MC2019 Part 1 Optimal estimation: Maximum likelihood and Bayesian estimation.

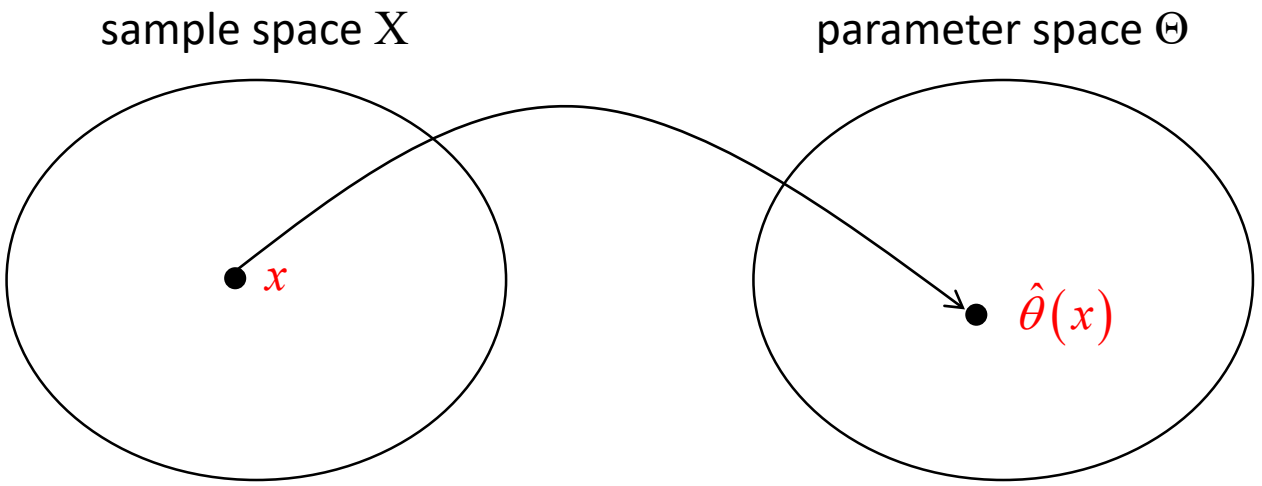
Hirokazu Tanaka

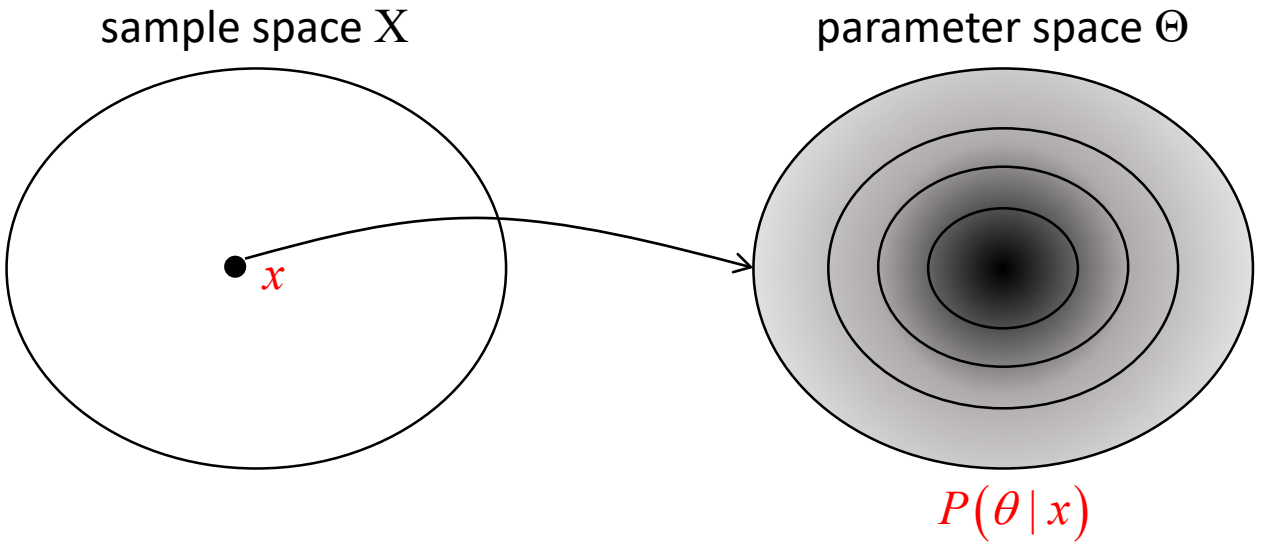
Estimation theory provides a theoretical framework for inference.

In this lecture, we will learn:

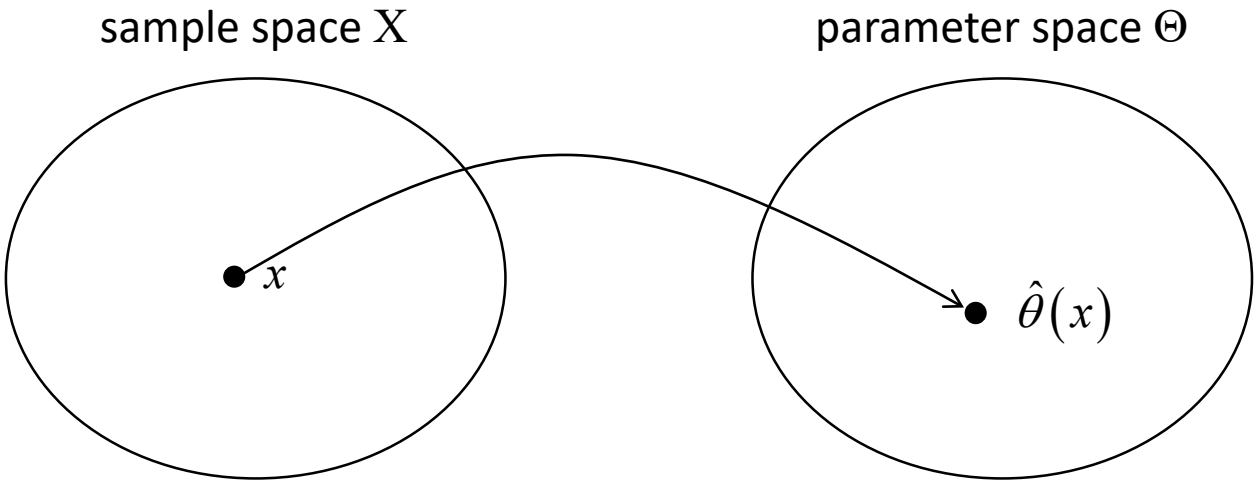
- Maximum-likelihood (ML) estimation
- Cramer-Rao's lower bound
- Bayesian estimation
- Wiener filter
- Kalman filter
- Kalman smoother
- Causal inference
- Information integration
- Postdiction

Classical estimation (point) and Bayesian estimation (density).





Maximum-likelihood (ML) estimation.



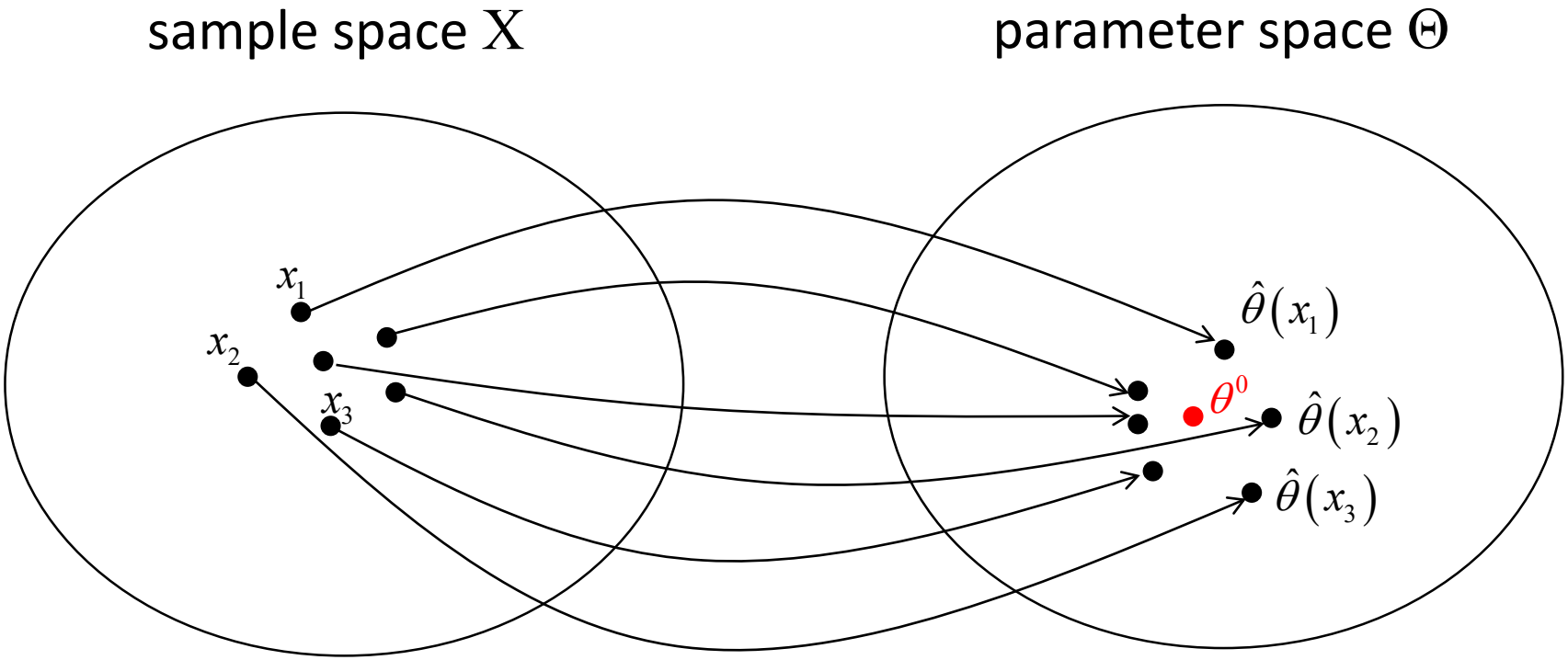
$$\hat{\theta}_{\mathrm{ML}}(x) = \arg\max_{\theta} P(x \mid \theta)$$

$$= \arg\max_{\theta} \log P(x \mid \theta)$$

ML is …

- Asymptotically unbiased (i.e., approaches to a true value).
- Asymptotically efficient (i.e., achieves minimum variance, CRLB).

Maximum-likelihood (ML) estimation.



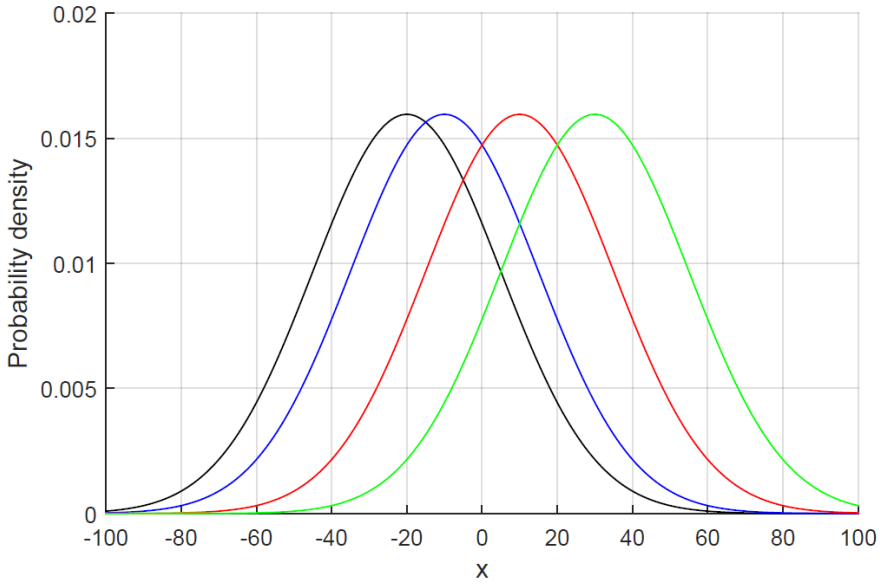
Kullback-Leibler divergence: a distance between two density functions.

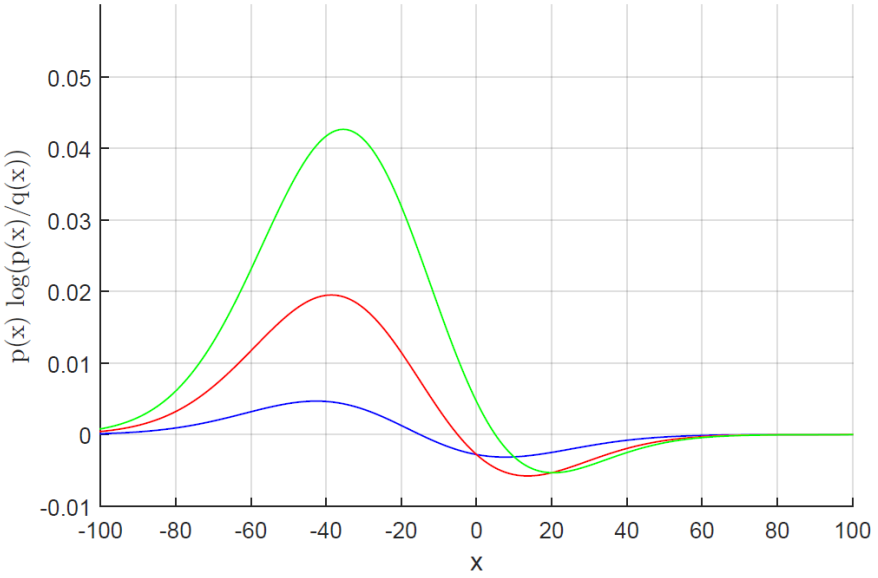
KL divergence for discrete variable

$$D_{\mathrm{KL}}\left[q;p\right]=\sum_i p_i \log\frac{p_i}{q_i}\geq 0$$

KL divergence for continuous variable

$$D_{\mathrm{KL}}\left[q;p\right]=\int dx p\left(x\right)\log\frac{p\left(x\right)}{q\left(x\right)}\geq 0$$





ML as a minimization of KL divergence.

KL divergence between true density $p(x)$ and parametrized density $q(x|\theta)$:

$$D_{\mathrm{KL}}\left[p(x);q(x\mid\theta)\right]=\int dx\,p(x)\log\frac{p(x)}{q(x\mid\theta)}$$
$$=\mathrm{E}\left[\log p(x)\right]-\mathrm{E}\left[\log q(x\mid\theta)\right]$$

Minimization of KL divergence $D_{\mathrm{KL}}\left[p(x);q(x\mid\theta)\right]$

$\Longleftrightarrow$ Maximization of $\mathrm{E}\left[\log q(x\mid\theta)\right]$

Sampling approximation:

$$\mathrm{E}\left[\log q(x\mid\theta)\right]\simeq\frac{1}{N}\sum_{i=1}^{N}\log q(x_i\mid\theta)$$

ML as a minimization of KL divergence.

Sampling approximation:

$$\mathrm{E}\left[\log q\left(x\mid\hat{\theta}\right)\right]-\frac{1}{N}\sum_{i=1}^{N}\log q\left(x_i\mid\hat{\theta}\right)\approx-\left(\hat{\theta}-\theta^0\right)^{\mathrm{T}}E\left[\frac{\partial^2}{\partial\theta\partial\theta^{\mathrm{T}}}\log q\left(x\mid\hat{\theta}\right)\right]\left(\hat{\theta}-\theta^0\right)$$

$$=\left(\hat{\theta}-\theta^0\right)^{\mathrm{T}}I\left(\hat{\theta}\right)\left(\hat{\theta}-\theta^0\right)$$

Fisher information:

$$I\left(\hat{\theta}\right)\equiv\mathrm{E}\left[\frac{\partial\log q\left(x\mid\hat{\theta}\right)}{\partial\theta}\frac{\partial\log q\left(x\mid\hat{\theta}\right)}{\partial\theta^{\mathrm{T}}}\right]=\mathrm{E}\left[-\frac{\partial^2}{\partial\theta\partial\theta^{\mathrm{T}}}\log q\left(x\mid\hat{\theta}\right)\right]$$

In the limit of large samples (infinite N), the ML estimator is unbiased and efficient.

$$\hat{\theta}\sim\mathcal{N}\left(\theta^0,\frac{1}{N}I^{-1}\left(\hat{\theta}\right)\right)$$

ML is …

- Asymptotically unbiased (i.e., approaches to a true value).
- Asymptotically efficient (i.e., achieves minimum variance, CRLB).

Cramer-Rao lower bound: scalar parameter case.

Suppose a random variable $X \sim p(x;\theta)$, where θ is fixed but unknown. Assume that $p(x;\theta)$ satisfies the "regularity" condition:

$$\mathrm{E}\left[\frac{\partial}{\partial\theta}\log p\left(x\,|\,\theta\right)\right] = 0,$$

where the expectation is with respect to $p(x;\theta)$. Then the variance of any unbiased estimator $\hat{\theta}$ satisfies

$$\mathrm{Var}\left[\hat{\theta}\right] \geq \frac{1}{I\left(\theta\right)}$$

Fisher information:

$$I\left(\theta\right) \equiv \mathrm{E}\left[\left(\frac{\partial \log p\left(x\,|\,\theta\right)}{\partial\theta}\right)^2\right] = \mathrm{E}\left[-\frac{\partial^2 \log p\left(x\,|\,\theta\right)}{\partial\theta^2}\right]$$

Cramer-Rao lower bound: vector case.

Suppose a random variable $X \sim p(x|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is fixed but unknown. Assume that $p(x|\boldsymbol{\theta})$ satisfies the "regularity" condition:

$$\mathrm{E}\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p\left(x \mid \boldsymbol{\theta}\right)\right] = 0,$$

where the expectation is with respect to $p(x;\theta)$. Then the variance of any unbiased estimator $\widehat{\boldsymbol{\theta}}$ satisfies

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] \geq \mathbf{I}^{-1}\left(\boldsymbol{\theta}\right)$$

Fisher information matrix:

$$\left\{\mathbf{I}\left(\boldsymbol{\theta}\right)\right\}_{ij} \equiv \mathrm{E}\left[\frac{\partial \log p\left(x \mid \boldsymbol{\theta}\right)}{\partial \theta_i} \frac{\partial \log p\left(x \mid \boldsymbol{\theta}\right)}{\partial \theta_j}\right] = \mathrm{E}\left[-\frac{\partial^2 \log p\left(x \mid \theta\right)}{\partial \theta_i \partial \theta_j}\right]$$

Width estimation: optimal integration of vision and haptics.

