
Kalman Filter for Data Assimilation

Gerard Castro Castillo	Flàvia Ferrús
University of Barcelona	University of Barcelona
Gran Via de les Corts Catalanes, 585	Gran Via de les Corts Catalanes, 585
08007, Barcelona	08007, Barcelona
gerardc98@gmail.com	flaviaferrus@gmail.com

Abstract

Data assimilation (DA) encompasses a group of statistical methods to improve accuracy of numerical simulation using experimental values. In this paper, DA is leveraged to improve CFD predictions using an optimal method: the Kalman Filter (KF). First, the necessary framework is provided; then, a review of related work is carried out and, finally, then the KF is implemented in R and applied to a turbulent flow use-case. The experimental data is obtained using the flow generated using a fan throughout a cylindrical pipe; while the simulated data obtaining the COMSOL CFD module. The results confirm the benefits of the KF algorithm as means of DA, in view of the huge improvement of the filtered time series in comparison to the raw CFD-simulated one, in terms of bias, variance and correlation with respect to the observations.

Abbreviations: Data Assimilation - DA, State Space Models - SSM, Kalman Filter - KF, Computational Fluid Dynamics - CFD.

1 Introduction

The purpose of DA is to determine the best possible forecast state using new real-time **observations** and short-range **predictions**. Intuitively, an initial forecast is modelled using the governing equations of the physics system. With this initial distribution, the next-step variable is predicted along with its corresponding covariance matrix (to estimate the forecast error). The groundbreaking idea of the DA approach is an **additional update step** that uses new real world **observations** on an experimental system, to improve the predicted values and obtain an analysed state that approximates more accurately the distribution of the real variable, called the **true state**. The improved predicted state is used on the **subsequent** time step with new real-time observations.

In order to understand the Kalman recursion we may first need to have a general idea on how to treat time series. We address this in terms of SSM in the following subsection.

1.1 State space models

By definition, a *time series* is a set of observations Y_t , each one being recorded at specific time t , conceived as a random variable that depends on time. In this particular research, we apply the theory for *discrete time series* since the observations are made at fixed-time intervals. The *state-space representation* arises as an alternative formulation to describe a system's dynamics using two equations.

31 **Definition 1.1.** A time series $\{Y_t\}$ has a state-space representation if there exists a SSM for $\{Y_t\}$,
 32 that consists of the observation equation 1 and the state equation 2 given by:

$$Y_t = M_t X_t + W_t, \quad (1)$$

$$X_{t+1} = F_t X_t + V_t, \quad (2)$$

33 for $t \in \{0, 1, 2, \dots\}$, where: X_t is the k -dimensional **state variable vector** of the n -dimensional
 34 **observation vector** Y_t ($k \leq n$); $\{W_t\}$, $\{V_t\}$ are distributed as a white noise with 0 mean and $\{R_t\}$,
 35 $\{Q_t\}$ their **covariance matrices**; $\{M_t\}$ is a sequence of $(n \times k)$ matrices called **observation matrices**;
 36 and $\{F_t\}$ is a sequence of $(k \times k)$ matrices called **prediction matrices**. Note that $\{M_t\}$ and $\{F_t\}$
 37 are **chosen** as suitable SSM to describe the system.

38 1.2 ARIMA process

The representation on the SSM is given by the time series distribution of the observations Y_t . In our study we focus on ARIMA adjusted models, *i.e.* we consider our Y_t can be modeled through a certain ARIMA process. The ARIMA(p,s,q) process corresponds to Autorregressive Integrated Moving Average process, with degrees p, s, q . The general expression of an ARMA(p,q) time series process, Y_t is given by the Autorregressive part, which indicates the dependency on the previous terms, and the Moving Average part, indicating the modification added by a time dependent white noise $\{X_t\}_{t \geq 0} \sim WN(0, \sigma^2)$. Formally,

$$Y_t = \sum_{i=1}^p \phi_i Y_{t-i} + X_t + \sum_{i=1}^q \theta_i X_{t-i},$$

39 The integrated part stands for the times we compute $\Delta Y_t = Y_t - Y_{t-1}$ on the previous time series.
 40 More formally, considering the backshift operator $BY_t = Y_{t-1}$, then $\{Z_t\}$ is an ARIMA(p,s,q)
 41 process if $Y_t := (1 - B)^s Z_t$ is an ARMA(p,q) model as described above.

