Introduction to Sequential Prediction and the Kalman Filter

Paul Schrater

Problem overview

- Need state of system that change over time using noisy measurements on the system
- Input:

– (Noisy) Sensor measurements: $\mathbf{Z} = \{ \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \ldots , \mathbf{z}_T \}$

- Goal: $\hat{\mathbf{x}}(t)$

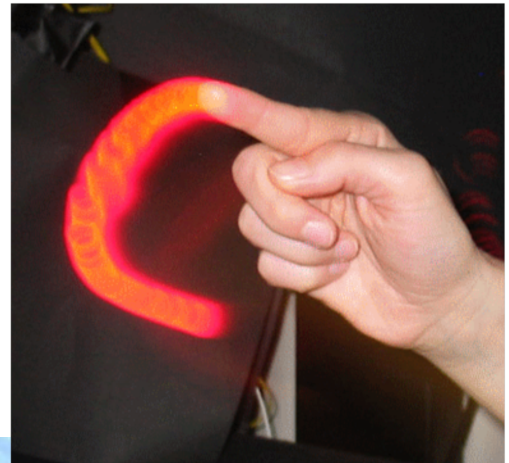
– Estimate most probable state at time τ using measurement up to time T

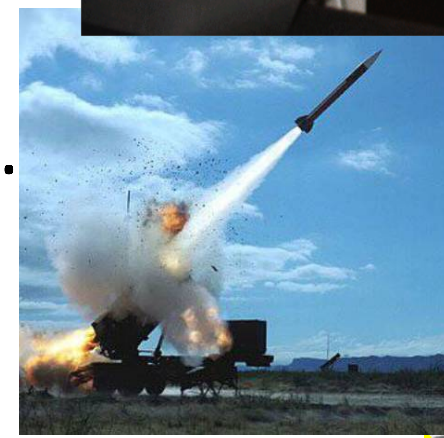
$\mathrm{T} < \tau$: **prediction** (extrapolation)

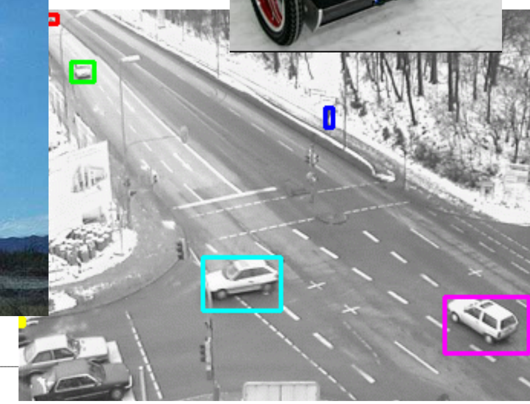
$\mathrm{T} > \tau$: **smoothing** (postdiction)

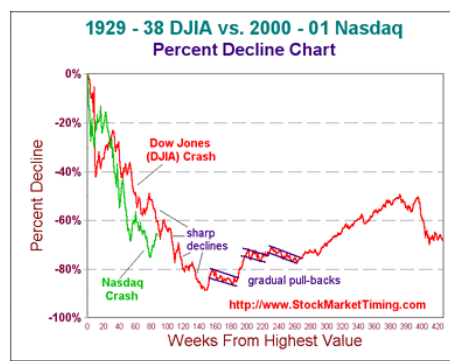
Applications

- Ballistics
- Motor Control
- Tracking hands/cars/…
- Econometrics
  - Stock prediction
- Cognitive Models
- GPS navigation
- Etc…









Stochastic Processes

- Deterministic process
  - Only one possible ‘reality’
- Random process
  - Several possible evolutions (starting point might be known)
  - Characterized by probability distributions

- Time series modeling
  - Sequence of random states/variables
  - Measurements available at discrete times

State space

- **The state vector** contains all available information to describe the investigated system
  - usually multidimensional: $X(k) \in R^{N_x}$
- **The measurement vector** represents observations related to the state vector $Z(k) \in R^{N_z}$
  - Generally (but not necessarily) of lower dimension than the state vector

Mean

Contribution of the N^{th} data point, x_N

$$
\begin{aligned}
\boldsymbol{\mu}_{\mathrm{ML}}^{(N)} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \\
&= \frac{1}{N}\mathbf{x}_N + \frac{1}{N}\sum_{n=1}^{N-1}\mathbf{x}_n \\
&= \frac{1}{N}\mathbf{x}_N + \frac{N-1}{N}\boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)} \\
&= \underbrace{\boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)}}_{\text{old estimate}} + \underbrace{\frac{1}{N}}_{\text{correction weight}}\underbrace{\left(\mathbf{x}_N - \boldsymbol{\mu}_{\mathrm{ML}}^{(N-1)}\right)}_{\text{correction given } \mathrm{x}_N}
\end{aligned}
$$

Breakpoint Exercises

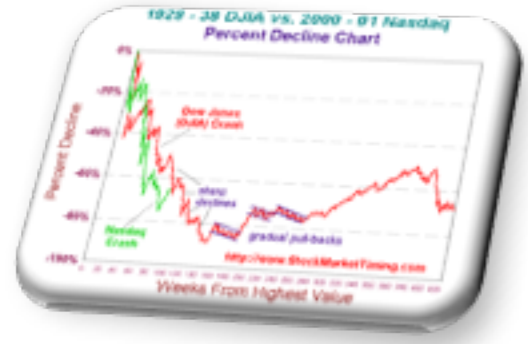
- Program recursive mean filter in matlab, apply to data in file: ‘kalmanjump.mat’
  - Replace the (1/N) weighting by a fixed constant, K
  - Try K = 1/10, and K=1/100

- Derive recursive variance filter

State space

- <u>Tracking:</u>

$N_x = 3$	$N_x = 4$
x	x
y	v_x
	y
	v_y



<u>Econometrics:</u>

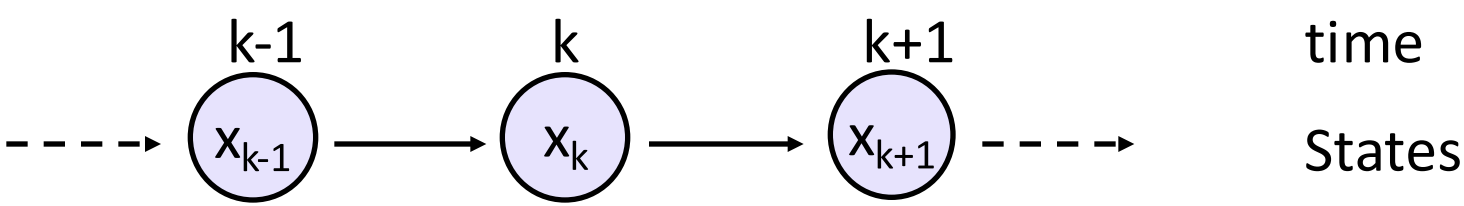
- Monetary flow
- Interest rates
- Inflation
- …

(First-order) Markov process

- The Markov property – the likelihood of a future state depends on present state only

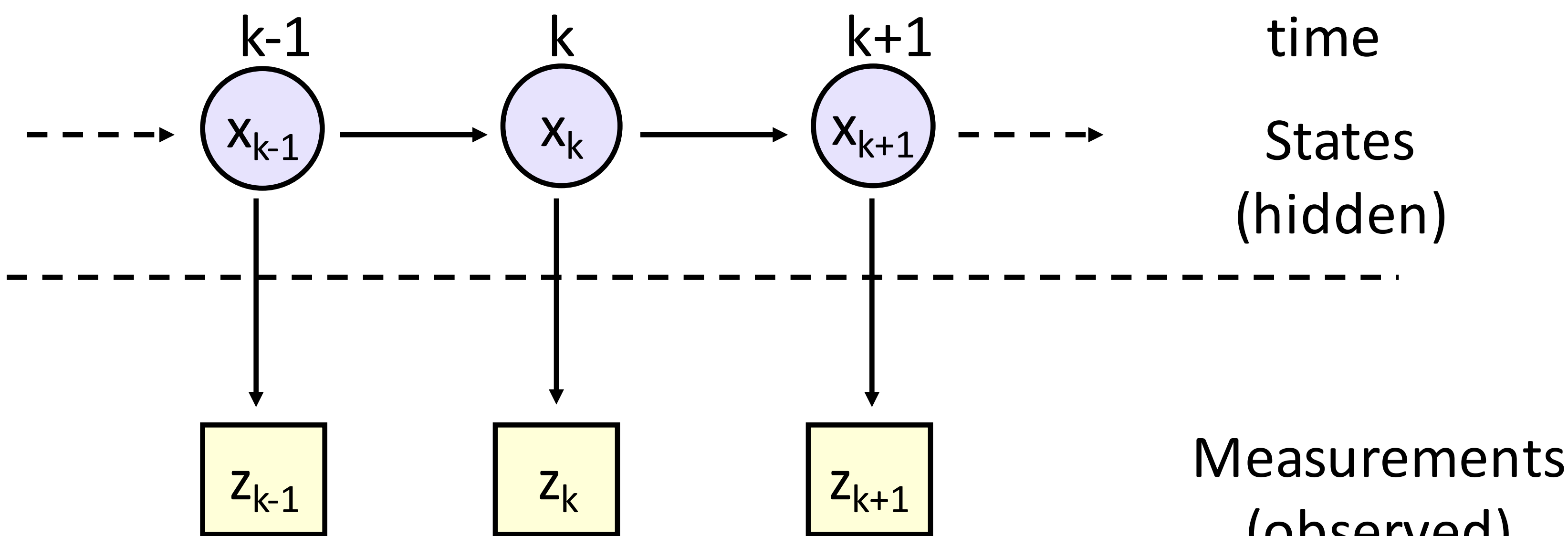
$$P(\ X(k+h) \mid X(k)\ )$$

- Markov chain – A stochastic process with Markov property

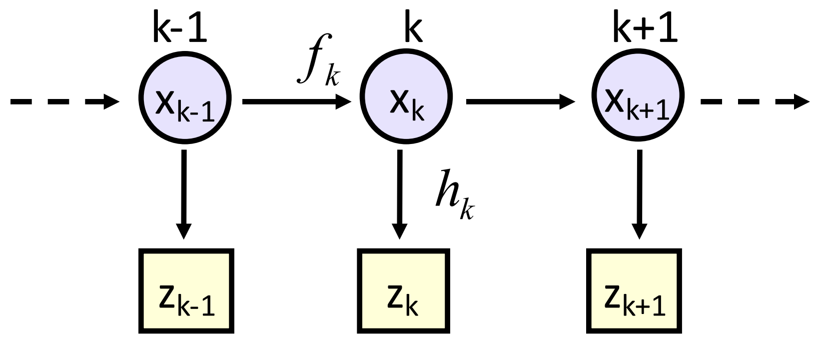


Hidden Markov Model (HMM)

- the state is not directly visible, but output dependent on the state is visible



Dynamic System



Stochastic diffusion

State equation: $x_k = f_k(x_{k-1}, \boxed{v_k})$

x_k state vector at time instant k

f_k state transition function, $f_k : R^{N_x} \cdot R^{N_v} \quad R^{N_x}$

v_k i.i.d process noise

Observation equation: $z_k = h_k(x_k, \boxed{w_k})$

z_k observations at time instant k

h_k observation function, $h_k : R^{N_x} \cdot R^{N_w} \quad R^{N_z}$

w_k i.i.d measurement noise

A simple dynamic system

- $X = [x, y, v_x, v_y]$ (4-dimensional state space)
- Constant velocity motion:

$$f(X, v) = [x + \quad t \;\; v_x, y + \quad t \;\; v_y, v_x, v_y] + v$$

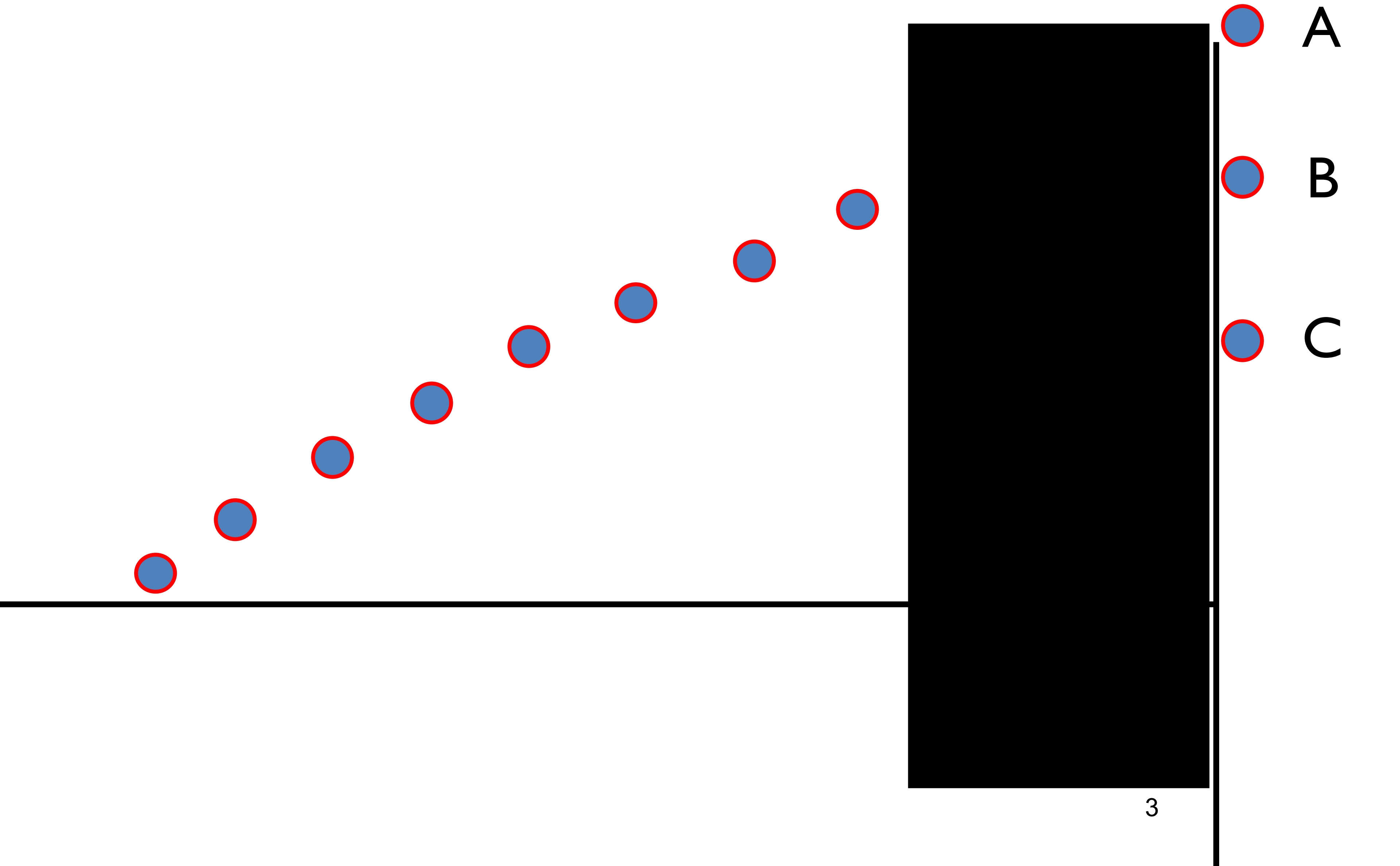
$$v \sim N(0, Q) \qquad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & q^2 \end{bmatrix}$$