ML estimation of object width from vision and haptics:

x_1: visually perceived width, σ_1: visual uncertainty
x_2: haptically perceived width σ_2 : haptic uncertainty

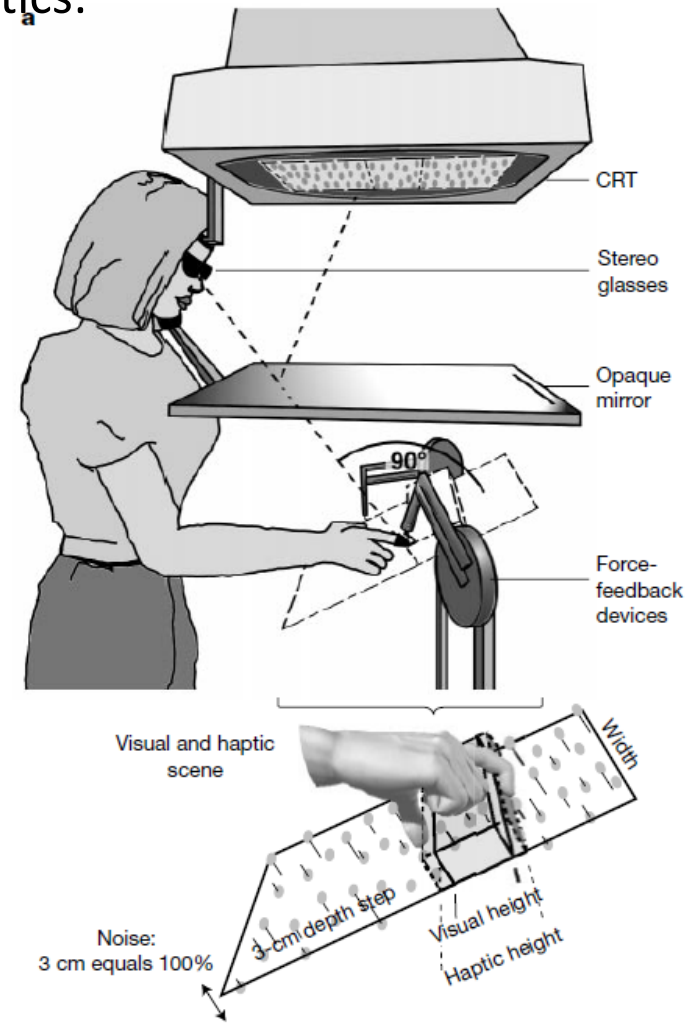
$$p(x_V, x_H \mid \mu) = p(x_V \mid \mu)\, p(x_H \mid \mu)$$

$$\propto \exp\left(-\frac{(x_V - \mu)^2}{2\sigma_V^2} - \frac{(x_H - \mu)^2}{2\sigma_H^2} \right)$$

$$\propto \exp\left(-\frac{(\hat{\mu} - \mu)^2}{2\sigma^2} \right)$$

$$\hat{\mu} = \frac{\sigma_H^2}{\sigma_V^2 + \sigma_H^2} x_V + \frac{\sigma_V^2}{\sigma_V^2 + \sigma_H^2} x_H$$

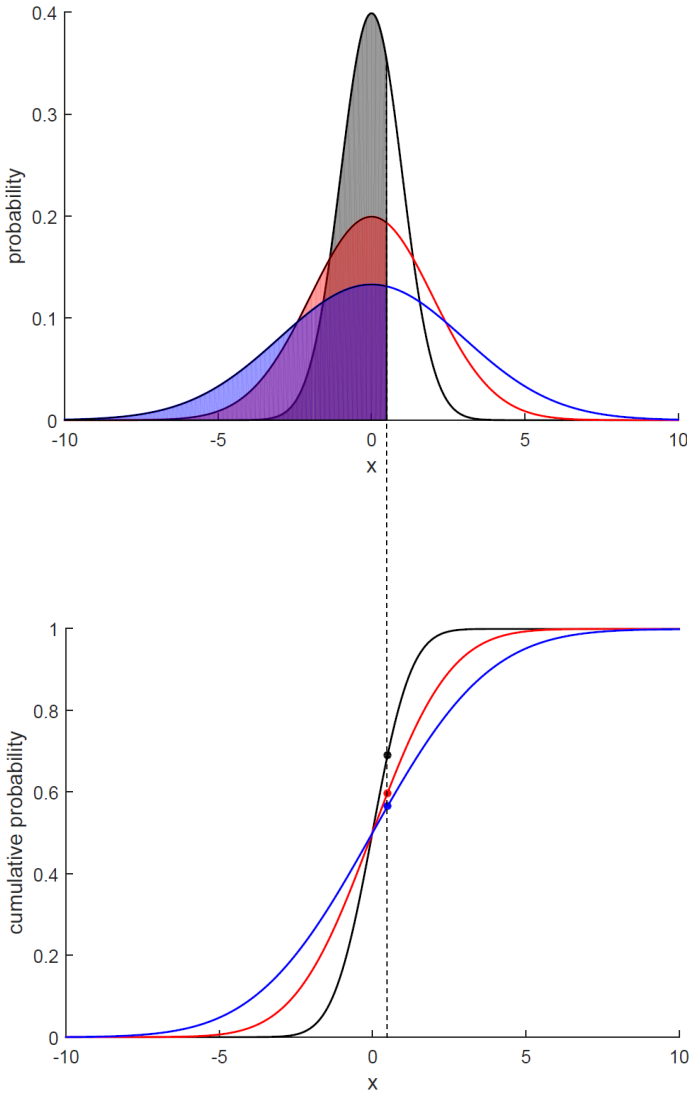
$$\frac{1}{\sigma^2} = \frac{1}{\sigma_V^2} + \frac{1}{\sigma_H^2}$$



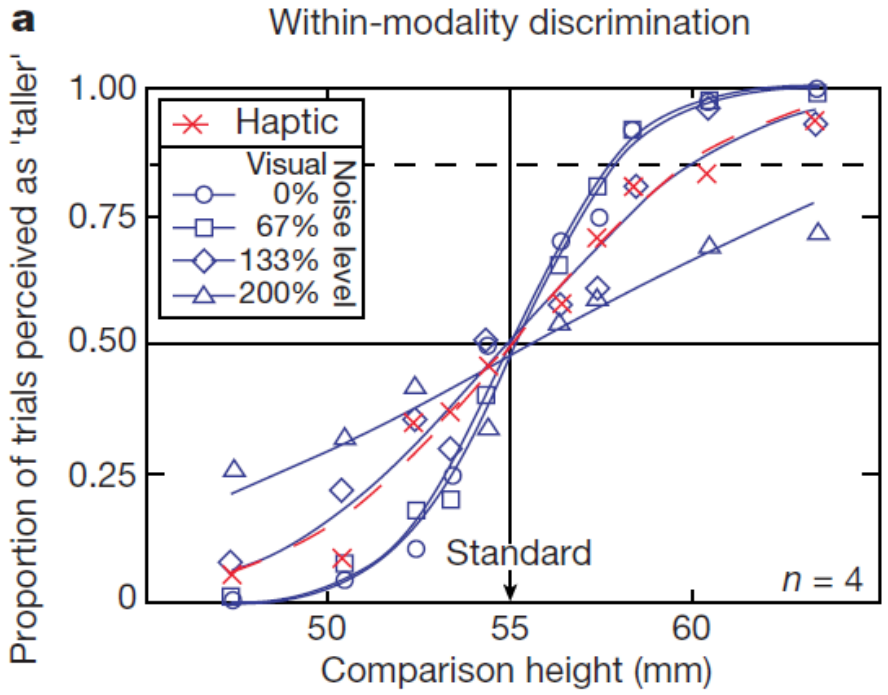
Ernst & Banks (2002) Nature

Width estimation: optimal integration of vision and haptics.

Density functions (sensory uncertainty)



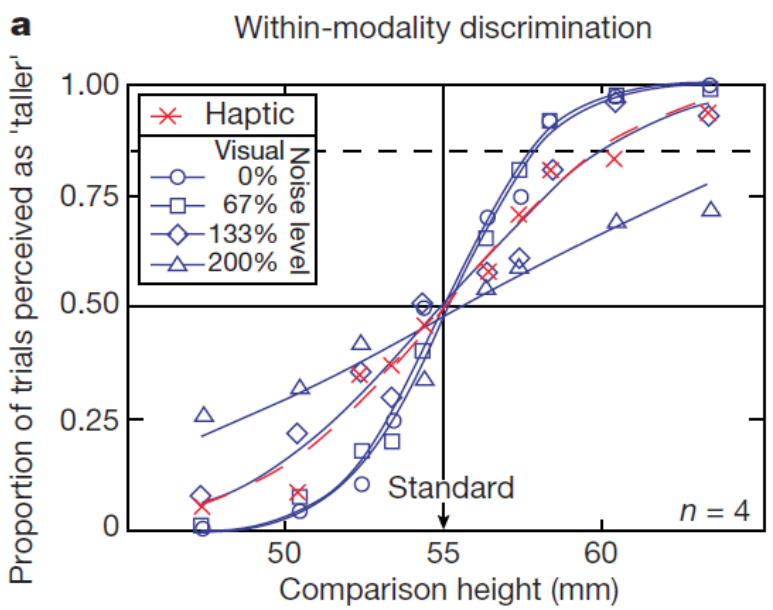
Distribution functions (human response)

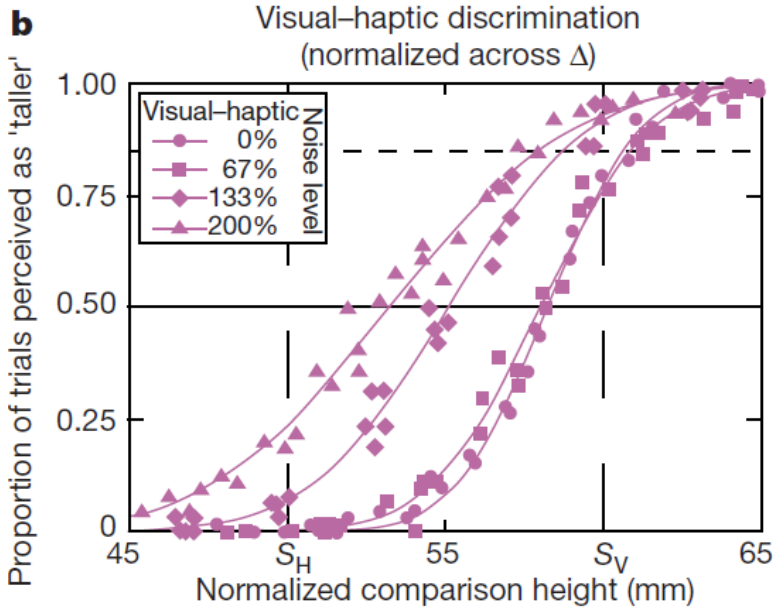


By examining human forced choice, the sensory density function can be estimated.

Ernst & Banks (2002) Nature

Width estimation: optimal integration of vision and haptics.





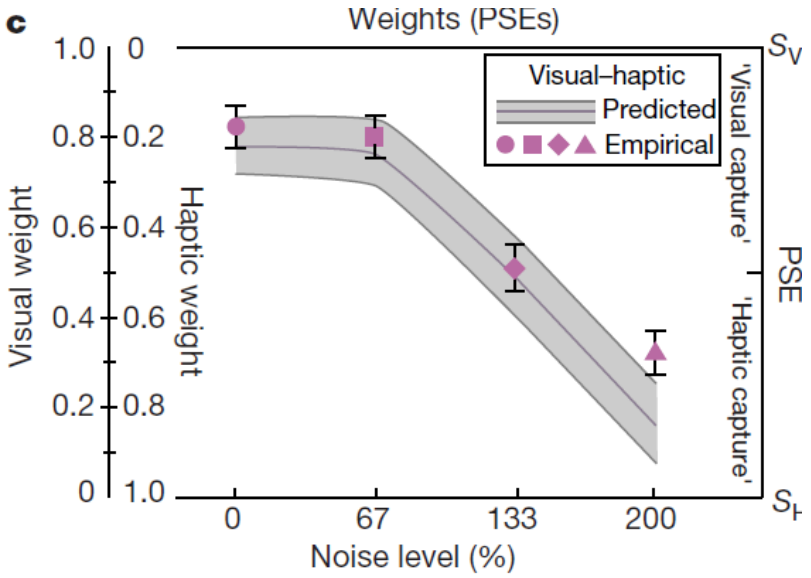
mean (x_V) and variance (σ_V^2) of vision
mean (x_H) and variance (σ_H^2) of haptics

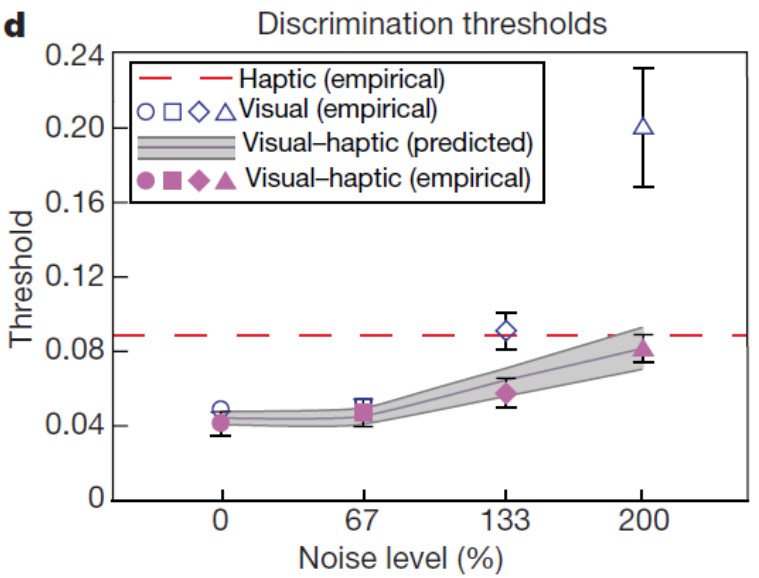
mean (x) and variance (σ_V) of multi-modal integration

Human performance in visual-haptic discrimination (RIGHT) is predicted by maximum-likelihood integration of those in visual and haptic discrimination measured separately (LEFT).

Ernst & Banks (2002) Nature

Width estimation: optimal integration of vision and haptics.



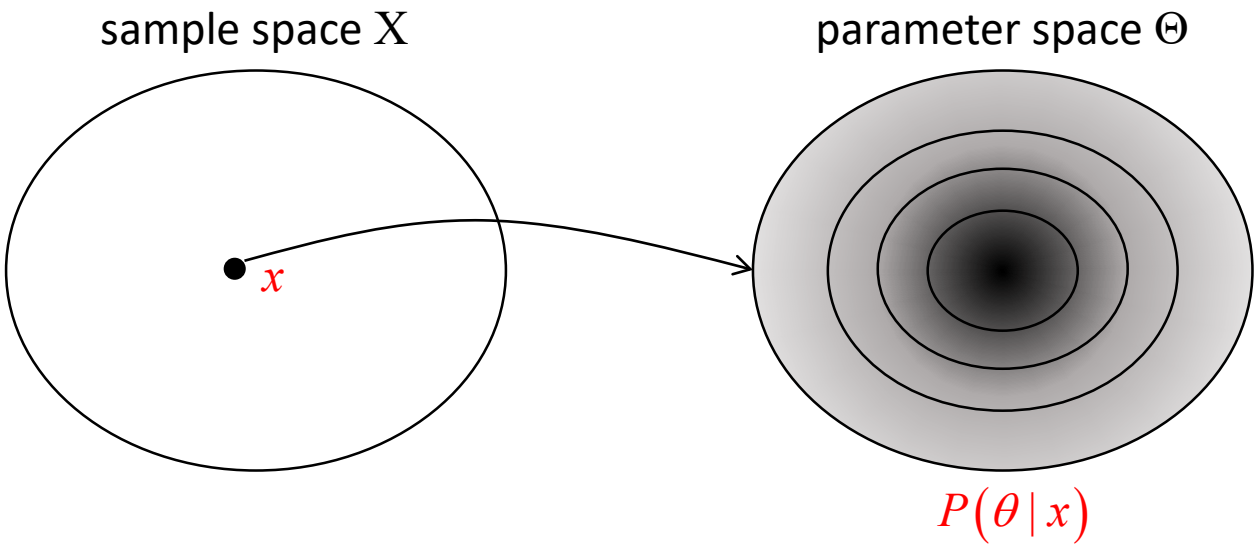


$$x = \frac{\sigma_H^2}{\sigma_V^2 + \sigma_H^2} x_1 + \frac{\sigma_V^2}{\sigma_V^2 + \sigma_H^2} x_H$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_V^2} + \frac{1}{\sigma_H^2}$$

Ernst & Banks (2002) Nature

Bayes' theorem: mapping observation to probability density.



$$P(\theta \mid x) = \frac{P(x \mid \theta) P(\theta)}{P(x)}$$

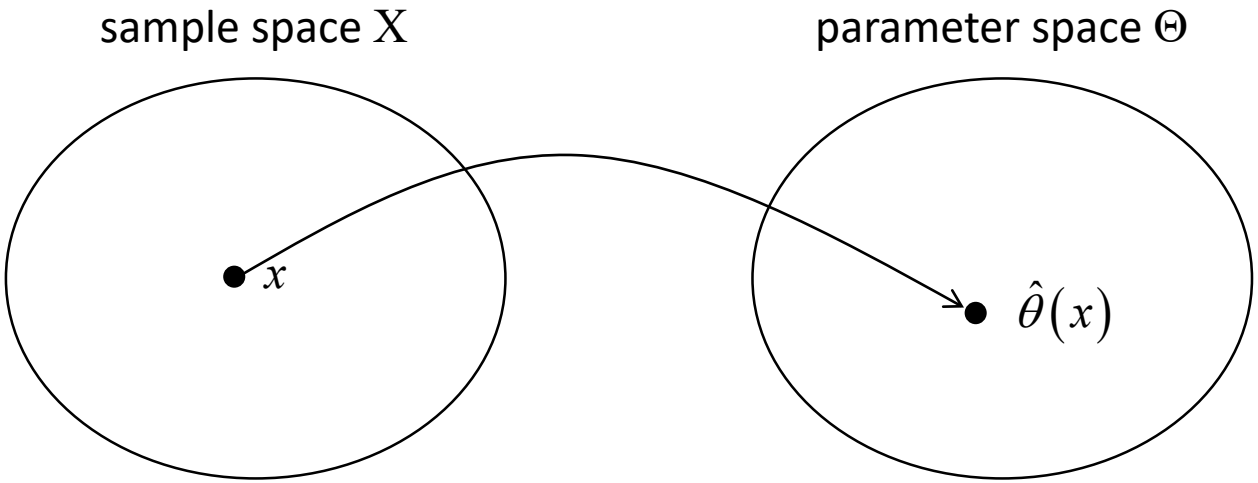
$P(\theta \mid x)$: Posterior probability

$P(x \mid \theta)$: Likelihood

$P(\theta)$: Prior probability

$P(x)$: Marginal probability

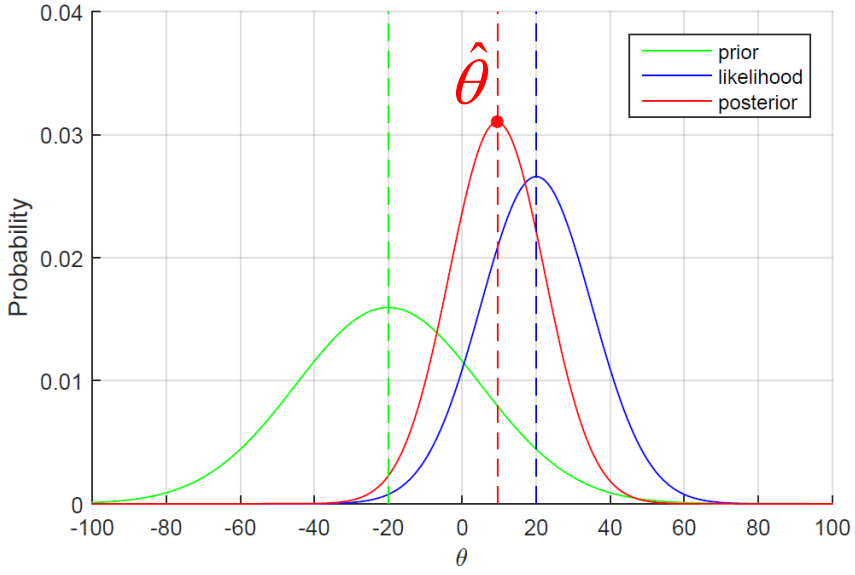
Bayesian estimation: Maximum A Posteriori (MAP).



