
Part 1: Revisiting linear filtering, the Bayesian paradigm & optimal state estimation

From least-squares to Kalman filter

S Ganga Prasath, Vishaal Krishnan

In this two part series, we discuss the theoretical underpinnings of two important control-theoretic formalisms: (i) optimal state-estimation; (ii) optimal control, with a tutorial approach. The Kalman filter is a classical technique for optimal state-estimation for linear systems, the purpose of which is to make the best possible quantitative estimates of the underlying system state from fluctuating measurements given the knowledge of the system's behavior as well as the measurement process. Its manifold applications, ranging from molecular imaging to aerospace engineering, has given it landmark status within control theory, while its foundations are rooted in the least-squares method accessible at the undergraduate level. Given its enormous utility, it is surprising to note that these ideas have not percolated into several other fields. In this article, we walk the reader through the calculation of the filtering technique, borrowed from well-known texts in the control theory literature.

1. Introduction

Linear dynamical systems with constant coefficients capture numerous physical phenomena and are the simplest form of differential equations that often reside in the pages of undergraduate textbooks. The solution to these dynamical systems have either growing, decaying, oscillating, saturating or stationary behavior depending on the constants in the system and the initial conditions pertaining to their dynamics. Even in these simple systems, experimentally



S Ganga Prasath is currently a visiting faculty in the Department of Applied Mechanics, Indian Institute of Technology Madras, Chennai. His research interest lies at the intersection of robotics and behavior.



Vishaal Krishnan is a post-doctoral researcher in the School of Engineering and Applied Sciences, Harvard University, Cambridge. His research interests are in control theory, cognition and navigation.

Keywords

least-squares method, Kalman filter, state-estimation problem etc.



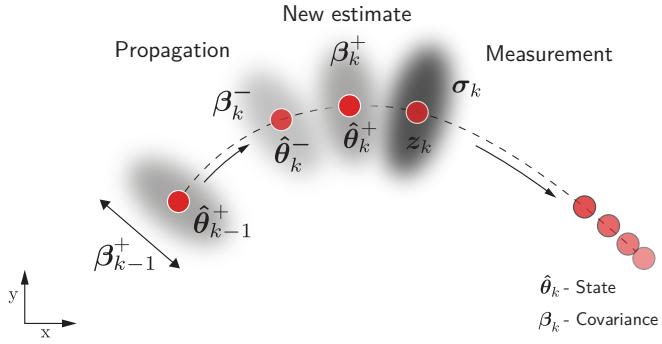
making quantitative estimates of the constants involved in the dynamics can be daunting. This is a common problem in engineering and is given the name, as one would expect, the *state-estimation* problem. The primary motivation behind this problem is to make the best possible estimate of a state-vector that captures the dynamics (or the constants) of the system. A more challenging task is to make these estimates in a fluctuating environment that adds to the uncertainty in the measurements. One of the landmark results in control theory, the field that studies such problems, is the exact solution of the best/optimal estimate (which will define soon) of the state-vector and is given by the name *Kalman filter* (denoted as KF from now on) [1].

In this pedagogical tutorial, we will look at the derivation of the Kalman filter for discrete linear dynamical system whose basic idea comes from the illustrious least-squares estimation problem. Following this, we briefly discuss the formal solution to the estimation problem for linear continuous dynamical systems i.e. the Kalman-Bucy filter [2] and highlight how the Kalman filter fits naturally in the Bayesian paradigm. We show examples throughout the article to make the approach and the notation transparent, building up to the ultimate problem of estimation of the position of a brownian particle in a harmonic trap. There is a supplementary [python](#)-code that goes with this tutorial with the numerical implementation of these algorithms. We leave the reader with a partial list of applications stampeded by Kalman/-Bucy filters and some of the open questions related to Kalman filter. The details of the calculation, the format of presentation and examples are heavily borrowed from well known texts [3, 4, 5].

2. Stochastic linear dynamical systems

Before we look at the steps leading to the derivation of the Kalman filter, we take a quick tour of stochastic differential





equations (SDEs) and their properties that will be relevant to our derivation. Stochastic dynamical systems are different from simple dynamical systems in that the dynamical variable is subjected to a source of noise (which have often singular, non-differentiable properties). The solution to these equations can be thought of as a continuous random variable capturing a process i.e. *stochastic process* [6]. Consider a stochastic dynamical system in $\theta(t) \in \mathbb{R}^d$ with white noise $\mathbf{w}(t) \in \mathbb{R}^d$ given by

$$d\theta(t) = f(t)\theta(t) dt + d\mathbf{w}(t), \quad (1)$$

$$\mathbb{E}[\mathbf{w}(t)] = 0, \quad (2)$$

$$\mathbb{E}[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{q}(t)\delta(t - \tau). \quad (3)$$

Here $f(t)$ is a known function of time, $\mathbf{w}(t)$ is the zero-mean Gaussian random variable and $\mathbf{q}(t)$ is its correlation (see Eqs. 1 for details). The evolution of the mean of the state vector $\theta(t)$, defined by $\mathbb{E}[\theta] = \mathbf{m}(t)$, can be written as $\dot{\mathbf{m}}(t) = f(t)\mathbf{m}(t)$, and the variance $\mathbb{E}[(\theta - \mathbf{m})(\theta - \mathbf{m})^T] = \beta(t)$ can be written as $\dot{\beta}(t) = f(t)\beta(t) + \beta(t)f^T(t) + \mathbf{q}(t)$. Here $\mathbb{E}[\bullet]$ denotes an ensemble average where several realizations of the same stochastic differential equation in Eq. 1 are used to obtain an average of the quantity in brackets. In 1-dimension when $f(t)$ is a constant, $f(t) = -\gamma$, such that $\gamma > 0$, the solution $\theta(t)$ is called an Ornstein-Uhlenbeck

Figure 1. Schematic showing the 3 steps involved in Kalman filtering for the case of a simple projectile motion where the state is the position & velocity of the ball (whose estimates are denoted by the vector $\hat{\theta}_k$ and the covariance by β_k). The 3 steps being: (i) estimate propagation, $\hat{\theta}_k^-, \beta_k^-$; (ii) measurement, \mathbf{z}_k (with its covariance σ_k); (iii) state-estimate and covariance estimate update $\hat{\theta}_k^+, \beta_k^+$.

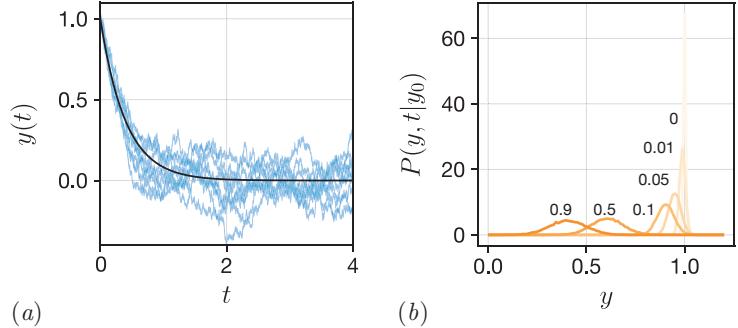


Figure 2. (a) Time evolution of $y(t)$ (in green) for different trial runs when $y_0 = 1$, $a = -1$ and $D = 10^{-2}$ in Eqs. 1. The solid line shows the mean $m(t)$ which is simply $m(t) = m_0 \exp(at)$. (b) Temporal evolution of the probability density function $P(y, t)$ shown at different time instants (indicated above the curve), computed from several runs of the same SDE starting at $y_0 = 1.0$. Refer to the [python](#)-notebook for further details.

process and when $f(t) = 0$, the solution is that of brownian motion. In Eqs. 1 we look at the evolution of a simple SDE whose evolution can be calculated analytically and provide a numerical implementation in the [python](#)-code.

