WIKIPEDIA

Kalman filter

In statistics and control theory, **Kalman filtering**, also known as **linear quadratic estimation** (**LQE**), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe. The filter is named after Rudolf E. Kálmán, one of the primary developers of its theory.
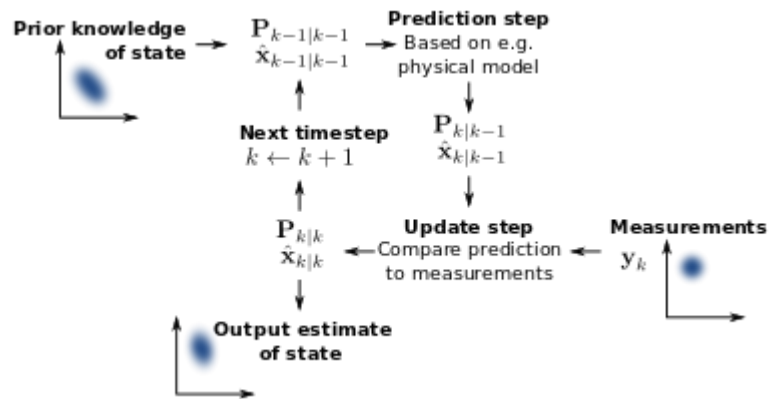
The Kalman filter has numerous applications in technology. A common application is for guidance, navigation, and control of vehicles, particularly aircraft and spacecraft.[1] Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and

The Kalman filter keeps track of the estimated state of the system and the variance or uncertainty of the estimate. The estimate is updated using a state transition model and measurements. $\hat{x}_{k|k-1}$ denotes the estimate of the system's state at time step k before the k-th measurement y_k has been taken into account; $P_{k|k-1}$ is the corresponding uncertainty.

econometrics. Kalman filters also are one of the main topics in the field of robotic motion planning and control, and they are sometimes included in trajectory optimization. The Kalman filter also works for modeling the central nervous system's control of movement. Due to the time delay between issuing motor commands and receiving sensory feedback, use of the Kalman filter supports a realistic model for making estimates of the current state of the motor system and issuing updated commands.[2]

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

Using a Kalman filter does not assume that the errors are Gaussian.[3] However, the filter yields the exact conditional probability estimate in the special case that all errors are Gaussian.

Extensions and generalizations to the method have also been developed, such as the extended Kalman filter and the unscented Kalman filter which work on nonlinear systems. The underlying model is similar to a hidden Markov model except that the state space of the latent variables is continuous and all latent and observed variables have Gaussian distributions.

Contents

History

The filter is named after Hungarian émigré Rudolf E. Kálmán, although Thorvald Nicolai Thiele[4][5] and Peter Swerling developed a similar algorithm earlier. Richard S. Bucy of the University of Southern California contributed to the theory, leading to it sometimes being called the Kalman–Bucy filter. Stanley F. Schmidt is generally credited with developing the first implementation of a Kalman filter. He realized that the filter could be divided into two distinct parts, with one part for time periods between sensor outputs and another part for incorporating measurements.[6] It was during a visit by Kálmán to the NASA Ames Research Center that Schmidt saw the applicability of Kálmán's ideas to the nonlinear problem of trajectory estimation for the Apollo program leading to its incorporation in the Apollo navigation computer. This Kalman filter was first described and partially developed in technical papers by Swerling (1958), Kalman (1960) and Kalman and Bucy (1961).

> The Apollo computer used 2k of magnetic core RAM and 36k wire rope [...]. The CPU was built from ICs [...]. Clock speed was under 100 kHz [...]. The fact that the MIT engineers were able to pack such good software (one of the very first applications of the Kalman filter) into such a tiny computer is truly remarkable.
>
> — Interview with Jack Crenshaw, by Matthew Reed, TRS-80.org (2009) [1] (http://www.trs-80.org/interview-jack-crenshaw/)

Kalman filters have been vital in the implementation of the navigation systems of U.S. Navy nuclear ballistic missile submarines, and in the guidance and navigation systems of cruise missiles such as the U.S. Navy's Tomahawk missile and the U.S. Air Force's Air Launched Cruise Missile. They are also used in the guidance and navigation systems of reusable launch vehicles and the attitude control and navigation systems of spacecraft which dock at the International Space Station.[7]

This digital filter is sometimes called the *Stratonovich–Kalman–Bucy filter* because it is a special case of a more general, non-linear filter developed somewhat earlier by the Soviet mathematician Ruslan Stratonovich.[8][9][10][11] In fact, some of the special case linear filter's equations appeared in these papers by Stratonovich that were published before summer 1960, when Kalman met with Stratonovich during a conference in Moscow.[12]

Overview of the calculation

The Kalman filter uses a system's dynamic model (e.g., physical laws of motion), known control inputs to that system, and multiple sequential measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using only one measurement alone. As such, it is a common sensor fusion and data fusion algorithm.

Noisy sensor data, approximations in the equations that describe the system evolution, and external factors that are not accounted for all place limits on how well it is possible to determine the system's state. The Kalman filter deals effectively with the uncertainty due to noisy sensor data and to some extent also with random external factors. The Kalman filter produces an estimate of the state of the system as an average of the system's predicted state and of the new measurement using a weighted average. The purpose of the weights is that values with better (i.e., smaller) estimated uncertainty are "trusted" more. The weights are calculated from the covariance, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated at every time step, with the new estimate and its covariance informing the prediction used in the following iteration. This means that the Kalman filter works recursively and requires only the last "best guess", rather than the entire history, of a system's state to calculate a new state.

The relative certainty of the measurements and current state estimate is an important consideration, and it is common to discuss the response of the filter in terms of the Kalman filter's *gain*. The Kalman gain is the relative weight given to the measurements and current state estimate, and can be "tuned" to achieve particular performance. With a high gain, the filter places more weight on the

most recent measurements, and thus follows them more responsively. With a low gain, the filter follows the model predictions more closely. At the extremes, a high gain close to one will result in a more jumpy estimated trajectory, while low gain close to zero will smooth out noise but decrease the responsiveness.

When performing the actual calculations for the filter (as discussed below), the state estimate and covariances are coded into matrices to handle the multiple dimensions involved in a single set of calculations. This allows for a representation of linear relationships between different state variables (such as position, velocity, and acceleration) in any of the transition models or covariances.

Example application

As an example application, consider the problem of determining the precise location of a truck. The truck can be equipped with a GPS unit that provides an estimate of the position within a few meters. The GPS estimate is likely to be noisy; readings 'jump around' rapidly, though remaining within a few meters of the real position. In addition, since the truck is expected to follow the laws of physics, its position can also be estimated by integrating its velocity over time, determined by keeping track of wheel revolutions and the angle of the steering wheel. This is a technique known as dead reckoning. Typically, the dead reckoning will provide a very smooth estimate of the truck's position, but it will drift over time as small errors accumulate.

In this example, the Kalman filter can be thought of as operating in two distinct phases: predict and update. In the prediction phase, the truck's old position will be modified according to the physical laws of motion (the dynamic or "state transition" model). Not only will a new position estimate be calculated, but a new covariance will be calculated as well. Perhaps the covariance is proportional to the speed of the truck because we are more uncertain about the accuracy of the dead reckoning position estimate at high speeds but very certain about the position estimate when moving slowly. Next, in the update phase, a measurement of the truck's position is taken from the GPS unit. Along with this measurement comes some amount of uncertainty, and its covariance relative to that of the prediction from the previous phase determines how much the new measurement will affect the updated prediction. Ideally, as the dead reckoning estimates tend to drift away from the real position, the GPS measurement should pull the position estimate back towards the real position but not disturb it to the point of becoming rapidly jumping and noisy.

Technical description and context

The Kalman filter is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements. It is used in a wide range of engineering and econometric applications from radar and computer vision to estimation of structural macroeconomic models,[13][14] and is an important topic in control theory and control systems engineering. Together with the linear-quadratic regulator (LQR), the Kalman filter solves the linear–quadratic–Gaussian control problem (LQG). The Kalman filter, the linear-quadratic regulator and the linear–quadratic–Gaussian controller are solutions to what arguably are the most fundamental problems in control theory.

In most applications, the internal state is much larger (more degrees of freedom) than the few "observable" parameters which are measured. However, by combining a series of measurements, the Kalman filter can estimate the entire internal state.

In Dempster–Shafer theory, each state equation or observation is considered a special case of a linear belief function and the Kalman filter is a special case of combining linear belief functions on a join-tree or Markov tree. Additional approaches include belief filters which use Bayes or evidential updates to the state equations.

A wide variety of Kalman filters have now been developed, from Kalman's original formulation, now called the "simple" Kalman filter, the Kalman–Bucy filter, Schmidt's "extended" filter, the information filter, and a variety of "square-root" filters that were developed by Bierman, Thornton and many others. Perhaps the most commonly used type of very simple Kalman filter is the phase-locked loop, which is now ubiquitous in radios, especially frequency modulation (FM) radios, television sets, satellite communications receivers, outer space communications systems, and nearly any other electronic communications equipment.

Underlying dynamical system model

The Kalman filters are based on <u>linear dynamical systems</u> discretized in the time domain. They are modeled on a <u>Markov chain</u> built on <u>linear operators</u> perturbed by errors that may include <u>Gaussian</u> <u>noise</u>. The <u>state</u> of the system is represented as a <u>vector</u> of <u>real numbers.</u> At each <u>discrete time</u> increment, a linear operator is applied to the state to generate the new state, with some noise mixed in, and optionally some information from the controls on the system if they are known. Then, another linear operator mixed with more noise generates the observed outputs from the true ("hidden") state. The Kalman filter may be regarded as analogous to the hidden Markov model, with the key difference that the hidden state variables take values in a continuous space (as opposed to a discrete state space as in the hidden Markov model). There is a strong analogy between the equations of the Kalman Filter and those of the hidden Markov model. A review of this and other models is given in Roweis and <u>Ghahramani</u> (1999),[15] and Hamilton (1994), Chapter 13.[16]

In order to use the Kalman filter to estimate the internal state of a process given only a sequence of noisy observations, one must model the process in accordance with the framework of the Kalman filter. This means specifying the following matrices:

- F_k, the state-transition model;
- H_k, the observation model;
- Q_k, the <u>covariance</u> of the process noise;
- R_k, the <u>covariance</u> of the observation noise;
- and sometimes B_k, the control-input model, for each time-step, k, as described below.
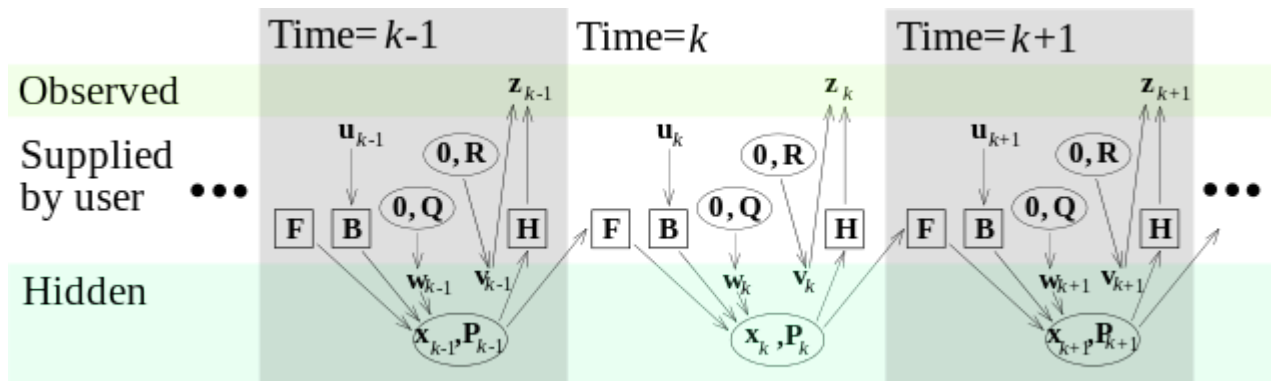
Model underlying the Kalman filter. Squares represent matrices. Ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors. In the simple case, the various matrices are constant with time, and thus the subscripts are dropped, but the Kalman filter allows any of them to change each time step.

