Chapter 10

Data Assimilation: Dynamical State/Parameter Estimation

After considering inverse UQ problems where the focus was parameter estimation in the case of time-independent parameters (no matter whether the state system was stationary or time-dependent), we now turn to the case of inverse UQ problems where the focus is parameter estimation in the case of time-dependent parameter. This problem shares several similarities with *state estimation* (in presence of measurements) and, more generally speaking, with Data Assimilation.

> Data assimilation (DA) is the process by which a numerical model of a given system, usually affected by noise or model uncertainties, is improved by incorporating system observations.

Hence, DA refers to a class of methods and approaches for combining observations with model output with the objective of improving the latter, when we want to predict the state of a system, or its future, in the best possible way (and thus, relying on models). However, when models are not corrected periodically by reality, they can be of little value: thus, we need to fit the model state in an optimal way to the observations, before an analysis or prediction is made. This fitting of a model to observations is a special case (but highly typical) of an inverse UQ problem.

Following [2], we will describe a sequential approach for (time-dependent) parameter estimation when dealing with dynamical systems, relying on *filtering algorithms*, among which the Kalman filter is the most relevant (and classical) instance. Suitable extension of the Kalman filter algorithm must be taken into account to enable its use when either nonlinear dynamical systems or non Gaussian assumptions are used. Since sequential approaches have been originally introduced to perform state estimation, we will mainly focus on this case, *augmenting* the state with the parameter vector in the case of parameter estimation. Note that sequential approaches can be conveniently used also in the case the parameter vector is time-independent, whenever we want to avoid to solve the whole dynamical system (that is, from the initial time on) for each evaluation of the likelihood function for the sake of parameter estimation within a Bayesian context.

10.1 Data Assimilation: Variational/Sequential Approaches

In the case of dynamical systems and time-dependent parameters, parameter estimation shares several similarities and contact points with data assimilation, even if the goal of this latter very often – such as in the case of geophysical fluids, for instance – goes beyond parameter estimation; indeed, it is more often related to *state estimation*, namely, to improving the outcome of the numerical model and of its initial state to correctly initialize forecasts, by assimilating available measurements into the numerical model itself. Data assimilation is intrinsically related to time-varying phenomena and deals with highly nonlinear dynamical systems, very often ill-posed.

the goal is to estimate dynamically the change of the state of the system (and parameters)

So now:
- parameters can change in time
- we want to perform state estimation from partial measurements of the state itself
- we don't want to wait until all the measurements has been acquired to perform estimation (like e.g. in the Bayesian framework (MCMC))

→ sequential estimation

10.1.1 Variational approach: PDE-constrained optimization*

Before focusing on filtering algorithms, we report some basic notions about the variational approach for (state and) parameter estimation in dynamical systems. even if we have not covered this aspect explicitly. This section can however be skipped without compromising the understanding of the following sections.

The variational approach recasts parameter estimation in the framework of PDE-constrained optimization, by considering the equations governing the problem at hand as the state system, and the discrepancy between the observation of the state and the measured data as the cost functional to be minimized. The parameters to be estimated (often involving initial conditions) play the role of optimization variables, just like control variables in the case of optimal control problems; however, unlike the optimal control case, parameters are quantities which no one can actually control – they are often coefficients of the operators appearing in the dynamical system.

Here we provide an abstract formulation of the variational approach, assuming that the state problem has already been discretized in space. We let $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ denote the semi-discrete state, $\boldsymbol{\theta} \in \mathbb{R}^p$ the parameters to be estimated, and $\mathbf{A}(t, \mathbf{x}(t), \boldsymbol{\theta})$ the (semi-discretized in space) state operator; note that usually $p \ll n_x$. The state variable then solves the dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t, \mathbf{x}(t), \boldsymbol{\theta}), \quad t \in (0, T), \tag{10.1a}$$

$$\mathbf{x}(0) = \mathbf{g}. \tag{10.1b}$$

We consider the case where the parameter vector $\boldsymbol{\theta}$ is unknown, and for which the estimation problem consists in finding $\widehat{\boldsymbol{\theta}}$ such that the discrepancy between the observation and a set of measurements $\mathbf{z}(t) \in \mathbb{R}^{n_z}$, $t \in (0, T)$, is minimized, e.g., in a least-squares sense. The case of unknown initial data $\mathbf{g}$ to be estimated can be treated in essentially the same way. We usually assume that measurements $\mathbf{z}(t)$ are related to observations of the true state through an *additive noise* model; that is,

$$\mathbf{z}(t) = H(t)\mathbf{x}(t) + \boldsymbol{\varepsilon}(t), \quad t \in (0, T),$$

where $H = H(t) \in \mathbb{R}^{n_z \times n_x}$ is an observation operator which maps the state space into the observation space $\mathbb{R}^{n_z}$, and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(t)$ is a *noise* term accounting for measurement errors. The following minimization problem is then solved:

$$J(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{2} \int_0^T \|\mathbf{z}(t) - H(t)\mathbf{x}(t)\|_M^2 \, dt + \frac{\alpha_\theta}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_{P_\theta^{-1}}^2 \to \min_{\boldsymbol{\theta} \in \mathcal{P}}, \tag{10.2}$$

where $\mathbf{x} = \mathbf{x}(t)$ is the solution of (10.1). Here $\mathcal{P} \subset \mathbb{R}^{n_x} \times \mathbb{R}^p$ denotes the set of admissible parameters. Additional information is usually added to the least-squares objective expressed by the first term in (10.2), via a background estimate $\boldsymbol{\theta}_0$ of $\boldsymbol{\theta}$; M and P_θ^{-1} are suitable symmetric positive definite matrices (the reason why we consider an inverse matrix to define this latter norm will be clarified in the following). This procedure goes by the name of *Levenberg–Marquardt– Tikhonov regularization*; note that the usual penalization coefficients are embedded in the definition of P_θ^{-1}.

The minimization problem (10.2) can be solved by an optimization algorithm based on the evaluation of the gradient of J with respect to $\boldsymbol{\theta}$; this latter can be computed through a suitable *adjoint problem*. This is a *four-dimensional variational* (4D-Var) assimilation; a three-dimensional variational (3D-Var) assimilation would arise in the case of steady-state systems.

Since measurements $\mathbf{z}$ are only available at a discrete number of times $\tau_1, \ldots, \tau_K$, we formulate the identification problem by replacing the dynamical system (10.1) with its time-discretized version:

$$\mathbf{x}_{k+1} = \mathbf{A}_{k+1}(\mathbf{x}_k, \boldsymbol{\theta}), \quad k = 0, \ldots, K-1, \tag{10.3a}$$

$$\mathbf{x}_0 = \mathbf{g} \tag{10.3b}$$

where $\mathbf{A}_{k+1}$ is a non-linear function describing the evolution of the state from time τ_k to time τ_{k+1} and $\mathbf{x}^k \approx \mathbf{x}(k\Delta\tau)$ denotes the state vector at time τ_k. Note that the length $\Delta\tau = \tau_{k+1} - \tau_k$

of the time window between two subsequent measurements is usually larger than the time step Δt used for time discretization, and that $\boldsymbol{\theta}$ does not depend on k. From here on k will denote the temporal index of system evolution.

We then formulate the following minimization problem

$$J_K(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{z}_k - H_k\mathbf{x}_k\|^2_{M_k} + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2_{P_\theta^{-1}} \rightarrow \min_{\boldsymbol{\theta}\in\mathcal{P}}, \tag{10.4}$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$, and we set

$$\mathbf{z}_k = H_k\mathbf{x}_k + \boldsymbol{\varepsilon}_k. \tag{10.5}$$

Here $\boldsymbol{\varepsilon}_k$ denotes the noise of the measurement device at $\tau_k = k\Delta\tau$; a possible choice for M_k is $M_k = \Delta\tau M$, whereas $H_k \approx H(k\Delta\tau)$. Also in this case a gradient-based optimization procedure can be used to solve the constrained optimization problem (10.3, 10.4) with gradients evaluated by introducing, e.g., a suitable adjoint problem.

