Kalman Filters

Olivier White, PhD, Ir

Associate Professor
INSERM - U1093 Cognition, Action, and Sensorimotor Plasticity

Dijon, Feb 2012

This lecture is adapted from Reza Shadmehr's, Gunnar Blohm and Jörn Diedrichsen
Johns Hopkins University, USA
Queens University, CA

Data processing

Given a time series of data points x_1, x_2, … x_N:

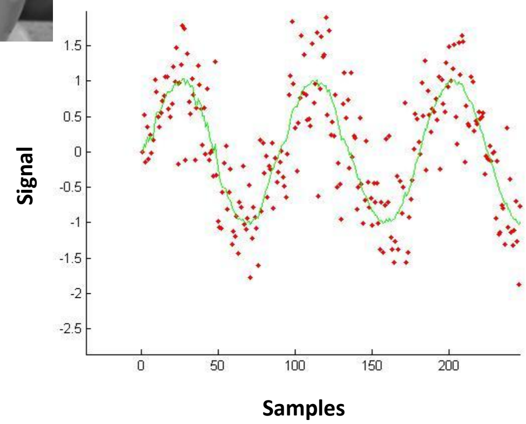
Forecaster
Computes the best guess for x_{N+1}

Smoother
Looks back at the data and computes the best possible x_i taking into account the points before and after x_i

Filter
Provides a correction for x_{N+1}, taking into account x_1, … x_N **and** an inaccurate measurement of x_{N+1}

Filters

A filter is used to throw out noise from interesting and meaningful but uncertain measurements



Kalman filters: intuition

The **Kalman filter** is an algorithm (used since the 1960s) for improving vehicle navigation, that yields an optimized estimate of the system's state (e.g. position and velocity).

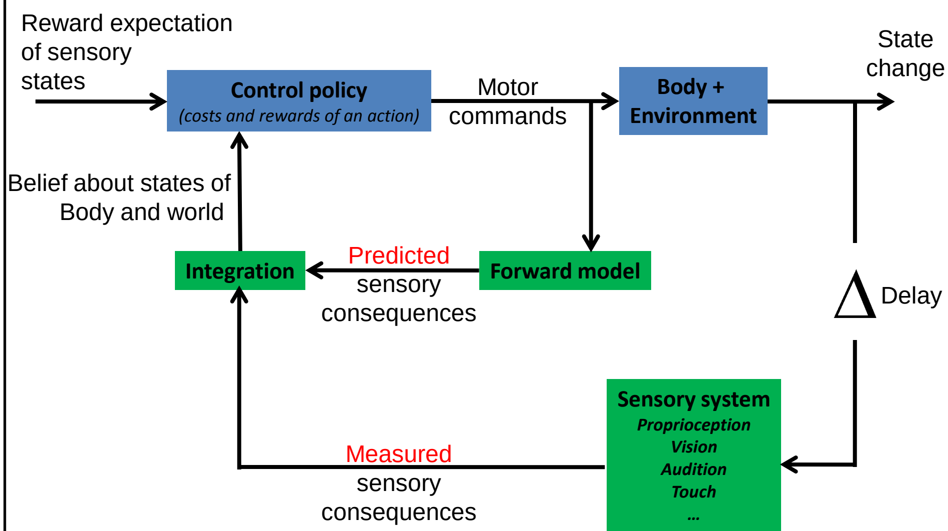
The algorithm works recursively in real time on streams of noisy input observation data (e.g. sensor measurements) and filters out errors using a least-squares curve-fit optimized with a mathematical prediction of future states generated through a modeling of the system's physical characteristics.

The model estimate is compared to the observation and this difference is scaled by a factor known as the **Kalman Gain**, which is then fed back as an input into the model for the purpose of improving subsequent predictions. The gain is adaptive for improved performance.
With a high gain, the filter follows the observations more closely.
With a low gain, the filter follows the model predictions more closely.

This method produces estimates that tend to be closer to the true unknown values than those that would be based on a single measurement alone or the model predictions alone.

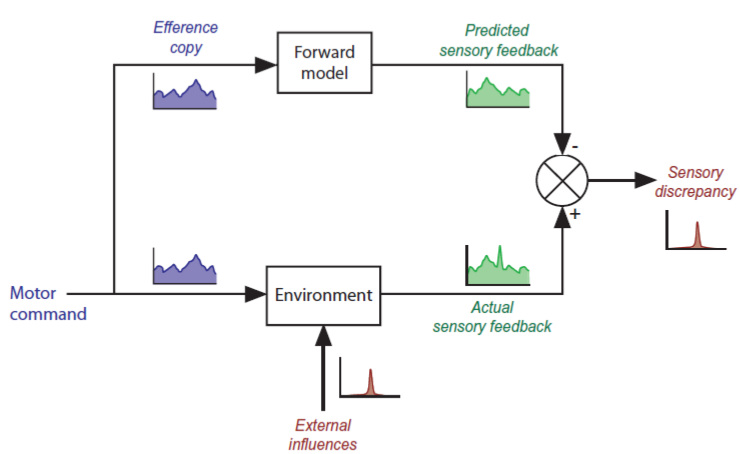
Kalman filters: intuition

The **Kalman filter** may be directly applicable to modeling motor control.

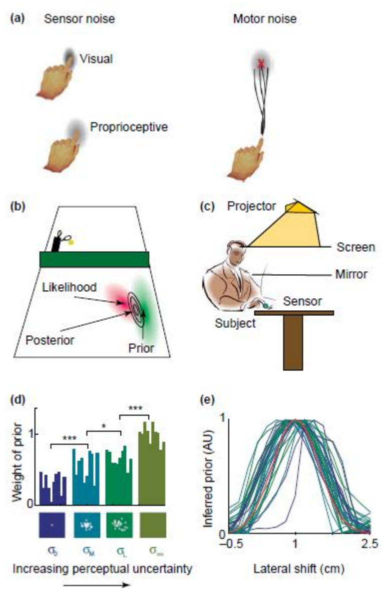


A real example (1)

Sensory cancellation mechanism. On the basis of efference copy, a forward model predicts the sensory feedback that will result from a planned action.
Subtracting this prediction from the actual sensory input reveals an estimate of the sensory feedback due to external influences



A real example (2)



(A) Sensory noise results from different uncertainties. Motor noise will induce variability on the target.

(B) Illustration in tennis: integration of likelihood with priors.

(C) Reliance on priors increase with uncertainty about the environment.

Koerding et al., 2006

Refresher

A Philosophical Essay on Probabilities (1814)

"Probability is the ratio of the number of favorable cases to that of all cases possible."

Suppose we throw a coin twice. What is the probability that we will throw exactly one head?

There are four equally possible cases that might arise:

1. One head and one tail.
2. One tail and one head.
3. Two tails.
4. Two heads.

So there are 2 favorable cases that will give us a head.

The probability that we seek is 2/4.