- Only position is observed:

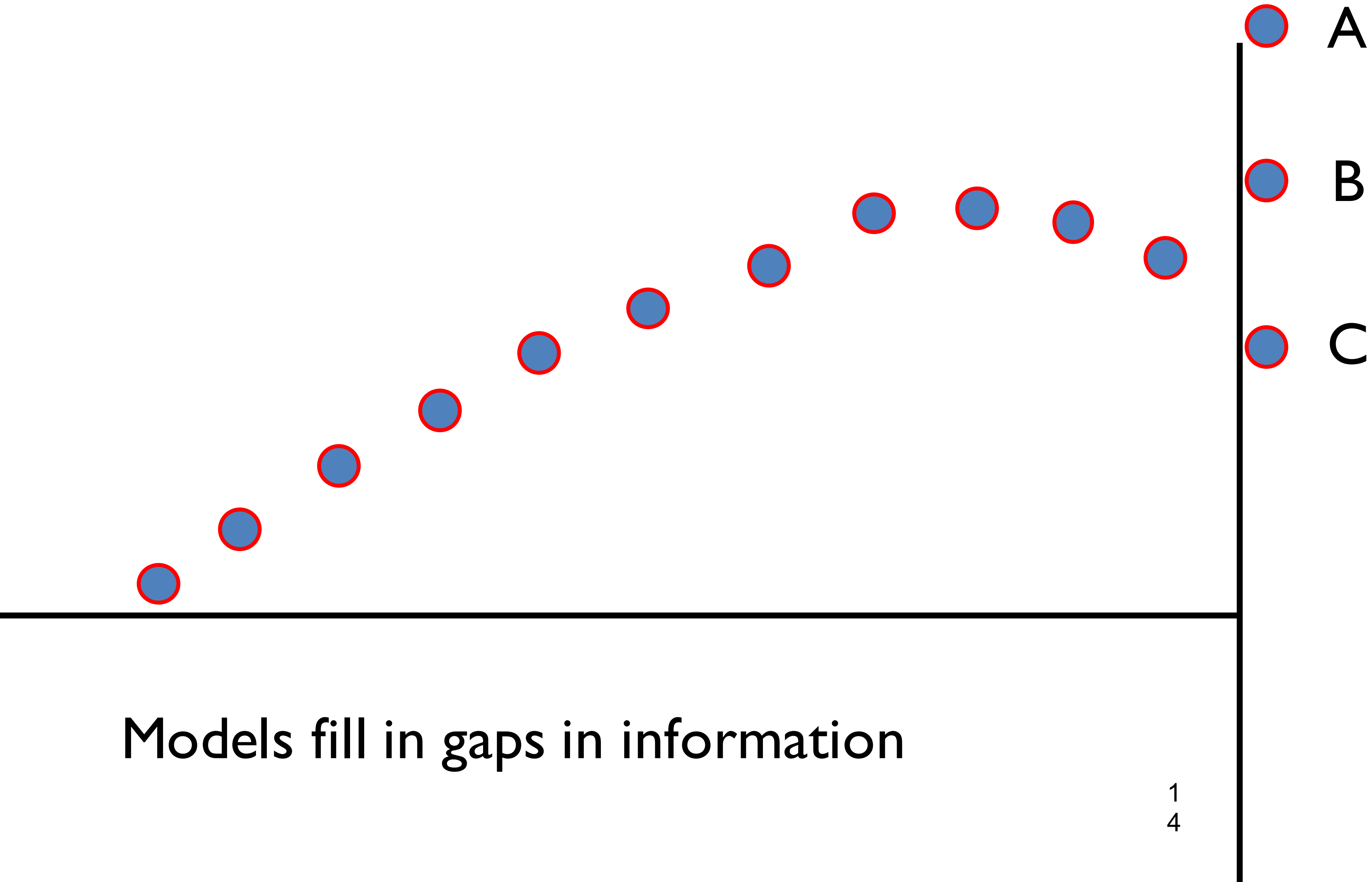
$$z = h(X, w) = [x, y] + w$$

$$w \sim N(0, R) \qquad R = \begin{bmatrix} r^2 & 0 \\ 0 & r^2 \end{bmatrix}$$

Model example

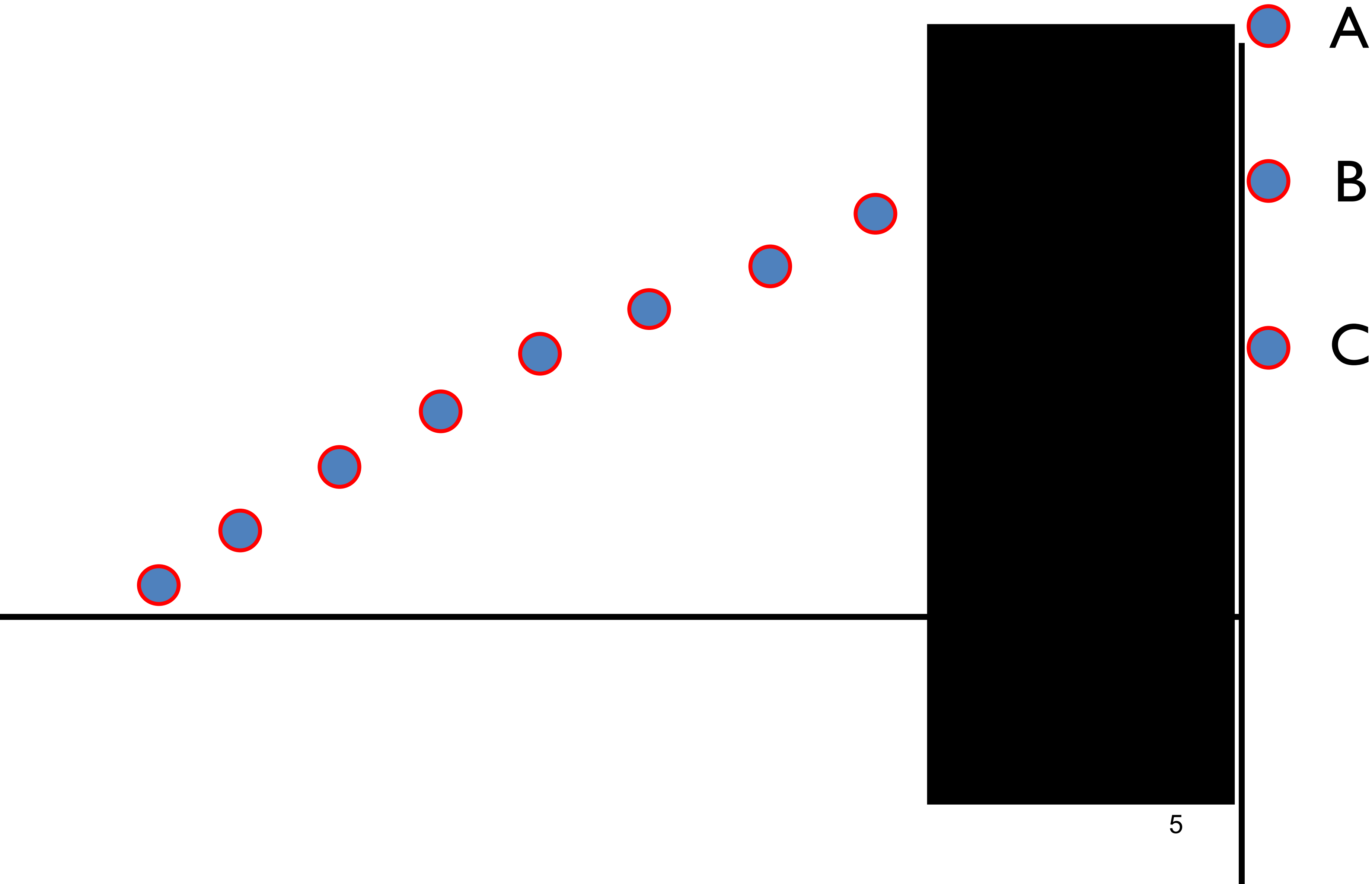


Model Example

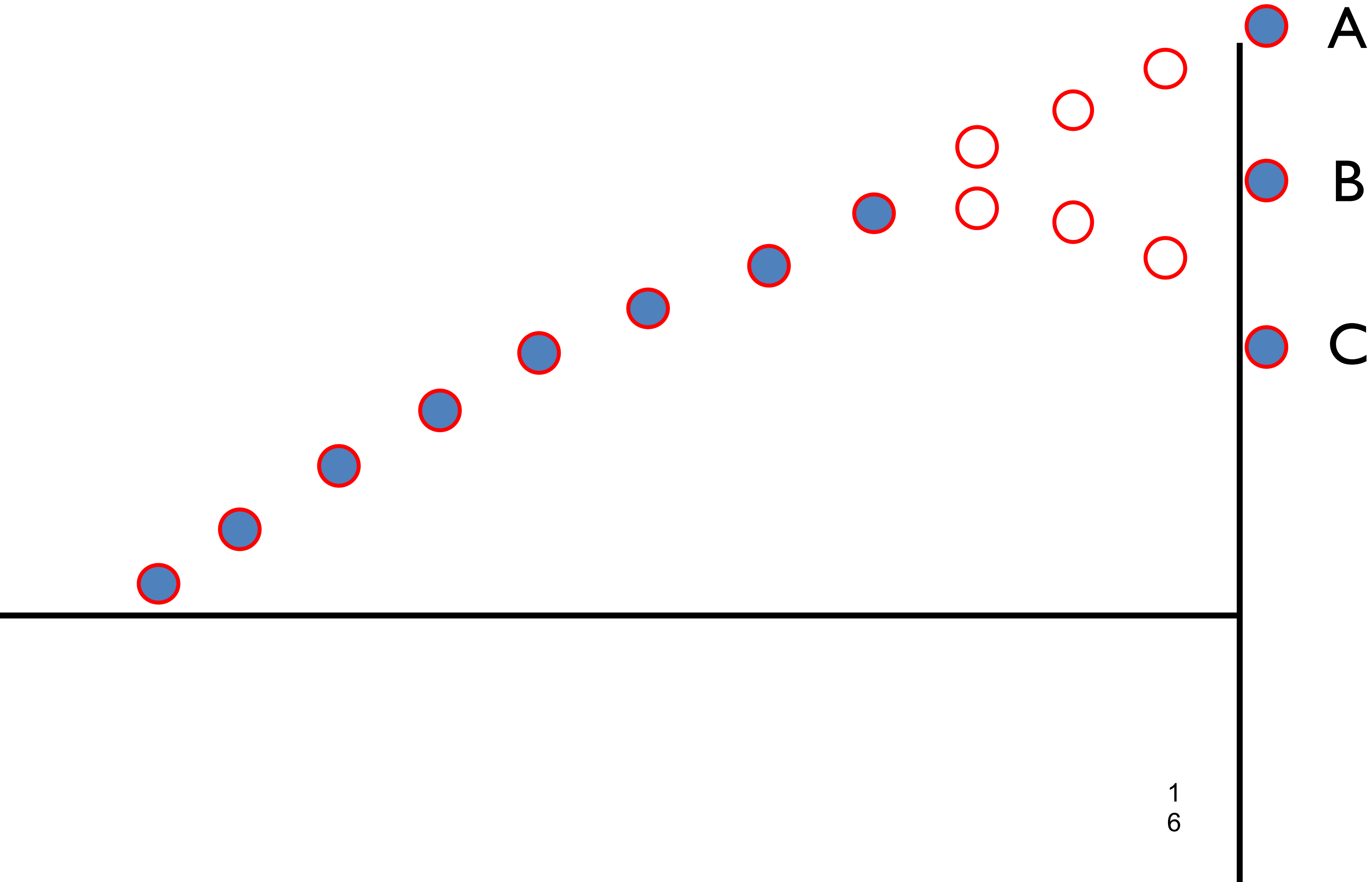


Models fill in gaps in information

Perceptual Kalman Example



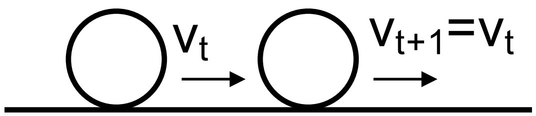
Extrapolation depends on model



What are Human default Motion Models?

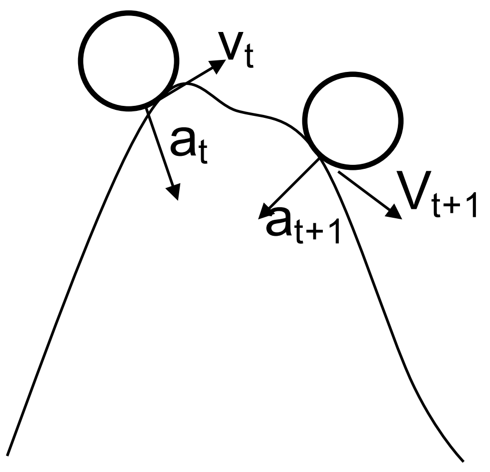
Object velocity: speed 5 m/s
direction south

1. Constant velocity (**CV**)



-maintain speed and direction

2. Constant acceleration (**CA**)



-constant change in speed and/or direction

Motion extrapolation task

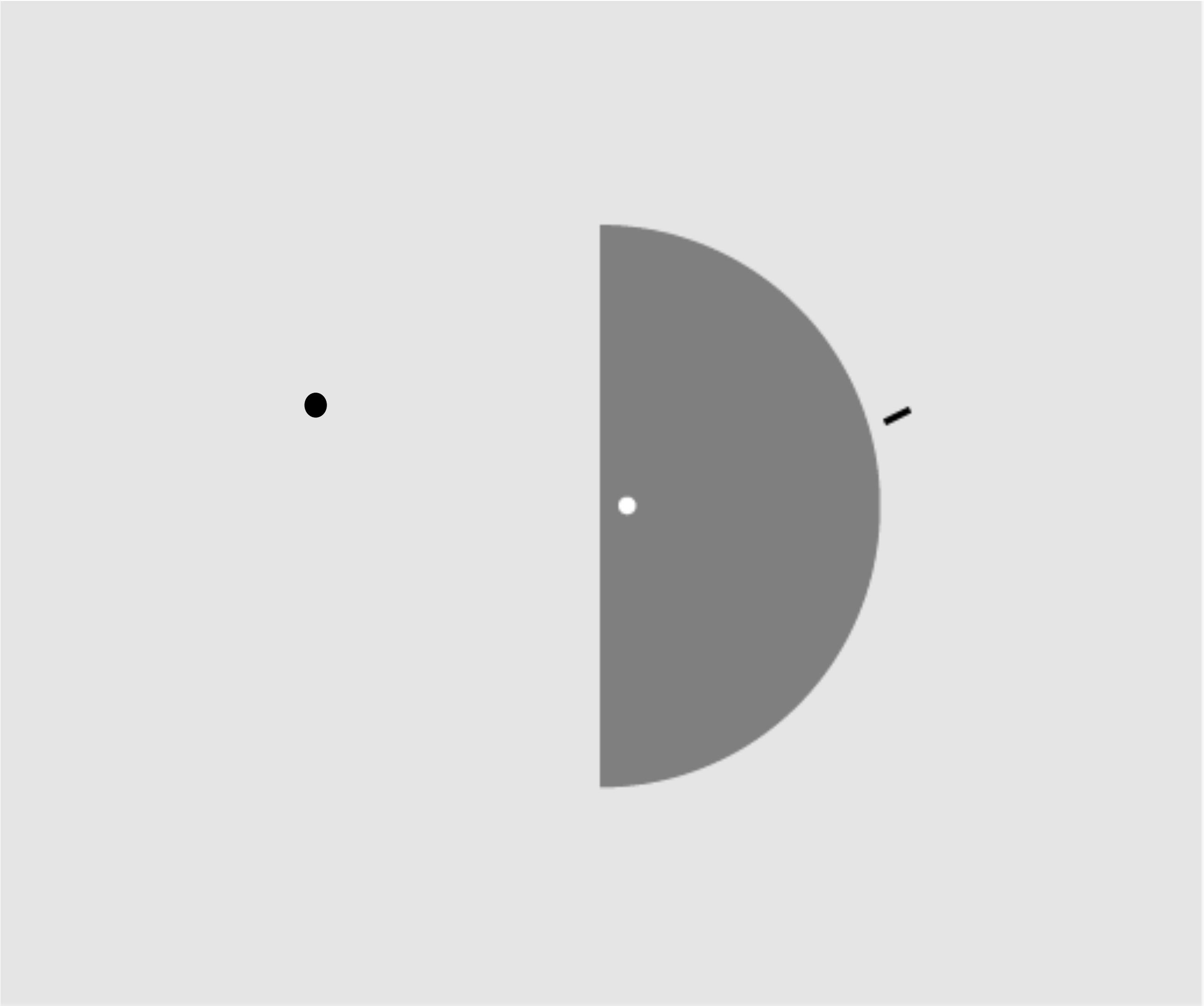
-Fixation

-After 500ms dot travels

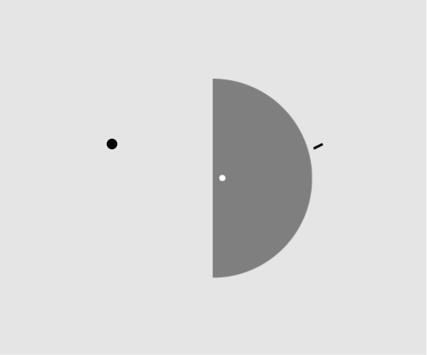
-Extrapolation judgment:
“above” or “below”

