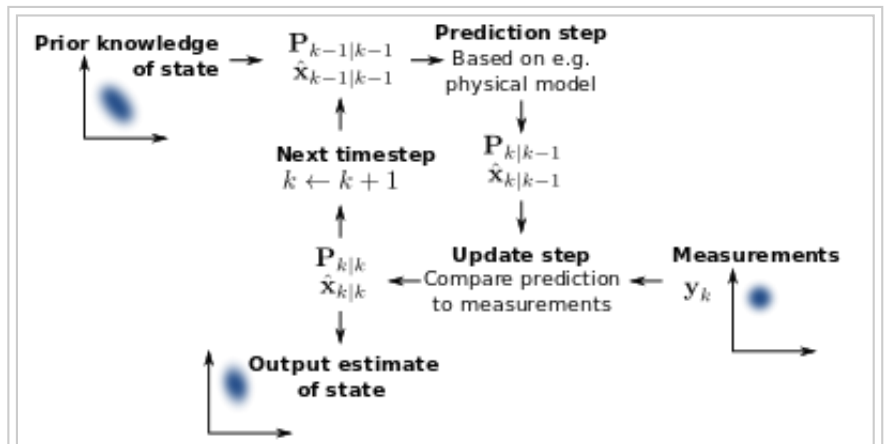# Kalman filter

From Wikipedia, the free encyclopedia

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 precise than those based on a single measurement alone, by using Bayesian inference and 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 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.

The Kalman filter has numerous applications in technology. A common application is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and 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 has also found use in 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 provides the needed model for making estimates of the current state of the motor system and issuing updated commands. [1]

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.

The Kalman filter does not require any assumption that the errors are Gaussian. [2] However, the filter yields the exact conditional probability estimate in the special case that all errors are Gaussian-distributed.

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 a Bayesian model similar to a hidden Markov model but where the state space of the latent variables is continuous and where all latent and observed variables have Gaussian distributions.

Contents

Naming and historical development

The filter is named after Hungarian émigré Rudolf E. Kálmán, although Thorvald Nicolai Thiele[3][4] and Peter Swerling developed a similar algorithm earlier. Richard S. Bucy of the University of Southern California contributed to the theory, leading to it often 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.[5] 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

Rudolf Emil Kálmán, co-inventor and developer of the Kalman filter.

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).

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. It is also used in the guidance and navigation systems of the NASA Space Shuttle and the attitude control and navigation systems of the International Space Station.

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.[6][7][8][9] 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.

Overview of the calculation

The Kalman filter uses a system's dynamics 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 any one measurement alone. As such, it is a common sensor fusion and data fusion algorithm.

All measurements and calculations based on models are estimated to some degree. Noisy sensor data, approximations in the equations that describe how a system changes, and external factors that are not accounted for introduce some uncertainty about the inferred values for a system's state. The Kalman filter averages a prediction of a system's state with a 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 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.

Because the certainty of the measurements is often difficult to measure precisely, it is common to discuss the filter's behavior in terms of gain. The Kalman gain is a function of the relative certainty of 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 measurements, and thus follows them more closely. With a low gain, the filter follows the model predictions more closely, smoothing out noise but decreasing the responsiveness. At the extremes, a gain of one causes the filter to ignore the state estimate entirely, while a gain of zero causes the measurements to be ignored.