When we move from continuous dynamical system to discrete system, the discrete version of the average equations can also be derived given the underlying evolution of θ_k (where k denotes the k -th step). A discrete linear dynamical system in θ_k can be written as

$$\theta_k = \Phi_{k-1} \theta_{k-1} + \Lambda_{k-1} w_{k-1}. \quad (4)$$

Here Φ_k is the propagator that evolves θ_{k-1} to θ_k and Λ_k is the amplitude of noise. The white noise has the property $\mathbb{E}[w_k] = 0, \mathbb{E}[w_k w_l^T] = q_k \delta_{k,l}$ where $\delta_{k,l}$ is the Kronecker delta function. With this the evolution of the mean then becomes $\mathbb{E}[\theta_k] = m_k = \Phi_{k-1} m_{k-1}$ and the variance can be written after little algebra as $\beta_k = \Phi_{k-1} \beta_{k-1} \Phi_{k-1}^T + \Lambda_{k-1} q_{k-1} \Lambda_{k-1}^T$ where $\beta_k = \mathbb{E}[(\theta_k - m_k)(\theta_k - m_k)^T]$ is the covariance matrix. We see that provided the initial probability distribution of the state vector i.e. $\hat{\theta}_0$ (or its mean m_0 and variance β_0), the evolution of $\hat{\theta}_k$ can be computed provided Φ_k, Λ_k are given for $k = 1 \dots n$.



Example 1: Evolution of a simple 1D SDE

Consider a simple dynamical system $\dot{y}(t) = ay(t) + b$ with $a, b \in \mathbb{R}$ being constants. This has a solution of the form $y(t) = [-b + (ay_0 + b)e^{at}]/a$ where the initial condition is $y(0) = y_0$. We see that there are 3 parameters in the system i.e. a, b and y_0 that determine the dynamics. When this dynamics is superposed with stochastic noise, the evolution can be written as a SDE of the form $dy(t) = ay(t) dt + dw(t)$ where $w(t)$ is white noise. The white noise is a random sequence characterized by $\mathbb{E}[w(t)] = 0$ and the correlation is given by $\mathbb{E}[w(t)w(\tau)] = q\delta(t - \tau)$. The magnitude of the correlation is often set to $q = 2D$ where D is the diffusion coefficient.

We know from sec. 2. that the mean evolves as $\dot{m}(t) = am(t)$ which has solution $m(t) = m_0 \exp at$ and covariance evolves as $\dot{\beta}(t) = 2a\beta(t) + 2D$ whose solution is $\beta(t) = -D/a(1 - \exp(2at))$. However, the SDE (called the Langevin equation, capturing the Ornstein-Uhlenbeck process) has a formal solution $y(t) = y_0 \exp at + \int_0^t \exp a(t-s) dw(s)$ and is valid for $a < 0$. Moreover, we can write the evolution of the $P(y, t|y_0)$ which is the probability of finding y at time t starting at y_0 (which is the corresponding Fokker-Plack equation whose details we don't go over here, refer [7, 6]) as

$$P(y, t|y_0) = \frac{1}{\sqrt{2\pi D(e^{2at} - 1)/a}} \exp \left[-\frac{(y - y_0 e^{at})^2}{2D(e^{2at} - 1)/a} \right].$$

In Fig. 2(a) we show the dynamics of $y(t)$ obtained by numerically simulating the Langevin equation and in Fig. 2(b) the probability density function $P(y, t|y_0)$ computed by simulating several trials. The `python`-notebook has further details of the implementation.

3. Least-squares estimation

We start with the derivation of the least-squares estimator (shortened as LSE hereafter) using which we can fit a model to data by minimizing the error between the model and the data. Least-squares fit/estimator has a long history and dates back to the days of the then young Karl Friedrich Gauss in 1795. Gauss used the least-squares estimator that he had developed to calculate the parameters necessary to describe the motion of the comet Ceres from astronomical measurements. Using this he managed to predict the time when the comet would reappear from behind



Least-squares estimation dates back to the days of Gauss in 1795. He used it to win the competition set by the Italian astronomer Giuseppe Piazzi by predicting the time when a comet would reappear from behind the sun's glare.

the sun's glare and win the competition set by the Italian astronomer Giuseppe Piazzi. Gauss being the inventor of the least-squares method, however, was not apparent during his days and it involves a fascinating anecdote between him and Legendre. We direct the interested readers to [4] for a detailed summary. A least-squares estimator provides the best possible fit to the data assuming that we are aware of the model that generated the data. In the example of Gauss, he already had the model of the comet's motion from initial measurements of its trajectory. The notion of best possible fit implies that the error between the data generated by the model and that of the measured data is minimized. In general, we know that models of physical systems involve physical parameters and that the primary motivation behind this estimation process is to come up with values or estimates of these parameters. To keep this simple, we look at an example where the model is a linear function of the parameters. In general when the model is a non-linear function of the parameters, more complex properties of the fit arise (see [8] for further details) and details of such fits are well outside the domain of interest of this tutorial.

In order to state things more concretely in mathematical form, let us consider a dynamical 1-dimensional trajectory of a particle $y(t; \theta) \in \mathbb{R}$ which is a function of time t and set of parameters $\theta_i \in \mathbb{R}$ where $i = 1 \dots k$, denoted using vector $\theta \in \mathbb{R}^k$ that are the physical constants and/or the initial conditions (depending on the details of the system we are interested). As the trajectory is a linear function of the parameters, we can write $y(t) = f(t)^T \theta$ where $f(t)$ is often called the feature vector or template vector containing the functional form of basis functions/model to which we would like to fit the data. We make discrete measurements of the particle trajectory at time instants t_i where $i = 1 \dots n$ to obtain a vector $m \in \mathbb{R}^n$.



Example 2: Least square estimation of constants for a one-dimensional trajectory

Consider a time-varying trajectory of a particle in 1D given by $y(t) = at + bt^{3/2} + ct^2 + dt^4$. From this we can write $f(t) = [t \ t^{3/2} \ t^2 \ t^4]$ and the state $\theta = [a, b, c, d]$. From this we can immediately compute the $\alpha = f(t_i)$ matrix of dimensions $n \times 4$ after n -measurements at $t = \Delta t[1, 2, \dots, n]$. We can compute the measurement $m \in \mathbb{R}^{n \times 1}$ from data collected at discrete time-instants t_j if we know the distribution of the noise that corrupted the data.