10.1.2 Sequential approach: Kalman filter and extensions

A drawback of the variational approach is the need to wait until the whole set of measurements has been acquired in order to perform an optimization step. A sequential approach instead performs the assimilation of acquired measurements *on the fly*, and updates the estimate of the unknown quantities accordingly. We can formulate the problem of *state estimation* as follows. At discrete times k, estimate the system state $\mathbf{x}_k$ using previous observations $\mathcal{D}_k = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, when the model and observation equations are given as

$$\mathbf{x}_k = \mathcal{M}(\mathbf{x}_{k-1}) + \boldsymbol{\varepsilon}^p_k,$$

$$\mathbf{z}_k = \mathcal{K}(\mathbf{x}_k) + \boldsymbol{\varepsilon}^o_k.$$

In the above system:

- $\mathcal{M}$ is the evolution model that evolves the state in time;
- $\mathcal{K}$ is the observation model that maps the state to the observations;
- error (or discrepancy) terms $\boldsymbol{\varepsilon}^p_k$ and $\boldsymbol{\varepsilon}^o_k$ represent model and observation error, respectively.

In sequential (or dynamical) state estimation problems, measurements are obtained in real-time and the state estimate needs to be updated after the measurements are obtained. This can be achieved through a sequential application of the Bayes' formula, where:

- the prior is given by evolving the posterior of the model state from the previous time step using the model $\mathcal{M}$ (prediction step);
- the obtained prior is updated with the likelihood of the measurement (update step) to get the posterior, which is evolved with the model and used as the prior in the next time step.

Repeating this procedure allows 'on-line' estimation of model states.

In formulas, we aim at estimating the marginal distribution of the states $\pi(\mathbf{x}_k \mid \mathcal{D}_k)$ given the measurements obtained until the current time. In the prediction step, the whole distribution of the state is moved with the dynamical model to the next time step:

$$\pi(\mathbf{x}_k \mid \mathcal{D}_{k-1}) = \int \pi(\mathbf{x}_k \mid \mathbf{x}_{k-1})\pi(\mathbf{x}_{k-1} \mid \mathcal{D}_{k-1})d\mathbf{x}_{k-1}. \tag{10.6}$$

When the new observation $\mathbf{z}_k$ is obtained, the model state is updated using the Bayes' rule, with (10.6) as the prior:

$$\pi(\mathbf{x}_k \mid \mathcal{D}_k) \propto \pi(\mathbf{z}_k \mid \mathbf{x_k})\pi(\mathbf{x_k} \mid \mathcal{D}_{\mathbf{k-1}}).$$

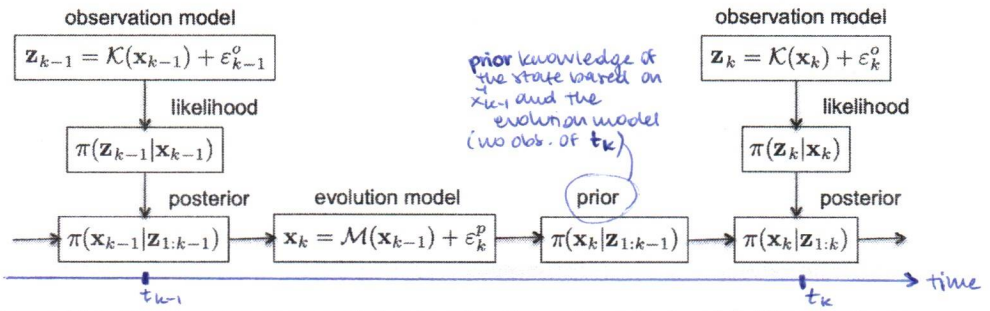


Figure 10.1: Two iterations of sequential state estimation at times $k-1$ and k. The prior is given by the model, which is combined with the observation to get the posterior. The posterior is further propagated to be the prior for the next time point.

This posterior is used inside integral (10.6) in the next prediction step. The term $\pi(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ includes the evolution model and describes the probability of having state $\mathbf{x}_k$ at time k, given that the state was $\mathbf{x}_{k-1}$ at the previous time step; see Figure 10.1.

Different state estimation methods can be derived, depending on the assumptions of the form of the state distribution, the likelihood and the techniques used to evolve the uncertainty of the state in time. Some of the most common methods are introduced in the following.

10.2 The Kalman Filter

It can be seen as:
1. recursive least squares problems
2. sequential bayesian estimation framework
3. conditioning of Gaussian measures

all these paths lead to Kalman filtering, the more simple is the 1. (we'll focus on that one)

A numerical milestone for the solution of sequential estimation problems, the Kalman filter (KF), was introduced in the 60s' as a recursive filter for the estimation of the state of a noisy dynamical system from a set of measurements, that is, to improve the prediction of the state dynamics by taking into account additional data. The KF is meant for situations where the model is linear and the model and observation errors are assumed to be zero mean Gaussians with given covariance matrices. The extended Kalman filter (EKF) is its extension to nonlinear situations, where the model is linearized and the KF formulas are applied.

In this section we formulate the KF, exploiting the analogy with the solution of a recursive least-squares problem yielding the best linear unbiased estimator for a linear model; then, we will show how the so-called push-forward and subsequent conditioning of Gaussian measures, as well as a sequential Bayesian estimation framework, are indeed closely related to the KF.

10.2.1 The link with least-squares estimators

The KF algorithm sequentially generates an estimate of the unknown quantity by means of a linear combination of the current estimate and the acquired measurement. Let us first consider the case where there is no dynamics, and data are generated by the linear model

$$\mathbf{z} = H\mathbf{x} + \boldsymbol{\varepsilon}, \tag{10.7}$$

where H is a given $n_z \times n_x$ matrix of rank n_z (usually, $n_z \ll n_x$), $\boldsymbol{\varepsilon}$ is an n_z-dimensional random variable with zero mean and known positive definite covariance $Q = Cov(\boldsymbol{\varepsilon}) = \mathbb{E}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top\right] > 0$, and $\mathbf{z}$ denotes known, but inexact, measurements of the state with errors given by $\boldsymbol{\varepsilon}$. The vector $\mathbf{x} \in \mathbb{R}^{n_x}$ is the quantity to be estimated from the observation $\mathbf{z}$. Which is the best estimator $\hat{\mathbf{x}}$ we can find from the measurements $\mathbf{z}$?

Among all linear estimators of $\mathbf{x}$, that is, estimators of the form $\hat{\mathbf{x}} = B\mathbf{z}$ for some matrix $B \in \mathbb{R}^{n_x \times n_z}$, which are unbiased (i.e. $\mathbb{E}[\hat{\mathbf{x}}] = \mathbf{x}$), the best choice is the one that minimizes the

mean-square error $\mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|^2\right]$, that is, the *best linear unbiased estimator* (BLUE), that is,

$$\hat{\mathbf{x}}_{BLUE} = \arg \min_{\hat{\mathbf{x}}=B\mathbf{z}:\ \mathbb{E}[\hat{\mathbf{x}}]=\mathbf{x}} J(\hat{\mathbf{x}}), \qquad J(\hat{\mathbf{x}}) = \mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|^2\right] \tag{10.8}$$

By the Gauss–Markov theorem, the best or minimum variance linear unbiased estimator, solution of (10.8), is given by

$$\hat{\mathbf{x}}_{BLUE} = \hat{B}\mathbf{z} \quad \text{where} \quad \hat{B} = (H^\top Q^{-1} H)^{-1} H^\top Q^{-1}, \tag{10.9}$$

provided H is full rank. Note that the covariance of $\hat{x}_{BLUE}$ is

$$Cov(\hat{\mathbf{x}}_{BLUE}) = \mathbb{E}\left[(\hat{\mathbf{x}}_{BLUE} - \mathbf{x})(\hat{\mathbf{x}}_{BLUE} - \mathbf{x})^\top\right] = (H^\top Q^{-1} H)^{-1} = (\nabla^2 J(\hat{x}_{BLUE}))^{-1},$$

that is, the covariance of $\hat{x}_{BLUE}$ is the inverse of the Hessian of J, and this latter does not depend on the estimate. These expressions should be quite familiar, if compared to the least squares estimators introduced in Section 9.1 (note, however, the change of notation compared to the problem of parameter estimation treated in Chapter 9).