Pierre Simon de Laplace (1749-1827)

Laplace firmly believed that, in reality, every event is fully determined by general laws of the universe. But nature is complex and we are woefully ignorant of her ways; we must therefore calculate probabilities to compensate for our limitations. Event, in other words, are probable only relative to our meager knowledge. In an epigram that has defined strict determinism ever since, Laplace boasted that if anyone could provide a complete account of the position and motion of every particle in the universe at any single movement, then total knowledge of nature's laws would permit a full determination of all future history. Laplace directly links the need for a theory of probability to human ignorance of nature's deterministic ways. He writes: "So it is that we owe to the weakness of the human mind one of the most delicate and ingenious of mathematical theories, the science of chance or probability." (Analytical Theory of Probabilities, as cited by Stephen J. Gould, Dinosaurs in a Haystack, p. 27.

A quick refresher on basic statistics
And
Matrix algebra

Refresher: Independence

If events **A** and **B** are independent of one another, the probability of their combined existence is the product of their respective probabilities.

$$P(A \wedge B) = P(A, B) = P(A)P(B)$$

Example: Suppose we throw two dice at once.
The probability of getting "snake eyes" (two ones) is 1/36.

$$P(A = 1 \wedge B = 1) = P(A = 1)P(B = 1) = \frac{1}{6} * \frac{1}{6} = \frac{1}{36}$$

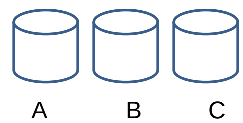
Consequence: The probability that a simple event in the same circumstances will occur consecutively a given number of times is equal to the probability of this simple event raised to the power indicated by this number."

$$P(\underbrace{A \wedge \ldots \wedge A}_{n}) = P(A)^n$$

Conditional probability

When two events depend upon each other, the probability of the compound event is the product of the probability of the first event **and** the probability that, this event having occurred, the second will occur.

Example:
Two urns contain only white balls (value = 1).
One urn contains only black balls (value = 0).
We take one ball from urn C.
What is the probability that it is white?



A	B	C
1	1	0
0	1	1
1	0	1

2 out of 3: $P(C=1) = \frac{2}{3}$

Now, knowing that one white ball has been picked from urn C, the probability of drawing a white ball from urn B is ½.

Bayes rule

Therefore, the probability of drawing 2 white balls from urns C and B is:

$$\begin{aligned} P(C=1, B=1) &= P(C=1|B=1)P(C=1) \\ &= \frac{1}{2} * \frac{2}{3} \\ &= \frac{1}{3} \end{aligned}$$

Conditional probabilities:

$$P(x,y) = P(x|y)P(y)$$
$$P(y,x) = P(y|x)P(x)$$
$$P(y,x) = P(x,y) \qquad \textit{Commutativity}$$

Hence, we derive the Bayes rule:

$$\boxed{P(x|y) = \frac{P(y|x)P(x)}{P(y)}}$$

Bayes rule: Example

In a group of people, 40% are male (M) and 60% are female (F). Unfortunately, 50% of males smoke (S) and 30% of females smoke.

What is the probability that a smoker is male?

We formalize the problem as follows:

$$x = M \text{ or } F$$
$$y = S \text{ or } H$$

$$p(x = M) = 0.4 \qquad P(y = S|x = M) = 0.5$$
$$p(x = F) = 0.6 \qquad P(y = S|x = F) = 0.3$$

$$P(x = M|y = S) = ?$$

Bayes rule: Example

We apply Bayes rule:

$$p(x = M) = 0.4 \qquad P(y = S|x = M) = 0.5$$
$$p(x = F) = 0.6 \qquad P(y = S|x = F) = 0.3$$

$$P(x = M|y = S) = \frac{P(y = S|x = M)P(x = M)}{\boxed{P(y = S)}}$$

$$P(u) = \int P(u|v)P(v)dv$$

In the discrete case, integral becomes a sum:

$$P(y = S) = P(y = S|x = M)P(x = M) + P(y = S|x = F)P(x = F)$$

By replacing:

$$P(x = M|y = S) = \frac{0.5 \times 0.4}{0.5 \times 0.4 + 0.3 \times 0.6} = \frac{0.2}{0.38} = 0.53$$

Expected value and variance

For x and y, scalar random variables and a and b, scalar:

$$E[x] = \int_{-\infty}^{\infty} x p(x) \mathrm{d}x = \sum_x x P(x) = \bar{x}$$

Linear operator: $E[ax + by] = aE[x] + bE[y]$

Expected value and variance

For x and y, scalar random variables and a, scalar:

$$\begin{aligned} var[x] =& E\left[(x - E[x])^2\right] = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x) \mathrm{d}x \\ =& E\left[(x - \bar{x})^2\right] \\ =& E\left[x^2 - 2x\bar{x} + \bar{x}^2\right] \\ =& E[x^2] - 2E[x\bar{x}] + E[\bar{x}^2] \\ =& E[x^2] - 2\bar{x} + \bar{x}^2 \\ =& E[x^2] - \bar{x}^2 \end{aligned}$$

Not a linear operator:

$$\begin{aligned} var[ax] =& E\left[a^2 x^2\right] - a^2 \bar{x}^2 \\ =& a^2 E\left([x^2] - \bar{x}^2\right) \\ =& a^2 var[x] \end{aligned}$$

$$var[x + y] = var[x] + var[y] \text{ if } x \perp\!\!\!\perp y$$

Binomial distribution

Probability distribution of number of successes of n independent yes/no experiments.

Boolean random variables: $x \in \{0,1\}, P(x=1) = q, P(x=0) = 1-q$

$$\vec{x} = (x_1, x_2, \ldots, x_n)$$

$$p(x_1) = q^{x_1}(1-q)^{1-x_1}$$

Probability to get a tail (x_1=1) when throwing a coin is ½:

$$p(x=1) = \frac{1}{2}^1 \left(1 - \frac{1}{2}\right)^0 = \frac{1}{2}$$

If N is # times a trial has succeeded: $n = \sum_{i=0}^{N} x_i$ $\quad x \sim \mathcal{B}(N,q)$

The probability to get N successes in N trials is:

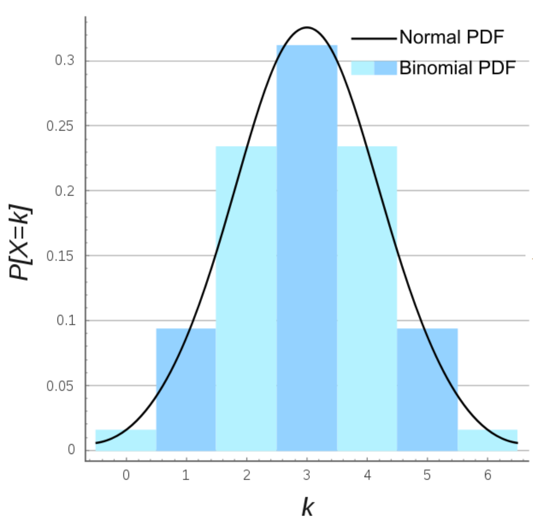
$$P(x) = \binom{N}{n} q^n (1-q)^{N-n} = \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n}$$

Expected values and variances of binomial random variables are:

$$E[x] = Nq \text{ and } var[x] = Nq(1-q)$$

Binomial and normal distribution

If *N is large enough*, then the skew of the distribution is not too great and an excellent approximation to B(n, q) is given by the normal distribution.