The Kalman filter model assumes the true state at time k is evolved from the state at $(k − 1)$ according to

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

where

- $\mathbf{F}_k$ is the state transition model which is applied to the previous state $\mathbf{x}_{k-1}$;
- $\mathbf{B}_k$ is the control-input model which is applied to the control vector $\mathbf{u}_k$;
- $\mathbf{w}_k$ is the process noise which is assumed to be drawn from a zero mean <u>multivariate normal distribution</u>, $\mathcal{N}$, with <u>covariance</u>, $\mathbf{Q}_k$: $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$.

At time k an observation (or measurement) $\mathbf{z}_k$ of the true state $\mathbf{x}_k$ is made according to

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

where

- $\mathbf{H}_k$ is the observation model which maps the true state space into the observed space and
- $\mathbf{v}_k$ is the observation noise which is assumed to be zero mean Gaussian <u>white noise</u> with covariance $\mathbf{R}_k$:
  $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$.

The initial state, and the noise vectors at each step $\{\mathbf{x}_0, \mathbf{w}_1, ..., \mathbf{w}_k, \mathbf{v}_1 ... \mathbf{v}_k\}$ are all assumed to be mutually <u>independent</u>.

Many real dynamical systems do not exactly fit this model. In fact, unmodeled dynamics can seriously degrade the filter performance, even when it was supposed to work with unknown stochastic signals as inputs. The reason for this is that the effect of unmodeled dynamics depends on the input, and, therefore, can bring the estimation algorithm to instability (it diverges). On the other hand, independent white noise signals will not make the algorithm diverge. The problem of distinguishing between measurement noise and unmodeled dynamics is a difficult one and is treated in control theory under the framework of <u>robust control</u>.[17][18]

Details

The Kalman filter is a <u>recursive</u> estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. In what follows, the notation $\hat{\mathbf{x}}_{n|m}$ represents the estimate of $\mathbf{x}$ at time n given observations up to and including at time $m \le n$.

The state of the filter is represented by two variables:

- $\hat{\mathbf{x}}_{k|k}$, the *a posteriori* state estimate at time k given observations up to and including at time k;
- $\mathbf{P}_{k|k}$, the *a posteriori* error covariance matrix (a measure of the estimated <u>accuracy</u> of the state estimate).

The Kalman filter can be written as a single equation, however it is most often conceptualized as two distinct phases: "Predict" and "Update". The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. This predicted state estimate is also known as the *a priori* state estimate because, although it is an estimate of the state at the current timestep, it does not include observation information from the current timestep. In the update phase, the current *a priori* prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the *a posteriori* state estimate.

Typically, the two phases alternate, with the prediction advancing the state until the next scheduled observation, and the update incorporating the observation. However, this is not necessary; if an observation is unavailable for some reason, the update may be skipped and multiple prediction steps performed. Likewise, if multiple independent observations are available at the same time, multiple update steps may be performed (typically with different observation matrices $\mathbf{H}_k$).[19][20]

Predict

Predicted (*a priori*) state estimate	$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$
Predicted (*a priori*) error covariance	$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\mathsf{T} + \mathbf{Q}_k$

Update

Innovation or measurement pre-fit residual	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$
Innovation (or pre-fit residual) covariance	$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\mathsf{T}$

Optimal Kalman gain	$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{S}_k^{-1}$
Updated (*a posteriori*) state estimate	$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$
Updated (*a posteriori*) estimate covariance	$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k|k-1}$
Measurement post-fit residual	$\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k}$

The formula for the updated (*a posteriori*) estimate covariance above is valid for any gain $\mathbf{K}_k$ and is sometimes called the **Joseph form**. For the optimal Kalman gain the formula further simplifies to $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k|k-1}$, in which form it is most widely used in applications. However, one must keep in mind, that it is valid only for the optimal gain that minimizes the residual error. Proof of the formulae is found in the *derivations* section.

Invariants

If the model is accurate, and the values for $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ accurately reflect the distribution of the initial state values, then the following invariants are preserved:

$$\mathbf{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = \mathbf{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] = 0$$
$$\mathbf{E}[\tilde{\mathbf{y}}_k] = 0$$

where $\mathbf{E}[\xi]$ is the expected value of ξ. That is, all estimates have a mean error of zero.

Also:

$$\mathbf{P}_{k|k} = \mathrm{cov}\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)$$
$$\mathbf{P}_{k|k-1} = \mathrm{cov}\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)$$
$$\mathbf{S}_k = \mathrm{cov}(\tilde{\mathbf{y}}_k)$$

so covariance matrices accurately reflect the covariance of estimates.

Estimation of the noise covariances Q_k and R_k

Practical implementation of the Kalman Filter is often difficult due to the difficulty of getting a good estimate of the noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$. Extensive research has been done in this field to estimate these covariances from data. One practical approach to do this is the *autocovariance least-squares (ALS)* technique that uses the time-lagged autocovariances of routine operating data to estimate the covariances.[21][22] The GNU Octave and Matlab code used to calculate the noise covariance matrices using the ALS technique is available online under the GNU General Public License.[23]

Optimality and performance

It follows from theory that the Kalman filter is the optimal linear filter in cases where a) the model perfectly matches the real system, b) the entering noise is white (uncorrelated) and c) the covariances of the noise are exactly known. Several methods for the noise covariance estimation have been proposed during past decades, including ALS, mentioned in the section above. After the covariances are estimated, it is useful to evaluate the performance of the filter; i.e., whether it is possible to improve the state estimation quality. If the Kalman filter works optimally, the innovation sequence (the output prediction error) is a white noise, therefore the whiteness

property of the innovations measures filter performance. Several different methods can be used for this purpose.[24] If the noise terms are non-Gaussian distributed, methods for assessing performance of the filter estimate, which use probability inequalities or large-sample theory, are known in the literature.[25][26]

Example application, technical

Consider a truck on frictionless, straight rails. Initially, the truck is stationary at position 0, but it is buffeted this way and that by random uncontrolled forces. We measure the position of the truck every Δt seconds, but these measurements are imprecise; we want to maintain a model of where the truck is and what is its velocity. We show here how we derive the model from which we create our Kalman filter.

Since $\mathbf{F}, \mathbf{H}, \mathbf{R}, \mathbf{Q}$ are constant, their time indices are dropped.

The position and velocity of the truck are described by the linear state space

Truth;   filtered process; observations.

$$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

where $\dot{x}$ is the velocity, that is, the derivative of position with respect to time.

We assume that between the $(k-1)$ and k timestep uncontrolled forces cause a constant acceleration of a_k that is normally distributed, with mean 0 and standard deviation σ_a. From Newton's laws of motion we conclude that

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

(note that there is no $\mathbf{B}u$ term since we have no known control inputs. Instead, we assume that a_k is the effect of an unknown input and $\mathbf{G}$ applies that effect to the state vector) where

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$$

so that

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k$$

where

$$\mathbf{w}_k \sim N(0, \mathbf{Q})$$

$$\mathbf{Q} = \mathbf{G}\mathbf{G}^\mathsf{T}\sigma_a^2 = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 \end{bmatrix} \sigma_a^2.$$

Please note that the matrix $\mathbf{Q}$ is not full rank (it is of rank one if $\Delta t \neq 0$). Hence, the distribution $N(0, \mathbf{Q})$ is not absolutely continuous and has no probability density function. Another way to express this, avoiding explicit degenerate distributions is given by

$$\mathbf{w}_k \sim \mathbf{G} \cdot N\left(0, \sigma_a^2\right).$$

At each time step, a noisy measurement of the true position of the truck is made. Let us suppose the measurement noise v_k is also normally distributed, with mean 0 and standard deviation σ_z.

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

where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and

$$\mathbf{R} = \mathbf{E}\left[\mathbf{v}_k \mathbf{v}_k^\mathsf{T}\right] = \begin{bmatrix} \sigma_z^2 \end{bmatrix}$$

We know the initial starting state of the truck with perfect precision, so we initialize

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and to tell the filter that we know the exact position and velocity, we give it a zero covariance matrix:

$$\mathbf{P}_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

If the initial position and velocity are not known perfectly, the covariance matrix should be initialized with suitable variances on its diagonal:

$$\mathbf{P}_{0|0} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_{\dot{x}}^2 \end{bmatrix}$$

The filter will then prefer the information from the first measurements over the information already in the model.

Derivations

Deriving the *posteriori* estimate covariance matrix

Starting with our invariant on the error covariance $\mathbf{P}_{k\,|\,k}$ as above

$$\mathbf{P}_{k|k} = \operatorname{cov}\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)$$

substitute in the definition of $\hat{\mathbf{x}}_{k|k}$

$$\mathbf{P}_{k|k} = \operatorname{cov}\left[\mathbf{x}_k - \left(\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k\right)\right]$$

and substitute $\tilde{\mathbf{y}}_k$

$$\mathbf{P}_{k|k} = \operatorname{cov}\left(\mathbf{x}_k - \left[\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}\right)\right]\right)$$

and $\mathbf{z}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}\Big(\mathbf{x}_k - \big[\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right)\big]\Big)$$

and by collecting the error vectors we get

$$\mathbf{P}_{k|k} = \mathrm{cov}\Big[(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right) - \mathbf{K}_k\mathbf{v}_k\Big]$$

Since the measurement error $\mathbf{v}_k$ is uncorrelated with the other terms, this becomes

$$\mathbf{P}_{k|k} = \mathrm{cov}\Big[(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\Big] + \mathrm{cov}[\mathbf{K}_k\mathbf{v}_k]$$

by the properties of <u>vector covariance</u> this becomes

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathrm{cov}\big(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\big)(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k\,\mathrm{cov}(\mathbf{v}_k)\mathbf{K}_k^\mathsf{T}$$

which, using our invariant on $\mathbf{P}_{k\,|\,k-1}$ and the definition of $\mathbf{R}_k$ becomes

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathsf{T}$$

This formula (sometimes known as the **Joseph form** of the covariance update equation) is valid for any value of $\mathbf{K}_k$. It turns out that if $\mathbf{K}_k$ is the optimal Kalman gain, this can be simplified further as shown below.