42 Naively determining the ARIMA hyper-parameters, *i.e.* p, s, q ; is generally a bad idea. Instead, to
 43 determine the suitable ARIMA distribution of a time series, normality tests are conducted, as well as
 44 the autocorrelation and partial correlation analysis. Once an intuitive idea of the time series behaviour
 45 is obtained, different ARIMA are models are adjusted to the time series and contrasted by considering
 46 the maximum likelihood and lower AIC (Akaike Information Criterion) criterion. Other information
 47 criterion can be used (*e.g.* BIC, ...) and there are several libraries which reproduces and optimizes the
 48 aforementioned process, such as `auto.arima` in R, Hyndman and Khandakar (2008) (*Flavia*).

49 1.2.1 MA(2): classical expression

50 Let us, for instance, consider an ARIMA(0, 0, 2) process (or simply *MA(2)*). If Y_t has mean 0, this
 51 can be expressed as:

$$Y_t = X_t + \theta_1 X_{t-1} + \theta_2 X_{t-2}, \quad \{X_t\}_{t \geq 0} \sim WN(0, \sigma_t^2) \quad (3)$$

52 The hyper-parameters, in this case θ_1, θ_2 are statistically estimated. Generally, the corresponding esti-
 53 mates are generally found by applying either the Least Squares method or the Maximum Likelihood
 54 method.

55 1.2.2 MA(2): SSM representation

56 Once known the time series fitted distribution (θ_1^*, θ_2^*) , we can find a SSM formulation. There are
 57 many possible representations for an ARIMA process, however following this example of a MA(2),
 58 an intuitive and possible SSM representation is given by the following following observation equation
 59 4 and state equation 5:

$$Y_t = (1, 0, 0) \begin{pmatrix} X_t \\ X_{t-1} \\ X_{t-2} \end{pmatrix} + (0) =: M X_t + W \quad (4)$$

60

$$\mathbf{X}_{t+1} := \begin{pmatrix} X_{t+1} \\ X_t \\ X_{t-1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_t \\ X_{t-1} \\ X_{t-2} \end{pmatrix} + V = F\mathbf{X}_t + V \quad (5)$$

61 Being W conceivable as white noise with mean 0 and covariance matrix $\mathbf{0}_3$; and V distributed as
 62 white noise with mean 0 and a covariance matrix Q , which should be given in terms of θ_1^*, θ_2^* . For
 63 instance, as seen later in 4, the `d1mModARMA` routine of the `d1m` library, proposes the same SSM
 64 representation with:

$$Q := \begin{pmatrix} 1 & \theta_1^* & \theta_2^* \\ \theta_1^* & \theta_1^{*2} & \theta_1^*\theta_2^* \\ \theta_2^* & \theta_1^*\theta_2^* & \theta_2^{*2} \end{pmatrix} \quad (6)$$

65 1.3 Kalman filter

66 Given the SSM (not necessarily unique for a system), there is only one optimal estimate of X_t
 67 given $\{\mathbf{Y}_t\}$ and to find it we will use the KF approach. This KF recursive process consists of two
 68 steps at every time point: a **forecast step**, providing the standard best linear predictor of the state
 69 vector, $\hat{\mathbf{X}}_t := P_{t-1}(\mathbf{X}_t)$; and an **update step**, which consists on computing the desired analysed
 70 estimate, $\mathbf{X}_{t|t} := P_t(\mathbf{X}_t)$ to incorporate the new observations. These two KF steps are described by
 71 the following theorems 1.2 and 1.3.