What does "N is large enough" mean?? Rules of thumb…

$$Nq > 5 \text{ and } Nq(1-q) > 5$$

$$x \sim \mathcal{N}(Nq, Nq(1-q))$$

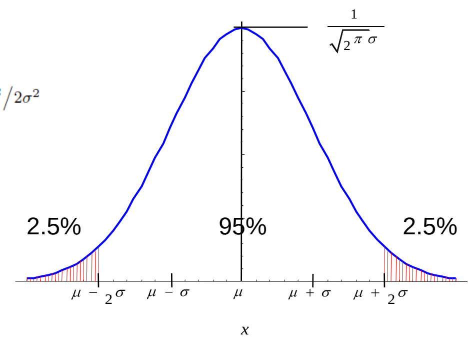
Normal distribution

Motivation: Central Limit Theorem:

"The mean of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed"

Scalar case

If $x \sim \mathcal{N}(\mu, \sigma^2)$ then $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$



Expected values and variances:

$$E[x] = \int_{-\infty}^{\infty} x p(x) \mathrm{d}x = \mu$$

$$\begin{aligned} var[x] &= E\left[(x - E[x])^2\right] \\ &= E\left[(x-\mu)^2\right] \\ &\stackrel{\text{def}}{=} \sigma^2 \end{aligned}$$

95% of the data: $|x - \mu| < 2\sigma^2$

Normal distribution

Vector case

$$x = [x_1, x_2, \ldots, x_n] \text{ for } x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

The vector x also follows a normal distribution with mean μ and covariance matrix C:
The pdf generalizes to the form below:

$$x \sim \mathcal{N}(\mu, C)$$

$$p(x) = \frac{1}{\sqrt{(2\pi)^n det(C)}} e^{\left[-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)\right]}$$

Expected values of a matrix are calculated element-wise

Scalar		**Vector**
$E[x] = \mu$	Expected value	$E[\vec{x}] = \vec{\mu}$
$E\left[(x-\mu)^2\right] = \sigma^2$	Variance	$E\left[(\vec{x}-\vec{\mu})^T(\vec{x}-\vec{\mu})\right] = C$

Covariance matrix

Covariance matrix C

$$cov(x) = \begin{bmatrix} var(x_1) & cov(x_1,x_2) & \cdots & cov(x_1,x_n) \\ cov(x_2,x_1) & var(x_2) & \cdots & cov(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_n,x_1) & cov(x_n,x_2) & \cdots & var(x_n) \end{bmatrix}$$

$$C = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \cdots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}\sigma_n\sigma_1 & \rho_{n2}\sigma_n\sigma_2 & \cdots & \sigma_n^2 \end{bmatrix}$$

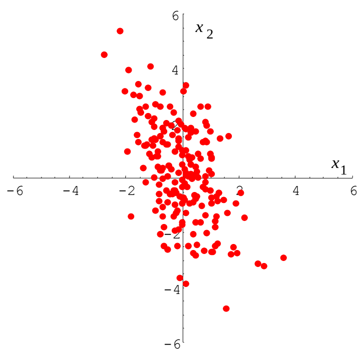
Properties:
- positive semi-definite $\quad x^T C x > 0$
- symmetric $\quad C^T = C$

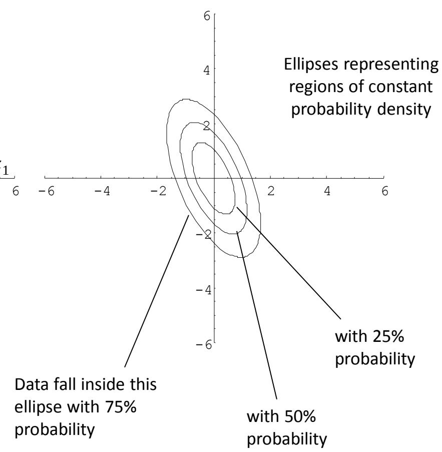
For independent variables, C becomes diagonal. Rhos measure the degree of correlation between x_i and x_j.

$$cov(x_i, y_j) = E\left[(x_i - \mu_i)(x_j - \mu_j)\right]$$
$$cov(x_i, x_i) = E\left[(x_i - \mu_i)^2\right] = \sigma_i^2$$

Covariance matrix: Example

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad x \sim \mathcal{N}(\mu, C) \qquad C = \begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix}$$





Observations about the data:

- x_1 and x_2 are not independent
- Variance of x_2 is greater than x_1
- x_1 and x_2 have a negative correlation

Variance and covariance: scalar

See before: $E[ax+by] = aE[x] + bE[y]$

Variance of the sum of two random variables:

$$\begin{aligned} var[x+y] =& E\left[(x+y-E[x+y])^2\right] \\ =& E\left[(x-\bar{x}+y-\bar{y})^2\right] \\ =& E\left[(x-\bar{x})^2\right] + E\left[(y-\bar{y})^2\right] + 2E\left[(x-\bar{x})(y-\bar{y})\right] \\ =& var[x] + var[y] + 2cov[x,y] \end{aligned}$$

Covariance of two random variables:

$$\begin{aligned} cov[x,y] =& E\left[(x-\bar{x})(y-\bar{y})\right] \\ =& E\left[xy - x\bar{y} + \bar{x}y - \bar{x}\bar{y}\right] \\ =& E[xy] - E[x]\bar{y} + \bar{x}E[y] - \bar{x}\bar{y} \\ =& E[xy] - \bar{x}\bar{y} \end{aligned}$$

Var and cov: vector and matrices

x and y, random vector variables; A and B, constant matrices; a, constant vector

$$\begin{aligned} var[x] =& E\left[(x-\bar{x})(x-\bar{x})^T\right] = E[xx^T] - \bar{x}\bar{x}^T \\ cov[x,y] =& E\left[(x-\bar{x})(y-\bar{y})^T\right] = E[xy^T] - \bar{x}\bar{y}^T \\ cov[Ax,By] =& E\left[A(x-\bar{x})(B(y-\bar{y}))^T\right] \\ =& E\left[A(x-\bar{x})(y-\bar{y})^T B^T\right] \\ =& AE\left[(x-\bar{x})(y-\bar{y})^T\right] B^T \\ =& Acov[x,y]B^T \end{aligned}$$

$$cov[x,y] = cov[x,y]^T$$

$$var[a^T x] = a^T var[x] a$$

$$var[Ax] = cov[Ax,Ax] = Acov[x,x]A^T = Avar[x]A^T$$

Var and cov of vector random variables produce symmetric positive definite matrices

Bases are set.
Derivation of the Kalman Filter

A simple model to illustrate uncertainty

Parameter variance depends only on input selection and noise.