Kalman gain derivation

The Kalman filter is a <u>minimum mean-square error</u> estimator. The error in the *a posteriori* state estimation is

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$$

We seek to minimize the expected value of the square of the magnitude of this vector, $\mathbf{E}\Big[\big\|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\big\|^2\Big]$. This is equivalent to minimizing the <u>trace</u> of the *a posteriori* estimate <u>covariance matrix</u> $\mathbf{P}_{k|k}$. By expanding out the terms in the equation above and collecting, we get:

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{K}_k^\mathsf{T} + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T} + \mathbf{R}_k\right)\mathbf{K}_k^\mathsf{T}$$

$$= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{K}_k^\mathsf{T} + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\mathsf{T}$$

The trace is minimized when its <u>matrix derivative</u> with respect to the gain matrix is zero. Using the <u>gradient matrix rules</u> and the symmetry of the matrices involved we find that

$$\frac{\partial\ \mathrm{tr}(\mathbf{P}_{k|k})}{\partial\ \mathbf{K}_k} = -2\big(\mathbf{H}_k\mathbf{P}_{k|k-1}\big)^\mathsf{T} + 2\mathbf{K}_k\mathbf{S}_k = 0.$$

Solving this for $\mathbf{K}_k$ yields the Kalman gain:

$$\mathbf{K}_k\mathbf{S}_k = \big(\mathbf{H}_k\mathbf{P}_{k|k-1}\big)^\mathsf{T} = \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}$$
$$\Rightarrow \mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{S}_k^{-1}$$

This gain, which is known as the *optimal Kalman gain*, is the one that yields MMSE estimates when used.

Simplification of the *posteriori* error covariance formula

The formula used to calculate the *a posteriori* error covariance can be simplified when the Kalman gain equals the optimal value derived above. Multiplying both sides of our Kalman gain formula on the right by $\mathbf{S}_k\mathbf{K}_k^\mathsf{T}$, it follows that

$$\mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\mathsf{T} = \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{K}_k^\mathsf{T}$$

Referring back to our expanded formula for the *a posteriori* error covariance,

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\mathbf{K}_k^\mathsf{T} + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\mathsf{T}$$

we find the last two terms cancel out, giving

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.$$

This formula is computationally cheaper and thus nearly always used in practice, but is only correct for the optimal gain. If arithmetic precision is unusually low causing problems with numerical stability, or if a non-optimal Kalman gain is deliberately used, this simplification cannot be applied; the *a posteriori* error covariance formula as derived above (Joseph form) must be used.

Sensitivity analysis

The Kalman filtering equations provide an estimate of the state $\hat{\mathbf{x}}_{k|k}$ and its error covariance $\mathbf{P}_{k|k}$ recursively. The estimate and its quality depend on the system parameters and the noise statistics fed as inputs to the estimator. This section analyzes the effect of uncertainties in the statistical inputs to the filter.[27] In the absence of reliable statistics or the true values of noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, the expression

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathsf{T}$$

no longer provides the actual error covariance. In other words, $\mathbf{P}_{k|k} \neq E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)^\mathsf{T}\right]$. In most real-time applications, the covariance matrices that are used in designing the Kalman filter are different from the actual (true) noise covariances matrices. This sensitivity analysis describes the behavior of the estimation error covariance when the noise covariances as well as the system matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ that are fed as inputs to the filter are incorrect. Thus, the sensitivity analysis describes the robustness (or sensitivity) of the estimator to misspecified statistical and parametric inputs to the estimator.

This discussion is limited to the error sensitivity analysis for the case of statistical uncertainties. Here the actual noise covariances are denoted by $\mathbf{Q}_k^a$ and $\mathbf{R}_k^a$ respectively, whereas the design values used in the estimator are $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively. The actual error covariance is denoted by $\mathbf{P}_{k|k}^a$ and $\mathbf{P}_{k|k}$ as computed by the Kalman filter is referred to as the Riccati variable. When $\mathbf{Q}_k \equiv \mathbf{Q}_k^a$ and $\mathbf{R}_k \equiv \mathbf{R}_k^a$, this means that $\mathbf{P}_{k|k} = \mathbf{P}_{k|k}^a$. While computing the actual error covariance using $\mathbf{P}_{k|k}^a = E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)^\mathsf{T}\right]$, substituting for $\widehat{\mathbf{x}}_{k|k}$ and using the fact that $E\left[\mathbf{w}_k\mathbf{w}_k^\mathsf{T}\right] = \mathbf{Q}_k^a$ and $E\left[\mathbf{v}_k\mathbf{v}_k^\mathsf{T}\right] = \mathbf{R}_k^a$, results in the following recursive equations for $\mathbf{P}_{k|k}^a$:

$$\mathbf{P}_{k|k-1}^a = \mathbf{F}_k\mathbf{P}_{k-1|k-1}^a\mathbf{F}_k^\mathsf{T} + \mathbf{Q}_k^a$$

and

$$\mathbf{P}^a_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \, \mathbf{P}^a_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k \mathbf{R}^a_k \mathbf{K}^\mathsf{T}_k$$

While computing $\mathbf{P}_{k|k}$, by design the filter implicitly assumes that $E\left[\mathbf{w}_k \mathbf{w}^\mathsf{T}_k\right] = \mathbf{Q}_k$ and $E\left[\mathbf{v}_k \mathbf{v}^\mathsf{T}_k\right] = \mathbf{R}_k$. Note that the recursive expressions for $\mathbf{P}^a_{k|k}$ and $\mathbf{P}_{k|k}$ are identical except for the presence of $\mathbf{Q}^a_k$ and $\mathbf{R}^a_k$ in place of the design values $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively. Researches have been done to analyze Kalman filter system's robustness.[28]

Square root form

One problem with the Kalman filter is its numerical stability. If the process noise covariance $\mathbf{Q}_k$ is small, round-off error often causes a small positive eigenvalue to be computed as a negative number. This renders the numerical representation of the state covariance matrix $\mathbf{P}$ indefinite, while its true form is positive-definite.

Positive definite matrices have the property that they have a triangular matrix square root $\mathbf{P} = \mathbf{S} \cdot \mathbf{S}^\mathsf{T}$. This can be computed efficiently using the Cholesky factorization algorithm, but more importantly, if the covariance is kept in this form, it can never have a negative diagonal or become asymmetric. An equivalent form, which avoids many of the square root operations required by the matrix square root yet preserves the desirable numerical properties, is the U-D decomposition form, $\mathbf{P} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{U}^\mathsf{T}$, where $\mathbf{U}$ is a unit triangular matrix (with unit diagonal), and $\mathbf{D}$ is a diagonal matrix.

Between the two, the U-D factorization uses the same amount of storage, and somewhat less computation, and is the most commonly used square root form. (Early literature on the relative efficiency is somewhat misleading, as it assumed that square roots were much more time-consuming than divisions,[29]:69 while on 21-st century computers they are only slightly more expensive.)

Efficient algorithms for the Kalman prediction and update steps in the square root form were developed by G. J. Bierman and C. L. Thornton.[29][30]

The $\mathbf{L} \cdot \mathbf{D} \cdot \mathbf{L}^\mathsf{T}$ decomposition of the innovation covariance matrix $\mathbf{S}_k$ is the basis for another type of numerically efficient and robust square root filter.[31] The algorithm starts with the LU decomposition as implemented in the Linear Algebra PACKage (LAPACK). These results are further factored into the $\mathbf{L} \cdot \mathbf{D} \cdot \mathbf{L}^\mathsf{T}$ structure with methods given by Golub and Van Loan (algorithm 4.1.2) for a symmetric nonsingular matrix.[32] Any singular covariance matrix is pivoted so that the first diagonal partition is nonsingular and well-conditioned. The pivoting algorithm must retain any portion of the innovation covariance matrix directly corresponding to observed state-variables $\mathbf{H}_k \cdot \mathbf{x}_{k|k-1}$ that are associated with auxiliary observations in $\mathbf{y}_k$. The $\mathbf{l} \cdot \mathbf{d} \cdot \mathbf{l}^\mathsf{t}$ square-root filter requires orthogonalization of the observation vector.[30][31] This may be done with the inverse square-root of the covariance matrix for the auxiliary variables using Method 2 in Higham (2002, p. 263).[33]

Relationship to recursive Bayesian estimation

The Kalman filter can be presented as one of the simplest dynamic Bayesian networks. The Kalman filter calculates estimates of the true values of states recursively over time using incoming measurements and a mathematical process model. Similarly, recursive Bayesian estimation calculates estimates of an unknown probability density function (PDF) recursively over time using incoming measurements and a mathematical process model.[34]

In recursive Bayesian estimation, the true state is assumed to be an unobserved Markov process, and the measurements are the observed states of a hidden Markov model (HMM).
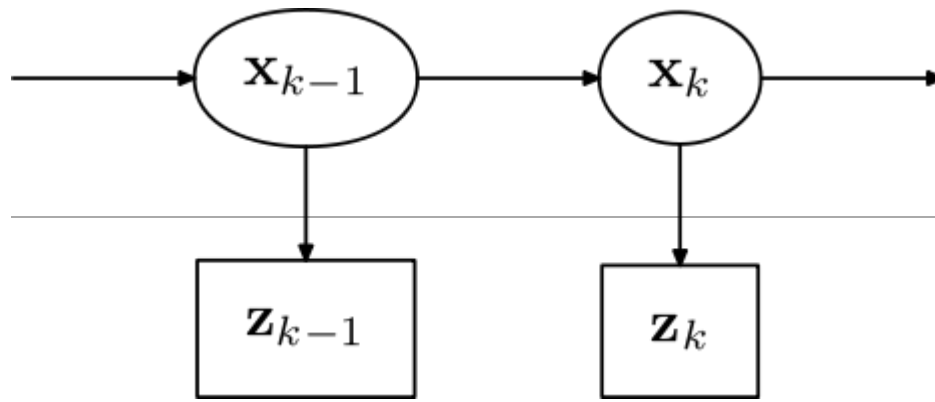
because of the Markov assumption, the true state is conditionally independent of all earlier states given the immediately previous state.

$$p(\mathbf{x}_k \mid \mathbf{x}_0, \ldots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

Similarly, the measurement at the k-th timestep is dependent only upon the current state and is conditionally independent of all other states given the current state.

$$p(\mathbf{z}_k \mid \mathbf{x}_0, \ldots, \mathbf{x}_k) = p(\mathbf{z}_k \mid \mathbf{x}_k)$$

Using these assumptions the probability distribution over all states of the hidden Markov model can be written simply as:

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_k, \mathbf{z}_1, \ldots, \mathbf{z}_k) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{z}_i \mid \mathbf{x}_i) p(\mathbf{x}_i \mid \mathbf{x}_{i-1})$$

However, when the Kalman filter is used to estimate the state $\mathbf{x}$, the probability distribution of interest is that associated with the current states conditioned on the measurements up to the current timestep. This is achieved by marginalizing out the previous states and dividing by the probability of the measurement set.

This leads to the *predict* and *update* steps of the Kalman filter written probabilistically. The probability distribution associated with the predicted state is the sum (integral) of the products of the probability distribution associated with the transition from the $(k − 1)$-th timestep to the k-th and the probability distribution associated with the previous state, over all possible Z_{k-1}.

$$p(\mathbf{x}_k \mid \mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1}) \, d\mathbf{x}_{k-1}$$

The measurement set up to time t is

$$\mathbf{Z}_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$$

The probability distribution of the update is proportional to the product of the measurement likelihood and the predicted state.

$$p(\mathbf{x}_k \mid \mathbf{Z}_k) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{Z}_{k-1})}{p(\mathbf{z}_k \mid \mathbf{Z}_{k-1})}$$

The denominator

$$p(\mathbf{z}_k \mid \mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{Z}_{k-1}) \, d\mathbf{x}_k$$

is a normalization term.

The remaining probability density functions are

$$p\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}\right) = \mathcal{N}\left(\mathbf{F}_k \mathbf{x}_{k-1}, \mathbf{Q}_k\right)$$
$$p\left(\mathbf{z}_k \mid \mathbf{x}_k\right) = \mathcal{N}\left(\mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k\right)$$
$$p\left(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1}\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}\right)$$

Note that the PDF at the previous timestep is inductively assumed to be the estimated state and covariance. This is justified because, as an optimal estimator, the Kalman filter makes best use of the measurements, therefore the PDF for $\mathbf{x}_k$ given the measurements $\mathbf{Z}_k$ is the Kalman filter estimate.