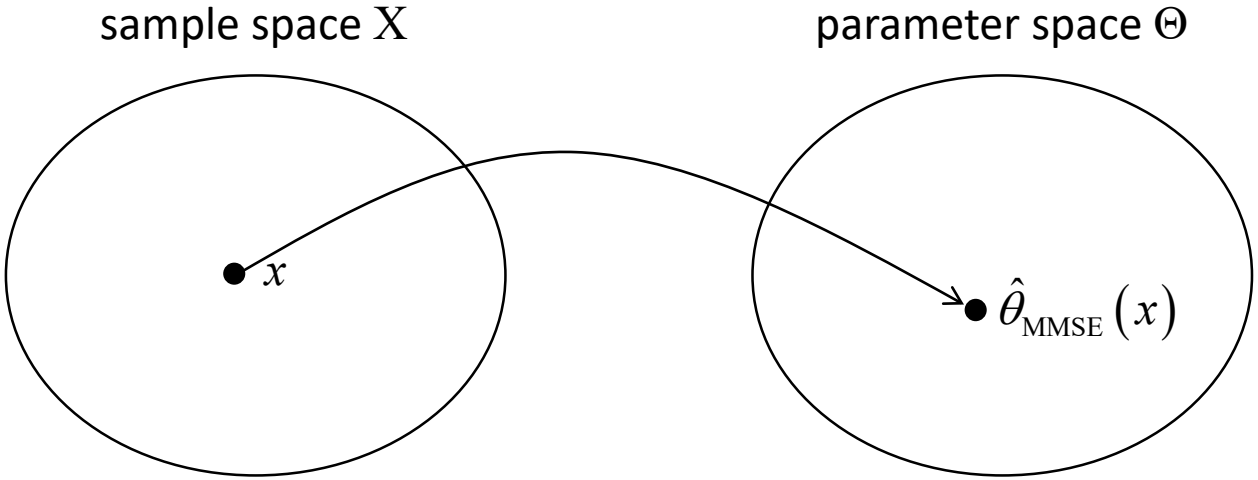
$$\hat{\theta}(x) = \arg\max_{\theta} P(\theta \mid x)$$

$$= \arg\max_{\theta} \frac{P(x \mid \theta) P(\theta)}{P(x)}$$

$$= \arg\max_{\theta} P(x \mid \theta) P(\theta)$$



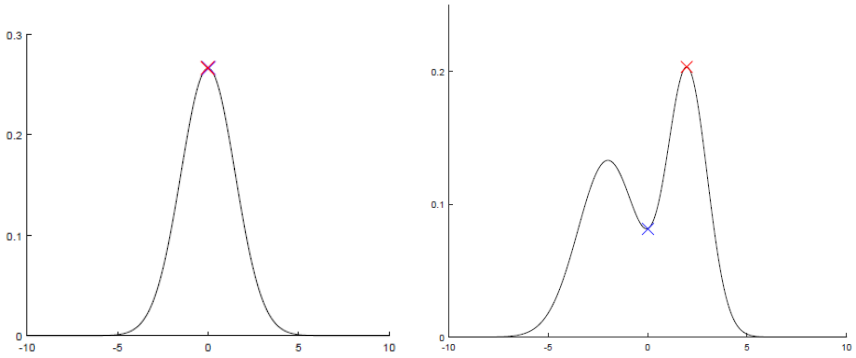
Bayesian estimation: Mean minimum-squared error estimator.



$$\hat{\theta}_{\mathrm{MMSE}}(x) = \arg\min_{\hat{\theta}} \mathrm{E}_{\theta}\left[\left(\theta-\hat{\theta}\right)^2 \middle| x\right]$$

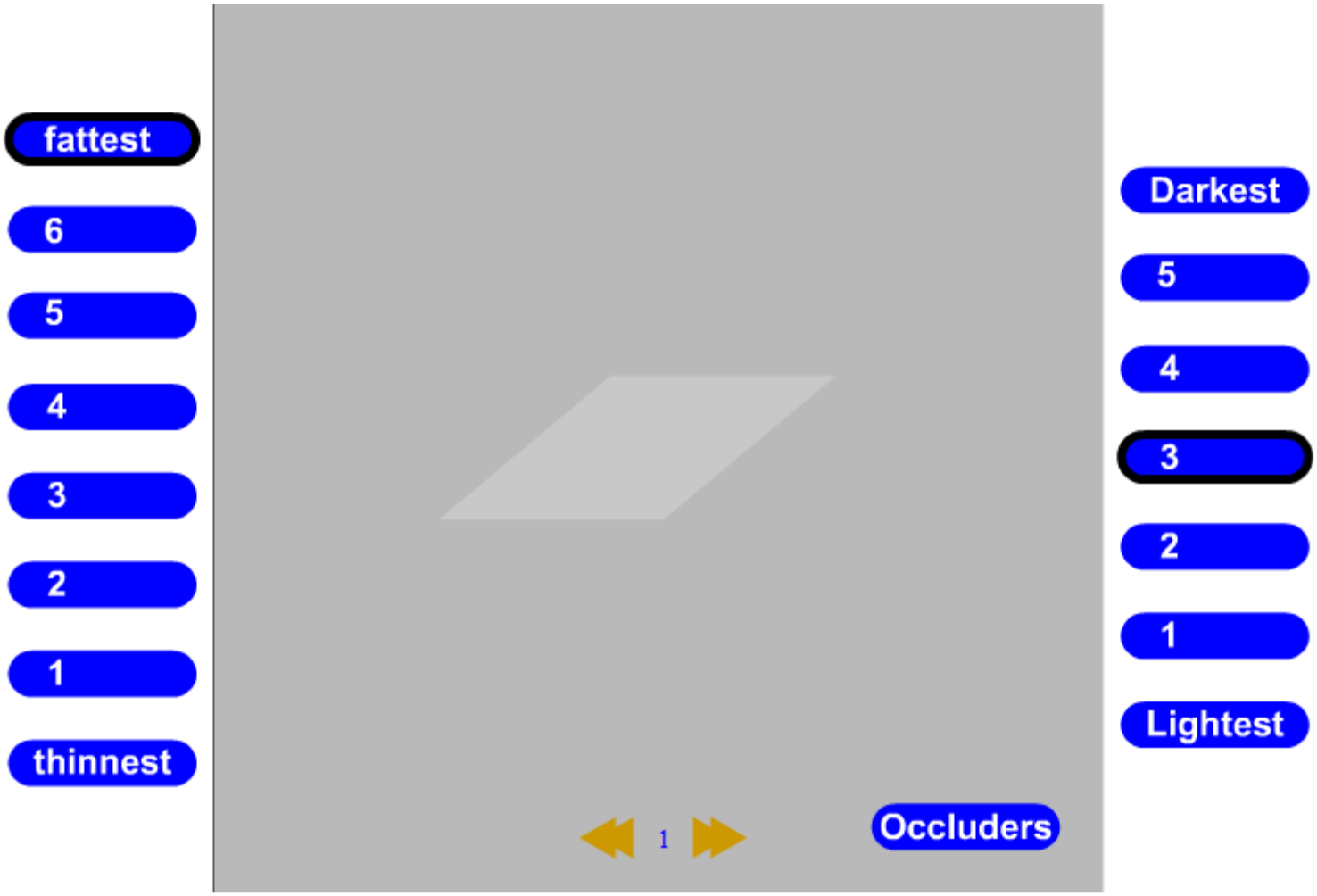
$$= \arg\min_{\hat{\theta}} \int \left(\theta-\hat{\theta}\right)^2 p\left(\theta \middle| x\right) d\theta$$

×: MAP, ×: MMSE



$$\hat{\theta}_{\mathrm{MMSE}}(x) = \int \theta\, p\left(\theta \middle| x\right) d\theta = E_{\theta}\left[\theta \middle| x\right]$$

Motion illusion as optimal percepts.



Use the left blue buttons to change the size/shape of the moving figure; use the right blue buttons to change the color of the background. Use the yellow buttons to control the speed.

http://www.cs.huji.ac.il/~yweiss/Rhombus/rhombus.html

Motion illusion as optimal percepts.

Prior density: most of objects have small velocity (not moving).

$$P\left(v_x, v_y\right) \propto \exp\left\{-\frac{1}{2\sigma_v^2}\left(v_x^2 + v_y^2\right)^2\right\}$$

Likelihood: Probability of observed image with given velocity (v_x, v_y).

Lucas-Kanade image model: $I\left(x + v_x\Delta t, y + v_y\Delta t, t + \Delta t\right) \approx I\left(x, y, t\right)$

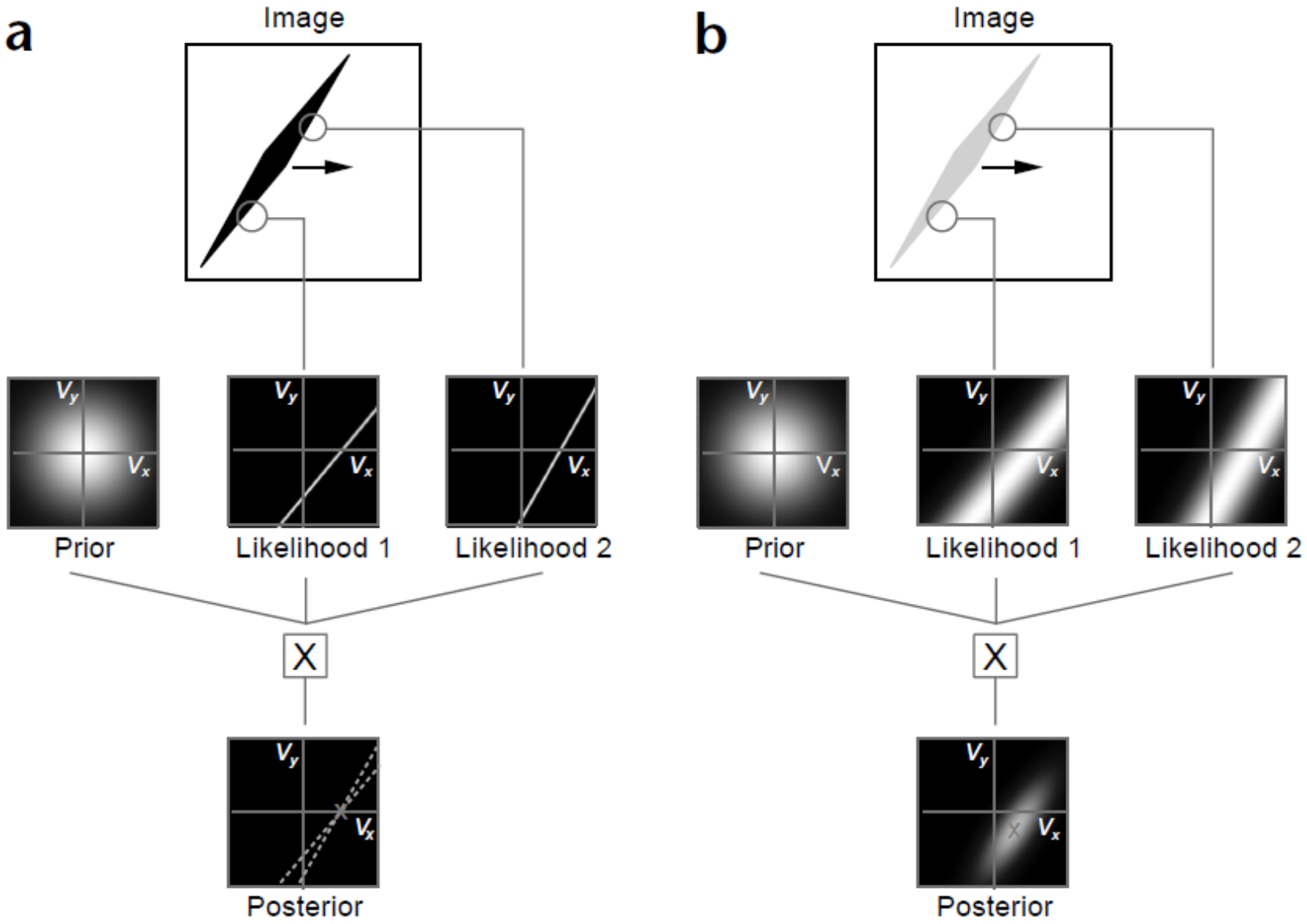
$$P\left(I\left(x_i, y_i, t\right) \mid v_x, v_y\right) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{\partial I\left(x_i, y_i, t\right)}{\partial x}v_x + \frac{\partial I\left(x_i, y_i, t\right)}{\partial y}v_y + \frac{\partial I\left(x_i, y_i, t\right)}{\partial t}\right)^2\right\}$$

Posterior density

$$P\left(v_x, v_y \mid I\left(x_i, y_i, t\right)\right) \propto \exp\left\{-\frac{1}{2\sigma_v^2}\left(v_x^2 + v_y^2\right)^2 - \frac{1}{2\sigma^2}\sum_i\left(\frac{\partial I\left(x_i, y_i, t\right)}{\partial x}v_x + \frac{\partial I\left(x_i, y_i, t\right)}{\partial y}v_y + \frac{\partial I\left(x_i, y_i, t\right)}{\partial t}\right)^2\right\}$$

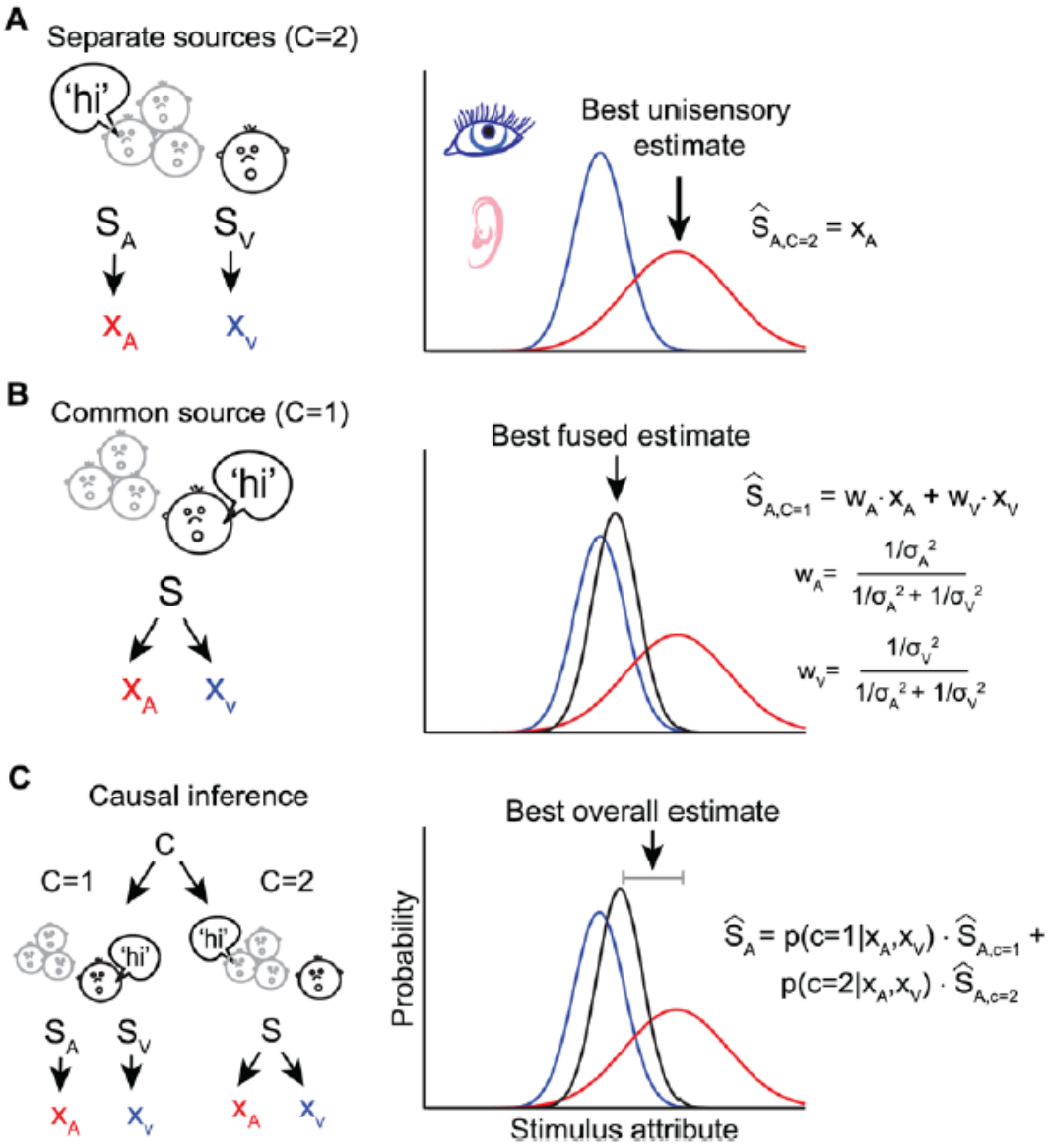
Weiss et al. (2002) Nature Neurosci

Motion illusion as optimal percepts.



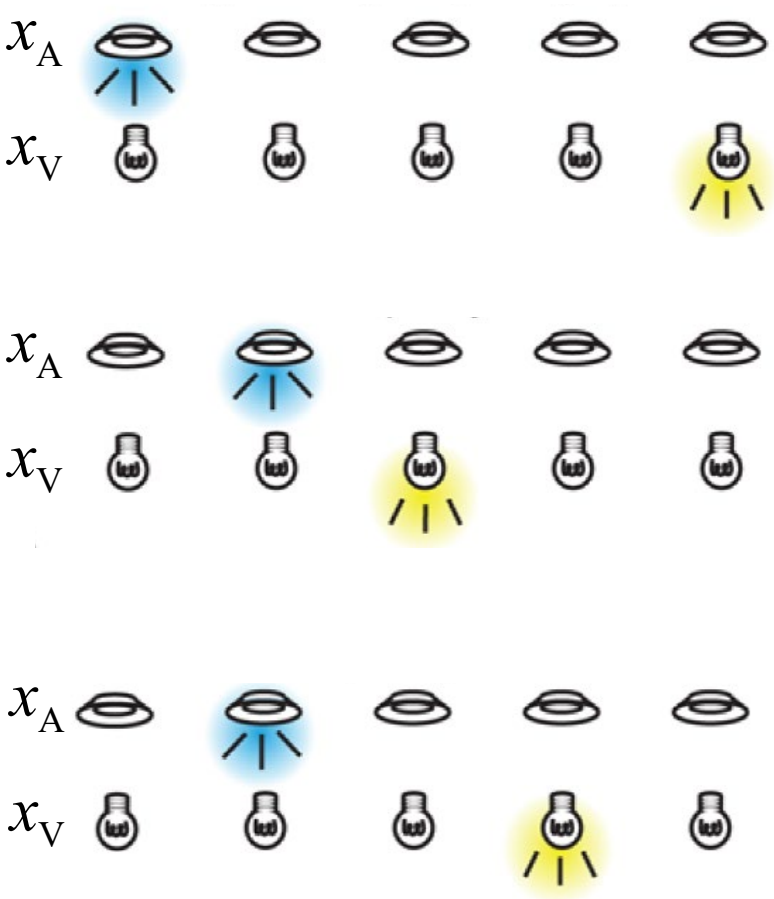
Weiss et al. (2002) Nature Neurosci

Causal inference of sensory information.



Kording et al. (2007) PLoS One; Kayer & Shams (2015) PLoS Biol

Causal inference of sensory information.



Given visual (x_V) and auditory (x_A) observations…

Q1: Do they belong to a single object (C=1) or come from two different objects (C=2)?

Q2: What are the most likely positions for the visual and the auditory objects?