A noisy process produces n data points (x,y) and we form a maximum likelihood estimate of w.

$$y_i^\star = w^{\star T} y_i + \epsilon \qquad\qquad \epsilon \sim N(0, \sigma^2)$$

Star denotes real but unknown parameter value. We assume zero-mean Gaussian noise with some variance.

$$D_1 = [\{x_1, y_{1,1}\}, \{x_2, y_{1,2}\}, \ldots, \{x_n, y_{1,n}\}]$$
$$w_{ML} = \left(X^T X\right)^{-1} X^T y_1$$

This is just a multiple regression.

A simple model to illustrate uncertainty

Parameter variance depends only on input selection and noise.

We run the same noisy process again with the same sequence of x's and we re-estimate w:

$$D_2 = [\{x_1, y_{2,1}\}, \{x_2, y_{2,2}\}, \ldots, \{x_n, y_{2,n}\}]$$
$$w_{ML} = \left(X^T X\right)^{-1} X^T y_2$$

Etc… until n.

The distribution of the resulting w will have a covariance that depends only on the sequence of inputs, the bases that encode those inputs, and the noise sigma.

$$w_{ML} \sim N\left(w^\star, \sigma^2 \left(X^T X\right)^{-1}\right)$$

Illustrate uncertainty: example 1

Input history (for each line, N measurements)

x_1	x_2	y*
1	0	0.5
1	0	0.5
1	0	0.5
1	0	0.5
0	1	0.5

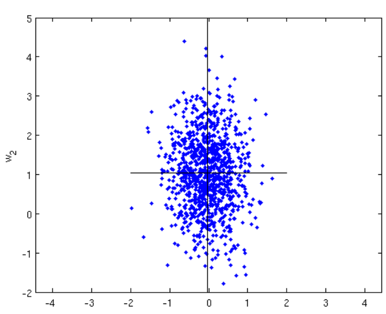
$$\hat{y} = w_1 x_1 + w_2 x_2 = x^T w$$

$$w_{ML} \sim \mathcal{N}\left(E[w], x = \begin{bmatrix} var[w_1] & cov[w_1, w_2] \\ cov[w_2, w_1] & var[w_2] \end{bmatrix}\right)$$
$$w_{ML} \sim \mathcal{N}\left(w^\star, \sigma^2 \left(X^T X\right)^{-1}\right)$$
$$w_{ML} \sim \mathcal{N}\left(\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$X = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

x_1 was "on" 80% of the time; so I'm pretty sure about w_1.

However, x_2 was "on" only once, so I'm uncertain about w_2.



Illustrate uncertainty: example 1

Simple matlab simulation:

```
sig=1;
N=1000;

X=[1 0 ; 1 0 ; 1 0 ; 1 0 ; 0 1]; % inp 1
yr=[0 0 0 0 1];

for i=1:5
    ye(i,:)=yr(i)*ones(1,N)+sig*randn(1,N);
end

w=inv(X'*X)*X'*ye;
plot(w(1,:),w(2,:),'.')
drawline(mean(w(1,:)));
drawline(mean(w(2,:)),'dir','horz');
mean(w')
axis equal
xlabel('w_1');
ylabel('w_2');
```

Illustrate uncertainty: example 2

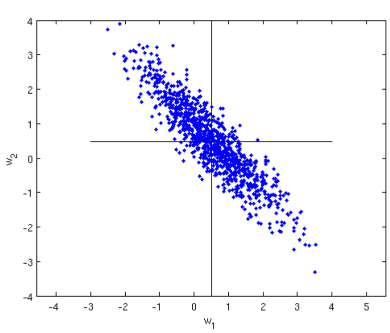
Input history (for each line, N measurements)

x_1	x_2	y^*
1	1	1
1	1	1
1	1	1
1	1	1
1	0	0.5

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$w_{ML} \sim \mathcal{N}\left(w^{\star}, \sigma^2 \left(X^T X\right)^{-1}\right)$$

$$w_{ML} \sim \mathcal{N}\left(\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & 1.25 \end{bmatrix}\right)$$



x_1 and x_2 were "on" mostly together. The weight var-cov matrix shows that what I learned is that:

I do not know individual values of w_1 and w_2 with much certainty.

x_1 appeared slightly more often than x_2, so I'm a little more certain about the value of w_1.

Illustrate uncertainty: example 2

$$w_{ML} \sim \mathcal{N}\left(\begin{bmatrix}0.5\\0.5\end{bmatrix}, \begin{bmatrix}1 & -1\\-1 & 1.25\end{bmatrix}\right)$$

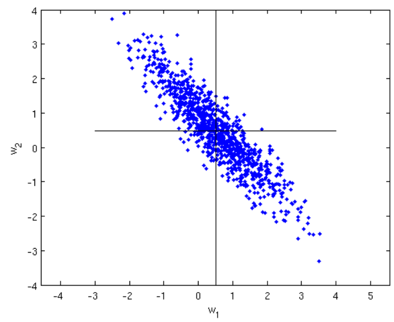
$$cov[x_1, x_2] = \rho_{12}\sigma_1\sigma_2$$

$$\begin{aligned}\rho_{12} =& \frac{cov[x_1, x_2]}{\sigma_1\sigma_2}\\ =& \frac{-1}{\sqrt{1}\sqrt{1.25}}\\ =& -0.894\end{aligned}$$

x_1 and x_2 were "on" mostly together. The weight var-cov matrix shows that what I learned is that:

I do not know individual values of w_1 and w_2 with much certainty.

x_1 appeared slightly more often than x_2, so I'm a little more certain about the value of w_1.



Illustrate uncertainty: example 3

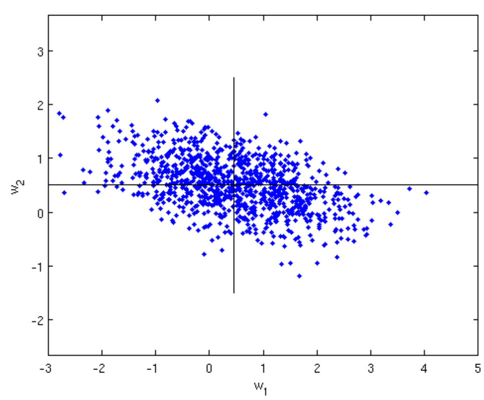
x_1	x_2	y*
0	1	0.5
0	1	0.5
0	1	0.5
0	1	0.5
1	1	1

$$w_{ML} \sim \mathcal{N}\left(\begin{bmatrix}0.5\\0.5\end{bmatrix}, \begin{bmatrix}1.25 & -0.25\\-0.25 & 0.25\end{bmatrix}\right)$$

x_2 was mostly "on". I'm pretty certain about w_2, but I am very uncertain about w_1.