- **Remark 10.2.1.** *Note that if* $Q = \sigma^2 I$,

$$\hat{\mathbf{x}}_{BLUE} = (H^\top H)^{-1} H^\top \mathbf{z}$$

and

$$\hat{\mathbf{x}}_{BLUE} = \arg\min_{\mathbf{x}} \frac{1}{2}\|H\mathbf{x} - \mathbf{z}\|^2;$$

in particular, $Cov(\hat{\mathbf{x}}_{BLUE}) = (H^\top H)^{-1} = (\nabla^2 J(\hat{x}_{BLUE}))^{-1}$.

Hence, in general, the BLUE (10.9) *can be obtained by solving the weighted least-squares problem*

$$\hat{\mathbf{x}}_{BLUE} = \arg\min_{\mathbf{x}} \frac{1}{2}\|H\mathbf{x} - \mathbf{z}\|^2_{Q^{-1}}.$$

Note that

$$\nabla J(\mathbf{x}) = H^\top Q^{-1} H\mathbf{x} - H^\top Q^{-1}\mathbf{z}.$$

What happens in the case where we already know an *a priori* (or background) estimate $\mathbf{x}_0$ of $\mathbf{x}$, with covariance matrix $Cov(\mathbf{x}_0) = P^-$? In this case, we can try to find

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|H\mathbf{x} - \mathbf{z}\|^2_{Q^{-1}} + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{(P^-)^{-1}};$$

(P^- where "$-$" because it is the background)

we try to minimize the error coming from both sources of information

this is also referred to as *3D variational data assimilation problem*. Computing the gradient of J and imposing that it must vanish, we have

$$\nabla J(\hat{\mathbf{x}}) = (P^-)^{-1}(\hat{\mathbf{x}} - \mathbf{x}_0) - H^\top Q^{-1}(\mathbf{z} - H\hat{\mathbf{x}}) = \mathbf{0}$$

so that

$$(P^-)^{-1}(\hat{\mathbf{x}} - \mathbf{x}_0) = H^\top Q^{-1}(\mathbf{z} - H\hat{\mathbf{x}}),$$

that is,

$$((P^-)^{-1} + H^\top Q^{-1} H)\hat{\mathbf{x}} = H^\top Q^{-1}\mathbf{z} + (P^-)^{-1}\mathbf{x}_0.$$

Hence, we obtain:

$$\begin{aligned}
\hat{\mathbf{x}} &= \underbrace{\left(((P^-)^{-1} + H^\top Q^{-1} H)\right)^{-1}}_{P^+} (H^\top Q^{-1}\mathbf{z} + (P^-)^{-1}\mathbf{x}_0) \\
&= P^+\left[(P^-)^{-1}\mathbf{x}_0 + H^\top Q^{-1} H\mathbf{x}_0 - H^\top Q^{-1} H\mathbf{x}_0 + H^\top Q^{-1}\mathbf{z}\right] \\
&= P^+\left[(P^+)^{-1}\mathbf{x}_0 - H^\top Q^{-1} H\mathbf{x}_0 + H^\top Q^{-1}\mathbf{z}\right] \\
&= \mathbf{x}_0 - P^+ H^\top Q^{-1} H\mathbf{x}_0 + P^+ H^\top Q^{-1}\mathbf{z}.
\end{aligned}$$

By defining the updated covariance matrix as

$$P^+ = \left(((P^-)^{-1} + H^\top Q^{-1} H)\right)^{-1} \tag{10.10}$$

= update of the covariance ($P^- \to P^+$)

and the *Kalman gain matrix* as

$$K = P^+ H^\top Q^{-1}$$

we finally obtain, instead of the BLUE (10.9), the estimator

$$\hat{\mathbf{x}} = \mathbf{x}_0 + K(\mathbf{z} - H\mathbf{x}_0). \tag{10.11}$$

Note that this estimate is given by a linear combination of the background estimate $\mathbf{x}_0$ and the so-called *innovation* $\mathbf{z} - H\mathbf{x}_0$. Note that, thanks to the so-called Sherman-Morrison-Woodbury identity, the Kalman gain matrix can also be evaluated by exploiting the prior covariance P^- instead of the updated covariance P^+, as

$$K = P^+ H^\top Q^{-1} = P^- H^\top (H P^- H^\top + Q)^{-1}. \tag{10.12}$$

10.2.2 The link with prior/posterior update of Gaussian distributions

Let us now go back to Example 9.4, and interpret it in light of what we have just obtained. Assume that the observation model $\mathbf{z} = H\mathbf{x}$ has a Gaussian prior and a Gaussian likelihood function, that is,

$$\mathbf{x} \sim N(\mathbf{x}_0, P^-)$$

and

$$\mathbf{z}|\mathbf{x} \sim N(H\mathbf{x}, Q),$$

then the posterior conditional distribution of $\mathbf{x}|\mathbf{z}$ takes the form

$$\mathbf{x}|\mathbf{z} \sim N(\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}, \Sigma_{\mathbf{x}|\mathbf{z}})$$

where

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}} = \mathbf{x}_0 + K(\mathbf{z} - H\mathbf{x}_0)$$

and

$$\Sigma_{\mathbf{x}|\mathbf{z}} = \left(((P^-)^{-1} + H^\top Q^{-1} H)\right)^{-1} = P^+,$$

so that we could obtain the expression of the estimator $\hat{\mathbf{x}}$ also by maximizing the likelihood function.

now time comes into play →

10.2.3 The Kalman Filter Algorithm

We now consider the case of a time-discretized linear system, with data acquired over a time interval. Then the model (10.7) is replaced by

$$\begin{aligned} \mathbf{x}_k &= A_k \mathbf{x}_{k-1} + \boldsymbol{\varepsilon}_k^p, \quad k = 1, \dots, K, \qquad \mathbf{x}_0 = \mathbf{g}, \\ \mathbf{z}_k &= H_k \mathbf{x}_k + \boldsymbol{\varepsilon}_k^o, \end{aligned} \tag{10.13}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ denotes the state vector and $\mathbf{z}_k \in \mathbb{R}^{n_z}$ are the measurements; for each k, $A_k \in \mathbb{R}^{n_x \times n_x}$ is a given matrix. Moreover, $\boldsymbol{\varepsilon}_k^p$ and $\boldsymbol{\varepsilon}_k^o$ are uncorrelated zero-mean random noise processes with positive-definite covariances Q_k and R_k, modeling the uncertainty of the model and the additive noise in the observation, respectively. The state estimation problem is the problem of finding the state $\mathbf{x}_k$ given k known observations $\mathbf{z}_1, \dots, \mathbf{z}_k$.

A_k ∈ evolution model represented through a (known) matrix

> The Kalman filter is a recursive algorithm that provides the best linear unbiased estimate $\mathbf{x}_k^a$ of $\mathbf{x}_k$ in terms of both the previous estimate $\mathbf{x}_{k-1}^a$ and the latest data $\mathbf{z}_k$ up to that point in time.

$\mathbf{x}_k^a$ where a stays for assimilation

The idea of filtering is essentially to estimate the state vector $\mathbf{x}_k$ for time steps $k = 1, 2, \ldots$. In practice this is done by applying the Bayes' rule sequentially so that the prediction from the previous time step is considered as the prior, which is updated with the new measurements that become available.
Hence, the Kalman filter is based on a *predictor-corrector* strategy, consisting of the following steps:

(1). a *prediction* step (called *forecast* or time update) consists in letting the system dynamics evolve from $\mathbf{x}_{k-1}^a$ without taking into account the observations, yielding the *forecast* state $\mathbf{x}_k^f$;

(2). a *correction* step (called *analysis* or measurement update) updates the forecast state $\mathbf{x}_k^f$ by *assimilating* the measurements into the model, yielding the *assimilated* state $\mathbf{x}_k^a$.