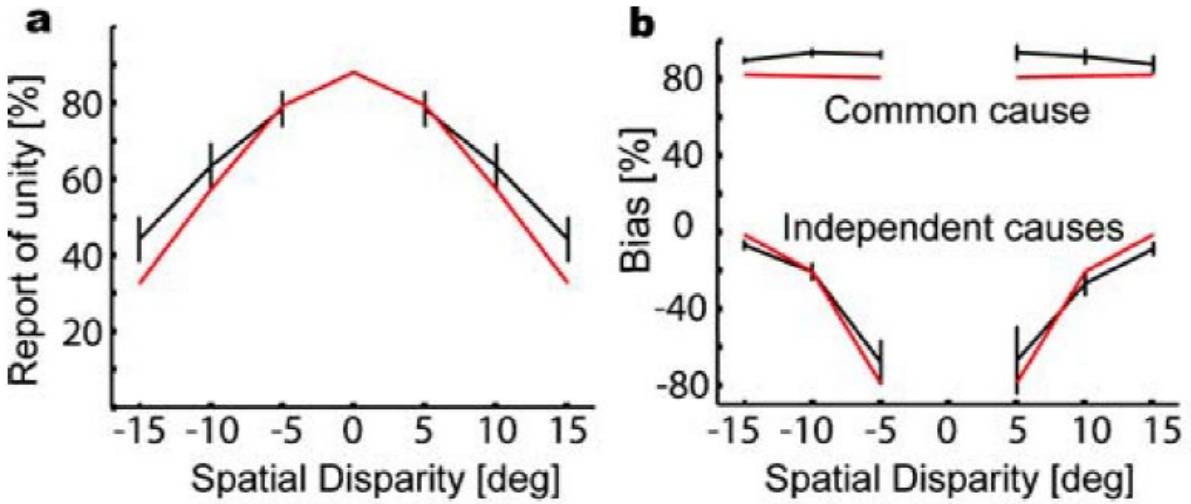
Kording et al. (2007) PLoS One; Kayer & Shams (2015) PLoS Biol

Causal inference of sensory information.

Bayes' theorem:

$$P(C=1 \mid x_A, x_V) = \frac{P(x_A, x_V \mid C=1)P(C=1)}{P(x_A, x_V \mid C=1)P(C=1) + P(x_A, x_V \mid C=2)P(C=2)}$$

$$P(C=2 \mid x_A, x_V) = \frac{P(x_A, x_V \mid C=2)P(C=2)}{P(x_A, x_V \mid C=1)P(C=1) + P(x_A, x_V \mid C=2)P(C=2)}$$



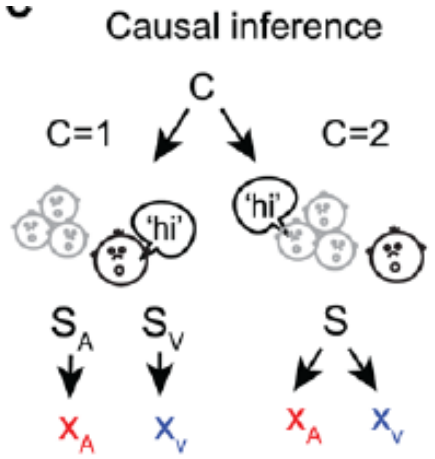
Kording et al. (2007) PLoS One; Kayer & Shams (2015) PLoS Biol

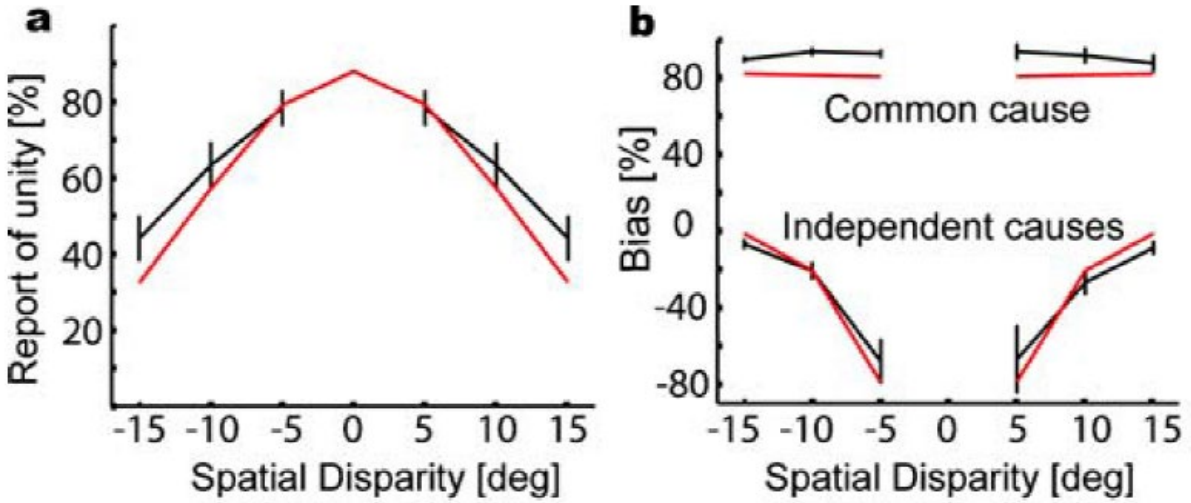
Causal inference of sensory information.

Model averaging.:

$$\hat{s}_{\mathrm{A}} = \hat{s}_{\mathrm{A},C=1} \times P\left(C=1 \mid x_{\mathrm{A}}, x_{\mathrm{V}}\right) + \hat{s}_{\mathrm{A},C=2} \times P\left(C=2 \mid x_{\mathrm{A}}, x_{\mathrm{V}}\right)$$

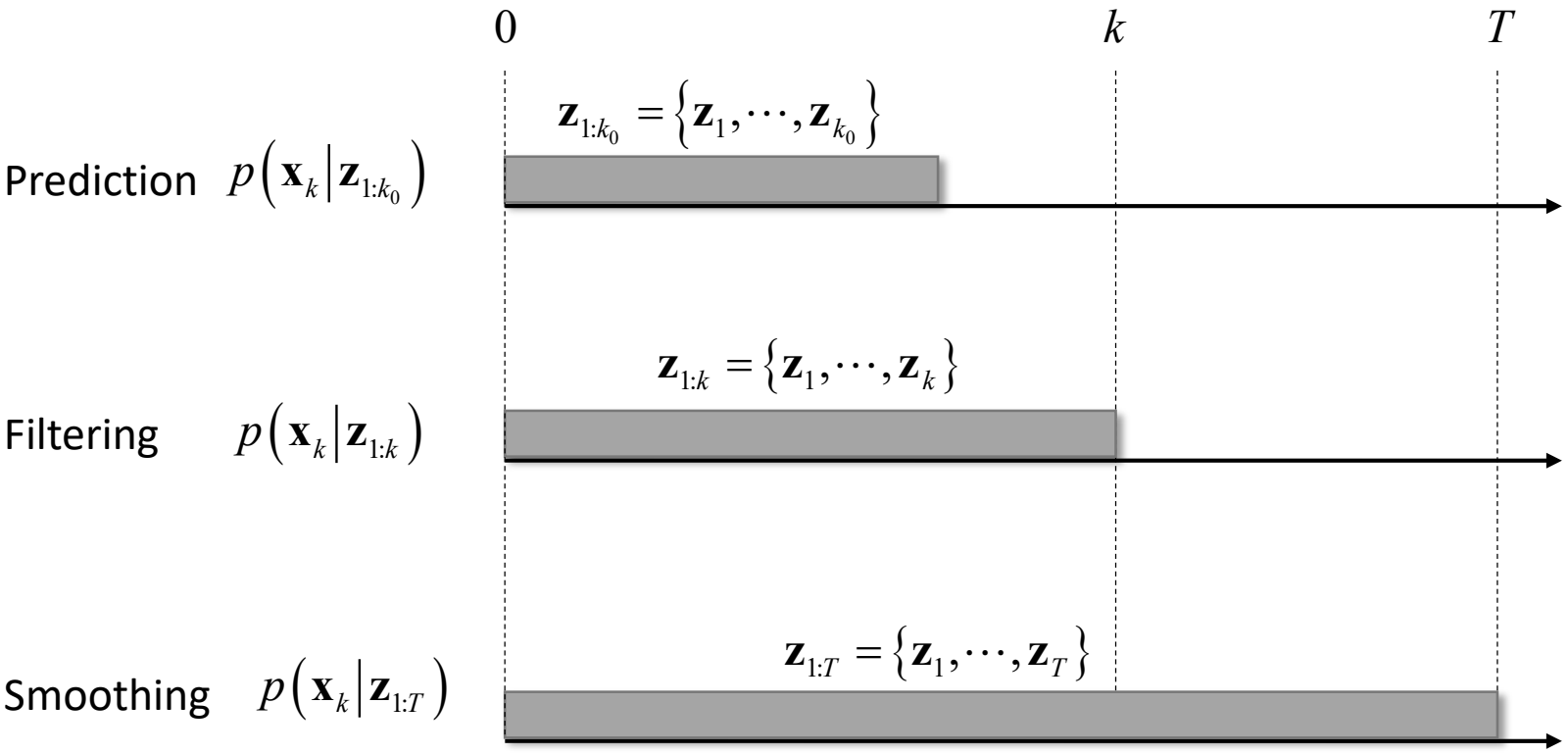
$$\hat{s}_{\mathrm{V}} = \hat{s}_{\mathrm{V},C=1} \times P\left(C=1 \mid x_{\mathrm{A}}, x_{\mathrm{V}}\right) + \hat{s}_{\mathrm{V},C=2} \times P\left(C=2 \mid x_{\mathrm{A}}, x_{\mathrm{V}}\right)$$





Kording et al. (2007) PLoS One; Kayer & Shams (2015) PLoS Biol

Prediction, filtering and smoothing of a dynamic system.



Estimation of dynamic systems: Kalman filter.

Stochastic state-space model

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k$$
$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

Problem: Given a sequence of measurements up to step k,

$$\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\},$$

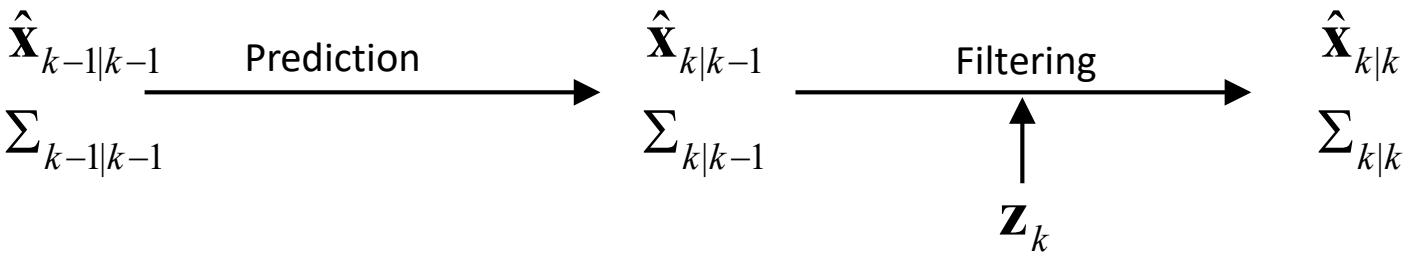
then estimate the conditional mean

$$\hat{\mathbf{x}}_{k|k} = E\left[\mathbf{x}_k \middle| \mathbf{z}_{1:k}\right],$$

and the conditional covariance of the state vector at step k,

$$\Sigma_{k|k} = \mathrm{Cov}\left[\mathbf{x}_k \middle| \mathbf{z}_{1:k}\right] = \mathrm{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k:k}\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k:k}\right)^{\mathrm{T}} \middle| \mathbf{z}_{1:k}\right],$$

Kalman filter optimally integrates prediction and measurement.



Prediction

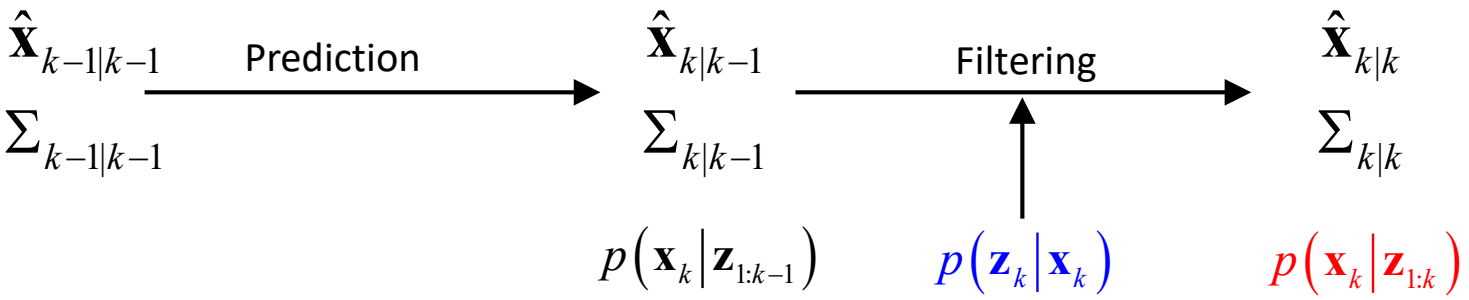
$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1}$$
$$\Sigma_{k|k-1} = \mathbf{A}\Sigma_{k-1|k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$$

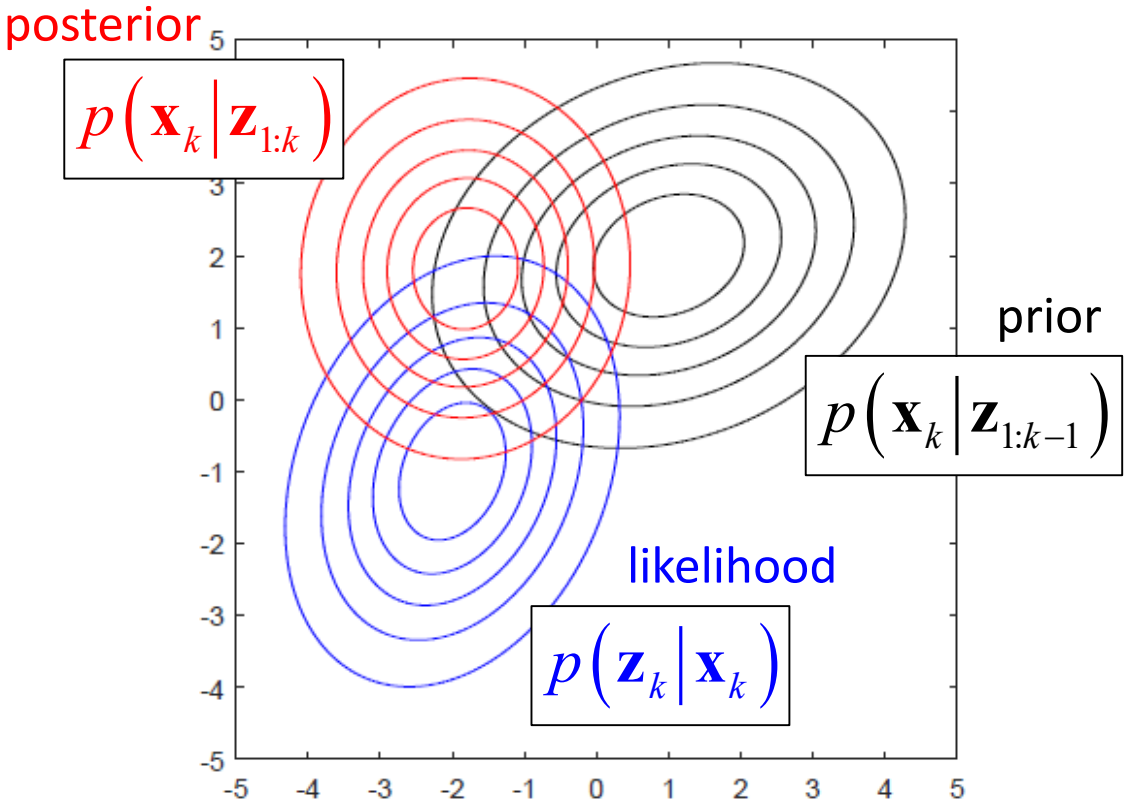
Filtering

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}\right)$$
$$\Sigma_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{C}\right)\Sigma_{k|k-1}$$

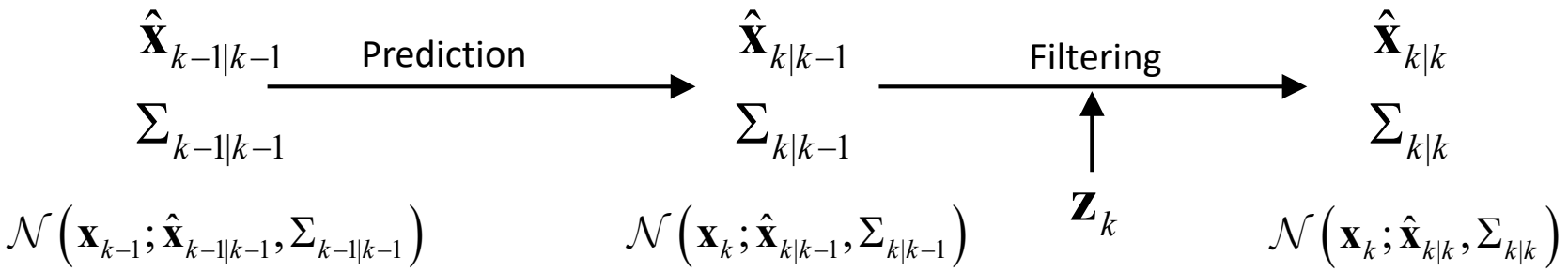
Kalman gain $$\mathbf{K}_k = \Sigma_{k|k-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\Sigma_{k|k-1}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\right)^{-1}$$

Kalman filter optimally integrates prediction and measurement.





Kalman filter: Derivation.



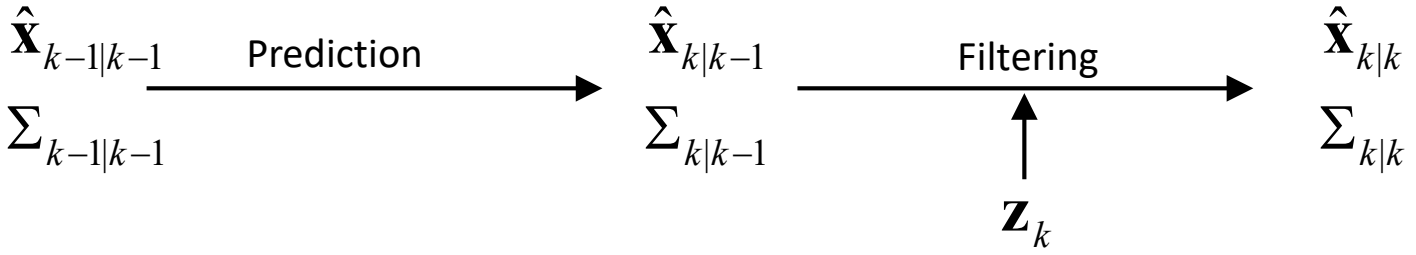
Prediction

$$
\begin{aligned}
p\left(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1}\right) &= \int d\mathbf{x}_{k-1}\, p\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1} \middle| \mathbf{z}_{1:k-1}\right) \\
&= \int d\mathbf{x}_{k-1}\, \mathcal{N}\left(\mathbf{x}_k;\mathbf{A}\mathbf{x}_{k-1},\mathbf{Q}\right) \mathcal{N}\left(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1},\Sigma_{k-1|k-1}\right) \\
&= \mathcal{N}\left(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},\Sigma_{k|k-1}\right)
\end{aligned}
$$

Filtering

$$
p\left(\mathbf{x}_k \middle| \mathbf{z}_{1:k}\right) = \frac{p\left(\mathbf{z}_k \middle| \mathbf{x}_k\right) p\left(\mathbf{x}_k \middle| \mathbf{z}_{1:k-1}\right)}{p\left(\mathbf{z}_k \middle| \mathbf{z}_{1:k-1}\right)} = \mathcal{N}\left(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},\Sigma_{k|k}\right)
$$

Kalman filter: Derivation.