Occasionally x_1 and x_2 were on together, so I have some reason to believe that $w_1 + w_2 = 1$.

$$\begin{aligned}\rho_{12} =& \frac{cov[x_1, x_2]}{\sigma_1\sigma_2}\\ =& \frac{-0.25}{\sqrt{1.25}\sqrt{0.25}}\\ =& -0.447\end{aligned}$$



Effect of uncertainty on learning rate

When you observe an error at time n, the amount that you should change x should depend on how certain you are about x:

The **more certain** you are, the less you should be influenced by the error.

The **less certain** you are, the more you should "pay attention" to the error.

$$x_{n+1} = x_n + \underset{\text{Kalman gain}}{K_n} \underbrace{(y_n - Hx_n)}_{\text{Error}}$$

Rudolph E. Kalman (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45.

Research Institute for Advanced Study
7212 Bellona Ave, Baltimore, MD

Example: Running estimate of average

We have measurements at different time steps $x = [x_1, x_2, \ldots, x_n]$

Goal: compute the average μ_n of **n** data points:

There are two approaches:

1. **Re-compute** the mean $\mu_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i$
2. **Adapt** the current value of the mean… computationally more efficient

$$\begin{aligned}
\mu_{n+1} &= \frac{n}{n+1}\left(\frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n}x_{n+1}\right)\\
&= \frac{n}{n+1}\mu_n + \frac{1}{n+1}x_{n+1}\\
&= \mu_n + K\underbrace{[x_{n+1} - \mu_n]}_{\text{error}} \text{ with } K = \frac{1}{n+1}
\end{aligned}$$

K=Kalman gain:
learning rate decreases as more measurements become available

How far is the current measurement from the calculated mean?

Initial assumptions

Objective: adjust learning gain in order to minimize model uncertainty

Gaussian model:
$$x_n = Ax_{n-1} + Bu_{n-1} + w_{n-1}$$
$$y_n = Hx_n + v_n$$

With:

$x \in \mathbb{R}^n$ True state (position, velocity, force etc

$u \in \mathbb{R}^l$ Command or excitation input

$y \in \mathbb{R}^m$ Measurement of state infected by noise

$A \in \mathbb{R}^{n\times n}$ Transition system matrix (dynamics of the system etc)

$B \in \mathbb{R}^{n\times l}$ Command or input matrix

$H \in \mathbb{R}^{m\times n}$ Observation matrix (Identity if fully observable)

$w \in \mathbb{R}^n, w \sim \mathcal{N}(0, Q)$ Process noise with covariance matrix Q

$v \in \mathbb{R}^n, v \sim \mathcal{N}(0, R)$ Measurement noise with covariance matrix R

Initial assumptions

- My estimate of x **before** I see y at time n, given that I have seen y up to time n-1: $x_{n|n-1}$
- Error at time n: $y_n - Hx_{n|n-1}$
- My estimate of x **after** I see y at time n: $x_{n|n} = x_{n|n-1} + K\left(y_n - Hx_{n|n-1}\right)$
- A <u>priori</u> error (***before*** *I observed the data*): $e_{n|n-1} \stackrel{\text{def}}{=} x_n - x_{n|n-1}$
- A <u>posteriori</u> error (***after*** *I observed the data*): $e_{n|n} \stackrel{\text{def}}{=} x_n - x_{n|n}$
- <u>Prior</u> covariance of parameters error: $P_{n|n-1} \stackrel{\text{def}}{=} cov[e_{n|n-1}, e_{n|n-1}] = E[e_{n|n-1}e^T_{n|n-1}]$ $\stackrel{\text{def}}{=} var[e_{n|n-1}]$
- <u>Posterior</u> covariance of parameters error: $P_{n|n} \stackrel{\text{def}}{=} cov[e_{n|n}, e_{n|n}] = E[e_{n|n}e^T_{n|n}]$ $\stackrel{\text{def}}{=} var[e_{n|n}]$

Covariances catch the uncertainty about our model parameters. Goal is to update the parameters such that **we minimize the uncertainty a posteriori** $P_{n|n}$

Where the trace comes into the game

The trace of parameters covariance matrix is the sum of squared parameters error.
When minimizing a function, always convenient to consider a quadratic cost.

$$Tr(A) \stackrel{\text{def}}{=} \sum_{i=1}^{N} a_{ii}$$ Sum of diagonal elements

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(0, P), P = E[xx^T]$$

$$x = \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} var[x_1] & cov[x_1, x_2] \\ cov[x_2, x_1] & var[x_2] \end{bmatrix}\right)$$

$$\begin{aligned} var[x_1] =& E\left[(x_1 - E[x_1])^2\right] \\ =& \frac{1}{n}\sum_{i=1}^{N}(x_{1i} - E[x_1])^2 \\ =& \frac{1}{n}\sum_{i=1}^{N} x_{1i}^2 \end{aligned}$$

Because mean is 0

$$\begin{aligned} Tr(P) =& var[x_1] + var[x_2] \\ =& \frac{1}{n}\sum_{i=1}^{N}(x_{1i}^2 + x_{2i}^2) \end{aligned}$$

Goal reformulated: find Kalman Gain K such that we minimize the sum of squared error in our parameter estimate.
This sum is the trace of matrix P.
Given an observation y_n, we want to find K such that we minimize the variance of our estimate x:

$$x_{n|n} = x_{n|n-1} + K\left(y_n - Hx_{n|n-1}\right)$$

Development of posterior covariance

By definition: $P_{n|n} = var[e_{n|n}]$ with $e_{n|n} = x_n - x_{n|n}$

$$\begin{aligned} e_{n|n} =& x_n - \left(x_{n|n-1} + K(y_n - Hx_{n|n-1})\right) \\ =& x_n - x_{n|n-1} + Ky_n + KHx_{n|n-1} \\ =& x_n - x_{n|n-1} - KHx_n - Kv_n + KHx_{n|n-1} \\ =& (I - KH)(x_n - x_{n|n-1}) - Kv_n \end{aligned}$$

$$\begin{aligned} P_{n|n} =& var\left[(I - KH)(x_n - x_{n|n-1}) - Kv_n\right] \\ =& (I - KH)var[x_n - x_{n|n-1}](I - KH)^T + Kvar[v_n]K^T \\ =& (I - KH)P_{n|n-1}(I - KH)^T + KRK^T \\ =& \left(P_{n|n-1} - KHP_{n|n-1}\right)(I - H^TK^T) + KRK^T \\ =& P_{n|n-1} - KHP_{n|n-1} - P_{n|n-1}H^TK^T + KHP_{n|n-1}H^TK^T + KRK^T \end{aligned}$$