To derive the expression of the prediction step, note that state prediction would yield

$$\mathbf{x}_k^f = A_k \mathbf{x}_{k-1}^a$$

and the error covariance prediction results from the fact that $Cov(\mathbf{x}_{k-1}^a) = P_{k-1}^a$ and the fact that $Cov(\boldsymbol{\varepsilon}_k^p) = Q_k$ as

$$P_k^f = A_k P_{k-1}^a A_K^\top + Q_k.$$

Indeed,

$$\begin{aligned} P_k^f &= \mathbb{E}\left[(\mathbf{x}_k^f - \mathbf{x}_k)(\mathbf{x}_k^f - \mathbf{x}_k)^\top\right] \\ &= \mathbb{E}\left[(A_k\mathbf{x}_{k-1}^a - A_k\mathbf{x}_{k-1} - \boldsymbol{\varepsilon}_k^p)(A_k\mathbf{x}_{k-1}^a - A_k\mathbf{x}_{k-1} - \boldsymbol{\varepsilon}_k^p)^\top\right] = A_k P_{k-1}^a A_k^\top + Q_k \end{aligned}$$

To derive the expression of the correction step, we need to assimilate the new observation $\mathbf{z}_k$, and suppose that the current prediction based on observations $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$ is $\mathbf{x}_k^f$, with covariance matrix P_k^f. The state correction is provided by the best linear unbiased estimator of this system (see (10.9)) and is given by

$$\mathbf{x}_k^a = \mathbf{x}_k^f + K_k(\mathbf{z}_k - H_k\mathbf{x}_k^f) \tag{10.14}$$

resulting from the linear combination of the current estimate and the *last* observation $\mathbf{z}_k$, where the Kalman gain is evaluated as

$$K_k = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}.$$

Finally, we correct the error covariance exploiting the prior/posterior correction of the covariance matrix, and the definition of the Kalman gain matrix, as

$$\begin{aligned} P_k^a &= ((P_k^f)^{-1} + H_k^\top R_k^{-1} H_k)^{-1} \\ &= (I - K_k H_k) P_k^f (I - K_k H_k)^\top + K_k R_k K_k^\top = (I - K_k H_k) P_k^f. \end{aligned}$$

Note the formal analogy with (10.10), where P_k^f and P_k^a now play the role of P^- and P^+, respectively.

Grouping the prediction and the correction steps together, we finally obtain the *k*th step of the KF algorithm:

$$\begin{aligned} \mathbf{x}_k^f &= A_k \mathbf{x}_{k-1}^a && \text{state prediction,} && (10.15a)\\ P_k^f &= A_k P_{k-1}^a A_k^\top + Q_k && \text{error covariance prediction,} && (10.15b)\\ K_k &= P_f^k H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1} && \text{Kalman gain evaluation,} && (10.15c)\\ \mathbf{x}_k^a &= \mathbf{x}_k^f + K_k(\mathbf{z}_k - H_k\mathbf{x}_k^f) && \text{state correction,} && (10.15d)\\ P_k^a &= (I - K_k H_k) P_k^f && \text{error covariance correction.} && (10.15e) \end{aligned}$$

(in a few words:) a prediction-correction strategy

also here (as in prior/posterior update of Gaussian distributions) we see that the updated information (posterior) is expressed as a merging of two different pieces of information: what we had before (looking only at the state prediction through the evolution model) and how the innovation regarding the new observations that have come can be considered.

posterior info = w_1 · old info + w_2 observations

Note that from (10.16d) only, the estimated state from the above step and the current measurement are needed to compute the estimate of the current state. The two prediction and correction steps alternate: the prediction advances the state until the next measurement is acquired, and then the correction incorporates this measurement.

Remark 10.2.2. * *Note that in the linear case, the variational and the sequential approaches yield the same result at the end of a time window, provided the following assumptions are made: the same background estimation and the same covariance matrices are used, and the same measurements are acquired – that is, both algorithms are optimal from a least-squares or minimum variance standpoint.*

Dynamical parameter estimation

Let us now return to our problem of estimating the parameter $\boldsymbol{\theta}$. We can apply the KF algorithm to the system

$$\begin{aligned} &\mathbf{x}_k = A_k \mathbf{x}_{k-1} + B_k \boldsymbol{\theta}_k + \boldsymbol{\varepsilon}_k^p, \quad k = 1, \dots, K, \quad \mathbf{x}_0 = \mathbf{g}, \\ &\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} \end{aligned}$$

with observations

$$\mathbf{z}_k = H_k \mathbf{x}_k + \boldsymbol{\varepsilon}_k^o, \quad k = 1, \dots, K.$$

Here $\boldsymbol{\theta}_k \in \mathbb{R}^p$ denotes the parameter vector and (under the linearity assumption) $B_k \in \mathbb{R}^{n_x \times p}$, and we assume that no random error is associated with model parameters. This is the so-called *state augmentation* technique. In order to exploit the KF algorithm, we consider $\widetilde{\mathbf{x}}_k = (\mathbf{x}_k, \boldsymbol{\theta}_k)^\top$ as state vector instead of $\mathbf{x}_k$, thus yielding the *augmented* KF algorithm

Grouping the prediction and the correction steps together, we finally obtain the *k*th step of the KF algorithm:

$$\begin{aligned}
\tilde{\mathbf{x}}_k^f &= \tilde{A}_k \tilde{\mathbf{x}}_{k-1}^a && \text{state prediction,} && (10.16a)\\
\tilde{P}_k^f &= \tilde{A}_k \tilde{P}_{k-1}^a \tilde{A}_k^\top + \tilde{Q}_k && \text{error covariance prediction,} && (10.16b)\\
K_k &= \tilde{P}_f^k \tilde{H}_k^\top (\tilde{H}_k \tilde{P}_k^f \tilde{H}_k^\top + \tilde{R}_k)^{-1} && \text{Kalman gain evaluation,} && (10.16c)\\
\tilde{\mathbf{x}}_k^a &= \tilde{\mathbf{x}}_k^f + K_k(\tilde{\mathbf{z}}_k - \tilde{H}_k \tilde{\mathbf{x}}_k^f) && \text{state correction,} && (10.16d)\\
\tilde{P}_k^a &= (I - K_k \tilde{H}_k) \tilde{P}_k^f && \text{error covariance correction.} && (10.16e)
\end{aligned}$$

where

$$\widetilde{A}_k = \begin{bmatrix} A_k & B_k \\ 0 & I \end{bmatrix}, \quad \tilde{\mathbf{z}}_k = \begin{bmatrix} \mathbf{z}_k \\ 0 \end{bmatrix},$$
$$\tilde{H}_k = \begin{bmatrix} H_k & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{Q}_k = \begin{bmatrix} Q_k \\ 0 \end{bmatrix}, \quad \tilde{R}_k = \begin{bmatrix} R_k & 0 \\ 0 & 0 \end{bmatrix}.$$

We point out that, by construction of the filtering procedure, the estimated parameter values evolve along the simulation period and the actual estimation is achieved with the final values; that is, the estimated parameter vector is $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_K^a$ (note that in the current formulation θ is independent of time). Hence, we expect these estimation *trajectories* to fluctuate less and less during the course of the simulation; the non-converging case would therefore denote the presence of persistent modeling errors.

We conclude this section by pointing out that when a sequential approach like the Kalman filter is used for parameter estimation, the dynamical system has to be solved only once, by

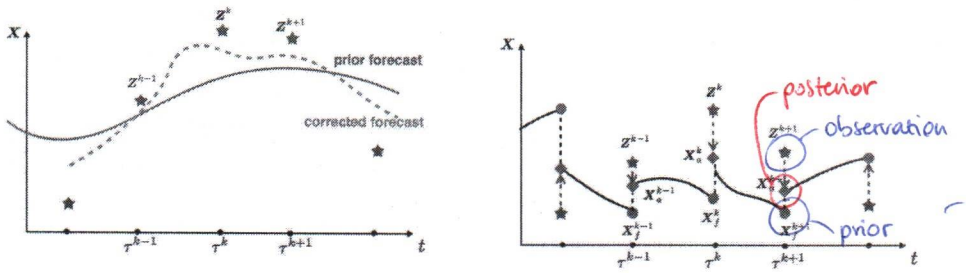


Figure 10.2: Variational approach (left) versus KF (right) approach: in the former, at each optimization stage the whole state dynamics has to be computed, whereas in the latter each measurement is sequentially used for the state (and parameter) correction.

updating the parameter value after each assimilation of new measurements. On the other hand, a variational approach would require us to solve the dynamical system on the whole time interval several times, assuming that an iterative approach is used to perform the optimization: see the sketch in Figure 10.2.