Lemma: When a joint density function of two variables x and y are given as

$$\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_{\mathbf{x}}\\ \boldsymbol{\mu}_{\mathbf{y}}\end{pmatrix}, \begin{pmatrix}\Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}}\\ \Sigma_{\mathbf{yx}} & \Sigma_{\mathbf{yy}}\end{pmatrix}\right),$$

then conditional probability densities are given as follows:

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\left(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}}\right), \Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\Sigma_{\mathbf{yx}}\right)$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{y}} + \Sigma_{\mathbf{yx}}\Sigma_{\mathbf{xx}}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}\right), \Sigma_{\mathbf{yy}} - \Sigma_{\mathbf{yx}}\Sigma_{\mathbf{xx}}^{-1}\Sigma_{\mathbf{xy}}\right)$$

Matlab code of Kalman filter and smoother (1/5)

```
function [x, z, w, v] = sim_statespace(m, N)

x = zeros(m.dx, N);
z = zeros(m.dz, N);

w = m.sQ*randn(m.dx, N);
v = m.sR*randn(m.dz, N);

x(:,1) = m.x0 + sqrtm(m.P0)*randn(m.dx, 1);
z(:,1) = m.C*x(:,1) + v(1);

for n=1:N-1
    x(:,n+1) = m.A*x(:,n) + w(:,n);
    z(:,n+1) = m.C*x(:,n+1) + v(:,n+1);
end
```

Matlab code of Kalman filter and smoother (3/5)

```
function [xp, xf, Pp, Pf, K] = kalmanfiter(m, z)

N = size(z,2);
xp = zeros(m.dx, N); xf = zeros(m.dx, N);
Pp = zeros(m.dx, m.dx, N); Pf = zeros(m.dx, m.dx, N);
K = zeros(m.dx, m.dz, N);

% initialize:
xp(:,1) = m.x0; Pp(:,:,1) = m.P0;

K(:,:,1) = Pp(:,:,1)*m.C'/(m.C*Pp(:,:,1)*m.C'+m.R);
xf(:,1) = xp(:,1) + K(:,:,1)*(z(:,1)-m.C*xp(:,1));
Pf(:,:,1) = (eye(size(m.x0,1))-K(:,:,1)*m.C)*Pp(:,:,1);

% main loop for kalman filter:
for n=1:N-1
    % prediction step:
    xp(:,n+1) = m.A*xf(:,n);
    Pp(:,:,n+1) = m.A*Pf(:,:,n)*m.A'+m.Q;

    % filtering step:
    K(:,:,n+1) = Pp(:,:,n+1)*m.C'/(m.C*Pp(:,:,n+1)*m.C'+m.R);
    xf(:,n+1) = xp(:,n+1) + K(:,:,n+1)*(z(:,n+1)-m.C*xp(:,n+1));
    Pf(:,:,n+1) = (eye(size(m.x0,1))-K(:,:,n+1)*m.C)*Pp(:,:,n+1);
end
```

Matlab code of Kalman filter and smoother (4/5)

```
function [xs, Ps, G] = kalmansmoother(m)

N = size(m.xp,2);
xs = zeros(m.dx, N);
Ps = zeros(m.dx, m.dx, N);
G = zeros(m.dx, m.dx, N);

% terminal conditions:
xs(:,N) = m.xf(:,N);
Ps(:,:,N) = m.Pf(:,:,N);

for n=N-1:(-1):1
    G(:,:,n) = m.Pf(:,:,n)*m.A'/m.Pp(:,:,n+1);
    xs(:,n) = m.xf(:,n) + G(:,:,n)*(xs(:,n+1)-m.xp(:,n+1));
    Ps(:,:,n) = m.Pf(:,:,n) + G(:,:,n)*(Ps(:,:,n+1)-
m.Pp(:,:,n+1))*G(:,:,n)';
end
```

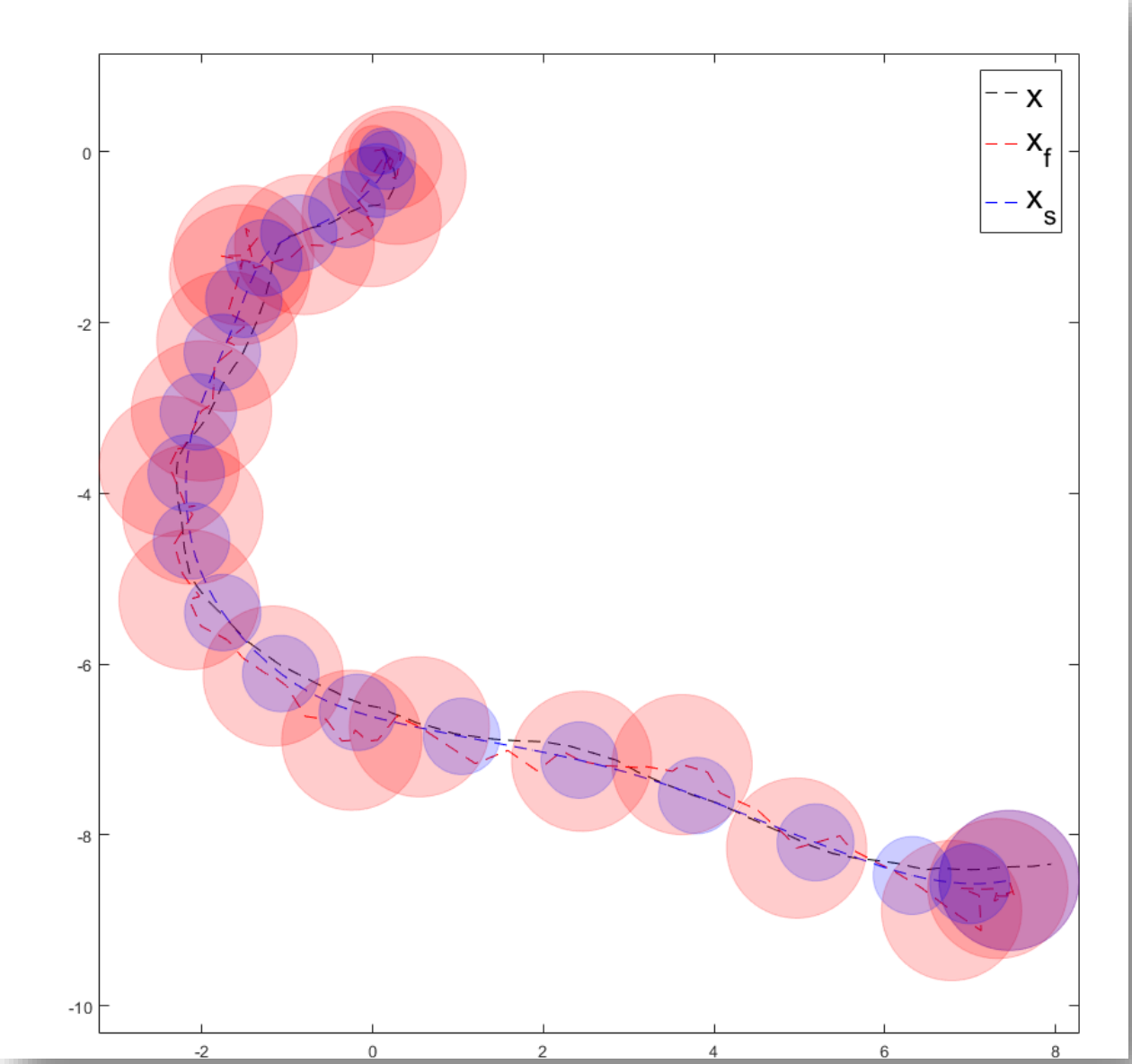
Kalman filter

```
dt = 0.1;
q1 = 1.;
q2 = 1.;
sigma1 = 0.5;
sigma2 = 0.5;

model.A = [1 0 dt 0; 0 1 0 dt; 0 0 1 0; 0 0 0 1];
model.C = [1 0 0 0; 0 1 0 0];
model.Q = [q1*dt^3/3 0 q1*dt^2/2 0; 0 q2*dt^3/3 0
q2*dt^2/2;
    q1*dt^2/2 0 q1*dt 0; 0 q2*dt^2/2 0 q2*dt];
model.sQ = sqrtm(model.Q);
model.R = [sigma1^2 0; 0 sigma2^2];
model.sR = sqrtm(model.R);
model.x0 = [0 0 0 0]';
model.P0 = diag([0.1^2 0.1^2 0.01^2 0.01^2]);
model.dx = size(model.A,1);
model.dz = size(model.C,1);
```

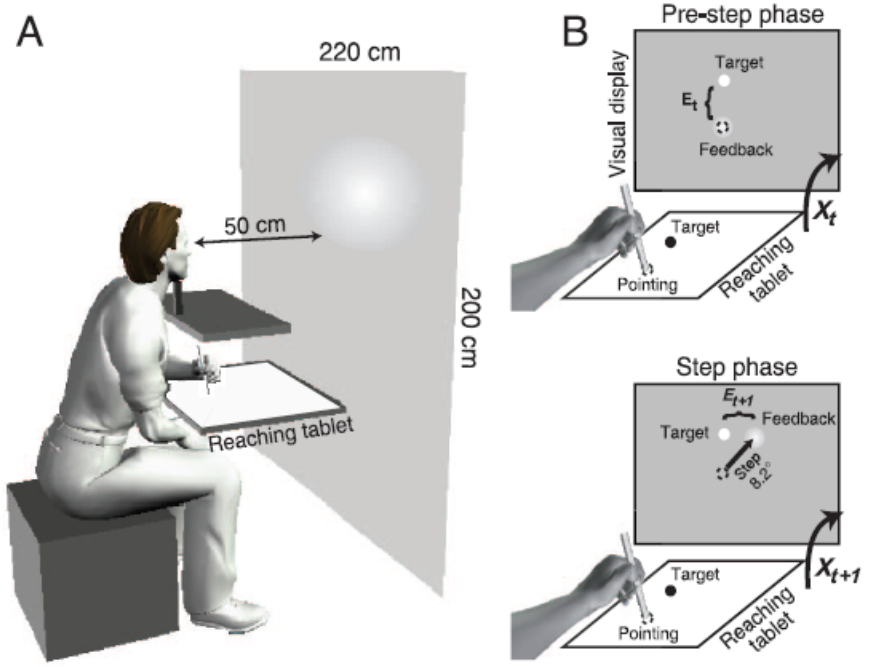
Parameters taken from: Example 4.3, “Bayesian Filtering and Smoothing” by Sarkka

Matlab code of Kalman filter and smoother (4/5)



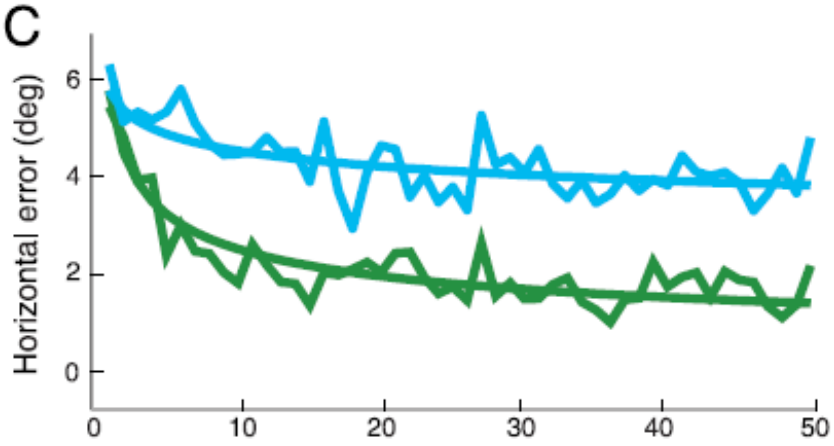
Kalman filter explains smooth eye pursuit movements.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}\right)$$



σ_{blur} = 24° x 24°

noisy sensory feedback: Kalman filter predicts *slower* adaptation.

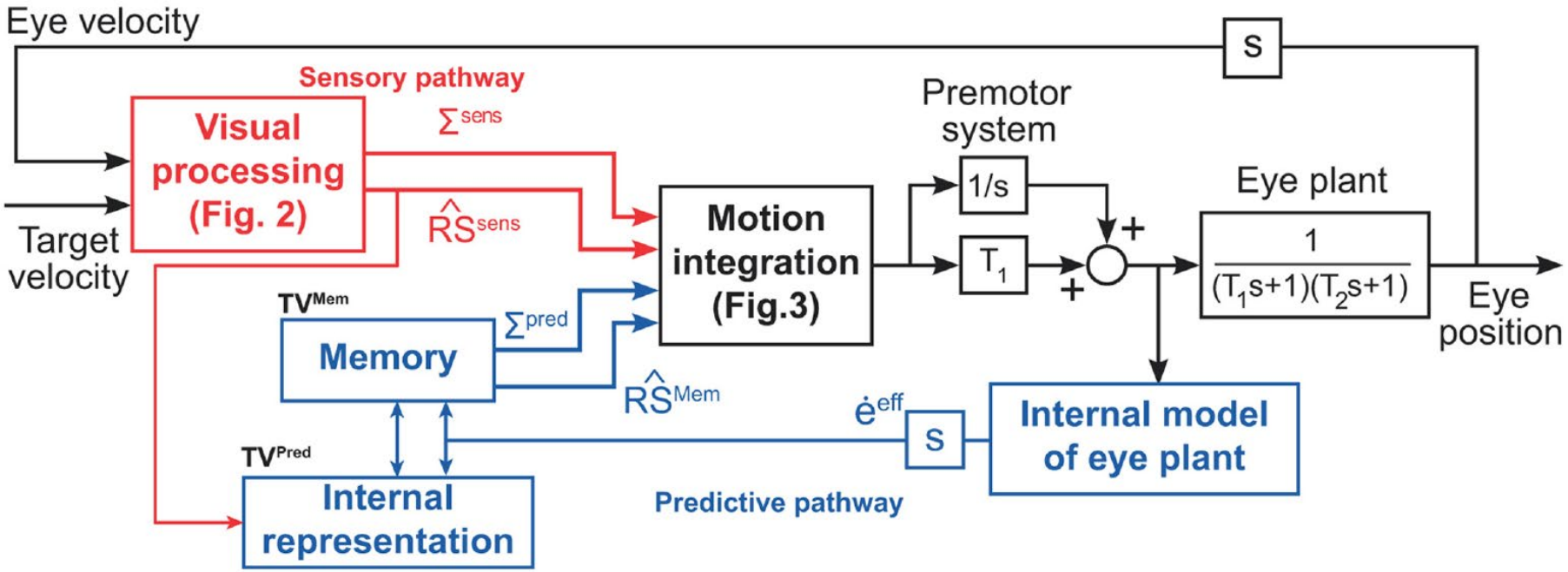


σ_{blur} = 4° x 4°

accurate sensory feedback: Kalman filter predicts *faster* adaptation.

Burge (2007) J Vision

Kalman filter explains smooth eye pursuit movements.



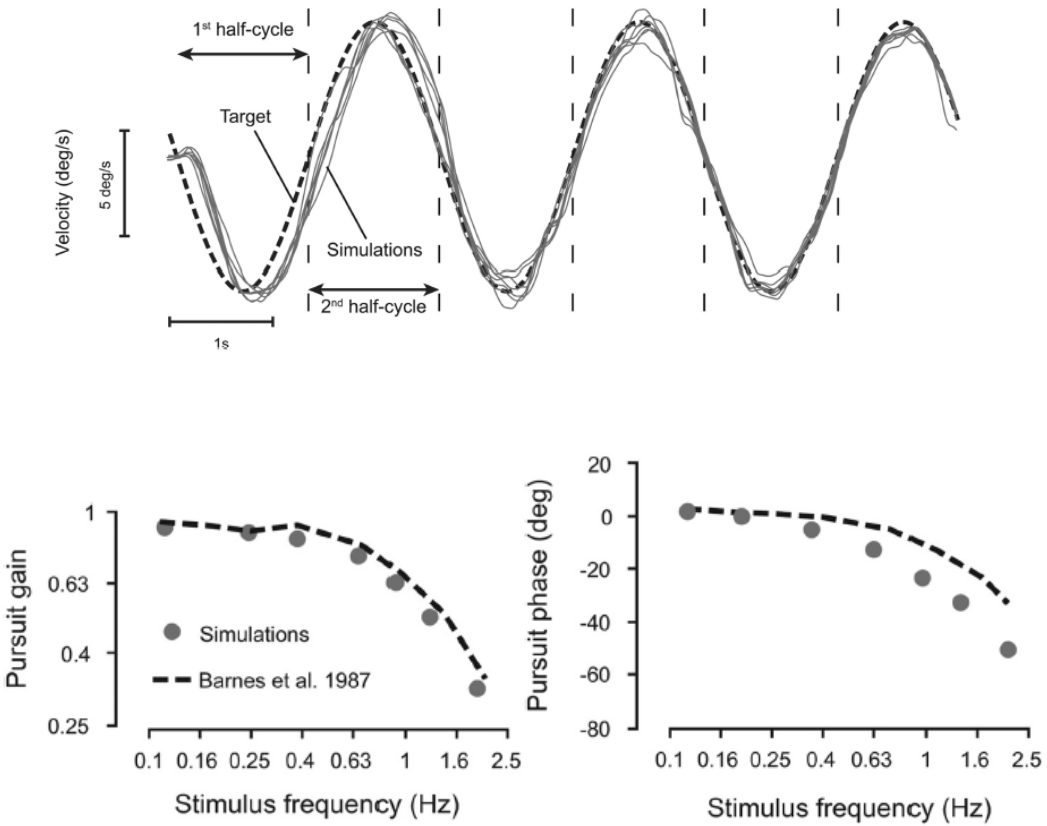
Kalman filter #1 for visual processing of retinal slip.

Kalman filter #2 for remembered target dynamics.

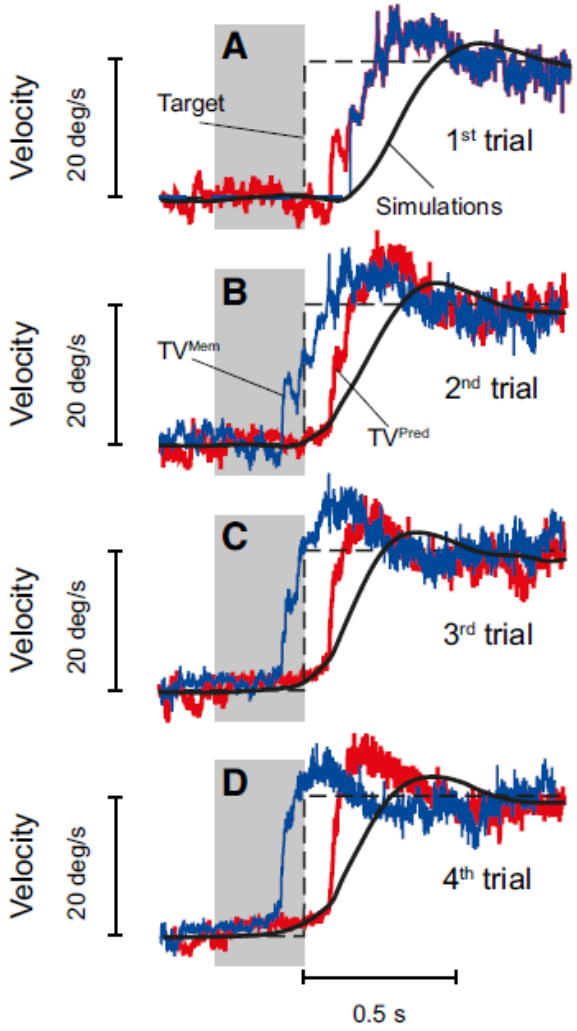
Orban de Xivry et al. (2013) J Neurosci

Kalman filter explains smooth eye pursuit movements.