We have to minimize the trace of the above expression relative to K:

$$\frac{\partial Tr(P_{n|n})}{\partial K} = 0$$

Development of trace of $P_{n|n}$

$$Tr(P_{n|n}) = Tr(P_{n|n-1}) - Tr(KHP_{n|n-1}) - Tr(P_{n|n-1}H^TK^T) + Tr(KHP_{n|n-1}H^TK^T) + Tr(KRK^T)$$

$$Tr(P_{n|n}) = Tr(P_{n|n-1}) - 2Tr(KHP_{n|n-1}) - Tr\left(K(HP_{n|n-1}H^T + R)K^T\right)$$

$$\begin{aligned}\frac{\partial Tr(P_{n|n})}{\partial K} =& 0 \\ =& -2\frac{\partial}{\partial K}Tr(KHP_{n|n-1}) + 2K(HP_{n|n-1}H^T + R) \\ =& -(HP_{n|n-1})^T + K(HP_{n|n-1}H^T + R) \\ =& K(HP_{n|n-1}H^T + R) - P_{n|n-1}H^T\end{aligned}$$

Kalman gain equation

To satisfy this equation: $\boxed{K = P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-1}}$

Lot of uncertainty about the model $P_{n|n-1} \gg R$: We learn a lot from the current error
Pretty sure about my model $P_{n|n-1} \ll R$: We ignore the current error

Update model uncertainty $P_{n|n}$

Having found the Kalman Gain, we need to update the current uncertainty from the prior covariance at time n-1 and K

$$\begin{aligned}P_{n|n} =& (I - KH)P_{n|n-1}(I - KH)^T + KRK^T \\ =& (I - KH)P_{n|n-1}(I - H^TK^T) + KRK^T \\ =& (P_{n|n-1} - KHP_{n|n-1})(I - H^TK^T) + KRK^T \\ =& P_{n|n-1} - P_{n|n-1}H^TK^T - KHP_{n|n-1} + KHP_{n|n-1}H^TK^T + KRK^T \\ =& P_{n|n-1} - P_{n|n-1}H^TK^T - KHP_{n|n-1} + K\left(HP_{n|n-1}H^T + R\right)K^T\end{aligned}$$

$$\text{With } K = P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-1}$$

We need to plug the Kalman Gain in the above and simplify

Update model uncertainty $P_{n|n}$

Which leads to the following:

$$
\begin{aligned}
P_{n|n} = P_{n|n-1} - P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-T} HP_{n|n-1} \\
-P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-1} HP_{n|n-1} \\
+\left(P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-1}\right)\left(HP_{n|n-1}H^T + R\right) \\
\times(\underbrace{HP_{n|n-1}H^T + R}_{S})^{-T} HP_{n|n-1}
\end{aligned}
$$

If we simplify notations, we get:

$$
\begin{aligned}
P_{n|n} =& P - PH^TS^{-T}HP - PH^TS^{-1}HP + PH^TS^{-1}SS^{-T}HP \\
=& P - \underbrace{PH^TS^{-1}}_{K}HP
\end{aligned}
$$

Which finally gives the update equation:

$$
\boxed{
\begin{aligned}
P_{n|n} =& P_{n|n-1} - KHP_{n|n-1} \\
=& (I - KH)P_{n|n-1}
\end{aligned}
}
$$

What we have so far

True state-space model :

$$
\begin{aligned}
x_n =& Ax_{n-1} + Bu_{n-1} + w_{n-1} \\
y_n =& Hx_n + v_n
\end{aligned}
$$

$w \in \mathbb{R}^n, w \sim \mathcal{N}(0, Q)$ Process noise with covariance matrix Q

$v \in \mathbb{R}^n, v \sim \mathcal{N}(0, R)$ Measurement noise with covariance matrix R

Predictor of next state from previous estimated state

$$
x_{n+1|n} = Ax_{n|n} + Bu_n + w_n \qquad \textbf{(1)}
$$

Next state given all previous measures

What we have so far

Updates

The Kalman Gain tells us how much we rely on the error:

$$K = P_{n|n-1}H^T \left(HP_{n|n-1}H^T + R\right)^{-1} \qquad \textbf{(2)}$$

Knowing the Kalman Gain, we can update our estimate of the current state:

$$x_{n|n} = x_{n|n-1} + K\left(y_n - Hx_{n|n-1}\right) \qquad \textbf{(3)}$$

We also need to update the covariance of our measurements:
after we observed a new input y, the uncertainty associated with the weight of that input decreases

$$P_{n|n} = (I - KH)P_{n|n-1} \qquad \textbf{(4)}$$

However, we still lack something… we also need to project our uncertainty about the state because state noise accumulates (Q)

Forecast state noise

From definition: $P_{n+1|n} \stackrel{\text{def}}{=} cov[e_{n|n+1}, e_{n|n+1}] = var[e_{n|n+1}]$

Where $e_{n|n-1}$ is the prior error (before we observe the new data)

$$\begin{aligned} e_{n+1|n} \stackrel{\text{def}}{=}& x_{n+1} - x_{n+1|n-1} \\ =& Ax_n + w_n - Ax_{n|n-1} \\ =& A(x_n - x_{n|n-1}) + w_n \\ =& Ae_{n|n-1} + w_n \end{aligned}$$

Projected state:

$$x_{n+1|n-1} = Ax_{n|n-1}$$

We calculate the variance of the prior error (shifted one state in the future)

$$\begin{aligned} P_{n+1|n} =& var[e_{n|n+1}] \\ =& var\left[Ae_{n|n-1} + w_n\right] \\ =& Avar\left[e_{n|n-1}\right]A^T + Q \end{aligned}$$

$$\boxed{P_{n+1|n} = AP_{n|n}A^T + Q} \qquad \textbf{(5)}$$

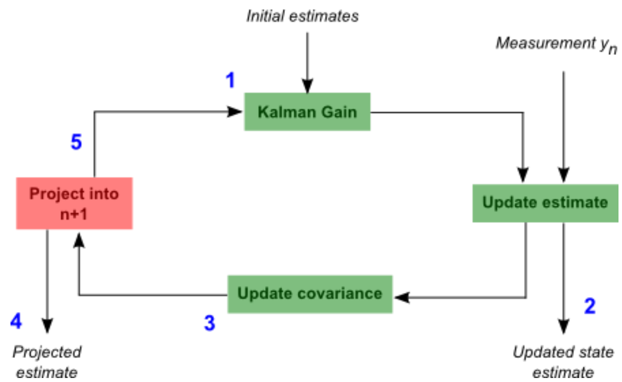
This last equation completes the picture

Summary of the Kalman Filter

How to set the initial values?

$$P_{1|0} = \left(H^T R^{-1} H\right)^{-1}$$

If we don't know anything about everything Before getting y1 then $P_{1|0} = \infty$



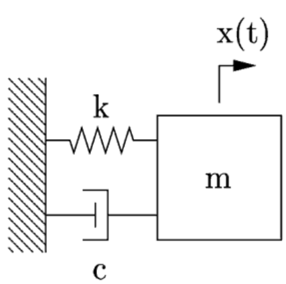
Update Or Correct

1. Kalman gain $\qquad K = P_{n|n-1}H^T\left(HP_{n|n-1}H^T + R\right)^{-1}$
2. Update state estimate $\qquad x_{n|n} = x_{n|n-1} + K\left(y_n - Hx_{n|n-1}\right)$
3. Update covariance of measurements $\qquad P_{n|n} = (I - KH)P_{n|n-1}$

