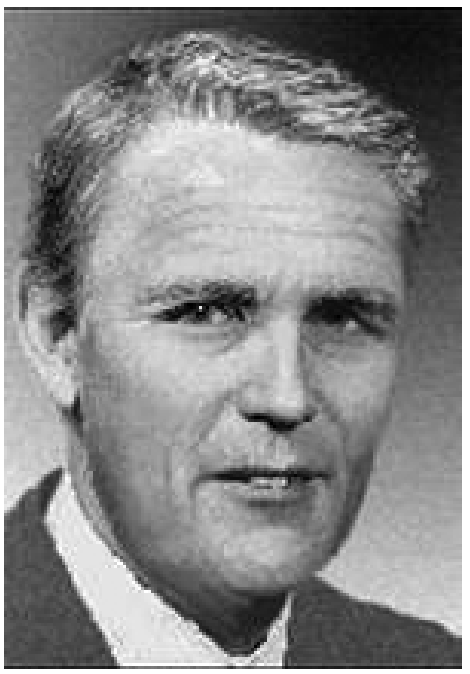
The Kalman Filter

"The *Kalman filter* is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown." (G. Welch and G. Bishop, 2004)



Named after Rudolf Emil Kalman (1930, Budapest/Hungary).

Kalman defined and published in 1960 a recursive solution to the discrete signal, linear filtering problem. Related basic ideas were also studied at that time by the US radar theoretician **Peter Swerling** (1929 – 2000). The Danish astronomer **Thorvald Nicolai Thiele** (1838 – 1910) is also cited for historic origins of involved ideas. See `en.wikipedia.org/wiki/Kalman_filter`.

The Kalman filter is a very powerful tool when it comes to controlling noisy systems.

Apollo 8 (December 1968), the first human spaceflight from the Earth to an orbit around the moon, would certainly not have been possible without the Kalman filter (see `www.ion.org/museum/item_view.cfm?cid=6&scid=5&iid=293`).

The basic idea of a Kalman filter:

Noisy data in $\Rightarrow$ Hopefully less noisy data out

The applications of a Kalman filter are numerous:

- Tracking objects (e.g., balls, faces, heads, hands)
- Fitting Bezier patches to point data
- Economics
- Navigation
- Many computer vision applications:
  - Stabilizing depth measurements
  - Feature tracking
  - Cluster tracking
  - Fusing data from radar, laser scanner and stereo-cameras for depth and velocity measurement
  - Many more

Structure of Presentation

We start with

(A) discussing briefly signals and noise, and

(B) recalling basics about random variables.

Then we start the actual subject with

(C) specifying *linear dynamic systems*, defined in continuous space.

This is followed by

(D) the goal of a Kalman filter and the discrete filter model, and

(E) a standard Kalman filter

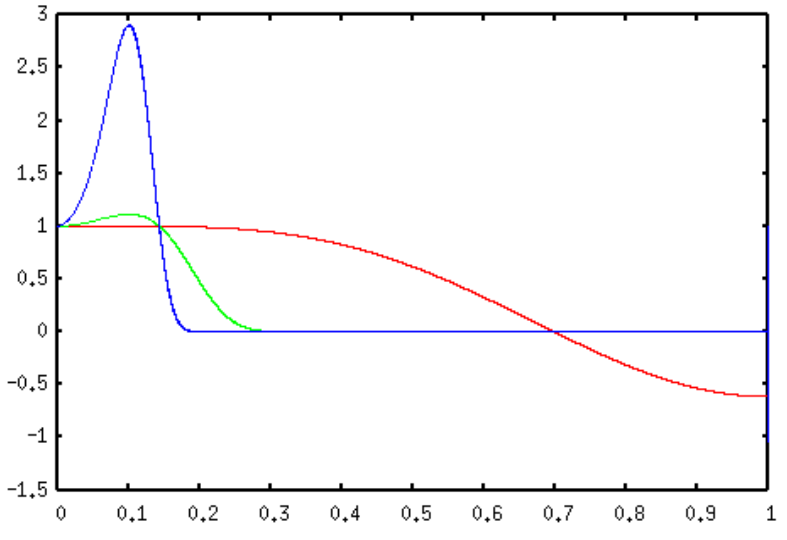
Note that there are many variants of such filters. - Finally (in this MI37) we outline

(F) a general scheme of applying a Kalman filter.

Two applications are then described in detail in subjects MI63 and MI64.

(A) Signals

A one-dimensional (1D) *signal* $x(t)$ has (typically) a time-varying amplitude. Axes are *amplitude* (vertical) and *time* (horizontal):



In its simplest form it is scalar-valued [e.g., a real-valued waveform such as $x(t) = \sin(2\pi t)$].

Quantization: A *discrete signal* is sampled at discrete positions in the signal's domain, and values are also (normally) discretized by allowing only values within a finite range. (For example, a digital gray-level picture is a discrete signal where spatial samples are taken at uniformly distributed grid point positions, and values within a finite set $\{0, 1, \ldots, G_{max}\}$.)