Assuming that $n \sim N(0, 1)$ where $N(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ , we now have the model matrix α and the measurement m which is all we require to compute the estimation of θ given by $\hat{\theta} = (\alpha^T \alpha)^{-1} \alpha^T m$. We show in Fig. 3(a) the comparison between the measurements and the solution trajectory obtained by using the estimated constants from least squares method. In Fig. 3(b) we show that the error in computed constants (defined by $\|\theta - \hat{\theta}\|$) goes to zero with increase in number of measurements n . The reader is strongly encouraged to look at the [python](#)-code that goes with the article for details.

The first assumption we make about the measurement process is that the measurement is a linear function of the state and is corrupted by a noise n (due to environmental fluctuations or sensor noise): $m = \alpha\theta + n$ where $\alpha = f(t_i) \in \mathbb{R}^{n \times k}$. Just to be clear, we now have access to α (because we know the functional form of the model $f(t)$) and m (which is the actual measurement) but not θ . Our aim is to make the best estimate of θ which we denote by $\hat{\theta}$ given α, m .

Residual error minimization

Although our motivation is to minimize the error $\|\theta - \hat{\theta}\|$, we do not have access to θ . So our best estimate has to be based on m and \hat{y} where $\hat{y} = \alpha\hat{\theta}$. We can define the error $\epsilon = (m - \hat{y})$ (also called the *measurement residual*) and from



this the quadratic cost function is given by

$$\begin{aligned}\mathcal{J}(\hat{\theta}) &= \frac{1}{2}\epsilon^T\epsilon = \frac{1}{2}(m - \hat{y})^T(m - \hat{y}) \\ &= \frac{1}{2}(m^T m - m^T \alpha \hat{\theta} - \hat{\theta}^T \alpha^T m + \hat{\theta}^T \alpha^T \alpha \hat{\theta}),\end{aligned}$$

which can be minimized by setting

$$\frac{\delta \mathcal{J}(\hat{\theta})}{\delta \hat{\theta}} = (\alpha^T \alpha \hat{\theta} - \alpha^T m)^T = 0.$$

Minimizing the measurement error, the solution to the least square estimation problem to compute θ is $\hat{\theta} = (\alpha^T \sigma^{-1} \alpha)^{-1} \alpha^T \sigma^{-1} m$ where α is the model, σ is a weight matrix (often the covariance of the error).

We see that the best estimate of θ is $\hat{\theta} = (\alpha^T \alpha)^{-1} \alpha^T m$. We can easily show that $\delta^2 \mathcal{J}(\hat{\theta}) / \delta \hat{\theta}^2 = \alpha^T \alpha$ which is positive definite providing the sufficiency condition that the solution obtained is a minimum. When you know more about the noise in the measurement process, one can also obtain an appropriately weighted estimate $\hat{\theta}$ with a weight matrix σ to get $\hat{\theta} = (\alpha^T \sigma^{-1} \alpha)^{-1} \alpha^T \sigma^{-1} m$ where σ is the co-variance of the noise. In Egs. 2 we look at an example problem where we deploy the fit derived here and Fig. 3 shows a sample fit.

Direct least-squares

If in addition to the description of the measurement process $m = \alpha \theta + n$, when we have prior knowledge of the covariance of the parameters θ , we can seek to directly minimize the squared error $\|\hat{\theta} - \theta\|^2$ of the estimate $\hat{\theta}$ on an average. Just as in the case of (measurement) residual error minimization, we seek an estimate $\hat{\theta}$ that is a linear function of the measurement m , i.e., $\hat{\theta} = \kappa m$ (where $\kappa \in \mathbb{R}^{k \times n}$ is called the *gain* and will play an important role later) by solving the following minimization problem

$$\min_{\kappa} \mathbb{E} [\|\hat{\theta} - \theta\|^2].$$

Here $\mathbb{E}[\bullet]$ denotes average over several trials of measurement of the same particle dynamics, just as described in sec. 2. Substituting the model $\hat{\theta} = \kappa m$ above, we obtain

$$\min_{\kappa} \mathbb{E} [\|\hat{\theta} - \theta\|^2] = \min_{\kappa} \mathbb{E} [\|\kappa m - \theta\|^2] = \min_{\kappa} \mathbb{E} [\|\kappa(\alpha \theta + n) - \theta\|^2].$$



Expanding the above equation, we get

$$\min_{\kappa} \mathbb{E} \left[\theta^\top (\kappa \alpha - \mathbb{I})^\top (\kappa \alpha - \mathbb{I}) \theta + 2\theta^\top (\kappa \alpha - \mathbb{I})^\top \kappa n + n^\top \kappa^\top \kappa n \right].$$

Note that the individual terms in the cost above are scalar inner products which can be rewritten as the trace of the corresponding outer products as follows

$$\min_{\kappa} \mathbb{E} \left[\text{Tr}\{(\kappa \alpha - \mathbb{I})\theta \theta^\top (\kappa \alpha - \mathbb{I})^\top + 2\kappa n \theta^\top (\kappa \alpha - \mathbb{I})^\top + \kappa n n^\top \kappa^\top\} \right].$$

From the commutativity of the expectation and trace operations and the fact that θ and n are independent random variables, the above reduces to

$$\begin{aligned} & \min_{\kappa} \text{Tr} \{ (\kappa \alpha - \mathbb{I}) \mathbb{E} [\theta \theta^\top] (\kappa \alpha - \mathbb{I})^\top + \kappa \mathbb{E} [nn^\top] \kappa^\top \} \\ &= \mathbb{E}[\theta \theta^\top] + \min_{\kappa} \text{Tr}\{\kappa (\alpha \mathbb{E}[\theta \theta^\top] \alpha^\top + \mathbb{E}[nn^\top]) \kappa^\top - 2\kappa \alpha \mathbb{E}[\theta \theta^\top]\}. \end{aligned}$$

Taking the covariances of θ and n to be $\beta = \mathbb{E}[\theta \theta^\top] \in \mathbb{R}^{k \times k}$ and $\sigma = \mathbb{E}[nn^\top] \in \mathbb{R}^{n \times n}$ respectively, and setting the first variation with respect to κ of the cost above to zero, we get the optimal κ (denoted by κ^*) as

$$\kappa^* = \beta \alpha^\top (\alpha \beta \alpha^\top + \sigma)^{-1},$$

and the optimal estimate then becomes

$$\hat{\theta} = \beta \alpha^\top (\alpha \beta \alpha^\top + \sigma)^{-1} m.$$

Minimizing the mean-squared error on the parameter estimate, the solution to the direct least-squares estimation problem to compute θ is $\hat{\theta} = \beta \alpha^\top (\alpha \beta \alpha^\top + \sigma)^{-1} m$ where α is the model, β is the covariance of the true parameter θ and σ is the covariance of the noise.