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 always 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) plus any changes produced by the accelerator pedal and steering wheel. 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, if 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 changing 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,[10][11] 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.
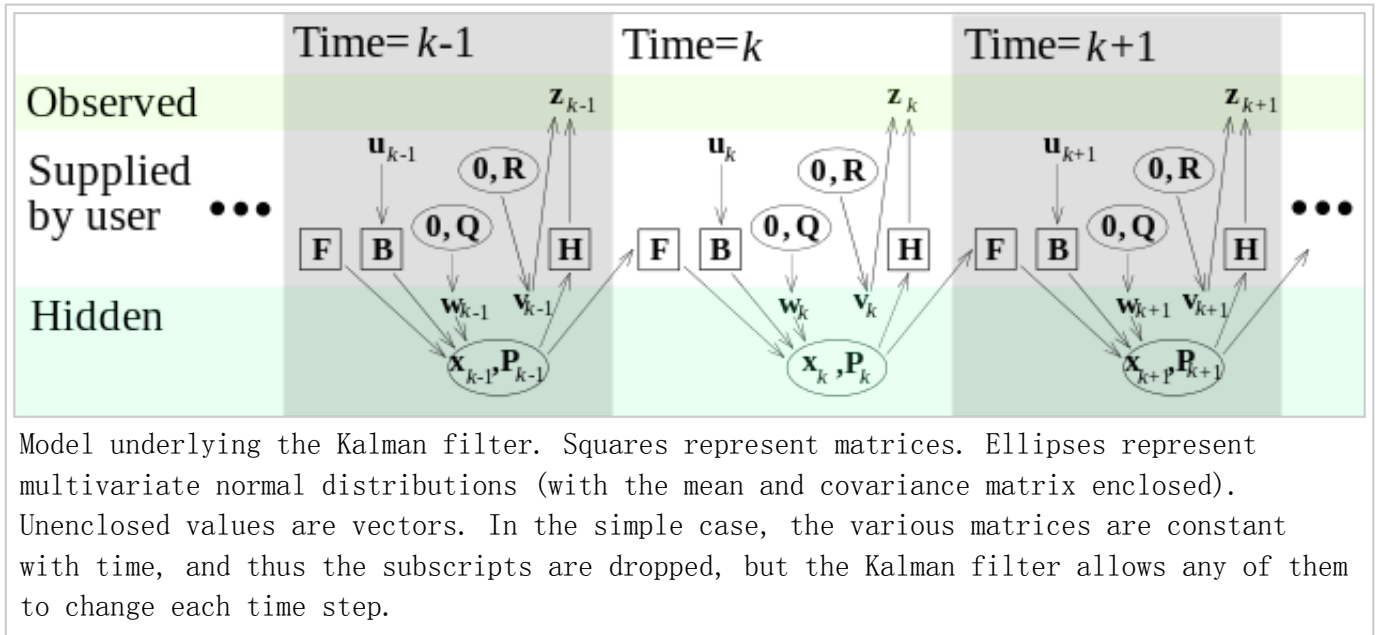
In Dempster–Shafer theory, each state equation or observation is considered a special case of a linear belief function and the Kalman filter is a special case of combining linear belief functions on a join-tree or Markov tree. Additional approaches include belief filters which use Bayes or evidential 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 dynamic system model

The Kalman filters are based on linear dynamic systems discretized in the time domain. They are modelled on a Markov chain built on linear operators perturbed by errors that may include Gaussian noise. The state of the system is represented as a vector of real numbers. At each discrete time 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 duality 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 Ghahramani (1999)[12] and Hamilton (1994), Chapter 13.[13]

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 covariance of the process noise; R_k, the covariance 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

- F_k is the state transition model which is applied to the previous state x_{k-1};
- B_k is the control-input model which is applied to the control vector u_k;
- w_k is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_k.

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

At time k an observation (or measurement) z_k of the true state x_k is made according to

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

where H_k is the observation model which maps the true state space into the observed space and v_k is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_k.

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$$

The initial state, and the noise vectors at each step $\{x_0, w_1, \cdots, w_k, v_1 \cdots v_k\}$ are all assumed to be mutually independent.

Many real dynamical systems do not exactly fit this model. In fact, unmodelled 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 unmodelled 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 separating between measurement noise and unmodelled dynamics is a difficult one and is treated in control theory under the framework of robust control. [14][15]

Details

The Kalman filter is a recursive 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 m, and including at time m $\leqslant$ 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 accuracy 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 H_k). [16][17]

Predict

Predicted (a priori) state estimate $\qquad \hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k$

Predicted (a priori) estimate covariance $\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

Update

Innovation or measurement residual	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$
Innovation (or residual) covariance	$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$
Optimal Kalman gain	$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$
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	$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$

The formula for the updated estimate covariance above is only valid for the optimal Kalman gain. Usage of other gain values requires a more complex formula 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: (all estimates have a mean error of zero)

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

where $\mathbf{E}[\xi]$ is the expected value of ξ, and covariance matrices accurately reflect the covariance of estimates

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

Optimality and performance

It follows from theory that the Kalman filter is optimal in cases where a) the model perfectly matches the real system, b) the entering noise is white and Gaussian and c) the covariances of the noise are exactly known. Several methods for the noise covariance estimation have been proposed during past decades. 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.[18] 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 given in [19] and.[20]

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 its velocity is. We show here how we derive the model from which we create our Kalman filter.

Black: truth, green: filtered process, red: observations
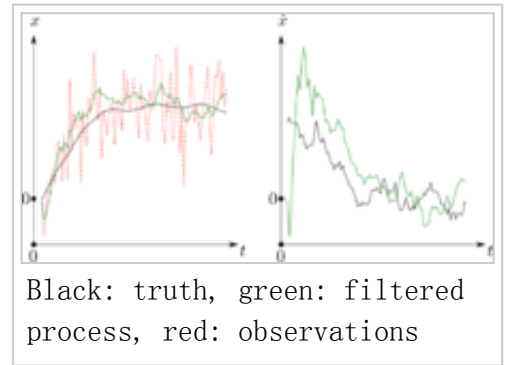
Since $\mathbf{F}, \mathbf{H}, \mathbf{R}, \mathbf{Q}$ are constant, their time indices are dropped.