Marginal likelihood

Related to the recursive Bayesian interpretation described above, the Kalman filter can be viewed as a generative model, i.e., a process for *generating* a stream of random observations $\mathbf{z} = (\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, ...)$. Specifically, the process is

1. Sample a hidden state $\mathbf{x}_0$ from the Gaussian prior distribution $p\left(\mathbf{x}_0\right) = \mathcal{N}\left(\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0}\right)$.
2. Sample an observation $\mathbf{z}_0$ from the observation model $p\left(\mathbf{z}_0 \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{H}_0 \mathbf{x}_0, \mathbf{R}_0\right)$.
3. For $k = 1, 2, 3, \ldots$, do
    1. Sample the next hidden state $\mathbf{x}_k$ from the transition model $p\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}\right) = \mathcal{N}\left(\mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k, \mathbf{Q}_k\right)$.
    2. Sample an observation $\mathbf{z}_k$ from the observation model $p\left(\mathbf{z}_k \mid \mathbf{x}_k\right) = \mathcal{N}\left(\mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k\right)$.

Note that this process has identical structure to the hidden Markov model, except that the discrete state and observations are replaced with continuous variables sampled from Gaussian distributions.

In some applications, it is useful to compute the *probability* that a Kalman filter with a given set of parameters (prior distribution, transition and observation models, and control inputs) would generate a particular observed signal. This probability is known as the marginal likelihood because it integrates over ("marginalizes out") the values of the hidden state variables, so it can be computed using only the observed signal. The marginal likelihood can be useful to evaluate different parameter choices, or to compare the Kalman filter against other models using Bayesian model comparison.

It is straightforward to compute the marginal likelihood as a side effect of the recursive filtering computation. By the chain rule, the likelihood can be factored as the product of the probability of each observation given previous observations,

$$p(\mathbf{z}) = \prod_{k=0}^{T} p\left(\mathbf{z}_k \mid \mathbf{z}_{k-1}, \ldots, \mathbf{z}_0\right),$$

and because the Kalman filter describes a Markov process, all relevant information from previous observations is contained in the current state estimate $\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}$. Thus the marginal likelihood is given by

$$p(\mathbf{z}) = \prod_{k=0}^{T} \int p\left(\mathbf{z}_k \mid \mathbf{x}_k\right) p\left(\mathbf{x}_k \mid \mathbf{z}_{k-1}, \ldots, \mathbf{z}_0\right) d\mathbf{x}_k$$

$$= \prod_{k=0}^{T} \int \mathcal{N}\left(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k\right) \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right) d\mathbf{x}_k$$

$$= \prod_{k=0}^{T} \mathcal{N}\left(\mathbf{z}_k; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathsf{T}}\right)$$

$$= \prod_{k=0}^{T} \mathcal{N}\left(\mathbf{z}_k; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k\right),$$

i.e., a product of Gaussian densities, each corresponding to the density of one observation $\mathbf{z}_k$ under the current filtering distribution $\mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k$. This can easily be computed as a simple recursive update; however, to avoid numeric underflow, in a practical implementation it is usually desirable to compute the *log* marginal likelihood $\ell = \log p(\mathbf{z})$ instead. Adopting the convention $\ell^{(-1)} = 0$, this can be done via the recursive update rule

$$\ell^{(k)} = \ell^{(k-1)} - \frac{1}{2}\left(\tilde{\mathbf{y}}_k^{\mathsf{T}} \mathbf{S}_k^{-1} \tilde{\mathbf{y}}_k + \log|\mathbf{S}_k| + d_y \log 2\pi\right),$$

where d_y is the dimension of the measurement vector.[35]

An important application where such a (log) likelihood of the observations (given the filter parameters) is used is multi-target tracking. For example, consider an object tracking scenario where a stream of observations is the input, however, it is unknown how many objects are in the scene (or, the number of objects is known but is greater than one). In such a scenario, it can be unknown apriori which observations/measurements were generated by which object. A multiple hypothesis tracker (MHT) typically will form different track association hypotheses, where each hypothesis can be viewed as a Kalman filter (in the linear Gaussian case) with a specific set of parameters associated with the hypothesized object. Thus, it is important to compute the likelihood of the observations for the different hypotheses under consideration, such that the most-likely one can be found.

Information filter

In the information filter, or inverse covariance filter, the estimated covariance and estimated state are replaced by the information matrix and information vector respectively. These are defined as:

$$\mathbf{Y}_{k|k} = \mathbf{P}_{k|k}^{-1}$$
$$\hat{\mathbf{y}}_{k|k} = \mathbf{P}_{k|k}^{-1} \hat{\mathbf{x}}_{k|k}$$

Similarly the predicted covariance and state have equivalent information forms, defined as:

$$\mathbf{Y}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1}$$
$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1} \hat{\mathbf{x}}_{k|k-1}$$

as have the measurement covariance and measurement vector, which are defined as:

$$\mathbf{I}_k = \mathbf{H}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \mathbf{H}_k$$
$$\mathbf{i}_k = \mathbf{H}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \mathbf{z}_k$$

The information update now becomes a trivial sum.[36]

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \mathbf{I}_k$$
$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \mathbf{i}_k$$

The main advantage of the information filter is that N measurements can be filtered at each timestep simply by summing their information matrices and vectors.

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{j=1}^{N} \mathbf{I}_{k,j}$$

$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \sum_{j=1}^{N} \mathbf{i}_{k,j}$$

To predict the information filter the information matrix and vector can be converted back to their state space equivalents, or alternatively the information space prediction can be used.[36]

$$\mathbf{M}_k = \left[\mathbf{F}_k^{-1}\right]^{\mathsf{T}} \mathbf{Y}_{k-1|k-1} \mathbf{F}_k^{-1}$$
$$\mathbf{C}_k = \mathbf{M}_k \left[\mathbf{M}_k + \mathbf{Q}_k^{-1}\right]^{-1}$$
$$\mathbf{L}_k = \mathbf{I} - \mathbf{C}_k$$
$$\mathbf{Y}_{k|k-1} = \mathbf{L}_k \mathbf{M}_k \mathbf{L}_k^{\mathsf{T}} + \mathbf{C}_k \mathbf{Q}_k^{-1} \mathbf{C}_k^{\mathsf{T}}$$
$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{L}_k \left[\mathbf{F}_k^{-1}\right]^{\mathsf{T}} \hat{\mathbf{y}}_{k-1|k-1}$$

Note that if F and Q are time invariant these values can be cached. Note also that F and Q need to be invertible.

Fixed-lag smoother

The optimal fixed-lag smoother provides the optimal estimate of $\hat{\mathbf{x}}_{k-N|k}$ for a given fixed-lag N using the measurements from $\mathbf{z}_1$ to $\mathbf{z}_k$.[37] It can be derived using the previous theory via an augmented state, and the main equation of the filter is the following:

$$
\begin{bmatrix}
\hat{\mathbf{x}}_{t|t} \\
\hat{\mathbf{x}}_{t-1|t} \\
\vdots \\
\hat{\mathbf{x}}_{t-N+1|t}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{I} \\
0 \\
\vdots \\
0
\end{bmatrix}
\hat{\mathbf{x}}_{t|t-1}
+
\begin{bmatrix}
0 & \cdots & 0 \\
\mathbf{I} & 0 & \vdots \\
\vdots & \ddots & \vdots \\
0 & \cdots & \mathbf{I}
\end{bmatrix}
\begin{bmatrix}
\hat{\mathbf{x}}_{t-1|t-1} \\
\hat{\mathbf{x}}_{t-2|t-1} \\
\vdots \\
\hat{\mathbf{x}}_{t-N+1|t-1}
\end{bmatrix}
+
\begin{bmatrix}
\mathbf{K}^{(0)} \\
\mathbf{K}^{(1)} \\
\vdots \\
\mathbf{K}^{(N-1)}
\end{bmatrix}
\mathbf{y}_{t|t-1}
$$

where:

- $\hat{\mathbf{x}}_{t|t-1}$ is estimated via a standard Kalman filter;
- $\mathbf{y}_{t|t-1} = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}$ is the innovation produced considering the estimate of the standard Kalman filter;
- the various $\hat{\mathbf{x}}_{t-i|t}$ with $i = 1, \ldots, N-1$ are new variables; i.e., they do not appear in the standard Kalman filter;
- the gains are computed via the following scheme:

$$\mathbf{K}^{(i+1)} = \mathbf{P}^{(i)} \mathbf{H}^{\mathsf{T}} \left[\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\right]^{-1}$$

and

$$\mathbf{P}^{(i)} = \mathbf{P}\left[(\mathbf{F} - \mathbf{KH})^{\mathsf{T}}\right]^{i}$$

where $\mathbf{P}$ and $\mathbf{K}$ are the prediction error covariance and the gains of the standard Kalman filter (i.e., $\mathbf{P}_{t|t-1}$).

If the estimation error covariance is defined so that

$$\mathbf{P}_i := E\left[\left(\mathbf{x}_{t-i} - \hat{\mathbf{x}}_{t-i|t}\right)^{*}\left(\mathbf{x}_{t-i} - \hat{\mathbf{x}}_{t-i|t}\right) \mid z_1 \ldots z_t\right],$$

then we have that the improvement on the estimation of $\mathbf{x}_{t-i}$ is given by:

$$\mathbf{P} - \mathbf{P}_i = \sum_{j=0}^{i}\left[\mathbf{P}^{(j)}\mathbf{H}^{\mathsf{T}}\left(\mathbf{HPH}^{\mathsf{T}} + \mathbf{R}\right)^{-1}\mathbf{H}\left(\mathbf{P}^{(i)}\right)^{\mathsf{T}}\right]$$

Fixed-interval smoothers

The optimal fixed-interval smoother provides the optimal estimate of $\hat{\mathbf{x}}_{k|n}$ ($k < n$) using the measurements from a fixed interval $\mathbf{z}_1$ to $\mathbf{z}_n$. This is also called "Kalman Smoothing". There are several smoothing algorithms in common use.

Rauch–Tung–Striebel

The Rauch–Tung–Striebel (RTS) smoother is an efficient two-pass algorithm for fixed interval smoothing.[38]

The forward pass is the same as the regular Kalman filter algorithm. These *filtered* a-priori and a-posteriori state estimates $\hat{\mathbf{x}}_{k|k-1}$, $\hat{\mathbf{x}}_{k|k}$ and covariances $\mathbf{P}_{k|k-1}$, $\mathbf{P}_{k|k}$ are saved for use in the backwards pass.

In the backwards pass, we compute the *smoothed* state estimates $\hat{\mathbf{x}}_{k|n}$ and covariances $\mathbf{P}_{k|n}$. We start at the last time step and proceed backwards in time using the following recursive equations:

$$\hat{\mathbf{x}}_{k|n} = \hat{\mathbf{x}}_{k|k} + \mathbf{C}_k\left(\hat{\mathbf{x}}_{k+1|n} - \hat{\mathbf{x}}_{k+1|k}\right)$$
$$\mathbf{P}_{k|n} = \mathbf{P}_{k|k} + \mathbf{C}_k\left(\mathbf{P}_{k+1|n} - \mathbf{P}_{k+1|k}\right)\mathbf{C}_k^{\mathsf{T}}$$

where

$$\mathbf{C}_k = \mathbf{P}_{k|k}\mathbf{F}_{k+1}^{\mathsf{T}}\mathbf{P}_{k+1|k}^{-1}.$$

Note that $\mathbf{x}_{k|k}$ is the a-posteriori state estimate of timestep k and $\mathbf{x}_{k+1|k}$ is the a-priori state estimate of timestep $k+1$. The same notation applies to the covariance.