72 **Theorem 1.2** (Kalman Filter: forecast step). *For the state-space model given by the equations*
 73 *(1) and (2), the one-step predictors $\hat{\mathbf{X}}_t := P_{t-1}(\mathbf{X}_t)$ and their error covariance matrices $\Sigma_t =$*
 74 *$E[(\mathbf{X}_t - \hat{\mathbf{X}}_t)(\mathbf{X}_t - \hat{\mathbf{X}}_t)^T]$ are uniquely determined by the initial conditions*

$$\hat{\mathbf{X}}_1 = P_0(\mathbf{X}_1|\mathbf{Y}_0), \quad \Sigma_1 = E[(\mathbf{X}_1 - \hat{\mathbf{X}}_1)(\mathbf{X}_1 - \hat{\mathbf{X}}_1)^T],$$

75 and the recursions for $t \in \mathbb{N}^+ \setminus \{1\}$,

$$\hat{\mathbf{X}}_t = F_{t-1}\widehat{\mathbf{X}}_{t-1} + F_{t-1}K(\mathbf{Y}_{t-1} - M_{t-1}\widehat{\mathbf{X}}_{t-1}), \quad (7)$$

$$\Sigma_t = F_{t-1}(I - KM_{t-1})\Sigma_{t-1}F_{t-1}^T + Q_{t-1}, \quad (8)$$

$$K = \Sigma_{t-1}M_{t-1}^T(M_{t-1}\Sigma_{t-1}M_{t-1}^T + R_{t-1})^{-1}, \quad (9)$$

76 Note the SSM observation equation 1 and the external data \mathbf{Y}_t have not been used yet, but it is on the
 77 **update step** when, according to 1.3, the forecast distribution $\widehat{\mathbf{X}}_t$ is modified using this new data.

78 **Theorem 1.3** (Kalman Filter: update step). *The updated estimates $\mathbf{X}_{t|t} = P_t(\mathbf{X}_t)$ and their covari-*
 79 *ance matrices $\Sigma_{t|t} = E[(\mathbf{X}_t - \mathbf{X}_{t|t})(\mathbf{X}_t - \mathbf{X}_{t|t})^T]$ are determined by:*

$$\mathbf{X}_{t|t} = \hat{\mathbf{X}}_t + K(\mathbf{Y}_t - M_t\hat{\mathbf{X}}_t), \quad (10)$$

$$\Sigma_{t|t} = \Sigma_t - KM_t\Sigma_t^T, \quad (11)$$

$$K = \Sigma_t M_t^T (M_t \Sigma_t M_t^T + R_t)^{-1}. \quad (12)$$

80 Notice these recursions are remarkably useful in practice. Given the true state variable distribution
 81 X_t for a given time t (e.g. obtained from numerical computations with CFD), as well as available
 82 observations $\{\mathbf{Y}_1, \dots, \mathbf{Y}_t\}$, it is possible to obtain a better description $\mathbf{X}_{t|t}$ with estimated error (given
 83 by the covariance matrix $\Sigma_{t|t}$). In particular, this improved estimation $\mathbf{X}_{t|t}$ serves in turn as input for
 84 further modeling for the next time step $t + 1$. To illustrate this recursive process, the KF algorithm is
 85 represented in Figure 1.

86 2 Related work

87 The KF was originally presented on Kalman (1960) (*Flàvia*), which introduces an equation for time
 88 evolution of the error covariance matrix. The EnKF was then introduced by Evensen (1994) (*Flàvia*),

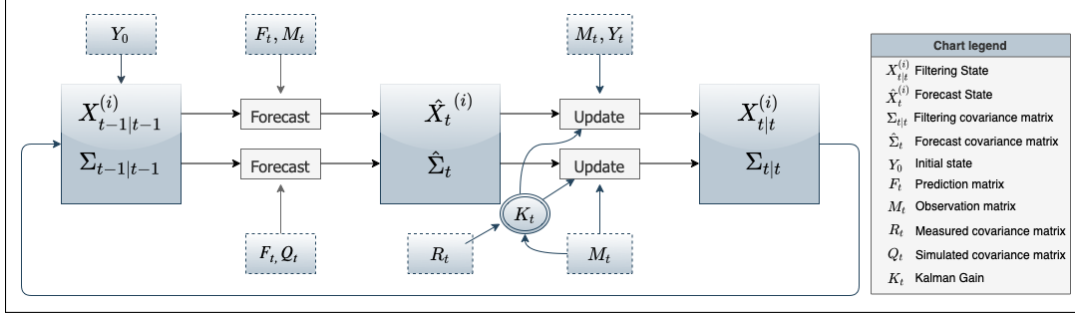


Figure 1: (KF) information flow chart diagram. Here $i \in \{1, \dots, N\}$, where N either corresponds to the sample dimension in the KF algorithm.