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

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

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

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

and

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

so that

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

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

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

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

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

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 a posteriori estimate covariance matrix

Starting with our invariant on the error covariance $P_{k|k}$ as above

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

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

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

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

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

and $\mathbf{z}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

and by collecting the error vectors we get

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

Since the measurement error $\mathrm{v_k}$ is uncorrelated with the other terms, this becomes

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

by the properties of vector covariance this becomes

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

which, using our invariant on $\mathrm{P_{k\,|\,k-1}}$ and the definition of $\mathrm{R_k}$ becomes

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

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

Kalman gain derivation

The Kalman filter is a minimum mean-square error 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, $\mathrm{E}\big[\|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\|^2\big]$. This is equivalent to minimizing the trace of the a posteriori estimate covariance matrix $\mathbf{P}_{k|k}$. By expanding out the terms in the equation above and collecting, we get:

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

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

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

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

Solving this for $\mathrm{K_k}$ yields the Kalman gain:

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

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

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

Simplification of the a 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 $S_k K_k{}^T$, it follows that

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

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

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

we find the last two terms cancel out, giving

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

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.[21] In the absence of reliable statistics or the true values of noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, the expression

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

no longer provides the actual error covariance. In other words, $\mathbf{P}_{k|k} \neq E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^{\mathbf{T}}]$. 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 $\mathbf{P}_{k|k}^a = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^\mathbf{T}]$, substituting for $\widehat{\mathbf{x}}_{k|k}$ and using the fact that $E[\mathbf{w}_k\mathbf{w}_k^\mathbf{T}] = \mathbf{Q}_k^a$ and $E[\mathbf{v}_k\mathbf{v}_k^\mathbf{T}] = \mathbf{R}_k^a$, results in the following recursive equations for $\mathbf{P}_{k|k}^a$:

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

and

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

While computing $\mathbf{P}_{k|k}$, by design the filter implicitly assumes that $E[\mathbf{w}_k\mathbf{w}_k^\mathbf{T}] = \mathbf{Q}_k$ and $E[\mathbf{v}_k\mathbf{v}_k^\mathbf{T}] = \mathbf{R}_k$. Note that the recursive expressions for $\mathbf{P}_{k|k}^a$ and $\mathbf{P}_{k|k}$ are identical except for the presence of $\mathbf{Q}_k^a$ and $\mathbf{R}_k^a$ in place of the design values $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively.

Square root form

One problem with the Kalman filter is its numerical stability. If the process noise covariance 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 P indefinite, while its true form is positive-definite.

Positive definite matrices have the property that they have a triangular matrix square root P = S • S$^\mathrm{T}$. 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, P = U • D • U$^\mathrm{T}$, where U is a unit triangular matrix (with unit diagonal), and 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,[22]: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.[22][23]

The $L \cdot D \cdot L^T$ decomposition of the innovation covariance matrix S_k is the basis for another type of numerically efficient and robust square root filter.[24] The algorithm starts with the LU decomposition as implemented in the Linear Algebra PACKage (LAPACK). These results are further factored into the $L \cdot D \cdot L^T$ structure with methods given by Golub and Van Loan (algorithm 4.1.2) for a symmetric nonsingular matrix.[25] 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 $H_k \cdot x_{k|k-1}$ that are associated with auxiliary observations in y_k. The $l \cdot d \cdot l^t$ square-root filter requires orthogonalization of the observation vector.[23][24] 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).[26]

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.[27]

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).
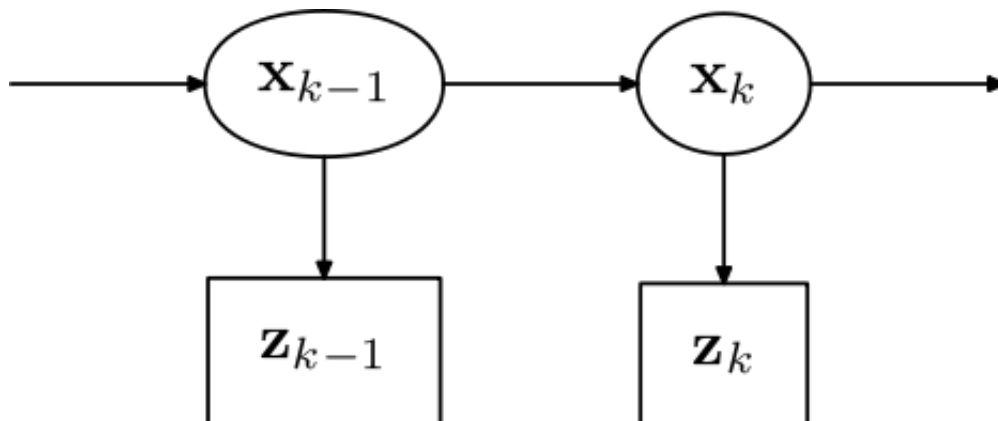
because of the Markov assumption, the true state is conditionally independent of all earlier states given the immediately previous state.