10.3 The extended Kalman filter

The classical KF formulation is well suited to low-dimensional, linear dynamical systems, although in real applications these assumptions are seldom verified. The extended Kalman filter (EKF) was introduced for non-linear dynamical systems, where (10.13) is replaced (here τ_k denotes discrete time instants) by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \tau_k) + \boldsymbol{\varepsilon}_k^p, \quad \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \tau_k) + \boldsymbol{\varepsilon}_k^o, \tag{10.17}$$

where $\mathbf{f}$ and $\mathbf{h}$ are two nonlinear functions. Nonlinearity here involves both the system dynamics and the observation model. The EKF consists in applying the KF algorithm (10.15) to a linearized version of (10.18) around the previous state, so that at each step we set

$$A_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^a_{k-1}}, \quad H_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}^f_k}.$$

At each step, two Jacobian matrices have to be evaluated at the current predicted state/parameters. A similar extension of the algorithm (10.16) provides the EKF for parameter estimation. Although feasible in principle, the EKF suffers from several drawbacks. For instance, it entails prohibitive computational costs to invert large matrices and to propagate the covariance matrix in time. Even more importantly, the EKF may lack stability, meaning that as the estimated state deviates from the true state, the linearized model becomes inaccurate, which may lead to an even larger error in state estimation. To mitigate these shortcomings, several strategies have been put in place: low-rank approximation of the covariance matrices have been considered, and other extensions of the KF, such as the Ensemble Kalman Filter (EnKF), have been introduced.

An extension to the state-parameter estimation problem is also possible, in the case where

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1}, \tau_k) + \boldsymbol{\varepsilon}_k^p, \quad \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \tau_k) + \boldsymbol{\varepsilon}_k^o, \tag{10.18}$$

with $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1}$, $k = 1, 2, \ldots$. In this case, considering $\tilde{\mathbf{x}}_k = (\mathbf{x}_k, \boldsymbol{\theta}_k)^\top$, we have

$$A_k = \left.\frac{\partial \mathbf{f}}{\partial \tilde{\mathbf{x}}}\right|_{\mathbf{x}^a_{k-1}, \boldsymbol{\theta}^k}, \quad H_k = \left.\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}}\right|_{\mathbf{x}^k_f}.$$

10.3.1 A remarkable example: data assimilation for the Lorenz system

The goal in this example is to simply illustrate the EKF algorithm. To demonstrate the data assimilation algorithm in practice, we consider the Lorenz system: for $t > 0$,

$$\begin{cases} x' &= \sigma(y - x) \\ y' &= rx - y - xz \\ z' &= xy - bz \end{cases} \tag{10.19}$$

parameters $[\sigma, r, b]$ can be difficult to estimate

moreover the system is very sensible to initial conditions → because of this we want to include some knowledge coming from observations as soon as we get obs.

The vector $\mathbf{x} = [x\ y\ z]^\top$ is the three-degree of freedom state vector whose nonlinear evolution $f(\mathbf{x})$ is specified by the right hand side (Lorenz model) in the system above. The Lorenz equations are a highly simplified model of convective- driven atmospheric motion. Here we follow [25] for the formulation and the analysis of an extended KF algorithm.

To begin, let us show a simulation of the Lorenz system that includes specified initial conditions without noise and an evolution which is free of stochastic forcing; this simulation will be the truth that the data assimilation will try to reproduce. The parameters to be simulated here, and in what follows, are standard in many examples: $\sigma = 10$, $b = 8/3$, $r = 28$. The large value of r puts the system in a dynamical state that is highly sensitive to initial conditions. The perfect initial conditions will be $\mathbf{x}_0 = [5\ 5\ 5]^\mathrm{T}$. The following Matlab code solves this problem for the time domain $t \in [0, 20]$ and plots it in a parametric way in three dimensions.

```
t=0:0.01:20;
sigma=10; b=8/3; r=28;
x_0=[5 5 5];
[t,xsol]=ode45('lor_rhs',t,x0,[],sigma,b,r)

x_true=xsol(:,1); y_true=xsol(:,2); z_true=xsol(:,3);
figure(1), plot3(x_true,y_true,z_true)
```

where the right-hand side of the differential equation, including the dynamics through $\mathbf{f}(\mathbf{x})$, is

```
function rhs=lor_rhs(t,x,dummy,sigma,b,r)
rhs=[sigma*(-x(1)+x(2))
            -x(1)*x(3)+r*x(1)-x(2)
             x(1)*x(2)-b*x(3)];
```

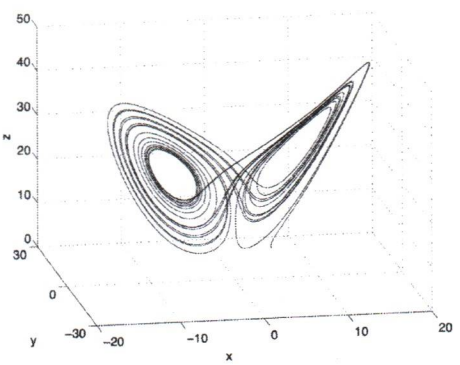


Figure 10.3: Evolution of the variables x(t), y(t) and z(t) over the time period $t \in [0, 20]$ for $\sigma = 10$, $b = 8/3$ and $r = 28$. The evolution is shown parametrically as a function of time.

The results of simulating the Lorenz equation are shown in Figs. 10.3–10.4. The first of these figures demonstrates the standard butterfly pattern of evolution of the strange attractor in three

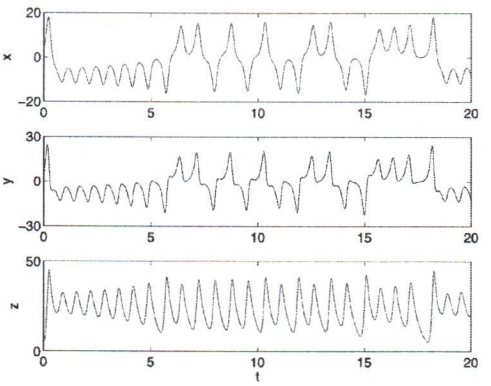


Figure 10.4: Time evolution of the variables $x(t)$, $y(t)$ and $z(t)$ over the time period $t \in [0, 20]$ for $\sigma = 10$, $b = 8/3$, $r = 28$ and $\mathbf{x}_0 = [5\ 5\ 5]^{\mathrm{T}}$. In this parameter regime, specifically for this large a value of r, the dynamics of the Lorenz equations are highly sensitive to initial conditions.

dimensions. The second of these figures illustrates the evolution of the variables $x(t)$, $y(t)$ and $z(t)$ over the time period $t \in [0, 20]$. In what follows, instead of tracking and comparing all the variables, we will focus on $x(t)$ for illustrative purposes only.

Sensitivity to initial conditions

The first thing that will be investigated is sensitivity of the evolution to small changes in the initial conditions. Let us assume that the error (in the initial data) has a Gaussian distribution so that

$$\mathbf{x}(0) = \mathbf{x}_0 + \sigma_2 q(0, 1)$$

where $\mathbf{x}_0$ is the perfect initial conditions and $q(0, 1)$ is a Gaussian distributed random vector with zero mean, independent components with unit variance. With this error, the evolution of the Lorenz system can once again be explored.

In Fig. 10.5 eight realizations of the evolution are given using this initial conditions. The exact evolution which we are trying to model (with `sigma2 = 0`) is the thinner solid line while the initial conditions with error (with $\sigma_2 = 1$) is the bolded line.