presenting improved approaches more orientated to solve nonlinear physical systems with larger number of observations. In particular, the EnKF is an approximate filtering method more capable and efficacious when solving high dimensional and nonlinear systems. In contrast to the standard KF, which works with the entire distribution of the state explicitly, the EnKF stores, propagates and updates an ensemble of vectors that approximates the state distribution, Katzfuss et al. (2016) (Gerard). Hence, in this EnKF case, the best estimator of the distribution is the mean of the filtered ensembles $\bar{X}_{t|t}$, analogously for the best estimator of the covariance matrix $\bar{\Sigma}_{t|t}$.

In our article, applying KF or EnKF would be essentially the same since the CFD simulations are deterministic and the state vector distribution will be a single value for each timestep (the same as EnKF with $n = 1$ ensembles). Again, the benefits of EnKF are seen in high dimensionality problems, in which it can be conceived as a dimensionality reduction technique. Both sequential methods have later been further developed, implemented and examined in a large number of published papers. Recent studies and reviews on the KF and the EnKF are given from a more theoretical point of view in Evensen et al. (2009) (Flàvia) and Brockwell and Davis (2002) (Flàvia), providing detailed information on the formulation, interpretation and implementation of the sequential DA method.

Kato and Obayashi (2011) (Gerard) used the EnKF to integrate surface pressures of a square cylinder obtained from computation and experiment, evidencing the capability of reproducing flow fields using the DA method, while they could not be replicated by only numerical simulation. Mons et al. (2016) (Flàvia), use the EnKF and other ensemble-based variational DA techniques to reconstruct a 2D unsteady flow past a cylinder combined with a compressible Navier-Stokes flow solver over different system's configurations in order to contrast the various sequential methods implemented. Deng et al. (2018) (Flàvia), is focused on the recovery of the global flow field from local quantity measurement data by optimizing the RANS model constants using the EnKF-based assimilation method, paying particular attention on the influence of different observational data on the DA performances and determining the optimal model.

3 Proposal

The proposed methodology is based on those of Deng et al. (2018) (Flàvia), Kato and Obayashi (2011) (Gerard), Mons et al. (2016) (Flàvia). On the one hand, in particular, it is modeled the turbulent flow of a 2D cylindrical configuration (or 3D but in virtue of cylindrical symmetry can be studied in 2 without loss of generality). On the other hand, the approximation of possible SSM to adjust the experimental distributions are based on ARIMA adjusted models, more specifically on a Moving Average (MA) model.

Essentially, the turbulent flow is modeled assuming the deviation from a parallel steady flow (represented theoretically by a white noise) is due to the interference effects created at the entrance region, as aforementioned, in which the wind flow goes past the inlet with a surface smaller than the pipe's cross-sectional area. The volume expansion provides a dilatation of the flow changing the velocity's

direction and affecting the velocity profile, causing a dependence on the past, given either by the MA process or by the differentiated time series describing.

3.1 Methodology

First, we define a physical model based on a wind flow in a finite cylindrical pipe, with associated initial and boundary conditions on the spatial domain using the COMSOL CFD module. The boundary conditions are given by zero velocity at the upper and lower pipe walls, atmospheric pressure at the outlet and a white noise distributed incoming velocity through the inlet (entrance region at the pipe's edge). The equations governing the system are assumed to be the Reynolds-averaged Navier Stokes (RANS). Of course, the real flow profile has some deviations from the so-called Poiseuille solution of the RANS. This is why DA will prove to be necessary.