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

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 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 x_{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

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

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 $z = (z_0, z_1, z_2, \cdots)$. Specifically, the process is

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

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,

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

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

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

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

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

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

i.e., a product of Gaussian densities, each corresponding to the density of one observation 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

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

where d_y is the dimension of the measurement vector. [28]

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:

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

The information update now becomes a trivial sum. [29]

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

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. [29]

$$\mathbf{M}_k = [\mathbf{F}_k^{-1}]^{\mathrm{T}}\mathbf{Y}_{k-1|k-1}\mathbf{F}_k^{-1}$$

$$\mathbf{C}_k = \mathbf{M}_k[\mathbf{M}_k + \mathbf{Q}_k^{-1}]^{-1}$$
$$\mathbf{L}_k = I - \mathbf{C}_k$$
$$\mathbf{Y}_{k|k-1} = \mathbf{L}_k\mathbf{M}_k\mathbf{L}_k^{\mathrm{T}} + \mathbf{C}_k\mathbf{Q}_k^{-1}\mathbf{C}_k^{\mathrm{T}}$$

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{L}_k[\mathbf{F}_k^{-1}]^{\mathrm{T}}\hat{\mathbf{y}}_{k-1|k-1}$$

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 $\mathbf{z}_k$. [30] 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:

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

and

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

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:

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

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. [31]

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:

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

where

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

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.[23] 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

$$\tilde{\Lambda}_k = \mathbf{H}_k^T \mathbf{S}_k^{-1}\mathbf{H}_k + \hat{\mathbf{C}}_k^T \hat{\Lambda}_k \hat{\mathbf{C}}_k$$
$$\hat{\Lambda}_{k-1} = \mathbf{F}_k^T \tilde{\Lambda}_k \mathbf{F}_k$$
$$\hat{\Lambda}_n = 0$$
$$\tilde{\lambda}_k = -\mathbf{H}_k^T \mathbf{S}_k^{-1}\mathbf{y}_k + \hat{\mathbf{C}}_k^T \hat{\lambda}_k$$
$$\hat{\lambda}_{k-1} = \mathbf{F}_k^T \tilde{\lambda}_k$$
$$\hat{\lambda}_n = 0$$

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. [32] 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

$$\hat{\mathbf{x}}_{k+1|k} = (\mathbf{F}_k - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{z}_k$$
$$\alpha_k = -\mathbf{S}_k^{-1/2}\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} + \mathbf{S}_k^{-1/2}\mathbf{z}_k$$

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

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

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. [33][34]

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. [35]

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

Accommodating Poisson noise

The Kalman filter is the optimum estimator from the set of all causal linear estimators when the underlying noise processes are zero-mean white processes. In some applications, the measurement noise may be Poisson-distributed. The filter recursions may be generalized to include a Poisson measurement noise component by adding the Poisson intensity Γ to the innovation (or residual) covariance:

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

An expectation-maximization algorithm may be employed to iteratively estimate an unknown Poisson noise intensity. [36]

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. [37] 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.

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

$$\mathbf{z}_k = h\big(\mathbf{x}_k\big) + \mathbf{v}_k$$

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. [38] This is because the covariance is propagated through linearization of the underlying non-linear model. The unscented Kalman filter (UKF) [38] uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered. The result is a filter which more accurately captures the true mean and covariance. (This can be verified using Monte Carlo sampling or through a 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).

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.

The estimated state and covariance are augmented with the mean and covariance of the process noise.

$$\mathbf{x}^a_{k-1|k-1} = [\hat{\mathbf{x}}^{\mathrm{T}}_{k-1|k-1} \quad E[\mathbf{w}^{\mathrm{T}}_k]\ ]^{\mathrm{T}}$$

$$\mathbf{P}^a_{k-1|k-1} = \begin{bmatrix} \mathbf{P}_{k-1|k-1} & 0 \\ 0 & \mathbf{Q}_k \end{bmatrix}$$

A set of 2L + 1 sigma points is derived from the augmented state and covariance where L is the dimension of the augmented state.

$$\chi^0_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1}$$
$$\chi^i_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1} + \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_i, \qquad i = 1,\ldots,L$$
$$\chi^i_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1} - \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_{i-L}, \qquad i = L+1,\ldots,2L$$

where

$$\left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_i$$

is the ith column of the matrix square root of

$$(L+\lambda)\mathbf{P}^a_{k-1|k-1}$$

using the definition: square root $\mathbf{A}$ of matrix $\mathbf{B}$ satisfies

$$\mathbf{B} \triangleq \mathbf{A}\mathbf{A}^{\mathrm{T}}.$$

The matrix square root should be calculated using numerically efficient and stable methods such as the Cholesky decomposition.