-No reemergence;
no feedback

-Determine the PSE
based on staircase
procedure



Motion extrapolation: Kalman filters for simple motions



Parameters of dot motion:

$$x_k = [x, y, vx, vy, ax, ay]_k^T$$

position velocity acceleration

Process:

True state: $$x_k = \boxed{A_k} x_{k-1} + \boxed{w_{k-1}}$$

$$\begin{pmatrix} x_k \\ y_k \\ vx_k \\ vy_k \\ ax_k \\ ay_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ vx_{k-1} \\ vy_{k-1} \\ ax_{k-1} \\ ay_{k-1} \end{pmatrix} + \begin{pmatrix} w_{x_{k-1}} \\ w_{y_{k-1}} \\ w_{vx_{k-1}} \\ w_{vy_{k}-1} \\ w_{ax_{k-1}} \\ w_{ay_{k-1}} \end{pmatrix}$$

"w" ~N(0,Q),
"Q" = covariance; reflects trust in prior ("A")

Q = 0 → complete trust

"A" represents the prior model in the absence of data

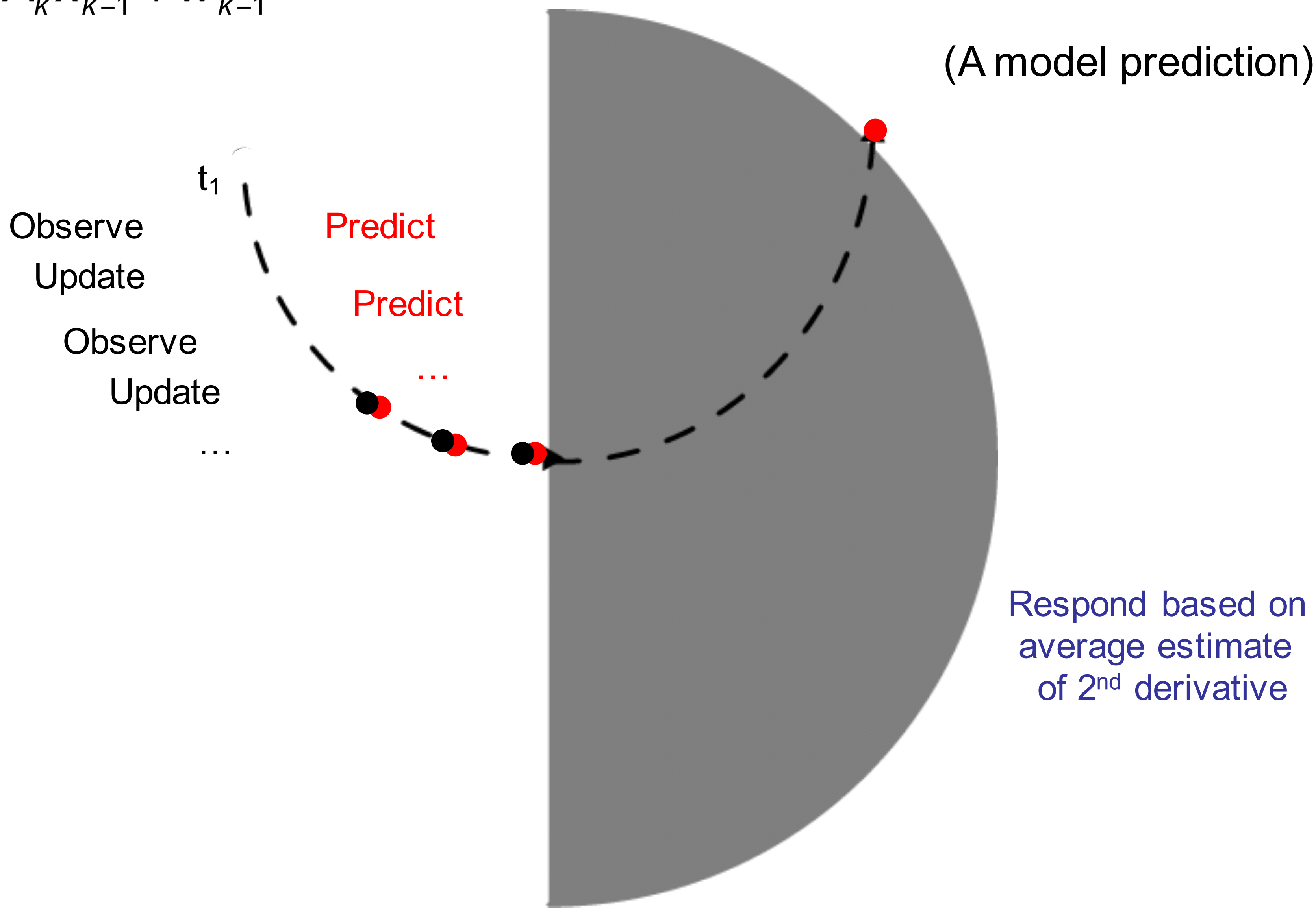
→ **CV:** constant speed & direction: Linear motion prior

→ **CA:** constant change in direction: Circular motion prior

Motion extrapolation: Model behavior

CA prediction using a Kalman filter

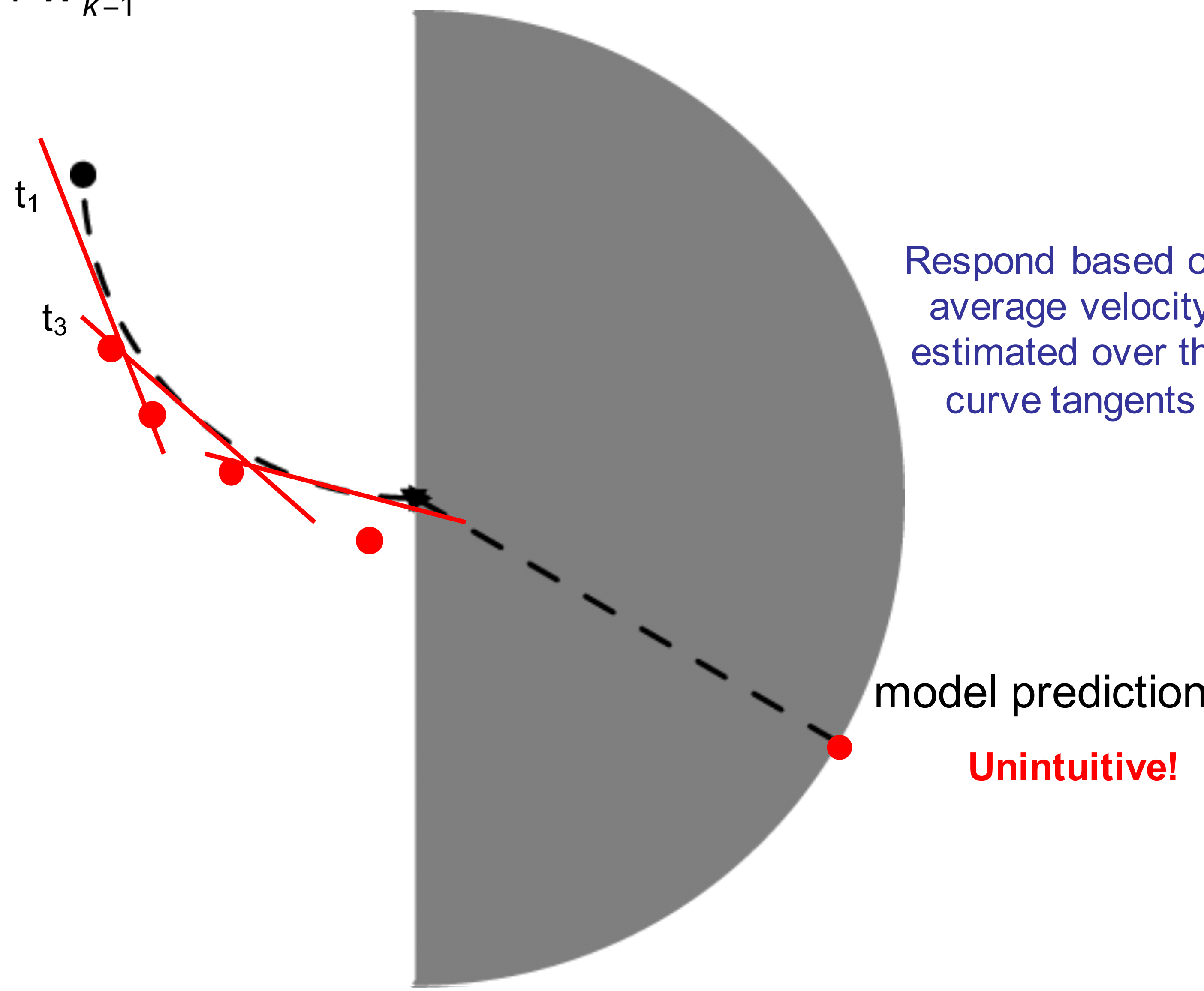
$$x_k = A_k x_{k-1} + w_{k-1}$$



Motion extrapolation: Model behavior

CV prediction using a Kalman filter

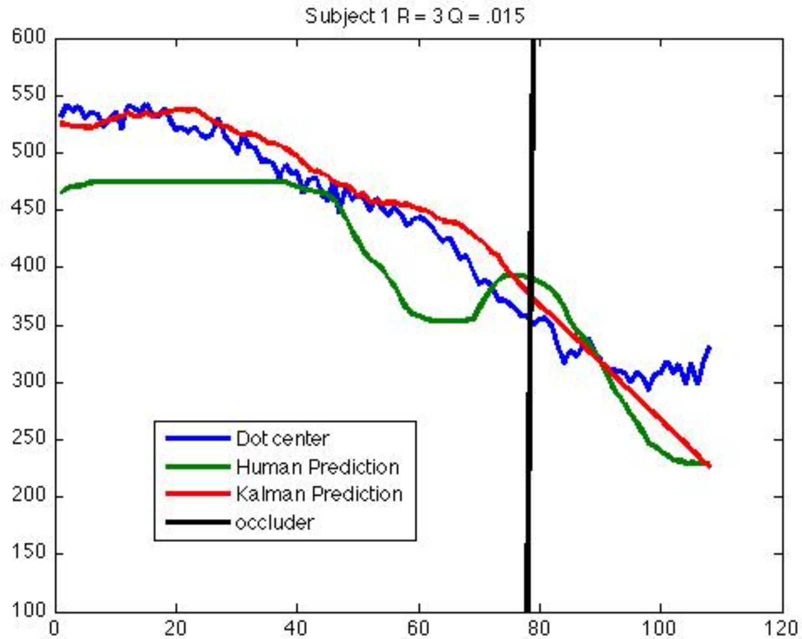
$$x_k = A_k x_{k-1} + w_{k-1}$$

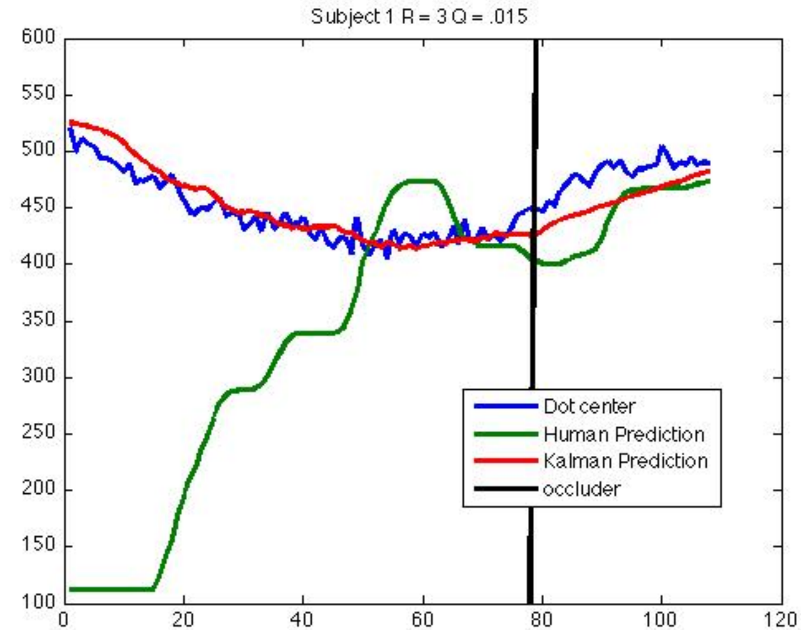


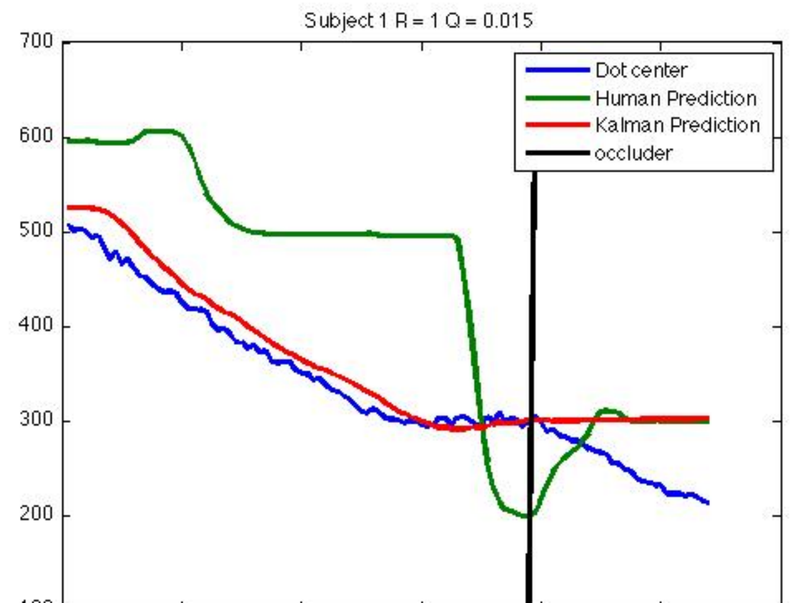
Respond based on average velocity estimated over the curve tangents

Unintuitive!