Yet, on the one hand, the observation vector, \mathbf{Y}_t , is given by the flow velocity measured experimentally at $p = 1$ observation point of the wind flow domain. On the other hand, the state vector, \mathbf{X}_t , represents the velocity flow representation and it is assumed to be distributed as a certain ARIMA, whose exact p, s, q hyper-parameters are found such as the ones adjusting the best the CFD-simulated timeseries in that point. Specifically, the COMSOL CFD simulations are launched at a set of $k \gg p = 1$ points and, then, the \mathbf{X}_t distribution is found out using the simulated timeseries interpolated at the $p = 1$ experimental point.

Finally, the KF algorithm is implemented using:

- The SSM formulation of the assumed ARIMA distribution for \mathbf{X}_t interpolated at point p . Of course, this SSM description yields the necessary matrices: M_t, F_t, R_t and Q_t .
- The observation vector \mathbf{Y}_t at point p .

4 Experiments and results

The code¹ was implemented in R, instead of Python, due to the existence of certain packages in this language that highly simplifies several different steps. In particular, the main obtained results are highlighted below:

- The best ARIMA model adjusting the simulated time series was found using the `auto.arima` library following the AIC optimization process described at 1.2. The MA(2) process was found as the suitable one with $(\log \text{likelihood}, \text{AIC}, \text{BIC}) = (400.76, -795.53, -787.16)$.
As described at 1.2.1, a MA(2) is uniquely determined by the (θ_1, θ_2) . In our case, to ensure the simulated time series $\tilde{\mathbf{Y}}_t$ had mean 0 and (3) could be used, the optimal values were found using the simulated 'displaced' time series $\tilde{\mathbf{Y}}_t - \tilde{\mu}$ with $\tilde{\mu} = 1.0566$ being its mean over time. The found values were: $(\theta_1^*, \theta_2^*) = (0.8505, 0.2416)$.
- The SSM formulation of the previous MA(2) was obtained using the `d1mModARMA` routine from the `d1m` library. The initial distribution assumed for X_0 and its covariance matrix Σ_0 can be generally determined using the data. However, since the convergence and the achieved results were great enough, the default values were left. The author proposes as initial seeds the following ones, Petris (2010) (*Gerard*):

$$X_0 = (0, 0, 0)^T, \quad \Sigma_0 = \mathbb{I}_3 \sigma_0 \quad (13)$$

with $\sigma_0 = 10^7$. Then, in this experimental case, the best SSM parameters M_t, F_t, R_t & Q_t found were constant in time and found with values:

$$M_t = (1, 0, 0); R_t = (0); F_t = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}; Q_t = \begin{pmatrix} 1 & \theta_1^* & \theta_2^* \\ \theta_1^* & \theta_1^{*2} & \theta_1^* \theta_2^* \\ \theta_2^* & \theta_1^* \theta_2^* & \theta_2^{*2} \end{pmatrix} \quad (14)$$

Note this is the same SSM representation than 1.2.2.

¹The code can be found at the authors' repository: <https://github.com/gcastro-98/kf-data-assimilation>

166 • Finally, using the former SSM, the KF is implemented as described by the Kalman recursions
 167 given by the Theorems 1.2 and 1.3. In our case, the `d1mFilter` routine of the `d1m` library
 168 was leveraged. The filtered distribution was obtained using as the 'displaced' observations
 169 $Y_t - \mu$, which after trending it again adding μ , is plotted in Figure 2.