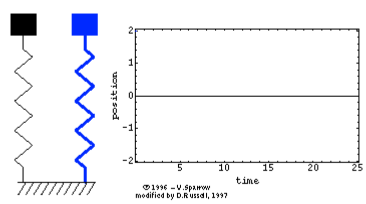
Predict Or Project

4. Project state to n+1 $\qquad x_{n+1|n} = Ax_{n|n} + Bu_n + w_n$
5. Project error covariance to state n+1 $\qquad P_{n+1|n} = AP_{n|n}A^T + Q$

Example: damped mass-spring system



Clean system

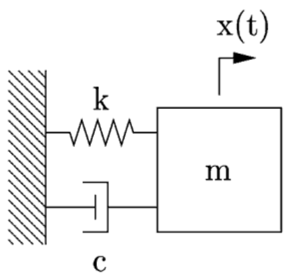


$$\begin{aligned} x_n &= Ax_{n-1} + Bu_{n-1} + w_{n-1} \\ y_n &= Hx_n + v_n \end{aligned} \qquad \begin{aligned} &w \in \mathbb{R}^n, w \sim \mathcal{N}(0, Q) \\ &v \in \mathbb{R}^n, v \sim \mathcal{N}(0, R) \end{aligned}$$

$$\begin{bmatrix} x \\ v \end{bmatrix}_n = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{2c}{m} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_{n-1} + \begin{bmatrix} 0 & 0 \\ 10 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_v \end{bmatrix}_{n-1} + v_{n-1}$$

$$y_n = [1 \ \ 0]_n \begin{bmatrix} x \\ v \end{bmatrix}_n + w_n \qquad Q = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-4} \end{bmatrix}, R = 0.1$$

Example: damped mass-spring system



Initial values

$$u_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, x_0 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

$$P_{1|0} = \left(H^T R^{-1} H\right)^{-1} = R = 0.1$$

This system will be simulated during 10 seconds (time step 25ms) and excited during 500ms after 5 seconds.

$$\begin{aligned} x_n &= A x_{n-1} + B u_{n-1} + w_{n-1} \\ y_n &= H x_n + v_n \end{aligned}$$

$$\begin{bmatrix} x \\ v \end{bmatrix}_n = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{2c}{m} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_{n-1} + \begin{bmatrix} 0 & 0 \\ 10 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_v \end{bmatrix}_{n-1} + v_{n-1}$$

$$y_n = \begin{bmatrix} 1 & 0 \end{bmatrix}_n \begin{bmatrix} x \\ v \end{bmatrix}_n + w_n$$

$$Q = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-4} \end{bmatrix}, R = 0.1$$

Playing with physical parameters

Run matlab simulation:
‘owh_launch_kalman’

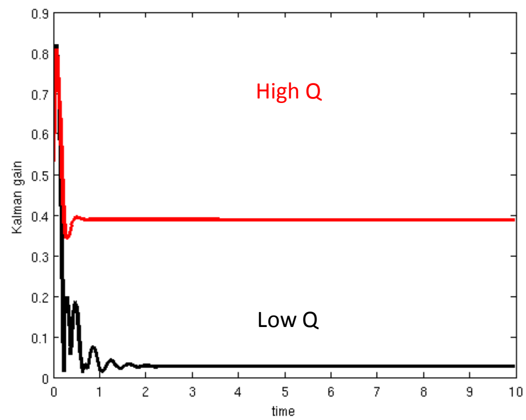
Play with physical parameters (k: 100 to 130 and c: 2 to 3).
Shows position, command and K.

Playing with uncertainties

Q: 10^{-4} to 50×10^{-4}

High noise in the **state** update model produces increased uncertainty in model parameters (k, m, c).

Therefore: high learning rates.

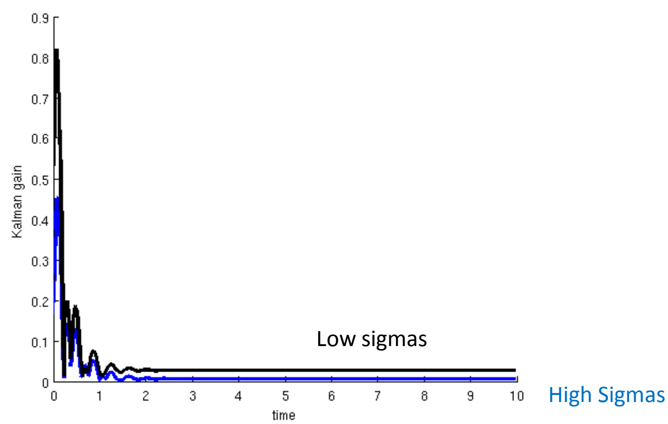


Playing with uncertainties

R: 0.1 to 0.5

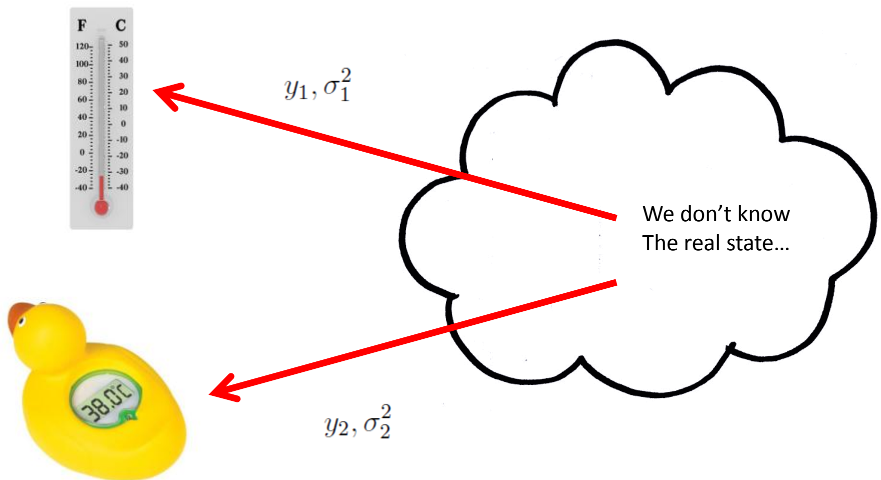
High noise in the **measurements** also increase our uncertainty about parameters. But this increase is small relative to measurement uncertainty.

Therefore, higher measurement noise leads to lower learning rates.



Application: Data fusion

We have two sensors that independently measure something. We would like to combine their measures to form a **better estimate** of the true state.