Recursive least-square estimator

The next step towards the derivation of the Kalman filter is to make estimate of the state $\hat{\theta}$ as and when the data is collected instead of assuming that the measurement is a fixed vector. The estimator we derive in this subsection is called the *recursive least-square estimator* (shortened



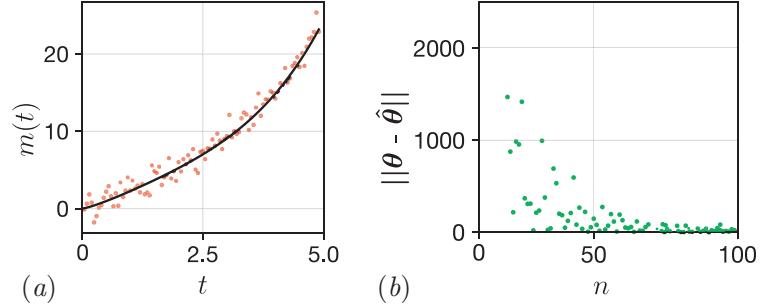


Figure 3. (a) Comparison between measurements $m(t_i)$ and the trajectory $\hat{y}(t)$ (solid black line) obtained by least square estimation. (b) L_2 -norm of error in computed constants defined as $\|\theta - \hat{\theta}\|$ as a function of the number of measurements n .

as RLSE). Let us consider the situation when we have collected n_1 measurements of the same dynamical process as earlier, given by $\mathbf{m}_1 \in \mathbb{R}^{n_1}$ and make an estimate $\hat{\theta}_1$. We know from our current algorithm that, this is given by $\hat{\theta}_1 = (\boldsymbol{\alpha}_1^T \boldsymbol{\sigma}_1^{-1} \boldsymbol{\alpha}_1)^{-1} \boldsymbol{\alpha}_1^T \boldsymbol{\sigma}_1^{-1} \mathbf{m}_1$ where $\boldsymbol{\alpha}_1 = f(t_i) \forall t_i, i = 1 \dots n_1 \in \mathbb{R}^{n_1 \times k}$. We would like to update $\hat{\theta}_1 \in \mathbb{R}^k$ when more measurements are made. Now new measurements $\mathbf{m}_2 \in \mathbb{R}^{n_2}$ are made such that $\boldsymbol{\alpha}_2 = f(t_i) \forall t_i, i = n_1 + 1 \dots n_1 + n_2 \in \mathbb{R}^{(n_1+n_2) \times k}$. With this new measurement, the new estimate that minimizes over entire error using both $\mathbf{m}_1, \mathbf{m}_2$ is given by $\hat{\theta}_2$. Assuming that we know the measurement error covariance for both the steps, $\sigma_j = \mathbb{E}[\mathbf{n}_j \mathbf{n}_j^T]$, we can write the total error as

$$\mathcal{J}(\mathbf{m}_1, \mathbf{m}_2) = [(\mathbf{m}_1 - \boldsymbol{\alpha}_1 \hat{\theta}_2)^T (\mathbf{m}_2 - \boldsymbol{\alpha}_2 \hat{\theta}_2)^T] \begin{bmatrix} \boldsymbol{\sigma}_1^{-1} & 0 \\ 0 & \boldsymbol{\sigma}_2^{-1} \end{bmatrix} [(\mathbf{m}_1 - \boldsymbol{\alpha}_1 \hat{\theta}_2) (\mathbf{m}_2 - \boldsymbol{\alpha}_2 \hat{\theta}_2)]. \quad (5)$$

Following a similar procedure as earlier, we can minimize the above function by setting $\delta \mathcal{J}(\hat{\theta}_2)/\delta \hat{\theta}_2 = 0$ to get

$$\hat{\theta}_2 = (\boldsymbol{\beta}_1^{-1} + \boldsymbol{\alpha}_2^T \boldsymbol{\sigma}_2^{-1} \boldsymbol{\alpha}_2)^{-1} (\boldsymbol{\alpha}_1^T \boldsymbol{\sigma}_1^{-1} \mathbf{m}_1 + \boldsymbol{\alpha}_2^T \boldsymbol{\sigma}_2^{-1} \mathbf{m}_2), \quad (6)$$

where we have used the notation $\boldsymbol{\beta}_1 = (\boldsymbol{\alpha}_1^T \boldsymbol{\sigma}_1^{-1} \boldsymbol{\alpha}_1)^{-1}$, which is the state residual covariance matrix, $\boldsymbol{\beta}_1 = \mathbb{E}[(\theta_1 - \hat{\theta}_1)(\theta_1 - \hat{\theta}_1)^T]$. We can use matrix inversion lemma to expand Eq. 6

to get:

$$\begin{aligned}\hat{\theta}_2 &= [\beta_1 - \beta_1 \alpha_2^T (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1} \alpha_2 \beta_1] (\alpha_1^T \sigma_1^{-1} m_1 + \alpha_2^T \sigma_2^{-1} m_2) \\ &= \hat{\theta}_1 - \beta_1 \alpha_2^T (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1} \alpha_2 \hat{\theta}_1 \\ &\quad + \beta_1 \alpha_2^T (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1} (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T) \sigma_2^{-1} m_2 \\ &\quad - \beta_1 \alpha_2^T (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1} \alpha_2 \beta_1 \alpha_2^T \sigma_2^{-1} m_2,\end{aligned}$$

where we have substituted $\hat{\theta}_1 = \beta_1 \alpha_1^T \sigma_1^{-1} m_1$. After using the identity $\mathbb{I} = (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1} (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)$, we get a compact form for the above equations as

$$\begin{aligned}\hat{\theta}_2 &= \hat{\theta}_1 + \kappa_2 (m_2 - \alpha_2 \hat{\theta}_1), \\ \kappa_2 &= \beta_1 \alpha_2^T (\sigma_2 + \alpha_2 \beta_1 \alpha_2^T)^{-1}.\end{aligned}$$

We can immediately see the striking similarity between the gain κ_2 here and κ^* derived using the direct least-squares method.

Algorithm 1: Recursive least-squares estimator

Known: Measurement m_j , Model $\alpha_j = f(t_j)$, Estimation error β_j for $j = 1, 2, \dots, n$;

Unknown: State θ_j ; for $j > 0$;

Initialize $\hat{\theta}_0, \beta_0, \sigma_1$;

foreach step of estimation, j **do**

$$\begin{aligned}\hat{\theta}_j &= \hat{\theta}_{j-1} + \kappa_j (m_j - \alpha_j \hat{\theta}_{j-1}) \\ \kappa_j &= \beta_{j-1} \alpha_j^T (\sigma_j + \alpha_j \beta_{j-1} \alpha_j^T)^{-1} \\ \beta_j &= (\beta_{j-1}^{-1} + \alpha_j^T \sigma_j^{-1} \alpha_j)^{-1}.\end{aligned}$$

end

Using this relation we can write the evolution of the estimate, $\hat{\theta}_j$ as new measurements, m_j are made as a *recursive least-squares estimator*. This can be written as

$$\hat{\theta}_j = \hat{\theta}_{j-1} + \kappa_j (m_j - \alpha_j \hat{\theta}_{j-1}) \quad (7)$$

$$\kappa_j = \beta_{j-1} \alpha_j^T (\sigma_j + \alpha_j \beta_{j-1} \alpha_j^T)^{-1} \quad (8)$$

$$\beta_j = (\beta_{j-1}^{-1} + \alpha_j^T \sigma_j^{-1} \alpha_j)^{-1}. \quad (9)$$



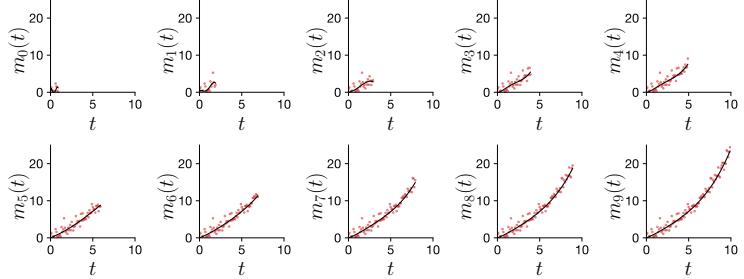


Figure 4. Evolution of measurements $m_j(t_i)$ and the solution obtained using the recursive least square estimator as new data is obtained for the model described in Eqs. 2. Different plots correspond to increase in acquired data where estimation is done for every 10 new data points for $j = 0, 1, \dots, 9$.