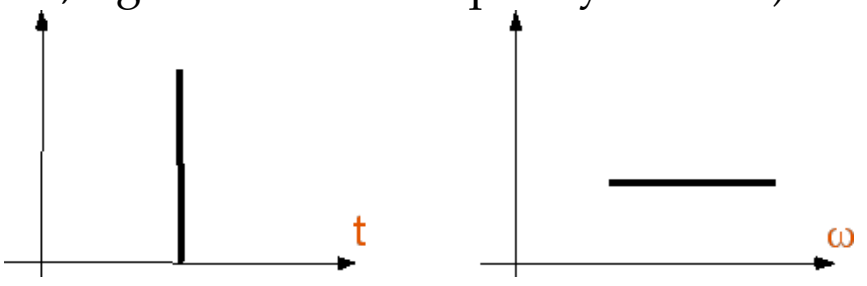
A single picture $I(i, j)$ is a two-dimensional (2D) discrete signal with scalar (i.e., gray levels) or vector [e.g. (R,G,B)] values; time t is replaced here by spatial coordinates i and j. A discrete time-sequence of digital images is a three-dimensional (3D) signal $x(t)(i, j) = I(i, j, t)$ that can be scalar- or vector-valued.

Noise

In a very general sense, “noise” is an unwanted contribution to a measured signal, and there are studies on various kinds of noise related to a defined context (acoustic noise, electronic noise, environmental noise, and so forth).

We are especially interested in *image noise* or *video noise*. Noise is here typically a high-frequency random perturbation of measured pixel values, caused by electronic noise of participating sensors (such as camera or scanner), or by transmission or digitization processes. For example, the Bayer pattern may introduce a noisy color mapping.

Example: *White noise* is defined by a constant (flat) spectrum within a defined frequency band, that means, it is something what is normally not assumed to occur in images (left: sketch of delta function; right: sketch in frequency domain).



Note: In image processing, “noise” is often also simply considered to be a measure for the variance of pixel values. For example, the *signal-to-noise ratio* (SNR) of a scalar image is commonly defined to be the ratio of mean to standard deviation of the image. Actually, this should be better called the *contrast ratio* (and we do so), to avoid confusion with the general perception that “noise” is “unwanted”.

mean: 114.32

standard deviation: 79.20

contrast ratio: 1.443

mean: 100.43 (darker, more contrast)

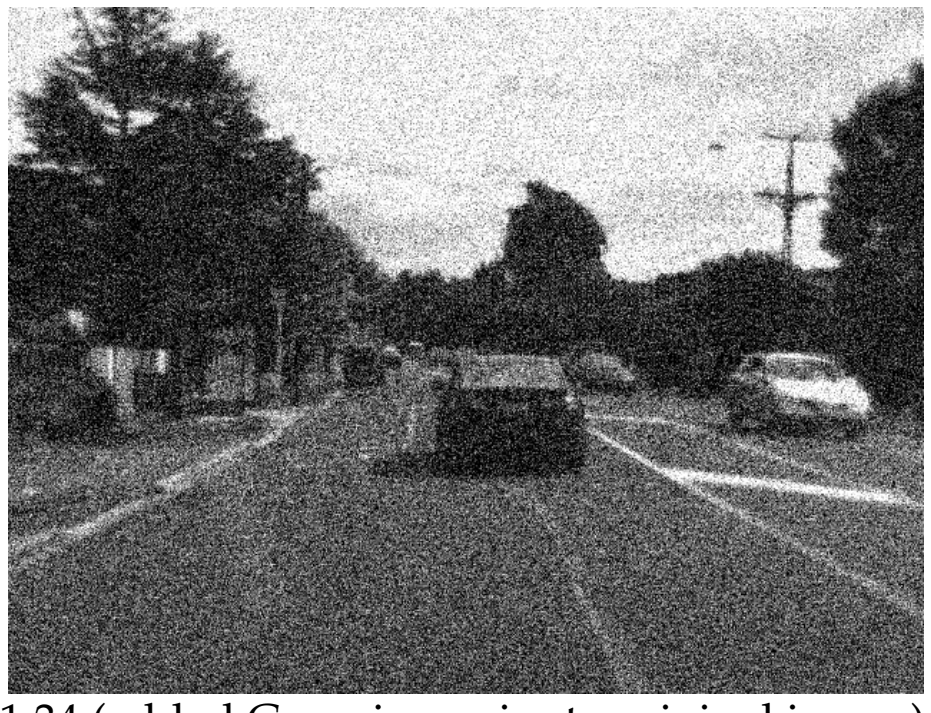
standard deviation: 92.26

contrast ratio: 1.089 (more contrast $\Rightarrow$ smaller ratio)

mean: 161.78 (brighter)

standard deviation: 60.41

contrast ratio: 2.678 (less contrast $\Rightarrow$ higher ratio)



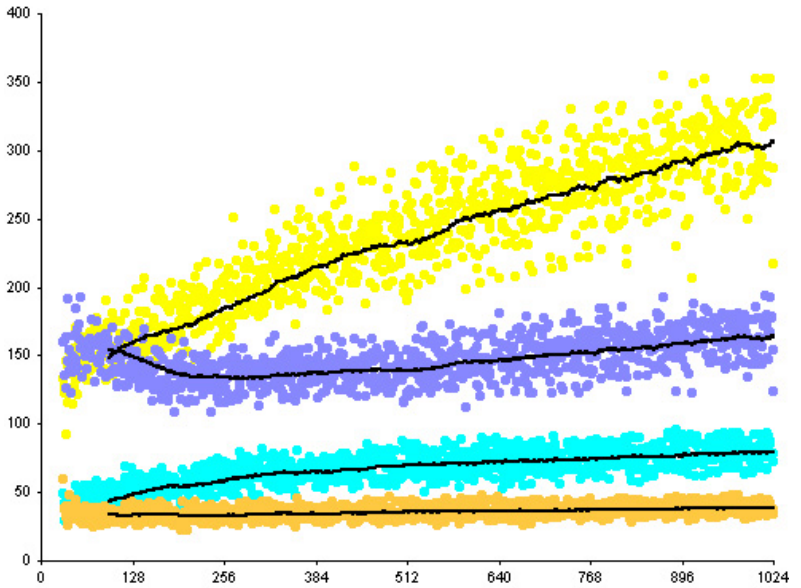
mean: 111.34 (added Gaussian noise to original image)

standard deviation: 82.20

contrast ratio: 1.354 (zero mean noise $\Rightarrow$ about the same ratio)