Humans vs. Kalman Filter

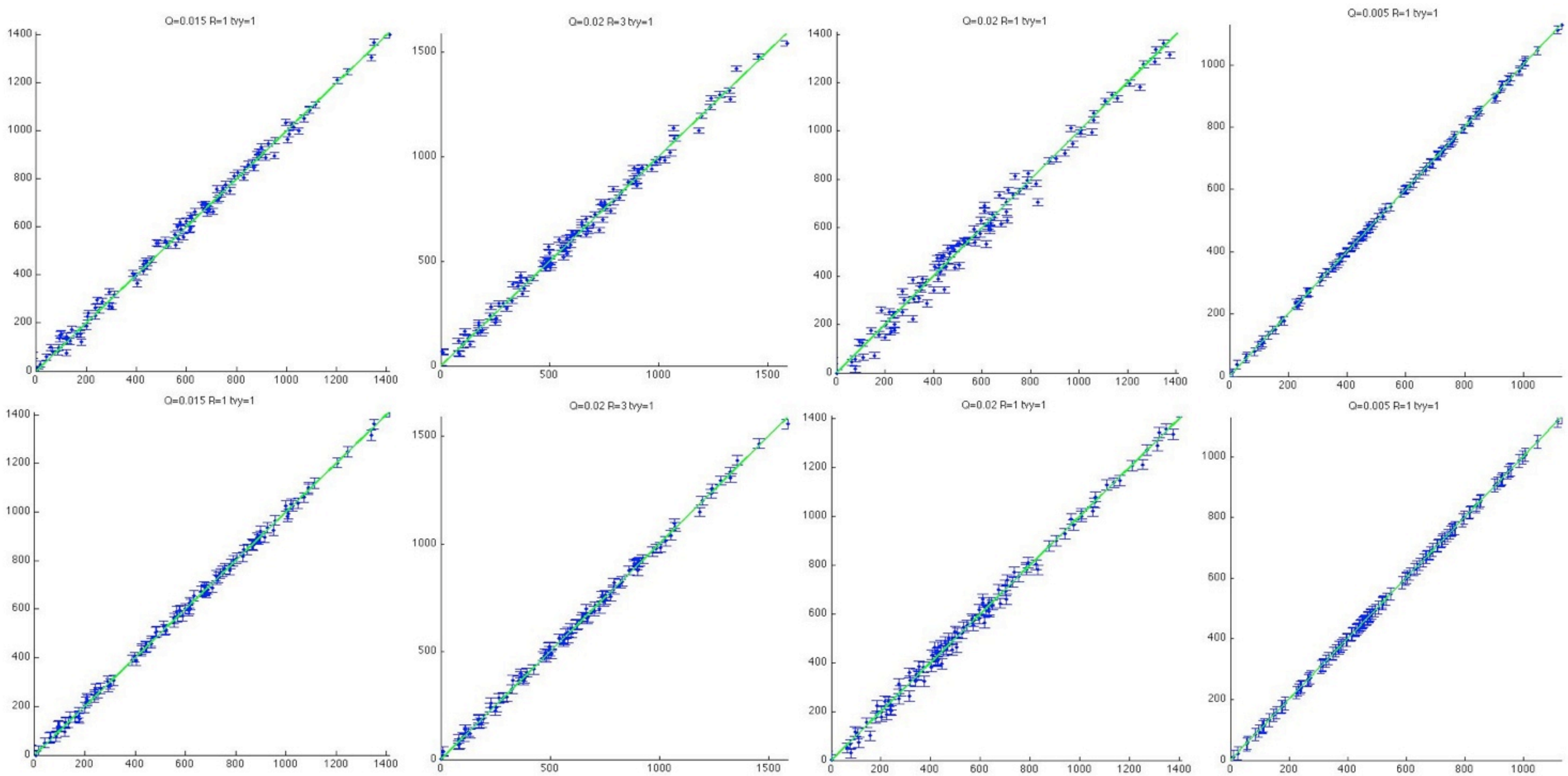






- Demonstration of the task, human vs. filter performance
- Kalman filter predicts human behavior well

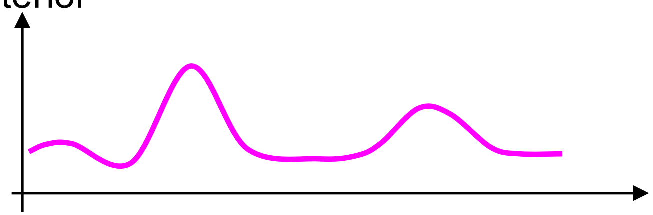
Matched Kalman excellent predictor



The Bayesian approach

Thomas Bayes

- Construct the posterior probability density function $p(x_k \mid z_{1:k})$ of the state based on all available information



- By knowing the posterior many kinds of estimates for x_k can be derived
  - mean (expectation), mode, median, …
  - Can also give estimation of the accuracy (e.g. covariance)

Recursive filters

- For many problems, estimate is required each time a new measurement arrives

- **Batch** processing
  - Requires *all* available data
- **Sequential** processing
  - New data is processed upon arrival
  - Need not store the complete dataset
  - Need not reprocess all data for each new measurement

  - Assume no out-of-sequence measurements (solutions for this exist as well…)

Recursive Bayes filters

- Given:
  - System models in probabilistic forms

$$x_k = f_k(x_{k-1}, v_k) \qquad p(x_k \mid x_{k-1})$$

$$z_k = h_k(x_k, w_k) \qquad p(z_k \mid x_k)$$

Markovian process

Measurements are conditionally independent given the state

    (known statistics of v_k, w_k)

  - Initial state $p(x_0 \mid z_0) = p(x_0)$ also known as the **prior**
  - Measurements $z_1, \ldots, z_k$

Recursive Bayes filters

- Prediction step (a-priori)

$$p(x_{k-1} \mid z_{1:k-1}) \qquad p(x_k \mid z_{1:k-1})$$

  - Uses the system model to predict forward
  - Deforms/translates/spreads state pdf due to random noise

- Update step (a-posteriori)

$$p(x_k \mid z_{1:k-1}) \qquad p(x_k \mid z_{1:k})$$

  - Update the prediction in light of new data
  - Tightens the state pdf

General prediction-update framework

- Assume $p(x_{k-1} \mid z_{1:k-1})$ is given at time k-1
- Prediction:

$$p(x_k \mid z_{1:k-1}) = \int \overbrace{p(x_k \mid x_{k-1})}^{\text{System model}} \overbrace{p(x_{k-1} \mid z_{1:k-1})}^{\text{Previous posterior}} dx_{k-1} \qquad (1)$$

- Using Chapman-Kolmogorov identity + Markov property

General prediction-update framework

- Update step

$$p(x_k \mid z_{1:k}) = p(x_k \mid z_k, z_{1:k-1})$$

$$\boxed{p(A \mid B, C) = \frac{p(B \mid A, C)\, p(A \mid C)}{p(B \mid C)}}$$

$$= \frac{p(z_k \mid x_k, z_{1:k-1})\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}$$

Measurement model

Current prior

$$\frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

$$= \frac{p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \qquad (2)$$

Normalization constant

Where $p(z_k \mid z_{1:k-1}) = \quad p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})\, dx_k$

Generating estimates

- Knowledge of $p(x_k \mid z_{1:k})$ enables to compute optimal estimate with respect to any criterion. e.g.
  - Minimum mean-square error (MMSE)

$$\hat{x}_{k|k}^{MMSE} \quad E\left[x_k \mid z_{1:k}\right] = \quad x_k\, p(x_k \mid z_{1:k}) dx_k$$

  - Maximum a-posteriori

$$\hat{x}_{k|k}^{MAP} \quad \arg\max_{x_k} p(x_k \mid z_k)$$

General prediction-update framework

➔So (1) and (2) give optimal solution for the recursive estimation problem!

- Unfortunately no… only conceptual solution
  - integrals are intractable…
  - Can only implement the pdf to finite representation!

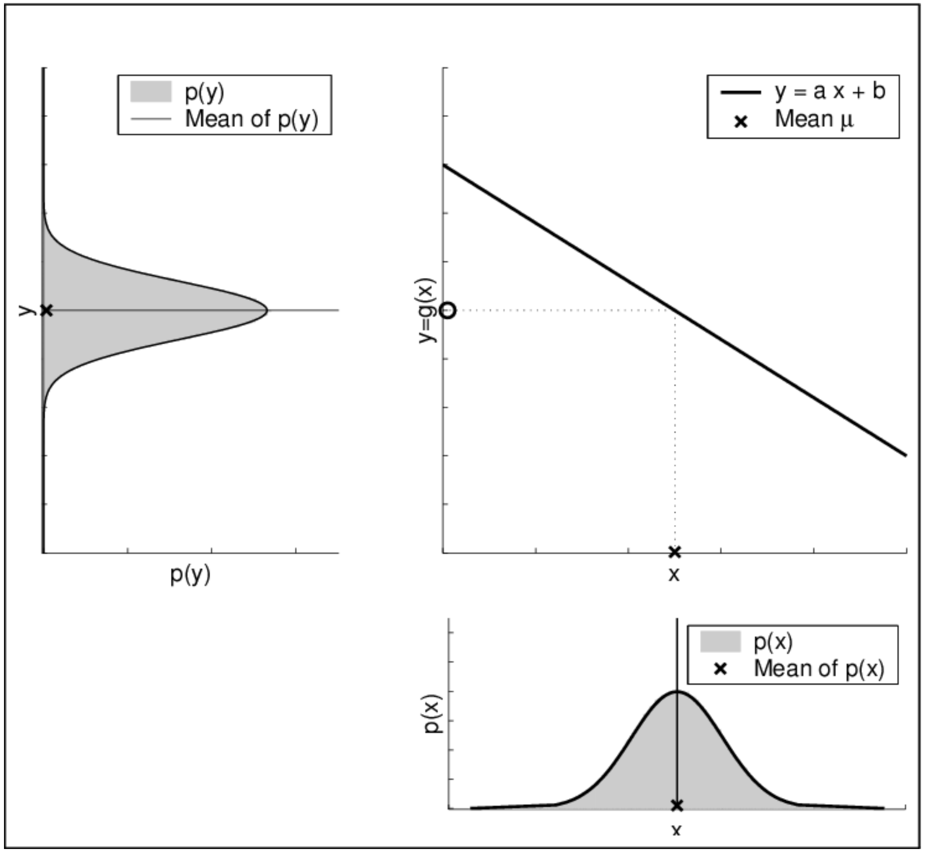
- However, optimal solution *does* exist for several restrictive cases

Restrictive case #1

- Posterior at each time step is Gaussian
  - Completely described by mean and covariance
- If $p(x_{k-1} \mid z_{1:k-1})$ is Gaussian it can be shown that $p(x_k \mid z_{1:k})$ is also Gaussian provided that:
  - v_k, w_k are Gaussian
  - f_k, h_k are linear

Restrictive case #1

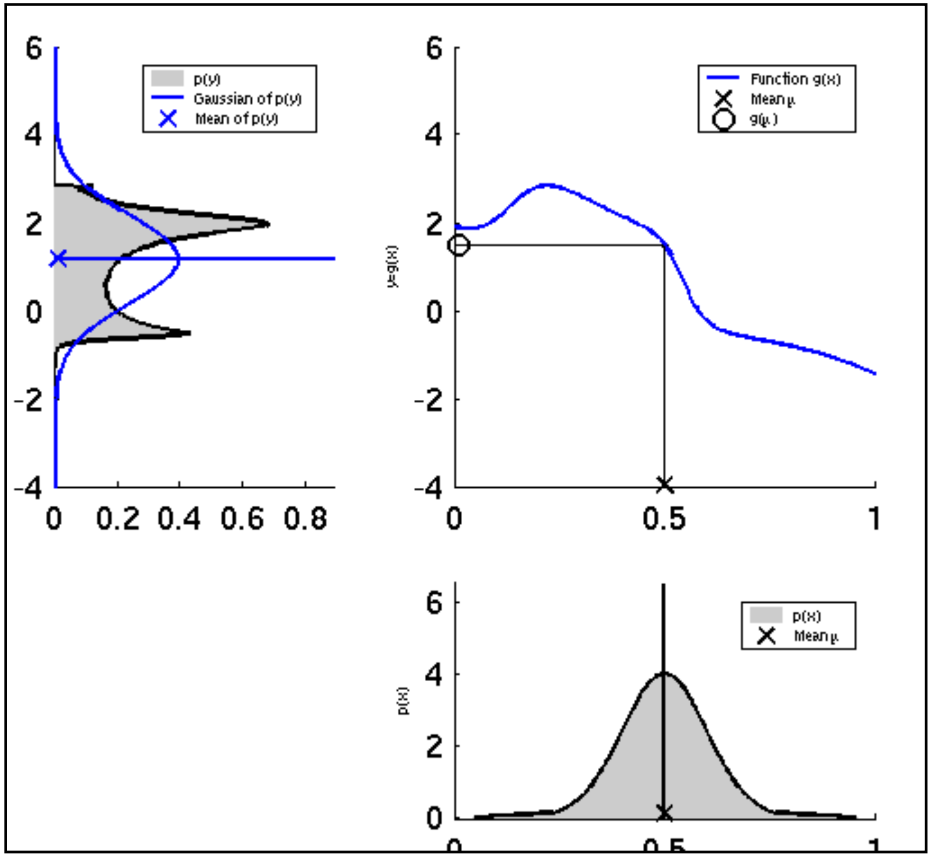
- Why Linear?



$$y = Ax + B \qquad p(y) \sim N\left(Ax + B, ACA^T\right)$$

Restrictive case #1

- Why Linear?



$$y = g(x) \;\; / \;\; p(y) \neq N(\;)$$

Restrictive case #1

- Linear system with additive noise

$$
\begin{aligned}
x_k &= f_k(x_{k-1}, v_k) \\
z_k &= h_k(x_k, w_k)
\end{aligned}
\qquad\qquad
\begin{aligned}
x_k &= F_k x_{k-1} + v_k \\
z_k &= H_k x_k + w_k \\
v_k &\sim N(0, Q_k) \\
w_k &\sim N(0, R_k)
\end{aligned}
$$

- Simple example again

$$f(X,v) = [x + \quad t \ \ v_x, y + \quad t \ \ v_y, v_x, v_y] + v \qquad\qquad z = h(X,w) = [x, y] + w$$

$$
\begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix}
= \underbrace{\begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{F}
\begin{bmatrix} x_k \\ y_k \\ v_{x,k-1} \\ v_{y,k-1} \end{bmatrix} + N(0, Q_k)
\qquad
\begin{bmatrix} x_{obs} \\ y_{obs} \end{bmatrix}
= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{H}
\begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix} + N(0, R_k)
$$