Sinusoidal target motion

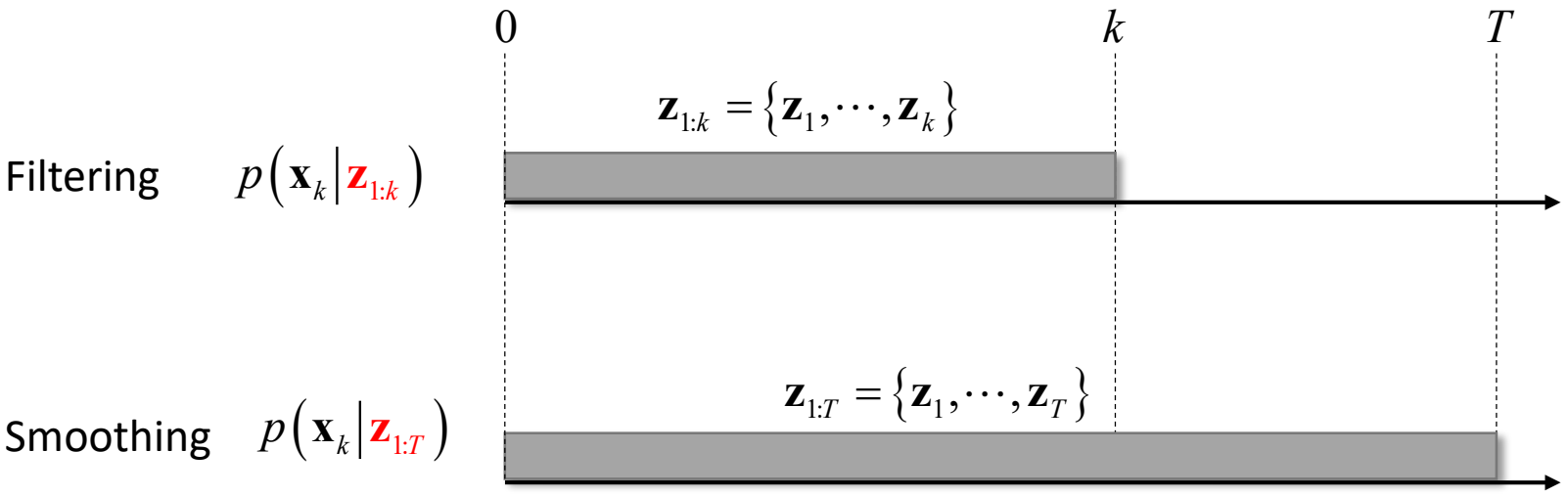


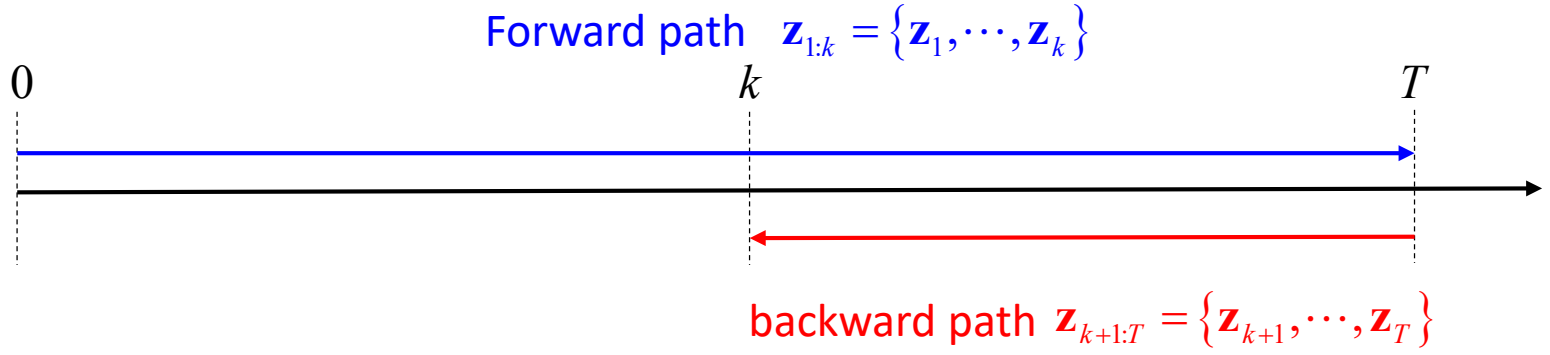
Anticipatory smooth pursuit



Orban de Xivry et al. (2013) J Neurosci

Smoothing consists of forward and backward paths.





Kalman smoother algorithm: RTS smoother.

Forward path: Using the standard algorithm of Kalman filter, compute the predicted mean and covariance,

$$\left(\hat{\mathbf{x}}_{k+1|k}, \Sigma_{k+1|k}\right) \quad k = 0, \cdots, T-1,$$

and the filtered mean and covariance,

$$\left(\hat{\mathbf{x}}_{k|k}, \Sigma_{k|k}\right) \quad k = 0, \cdots, T.$$

Backward path: Then, the smoothed mean and covariance,

$$\left(\hat{\mathbf{x}}_{k|T}, \Sigma_{k|T}\right) \quad k = 0, \cdots, T.$$

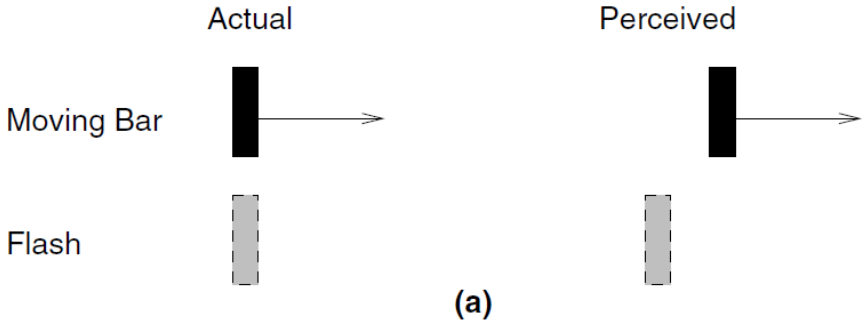
are solved backward in time as follows:

$$\hat{\mathbf{x}}_{k|T} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\left(\hat{\mathbf{x}}_{k+1|T} - \hat{\mathbf{x}}_{k+1|k}\right)$$

$$\Sigma_{k|T} = \Sigma_{k|k} + \mathbf{G}_k\left(\Sigma_{k+1|T} - \Sigma_{k+1|k}\right)\mathbf{G}_k^T$$

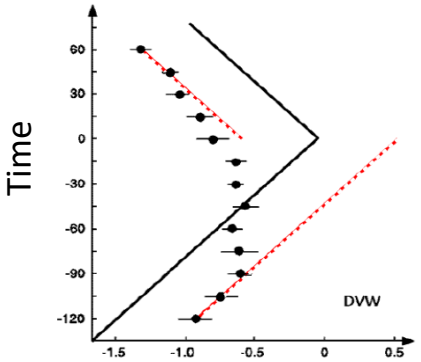
where $\mathbf{G}_k = \Sigma_{k|k}\mathbf{A}^T\left(\Sigma_{k+1|k}\right)^{-1}$

Flash-lag effect explained by Kalman smoother.

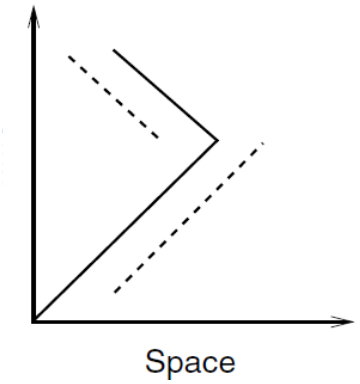


http://visionlab.harvard.edu/Members/Alumni/David/flash-lag.htm

Psychophysics

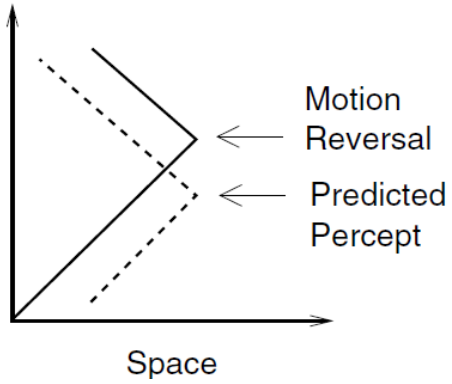


Extrapolation model



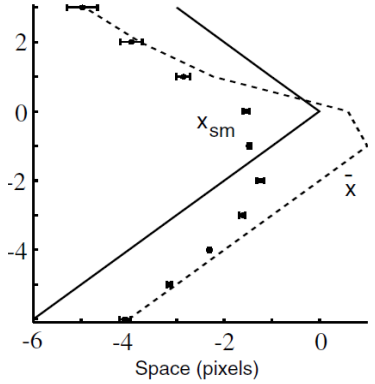
Nijhawan (1994)

Latency difference model



Whitney & Murakami (1998)

Optimal smoothing model



Rao et al. (2001)

Rao et al. (2001) Neural Comput

Attractor neural networks for ML estimation.

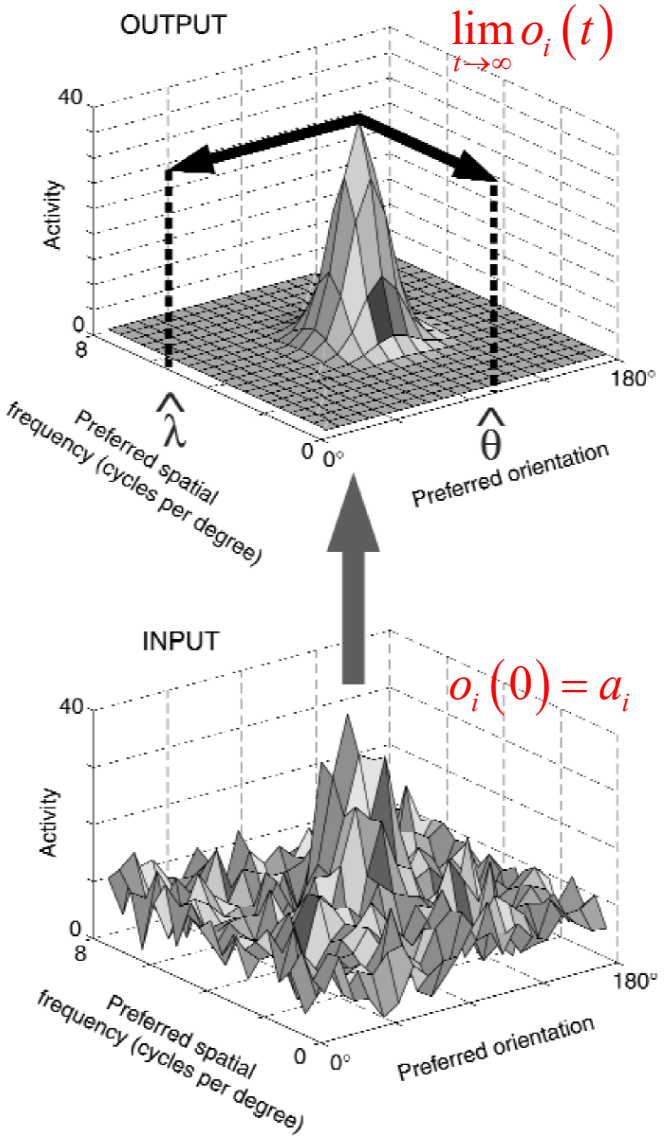
Dynamics of recurrent neural network

Local averaging

$$u_i(t+1) = \sum_j w_{ij} o_j(t)$$

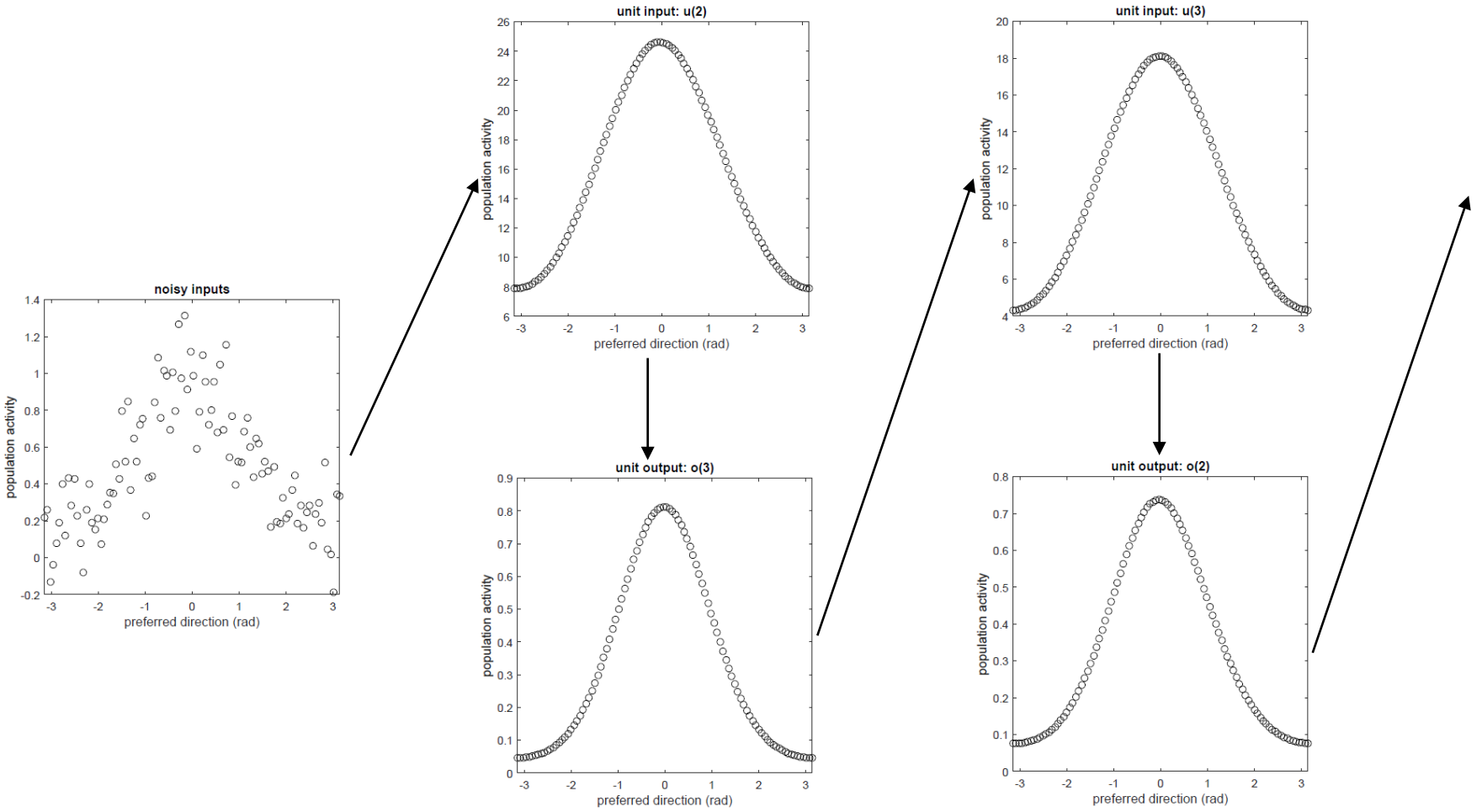
Divisive normalization ("winner-takes-all")

$$o_i(t+1) = \frac{u_i^2(t+1)}{1+\mu\sum_j u_j^2(t+1)}$$



Deneve et al. (1999) Nature Neurosci

Attractor neural networks for ML estimation.



Deneve et al. (1999) Nature Neurosci

Neural network algorithm for maximum likelihood.

Probabilistic neural responses to stimulus s:

$$n_i \sim \text{Poisson}\left(f_i(s)\right)$$

$$P(r_i \mid s) = \frac{e^{-f_i(s)} f_i(s)^{n_i}}{n_i!}$$

Likelihood function:

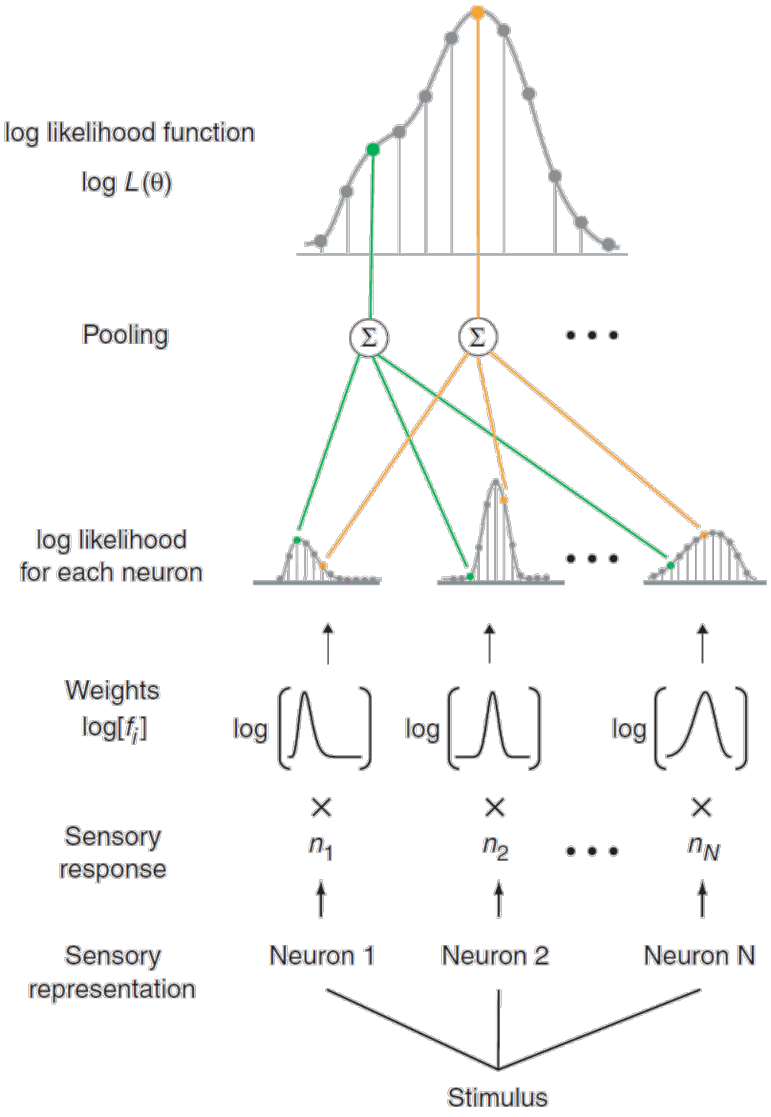
$$P(\mathbf{n} \mid s) = \prod_{i=1}^{N} P(n_i \mid s) = \prod_{i=1}^{N} \frac{e^{-f_i(s)} f_i(s)^{n_i}}{n_i!}$$

Log likelihood function:

$$\log P(\mathbf{n} \mid s)$$

$$= \sum_{i=1}^{N} \log P(n_i \mid s)$$

$$= -\sum_{i=1}^{N} f_i(s) + \sum_{i=1}^{N} n_i \log f_i(s) + \sum_{i=1}^{N} n_i!$$



Jazayeri & Movshon (2006) Nature Neurosci

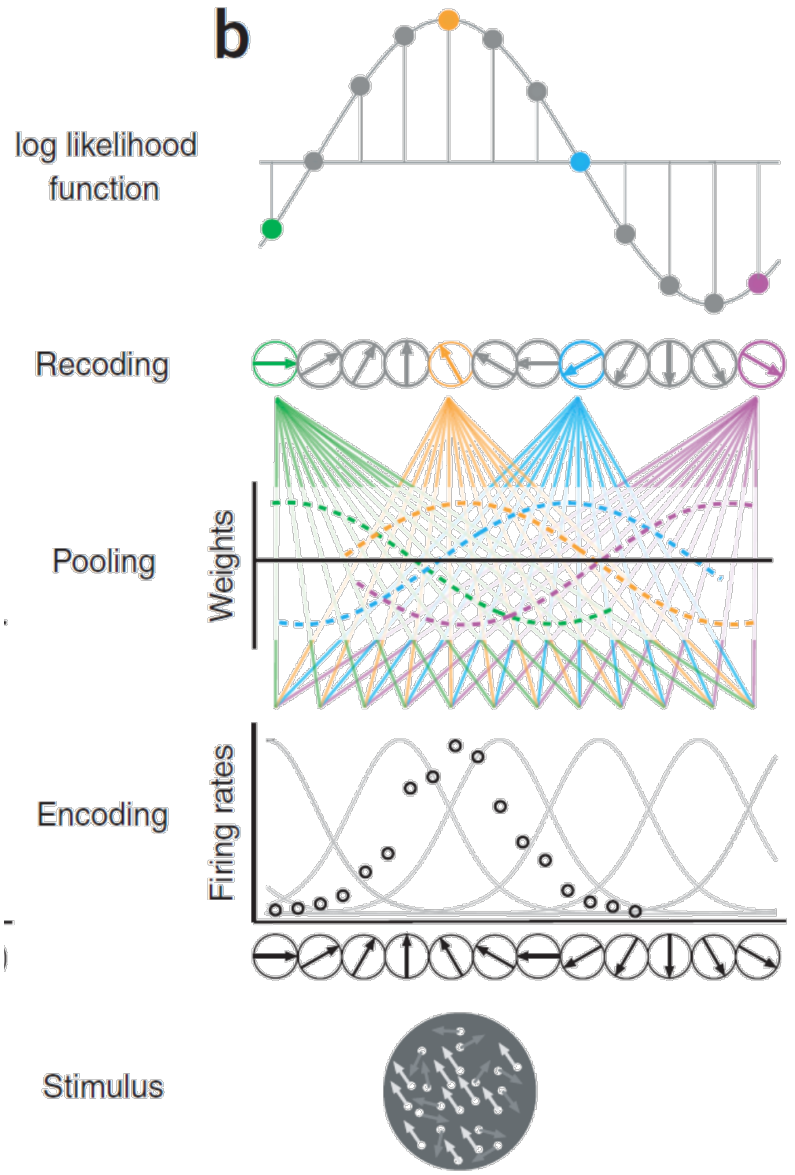
Neural network algorithm for maximum likelihood.

$$\log P(\mathbf{r} \mid s) = \sum_{i=1}^{N} \log P(r_i \mid s)$$

$$= -\sum_{i=1}^{N} f_i(s) + \sum_{i=1}^{N} r_i \log f_i(s) + \sum_{i=1}^{N} r_i!$$

Maximize $\sum_{i=1}^{N} r_i \log f_i(s)$

under constraint $\sum_{i=1}^{N} f_i(s) = \text{constant}$



Jazayeri & Movshon (2006) Nature Neurosci

Neural implementation of multimodal integration.