The Need of Modeling Noise

The diagram below shows measurements (in the scale 0 to 400) for four different algorithms (the input size n varied between 32 and 1024). Each algorithm produced exactly one scattered value, for each n. The sliding mean of these values (taken by using also the last 32 and the next 32 values) produces "arcs", which illustrate "expected values" for the four processes.



Assume we replace input size n by time t; now, only values at earlier time slots are available at t. We cannot estimate anymore the expected value accurately, *having no knowledge about the future at hand*. An estimation error for the bottom-most curve might be smaller than for the top-most curve (i.e., a signal with changing amplitudes) - **IF** prediction is supported.

For accurate estimation of values of a time-dependent process, we have to *model the process* itself, including future noise. An optimum (!) prediction for a given model can be achieved by applying the appropriate Kalman filter.

(B) Random Variables

A *random variable* is the numerical outcome of a random process, such as measuring gray values by a camera within some field of view, or estimating disparities l_p at pixel positions $p \in \Omega$.

Mathematically, a random variable X is a function

$$X : \Psi \to \mathbb{R}$$

where Ψ is the space of all possible outcomes (e.g., all labelings l on Ω) of the corresponding random process.

Normally, it is described by its *probability distribution function*

$$Pr : \wp(\Psi) \to [0, 1]$$

with $Pr(\Psi) = 1$, and $A \subseteq B \subseteq \Psi$ implies $Pr(A) \leq Pr(B)$. Note that $\wp(\Psi)$ denotes the power set (i.e., set of all subsets of Ψ).

Two *events* A, B are *independent* iff $Pr(A \cap B) = Pr(A)Pr(B)$.

It is also convenient to describe a random variable X either by its *cumulative (probability) distribution function*

$$Pr(X \leq a)$$

for $a \in \mathbb{R}$.

"$X \leq a$" is short for the event $\{\psi : \psi \in \Psi \wedge X(\psi) \leq a\} \subseteq \Psi$.

The *probability density function* $f_X : \mathbb{R} \to \mathbb{R}$ satisfies

$$Pr(a \leq X \leq b) = \int_a^b f_X(x)\, \mathrm{d}x$$

Discrete Random Variables

Toss a coin three times at random, and X is the total number of heads

What is Ψ in this case? Specify the probability distribution, density, and cumulative distribution function.

Throw two dice together; let X be the total number of the shown points.

Stereo analysis: In case of calculated disparities at pixel positions in digital stereo image pairs, we may also consider random variables X_p for each pixel position p. Disparities at all pixel positions $p \in \Omega$ define then a matrix (or vector) of discrete random variables, of the same size as discrete points in Ω.

Continuous Random Variables

Measurements X (e.g., of speed, curvature, height, or yaw rate) are often modeled as being continuous random variables

Optic flow calculation: Estimated motion parameters at one pixel position in digital image sequences

Optic flow values at all pixel positions define a matrix (or vector) of continuous random variables.

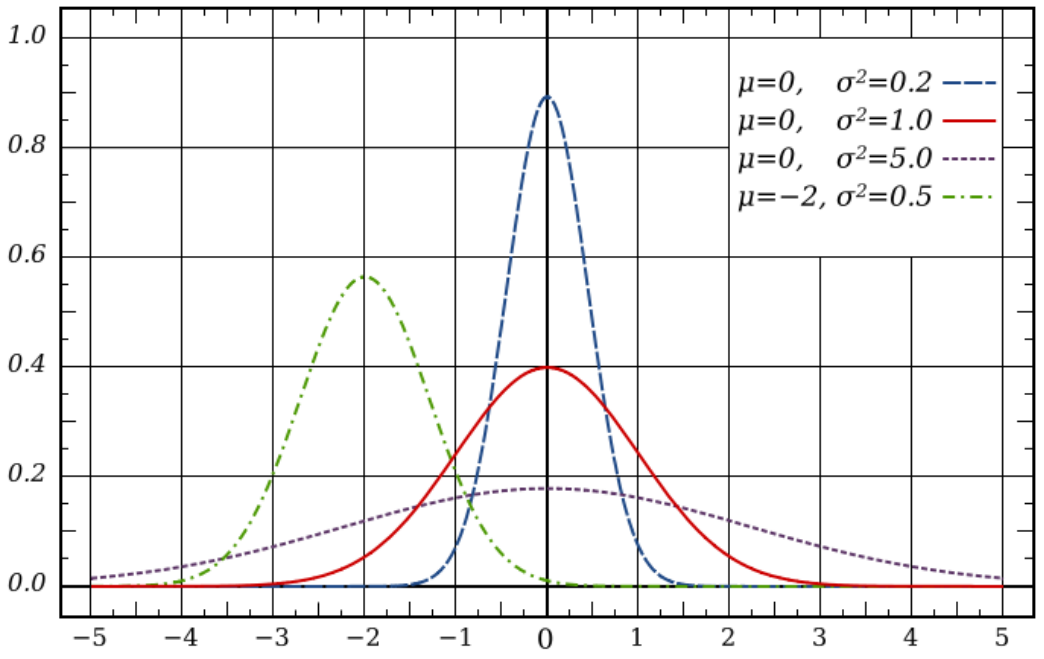
Two Continuous Distributions

Gaussian Distribution (also called *normal distribution*).