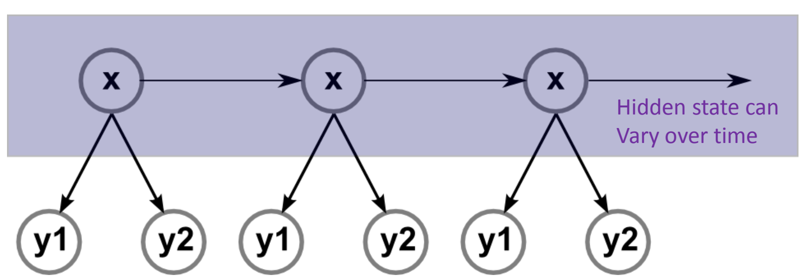
What should the weight be?

Application: Data fusion

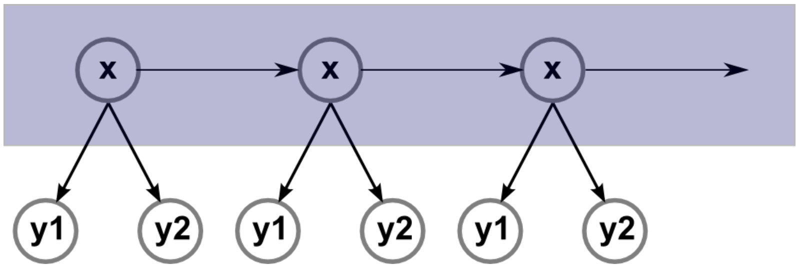
What should the weight be?
Intuitively, the largest on the most reliable.

$$\hat{x} = \frac{\frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} y_2$$

To validate this intuition, we need to formalize our hypothesis about how these data are generated. We design a state-space model:



Application: Data fusion



General state-space equations:

$$x_n = Ax_{n-1} + Bu_{n-1} + w_{n-1} \qquad w \in \mathbb{R}^n, w \sim \mathcal{N}(0, Q)$$
$$y_n = Hx_n + v_n \qquad v \in \mathbb{R}^n, v \sim \mathcal{N}(0, R)$$

This simplifies in our particular case to:

$$x_{n+1} = Ax_n + v_n$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_n = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_n + w_n$$

Initial values:

$$x_{1|0} = 0$$
$$P_{1|0} = \infty$$
$$\vec{y_1} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_1$$

Application: Data fusion

It can be shown that:

$$K = \begin{bmatrix} \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} \\ \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2} \end{bmatrix}$$

The predicted state now becomes:

$$x_{1|1} = x_{1|0} + K(y_1 - Hx_{1|0})$$
$$x_{1|1} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} y_2$$

Covariance of the posterior is (details not shown):

$$P_{1|1} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} < \{\sigma_1^2, \sigma_2^2\}$$

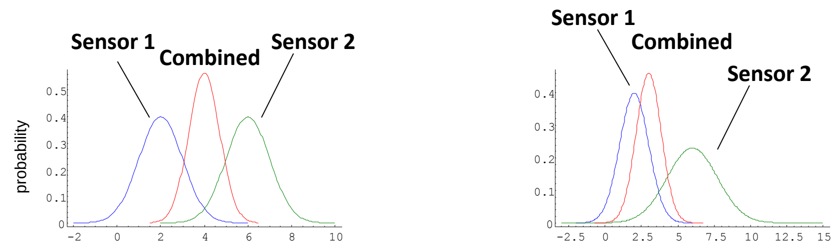
Better than the variance of either sensor taken separately.

Application: Data fusion

Covariance of the posterior is (details not shown):

$$P_{1|1} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} < \{\sigma_1^2, \sigma_2^2\}$$

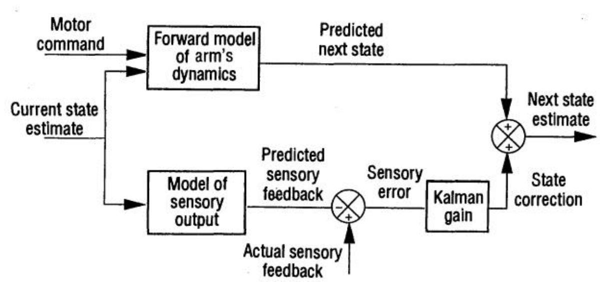
Better than the variance of either sensor taken separately.

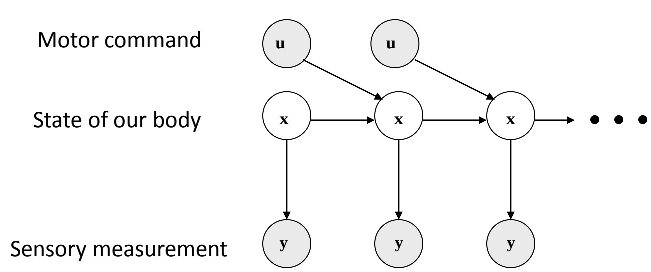


$$\hat{x} \sim \mathcal{N}\left(\underbrace{\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} y_2}, \underbrace{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \right)$$

Mean of the posterior, and its variance

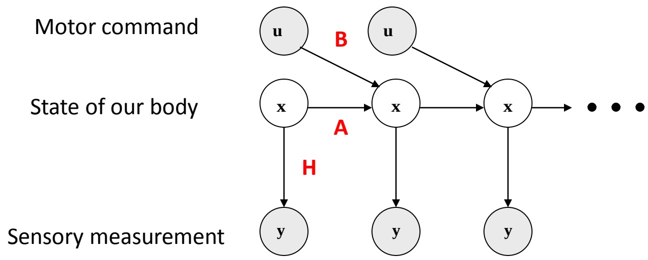
Application of KF in sensorimotor control





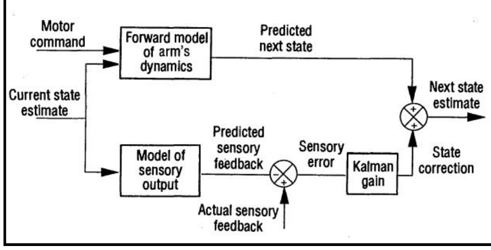
DM Wolpert et al. (1995) *Science* 269:1880

Application of KF in sensorimotor control



$$x_{n+1} = Ax_n + Bu_n + w_n \qquad w \in \mathbb{R}^n, w \sim \mathcal{N}(0, Q)$$
$$y_n = Hx_n + v_n \qquad v \in \mathbb{R}^n, v \sim \mathcal{N}(0, R)$$

The model for estimation of sensory state from sensory feedback



$$\boxed{\begin{aligned} x_{n+1} &= \hat{A}x_n + \hat{B}u_n + w_n \\ y_n &= \hat{H}x_n + v_n \\ B &= 1.4\hat{B} \end{aligned}}$$

For whatever reason, the brain has an incorrect model of the arm. It overestimates the effect of motor commands on changes in limb position.

Next step…

We can optimally integrate measurements.
However, we still don't know how to use that information to generate appropriate commands.

The next step is to integrate Kalman filtering techniques in optimal control in order to find the best series of command under a certain cost function to accomplish a given task.

See you in next chapter…