The Kalman filter

Rudolf E. Kalman

$$p(x_{k-1} \mid z_{1:k-1}) = N(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$$
$$p(x_k \mid z_{1:k-1}) = N(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$$
$$p(x_k \mid z_{1:k}) = N(x_k; \hat{x}_{k|k}, P_{k|k})$$

- Substituting into (1) and (2) yields the predict and update equations

The Kalman filter

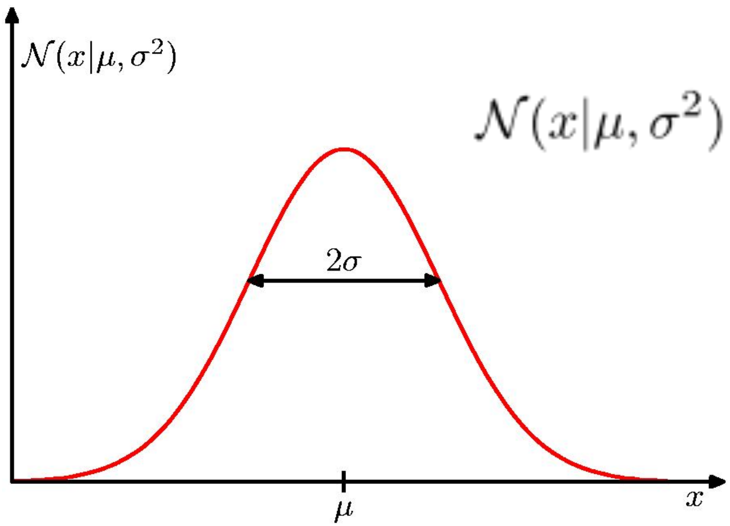
Predict:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

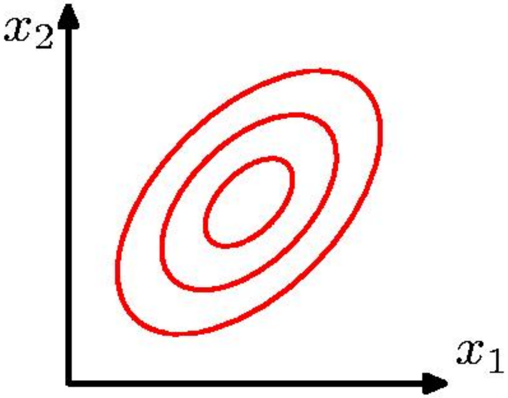
Update:

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right)$$
$$P_{k|k} = \left[I - K_k H_k \right] P_{k|k-1}$$

The Gaussian Distribution



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

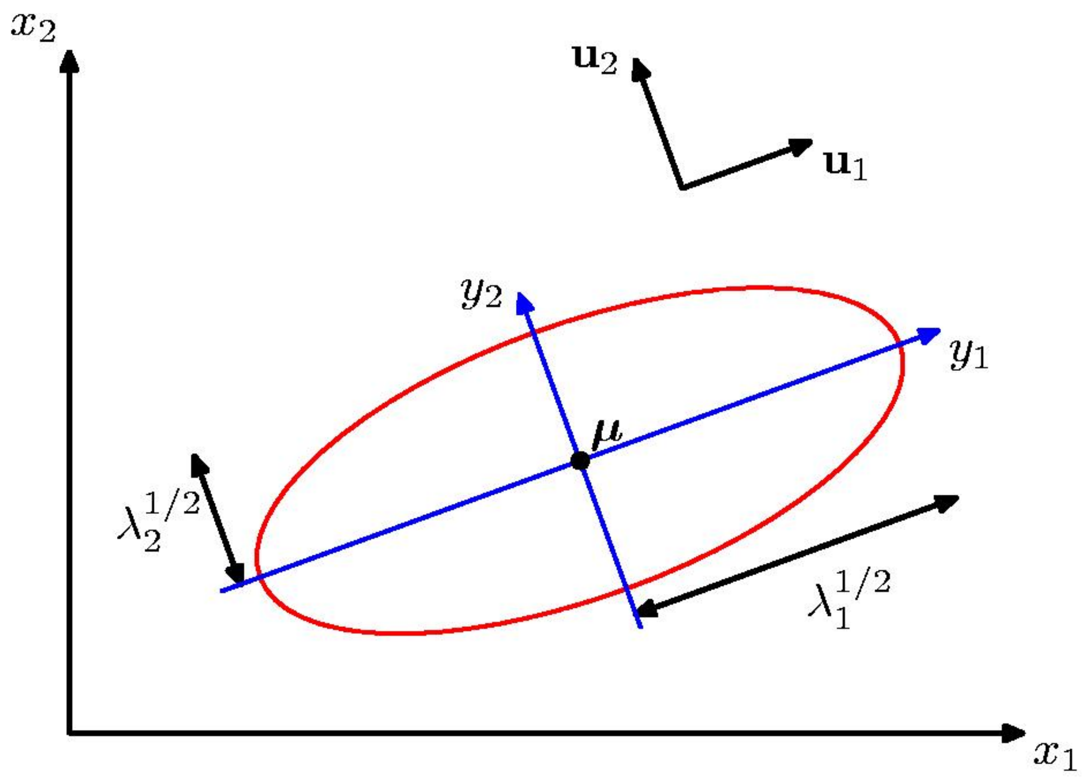
Geometry of the Multivariate Gaussian

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$$

$$\Delta^2 = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^{\mathrm{T}} (\mathbf{x} - \boldsymbol{\mu})$$



Moments of the Multivariate Gaussian (1)

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}\mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\}(\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z}\end{aligned}$$

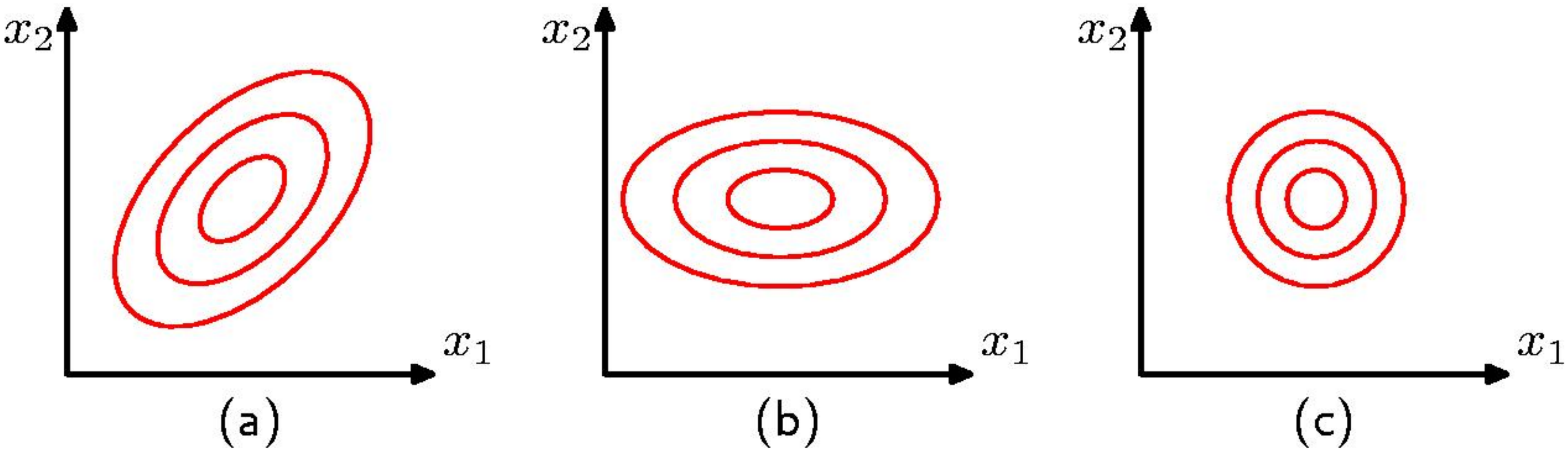
thanks to anti-symmetry of z

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

Moments of the Multivariate Gaussian (2)

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma}$$

$$\operatorname{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$



Partitioned Conditionals and Marginals

Conditionals

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \\
\boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\right\} \\
&= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\
&= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)
\end{aligned}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}$$

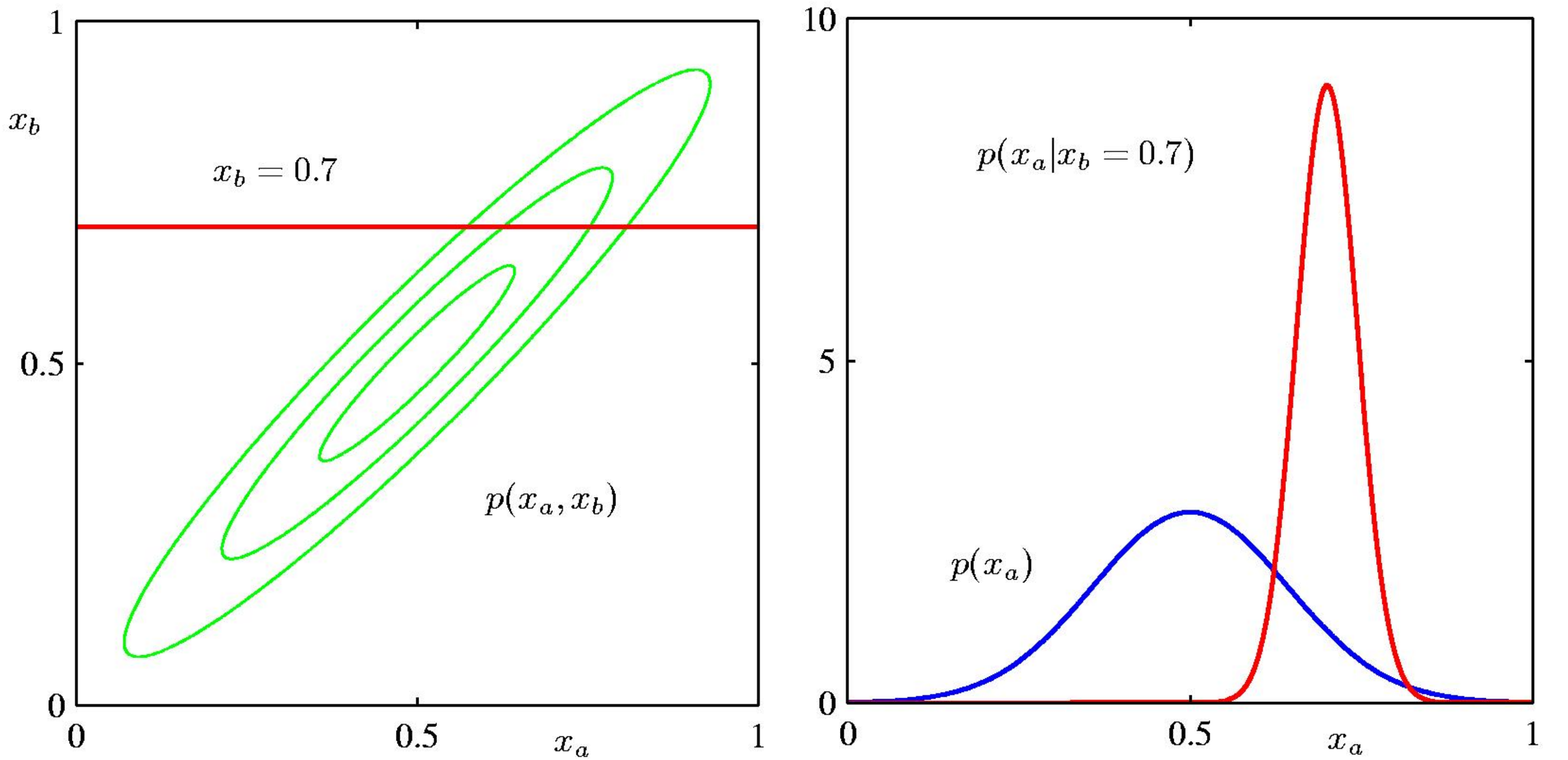
Marginals

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

$$\begin{aligned}
p(\mathbf{x}_a) &= \int p(\mathbf{x}_a, \mathbf{x}_b)\,\mathrm{d}\mathbf{x}_b \\
&= \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})
\end{aligned}$$

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}$$

Partitioned Conditionals and Marginals



Bayes' Theorem for Gaussian Variables

- Given

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right) \end{aligned}$$

- we have

$$\begin{aligned} p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \\ p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma})^{-1} \end{aligned}$$

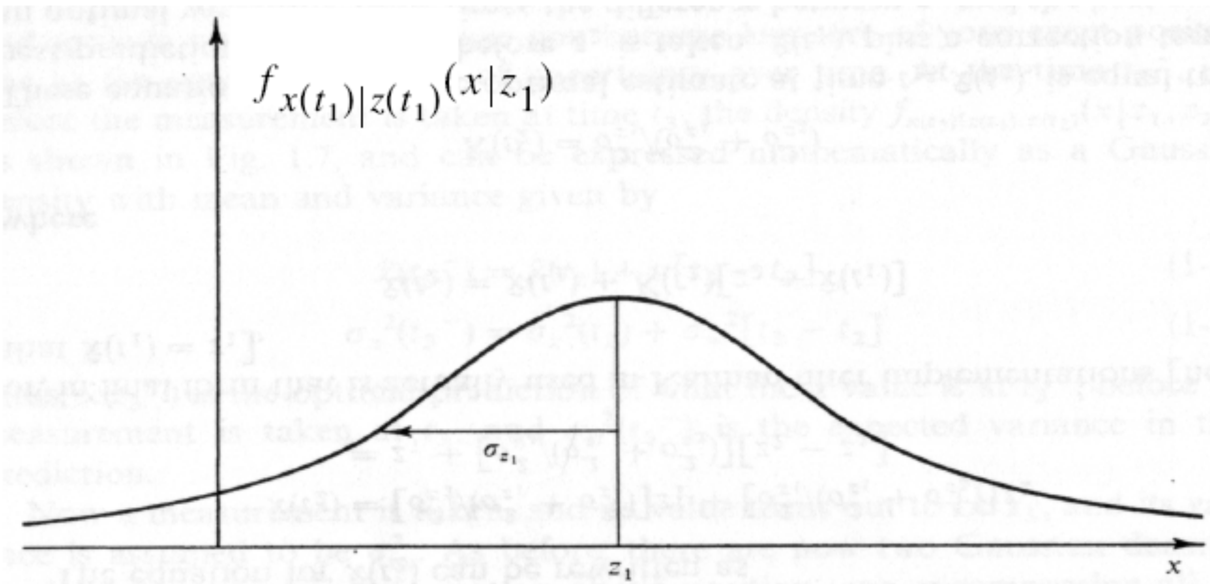
- where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$$

Example – cont'd

- Time t1: Star Sighting
  - Denote $x(t1)=z1$
- Uncertainty (inaccuracies, human error, etc)
  - Denote $\sigma 1$ (normal)
- Can establish the conditional probability of $x(t1)$ given measurement $z1$