A Gaussian random variable X is defined by a probability density

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}D_M^2(x)}$$

for reals μ and $\sigma > 0$ and Mahalanobis distance D_M (for a general definition of this distance function - see below).



(figure reproduced from Wikipedia's common domain)

Continuous Uniform Distribution.

This is defined by an interval $[a, b]$ and the probability density

$$f_X(x) = \frac{\mathrm{sgn}(x-a) - \mathrm{sgn}(x-b)}{2(b-a)}$$

for $\mathrm{sgn}(x) = -1$ for $x < 0$, $= 0$ for $x = 0$, and $= 1$ for $x > 0$.

Parameters of Distributions

Expected Value μ (also called *mean* or *expectation value*).

For a random variable X, this is defined by

$$E[X] = \int_{-\infty}^{\infty} x f_X(x)\, \mathrm{d}x$$

The mean of a random variable equals μ if Gaussian, and $(a+b)/2$ if continuous uniform.

Variance σ^2.

This parameter defines how possible values are spread around the mean μ. It is defined by the following:

$$\mathrm{var}(X) = E[(X-\mu)^2]$$

The variance of a random variable equals σ^2 if Gaussian, and $(b-a)^2/12$ if continuous uniform. We have that

$$E[(X-\mu)^2] = E[X^2] - \mu^2$$

Standard Deviation σ.

Square root of the variance.

Two Examples of Discrete Distributions

Image histograms.

An image histogram $H(u) = \text{card}\{(i,j) : I(i,j) = u\}$ is a discrete version of a probability density function, and the cumulative image histogram

$$C(u) = \sum_{v=0}^{u} H(v)$$

is a discrete version of a cumulative probability distribution function.

Discrete Uniform Distribution.

This is used for modeling that all values of a finite set S are equally probable. For $\text{card}(S) = n > 0$, we have the density function $f_X(x) = \frac{1}{n}$, for all $x \in S$. Let $S = \{a, a+1, \ldots, b\}$ with $n = b - a + 1$. It follows that $\mu = (a+b)/2$ and $\sigma^2 = (n^2-1)/12$. The cumulative distribution function is the step function

$$Pr(X \leq c) = \frac{1}{n} \sum_{i=1}^{n} H(c - k_i)$$

for $k_1, k_2, \ldots, k_n$ being the possible values of X (here: $a, a+1, \ldots, b$), and H is here the *Heaviside step function* (see next page).

Two Discontinuous Functions

Heaviside Step Function (also called *unit step function*). This discontinuous function is defined as follows:

$$H(x) = \begin{cases} 0, & x < 0 \\ \frac{1}{2}, & x = 0 \\ 1, & x > 0 \end{cases}$$

The value $H(0)$ is often of no importance when H is used for modeling a probability distribution. The Heaviside function is used as an antiderivative of the *Dirac delta function* δ; that means $H' = \delta$.

Dirac Delta Function (also called *unit impulse function*). Named after the British physicist Paul Dirac (1902 - 1984), the function $\delta(x)$ is (informally) equals $+\infty$ at $x = 0$, and equals 0 otherwise, and also constrained by the following:

$$\int_{-\infty}^{\infty} \delta(x)\, \mathrm{d}x = 1$$

Note that this is not yet a formal definition of this function (and that is also not needed for the purpose of this lecture).

Example: White noise

Mathematically, white noise of a random time process X_t is defined by zero *mean* $\mu_t = 0$ and an *autocorrelation matrix* (see below) with elements $a_{t_1 t_2} = E[X_{t_1} X_{t_2}] = \sigma^2 \cdot \delta(t_1 - t_2)$, where δ is the *Dirac delta function* (see below) and σ^2 the *variance*.

Random Vectors

The $n > 1$ components X_i of a *random vector* $\mathbf{X} = (X_1, \ldots, X_n)^T$ are random variables, where each X_i is described by its *marginal probability distribution function* $Pr_i : \wp(\Omega) \to [0, 1]$. Functions $Pr_1, \ldots, Pr_n$ define the *joint distribution* for the given random vector. For example, a static camera capturing a sequence of $N \times N$ images, defines a random vector of N^2 components (i.e., pixel values), where sensor noise contributes to the joint distribution.

Covariance Matrix. Let $\mathbf{X}$ and $\mathbf{Y}$ be two random vectors, both with $n > 1$ components (e.g., two N^2 images captured by two static binocular stereo cameras). The $n \times n$ *covariance matrix*

$$\mathrm{cov}(\mathbf{X}, \mathbf{Y}) = E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{Y} - E[\mathbf{Y}])^T]$$

generalizes the concept of variance of a random variable.