Modified Bryson–Frazier smoother

An alternative to the RTS algorithm is the modified Bryson–Frazier (MBF) fixed interval smoother developed by Bierman.[30] This also uses a backward pass that processes data saved from the Kalman filter forward pass. The equations for the backward pass involve the recursive computation of data which are used at each observation time to compute the smoothed state and covariance.

The recursive equations are

$$\tilde{\Lambda}_k = \mathbf{H}_k^\mathsf{T}\mathbf{S}_k^{-1}\mathbf{H}_k + \hat{\mathbf{C}}_k^\mathsf{T}\hat{\Lambda}_k\hat{\mathbf{C}}_k$$

$$\hat{\Lambda}_{k-1} = \mathbf{F}_k^\mathsf{T}\tilde{\Lambda}_k\mathbf{F}_k$$

$$\hat{\Lambda}_n = 0$$

$$\tilde{\lambda}_k = -\mathbf{H}_k^\mathsf{T}\mathbf{S}_k^{-1}\mathbf{y}_k + \hat{\mathbf{C}}_k^\mathsf{T}\hat{\lambda}_k$$

$$\hat{\lambda}_{k-1} = \mathbf{F}_k^\mathsf{T}\tilde{\lambda}_k$$

$$\hat{\lambda}_n = 0$$

where $\mathbf{S}_k$ is the residual covariance and $\hat{\mathbf{C}}_k = \mathbf{I} - \mathbf{K}_k\mathbf{H}_k$. The smoothed state and covariance can then be found by substitution in the equations

$$\mathbf{P}_{k|n} = \mathbf{P}_{k|k} - \mathbf{P}_{k|k}\hat{\Lambda}_k\mathbf{P}_{k|k}$$

$$\mathbf{x}_{k|n} = \mathbf{x}_{k|k} - \mathbf{P}_{k|k}\hat{\lambda}_k$$

or

$$\mathbf{P}_{k|n} = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\tilde{\Lambda}_k\mathbf{P}_{k|k-1}$$

$$\mathbf{x}_{k|n} = \mathbf{x}_{k|k-1} - \mathbf{P}_{k|k-1}\tilde{\lambda}_k.$$

An important advantage of the MBF is that it does not require finding the inverse of the covariance matrix.

Minimum-variance smoother

The minimum-variance smoother can attain the best-possible error performance, provided that the models are linear, their parameters and the noise statistics are known precisely.[39] This smoother is a time-varying state-space generalization of the optimal non-causal Wiener filter.

The smoother calculations are done in two passes. The forward calculations involve a one-step-ahead predictor and are given by

$$\hat{\mathbf{x}}_{k+1|k} = (\mathbf{F}_k - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{z}_k$$

$$\alpha_k = -\mathbf{S}_k^{-\frac{1}{2}}\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} + \mathbf{S}_k^{-\frac{1}{2}}\mathbf{z}_k$$

The above system is known as the inverse Wiener-Hopf factor. The backward recursion is the adjoint of the above forward system. The result of the backward pass β_k may be calculated by operating the forward equations on the time-reversed α_k and time reversing the result. In the case of output estimation, the smoothed estimate is given by

$$\hat{\mathbf{y}}_{k|N} = \mathbf{z}_k - \mathbf{R}_k\beta_k$$

Taking the causal part of this minimum-variance smoother yields

$$\hat{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{R}_k\mathbf{S}_k^{-\frac{1}{2}}\alpha_k$$

which is identical to the minimum-variance Kalman filter. The above solutions minimize the variance of the output estimation error. Note that the Rauch–Tung–Striebel smoother derivation assumes that the underlying distributions are Gaussian, whereas the minimum-variance solutions do not. Optimal smoothers for state estimation and input estimation can be constructed similarly.

A continuous-time version of the above smoother is described in.[40][41]

Expectation-maximization algorithms may be employed to calculate approximate maximum likelihood estimates of unknown state-space parameters within minimum-variance filters and smoothers. Often uncertainties remain within problem assumptions. A smoother that accommodates uncertainties can be designed by adding a positive definite term to the Riccati equation.[42]

In cases where the models are nonlinear, step-wise linearizations may be within the minimum-variance filter and smoother recursions (extended Kalman filtering).

Frequency-weighted Kalman filters

Pioneering research on the perception of sounds at different frequencies was conducted by Fletcher and Munson in the 1930s. Their work led to a standard way of weighting measured sound levels within investigations of industrial noise and hearing loss. Frequency weightings have since been used within filter and controller designs to manage performance within bands of interest.

Typically, a frequency shaping function is used to weight the average power of the error spectral density in a specified frequency band. Let $\mathbf{y} - \hat{\mathbf{y}}$ denote the output estimation error exhibited by a conventional Kalman filter. Also, let $\mathbf{W}$ denote a causal frequency weighting transfer function. The optimum solution which minimizes the variance of $\mathbf{W}(\mathbf{y} - \hat{\mathbf{y}})$ arises by simply constructing $\mathbf{W}^{-1}\hat{\mathbf{y}}$.

The design of $\mathbf{W}$ remains an open question. One way of proceeding is to identify a system which generates the estimation error and setting $\mathbf{W}$ equal to the inverse of that system.[43] This procedure may be iterated to obtain mean-square error improvement at the cost of increased filter order. The same technique can be applied to smoothers.

Non-linear filters

The basic Kalman filter is limited to a linear assumption. More complex systems, however, can be nonlinear. The non-linearity can be associated either with the process model or with the observation model or with both.

Extended Kalman filter

In the extended Kalman filter (EKF), the state transition and observation models need not be linear functions of the state but may instead be non-linear functions. These functions are of differentiable type.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k$$
$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed.

At each timestep the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

Unscented Kalman filter

When the state transition and observation models—that is, the predict and update functions f and h—are highly non-linear, the extended Kalman filter can give particularly poor performance.[44] This is because the covariance is propagated through linearization of the underlying non-linear model. The unscented Kalman filter (UKF) [44] uses a deterministic sampling technique known as the

unscented transformation (UT) to pick a minimal set of sample points (called sigma points) around the mean. The sigma points are then propagated through the non-linear functions, from which a new mean and covariance estimate are then formed. The resulting filter depends on how the transformed statistics of the UT are calculated and which set of sigma points are used—it should be remarked that is always possible to construct new UKFs in a consistent way.[45] For certain systems, the resulting UKF more accurately estimates the true mean and covariance.[46] This can be verified with Monte Carlo sampling or Taylor series expansion of the posterior statistics. In addition, this technique removes the requirement to explicitly calculate Jacobians, which for complex functions can be a difficult task in itself (i.e., requiring complicated derivatives if done analytically or being computationally costly if done numerically), if not impossible (if those functions are not differentiable).

Sigma points

For a random vector $\mathbf{x} = (x_1, \ldots, x_L)$, sigma points are any set of vectors

$$\{\mathbf{s}_0, \ldots, \mathbf{s}_N\} = \left\{ \begin{pmatrix} s_{0,1} & s_{0,2} & \cdots & s_{0,L} \end{pmatrix}, \ldots, \begin{pmatrix} s_{N,1} & s_{N,2} & \cdots & s_{N,L} \end{pmatrix} \right\}$$

attributed with

- first-order weights $W_0^a, \ldots, W_N^a$ that fulfill

1. $\displaystyle\sum_{j=0}^{N} W_j^a = 1$

2. for all $i = 1, \ldots, L$: $E[x_i] = \displaystyle\sum_{j=0}^{N} W_j^a s_{j,i}$

- second-order weights $W_0^c, \ldots, W_N^c$ that fulfill

1. $\displaystyle\sum_{j=0}^{N} W_j^c = 1$

2. for all pairs $(i, l) \in \{1, \ldots, L\}^2 : E[x_i x_l] = \displaystyle\sum_{j=0}^{N} W_j^c s_{j,i} s_{j,l}$.

A simple choice of sigma points and weights for $\mathbf{x}_{k-1|k-1}$ in the UKF algorithm is

$$\mathbf{s}_0 = \hat{\mathbf{x}}_{k-1|k-1}$$
$$-1 < W_0^a = W_0^c < 1$$
$$\mathbf{s}_j = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{\frac{L}{1 - W_0}} \mathbf{A}_j, \quad j = 1, \ldots, L$$
$$\mathbf{s}_{L+j} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{\frac{L}{1 - W_0}} \mathbf{A}_j, \quad j = 1, \ldots, L$$
$$W_j^a = W_j^c = \frac{1 - W_0}{2L}, \quad j = 1, \ldots, 2L$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the mean estimate of $\mathbf{x}_{k-1|k-1}$. The vector $\mathbf{A}_j$ is the jth column of $\mathbf{A}$ where $\mathbf{P}_{k-1|k-1} = \mathbf{A}\mathbf{A}^\mathsf{T}$. The matrix $\mathbf{A}$ should be calculated using numerically efficient and stable methods such as the Cholesky decomposition. The weight of the mean value, W_0, can be chosen arbitrarily.

Another popular parameterization (which generalizes the above) is

$$\mathbf{s}_0 = \hat{\mathbf{x}}_{k-1|k-1}$$

$$W_0^a = \frac{\alpha^2\kappa - L}{\alpha^2\kappa}$$

$$W_0^c = W_0^a + 1 - \alpha^2 + \beta$$

$$\mathbf{s}_j = \hat{\mathbf{x}}_{k-1|k-1} + \alpha\sqrt{\kappa}\mathbf{A}_j, \quad j = 1, \ldots, L$$

$$\mathbf{s}_{L+j} = \hat{\mathbf{x}}_{k-1|k-1} - \alpha\sqrt{\kappa}\mathbf{A}_j, \quad j = 1, \ldots, L$$

$$W_j^a = W_j^c = \frac{1}{2\alpha^2\kappa}, \quad j = 1, \ldots, 2L.$$

α and κ control the spread of the sigma points. β is related to the distribution of x.

Appropriate values depend on the problem at hand, but a typical recommendation is $\alpha = 10^{-3}$, $\kappa = 1$, and $\beta = 2$. However, a larger value of α (e.g., $\alpha = 1$) may be beneficial in order to better capture the spread of the distribution and possible non-linearities.[47] If the true distribution of x is Gaussian, $\beta = 2$ is optimal.[48]

Predict

As with the EKF, the UKF prediction can be used independently from the UKF update, in combination with a linear (or indeed EKF) update, or vice versa.

Given estimates of the mean and covariance, $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$, one obtains $N+1$ sigma points as described in the section above. The sigma points are propagated through the transition function f.

$$\mathbf{z}_j = f(\mathbf{s}_j) \quad j = 0, \ldots, N.$$

The propagated sigma points are weighed to produce the predicted mean and covariance.

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{j=0}^{N} W_j^a \mathbf{z}_j$$

$$\mathbf{P}_{k|k-1} = \sum_{j=0}^{N} W_j^c \left(\mathbf{z}_j - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathbf{z}_j - \hat{\mathbf{x}}_{k|k-1}\right)^{\mathsf{T}} + \mathbf{Q}_k$$

where W_j^a are the first-order weights of the original sigma points, and W_j^c are the second-order weights. The matrix $\mathbf{Q}_k$ is the covariance of the transition noise, $\mathbf{w}_k$.

