

One usually has to choose between

- linear Kalman filters
- ensemble Kalman filters
- nonlinear filters
- hybrid variational-filter methods.

These questions are resumed in the following Table:

Estimator	Model type	pdf	CPU-time
KF	linear	Gaussian	low
EKF	locally linear	Gaussian	low-medium
UKF	nonlinear	Gaussian	medium
EnKF	nonlinear	Gaussian	medium-high
PF	nonlinear	non-Gaussian	high

Table 1: Decision matrix for choice of Kalman filters.

EXAMPLES

Codes

Various open-source repositories and codes are available for both academic and operational data assimilation.

1. DARC: https://research.reading.ac.uk/met-darc/ from Reading, UK.

2. DAPPER: https://github.com/nansencenter/DAPPER from Nansen, Norway.

3. DART: https://dart.ucar.edu/ from NCAR, US, specialized in ensemble DA.

4. OpenDA: https://www.openda.org/.

5. Verdandi: http://verdandi.sourceforge.net/ from INRIA, France.

6. PyDA: `https://github.com/Shady-Ahmed/PyDA`, a Python implementation for academic use.

7. Filterpy: `https://github.com/rlabbe/filterpy`, dedicated to KF variants.

8. EnKF; `https://enkf.nersc.no/`, the original Ensemble KF from Geir Evensen.

References

1. K. Law, A. Stuart, K. Zygalakis. *Data Assimilation. A Mathematical Introduction*. Springer, 2015.

2. S. Sarkka. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

3. S. Sarkka, A. Solin. Applied Stochastic Differential Equations. Cambridge University Press, 2019.

4. G. Evensen. *Data assimilation, The Ensemble Kalman Filter*, 2nd ed., Springer, 2009.

5. A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* SIAM. 2005.

6. O. Talagrand. Assimilation of observations, an introduction. *J. Meteorological Soc. Japan*, **75**, 191–209, 1997.

7. F.X. Le Dimet, O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus,* **38**(2), 97–110, 1986.

8. J.-L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Rev.*, **30**(1):1–68, 1988.

9. J. Nocedal, S.J. Wright. *Numerical Optimization*. Springer, 2006.

10. F. Tröltzsch. *Optimal Control of Partial Differential Equations*. AMS, 2010.