Variance Matrix. In particular, if $\mathbf{X} = \mathbf{Y}$, then we have the $n \times n$ *variance matrix*

$$\mathrm{var}(\mathbf{X}) = \mathrm{cov}(\mathbf{X}, \mathbf{X}) = E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T]$$

For example, an image sequence captured by one $N \times N$ camera allows to analyze the $N^2 \times N^2$ variance matrix of this random process. – (Note: the variance matrix is also often called "covariance matrix", meaning the covariance between components of vector $\mathbf{X}$ rather than the covariance between two random vectors $\mathbf{X}$ and $\mathbf{Y}$.)

Mahalanobis distance

For a random vector $\mathbf{X} = (X_1, \ldots, X_n)^T$ with variance matrix $\mathrm{var}(\mathbf{X})$ and mean $\mu = (\mu_1, \ldots, \mu_n)^T$, the *Mahalanobis distance* is defined as

$$D_M(\mathbf{X}) = \sqrt{(\mathbf{X} - \mu)^T \mathrm{var}^{-1}(\mathbf{X})(\mathbf{X} - \mu)}$$

P. C. Mahalanobis (1893 – 1972) introduced (at ISI, Kolkata) this distance in 1936 into statistics.

On `en.wikipedia.org/wiki/Mahalanobis_distance`, there is a good intuitive explanation for this measure. We quote:

"Consider the problem of estimating the probability that a test point in N-dimensional Euclidean space belongs to a set, where we are given sample points that definitely belong to that set. Our first step would be to find the average or center of mass of the sample points. Intuitively, the closer the point in question is to this center of mass, the more likely it is to belong to the set. However, we also need to know how large the set is. The simplistic approach is to estimate the standard deviation of the distances of the sample points from the center of mass. If the distance between the test point and the center of mass is less than one standard deviation, then we conclude that it is highly probable that the test point belongs to the set. The further away it is, the more likely that the test point should not be classified as belonging to the set.

This intuitive approach can be made quantitative by ... "

In detail, the variance matrix $\mathrm{var}(\mathbf{X})$ of a random vector $\mathbf{X}$ is as follows (where μ_i is the expected value of component X_i):

$$\begin{bmatrix} E[(X_1-\mu_1)(X_1-\mu_1)] & E[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & E[(X_1-\mu_1)(X_n-\mu_n)] \\ E[(X_2-\mu_2)(X_1-\mu_1)] & E[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & E[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n-\mu_n)(X_1-\mu_1)] & E[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & E[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$

The main diagonal of $\mathrm{var}(\mathbf{X})$ contains all the variances σ_i^2 of components X_i, for $i = 1, 2, \ldots, n$. All other elements are covariances between two different components X_i and X_j. In general, we have that

$$\mathrm{var}(\mathbf{X}) = E[\mathbf{X}\mathbf{X}^T] - \mu\mu^T$$

where $\mu = E[\mathbf{X}] = (\mu_1, \mu_2, \ldots, \mu_n)^T$.

Autocorrelation Matrix. $\mathbf{A_X} = E[\mathbf{X}\mathbf{X}^T] = [a_{ij}]$ is the (real-valued) *autocorrelation matrix* of the random vector $\mathbf{X}$. Due to the commutativity $a_{ij} = E[X_iX_j] = E[X_jX_i] = a_{ji}$ it follows that this matrix is symmetric (or *Hermitian*), that means

$$\mathbf{A_X} = \mathbf{A_X}^T$$

It can also be shown that this matrix is positive definite, that means, for any vector $\mathbf{w} \in \mathbb{R}^n$, we have that

$$\mathbf{w}^T\mathbf{A_X}\mathbf{w} > 0$$

In particular, that means that $\det(\mathbf{A_X}) > 0$ (i.e., matrix $\mathbf{A_X}$ is non-singular), and $a_{ii} > 0$ and $a_{ii} + a_{jj} > 2a_{ij}$, for $i \neq j$ and $i, j = 1, 2, \ldots, n$.

(C) Linear Dynamic Systems

We assume a continuous *linear dynamic system*, defined by

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

The n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$ specifies the *state* of the process, and $\mathbf{A}$ is the constant $n \times n$ *system matrix*. The notion $\dot{\mathbf{x}}$ is (as common) short for the derivative of $\mathbf{x}$ with respect to time t. Signs and magnitudes of the roots of the characteristic polynomial $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ (i.e., the *eigenvalues* of $\mathbf{A}$) determine the *stability* of the dynamic system. *Observability* and *controllability* are further properties of dynamic systems.

Example 1: A video camera captures an object moving along a straight line. Its centroid (location) is described by coordinate x (on this line), and its move by speed v and a *constant* acceleration a. We do not consider start or end of this motion. The process state is characterized by vector $\mathbf{x} = (x, v, a)^T$, and we have that $\dot{\mathbf{x}} = (v, a, 0)^T$ because of

$$\dot{x} = v, \quad \dot{v} = a, \quad \dot{a} = 0$$

It follows that

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathrm{v} \\ \mathrm{a} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathrm{x} \\ \mathrm{v} \\ \mathrm{a} \end{bmatrix}$$

This defines the 3×3 system matrix $\mathbf{A}$. It follows that

$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^3, \quad \text{i.e.} \quad \lambda_{1,2,3} = 0 \qquad (\text{"very stable"})$$

(D) Goal of a Time-Discrete Filter

Given is a sequence of noisy observations $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{t-1}$ for a linear dynamic system. The goal is to estimate the internal state $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{n,t})$ of the system such that the estimation error is minimized (i.e., we want to look "one step ahead").