Example – cont'd

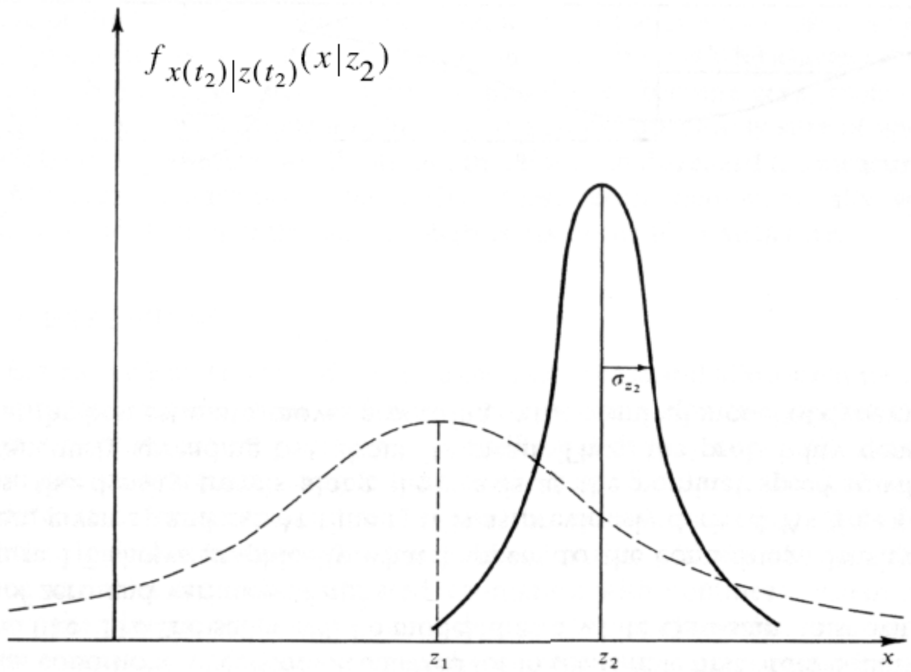


- Probability for any location, based on measurement
- For Gaussian density – 68.3% within $\pm\sigma 1$
- Best estimate of position: Mean/Mode/Median

$$\hat{x}(t_1) = z_1 \qquad \sigma_x^2(t_1) = \sigma_{z_1}^2$$

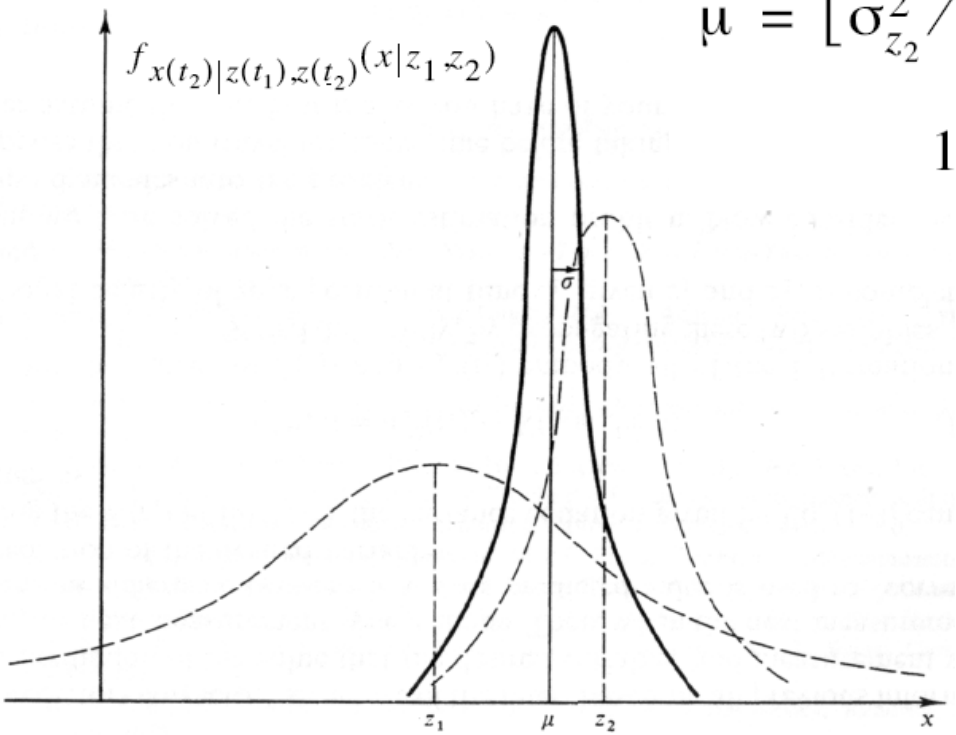
Example – cont'd

- Time t2≅t1: friend (more trained)
  - x(t2)=z2, σ(t2)=σ2
  - Since she has higher skill: σ2<σ1



Example – cont'd

- f(x(t2)|z1,z2) also Gaussian



$$\mu = [\sigma_{z_2}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2+\sigma_{z_2}^2)]z_2$$

$$1/\sigma^2 = (1/\sigma_{z_1}^2) + (1/\sigma_{z_2}^2)$$

Intuition via 1D example

- Lost at sea
  - Night
  - Drifting
  - Vague idea of location
  - For simplicity – let's assume 1D

* Example and plots by Maybeck, *"Stochastic models, estimation and control, volume 1"*

Example – cont'd

$$\mu = [\sigma_{z_2}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2)]z_2$$

$$1/\sigma^2 = (1/\sigma_{z_1}^2) + (1/\sigma_{z_2}^2)$$

- σ less than both $\sigma 1$ and $\sigma 2$
- $\sigma 1 = \sigma 2$: average
- $\sigma 1 > \sigma 2$: more weight to z2
- Rewrite:

$$\begin{aligned}\hat{x}(t_2) &= [\sigma_{z_2}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2)]z_2 \\ &= z_1 + [\sigma_{z_1}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2)][z_2 - z_1]\end{aligned}$$

Example – cont'd

- The Kalman update rule:

Best estimate
Given z2
(a poseteriori)

$$\boxed{\hat{x}(t_2)} = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)]$$

$$K(t_2) = \sigma_{z_1}^2 / (\sigma_{z_1}^2 + \sigma_{z_2}^2)$$

Best Prediction prior to z2
(*a priori*)

Optimal Weighting
(*Kalman Gain*)

Residual

Kalman gain

$$
\begin{aligned}
S_k &= H_k P_{k|k-1} H_k^T + R_k \\
K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right) \\
P_{k|k} &= \left[I - K_k H_k \right] P_{k|k-1}
\end{aligned}
$$

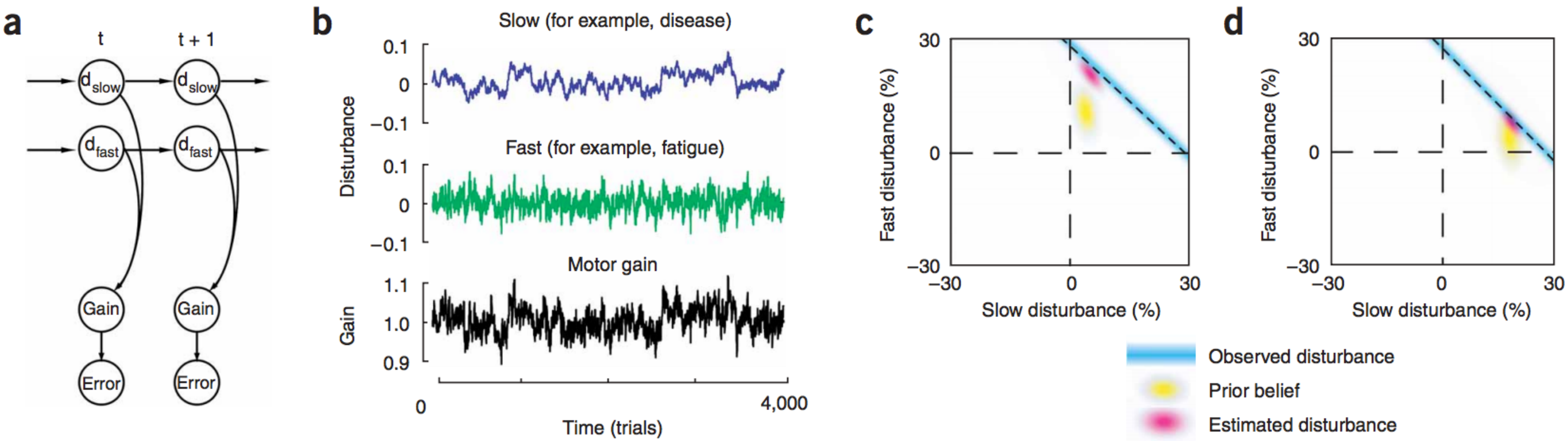
- Small measurement error:

$$\lim_{\mathbf{R}_k \to 0} K_k = H_k \qquad \lim_{\mathbf{R}_k \to 0} \hat{x}_{k|k} = H_k \; z_k$$

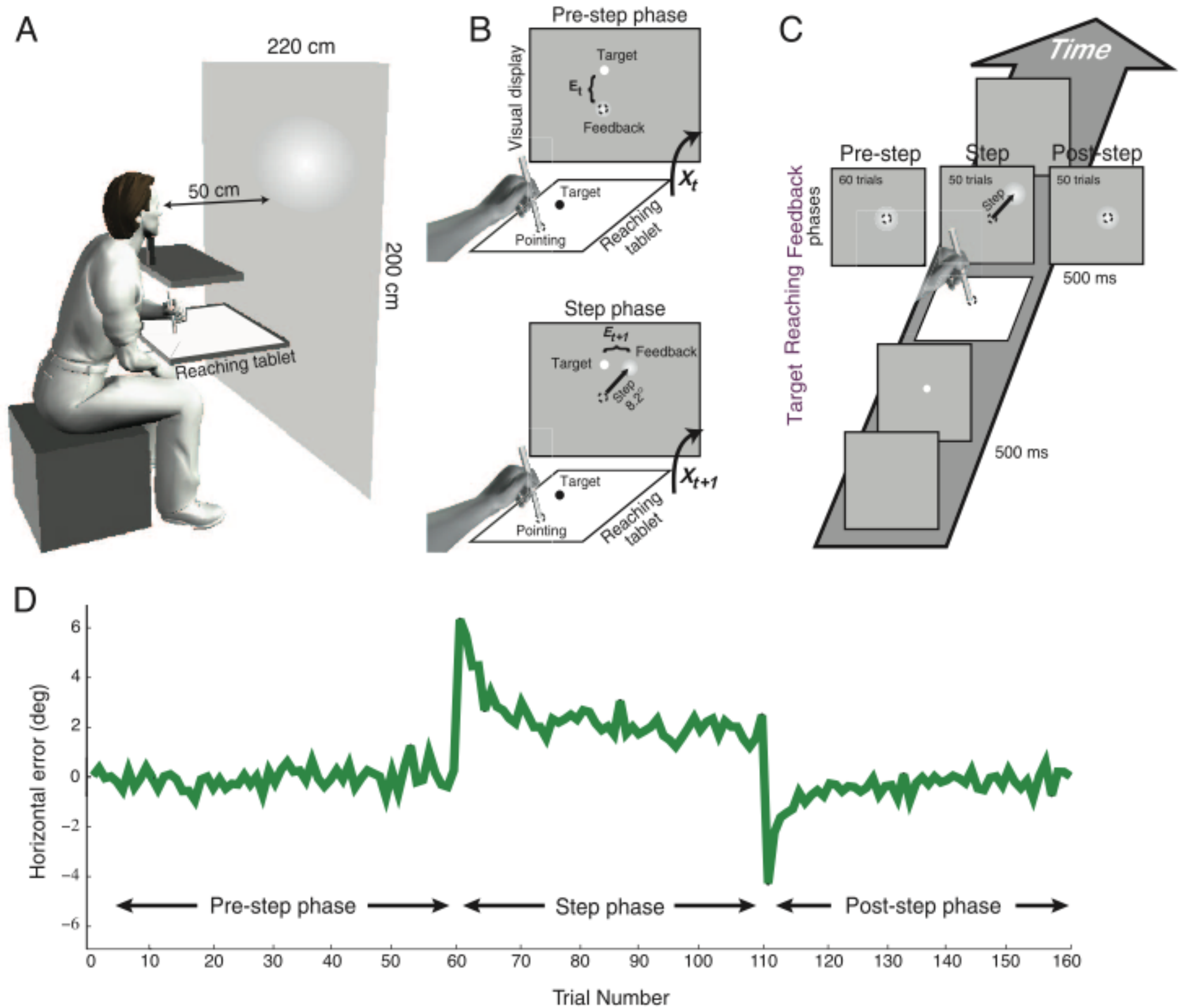
- Small prediction error:

$$\lim_{\mathbf{P}_k \to 0} K_k = 0 \qquad \lim_{\mathbf{P}_k \to 0} \hat{x}_{k|k} = \hat{x}_{k|k-1}$$