In Fig. 4 we show the results from the implementation of this technique to determine the constants, $\hat{\theta}_k$ recursively for the simple 1D dynamics described in Eqs. 2.

Before proceeding to the derivation of the Kalman filter, we take a brief look at the structural conditions under which the system dynamics and measurement model permit a solution to the estimation problem. To this end, we consider the idealized case of the measurement model without noise ($n = 0$). Given measurements m_1, \dots, m_ℓ (with $m_j = \alpha_j \theta$ for $j = 1, \dots, \ell$), we can write

$$\begin{pmatrix} m_1 \\ \vdots \\ m_\ell \end{pmatrix} = \Psi \theta,$$

where $\Psi = (\alpha_1^\top, \dots, \alpha_\ell^\top)^\top$ is called the *observability matrix*. The estimation problem in this noiseless case requires the invertibility of Ψ for some ℓ , in order to be fully soluble. In retrospect, this structural condition is necessary even for the estimators developed earlier, by residual error and direct least-squares minimization. In the absence of such a condition, the estimate $\hat{\theta}$ only recovers the true parameter θ up to a projection onto a lower dimensional subspace and parts of the parameter θ will remain inestimable from the measurements m_k . Intuitively, the matrix Ψ determines the observable space for underlying parameter θ , with the various snapshots of θ encapsulated in the measurement vector (m_1, \dots, m_ℓ) . We then see that the invertibility of Ψ affords



a “full view” of the parameter θ from this observable space. We will derive the corresponding structural condition underlying the Kalman filter, where the model of the system dynamics is explicitly accounted for in the definition of the matrix Ψ , in the next section.

4. Kalman filter as an extension of RLSE

We are now equipped with the required tools to derive the Kalman filter. The purpose of the filter is to estimate the true state of the system, θ_k as $\hat{\theta}_k$ such that the variance/spread of the estimate error distribution or the estimation error is minimum. Minimizing the estimation error ensures that the likelihood of estimating θ_k is maximized in the process (which is true in general for unimodal distributions of error). The KF setup follows a multi-step ‘predictor-corrector’ procedure (see Fig. 1 for a schematic) where we use the model/propagator to update the previous state and covariance estimates, then compute the gain that is required to correct the estimates given the knowledge of the error. Then use this gain to update the predicted estimates with new measurements. The derivation in this section closely follows chapter 4 of [3].

Let us start by phrasing the KF setup in mathematical terms. Just as in the sec. 2., we consider a discrete dynamical system whose true state evolves as

$$\theta_k = \Phi_{k-1}\theta_{k-1} + \Lambda_{k-1}\mathbf{w}_{k-1}, \quad (10)$$

$$\mathbb{E}[\mathbf{w}_k] = 0, \quad (11)$$

$$\mathbb{E}[\mathbf{w}_k \mathbf{w}_l^T] = \mathbf{q}_k \delta_{k,l}. \quad (12)$$

Here \mathbf{w}_k is discrete delta-correlated noise with correlation amplitude \mathbf{q}_k and Φ_k, Λ_k are known quantities that describe the system. We make observations/measurements \mathbf{z}_k at time k of the true state θ_k according to

$$\mathbf{z}_k = \alpha_k \theta_k + \mathbf{n}_k.$$



where we assume that the model of measurement, α_k is given and the statistics of the noise n_k follow

$$\mathbb{E}[n_k] = 0, \quad (13)$$

$$\mathbb{E}[n_k n_l^T] = \sigma_k \delta_{k,l}. \quad (14)$$

This setup also closely resembles that of RLSEs we derived in sec. 3. So far we have assumed that the model quantities Φ_{k-1} , error amplitude Λ_{k-1} are known as well as the measurement model α_k and the variance in error and noise q_k, σ_k are given. The unknowns are the true state θ_k (whose value we estimate using $\hat{\theta}_k$) and the covariance estimate, β_k . The Kalman filter has 3 steps: (i) state and covariance estimate propagation, (ii) filter gain computation, (iii) state and covariance estimate update.

Kalman filter estimates the value of the state θ_k such that the variance/spread of the estimate error, $(z - \alpha_k \hat{\theta}_k)$ distribution or the estimation error is minimum. The setup involves multiple steps: a predictor step and a corrector step with the appropriate gain to make the right estimates.

Before proceeding to derive the above three steps, we briefly comment on the necessary structural condition for solvability of the estimation problem. As introduced in the previous section, this involves the observability matrix Ψ , which is now redefined to account for the system dynamics (10). In the deterministic case, i.e. $w_k = 0, n_k = 0$, the state sequence $\theta_0, \theta_1, \dots$ is completely determined from the initial state θ_0 and the state transition matrices Φ_k . Consequently, the structural condition we seek involves the state transition matrices Φ_k and the measurement matrices α_k , and requires invertibility of the following *observability matrix* for some ℓ

$$\Psi = \begin{pmatrix} \alpha_0 \\ \alpha_1 \Phi_0 \\ \alpha_2 \Phi_1 \Phi_0 \\ \vdots \\ \alpha_\ell \Phi_{\ell-1} \Phi_{\ell-2} \dots \Phi_0 \end{pmatrix}.$$

Note that for a constant α , the rank of Ψ saturates at $\ell = d$, the system dimension. Just as before, in the absence of invertibility of the observability matrix, parts of the system state will remain inestimable from measurements.



Estimates propagation step

The state and covariance estimate extrapolation follows exactly the dynamics of the mean and covariance of an SDE discussed in sec. 2. This is given by

$$\hat{\theta}_k^- = \Phi_{k-1} \hat{\theta}_{k-1}^+, \quad (15)$$

$$\beta_k^- = \Phi_{k-1} \beta_{k-1}^+ \Phi_{k-1}^T + \Lambda_{k-1} \mathbf{q}_{k-1} \Lambda_{k-1}^T. \quad (16)$$

Here we use $(-)$ to denote the propagation step without considering the effects of the measurement and $(+)$ to denote the update to the estimate after considering measurement.