Standard Discrete Filtering Model

We assume that the following is given:

- a *state transition matrix* $\mathbf{F}$ which is applied to the (known) previous state $\mathbf{x}_{t-1}$,
- a *control matrix* $\mathbf{B}$ which is applied to a *control vector* $\mathbf{u}_t$, and
- a *process noise vector* $\mathbf{w}_t$ whose joint distribution is a multivariate Gaussian distribution with variance matrix $\mathbf{Q}_t$ and $\mu_{i,t} = E[w_{i,t}] = 0$, for $i = 1, 2, \ldots, n$.

We also assume to have an

- *observation vector* $\mathbf{y}_t$ of state $\mathbf{x}_t$,
- an *observation matrix* $\mathbf{H}$ ("how to observe $\mathbf{y}_t$"?), and
- an *observation noise vector* $\mathbf{v}_t$, whose joint distribution is also a multivariate Gaussian distribution with variance matrix $\mathbf{R}_t$ and $\mu_{i,t} = E[v_{i,t}] = 0$, for $i = 1, 2, \ldots, n$.

Kalman Filter Equations

Initial state $\mathbf{x}_0$ and noise vectors $\mathbf{w}_1, \ldots, \mathbf{w}_t, \ldots, \mathbf{v}_1, \ldots, \mathbf{v}_t, \ldots$, are all assumed to be mutually independent. Let Δt be the actual time difference between t and $t+1$. Recall that

$$e^x = 1 + \sum_{i=1}^{\infty} \frac{x^i}{i!}$$

The *defining equations of a Kalman filter* are as follows:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \quad \text{with} \quad \mathbf{F} = e^{\Delta t \mathbf{A}} = \mathbf{I} + \sum_{i=1}^{\infty} \frac{\Delta t^i \mathbf{A}^i}{i!}$$

$$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

Note that there is often an $i_0 > 0$ such that $\mathbf{A}^i$ equals a matrix having zero in all of its components, for all $i \geq i_0$, thus defining a finite sum only for $\mathbf{F}$.

This model is used for deriving the *standard Kalman filter* - see below. This *discrete model* represents the *continuous linear system*

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

with respect to discrete time samples.

There exist modifications of this discrete model, and related modifications of the Kalman filter (not discussed in these lecture notes).

Continuation of Example 1: We continue with considering linear motion with *constant* acceleration. We have a system vector $\mathbf{x}_t = [x_t, v_t, a_t]^T$ (note: $a_t = a$) and a state transition matrix $\mathbf{F}$ (verify for the provided $\mathbf{A}$!) defined by the following equation:

$$\mathbf{x}_{t+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{x}_t = \begin{bmatrix} x_t + \Delta t \cdot v_t + \frac{1}{2}\Delta t^2 a \\ v_t + \Delta t \cdot a \\ a \end{bmatrix}$$

Note that "time t" is short for time $t_0 + t \cdot \Delta t$, that means, Δt is assumed to be a constant between time slots t and $t + 1$.

For observation $\mathbf{y}_t = (x_t, 0, 0)^T$ (note: we only observe the recent location), we obtain the observation matrix $\mathbf{H}$ defined by the following equation:

$$\mathbf{y}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{x}_t$$

Noise vectors $\mathbf{w}_t$ and $\mathbf{v}_t$ were not part of Example 1, and would be zero vectors under the given ideal assumptions. Control vector and control matrix are also not used in this example, and are zero vector and zero matrix, respectively. (In general, *control* defines some type of influence at time t which is not inherent to the process itself.)

The example needs to be modified by introducing the existence of noise (in process or measurement) for making a proper use of the Kalman filter.

(E) Standard Predict-Update Equations

With $\hat{\mathbf{x}}_{t_1|t_2}$ we denote the *estimate* of state $\mathbf{x}_{t_1}$ based at information available at time t_2. Let $\mathbf{P}_{t_1|t_2}$ be the variance matrix of the error $\mathbf{x}_{t_1} - \hat{\mathbf{x}}_{t_1|t_2}$. The goal is to minimize $\mathbf{P}_{t|t}$ in some defined (i.e., mathematical) way.

Predict Phase of the Filter. In this first phase of a standard Kalman filter, we calculate the predicted state and the predicted variance matrix as follows (using state transition matrix $\mathbf{F}$, control matrix $\mathbf{B}$, and process noise variance matrix $\mathbf{Q}_t$, as given in the model):

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\mathbf{u}_t$$
$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q}_t$$

Update Phase of the Filter. In the second phase of a standard Kalman filter, we calculate the measurement residual vector $\tilde{\mathbf{z}}_t$ and the residual variance matrix $\mathbf{S}_t$ as follows, using observation matrix $\mathbf{H}$ and observation noise variance $\mathbf{R}_t$, as given in the model:

$$\tilde{\mathbf{z}}_t = \mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}$$
$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}_t$$

The updated state estimation vector (i.e., the solution for time t) is calculated in the final (at time t) *innovation step* by the *filter*

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{z}}_t \tag{1}$$

Optimal Kalman Gain

So: which matrix $\mathbf{K}_t$? – The *standard Kalman Filter* is defined by the use of the following matrix $\mathbf{K}_t$ known as the *optimal Kalman gain* (and that's the important result of R. E. Kalman):

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

Optimality.