The sigma points are propagated through the transition function f.

$$\chi^i_{k|k-1} = f(\chi^i_{k-1|k-1}) \quad i = 0, \ldots, 2L$$

where $f : R^L \to R^{|\mathbf{X}|}$. The weighted sigma points are recombined to produce the predicted state and covariance.

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} W^i_s \chi^i_{k|k-1}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2L} W^i_c \left[\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]\left[\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]^{\mathrm{T}}$$

where the weights for the state and covariance are given by:

$$W^0_s = \frac{\lambda}{L + \lambda}$$
$$W^0_c = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta)$$
$$W^i_s = W^i_c = \frac{1}{2(L + \lambda)}$$
$$\lambda = \alpha^2(L + \kappa) - L$$

α and κ control the spread of the sigma points. β is related to the distribution of x. Normal values are $\alpha = 10^{-3}$, $\kappa = 0$ and $\beta = 2$. If the true distribution of x is Gaussian, $\beta = 2$ is optimal. [39]

Update

The predicted state and covariance are augmented as before, except now with the mean and covariance of the measurement noise.

$$\mathbf{x}^a_{k|k-1} = [\hat{\mathbf{x}}^{\mathrm{T}}_{k|k-1} \quad E[\mathbf{v}^{\mathrm{T}}_k]]^{\mathrm{T}}$$

$$\mathbf{P}^a_{k|k-1} = \begin{bmatrix} \mathbf{P}_{k|k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix}$$

As before, a set of 2L + 1 sigma points is derived from the augmented state and covariance where L is the dimension of the augmented state.

$$\chi^0_{k|k-1} = \mathbf{x}^a_{k|k-1}$$

$$\chi^i_{k|k-1} = \mathbf{x}^a_{k|k-1} + \left(\sqrt{(L + \lambda)\mathbf{P}^a_{k|k-1}}\right)_i, \qquad i = 1, \ldots, L$$

$$\chi^i_{k|k-1} = \mathbf{x}^a_{k|k-1} - \left(\sqrt{(L + \lambda)\mathbf{P}^a_{k|k-1}}\right)_{i-L}, \qquad i = L + 1, \ldots, 2L$$

Alternatively if the UKF prediction has been used the sigma points themselves can be augmented along the following lines

$$\chi_{k|k-1} := [\chi_{k|k-1}^{\mathrm{T}} \quad E[\mathbf{v}_k^{\mathrm{T}}]]^{\mathrm{T}} \pm \sqrt{(L+\lambda)\mathbf{R}_k^a}$$

where

$$\mathbf{R}_k^a = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{R}_k \end{bmatrix}$$

The sigma points are projected through the observation function h.

$$\gamma_k^i = h(\chi_{k|k-1}^i) \quad i = 0..2L$$

The weighted sigma points are recombined to produce the predicted measurement and predicted measurement covariance.

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2L} W_s^i \gamma_k^i$$

$$\mathbf{P}_{z_k z_k} = \sum_{i=0}^{2L} W_c^i [\gamma_k^i - \hat{\mathbf{z}}_k][\gamma_k^i - \hat{\mathbf{z}}_k]^{\mathrm{T}}$$

The state-measurement cross-covariance matrix,

$$\mathbf{P}_{x_k z_k} = \sum_{i=0}^{2L} W_c^i [\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}][\gamma_k^i - \hat{\mathbf{z}}_k]^{\mathrm{T}}$$

is used to compute the UKF Kalman gain.

$$K_k = \mathbf{P}_{x_k z_k} \mathbf{P}_{z_k z_k}^{-1}$$

As with the Kalman filter, the updated state is the predicted state plus the innovation weighted by the Kalman gain,

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

And the updated covariance is the predicted covariance, minus the predicted measurement covariance, weighted by the Kalman gain.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - K_k \mathbf{P}_{z_k z_k} K_k^{\mathrm{T}}$$

Kalman–Bucy filter

The Kalman–Bucy filter (named after Richard Snowden Bucy) is a continuous time version of the Kalman filter. [40][41]

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 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:

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{K}(t)(\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t))$$

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t)$$

where the Kalman gain is given by

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

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. [42]