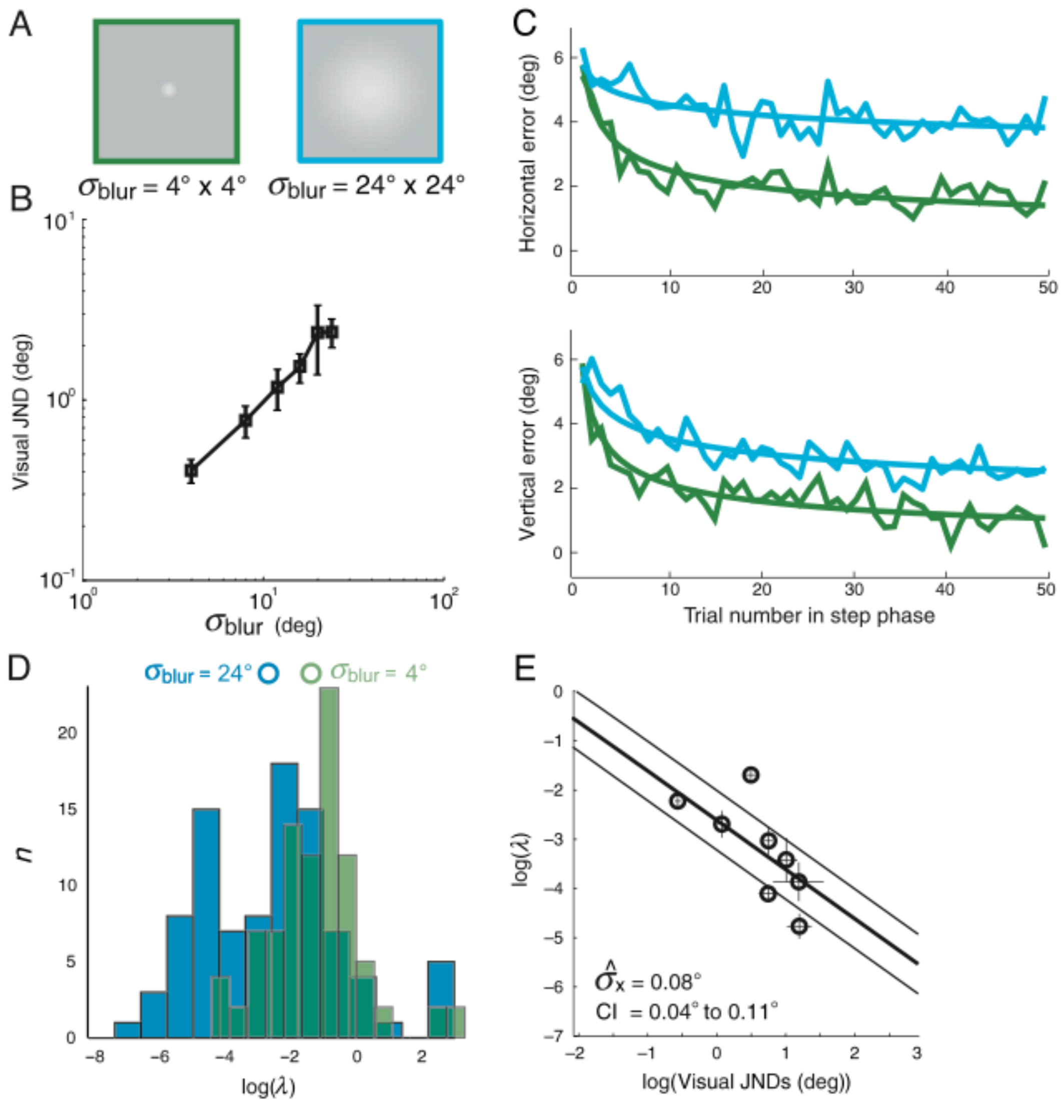
Multiple Time Scales



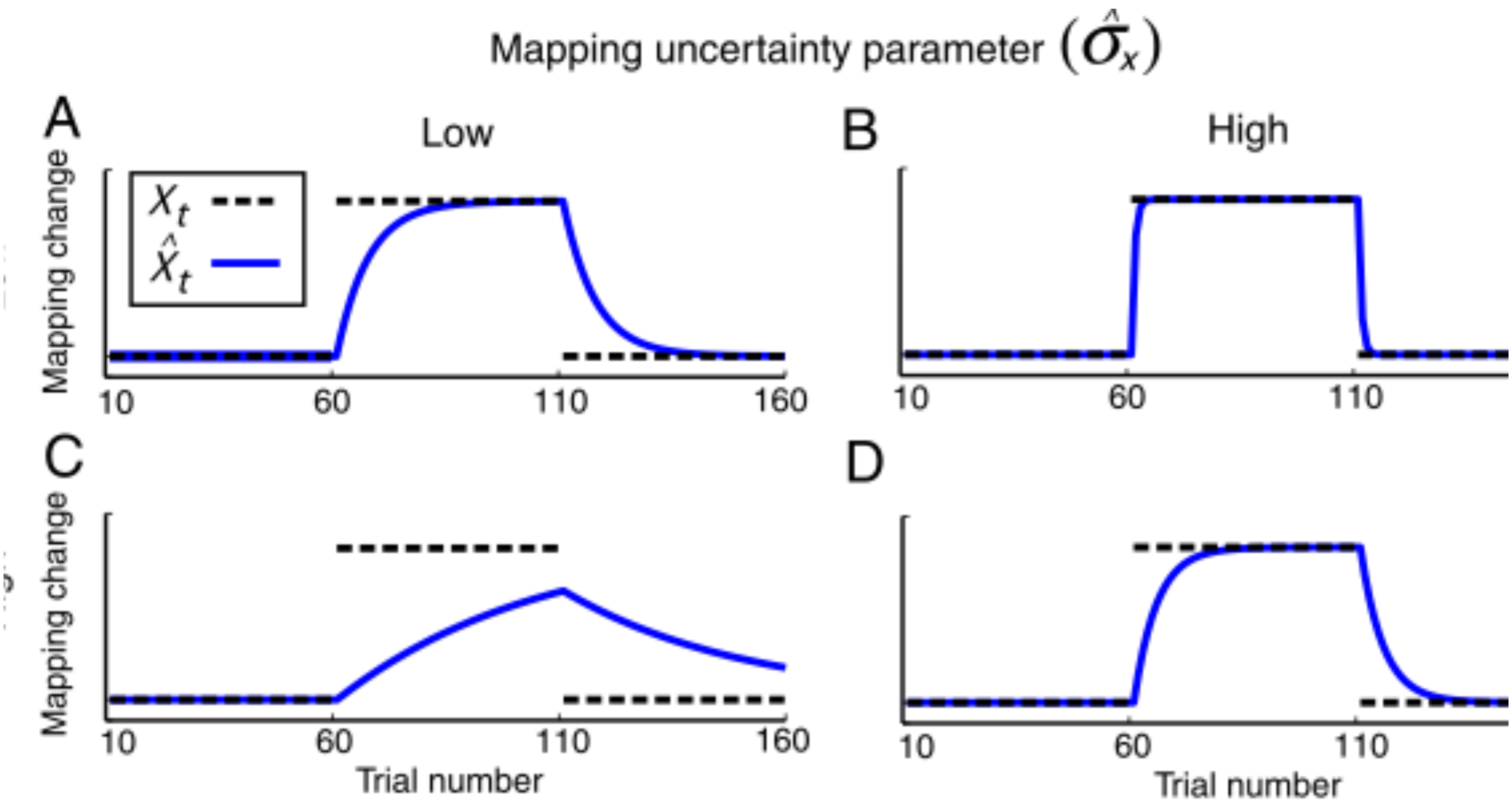
Temporal dependence in cue weighting



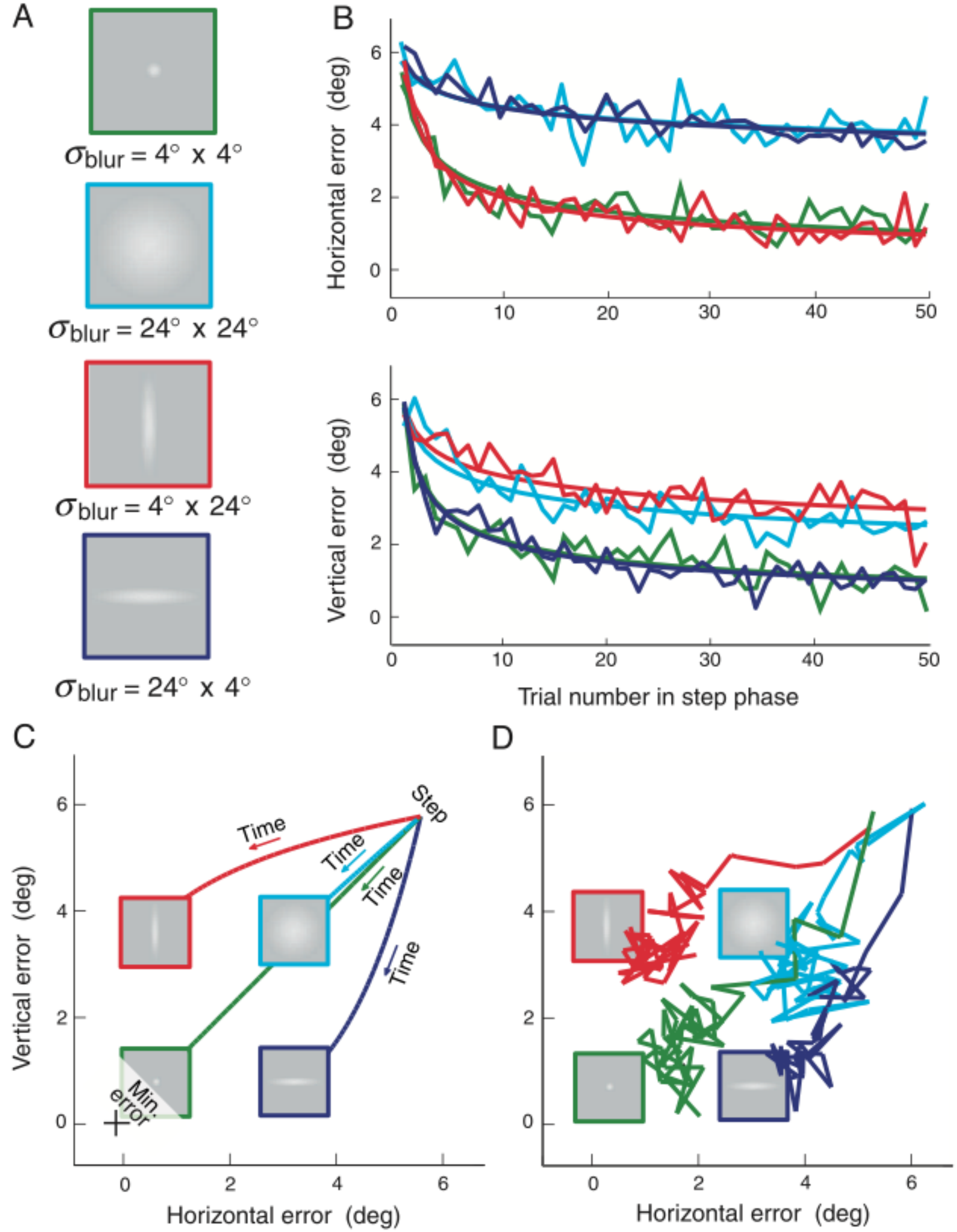
Position uncertainty and blur



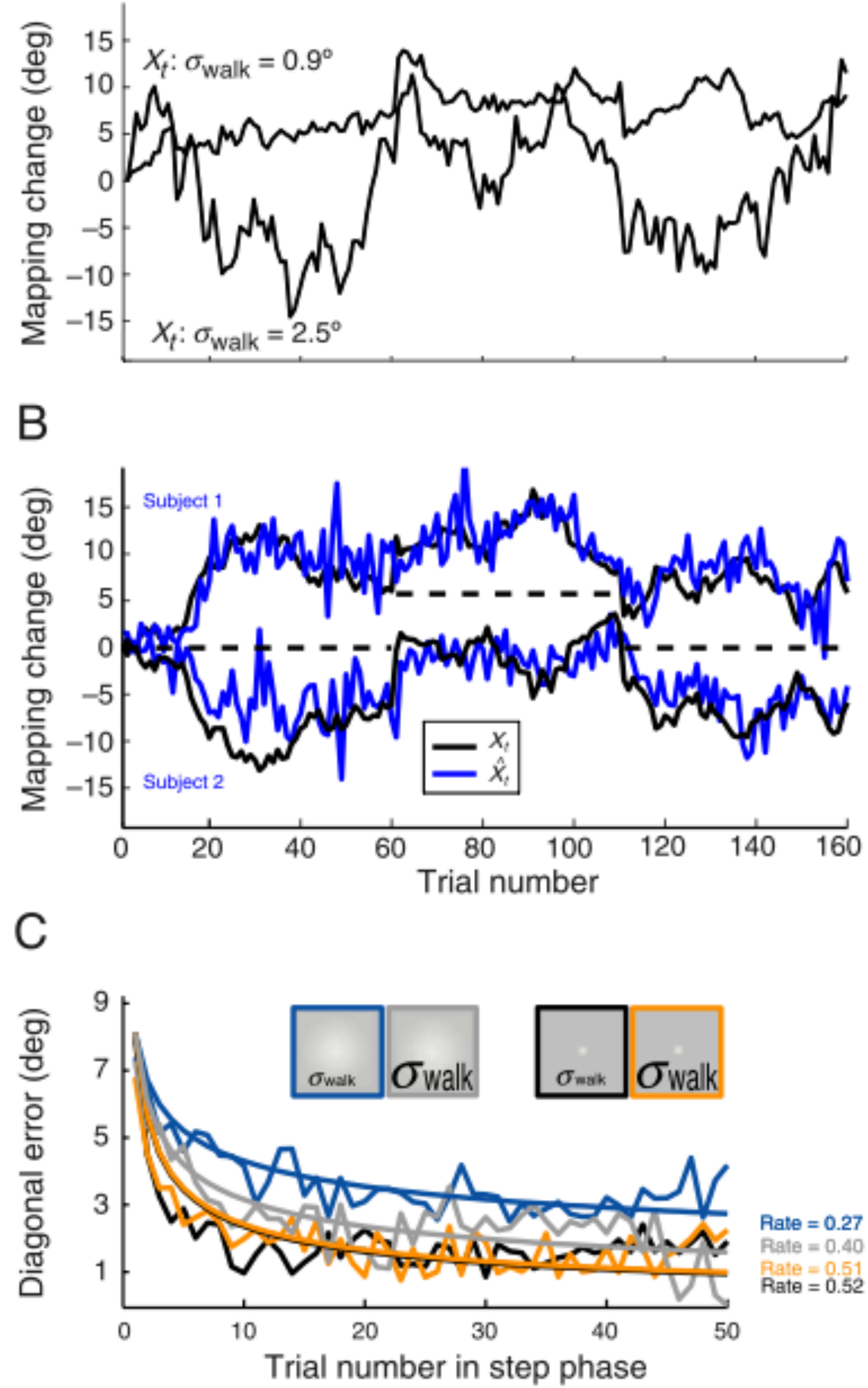
Predictions



Directional Blur

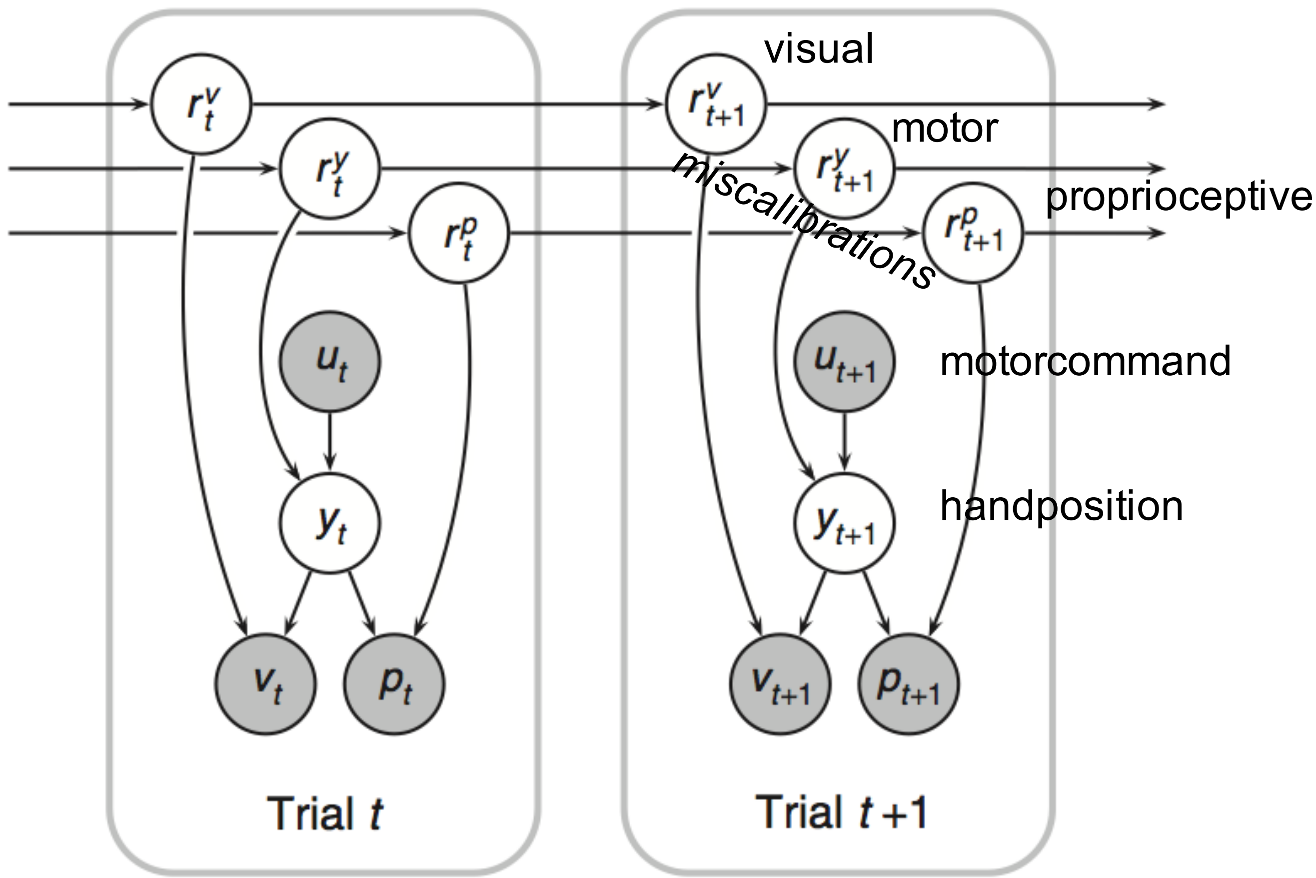


Random walk increases adaptation rate



Bayesian sensory- and motor-adaptation model.