The use of this optimal Kalman gain in Equation (1) minimizes the mean square error $E[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^2]$, which is equivalent to minimizing the trace (= sum of elements on the main diagonal) of $\mathbf{P}_{t|t}$.

For a proof of the optimality of the Kalman gain, see, for example, online publications about the Kalman filter. This is one mathematical theorem, and it is due to R. E. Kalman.

Note that the updated estimate variance matrix

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}$$

is also required for the predict phase at time $t+1$. The variance matrix $\mathbf{P}_{0|0}$ needs to be initialized at the begin of the process.

Example 2. We modify Example 1. The object (e.g., a car) is still assumed to move (in front of our camera) along a straight line, but now with *random* acceleration a_t (we assume Gaussian distribution with zero mean and variance σ_a^2) between time $t-1$ and time t.

The measurements of the positions of the object are also assumed to be noisy (Gaussian noise with zero mean and variance σ_y^2).

The state vector of this process is given by $\mathbf{x}_t = (x_t, \dot{x}_t)^T$, where $\dot{x}_t$ denotes the speed v_t.

Again, we do not assume any process control (i.e., $\mathbf{u}_t$ is the zero vector). We have that (note: a random acceleration cannot be part of the state anymore; what is matrix $\mathbf{A}$ of the continuous model?)

$$\mathbf{x}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ v_{t-1} \end{bmatrix} + a_t \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t$$

with the variance matrix $\mathbf{Q}_t = \mathrm{var}(\mathbf{w}_t)$ $\left[\text{let } \mathbf{G}_t = (\frac{\Delta t^2}{2}, \Delta t)^T\right]$:

$$\mathbf{Q}_t = E[\mathbf{w}_t \mathbf{w}_t^T] = \mathbf{G}_t E[a_t^2] \mathbf{G}_t^T = \sigma_a^2 \mathbf{G}_t \mathbf{G}_t^T = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}$$

That means, not only $\mathbf{F}$, also $\mathbf{Q}_t$ and $\mathbf{G}_t$ are independent of t, and we just call them $\mathbf{Q}$ and $\mathbf{G}$ for this reason. (In general, matrix $\mathbf{Q}$ is often only specified in form of a diagonal matrix.)

In the assumed example, we measure the position of the object at time t (but not its speed); that means that we have the following:

$$\mathbf{y}_t = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} v_t \\ 0 \end{bmatrix} = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

(note: $\mathbf{v}_t$ is observation noise) with variance matrix

$$\mathbf{R} = E[\mathbf{v_t}\mathbf{v}_t^T] = \begin{bmatrix} \sigma_y^2 & 0 \\ 0 & 0 \end{bmatrix}$$

The initial position equals $\hat{\mathbf{x}}_{0|0} = (0,0)^T$. If this position is accurately known then we have the zero variance matrix

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

Otherwise we have that

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

with a suitably large real $c > 0$.

Now we are ready to deal with $t = 1$. At first, we predict $\hat{\mathbf{x}}_{1|0}$ and calculate its variance matrix $\mathbf{P}_{1|0}$, following the predict equations

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\mathbf{u}_t = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}$$
$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q}_t = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q}$$

Then we calculate the auxiliary data $\tilde{\mathbf{z}}_1$ and $\mathbf{S}_1$, following the update equations

$$\tilde{\mathbf{z}}_t = \mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}$$
$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}$$

This allows us to calculate the optimal Kalman gain $\mathbf{K}_1$ and to update $\hat{\mathbf{x}}_{1|1}$, following the equations

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

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{z}}_t$$

Finally, we calculate $\mathbf{P}_{1|1}$ to prepare for $t = 2$, following the equation

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}$$

Note that those calculations are basic matrix or vector algebra operations, but formally already rather complex, excluding (for common standards) manual calculations. On the other hand, implementation is quite straightforward.

Tuning the Kalman Filter. The specification of the variance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$, or of the constant $c \geq 0$ in $\mathbf{P}_{0|0}$, influences the number of time slots (say, the "convergence") of the Kalman filter such that the predicted states converge to the true states. Basically, assuming a higher uncertainty (i.e., larger $c \geq 0$, or larger values in $\mathbf{Q}_t$ and $\mathbf{R}_t$), increases values in $\mathbf{P_{t|t-1}}$ or $\mathbf{S}_t$; due to the use of the inverse $\mathbf{S}_t^{-1}$ in the definition of the optimal Kalman gain, this decreases values in $\mathbf{K}_t$ and the contribution of the measurement residual vector in the (update) Equation (1).

For example, in the extreme case that we are totally sure about the correctness of the initial state $\mathbf{z}_{0|0}$ (i.e., $c = 0$), and that we do not have to assume any noise in the system and in the measurement processes (as in Example 1), then matrices $\mathbf{P}_{t|t-1}$ and $\mathbf{S}_t$ degenerate to zero matrices; the inverse $\mathbf{S}_t^{-1}$ does not exist (note: consider this case in your program!), and $\mathbf{K}_t$ remains undefined. The predicted state is equal to the updated state; this is the fastest possible convergence of the filter.