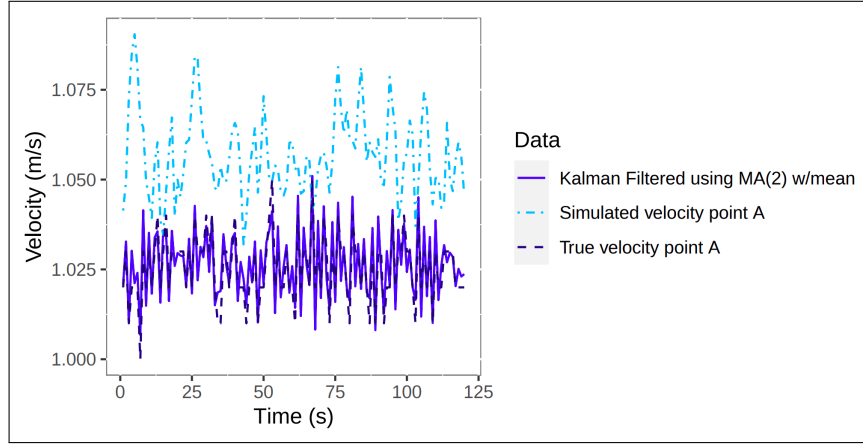


Figure 2: Observation (dashed), CFD-simulated (dot-dashed) and KF-filtered (solid) time series of wind flow velocity over time, for an interval of 2 minutes.

170 Of course, as it can be appreciated in Figure 2, the KF-filtered time series does not adjust completely
 171 and perfectly the observations, but does a much better job than the CFD simulation in terms of bias,
 172 variance and correlation.

173 5 Conclusions

174 After this brief and self-contained introduction to the Kalman Filter (KF) algorithm, as well as
 175 the presented experimental use-case modeling the wind flow velocity in a finite cylinder, it can be
 176 concluded that:

- 177 • KF constitutes a **suitable data assimilation strategy** capable of **highly improving the**
 178 **results** of complex simulations, such as CFD-based ones.

179 If the prior distribution of the wind flow profile was known, then KF would not only constitute a
 180 suitable way to integrate observations to the predictions, but a powerful tool to model in its own.
 181 Powerful but with limitations, nevertheless, because higher lead times (than the very next time step)
 182 would yield worse predictions.

183 Nevertheless, to experience the full potential of the KF algorithm, the KF would have to be imple-
 184 mented iteratively along with the CFD. This is, to apply the filtered states $X_{t|t}$ as the new input for the
 185 next time step CFD simulation. This would constitute the most natural way to integrate observations
 186 and refine CFD simulations. Of course, this would serve as short-term forecast or now-cast model,
 187 but other problems would arise: such as the estimation of the suitable distribution for the initial time
 188 steps, when not enough simulated data would have been generated... However, since with the use of
 189 warm-up rounds and other advanced strategies may be of use (and which were out of the scope of
 190 this work), the authors propose it as a new research line.

191 References

- 192 Brockwell, P.J., Davis, R.A., 2002. Introduction to time series and forecasting. Springer.
- 193 Deng, Z., He, C., Wen, X., Liu, Y., 2018. Recovering turbulent flow field from local quantity
 194 measurement: turbulence modeling using ensemble-kalman-filter-based data assimilation. Journal
 195 of Visualization 21, 1043–1063.

196 Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using
 197 monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans* 99,
 198 10143–10162.

199 Evensen, G., et al., 2009. *Data assimilation: the ensemble Kalman filter*. volume 2. Springer.

200 Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: The forecast package for
 201 r. *Journal of Statistical Software* 27, 1–22. URL: [https://www.jstatsoft.org/index.php/](https://www.jstatsoft.org/index.php/jss/article/view/v027i03)
 202 [jss/article/view/v027i03](https://www.jstatsoft.org/index.php/jss/article/view/v027i03), doi:10.18637/jss.v027.i03.

203 Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *ASME - Journal of*
 204 *Basic Engineering* 82, 35–45.

205 Kato, H., Obayashi, S., 2011. Hybrid wind tunnel based on ensemble kalman filter, in: *14th*
 206 *International Conference on Information Fusion*, IEEE. pp. 1–8.

207 Katzfuss, M., Stroud, J.R., Wikle, C.K., 2016. Understanding the ensemble kalman filter. *The*
 208 *American Statistician* 70, 350–357.

209 Mons, V., Chassaing, J.C., Gomez, T., Sagaut, P., 2016. Reconstruction of unsteady viscous flows
 210 using data assimilation schemes. *Journal of Computational Physics* 316, 255–280.

211 Petris, G., 2010. An r package for dynamic linear models. *Journal of Statistical Software*
 212 36, 1–16. URL: <https://www.jstatsoft.org/index.php/jss/article/view/v036i12>,
 213 doi:10.18637/jss.v036.i12.