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

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim N\big(\mathbf{0}, \mathbf{Q}(t)\big)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \qquad\qquad \mathbf{v}_k \sim N\big(\mathbf{0}, \mathbf{R}_k\big)$$

where

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

Initialize
$$\hat{\mathbf{x}}_{0|0} = E\big[\mathbf{x}(t_0)\big], \mathbf{P}_{0|0} = Var\big[\mathbf{x}(t_0)\big]$$
Predict

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

The prediction equations are derived from those of continuous-time Kalman filter without update from measurements, i.e., $\mathbf{K}(t) = 0$. 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

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

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[43][44][45] 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[46][47]
- Brain-computer interface
- Chaotic signals
- Tracking and vertex fitting of charged particles in particle detectors[48]
- Tracking of objects in computer vision
- Dynamic positioning
- Economics, in particular macroeconomics, time series analysis, and econometrics[49][50]
- Inertial guidance system
- Nuclear medicine - single photon emission computed tomography image restoration[51]

- Orbit Determination
- Power system state estimation
- Radar tracker
- Satellite navigation systems
- Seismology[52]
- 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[53]

See also

- Alpha beta filter
- Bayesian MMSE estimator
- Covariance intersection
- Data assimilation
- Ensemble Kalman filter
- Extended Kalman filter
- Fast Kalman filter
- Filtering problem (stochastic processes)

- Generalized filtering
- Invariant extended Kalman filter
- Kernel adaptive filter
- Linear-quadratic-Gaussian control
- Masreliez's theorem
- Moving horizon estimation
- Non-linear filter
- Particle filter estimator
- Predictor corrector

- Recursive least squares
- Schmidt–Kalman filter
- Separation principle
- Sliding mode control
- Stochastic differential equations
- Volterra series
- Wiener filter
- Zakai equation

References

1. Wolpert, Daniel; Ghahramani, Zoubin (2000). "Computational principles of movement neuroscience". Nature Neuroscience 3: 1212–7. doi:10.1038/81497. PMID 11127840.
2. Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering 82: 35. doi:10.1115/1.3662552.
3. Steffen L. Lauritzen (http://www.stats.ox.ac.uk/~steffen/). "Time series analysis in 1880. A discussion of contributions made by T.N. Thiele". International Statistical Review 49, 1981, 319–333. JSTOR 1402616 (https://www.jstor.org/stable/1402616)
4. Steffen L. Lauritzen, Thiele: Pioneer in Statistics (http://www.oup.com/uk/catalogue/?ci=9780198509721), Oxford University Press, 2002. ISBN 0-19-850972-3.
5. Mohinder S. Grewal and Angus P. Andrews (http://ieeecss.org/CSM/library/2010/june10/11-HistoricalPerspectives.pdf)
6. 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.
7. 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.
8. Stratonovich, R. L. (1960) Application of the Markov processes theory to optimal filtering. Radio Engineering and Electronic Physics, 5:11, pp. 1–19.
9. Stratonovich, R. L. (1960). Conditional Markov Processes. Theory of Probability and its Applications, 5, pp. 156–178.
10. Ingvar Strid; Karl Walentin (April 2009). "Block Kalman Filtering for Large-Scale DSGE Models". Computational Economics (Springer) 33 (3): 277–304. doi:10.1007/s10614-008-9160-4.
11. Martin Møller Andreasen (2008). "Non-linear DSGE Models, The Central Difference Kalman Filter, and The Mean Shifted Particle Filter" (PDF).
12. Roweis, S; Ghahramani, Z (1999). "A unifying review of linear gaussian models". Neural computation 11 (2): 305–45. doi:10.1162/089976699300016674. PMID 9950734.
13. Hamilton, J. (1994), Time Series Analysis, Princeton University Press. Chapter 13, 'The Kalman Filter'.
14. 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.
15. 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.
16. Kelly, Alonzo (1994). "A 3D state space formulation of a navigation Kalman filter for autonomous vehicles" (PDF). DTIC Document: 13. 2006 Corrected Version (http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/kalman_V2.pdf)