If tuning function is Gaussian with mean s_i and variance ${\sigma_0}^2$,

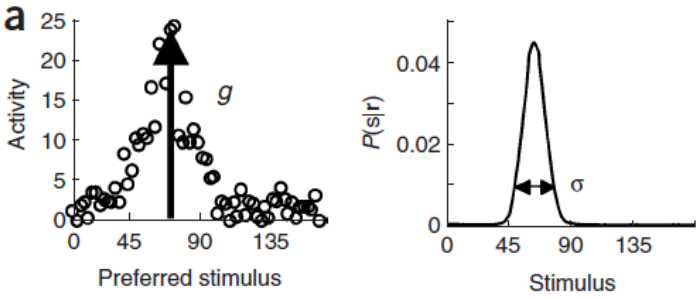
$$f_i(s) = \mathcal{N}\left(s_i, \sigma_0^2\right) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(s-s_i)^2}{2\sigma_0^2}\right),$$

then, the likelihood function is also Gaussian:

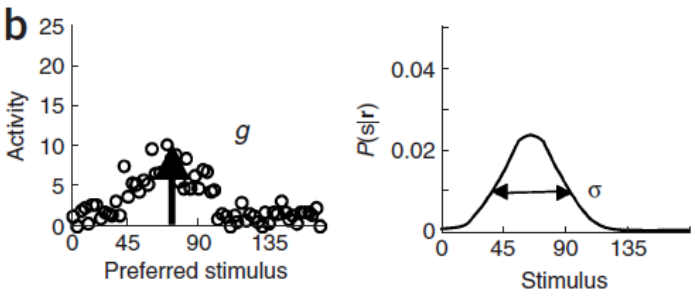
$$\log P(\mathbf{n} \mid s) = \sum_{i=1}^{N} n_i \log f_i(s) + \text{const.} = -\sum_{i=1}^{N} \frac{n_i}{2\sigma_0^2}(s-s_i)^2 + \text{const.}$$

$$\therefore \quad P(\mathbf{n} \mid s) = \mathcal{N}\left(s; \mu, \sigma^2\right) \quad \text{where} \quad \mu = \frac{\sum_{i=1}^{N} n_i s_i}{\sum_{i=1}^{N} n_i}, \quad \sigma^2 = \frac{\sigma_0^2}{\sum_{i=1}^{N} n_i}$$

Population activity Likelihood



Population activity Likelihood



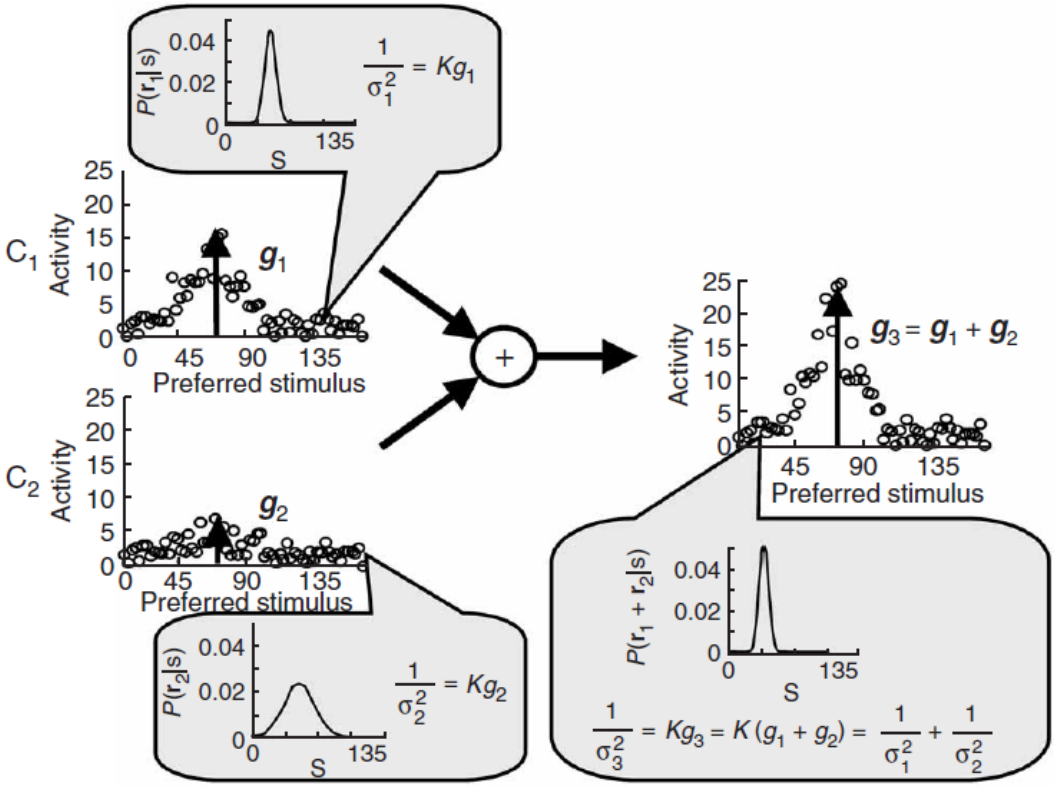
Ma et al. (2006) Nature Neurosci

Neural implementation of multimodal integration.

$$P\left(\mathbf{n}^{(3)} \mid s\right) = P\left(\mathbf{n}^{(1)} + \mathbf{n}^{(2)} \mid s\right)$$
$$= P\left(\mathbf{n}^{(1)} \mid s\right) P\left(\mathbf{n}^{(2)} \mid s\right)$$
$$= \mathcal{N}\left(\mu_1, \sigma_1^2\right) \mathcal{N}\left(\mu_2, \sigma_2^2\right)$$
$$= \mathcal{N}\left(\mu_3, \sigma_3^2\right)$$

$$\mu_3 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$$

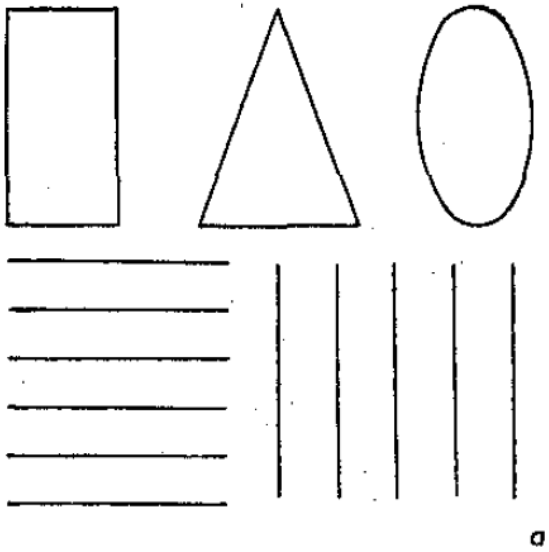
$$\frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

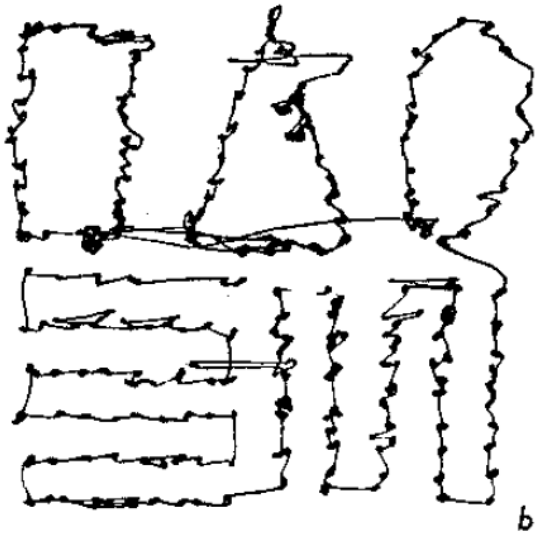


Ma et al. (2006) Nature Neurosci

Eye movements are goal oriented.

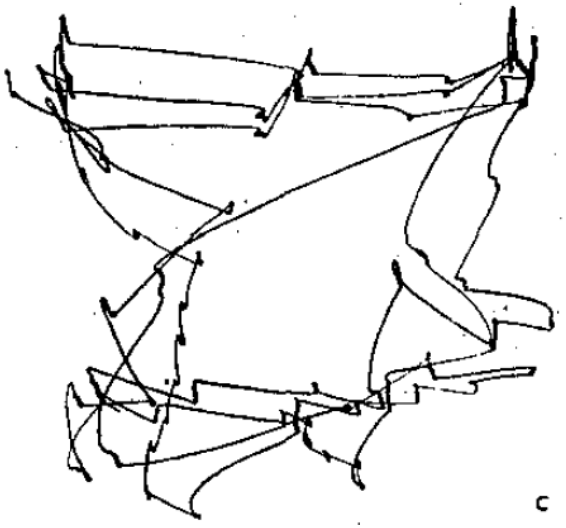
Original figures

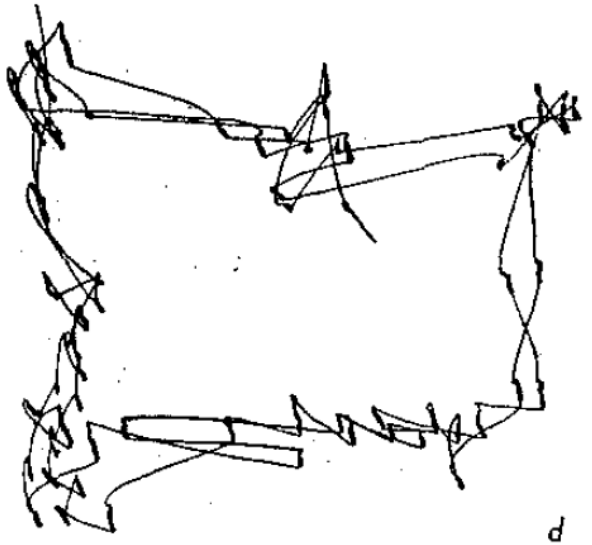




"Trace the lines smoothly."

Free viewing
(no instruction)



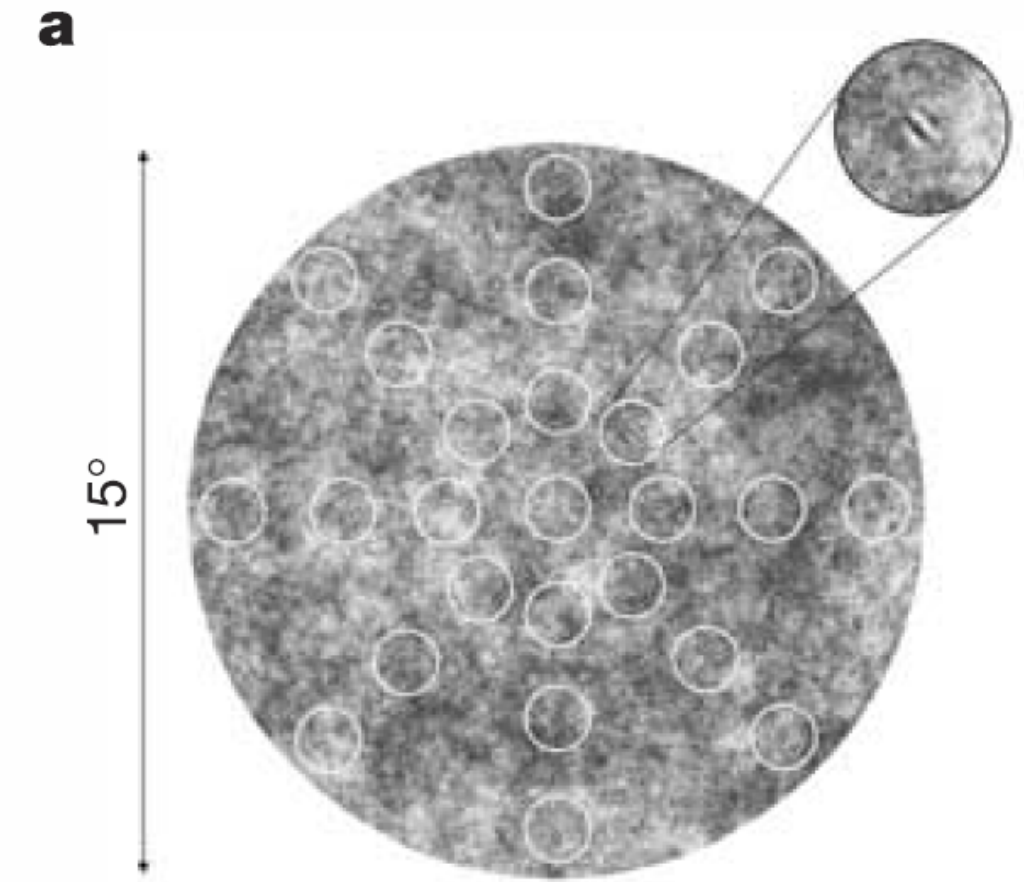


"Count the number of straight lines"

Yarbus (1967) "Eye Movements and Vision."

Eye movements as optimal decision making

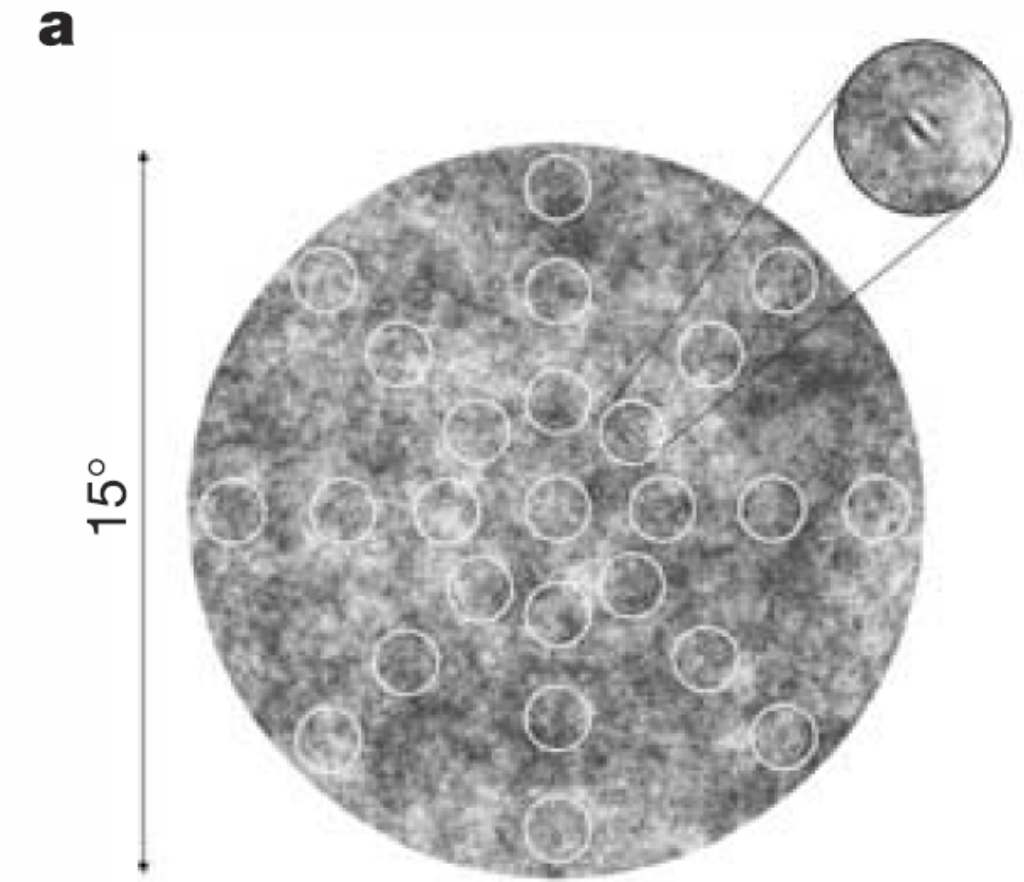
"Look for the Gabor patch in the pink-noise background."



Najemnik & Geisler (2005) Nature

Eye movements as optimal decision making

"Look for the Gabor patch in the pink-noise background."



Najemnik & Geisler (2005) Nature

Summary

- Inference from noisy measurements is formulated as a statistical inference problem.
- Statistical inference is categorized into classical point estimation and Bayesian probability estimation.
- Human performance in psychophysical experiments matches the predictions of optimal estimation.
- The computation of optimal inference can be implemented by a few neural mechanisms.

References

- Ernst, M. O., & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. Nature, 415(6870), 429-433.
- Weiss, Y., Simoncelli, E. P., & Adelson, E. H. (2002). Motion illusions as optimal percepts. Nature neuroscience, 5(6), 598-604.
- Kording, K. P., & Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. Nature, 427(6971), 244-247.
- Kording, K. P., Beierholm, U., Ma, W. J., Quartz, S., Tenenbaum, J. B., & Shams, L. (2007). Causal inference in multisensory perception.
- Kayser, C., & Shams, L. (2015). Multisensory causal inference in the brain. PLoS biology, 13(2).
- Burge, J., Ernst, M. O., & Banks, M. S. (2008). The statistical determinants of adaptation rate in human reaching. Journal of Vision, 8(4), 20.
- de Xivry, J. J. O., Coppe, S., Blohm, G., & Lefevre, P. (2013). Kalman filtering naturally accounts for visually guided and predictive smooth pursuit dynamics. The Journal of Neuroscience, 33(44), 17301-17313.
- Rao, R. P., Eagleman, D. M., & Sejnowski, T. J. (2001). Optimal smoothing in visual motion perception. Neural Computation, 13(6), 1243-1253.
- Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. Nature neuroscience, 2(8), 740-745.
- Jazayeri, M., & Movshon, J. A. (2006). Optimal representation of sensory information by neural populations. Nature neuroscience, 9(5), 690-696.
- Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. Nature neuroscience, 9(11), 1432-1438.