Alternative Model for Predict Phase. If we have the continuous model matrix $\mathbf{A}$ for the given linear dynamic process $\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$, it is more straightforward to use the equations

$$\dot{\hat{\mathbf{x}}}_{t|t-1} = \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t$$

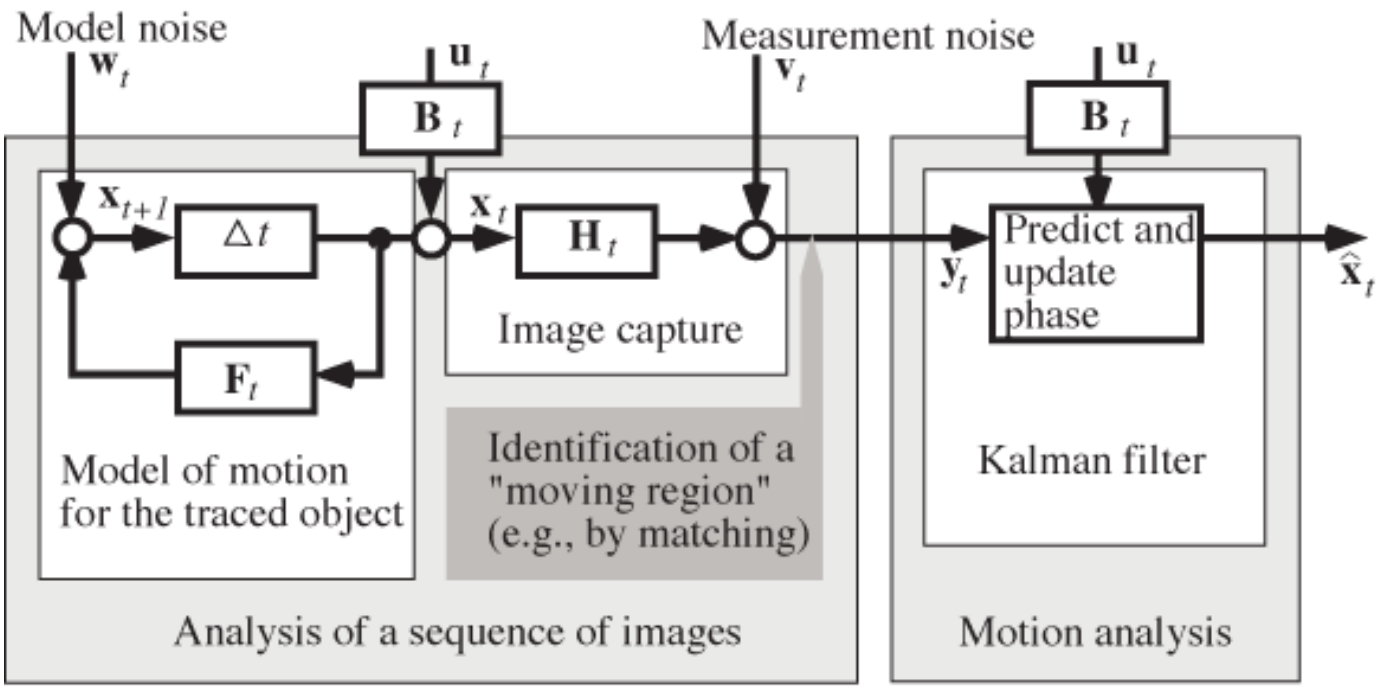
$$\mathbf{P}_{t|t-1} = \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^T + \mathbf{Q}_t$$

rather than those using discrete matrices $\mathbf{F}$. (Of course, this also defines modified matrices $\mathbf{B}$, now defined by the impact of control on the derivatives of state vectors.) This modification in the predict phase does not have any formal consequence on the update phase.

(F) Applications of the Kalman Filter

The Kalman filter had already many “spectacular” applications; for example, it was crucial for the Apollo flights to the moon. In the context of this lecture, we are in particular interested in applications in image analysis, computer vision, or driver assistance.

Here, the time-discrete process is typically a sequence of images (i.e., of fast cameras) or frames (i.e., of video cameras), and the process to be modeled can be something like tracing objects in those images, calculation of optical flow, determining the ego-motion of the capturing camera (or, of the car where the camera has been installed), determining the lanes in the field of view of (e.g., binocular) cameras installed in a car, and so forth. We consider two applications in detail in MI63 and MI64.



Coursework

37.1. [possible lab project] Implement the Kalman filter described in Example 2 (There are links to software downloads on `www.cs.unc.edu/~welch/kalman/`.)

Assume a random sequence of increments $\Delta x_t = x_{t+1} - x_t$ between subsequent positions, e.g. by using a system function RANDOM modeling uniform distribution.

Modify (increase or decrease) the input parameters $c \geq 0$ and the noise parameters in the variance matrices $\mathbf{Q}$ and $\mathbf{R}$.

Discuss the observed impact on the filter's convergence (i.e., the relation between predicted and updated states of the process).

Note that you have to apply the assumed measurement noise model on the generation of the available data $\mathbf{y}_t$ at time t.

37.2. See `www.cs.unc.edu/$\sim$welch/kalman/` for various materials related to Kalman filtering (possibly also follow links specified on this web site, which is dedicated to Kalman filters).

37.3. Show for Example 1, that $\mathbf{F}_t = \mathbf{I} + \Delta t \mathbf{A} + \frac{\Delta t^2}{2}\mathbf{A}^2$.

37.4. Discuss the figure given on the previous page.

37.5. What is the *Mahalanobis dissimilarity measure* $d_M(\mathbf{X}, \mathbf{Y})$ and what is the *normalized Euclidean distance* $d_{e,M}(\mathbf{X}, \mathbf{Y})$, between two random vectors?