17. Reid, Ian; Term, Hilary. "Estimation II" (PDF). www.robots.ox.ac.uk. Oxford University. Retrieved 6 August 2014.
18. Three optimality tests with numerical examples are described in Peter, Matisko, (2012). "Optimality Tests and Adaptive Kalman Filter". 16th IFAC Symposium on System Identification. 16th IFAC Symposium on System Identification. p. 1523. doi:10.3182/20120711-3-BE-2027.00011. ISBN 978-3-902823-06-9.
19. Spall, James C. (1995). "The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions". Automatica 31 (10): 1513. doi:10.1016/0005-1098(95)00069-9.
20. 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. doi:10.1109/TAC.2003.821415.
21. Anderson, Brian D. O.; Moore, John B. (1979). Optimal Filtering. New York: Prentice Hall. pp. 129–133. ISBN 0-13-638122-7.
22. Thornton, Catherine L. (15 October 1976). "Triangular Covariance Factorizations for Kalman Filtering" (PDF). (PhD thesis). NASA. NASA Technical Memorandum 33-798.
23. Bierman, G.J. (1977). "Factorization Methods for Discrete Sequential Estimation". Factorization Methods for Discrete Sequential Estimation (Academic Press). Bibcode:1977fmds.book.....B.
24. 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.
25. 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.
26. 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.
27. 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.
28. Lütkepohl, Helmut (1991). Introduction to Multiple Time Series Analysis. Heidelberg: Springer-Verlag Berlin,. p. 435.
29. Gabriel T. Terejanu (2012-08-04). "Discrete Kalman Filter Tutorial" (PDF). Retrieved 2016-04-13.
30. Anderson, Brian D. O.; Moore, John B. (1979). Optimal Filtering. Englewood Cliffs, NJ: Prentice Hall, Inc. pp. 176 – 190. ISBN 0-13-638122-7.
31. Rauch, H.E.; Tung, F.; Striebel, C. T. (August 1965). "Maximum likelihood estimates of linear dynamic systems". AIAA J 3 (8): 1445–1450. Bibcode:1965AIAAJ...3.1445. doi:10.2514/3.3166.
32. Einicke, G.A. (March 2006). "Optimal and Robust Noncausal Filter Formulations". IEEE Trans. Signal Processing 54 (3): 1069–1077. Bibcode:2006ITSP...54.1069E. doi:10.1109/TSP.2005.863042.
33. Einicke, G.A. (April 2007). "Asymptotic Optimality of the Minimum-Variance Fixed-Interval Smoother". IEEE Trans. Signal Processing 55 (4): 1543–1547. Bibcode:2007ITSP...55.1543E. doi:10.1109/TSP.2006.889402.
34. 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.
35. Einicke, G.A. (December 2009). "Asymptotic Optimality of the Minimum-Variance Fixed-Interval Smoother". IEEE Trans. Automatic Control 54 (12): 2904–2908. Bibcode:2007ITSP...55.1543E. doi:10.1109/TSP.2006.889402.
36. Einicke, G.A. (July 2015). "Iterative Filtering and Smoothing of Measurements Possessing Poisson Measurement Noise". IEEE Transactions on Aerospace and Electronic Systems 51 (3): 2205–2211. doi:10.1109/TAES.2015.140843.
37. Einicke, G.A. (December 2014). "Iterative Frequency-Weighted Filtering and Smoothing Procedures". IEEE Signal Processing Letters 21 (12): 1467–1470. doi:10.1109/LSP.2014.2341641.

38. Julier, Simon J.; Uhlmann, Jeffrey K. (1997). "A new extension of the Kalman filter to nonlinear systems" (PDF). Int. Symp. Aerospace/Defense Sensing, Simul. and Controls. Signal Processing, Sensor Fusion, and Target Recognition VI 3: 182. Bibcode:1997SPIE.3068..182J. doi:10.1117/12.280797. Retrieved 2008-05-03.

39. Wan, E.A.; Van Der Merwe, R. (2000). "The unscented Kalman filter for nonlinear estimation". Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373) (PDF). p. 153. doi:10.1109/ASSPCC.2000.882463. ISBN 0-7803-5800-7.

40. 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

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

42. 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.

43. Vaswani, Namrata (2008). "Kalman filtered Compressed Sensing". 2008 15th IEEE International Conference on Image Processing. p. 893. doi:10.1109/ICIP.2008.4711899. ISBN 978-1-4244-1765-0.

44. 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. doi:10.1109/TSP.2009.2038959.

45. Zachariah, Dave; Chatterjee, Saikat; Jansson, Magnus (2012). "Dynamic Iterative Pursuit". IEEE Transactions on Signal Processing 60 (9): 4967–4972. doi:10.1109/TSP.2012.2203813.

46. 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: 30–40. doi:10.1016/j.jpowsour.2007.04.011.

47. 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.

48. Fruhwirth, R. (1987). "Application of Kalman filtering to track and vertex fitting". Nucl. Instrum. Meth. A262 (2–3): 444–450. Bibcode:1987NIMPA.262..444F. doi:10.1016/0168-9002(87)90887-4.

49. Harvey, Andrew C. (1994). "Applications of the Kalman filter in econometrics". In Bewley, Truman. Advances in Econometrics. New York: Cambridge University Press. pp. 285f. ISBN 0-521-46726-8.