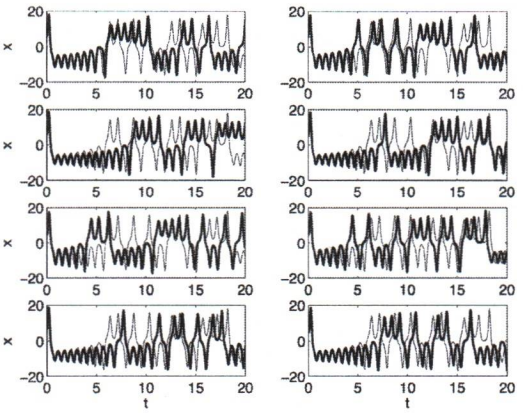
The following `Matlab` code generates the eight realizations:

```
sigma2 = 1; % error variance
for j = 1 : 8
   xic = x0 + sigma2*randn(1,3); % perturb initial conditions
   [t, xsol] = ode45('lor_rhs', t, xic, [], sigma, b, r);
   x = xsol(:,1); % projected x values
   subplot(4,2,j), plot(t, x_true, 'k'), hold on
   plot(t, x, 'k', 'Linewidth', 2)
end
```

For all these realizations, the projected state fails to model the true dynamics after $t \approx 5$. Indeed, after this time, there is almost no correlation of closeness between the truth and our projection based upon noisy initial data. This illustrates the need for data assimilation. Specifically, it is hoped that occasional measurements of the data would allow for an accurate prediction of the true future state far beyond $t \approx 5$.

A simple implementation of the EKF

We consider a simple case which can be completely coded in `Matlab` in a fairly straightforward manner. In this simple example, no stochastic forcing of the differential equations will be consid-



– 8 different initial conditions lead to 8 different situations

Figure 10.5: Time evolution of $x(t)$, $y(t)$ and $z(t)$ over the time period $t \in [0, 20]$ for $\sigma = 10$, $b = 8/3$, $r = 28$ and $\mathbf{x}_0 = [5\ 5\ 5]^{\mathrm{T}}$. Here, eight realizations are shown for the perturbed initial conditions with $\sigma_2 = 1$. Note that for all simulations, after $t \approx 5$ the true dynamics (light line) differ from the dynamics with perturbed initial conditions (bold line). Predicting the dynamics beyond this time is virtually impossible given such perturbations (errors) in the initial data.

ered, so that the dynamics are exactly as specified in the ODE system – that is, $\boldsymbol{\varepsilon}_k^p = \mathbf{0}$. However, there will be both error in the measurements $\boldsymbol{\eta}_k^o$ and the initial conditions $\mathbf{g}$.

Having seen how much sensitive is the Lorenz system to initial conditions, the initial noise will greatly compromise the ability of the model to predict the future state of the system. The hope is that data assimilation will mitigate this problem at some extent, and allow for much more accurate, and longer time, predictions for the future state of the system.

In this example, the data collection points will be at simulation time points already specified by the differential equation solver. Moreover, data will be taken from all variables. Thus the mapping matrix is $\mathbf{h}(\mathbf{x}_k, \tau_k) = \mathbf{x}_k$, and we have

$$\mathbf{z}_k = \mathbf{x}_k + \boldsymbol{\varepsilon}_k^o$$

where

$$\boldsymbol{\varepsilon}_k^o = \sigma_3 \mathbf{q}(0, 1)$$

with $\mathbf{q}(0, 1)$ a Gaussian distributed random variable with mean zero and unit variance. Thus σ_3 is chosen to make the error variance either larger or smaller in the data measurements. Figure 10.6 shows the true dynamics (line) and data collected every half-unit of time with $\sigma_3 = 4$ (circles). Note that data is collected under error and does not match up perfectly with the true dynamics. However, it does follow the true dynamics fairly closely over the time period of integration.

To compute these data points in `Matlab`, the following code is used in conjunction with the code that generates the true dynamics:

```
% noisy obserations every t=0.5
tdata = t(1 : 50 : end);
n = length(tdata)
xn = randn(n,1);  yn = randn(n,1);  zn = randn(n,1);
sigma3 = 4; % error variance in data
xdata = x(1:50:end) + sigma3*xn;
ydata = y(1:50:end) + sigma3*yn;
zdata = z(1:50:end) + sigma3*zn;
```

The idea is to use these experimental points along with the noisy initial conditions shown in Figure 10.5 in order to enhance our prediction for the future state. Given that $H_k = I$ in this example, this reduces the EKF algorithm to computing

$$\mathbf{x}_k^a = \mathbf{x}_k^f + K_k(\mathbf{z}^k - \mathbf{x}_k^f)$$

with the Kalman gain matrix given by

$$K_k = P_k^f (P_k^f + R_k)^{-1}$$

and R_k is the noise covariance matrix. The error covariance of the updated state vector is given by

$$P_k^a = (I - K_k)P_k^f$$

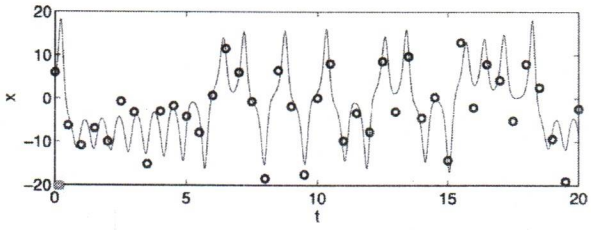


Figure 10.6: Time evolution of the variables $x(t)$, $y(t)$ and $z(t)$ over the time period $t \in [0, 20]$ for $\sigma = 10$, $b = 8/3$, $r = 28$ and $\mathbf{x}_0 = [5\ 5\ 5]^{\mathrm{T}}$. The circles represent experimental measurements at every half-unit of time of the true dynamics with Gaussian (independent) error, $\sigma_3 = 4$. The dynamics of the variables $y(t)$ and $z(t)$ are similar. Data assimilation makes use of the experimental measurements to keep the model predictions closer to the true dynamics.

Using the fact that the dynamics propagates in an error free fashion, then $P_k^f = \mathbf{J}(\mathbf{f})P_{k-1}^f\mathbf{J}(\mathbf{f})^{\top}$, where the Jacobian for the Lorenz equation can be easily computed to give

$$\mathbf{J}(\mathbf{f}) = \begin{pmatrix} \sigma & \sigma & 0 \\ r - z & -1 & -z \\ y & x & -b \end{pmatrix}$$

and the matrix P_{k-1}^f measures the error in estimating the initial state of the system at time τ_{k-1}.

Note that the Lorenz equation is a fairly trivial example to consider, and that our treatment of the data assimilation is the simplest case possible. In this example, the error variance for both the initial conditions and data measurements will both be unity so that $\sigma_2 = \sigma_3 = 1$ respectively. The Kalman gain matrix will then be given as

$$K = \frac{\sigma_2}{\sigma_2 + \sigma_3}.$$

A code for simulating this system and making adjustments based upon the data measurements is as follows:

```
x_da = [];                                          % DA solution
for j = 1 : length(tdata) - 1                       % step through every t=0.5
   tspan = 0 : 0.01 : 0.5;
   [tspan,xsol] = ode45('lor_rhs',tspan,xic,[],sigma,b,r);
   xic0 = [xsol(end,1); xsol(end,2); xsol(end,3)]   % model estimate
```

```
    xdat = [xdata(j+1); ydata(j+1); zdata(j+1)]          % data estimate
    K = sigma2 / (sigma2 + sigma3);                      % Kalman gain
    xic = xic0 + (K*[xdat - xic0])                       % adjusted state vector
    x_da = [x_da; xsol(1:end-1,:)];                      % store the data
end
x_da = [x_da; xsol(end,:)];                              % store last data time
```

In this simulation, the vector `xdat - xic0` is the innovation vector that is weighted according to the Kalman gain matrix.

Figure 10.7 shows the results of this simulation for one representative realization of the error vectors. In the top panels, the non-data assimilated computation is shown showing that the simulation solution diverges from the true solution around $t \approx 5$. The error between the true solution and the model solution is shown in the right panel. Note that the error is quite large around $t \approx 5$, thus making any prediction beyond this time fairly useless.