Gain computation step

The next step in the KF setup is to compute the gain, known in literature as the *Kalman gain*. The gain, denoted by κ_k , follows analogous to the RLSE framework in Eq. 8 and has the functional form,

$$\kappa_k = \beta_k^- \alpha_k^T [\alpha_k \beta_k^- \alpha_k^T + \sigma_k]^{-1}.$$

The gain serves the purpose of a multiplier that corrects the predictions of the estimation from previously available data. As we see, it depends both on the covariance estimation from previous step as well as the measurement error covariance.



Algorithm 2: Kalman filter

Known: Dynamics model, (Φ_j, Λ_j) ;
 Noise covariance, \mathbf{q}_j ;
 Measurement model, α_j ; Measurement, \mathbf{z}_j ;
 Error covariance, σ_j ;
Unknown: Mean estimation of state, $\hat{\theta}_j$;
 Covariance estimation, β_k for $j > 0$;
 Initialize $\hat{\theta}_0, \beta_0$;
foreach step of estimation, j **do**

- Estimates propagation:
$$\hat{\theta}_j^- = \Phi_{j-1} \hat{\theta}_{j-1}^+$$

$$\beta_j^- = \Phi_{j-1} \beta_{j-1}^+ \Phi_{j-1}^T + \Lambda_{j-1} \mathbf{q}_{j-1} \Lambda_{j-1}^T.$$
- Gain computation:
$$\kappa_j = \beta_j^- \alpha_j^T [\alpha_j \beta_j^- \alpha_j^T + \sigma_j]^{-1}.$$
- Estimates update:
$$\hat{\theta}_j^+ = \hat{\theta}_j^- + \kappa_j [\mathbf{z}_j - \alpha_j \hat{\theta}_j^-],$$

$$\beta_j^+ = (\mathbb{I} - \kappa_j \alpha_j) \beta_j^-.$$

end

Estimates update step

Given the values of the gain, we can update the state and covariance estimates just as in Eqs. 7, 9 to get

$$\hat{\theta}_k^+ = \hat{\theta}_k^- + \kappa_k [\mathbf{z}_k - \alpha_k \hat{\theta}_k^-], \quad (17)$$

$$\beta_k^+ = [(\beta_k^-)^{-1} + \alpha_k^T \sigma_k^{-1} \alpha_k]^{-1}. \quad (18)$$

Eq. 18 can be rewritten after some algebra (primarily using Woodbury matrix identity) as $\beta_k^+ = (\mathbb{I} - \kappa_k \alpha_k) \beta_k^-$, which is the standard form seen in literature. We see that the Kalman gain multiplies the error between ‘prediction’ and measurement, loosely acting as a lagrange multiplier. The covariance update on the other hand, uses the details of the measurement statistics to update. In the simple RLSE setting, the updates on the state estimates were based on the

minimizing the error defined as the difference between the measurement and the prediction weighted appropriately by the covariance matrix. In the KF setting, we see that the update uses a similar update using the Kalman gain thus ensuring that the estimate error is minimum.

To simplify the idea behind this entire procedure, we know from (R)LSE that with a given model of the dynamics, we can fit measurements to the model to estimate constants/state vector. However, in order to make better estimates it is useful to weight the measurements with the appropriate variance of the dynamics. Thus good variance of the dynamics provides good estimates. Subsequently, we also know that the measurement process introduces error in estimation. In order to account for this, the variance of the dynamics has to be appropriately corrected. The KF setup does exactly this. The propagation step keeps track of both the mean and variance using the model of the dynamics. Second, the gain computation provides the appropriate correction to the simple propagation accounting for the stochasticity in dynamics and error in the measurement. Lastly the estimates are updated, just as in RLSE, utilizing the new measurement.

Kalman filter has 3 steps: (i) estimate propagation, (ii) gain computation, (iii) estimate update.

Estimate propagation projects forward the estimates from previous time-step while the gain computation provides the right *Kalman gain* required to correct this estimate. The update step uses this gain to calculate the new estimate with new measurements.



Example 3: Estimating position of a Brownian particle in a harmonic potential

Evolution of a Brownian particle in a harmonic potential is a classic problem in non-equilibrium statistical mechanics. The dynamics of the particle is given by

$$\begin{aligned} dx &= v dt, \\ dv &= -\gamma v dt - \omega^2 x + \sqrt{\frac{2\gamma k_B T}{m}} dw(t), \end{aligned}$$

where $x(t), v(t)$ are the position and velocity of the particle at time t , γ is the viscous damping coefficient, $\omega = \sqrt{k/m}$ is the frequency of oscillation due to a harmonic potential $V(x) = kx^2/2$, k_B is the Boltzmann constant, T is the temperature and $w(t)$ is the Wiener process with $\mathbb{E}[w(t)] = 0$, $\mathbb{E}[w(t)w(t')] = \delta(t-t')$. We can immediately see that in the limit when inertia is not relevant, the above equation reduces to $dx = -\omega^2 x / \gamma dt + \sqrt{2k_B T / \gamma m} dw(t)$. This immediately looks similar to the dynamics we saw in Eqs. 1. Our aim now is to use Kalman-Bucy filter to estimate $\hat{x}(t)$ given we know $f(t) = -\omega^2 / \gamma$, $q(t) = 2k_B T / \gamma m$. We simulate the dynamics of this brownian particle and estimate its position as shown in Fig. 5.

Kalman filter from direct least-squares

Just as we saw with LSE, we can derive the Kalman-gain using direct least-squares. The structure of the Kalman filter for the state-estimate can be rewritten from Eq. 17 as

$$\hat{\theta}_k = \Phi_{k-1} \hat{\theta}_{k-1} + \kappa_k [z_k - \alpha_k \Phi_{k-1} \hat{\theta}_{k-1}].$$

The state-estimation error, $e_k = (\theta_k - \hat{\theta}_k)$ can be written as

$$\begin{aligned} e_k &= \Phi_{k-1} [\theta_{k-1} - \hat{\theta}_{k-1}] - \kappa_k [z_k - \alpha_k \Phi_{k-1} \hat{\theta}_{k-1}] \\ &= \Phi_{k-1} e_{k-1} - \kappa_k \alpha_k \Phi_{k-1} e_{k-1} + \kappa_k n_k + \Lambda_{k-1} w_{k-1} \\ &= (\mathbb{I} - \kappa_k \alpha_k) \Phi_{k-1} e_{k-1} + \kappa_k n_k + \Lambda_{k-1} w_{k-1}. \end{aligned}$$

The covariance update, $\beta_k = \mathbb{E} [e_k e_k^\top]$ then becomes

$$\beta_k = (\mathbb{I} - \kappa_k \alpha_k) \Phi_{k-1} \beta_{k-1} \Phi_{k-1}^\top (\mathbb{I} - \kappa_k \alpha_k)^\top + \kappa_k \sigma_k \kappa_k^\top + \Lambda_{k-1} q_{k-1} \Lambda_{k-1}^\top.$$

The Kalman gain κ_k is obtained as the minimizer of $\text{Tr}(\beta_k)$, and satisfies $(\partial \text{Tr}(\beta_k) / \partial \kappa_k)|_{\kappa_k^*} = 0$. The functional form of κ_k



obtained by following the minimization procedure is again

$$\kappa_k = \beta_{k-1} \alpha_k^T [\alpha_k \beta_{k-1} \alpha_k^T + \sigma_k]^{-1}.$$