50. Dimitri O. Ledenyov, Viktor O. Ledenyov, On the Stratonovich – Kalman – Bucy filtering algorithm application for accurate characterization of financial time series with use of state-space model by central banks, 23rd Conference on the Theories and Practices of Securities and Financial Markets, National Sun Yat-sen University, Kaohsiung, Taipei, Taiwan, December 11 – 12, 2015, 52 pages article (http://sfm.finance.nsysu.edu.tw/php/Papers/CompletePaper/014-1856280412.pdf)

51. D. Boulfelfel, R.M. Rangayyan, L.J. Hahn, R. Kloiber, and G.R. Kuduvalli (1994). "Restoration of single photon emission computed tomography images by the Kalman filter (http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=276148&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D276148)". IEEE Transactions on Medical Imaging 13(1): 102–109

52. 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: 01. Bibcode:2008AGUFM.G43B..01B.

53. 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. PMID 12662535.

Further reading

- Einicke, G.A. (2012). Smoothing, Filtering and Prediction: Estimating the Past, Present

and Future. Rijeka, Croatia: Intech. ISBN 978-953-307-752-9.

- 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.
- Gelb, A. (1974). Applied Optimal Estimation. MIT Press.
- Kalman, R.E. (1960). "A new approach to linear filtering and prediction problems" (PDF). Journal of Basic Engineering 82 (1): 35–45. doi:10.1115/1.3662552. Retrieved 2008-05-03.
- Kalman, R.E.; Bucy, R.S. (1961). "New Results in Linear Filtering and Prediction Theory".
- 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". Neural Computation 11 (2): 305–345. doi:10.1162/089976699300016674. PMID 9950734.
- Simon, D. (2006). Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience.
- Stengel, R.F. (1994). Optimal Control and Estimation. Dover Publications. ISBN 0-486-68200-5.
- Warwick, K. (1987). "Optimal observers for ARMA models" (PDF). International Journal of Control 46 (5): 1493–1503. doi:10.1080/00207178708933989. Retrieved 2008-05-03.
- 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" (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. ISBN 0-89791-896-7.
- Jazwinski, Andrew H. (1970). Stochastic Processes and Filtering. Mathematics in Science and Engineering. New York: Academic Press. p. 376. ISBN 0-12-381550-9.
- Maybeck, Peter S. (1979). "Chapter 1". Stochastic Models, Estimation, and Control (PDF). Mathematics in Science and Engineering. 141-1. New York: Academic Press. ISBN 0-12-480701-1.
- 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". 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. 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.
- 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.
- Dimitri O. Ledenyov, Viktor O. Ledenyov, On the Stratonovich – Kalman – Bucy filtering algorithm application for accurate characterization of financial time series with use of state-space model by central banks, 23rd Conference on the Theories and Practices of Securities and Financial Markets, National Sun Yat-sen University, Kaohsiung, Taipei, Taiwan, December 11 – 12, 2015, 52 pages article (http://sfm.finance.nsysu.edu.tw/php/Papers/CompletePaper/014-1856280412.pdf).

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‑Bucy Filter (http://www.eng.tau.ac.il/~liptser/lectures1/lect6.pdf), a good 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 Filtering". Archived from the original on 2013-06-23.
- Kalman Filters, thorough introduction to several types, together with applications to Robot Localization (http://www.negenborn.net/kal_loc/)
- "Kalman filters used in Weather models" (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.
- Kalman and Bayesian Filters in Python (http://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python) Free book on Kalman Filtering implemented in IPython Notebook.
- Source code for the propeller microprocessor (http://obex.parallax.com/object/326): Well documented source code written for the Parallax propeller processor.
- 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
- Derivation of a 6D EKF solution to Simultaneous Localization and Mapping (http://www.mrpt.org/6D-SLAM) (In old version PDF (http://mapir.isa.uma.es/~jlblanco/papers/RangeBearingSLAM6D.pdf)). See also the tutorial on implementing a Kalman Filter (http://www.mrpt.org/Kalman_Filters) with the MRPT C++ libraries.
- The Kalman Filter Explained (http://www.tristanfletcher.co.uk/LDS.pdf) A very simple tutorial.
- The Kalman Filter in Reproducing Kernel Hilbert Spaces (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.
- Extended Kalman Filters (http://apmonitor.com/wiki/index.php/Main/Background)

explained in the context of Simulation, Estimation, Control, and Optimization

- 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". Sensors 11 (12): 1251–1259. doi:10.3390/s110808164.
- Hookes Law and the Kalman Filter (http://finmathblog.blogspot.com/2013/10/hookes-law-and-kalman-filter-little.html) A little "spring theory" emphasizing the connection between statistics and physics.
- 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=715306774"

Categories: Control theory │ Nonlinear filters │ Linear filters │ Signal estimation │ Stochastic differential equations │ Robot control │ Markov models

---