Update

Given prediction estimates $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, a new set of $N+1$ sigma points $\mathbf{s}_0, \ldots, \mathbf{s}_N$ with corresponding first-order weights $W_0^a, \ldots W_N^a$ and second-order weights $W_0^c, \ldots, W_N^c$ is calculated.[49] These sigma points are transformed through h.

$$\mathbf{z}_j = h(\mathbf{s}_j), \quad j = 0, 1, \ldots, N.$$

Then the empirical mean and covariance of the transformed points are calculated.

$$\hat{\mathbf{z}} = \sum_{j=0}^{N} W_j^a \mathbf{z}_j$$

$$\hat{\mathbf{S}}_k = \sum_{j=0}^{N} W_j^c (\mathbf{z}_j - \hat{\mathbf{z}})(\mathbf{z}_j - \hat{\mathbf{z}})^\mathsf{T} + \mathbf{R}_k$$

where $\mathbf{R}_k$ is the covariance matrix of the observation noise, $\mathbf{v}_k$. Additionally, the cross covariance matrix is also needed

$$\mathbf{C}_{\mathbf{sz}} = \sum_{j=0}^{N} W_j^c (\mathbf{z}_j - \hat{\mathbf{z}})(\mathbf{s}_j - \hat{\mathbf{x}}_{k|k-1})^\mathsf{T}$$

where $\mathbf{s}_j$ are the untransformed sigma points created from $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$.

The Kalman gain is

$$\mathbf{K}_k = \mathbf{C}_{\mathbf{sz}} \hat{\mathbf{S}}_k^{-1}.$$

The updated mean and covariance estimates are

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}})$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \hat{\mathbf{S}}_k \mathbf{K}_k^\mathsf{T}.$$

Kalman–Bucy filter

The Kalman–Bucy filter (named after Richard Snowden Bucy) is a continuous time version of the Kalman filter.[50][51]

It is based on the state space model

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t)$$
$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t)$$

where $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ represent the intensities (or, more accurately: the Power Spectral Density - PSD - matrices) of the two white noise terms $\mathbf{w}(t)$ and $\mathbf{v}(t)$, respectively.

The filter consists of two differential equations, one for the state estimate and one for the covariance:

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{K}(t)\left(\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)\right)$$
$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^\mathsf{T}(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^\mathsf{T}(t)$$

where the Kalman gain is given by

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^\mathsf{T}(t)\mathbf{R}^{-1}(t)$$

Note that in this expression for $\mathbf{K}(t)$ the covariance of the observation noise $\mathbf{R}(t)$ represents at the same time the covariance of the prediction error (or *innovation*) $\tilde{\mathbf{y}}(t) = \mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)$; these covariances are equal only in the case of continuous time.[52]

The distinction between the prediction and update steps of discrete-time Kalman filtering does not exist in continuous time.

The second differential equation, for the covariance, is an example of a Riccati equation.

Hybrid Kalman filter

Most physical systems are represented as continuous-time models while discrete-time measurements are frequently taken for state estimation via a digital processor. Therefore, the system model and measurement model are given by

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t), \qquad \mathbf{w}(t) \sim N\left(\mathbf{0}, \mathbf{Q}(t)\right)$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \qquad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$$

where

$$\mathbf{x}_k = \mathbf{x}(t_k).$$

Initialize

$$\hat{\mathbf{x}}_{0|0} = E\left[\mathbf{x}(t_0)\right], \mathbf{P}_{0|0} = \mathrm{Var}[\mathbf{x}(t_0)]$$

Predict

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t), \text{ with } \hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1|k-1}$$
$$\Rightarrow \hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}(t_k)$$
$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^{\mathsf{T}} + \mathbf{Q}(t), \text{ with } \mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1|k-1}$$
$$\Rightarrow \mathbf{P}_{k|k-1} = \mathbf{P}(t_k)$$

The prediction equations are derived from those of continuous-time Kalman filter without update from measurements, i.e., $\mathbf{K}(t) = \mathbf{0}$. The predicted state and covariance are calculated respectively by solving a set of differential equations with the initial value equal to the estimate at the previous step.

Update

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathsf{T}}\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k\right)^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right)$$
$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1}$$

The update equations are identical to those of the discrete-time Kalman filter.

Variants for the recovery of sparse signals

The traditional Kalman filter has also been employed for the recovery of sparse, possibly dynamic, signals from noisy observations. Recent works[53][54][55] utilize notions from the theory of compressed sensing/sampling, such as the restricted isometry property and related probabilistic recovery arguments, for sequentially estimating the sparse state in intrinsically low-dimensional systems.

Applications

- Attitude and heading reference systems
- Autopilot
- Battery state of charge (SoC) estimation[56][57]
- Brain-computer interface
- Chaotic signals
- Tracking and vertex fitting of charged particles in particle detectors[58]
- Tracking of objects in computer vision
- Dynamic positioning
- Economics, in particular macroeconomics, time series analysis, and econometrics[59]
- Inertial guidance system
- Nuclear medicine – single photon emission computed tomography image restoration[60]
- Orbit Determination

- Power system state estimation
- Radar tracker
- Satellite navigation systems
- Seismology[61]
- Sensorless control of AC motor variable-frequency drives
- Simultaneous localization and mapping
- Speech enhancement
- Visual odometry
- Weather forecasting
- Navigation system
- 3D modeling
- Structural health monitoring
- Human sensorimotor processing[62]

See also

- Alpha beta filter
- Covariance intersection
- Ensemble Kalman filter
- Fast Kalman filter
- Filtering problem (stochastic processes)
- Generalized filtering
- Invariant extended Kalman filter
- Kernel adaptive filter
- Masreliez's theorem
- Moving horizon estimation
- Particle filter estimator
- PID controller
- Predictor–corrector method
- Recursive least squares filter
- Schmidt–Kalman filter
- Separation principle
- Sliding mode control
- Stochastic differential equations
- Switching Kalman filter

References

1. Paul Zarchan; Howard Musoff (2000). *Fundamentals of Kalman Filtering: A Practical Approach* (https://books.google. com/books?id=AQxRAAAAMAAJ). American Institute of Aeronautics and Astronautics, Incorporated. ISBN 978-1-56347-455-2.

2. Wolpert, Daniel; Ghahramani, Zoubin (2000). "Computational principles of movement neuroscience". *Nature Neuroscience*. **3**: 1212–7. doi:10.1038/81497 (https://doi.org/10.1038%2F81497). PMID 11127840 (https://www.ncbi. nlm.nih.gov/pubmed/11127840).

3. Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering*. **82**: 35. doi:10.1115/1.3662552 (https://doi.org/10.1115%2F1.3662552).

4. Lauritzen, S. L. (December 1981). "Time series analysis in 1880. A discussion of contributions made by T.N. Thiele". *International Statistical Review*. **49** (3): 319–331. doi:10.2307/1402616 (https://doi.org/10.2307%2F1402616). JSTOR 1402616 (https://www.jstor.org/stable/1402616). "He derives a recursive procedure for estimating the regression component and predicting the Brownian motion. The procedure is now known as Kalman filtering."

5. Lauritzen, S. L. (2002). *Thiele: Pioneer in Statistics* (https://books.google.com/books?id=irugmNUwuG4C&q=kalman#v=snippet&q=kalman&f=false). New York: Oxford University Press. p. 41. ISBN 978-0-19-850972-1. "He solves the problem of estimating the regression coefficients and predicting the values of the Brownian motion by the method of least squares and gives an elegant recursive procedure for carrying out the calculations. The procedure is nowadays known as *Kalman filtering*."

6. Mohinder S. Grewal and Angus P. Andrews (http://ieeecss.org/CSM/library/2010/june10/11-HistoricalPerspectives.pdf)

7. Gaylor, David; Lightsey, E. Glenn (2003). "GPS/INS Kalman Filter Design for Spacecraft Operating in the Proximity of International Space Station". *AIAA Guidance, Navigation, and Control Conference and Exhibit*. doi:10.2514/6.2003-5445 (https://doi.org/10.2514%2F6.2003-5445). ISBN 978-1-62410-090-1.

8. Stratonovich, R. L. (1959). *Optimum nonlinear systems which bring about a separation of a signal with constant parameters from noise*. Radiofizika, 2:6, pp. 892–901.

9. Stratonovich, R. L. (1959). *On the theory of optimal non-linear filtering of random functions*. Theory of Probability and its Applications, 4, pp. 223–225.

10. Stratonovich, R. L. (1960) *Application of the Markov processes theory to optimal filtering*. Radio Engineering and Electronic Physics, 5:11, pp. 1–19.

11. Stratonovich, R. L. (1960). *Conditional Markov Processes*. Theory of Probability and its Applications, 5, pp. 156–178.

12. Stepanov, O. A. (15 May 2011). "Kalman filtering: Past and present. An outlook from Russia. (On the occasion of the 80th birthday of Rudolf Emil Kalman)". *Gyroscopy and Navigation*. **2** (2): 105. doi:10.1134/S2075108711020076 (https://doi.org/10.1134%2FS2075108711020076).

13. Ingvar Strid; Karl Walentin (April 2009). "Block Kalman Filtering for Large-Scale DSGE Models" (http://www.riksbank.se/en/Press-and-published/Published-from-the-Riksbank/Other-reports/Working-Paper-Series/2008/No-224-Block-Kalman-filtering-for-large-scale-DSGE-models/). *Computational Economics*. **33** (3): 277–304. CiteSeerX 10.1.1.232.3790 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.232.3790). doi:10.1007/s10614-008-9160-4 (https://doi.org/10.1007%2Fs10614-008-9160-4).

14. Martin Møller Andreasen (2008). "Non-linear DSGE Models, The Central Difference Kalman Filter, and The Mean Shifted Particle Filter" (ftp://ftp.econ.au.dk/creates/rp/08/rp08_33.pdf) (PDF).

15. Roweis, S; Ghahramani, Z (1999). "A unifying review of linear gaussian models" (https://authors.library.caltech.edu/13697/1/ROWnc99.pdf) (PDF). *Neural Computation*. **11** (2): 305–45. doi:10.1162/089976699300016674 (https://doi.org/10.1162%2F089976699300016674). PMID 9950734 (https://www.ncbi.nlm.nih.gov/pubmed/9950734).

16. Hamilton, J. (1994), *Time Series Analysis*, Princeton University Press. Chapter 13, 'The Kalman Filter'

17. Ishihara, J.Y.; Terra, M.H.; Campos, J.C.T. (2006). "Robust Kalman Filter for Descriptor Systems". *IEEE Transactions on Automatic Control*. **51** (8): 1354. doi:10.1109/TAC.2006.878741 (https://doi.org/10.1109%2FTAC.2006.878741).

18. Terra, Marco H.; Cerri, Joao P.; Ishihara, Joao Y. (2014). "Optimal Robust Linear Quadratic Regulator for Systems Subject to Uncertainties". *IEEE Transactions on Automatic Control*. **59** (9): 2586–2591. doi:10.1109/TAC.2014.2309282 (https://doi.org/10.1109%2FTAC.2014.2309282).

19. Kelly, Alonzo (1994). "A 3D state space formulation of a navigation Kalman filter for autonomous vehicles" (http://www.dtic.mil/dtic/tr/fulltext/u2/a282853.pdf) (PDF). *DTIC Document*: 13. 2006 Corrected Version (http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/kalman_V2.pdf)

20. Reid, Ian; Term, Hilary. "Estimation II" (http://www.robots.ox.ac.uk/~ian/Teaching/Estimation/LectureNotes2.pdf) (PDF). *www.robots.ox.ac.uk*. Oxford University. Retrieved 6 August 2014.