Kalman-Bucy filter

The continuous time version of the KF setup is known as the Kalman-Bucy filter. In this setting the true state evolves as

$$\dot{\theta}(t) = f(t)\theta(t) + \mathbf{w}(t), \quad (19)$$

$$\mathbb{E}[\mathbf{w}(t)] = 0, \quad (20)$$

$$\mathbb{E}[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{q}(t)\delta(t-\tau). \quad (21)$$

Just as in the discrete version, the measurement model is given by $\mathbf{z}(t) = \alpha(t)\theta(t) + \mathbf{n}(t)$, where the noise statistics again follow $\mathbb{E}[\mathbf{n}(t)] = 0, \mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \sigma(t)\delta(t-\tau)$. Following a similar procedure as earlier, we arrive at the evolution of the state and variance estimates as

$$\dot{\hat{\theta}}(t) = f(t)\hat{\theta}(t) + \kappa(t)\{\mathbf{z}(t) - \alpha(t)\hat{\theta}(t)\}, \quad (22)$$

$$\dot{\beta}(t) = f(t)\beta(t) + \beta(t)f^T(t) + \mathbf{q}(t) - \kappa(t)\sigma(t)\kappa^T(t), \quad (23)$$

$$\kappa(t) = \beta(t)\alpha^T(t)\sigma^{-1}(t). \quad (24)$$

We see that the evolution equations for the state estimate and the covariance follow closely the discrete version. We illustrate the implementation of this technique through the example of estimating the position of a brownian particle in a harmonic potential, detailed in Egs. 3.

5. Kalman filtering as recursive Bayesian estimation

The problem of estimation is fundamentally an inverse problem of obtaining the underlying system state from measurements, whereas the known dynamics and measurement models specify the forward relationship from the system state to the measurements. In principle, the problem of obtaining the optimal state estimate at any given time can be



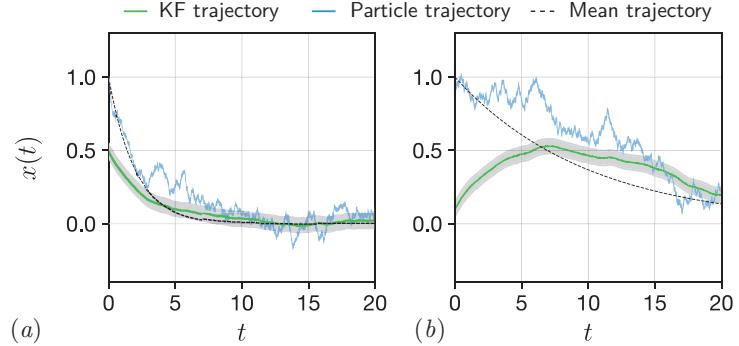


Figure 5. Evolution of the position of a Brownian particle in a harmonic potential (in blue) and its estimate using Kalman-Bucy filter (in green) when $\alpha(t) = \sigma(t) = 1$, in (a) $f(t) = 0.5, \hat{x}(0) = 0.5, \beta(0) = 0.01$ and (b) $f(t) = 0.1, \hat{x}(0) = 0.1, \beta(0) = 0.08$. The dashed line is the mean of state vector $\mathbb{E}[x(t)] = x_0 \exp(ft)$. Further details are available in the [python-notebook](#).

cast as a *batch estimation* problem solving for the state trajectory most likely to produce the observed measurement sequence. This formulation is natural since the measurement sequence constitutes the available data in the estimation problem. However, the Kalman filter reformulates the batch estimation problem into a recursive filtering problem, maintaining a running state estimate $\hat{\theta}_k$ that is recursively updated with only a single incoming measurement m_k at time k , as opposed to working with the entire (batch) measurement sequence m_1, \dots, m_k . This recursive structure of the Kalman filter opens up the possibility for comparison with Bayesian estimation. To this end, we first recall the Bayes rule given by

$$p(\boldsymbol{\theta} | \mathbf{m}) = \frac{p(\boldsymbol{\theta}) p(\mathbf{m} | \boldsymbol{\theta})}{\int p(\boldsymbol{\theta}) p(\mathbf{m} | \boldsymbol{\theta}) d\boldsymbol{\theta}}.$$

Here $p(\boldsymbol{\theta} | \mathbf{m})$ is the posterior probability of the parameter $\boldsymbol{\theta}$ given the measurement \mathbf{m} , $p(\boldsymbol{\theta})$ is the prior probability on the parameter space and $p(\mathbf{m} | \boldsymbol{\theta})$ is the likelihood of the measurement \mathbf{m} given the parameter $\boldsymbol{\theta}$. Ignoring the normalization constant (which is automatically retrieved when the posterior is constrained to be a probability distribution), Bayes rule gives us the relation $p(\boldsymbol{\theta} | \mathbf{m}) \propto p(\boldsymbol{\theta}) p(\mathbf{m} | \boldsymbol{\theta})$. Loosely interpreting Kalman filtering in Bayesian terms, the estimate propagation step of the Kalman filter propagates the state estimate (mean and covariance) from the previous time step to generate a prior for the current step, whereas



the estimate update step corrects the prior using the measurement residual to produce the posterior.

In Bayesian filtering, the current prior is obtained by propagation of the posterior from the previous time using the known system dynamics and is then corrected according to Bayes rule (using the current measurement and the likelihood function) to obtain the current posterior, and the entire process is carried out recursively. We will now show precisely that the recursive Bayesian estimator yields the Kalman filter when the underlying system dynamics is linear and the probability distributions of the noise (in the system dynamics and measurement model) are Gaussian.

Bayesian prior

The prior probability for the current time k is obtained by forward propagating the posterior probability from the previous time $k - 1$ using the system dynamics. Since the underlying probability distributions are Gaussian (this can be verified from the fact that for any initial state of the system, the linear dynamics and Gaussian noise generate a sequence of states θ_k which follow a Gaussian distribution for every k), this step involves forward propagating the mean and covariance of the posterior from time $k - 1$. The posterior mean at time $k - 1$, denoted by $\hat{\theta}_{k-1}$ is propagated by the dynamics to $\Phi_{k-1}\hat{\theta}_{k-1}$. Similarly, the prior covariance at time k is obtained as $\beta_k^- = \Phi_{k-1}\beta_{k-1}\Phi_{k-1}^\top + \Lambda_{k-1}\mathbf{q}_{k-1}\Lambda_{k-1}^\top$, to define the prior probability density as

$$p_k(\boldsymbol{\theta} | \hat{\theta}_{k-1}) \sim e^{-(\boldsymbol{\theta} - \Phi_{k-1}\hat{\theta}_{k-1})^\top \beta_k^{-1} (\boldsymbol{\theta} - \Phi_{k-1}\hat{\theta}_{k-1})}.$$

Note that the generation of the above prior merely involves forward propagation through the system dynamics and is identical to the estimates propagation step of the Kalman filter.