In the bottom panel, the data assimilated solution is demonstrated (bold line) and compared to the true dynamics (line). The data assimilated solution is nearly indistinguishable from the true dynamics. The experimental observations are shown by circles at every half-unit of time. The error between the data assimilated solution and true dynamics is shown in the right panel. Note that the error remains quite small in comparison to the direct simulation from noisy initial data.

Regardless, there is a build up of error that will eventually grow large as the data assimilated solution also diverges from the true dynamics. Ultimately, the data assimilated solution allows for significant extension of the time window where the model prediction is valid.

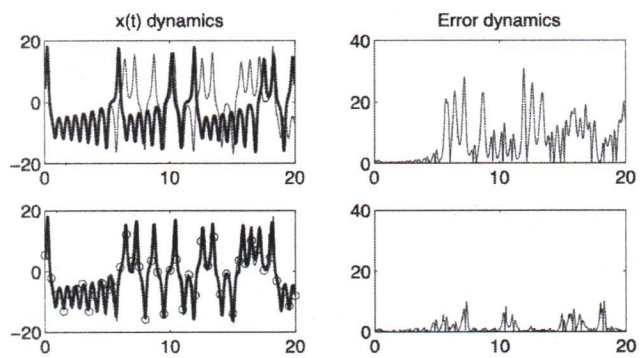


Figure 10.7: Comparison of the model dynamics using a direct numerical simulation of the noisy initial conditions (top) with the data assimilated solution with noisy initial conditions and noisy data measurements (bottom). The direct simulation (bold line) fails to predict the true dynamics (line) beyond $t \approx 5$ (top). Indeed, the error in this case grows quite large at this time (top, right). When making use of DA (bottom) and the data measurements (circles), the solution stays close to the true solution for a much longer time with much smaller error (bottom, right). Thus DA can greatly extend the time window under which the model can be useful.

Data assimilation, in general, is much more sophisticated than what has been applied here to a simple 3×3 system. The hope here was simply to illustrate the basic ideas of this tremendously powerful methodology.

10.4 Inverse UQ for time-dependent problems

So far, we have derived the Kalman filter algorithm using a *variational* argument, and shown a connection with the prior/posterior conditioning of Gaussian distributions. However, we can exploit a Bayesian framework similar to the one already introduced for parameter estimation problems, to describe filtering methods as well: this is the goal of this section, which will then prove to be useful for the sake of describing further, more general, filtering algorithms.

Inverse UQ problems can be cast in the form of a Bayesian inverse problem, as seen in the previous chapter, provided that: *(i)* the forward system is stationary, *(ii)* observations are not acquired sequentially in time, and *(iii)* parameters to be estimated does not change over time.

> Inverse UQ problem that do not verify all the three conditions above can however be formulated in the Bayesian framework, too, by means of sequential approaches like *Bayesian filtering methods*, among which the Kalman filter can be seen as a particular instance. These problems are also referred to as *non-stationary inverse problems*.

We also remark that sequential approaches can be conveniently used also in the case the parameter vector is time-independent, whenever we want to avoid to solve the whole dynamical system (that is, from the initial time on) for each evaluation of the likelihood function, for the sake of parameter estimation within a Bayesian context.

Depending on the quantities that have to be estimated, the problem can be formulated as a state estimation or a joint state/parameter estimation problem; in this latter case, *state augmentation* can be performed similarly to what we did for the Kalman Filter. In particular, we will limit our discussion to finite-dimensional models (which usually arise from space and time discretization of unsteady PDEs) and using time-discretized evolution models. Further discussions can be found in [22], [44], [47] for instance.

Let $\{\mathbf{x}_k\}_{k=0}^K$ and $\{\mathbf{z}_k\}_{k=1}^K$ denote two stochastic processes; the former is related to the quantity we are interested in, whereas the latter represents the measurement. In particular, the random vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is referred to as the *state vector*, whereas the random vector $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is referred to as the *observation*, both considered at the k-th time τ_k. From a Bayesian standpoint, the goal is to use the observations until time τ_k to get information about the state $\mathbf{x}_k$ and quantify the uncertainty related to this estimate.

To frame this problem in the Bayesian setting, we assume that $\{\mathbf{x}_k\}_{k=0}^K$ and $\{\mathbf{z}_k\}_{k=1}^K$ are an *evolution/observation model*, as follows:

1. $\{\mathbf{x}_k\}_{k=0}^K$ and $\{\mathbf{z}_k\}_{k=1}^K$ are Markov processes, that is,

$$\pi(\mathbf{x}^{k+1} \mid \mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}_k) = \pi(\mathbf{x}^{k+1} \mid \mathbf{x}_k), \quad k = 0, 1, \dots,$$
$$\pi(\mathbf{z}^k \mid \mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}_k) = \pi(\mathbf{z}^k \mid \mathbf{x}_k), \quad k = 1, 2, \dots.$$

2. $\{\mathbf{x}_k\}_{k=0}^K$ depends on the past observations only through its own history, that is,

$$\pi(\mathbf{x}^{k+1} \mid \mathbf{x}_k, \mathbf{z}^1, \dots, \mathbf{z}_k) = \pi(\mathbf{x}^{k+1} \mid \mathbf{x}_k), \quad k = 0, 1, \dots.$$

Here $\mathbf{x}_k$, $\mathbf{z}_k$ denote the realizations of the processes $\{\mathbf{x}_k\}_{k=0}^K$ and $\{\mathbf{z}_k\}_{k=1}^K$, respectively. In order to characterize such a model, we need to specify the PDF of the initial state $\mathbf{X}^0$, $\pi_{\text{prior}}(\mathbf{x}^0)$, the so-called *transition kernel* $\pi(\mathbf{x}^{k+1} \mid \mathbf{x}_k)$, $k = 0, 1, \dots$ and the conditional probability $\pi(\mathbf{z}^k \mid \mathbf{x}_k)$, $k = 1, 2, \dots$, the so-called (conditional) likelihood function.

We assume a *state evolution equation* of the form

$$\mathbf{x}_{k+1} = \mathbf{f}_{k+1}(\mathbf{x}_k, \boldsymbol{\epsilon}_{k+1}^p), \quad k = 0, 1, \dots \tag{10.20}$$

and an *observation equation* of the form

$$\mathbf{z}^k = \mathbf{h}^k(\mathbf{x}_k, \boldsymbol{\varepsilon}^o_{k+1}), \quad k = 1, 2, \dots. \tag{10.21}$$

Here $\mathbf{f}_{k+1}$ and $\mathbf{h}_k$ are known functions, whereas $\boldsymbol{\varepsilon}^p_{k+1} \in \mathbb{R}^{n_x}$ and $\boldsymbol{\varepsilon}^p_k \in \mathbb{R}^{n_z}$ denote the *state noise* and the *observation noise*, respectively.

We want to determine the conditional PDF $\pi(\mathbf{x}_k \mid \mathcal{D}_k)$ of the state at the k-th time instant given the observations $\mathcal{D}_k = (\mathbf{z}^1, \dots, \mathbf{z}_k)$ up to the same time instant; this procedure is usually referred to as *filtering problem*. By recursive application of Bayes' theorem, we have the following.

- The *time-evolution updating*, i.e., the problem of determining $\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k)$ given $\pi(\mathbf{x}_k \mid \mathcal{D}_k)$ and the transition kernel $\pi(\mathbf{x}_{k+1} \mid \mathbf{x}_k)$, provides

$$\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k) = \int_{\mathbb{R}^{n_x}} \pi(\mathbf{x}_{k+1} \mid \mathbf{x}_k)\pi(\mathbf{x}_k \mid \mathcal{D}_k)d\mathbf{x}_k. \tag{10.22}$$

- The *observation updating*, i.e., the problem of determining the posterior distribution $\pi(\mathbf{x}_{k+1} \mid \mathbf{D}^{k+1})$ of $\mathbf{x}_k \mid \mathcal{D}_k$ based on the new observation $\mathbf{z}_{k+1}$ given $\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k)$ and the likelihood function $\pi(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1})$, provides

$$\pi(\mathbf{x}_{k+1} \mid \mathbf{D}_{k+1}) = \frac{\pi(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1})\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k)}{\int_{\mathbb{R}^{n_x}} \pi(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1})\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k)d\mathbf{x}_{k+1}}, \tag{10.23}$$

  through the Bayes' formula, where $\pi(\mathbf{x}_{k+1} \mid \mathcal{D}_k)$ is considered as the prior distribution for $\mathbf{x}_{k+1}$.