21. Rajamani, Murali (October 2007). *Data-based Techniques to Improve State Estimation in Model Predictive Control* (http://jbrwww.che.wisc.edu/theses/rajamani.pdf) (PDF) (PhD Thesis). University of Wisconsin–Madison.

22. Rajamani, Murali R.; Rawlings, James B. (2009). "Estimation of the disturbance structure from data using semidefinite programming and optimal weighting". *Automatica*. **45** (1): 142–148. doi:10.1016/j.automatica.2008.05.032 (https://doi.org/10.1016%2Fj.automatica.2008.05.032).

23. "Autocovariance Least-Squares Toolbox" (http://jbrwww.che.wisc.edu/software/als/). Jbrwww.che.wisc.edu. Retrieved 2014-06-02.

24. Three optimality tests with numerical examples are described in Peter, Matisko (2012). "Optimality Tests and Adaptive Kalman Filter". *16th IFAC Symposium on System Identification*. *Ifac Proceedings Volumes*. 16th IFAC Symposium on System Identification. **45**. pp. 1523–1528. doi:10.3182/20120711-3-BE-2027.00011 (https://doi.org/10.3182%2F2012 0711-3-BE-2027.00011). ISBN 978-3-902823-06-9.

25. Spall, James C. (1995). "The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions". *Automatica*. **31** (10): 1513–1517. doi:10.1016/0005-1098(95)00069-9 (https://doi.org/10.1016%2F0005-1098%2895%2900069-9).

26. Maryak, J.L.; Spall, J.C.; Heydon, B.D. (2004). "Use of the Kalman Filter for Inference in State-Space Models with Unknown Noise Distributions". *IEEE Transactions on Automatic Control*. **49**: 87–90. doi:10.1109/TAC.2003.821415 (https://doi.org/10.1109%2FTAC.2003.821415).

27. Anderson, Brian D. O.; Moore, John B. (1979). *Optimal Filtering*. New York: Prentice Hall. pp. 129–133. ISBN 978-0-13-638122-8.

28. Jingyang Lu. "False information injection attack on dynamic state estimation in multi-sensor systems" (http://ieeexplore.ieee.org/abstract/document/6916211/,), Fusion 2014

29. Thornton, Catherine L. (15 October 1976). "Triangular Covariance Factorizations for Kalman Filtering" (https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770005172_1977005172.pdf) (PDF). (PhD thesis). NASA. NASA Technical Memorandum 33-798.

30. Bierman, G.J. (1977). "Factorization Methods for Discrete Sequential Estimation". *Factorization Methods for Discrete Sequential Estimation*. Bibcode:1977fmds.book.....B (http://adsabs.harvard.edu/abs/1977fmds.book.....B).

31. Bar-Shalom, Yaakov; Li, X. Rong; Kirubarajan, Thiagalingam (July 2001). *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons. pp. 308–317. ISBN 978-0-471-41655-5.

32. Golub, Gene H.; Van Loan, Charles F. (1996). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences (Third ed.). Baltimore, Maryland: Johns Hopkins University. p. 139. ISBN 978-0-8018-5414-9.

33. Higham, Nicholas J. (2002). *Accuracy and Stability of Numerical Algorithms* (Second ed.). Philadelphia, PA: Society for Industrial and Applied Mathematics. p. 680. ISBN 978-0-89871-521-7.

34. Masreliez, C. Johan; Martin, R D (1977). "Robust Bayesian estimation for the linear model and robustifying the Kalman filter". *IEEE Transactions on Automatic Control*. **22** (3): 361–371. doi:10.1109/TAC.1977.1101538 (https://doi.org/10.1109%2FTAC.1977.1101538).

35. Lütkepohl, Helmut (1991). *Introduction to Multiple Time Series Analysis*. Heidelberg: Springer-Verlag Berlin. p. 435.

36. Gabriel T. Terejanu (2012-08-04). "Discrete Kalman Filter Tutorial" (https://cse.sc.edu/~terejanu/files/tutorialKF.pdf) (PDF). Retrieved 2016-04-13.

37. Anderson, Brian D. O.; Moore, John B. (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, Inc. pp. 176–190. ISBN 978-0-13-638122-8.

38. Rauch, H.E.; Tung, F.; Striebel, C. T. (August 1965). "Maximum likelihood estimates of linear dynamic systems". *AIAA Journal*. **3** (8): 1445–1450. Bibcode:1965AIAAJ...3.1445. (http://adsabs.harvard.edu/abs/1965AIAAJ...3.1445.). doi:10.2514/3.3166 (https://doi.org/10.2514%2F3.3166).

39. Einicke, G.A. (March 2006). "Optimal and Robust Noncausal Filter Formulations". *IEEE Trans. Signal Process*. **54** (3): 1069–1077. Bibcode:2006ITSP...54.1069E (http://adsabs.harvard.edu/abs/2006ITSP...54.1069E). doi:10.1109/TSP.2005.863042 (https://doi.org/10.1109%2FTSP.2005.863042).

40. Einicke, G.A. (April 2007). "Asymptotic Optimality of the Minimum-Variance Fixed-Interval Smoother". *IEEE Trans. Signal Process*. **55** (4): 1543–1547. Bibcode:2007ITSP...55.1543E (http://adsabs.harvard.edu/abs/2007ITSP...55.1543E). doi:10.1109/TSP.2006.889402 (https://doi.org/10.1109%2FTSP.2006.889402).

41. Einicke, G.A.; Ralston, J.C.; Hargrave, C.O.; Reid, D.C.; Hainsworth, D.W. (December 2008). "Longwall Mining Automation. An Application of Minimum-Variance Smoothing". *IEEE Control Systems Magazine*. **28** (6): 28–37. doi:10.1109/MCS.2008.929281 (https://doi.org/10.1109%2FMCS.2008.929281).

42. Einicke, G.A. (December 2009). "Asymptotic Optimality of the Minimum-Variance Fixed-Interval Smoother". *IEEE Transactions on Automatic Control*. **54** (12): 2904–2908. Bibcode:2007ITSP...55.1543E (http://adsabs.harvard.edu/abs/2007ITSP...55.1543E). doi:10.1109/TSP.2006.889402 (https://doi.org/10.1109%2FTSP.2006.889402).

43. Einicke, G.A. (December 2014). "Iterative Frequency-Weighted Filtering and Smoothing Procedures". *IEEE Signal Processing Letters*. **21** (12): 1467–1470. Bibcode:2014ISPL...21.1467E (http://adsabs.harvard.edu/abs/2014ISPL...21.1467E). doi:10.1109/LSP.2014.2341641 (https://doi.org/10.1109%2FLSP.2014.2341641).

44. Julier, Simon J.; Uhlmann, Jeffrey K. (1997). "A new extension of the Kalman filter to nonlinear systems" (http://www.cs.unc.edu/~welch/kalman/media/pdf/Julier1997_SPIE_KF.pdf) (PDF). *Int. Symp. Aerospace/Defense Sensing, Simul. And Controls*. Signal Processing, Sensor Fusion, and Target Recognition VI. **3**: 182. Bibcode:1997SPIE.3068..182J (http://adsabs.harvard.edu/abs/1997SPIE.3068..182J). CiteSeerX 10.1.1.5.2891 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.2891). doi:10.1117/12.280797 (https://doi.org/10.1117%2F12.280797). Retrieved 2008-05-03.

45. Menegaz, H. M. T.; Ishihara, J. Y.; Borges, G. A.; Vargas, A. N. (October 2015). "A Systematization of the Unscented Kalman Filter Theory". *IEEE Transactions on Automatic Control*. **60** (10): 2583–2598. doi:10.1109/tac.2015.2404511 (https://doi.org/10.1109%2Ftac.2015.2404511). hdl:20.500.11824/251 (https://hdl.handle.net/20.500.11824%2F251). ISSN 0018-9286 (https://www.worldcat.org/issn/0018-9286).

46. Gustafsson, Fredrik; Hendeby, Gustaf (2012). "Some Relations Between Extended and Unscented Kalman Filters" (http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-75272). *IEEE Transactions on Signal Processing*. **60** (2): 545–555. Bibcode:2012ITSP...60..545G (http://adsabs.harvard.edu/abs/2012ITSP...60..545G). doi:10.1109/tsp.2011.2172431 (https://doi.org/10.1109%2Ftsp.2011.2172431).

47. Bitzer, S. (2016). "The UKF exposed: How it works, when it works and when it's better to sample" (https://github.com/sbitzer/UKF-exposed). doi:10.5281/zenodo.44386 (https://doi.org/10.5281%2Fzenodo.44386).

48. Wan, E.A.; Van Der Merwe, R. (2000). "The unscented Kalman filter for nonlinear estimation" (http://www.lara.unb.br/~gaborges/disciplinas/efe/papers/wan2000.pdf) (PDF). *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. p. 153. CiteSeerX 10.1.1.361.9373 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.361.9373). doi:10.1109/ASSPCC.2000.882463 (https://doi.org/10.1109%2FASSPCC.2000.882463). ISBN 978-0-7803-5800-3.

49. Sarkka, Simo (September 2007). "On Unscented Kalman Filtering for State Estimation of Continuous-Time Nonlinear Systems". *IEEE Transactions on Automatic Control*. **52** (9): 1631–1641. doi:10.1109/TAC.2007.904453 (https://doi.org/10.1109%2FTAC.2007.904453).

50. Bucy, R.S. and Joseph, P.D., *Filtering for Stochastic Processes with Applications to Guidance,* John Wiley & Sons, 1968; 2nd Edition, AMS Chelsea Publ., 2005. ISBN 0-8218-3782-6

51. Jazwinski, Andrew H., *Stochastic processes and filtering theory,* Academic Press, New York, 1970. ISBN 0-12-381550-9

52. Kailath, T. (1968). "An innovations approach to least-squares estimation--Part I: Linear filtering in additive white noise". *IEEE Transactions on Automatic Control*. **13** (6): 646–655. doi:10.1109/TAC.1968.1099025 (https://doi.org/10.1109%2FTAC.1968.1099025).

53. Vaswani, Namrata (2008). "Kalman filtered Compressed Sensing". *2008 15th IEEE International Conference on Image Processing*. pp. 893–896. arXiv:0804.0819 (https://arxiv.org/abs/0804.0819). doi:10.1109/ICIP.2008.4711899 (https://doi.org/10.1109%2FICIP.2008.4711899). ISBN 978-1-4244-1765-0.

54. Carmi, Avishy; Gurfil, Pini; Kanevsky, Dimitri (2010). "Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms". *IEEE Transactions on Signal Processing*. **58** (4): 2405–2409. Bibcode:2010ITSP...58.2405C (http://adsabs.harvard.edu/abs/2010ITSP...58.2405C). doi:10.1109/TSP.2009.2038959 (https://doi.org/10.1109%2FTSP.2009.2038959).

55. Zachariah, Dave; Chatterjee, Saikat; Jansson, Magnus (2012). "Dynamic Iterative Pursuit". *IEEE Transactions on Signal Processing*. **60** (9): 4967–4972. arXiv:1206.2496 (https://arxiv.org/abs/1206.2496). Bibcode:2012ITSP...60.4967Z (http://adsabs.harvard.edu/abs/2012ITSP...60.4967Z). doi:10.1109/TSP.2012.2203813 (https://doi.org/10.1109%2FTSP.2012.2203813).