Likelihood

The measurement model, with Gaussian noise of covariance σ_k at time k , yields the likelihood function

$$p_k(\mathbf{m} \mid \boldsymbol{\theta}) \sim e^{-(\mathbf{m} - \boldsymbol{\alpha}_k \boldsymbol{\theta})^\top \boldsymbol{\sigma}_k^{-1} (\mathbf{m} - \boldsymbol{\alpha}_k \boldsymbol{\theta})}.$$

Posterior

The posterior probability is then obtained by combining the prior probability and likelihood according to Bayes rule, and is given by

$$p_k(\boldsymbol{\theta} \mid \mathbf{m}_k, \hat{\boldsymbol{\theta}}_{k-1}) \sim e^{-(\boldsymbol{\theta} - \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1})^\top \boldsymbol{\beta}_k^{-1} (\boldsymbol{\theta} - \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1})} \cdot e^{-(\mathbf{m}_k - \boldsymbol{\alpha}_k \boldsymbol{\theta})^\top \boldsymbol{\sigma}_k^{-1} (\mathbf{m}_k - \boldsymbol{\alpha}_k \boldsymbol{\theta})}.$$

We now compute the posterior mean using the fact that the mean of a Gaussian distribution is also its mode (and the only critical point of its probability density function). The posterior mean $\hat{\boldsymbol{\theta}}_k$ is then obtained as the solution to

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left[p_k(\boldsymbol{\theta} \mid \mathbf{m}_k, \hat{\boldsymbol{\theta}}_{k-1}) \right] \Big|_{\hat{\boldsymbol{\theta}}_k} = 0,$$

which yields

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} & \left[(\boldsymbol{\theta} - \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1})^\top \boldsymbol{\beta}_k^{-1} (\boldsymbol{\theta} - \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1}) \right. \\ & \left. + (\mathbf{m}_k - \boldsymbol{\alpha}_k \boldsymbol{\theta})^\top \boldsymbol{\sigma}_k^{-1} (\mathbf{m}_k - \boldsymbol{\alpha}_k \boldsymbol{\theta}) \right] \Big|_{\hat{\boldsymbol{\theta}}_k} = 0. \end{aligned}$$

Solving the above, we get

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{\beta}_k^{-1} + \boldsymbol{\alpha}_k^\top \boldsymbol{\sigma}_k^{-1} \boldsymbol{\alpha}_k \right)^{-1} \left(\boldsymbol{\beta}_k^{-1} \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{\alpha}_k^\top \boldsymbol{\sigma}_k^{-1} \mathbf{m}_k \right).$$

It can be verified by the application of the Woodbury matrix identity that $\hat{\boldsymbol{\theta}}_k$ above can be expressed as

$$\hat{\boldsymbol{\theta}}_k = \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{\kappa}_k \left(\mathbf{m}_k - \boldsymbol{\alpha}_k \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{\theta}}_{k-1} \right),$$



where $\kappa_k = \beta_k^- \alpha_k^\top \left(\sigma_k + \alpha_k \beta_k^- \alpha_k^\top \right)^{-1}$. Note that the posterior mean $\hat{\theta}_k$ and the gain κ_k above are identical to the Kalman filter updated estimate and the Kalman gain, respectively. It can also be verified (we skip the calculation here) that the posterior covariance matches the updated covariance from the Kalman filter.

6. Beyond Kalman filter

Following the advent of KF, several techniques were developed to accommodate the non-linearity of the dynamics and the non-gaussian aspects of the noise that often infect real world problems. Some of the popular techniques in literature include Extended Kalman Filter, Particle Filter, Unscented Kalman Filter, Ensemble Kalman Filter, Moving horizon estimation, and Recursive Bayesian estimation. Here we describe in brief the ideas behind Extended Kalman Filter and Particle Filter and point the interested reader to [9] for details of other techniques.

Extended Kalman Filter

The process of state estimation under Kalman Filter is valid only for linear dynamics of the state vector. Exact updates of the state vector for non-linear systems are in-general not available as the non-linearity results in a non-gaussian distribution of the state vector. An extension of the KF technique that is applied to non-linear systems with Gaussian noise is by locally linearizing the dynamics and is termed as the Extended Kalman Filter (shortened as EKF). This technique has been applied to several real-world scenarios and shown to perform well. However, EKF does not preserve the true mean and variance of the state vector and does not work for systems with multi-modal distributions.



Particle Filter

Particle filtering is another technique used for state estimation when the underlying dynamics is non-linear and/or the process is non-gaussian. This technique uses particles to represent the posterior distribution of the state with the noisy measurements. Each particle has a likelihood weight that represents the probability of that particle/state being sampled from the probability density function of the state determined by sampling techniques such Monte Carlo methods. These weights and samples are then used to estimate the state of the system and as the number of samples becomes large, this distribution becomes equivalent to the true posterior distribution. Particle filtering technique is very popular as it makes little assumption about the underlying dynamics. However, this technique does not perform well when applied to high-dimensional dynamics and is often computationally expensive.

7. Conclusion

Kalman Filter was developed in 1960 first for aerospace applications and it played a pivotal role in the successful completion of the Apollo moon project, where it was used for trajectory estimation and navigation [10]. Post that, it was used for radar tracking of aircraft and was also critical in the development of the Global Positioning System (GPS) (which was then described as “one enormous Kalman filter” [10]). Since its origin, KF has remained the go-to tool for sensor calibration and error correction. They have even been used in detecting constrictions of cerebral arterial blood vessels, detecting cancerous tumours, control of protein aggregation and in a gazillion other scenarios [10, 11, 12].

In summary, we see that the Kalman filter is a powerful tool with wide ranging practical applications due to the recursive nature of the computations involved (no memory constraint), ease of implementation and a minimal set of



assumptions required for its working. The foundation of the theory, however, is the famous least-squares fit which features in undergraduate textbooks. Several open questions remain till date as to the extension of the Kalman filter idea to non-linear/complex systems without simplifying assumptions that works for a wide range of dynamics. The physics community, in particular, is yet to dive into the estimation question and use some of the understandings of complex systems to answer the estimation question for systems that are physically constrained.

Suggested Reading

- [1] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [2] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. 1961.
- [3] Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 1994.
- [4] Harold W Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE spectrum*, 7(7):63–68, 1970.
- [5] Peter S Maybeck. *Stochastic models, estimation, and control*. Academic press, 1982.
- [6] Venkataraman Balakrishnan. *Elements of nonequilibrium statistical mechanics*, volume 3. Springer, 2008.
- [7] Subrahmanyam Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of modern physics*, 15(1):1, 1943.
- [8] Mark K Transtrum, Benjamin B Machta, Kevin S Brown, Bryan C Daniels, Christopher R Myers, and James P Sethna. Perspective: Sloppiness and emergent theories in physics, biology, and beyond. *The Journal of chemical physics*, 143(1):07B201_1, 2015.
- [9] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [10] Mohinder S Grewal and Angus P Andrews. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78, 2010.
- [11] Vicente I Fernandez, Pallav Kosuri, Vicente Parot, and Julio M Fernandez. Extended kalman filter estimates the contour length of a protein in single molecule atomic force microscopy experiments. *Review of Scientific Instruments*, 80(11):113104, 2009.
- [12] Victor M Moreno and Alberto Pigazo. Kalman filter: recent advances and applications. 2009.