Exercise

- Write a Matlab code for Kalman filter and Kalman smoothing, and examine the difference.

- Psychophysics: Smooth eye pursuit experiment using GazeParser (http://gazeparser.sourceforge.net/).

Appendix A: Derivation of Kalman filter equations (1)

- Strategy 1.
  - Compute the mean and average.
  - Intuitive but a bit heuristic.

- Strategy 2.
  - Use Bayes' formula.
  - Complete the square.

- Strategy 3.
  - Use Bayes' formula.
  - Use the formula for conditional normal probability.

Appendix A: Derivation of Kalman filter equations (1)

- Prediction step:

$$\hat{\mathbf{x}}_{k+1|k} = \mathrm{E}\left[\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k}\right] = \mathrm{E}\left[\mathbf{A}\mathbf{x}_k + \mathbf{w}_k \mid \mathbf{z}_{1:k}\right] = \mathbf{A}\mathbf{x}_{k|k}$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{k+1|k} &= \mathrm{Var}\left[\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k}\right] \\
&= \mathrm{E}\left[\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}\right)^{\mathrm{T}} \mid \mathbf{z}_{1:k}\right] \\
&= \mathrm{E}\left[\left(\mathbf{A}\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right) + \mathbf{w}_k\right)\left(\mathbf{A}\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right) + \mathbf{w}_k\right)^{\mathrm{T}} \mid \mathbf{z}_{1:k}\right] \\
&= \mathbf{A}\mathrm{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\right)^{\mathrm{T}} \mid \mathbf{z}_{1:k}\right]\mathbf{A}^{\mathrm{T}} + \mathrm{E}\left[\mathbf{w}_k \mathbf{w}_k^{T} \mid \mathbf{z}_{1:k}\right] \\
&= \mathbf{A}\boldsymbol{\Sigma}_{k|k}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}
\end{aligned}$$

$$\boxed{\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}\mathbf{x}_{k|k} \\
\boldsymbol{\Sigma}_{k+1|k} &= \mathbf{A}\boldsymbol{\Sigma}_{k|k}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}
\end{aligned}}$$

Appendix A: Derivation of Kalman filter equations (1)

- Filter step:

$$\hat{\mathbf{x}}_{k+1|k+1} = \mathrm{E}\left[\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k}\right] = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k+1|k+1} &= \mathrm{Var}\left[\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k+1}\right] \\ &= \mathrm{E}\left[\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1}\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1}\right)^{\mathrm{T}} \mid \mathbf{z}_{1:k+1}\right] \\ &= \cdots \\ &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)\boldsymbol{\Sigma}_{k+1|k}\left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)^{\mathrm{T}} + \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}_{k+1}^{\mathrm{T}}\end{aligned}$$

$$0 = \frac{1}{2}\frac{\partial}{\partial \mathbf{K}_{k+1}}\mathrm{tr}\boldsymbol{\Sigma}_{k+1|k+1} = -\left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}} + \mathbf{K}_{k+1}\mathbf{R}$$

$$\mathbf{K}_{k+1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\right)^{-1}$$

Appendix A: Derivation of Kalman filter equations (1)

- Note on matrix derivatives.

$$\frac{\partial}{\partial K_{ij}}\mathrm{tr}\left(\mathbf{KAK}^{\mathrm{T}}\right)=\frac{\partial}{\partial K_{ij}}\sum_{k,l,m}K_{kl}A_{lm}K_{mk}^{T}=\frac{\partial}{\partial K_{ij}}\sum_{k,l,m}K_{kl}A_{lm}K_{km}$$

$$=\sum_{k,l,m}\left(\delta_{i,k}\delta_{j,l}A_{lm}K_{km}+K_{kl}A_{lm}\delta_{i,k}\delta_{m,j}\right)=\sum_{m}A_{jm}K_{im}+\sum_{l}K_{il}A_{lj}=\left(\mathbf{KA}^{\mathrm{T}}\right)_{ij}+\left(\mathbf{KA}\right)_{ij}=\left(2\mathbf{KA}\right)_{ij}$$

$$\frac{\partial}{\partial\mathbf{K}}\mathrm{tr}\left(\mathbf{KAK}^{\mathrm{T}}\right)=2\mathbf{KA}$$

Appendix A: Derivation of Kalman filter equations (1)

$$\hat{\mathbf{x}}_{k+1|k+1} = \left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\left(\boldsymbol{\Sigma}_{k+1|k}^{-1}\hat{\mathbf{x}}_{k+1|k} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{z}_{k+1}\right)$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}$$

$$\left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\boldsymbol{\Sigma}_{k+1|k}^{-1} = \mathbf{I} - \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C} = \mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}$$

$$\left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\right)^{-1} = \mathbf{K}_{k+1}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k+1|k+1} &= \left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1} \\ &= \boldsymbol{\Sigma}_{k+1|k} - \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C}\boldsymbol{\Sigma}_{k+1|k} \\ &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)\boldsymbol{\Sigma}_{k+1|k}\end{aligned}$$

Appendix A: Derivation of Kalman filter equations (1)

- "Weighted average" expression:

$$\hat{\mathbf{x}}_{k+1|k+1} = \left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\left(\boldsymbol{\Sigma}_{k+1|k}^{-1}\hat{\mathbf{x}}_{k+1|k} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{z}_{k+1}\right)$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \left(\boldsymbol{\Sigma}_{k+1|k}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C}\right)^{-1}$$

- "Innovation-correction" expression:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)\boldsymbol{\Sigma}_{k+1|k}$$

$$\mathbf{K}_{k+1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\right)^{-1}$$

Matrix identities

- For matrices A, B, C and D:

$$\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{B} \in \mathbb{R}^{n \times m}, \mathbf{C} \in \mathbb{R}^{m \times m}, \mathbf{D} \in \mathbb{R}^{m \times n},$$

$$\left(\mathbf{A}+\mathbf{BCD}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}$$

$$\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{A}+\mathbf{BCD}\right)^{-1} = \left(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}$$

- Rank-1 modification

$$\left(\mathbf{A}+\mathbf{uv}^{\mathrm{T}}\right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{uv}^{\mathrm{T}}\mathbf{A}^{-1}}{1+\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}}$$

Appendix B: Kalman filter in continuous-time dynamics

- Let us start with a discrete-time state-space model:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k$$
$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

If the matrix A is expanded for a small time step dt, the prediction update reads as:

$$\hat{\mathbf{x}}_{k+1|k} = \left(\mathbf{I} + \bar{\mathbf{A}}dt\right)\hat{\mathbf{x}}_{k|k}$$

$$\boldsymbol{\Sigma}_{k+1|k} = \left(\mathbf{I} + \bar{\mathbf{A}}dt\right)\boldsymbol{\Sigma}_{k|k}\left(\mathbf{I} + \bar{\mathbf{A}}dt\right)^{\mathrm{T}} + \mathbf{Q} = \boldsymbol{\Sigma}_{k|k} + \left(\bar{\mathbf{A}}\boldsymbol{\Sigma}_{k|k} + \boldsymbol{\Sigma}_{k|k}\bar{\mathbf{A}}^{\mathrm{T}}\right)dt + \mathbf{Q}$$

From the covariance update, the matrix Q should be of the order of dt:

$$\mathbf{Q} = \bar{\mathbf{Q}}dt$$

then, the covariance update now becomes:

$$\boldsymbol{\Sigma}_{k+1|k} = \boldsymbol{\Sigma}_{k|k} + \left(\bar{\mathbf{A}}\boldsymbol{\Sigma}_{k|k} + \boldsymbol{\Sigma}_{k|k}\bar{\mathbf{A}}^{\mathrm{T}} + \bar{\mathbf{Q}}\right)dt$$

Appendix B: Kalman filter in continuous-time dynamics

- Next, consider the filter step:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k+1|k+1} &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)\boldsymbol{\Sigma}_{k+1|k}\left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}\right)^{\mathrm{T}} + \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}_{k+1}^{\mathrm{T}} \\ &= \underbrace{\boldsymbol{\Sigma}_{k+1|k}}_{\mathcal{O}(1)} - \underbrace{\underbrace{\mathbf{K}_{k+1}}_{\mathcal{O}(dt)}\underbrace{\mathbf{C}}_{\mathcal{O}(1)}\underbrace{\boldsymbol{\Sigma}_{k+1|k}}_{\mathcal{O}(1)}}_{\mathcal{O}(dt)} - \underbrace{\underbrace{\boldsymbol{\Sigma}_{k+1|k}}_{\mathcal{O}(1)}\underbrace{\mathbf{C}^{\mathrm{T}}}_{\mathcal{O}(1)}\underbrace{\mathbf{K}_{k+1}^{\mathrm{T}}}_{\mathcal{O}(dt)}}_{\mathcal{O}(dt)} + \underbrace{\underbrace{\mathbf{K}_{k+1}}_{\mathcal{O}(dt)}\underbrace{\mathbf{C}}_{\mathcal{O}(1)}\underbrace{\boldsymbol{\Sigma}_{k+1|k}}_{\mathcal{O}(1)}\underbrace{\mathbf{C}^{\mathrm{T}}}_{\mathcal{O}(1)}\underbrace{\mathbf{K}_{k+1}^{\mathrm{T}}}_{\mathcal{O}(dt)}}_{\mathcal{O}(dt^2)} + \underbrace{\underbrace{\mathbf{K}_{k+1}}_{\mathcal{O}(dt)}\underbrace{\mathbf{R}}_{\mathcal{O}\left(\frac{1}{dt}\right)}\underbrace{\mathbf{K}_{k+1}^{\mathrm{T}}}_{\mathcal{O}(dt)}}_{\mathcal{O}(dt)}\end{aligned}$$

Hence, it is appropriate to define:

$$\mathbf{R} = \frac{\bar{\mathbf{R}}}{dt}$$

In this limit, the Kalman gain is of the order of dt:

$$\mathbf{K}_{k+1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}} + \frac{\bar{\mathbf{R}}}{dt}\right)^{-1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\bar{\mathbf{R}}^{-1}dt + \mathcal{O}\left(dt^2\right)$$

Appendix B: Kalman filter in continuous-time dynamics

- The prediction equations up to the order of dt are:

$$\hat{\mathbf{x}}_{k+1|k} = \left(\mathbf{I} + \overline{\mathbf{A}}dt\right)\hat{\mathbf{x}}_{k|k}$$

$$\boldsymbol{\Sigma}_{k+1|k} = \boldsymbol{\Sigma}_{k|k} + \left(\overline{\mathbf{A}}\boldsymbol{\Sigma}_{k|k} + \boldsymbol{\Sigma}_{k|k}\overline{\mathbf{A}}^{\mathrm{T}} + \overline{\mathbf{Q}}\right)dt$$

  and the filter equations up to the order of dt are:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \boldsymbol{\Sigma}_{k+1|k} - \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\mathbf{C}\boldsymbol{\Sigma}_{k+1|k}dt$$

- Combining these equations and eliminating the prediction variables, the update equations for the posterior mean and covariance are:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k|k} + \left\{\overline{\mathbf{A}}\mathbf{x}_{k|k} + \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)\right\}dt$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \boldsymbol{\Sigma}_{k|k} + \left(\overline{\mathbf{A}}\boldsymbol{\Sigma}_{k|k} + \boldsymbol{\Sigma}_{k|k}\overline{\mathbf{A}}^{\mathrm{T}} + \overline{\mathbf{Q}} - \boldsymbol{\Sigma}_{k|k}\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\mathbf{C}\boldsymbol{\Sigma}_{k|k}\right)dt$$

Appendix B: Kalman filter in continuous-time dynamics

- By taking the limit of infinitesimal dt

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k|k} + \left\{ \overline{\mathbf{A}}\mathbf{x}_{k|k} + \boldsymbol{\Sigma}_{k+1|k}\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\left(\mathbf{z}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}\right)\right\} dt$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \boldsymbol{\Sigma}_{k|k} + \left(\overline{\mathbf{A}}\boldsymbol{\Sigma}_{k|k} + \boldsymbol{\Sigma}_{k|k}\overline{\mathbf{A}}^{\mathrm{T}} + \overline{\mathbf{Q}} - \boldsymbol{\Sigma}_{k|k}\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\mathbf{C}\boldsymbol{\Sigma}_{k|k}\right) dt$$

and introducing continuous-time variables as

$$\hat{\mathbf{x}}(t) \equiv \hat{\mathbf{x}}_{k|k}, \ \boldsymbol{\Sigma}(t) \equiv \boldsymbol{\Sigma}_{k|k}, \ \mathbf{z}(t) \equiv \mathbf{z}_k,$$

the equations of Kalman filter for a continuous-time system are obtained as:

$$\dot{\hat{\mathbf{x}}}(t) = \overline{\mathbf{A}}\hat{\mathbf{x}}(t) + \boldsymbol{\Sigma}(t)\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\left(\mathbf{z}(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right)$$

$$\dot{\boldsymbol{\Sigma}}(t) = \overline{\mathbf{A}}\boldsymbol{\Sigma}(t) + \boldsymbol{\Sigma}(t)\overline{\mathbf{A}}^{\mathrm{T}} + \overline{\mathbf{Q}} - \boldsymbol{\Sigma}(t)\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}\mathbf{C}\boldsymbol{\Sigma}(t)$$

Matrix formulas

- Matrix inversion lemma

$$\left(\mathbf{A}+\mathbf{BCD}\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}$$

$$\begin{aligned}
&\left(\mathbf{A}+\mathbf{BCD}\right)\left(\mathbf{A}+\mathbf{BCD}\right)^{-1}\\
&=\left(\mathbf{A}+\mathbf{BCD}\right)\left[\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}\right]\\
&=\mathbf{I}+\mathbf{BCDA}^{-1}-\left(\mathbf{I}+\mathbf{BCDA}^{-1}\right)\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}\\
&=\mathbf{I}+\mathbf{BCDA}^{-1}-\mathbf{B}\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}\\
&=\mathbf{I}+\mathbf{BCDA}^{-1}-\mathbf{BC}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}\\
&=\mathbf{I}+\mathbf{BCDA}^{-1}-\mathbf{BCDA}^{-1}\\
&=\mathbf{I}
\end{aligned}$$

$$\left(\mathbf{I}+\mathbf{BCDA}^{-1}\right)\mathbf{B}=\mathbf{B}\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)$$

$$\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)=\mathbf{C}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)$$

Matrix formulas

- Matrix inversion lemma

$$\left(\mathbf{A}+\mathbf{BCD}\right)^{-1}\mathbf{BC}=\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}$$

$$\begin{aligned}&\left(\mathbf{A}+\mathbf{BCD}\right)^{-1}\mathbf{BC}\\&=\mathbf{A}^{-1}\left(\mathbf{I}+\mathbf{BCDA}^{-1}\right)^{-1}\mathbf{BC}\\&=\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\\&=\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\end{aligned}$$

$$\left(\mathbf{I}+\mathbf{BCDA}^{-1}\right)\mathbf{B}=\mathbf{B}\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)$$

$$\left(\mathbf{I}+\mathbf{CDA}^{-1}\mathbf{B}\right)=\mathbf{C}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)$$

Recursive least squares

- Least squares problem:

$$\sum_{k=1}^{n}\left(y_k - \mathbf{w}^{\mathrm{T}}\mathbf{x}_k\right)^2$$

$$\hat{\mathbf{w}}_n = \arg\min_{\mathbf{w}} \sum_{k=1}^{n}\left(y_k - \mathbf{w}^{\mathrm{T}}\mathbf{x}_k\right)^2 = \left(\sum_{k=1}^{n}\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}}\right)^{-1}\left(\sum_{k=1}^{n}y_k\mathbf{x}_k^{\mathrm{T}}\right)$$

- We would like a recursive formula for n+1 when the optimal weight vector up to n is available.

$$\hat{\mathbf{w}}_{n+1} = \arg\min_{\mathbf{w}} \sum_{k=1}^{n+1}\left(y_k - \mathbf{w}^{\mathrm{T}}\mathbf{x}_k\right)^2 = \left(\sum_{k=1}^{n+1}\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}}\right)^{-1}\left(\sum_{k=1}^{n+1}y_k\mathbf{x}_k^{\mathrm{T}}\right)$$

Recursive least squares

- Least squares problem:

$$\hat{\mathbf{w}}_{n+1} = \left(\sum_{k=1}^{n} \mathbf{x}_k \mathbf{x}_k^{\mathrm{T}} + \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \right)^{-1} \left(\sum_{k=1}^{n} y_k \mathbf{x}_k^{\mathrm{T}} + y_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \right)$$

$$\begin{aligned}
\hat{\mathbf{w}}_{n+1} &= \left(\mathbf{P}_n + \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \right)^{-1} \left(\sum_{k=1}^{n} y_k \mathbf{x}_k + y_{n+1} \mathbf{x}_{n+1} \right) \\
&= \left(\mathbf{P}_n + \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \right)^{-1} \sum_{k=1}^{n} y_k \mathbf{x}_k + \left(\mathbf{P}_n + \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \right)^{-1} \mathbf{x}_{n+1} y_{n+1} \\
&= \left(\mathbf{P}_n^{-1} - \frac{\mathbf{P}_n^{-1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n^{-1}}{1 + \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n^{-1} \mathbf{x}_{n+1}} \right) \sum_{k=1}^{n} y_k \mathbf{x}_k + \frac{\mathbf{P}_n^{-1} \mathbf{x}_{n+1}}{1 + \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n^{-1} \mathbf{x}_{n+1}} y_{n+1} \\
&= \mathbf{P}_n^{-1} \sum_{k=1}^{n} y_k \mathbf{x}_k^{\mathrm{T}} + \frac{\mathbf{P}_n^{-1} \mathbf{x}_{n+1}}{1 + \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{P}_n^{-1} \mathbf{x}_{n+1}} \left(y_{n+1} - \hat{\mathbf{w}}_n^{\mathrm{T}} \mathbf{x}_{n+1} \right) \\
&= \hat{\mathbf{w}}_{n+1} + \mathbf{K}_{n+1} \left(y_{n+1} - \hat{\mathbf{w}}_n^{\mathrm{T}} \mathbf{x}_{n+1} \right)
\end{aligned}$$

Recursive least squares

- Least squares problem:

$$\mathbf{P}_1 = \mathbf{x}_1 \mathbf{x}_1^{\mathrm{T}}$$

$$\mathbf{P}_{n+1}^{-1} = \mathbf{P}_n^{-1} - \frac{\mathbf{P}_n^{-1}\mathbf{x}_{n+1}\mathbf{x}_{n+1}^{\mathrm{T}}\mathbf{P}_n^{-1}}{1 + \mathbf{x}_{n+1}^{\mathrm{T}}\mathbf{P}_n^{-1}\mathbf{x}_{n+1}} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{x}_{n+1}^{\mathrm{T}}\right)\mathbf{P}_n^{-1}$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_{n+1} + \mathbf{K}_{n+1}\left(y_{n+1} - \hat{\mathbf{w}}_n^{\mathrm{T}}\mathbf{x}_{n+1}\right)$$

$$\mathbf{K}_{n+1} = \frac{\mathbf{P}_n^{-1}\mathbf{x}_{n+1}}{1 + \mathbf{x}_{n+1}^{\mathrm{T}}\mathbf{P}_n^{-1}\mathbf{x}_{n+1}}$$