The Kalman filter introduced in Section 10.2.3 is a remarkable instance of Bayesian filter method. Indeed, let us assume that the state and the observation equations are linear with additive noise processes; that is,

$$\mathbf{f}_{k+1}(\mathbf{x}_k, \boldsymbol{\varepsilon}^p_k) = A_{k+1}\mathbf{x}_k + \boldsymbol{\varepsilon}^p_k, \qquad \mathbf{h}_k = H_k\mathbf{x}_k + \boldsymbol{\varepsilon}^o_k,$$

for given matrices A_{k+1}, H_k, that the noise vectors $\boldsymbol{\varepsilon}^p_{k+1}$ and $\boldsymbol{\varepsilon}^o_k$ are mutually independent, Gaussian, with zero mean and known covariances Q_{k+1} and R_k, respectively, and that the prior PDF of $\mathbf{x}_0$ is Gaussian with mean $\mathbf{m}_0$ and covariance P_0.

Under these assumptions, the time evolution and the observation updating formulas (10.20, 10.21) involve Gaussian distributions, whose means and covariances can be updated at each step according to prediction formulas (10.16a, 10.16b) and correction formulas (10.16d, 10.16e), respectively. In particular, we have that

$$\begin{aligned} \pi(\mathbf{x}_{k+1} \mid \mathbf{x}_k) &\sim N(A_{k+1}\mathbf{x}_k, Q_k), \\ \pi(\mathbf{z}_k \mid \mathbf{x}_k) &\sim N(H_k\mathbf{x}_k, R_k). \end{aligned}$$

The Bayesian filtering equations can be evaluated in closed form, yielding the following Gaussian distributions:

$$\begin{aligned} \pi(\mathbf{x}_k \mid \mathcal{D}_{k-1}) &\sim \mathcal{N}(\mathbf{m}^f_k, P^f_k), \\ \pi(\mathbf{x}_k \mid \mathcal{D}_k) &\sim \mathcal{N}(\mathbf{m}^a_k, P^a_k), \\ \pi(\mathbf{z}_k \mid \mathcal{D}_{k-1}) &\sim \mathcal{N}(H_k\mathbf{m}^f_k, H_kP^f_kH^\top_k + R_k), \end{aligned}$$

where the means and the variances can be computed with the KF prediction step

$$\begin{aligned} \mathbf{m}^f_k &= A_k\mathbf{m}_{k-1}, \\ P^f_k &= A_kP^a_{k-1}A^\top_k + Q_{k-1} \end{aligned} \tag{10.24}$$

and the consequent correction step, setting $K_k = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$,

$$\begin{aligned} \mathbf{m}_k^a &= \mathbf{m}_k^f + K_k(\mathbf{z}_k - H_k \mathbf{m}_f^k), \\ P_k^a &= (I - K_k H_k) P_k^f. \end{aligned} \tag{10.25}$$

In other words, under the Gaussian assumption, the density is updated only through the mean and the covariance. A similar interpretation also holds for the EKF, as soon as a Gaussian approximation of the densities is considered, and the evolution of these densities is taken into account. In this respect, Bayesian filtering can be seen as a generalization of the Kalman filter and the Extended Kalman Filters introduced before.

10.5 Particle Filters: the Ensemble Kalman Filter

As already remarked in Section 10.3, when the evolution model is fully nonlinear, the EKF may perform badly: this can be explained by considering that the push-forward of the previous state estimate (which has a Gaussian distribution) by a nonlinear map is poorly approximated by a Gaussian distribution. To avoid the linearization of the evolution and the observation models, one can rely on Monte Carlo methods to simulate the distributions by random samples: this strategy yields the so-called *particle filters* (also referred to as *sequential Monte Carlo methods*), now very popular for inverse UQ problems. Particle filtering methods are fully statistical state estimation methods in the sense that they do not rely on any assumptions about the form of the target, and all inference is carried out by MC sampling.

A particle filter sequentially produces an ensemble $\{\mathbf{x}_1^{k|k}, \dots, \mathbf{x}_{N_e}^{k|k}\}$ of N_e *particles*, i.e., a random sample distributed according to the conditional distribution $\pi(\mathbf{x}_k \mid \mathcal{D}_k)$.

The *ensemble Kalman filter* (EnKF) [14] is the simplest case of particle filter exploiting the idea of approximating the means and the covariances of the current estimate involved in the Kalman filter prediction-correction strategy by a set of particles sampled from the distribution. Unlike the KF itself, we evaluate the error covariance predictions and corrections by the ensemble covariance matrices around the corresponding ensemble mean, instead of classical covariance equations (10.16b–10.16e) given in the KF algorithm. The covariance matrices of the state vector $\mathbf{x}$ need not be evolved, thus eliminating the costs associated with storing, multiplying and inverting the matrices appearing in the equations (10.16b–10.16e).

The ensemble is initialized by drawing N_e independent particles from, say, a Gaussian distribution with mean $\mathbf{m}_0$ and covariance θ_0.

- At each *prediction step*, each particle is evolved using the KF prediction step,

$$\begin{aligned} \mathbf{x}_e^{k|k-1} &= A_k \mathbf{x}_e^{k-1|k-1} + \boldsymbol{\varepsilon}_{k-1}^p && \text{if the system is linear, or} \\ \mathbf{x}_e^{k|k-1} &= \mathbf{f}_k(\mathbf{x}_e^{k-1|k-1}, \boldsymbol{\varepsilon}_{k-1}^p) && \text{if the system is nonlinear.} \end{aligned}$$

- At each *correction step*, the observation $\mathbf{z}_k$ is replicated N_e times, obtaining

$$\mathbf{d}_k^e = \mathbf{z}_k + \boldsymbol{\eta}_k^e, \quad \boldsymbol{\eta}_k^e \sim N(\mathbf{0}, R_k).$$

Then, the sample mean

$$\bar{\mathbf{x}}_e^{k|k-1} = \frac{1}{N_e} \sum_{e=1}^{N_e} \mathbf{x}_e^{k|k-1}$$

and the sample covariance

$$C_e^{k|k-1} = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} \left(\mathbf{x}_e^{k|k-1} - \bar{\mathbf{x}}_e^{k|k-1}\right)\left(\mathbf{x}_e^{k|k-1} - \bar{\mathbf{x}}_e^{k|k-1}\right)^\top$$

of the particles set $\{\mathbf{x}_e^{k|k-1}\}_{e=1}^{N_e}$ are computed. The exact Kalman gain is then approximated by

$$K_k^E = C_e^{k|k-1} H_k^\top (H_k C_e^{k|k-1} H_k^\top + R_k)^{-1}$$

and, finally, the state correction is obtained by applying the formula (10.16d) to each particle; that is,

$$\mathbf{x}_e^{k|k} = \mathbf{x}_e^{k|k-1} + K_k^E(\mathbf{d}_k^e - H_k \mathbf{x}_e^{k|k-1}).$$

Hence, the ensemble Kalman filter is a Monte Carlo implementation of the Bayesian update problem. Several alternative implementations can be found; see, e.g., [20], [23], [13].

If the inverse UQ problem also involves random inputs (as in the case of model parameters $\boldsymbol{\theta}$, which can also be time-varying), the problem of state estimation and parameter estimation simultaneously arises. Generally speaking, there is no unique optimal solution for this problem. Similarly to the *state augmentation* technique presented in Section 10.1.2, a possible way of facing this problem is to treat the unknown parameters $\boldsymbol{\theta}$ as part of the state, and use conventional filtering technique to infer the parameter and state simultaneously – this strategy goes by the name of joint estimation.