56. Vasebi, Amir; Partovibakhsh, Maral; Bathaee, S. Mohammad Taghi (2007). "A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications". *Journal of Power Sources*. **174** (1): 30–40. Bibcode:2007JPS...174...30V (http://adsabs.harvard.edu/abs/2007JPS...174...30V). doi:10.1016/j.jpowsour.2007.04.011 (https://doi.org/10.1016%2Fj.jpowsour.2007.04.011).

57. Vasebi, A.; Bathaee, S.M.T.; Partovibakhsh, M. (2008). "Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended Kalman filter". *Energy Conversion and Management*. **49**: 75–82. doi:10.1016/j.enconman.2007.05.017 (https://doi.org/10.1016%2Fj.enconman.2007.05.017).

58. Fruhwirth, R. (1987). "Application of Kalman filtering to track and vertex fitting". *Nucl. Instrum. Methods*. **A262** (2–3): 444–450. Bibcode:1987NIMPA.262..444F (http://adsabs.harvard.edu/abs/1987NIMPA.262..444F). doi:10.1016/0168-9002(87)90887-4 (https://doi.org/10.1016%2F0168-9002%2887%2990887-4).

59. Harvey, Andrew C. (1994). "Applications of the Kalman filter in econometrics". In Bewley, Truman (ed.). *Advances in Econometrics*. New York: Cambridge University Press. pp. 285f. ISBN 978-0-521-46726-1.

60. Boulfelfel, D.; Rangayyan, R.M.; Hahn, L.J.; Kloiber, R.; Kuduvalli, G.R. (1994). "Two-dimensional restoration of single photon emission computed tomography images using the Kalman filter". *IEEE Transactions on Medical Imaging*. **13** (1): 102–109. doi:10.1109/42.276148 (https://doi.org/10.1109%2F42.276148). PMID 18218487 (https://www.ncbi.nlm.nih.gov/pubmed/18218487).

61. Bock, Y.; Crowell, B.; Webb, F.; Kedar, S.; Clayton, R.; Miyahara, B. (2008). "Fusion of High-Rate GPS and Seismic Data: Applications to Early Warning Systems for Mitigation of Geological Hazards". *American Geophysical Union*. **43**: G43B–01. Bibcode:2008AGUFM.G43B..01B (http://adsabs.harvard.edu/abs/2008AGUFM.G43B..01B).

62. Wolpert, D. M.; Miall, R. C. (1996). "Forward Models for Physiological Motor Control". *Neural Netw*. **9** (8): 1265–1279. doi:10.1016/S0893-6080(96)00035-4 (https://doi.org/10.1016%2FS0893-6080%2896%2900035-4). PMID 12662535 (https://www.ncbi.nlm.nih.gov/pubmed/12662535).

Further reading

- Einicke, G.A. (2019). *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future (2nd ed.)*. Amazon Prime Publishing. ISBN 978-0-6485115-0-2.

- Jinya Su; Baibing Li; Wen-Hua Chen (2015). "On existence, optimality and asymptotic stability of the Kalman filter with partially observed inputs". *Automatica*. **53**: 149–154. doi:10.1016/j.automatica.2014.12.044 (https://doi.org/10.1016%2Fj.automatica.2014.12.044).

- Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press.

- Kalman, R.E. (1960). "A new approach to linear filtering and prediction problems" (https://web.archive.org/web/20080529105724/http://www.elo.utfsm.cl/~ipd481/Papers%20varios/kalman1960.pdf) (PDF). *Journal of Basic Engineering*. **82** (1): 35–45. doi:10.1115/1.3662552 (https://doi.org/10.1115%2F1.3662552). Archived from the original (http://www.elo.utfsm.cl/~ipd481/Papers%20varios/kalman1960.pdf) (PDF) on 2008-05-29. Retrieved 2008-05-03.

- Kalman, R.E.; Bucy, R.S. (1961). "New Results in Linear Filtering and Prediction Theory". *Trans. Asme, Ser. D, J. Basic Eng*: 109. CiteSeerX 10.1.1.361.6851 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.361.6851).

- Harvey, A.C. (1990). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

- Roweis, S.; Ghahramani, Z. (1999). "A Unifying Review of Linear Gaussian Models" (https://authors.library.caltech.edu/13697/1/ROWnc99.pdf) (PDF). *Neural Computation*. **11** (2): 305–345. doi:10.1162/089976699300016674 (https://doi.org/10.1162%2F089976699300016674). PMID 9950734 (https://www.ncbi.nlm.nih.gov/pubmed/9950734).

- Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches* (http://academic.csuohio.edu/simond/estimation/). Wiley-Interscience.

- Stengel, R.F. (1994). *Optimal Control and Estimation* (http://www.princeton.edu/~stengel/OptConEst.html). Dover Publications. ISBN 978-0-486-68200-6.

- Warwick, K. (1987). "Optimal observers for ARMA models". *International Journal of Control*. **46** (5): 1493–1503. doi:10.1080/00207178708933989 (https://doi.org/10.1080%2F00207178708933989).

- Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation*. *Mathematics in Science and Engineering*. **128**. Mineola, N.Y.: Dover Publications. ISBN 978-0-486-44981-4.

- Bozic, S.M. (1994). *Digital and Kalman filtering*. Butterworth–Heinemann.

- Haykin, S. (2002). *Adaptive Filter Theory*. Prentice Hall.

- Liu, W.; Principe, J.C. and Haykin, S. (2010). *Kernel Adaptive Filtering: A Comprehensive Introduction*. John Wiley.

- Manolakis, D.G. (1999). *Statistical and Adaptive signal processing*. Artech House.

- Welch, Greg; Bishop, Gary (1997). "SCAAT: incremental tracking with incomplete information" (http://www.cs.unc.edu/~welch/media/pdf/scaat.pdf) (PDF). *SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. pp. 333–344. doi:10.1145/258734.258876 (https://doi.org/10.1145%2F258734.258876). ISBN 978-0-89791-896-1.
- Jazwinski, Andrew H. (1970). *Stochastic Processes and Filtering*. Mathematics in Science and Engineering. New York: Academic Press. p. 376. ISBN 978-0-12-381550-7.
- Maybeck, Peter S. (1979). "Chapter 1" (http://www.cs.unc.edu/~welch/kalman/media/pdf/maybeck_ch1.pdf) (PDF). *Stochastic Models, Estimation, and Control*. Mathematics in Science and Engineering. 141-1. New York: Academic Press. ISBN 978-0-12-480701-3.
- Moriya, N. (2011). *Primer to Kalman Filtering: A Physicist Perspective*. New York: Nova Science Publishers, Inc. ISBN 978-1-61668-311-5.
- Dunik, J.; Simandl M.; Straka O. (2009). "Methods for Estimating State and Measurement Noise Covariance Matrices: Aspects and Comparison" (https://www.researchgate.net/publication/216411106). *15th IFAC Symposium on System Identification, 2009*. 15th IFAC Symposium on System Identification, 2009. France. pp. 372–377. doi:10.3182/20090706-3-FR-2004.00061 (https://doi.org/10.3182%2F20090706-3-FR-2004.00061). ISBN 978-3-902661-47-0.
- Chui, Charles K.; Chen, Guanrong (2009). *Kalman Filtering with Real-Time Applications*. Springer Series in Information Sciences. **17** (4th ed.). New York: Springer. p. 229. ISBN 978-3-540-87848-3.
- Spivey, Ben; Hedengren, J. D. and Edgar, T. F. (2010). "Constrained Nonlinear Estimation for Industrial Process Fouling". *Industrial & Engineering Chemistry Research*. **49** (17): 7824–7831. doi:10.1021/ie9018116 (https://doi.org/10.1021%2Fie9018116).
- Thomas Kailath; Ali H. Sayed; Babak Hassibi (2000). *Linear Estimation*. NJ: Prentice–Hall. ISBN 978-0-13-022464-4.
- Ali H. Sayed (2008). *Adaptive Filters*. NJ: Wiley. ISBN 978-0-470-25388-5.

External links

- A New Approach to Linear Filtering and Prediction Problems (http://www.cs.unc.edu/~welch/kalman/kalmanPaper.html), by R. E. Kalman, 1960
- Kalman and Bayesian Filters in Python (https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python). Open source Kalman filtering textbook.
- How a Kalman filter works, in pictures (http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/). Illuminates the Kalman filter with pictures and colors
- *Kalman–Bucy Filter* (http://www.eng.tau.ac.il/~liptser/lectures1/lect6.pdf), a derivation of the Kalman–Bucy Filter
- MIT Video Lecture on the Kalman filter (https://www.youtube.com/watch?v=d0D3VwBh5UQ) on YouTube
- An Introduction to the Kalman Filter (http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf), SIGGRAPH 2001 Course, Greg Welch and Gary Bishop
- Kalman Filter (http://www.cs.unc.edu/~welch/kalman/) webpage, with lots of links
- "Kalman filters used in Weather models" (http://www.siam.org/pdf/news/362.pdf) (PDF). *SIAM News*. **36** (8). October 2003.
- Haseltine, Eric L.; Rawlings, James B. (2005). "Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation". *Industrial & Engineering Chemistry Research*. **44** (8): 2451. doi:10.1021/ie034308l (https://doi.org/10.1021%2Fie034308l).
- Gerald J. Bierman's Estimation Subroutine Library (http://netlib.org/a/esl.tgz): Corresponds to the code in the research monograph "Factorization Methods for Discrete Sequential Estimation" originally published by Academic Press in 1977. Republished by Dover.
- Matlab Toolbox implementing parts of Gerald J. Bierman's Estimation Subroutine Library (http://www.mathworks.com/matlabcentral/fileexchange/32537): UD / UDU' and LD / LDL' factorization with associated time and measurement updates making up the Kalman filter.
- Matlab Toolbox of Kalman Filtering applied to Simultaneous Localization and Mapping (http://eia.udg.es/~qsalvi/Slam.zip): Vehicle moving in 1D, 2D and 3D
- The Kalman Filter in Reproducing Kernel Hilbert Spaces (https://web.archive.org/web/20160304042652/http://www.cnel.ufl.edu/~weifeng/publication.htm) A comprehensive introduction.

- Matlab code to estimate Cox–Ingersoll–Ross interest rate model with Kalman Filter (http://www.mathfinance.cn/kalman-filter-finance-revisited/): Corresponds to the paper "estimating and testing exponential-affine term structure models by kalman filter" published by Review of Quantitative Finance and Accounting in 1999.
- Online demo of the Kalman Filter (http://www.data-assimilation.net/Tools/AssimDemo/?method=KF). Demonstration of Kalman Filter (and other data assimilation methods) using twin experiments.
- Botella, Guillermo; Martín h., José Antonio; Santos, Matilde; Meyer-Baese, Uwe (2011). "FPGA-Based Multimodal Embedded Sensor System Integrating Low- and Mid-Level Vision" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231703). *Sensors*. **11** (12): 1251–1259. doi:10.3390/s110808164 (https://doi.org/10.3390%2Fs110808164). PMC 3231703 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231703). PMID 22164069 (https://www.ncbi.nlm.nih.gov/pubmed/22164069).
- Examples and how-to on using Kalman Filters with MATLAB (http://www.mathworks.com/discovery/kalman-filter.html) A Tutorial on Filtering and Estimation
- Explaining Filtering (Estimation) in One Hour, Ten Minutes, One Minute, and One Sentence (http://blog.sciencenet.cn/home.php?mod=space&uid=1565&do=blog&id=851754) by Yu-Chi Ho

---

Retrieved from "https://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=895067287"