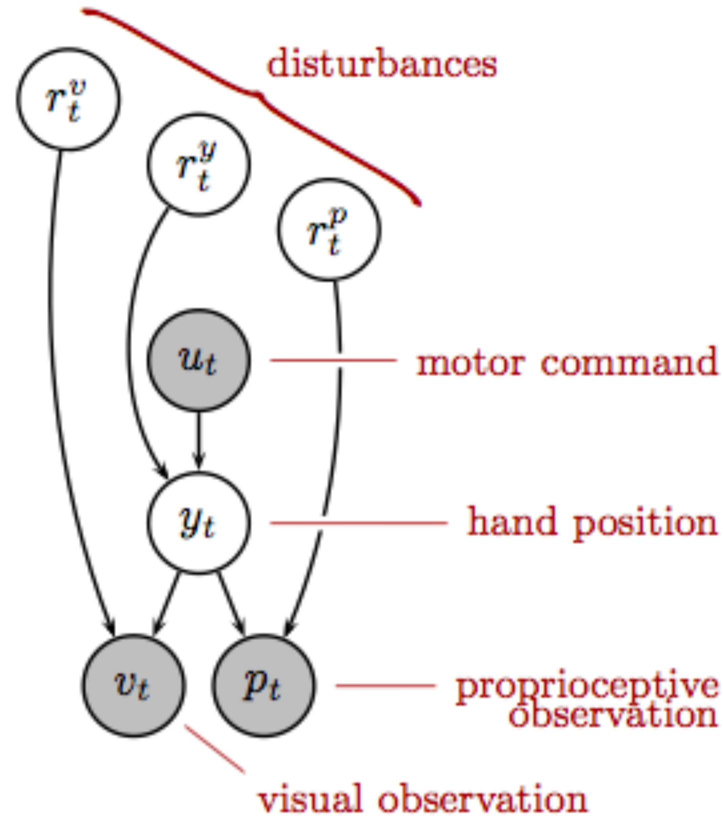
Shaded circles represent observed random variables
Unshaded circles represent unobserved random variables



Rewrite as Kalman

$$v_t = y_t + r_t^v + \varepsilon_t^v$$
$$p_t = y_t + r_t^p + \varepsilon_t^p$$
$$y_t = u_t + r_t^y + \varepsilon_t^y$$

Problem: This mixes observable and unobserved variables



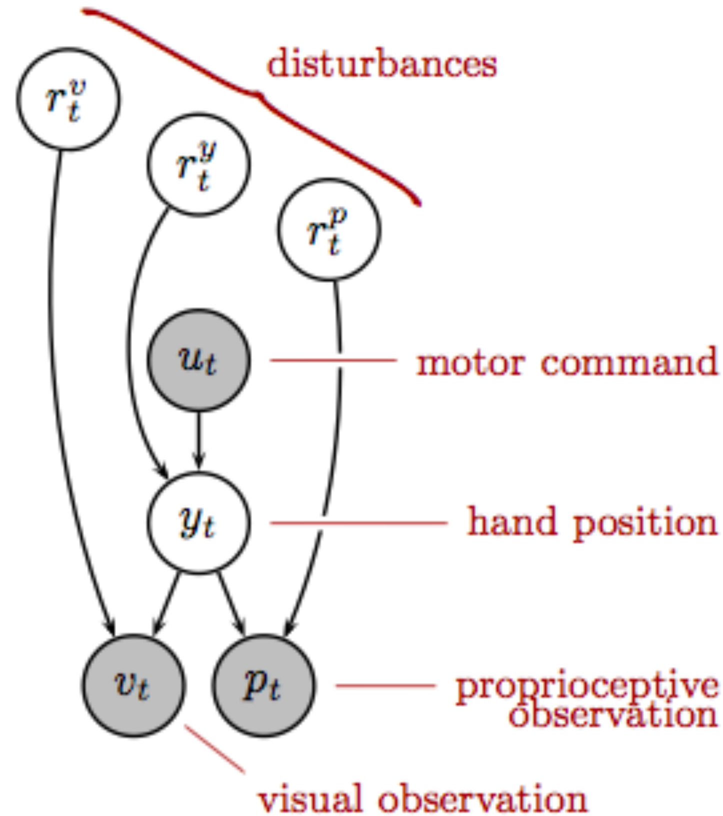
Rewrite as Kalman

Because Linear and Gaussian, we can rewrite:

$$v_t = y_t + r_t^v + \varepsilon_t^v$$

$$p_t = y_t + r_t^p + \varepsilon_t^p$$

$$u_t = y_t - r_t^y - \varepsilon_t^y$$



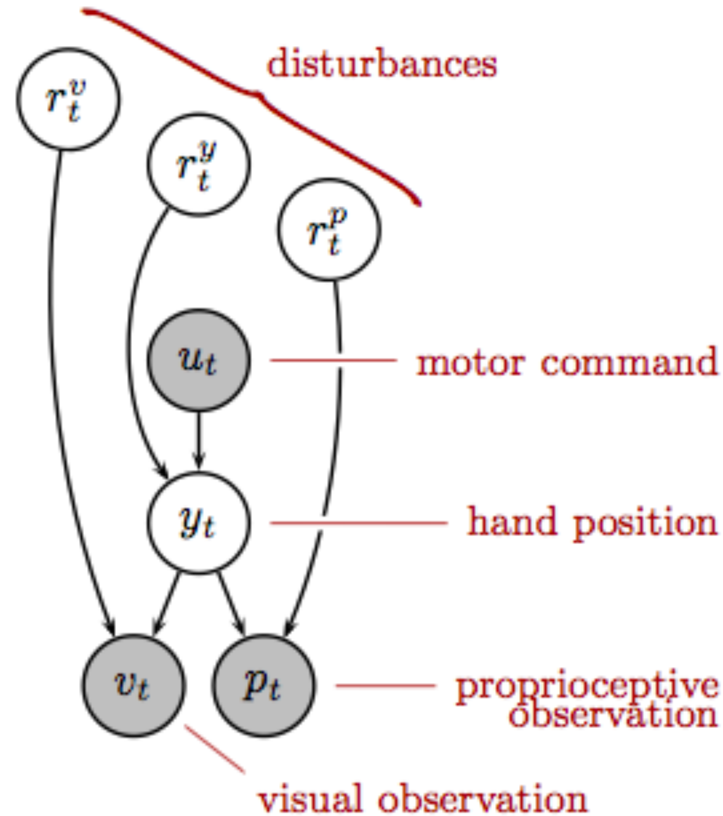
Rewrite as Kalman

$$v_t = y_t + r_t^v + \varepsilon_t^v$$

$$p_t = y_t + r_t^p + \varepsilon_t^p$$

$$u_t = y_t - r_t^y - \varepsilon_t^y$$

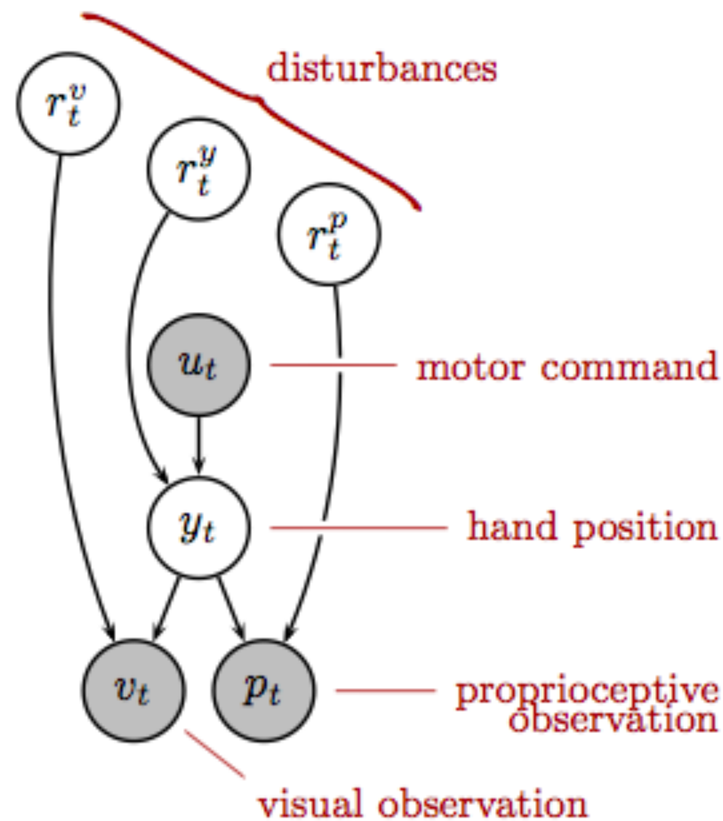
$$\begin{bmatrix} v_t \\ p_t \\ u_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} r_t^v \\ r_t^p \\ r_t^y \\ y_t \end{bmatrix} + \begin{bmatrix} \varepsilon_t^v \\ \varepsilon_t^p \\ -\varepsilon_t^y \end{bmatrix}$$



Rewrite as Kalman

$$\begin{bmatrix} v_t \\ p_t \\ u_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} r_t^v \\ r_t^p \\ r_t^y \\ y_t \end{bmatrix} + \begin{bmatrix} \varepsilon_t^v \\ \varepsilon_t^p \\ -\varepsilon_t^y \end{bmatrix}$$

THEY DIDN'T DO THIS, BUT COULD HAVE



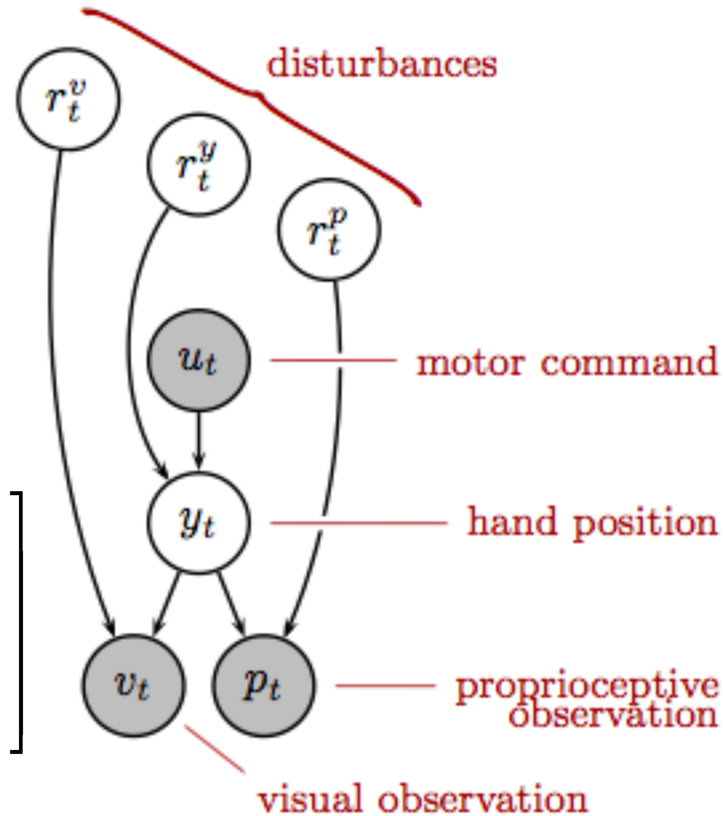
Rewrite as Kalman

$$y_t = u_t + r_t^y + \varepsilon_t^y$$

$$v_t = (u_t + r_t^y + \varepsilon_t^y) + r_t^v + \varepsilon_t^v$$

$$p_t = \left(u_t + r_t^y + \varepsilon_t^y\right) + r_t^p + \varepsilon_t^p$$

$$\begin{bmatrix} v_t - u_t \\ p_t - u_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_t^v \\ r_t^p \\ r_t^y \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_t^v \\ \varepsilon_t^p \\ \varepsilon_t^y \end{bmatrix}$$



$$\mathbf{z}_t = H\mathbf{r}_t + H\varepsilon_t$$

In Matrix Form

Dynamics Model

$$\mathbf{r}_{t+1} = A\mathbf{r}_t + \eta_t$$

$$\eta_t \sim N(0, Q)$$

$$A = \begin{bmatrix} a^v & 0 & 0 \\ 0 & a^p & 0 \\ 0 & 0 & a^y \end{bmatrix} \qquad Q = \begin{bmatrix} q^v & 0 & 0 \\ 0 & q^p & 0 \\ 0 & 0 & q^y \end{bmatrix}$$

Measurement Model

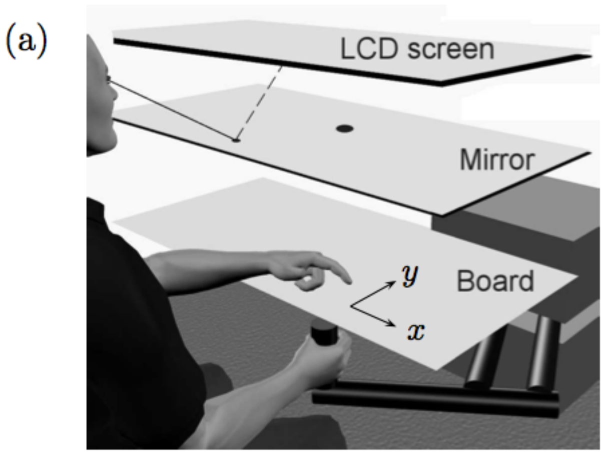
$$\mathbf{z}_t = H\mathbf{r}_t + H\varepsilon_t$$

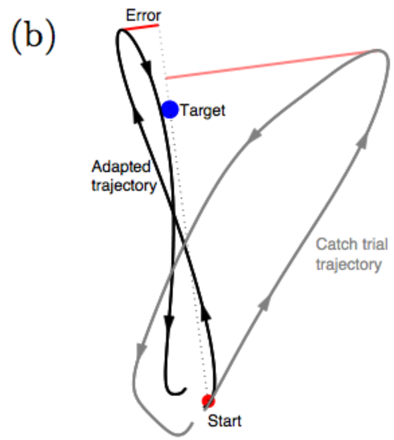
$$\eta_t \sim N(0, R)$$

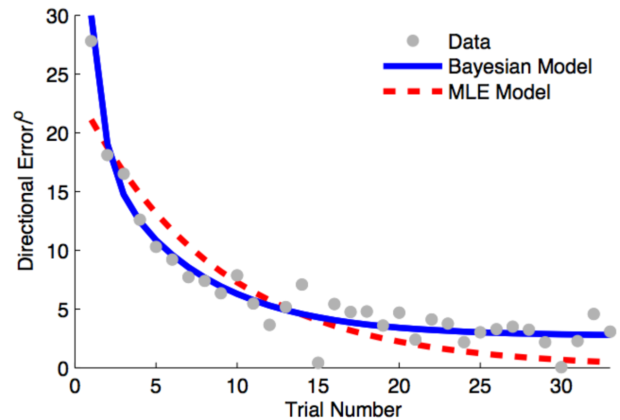
$$H = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

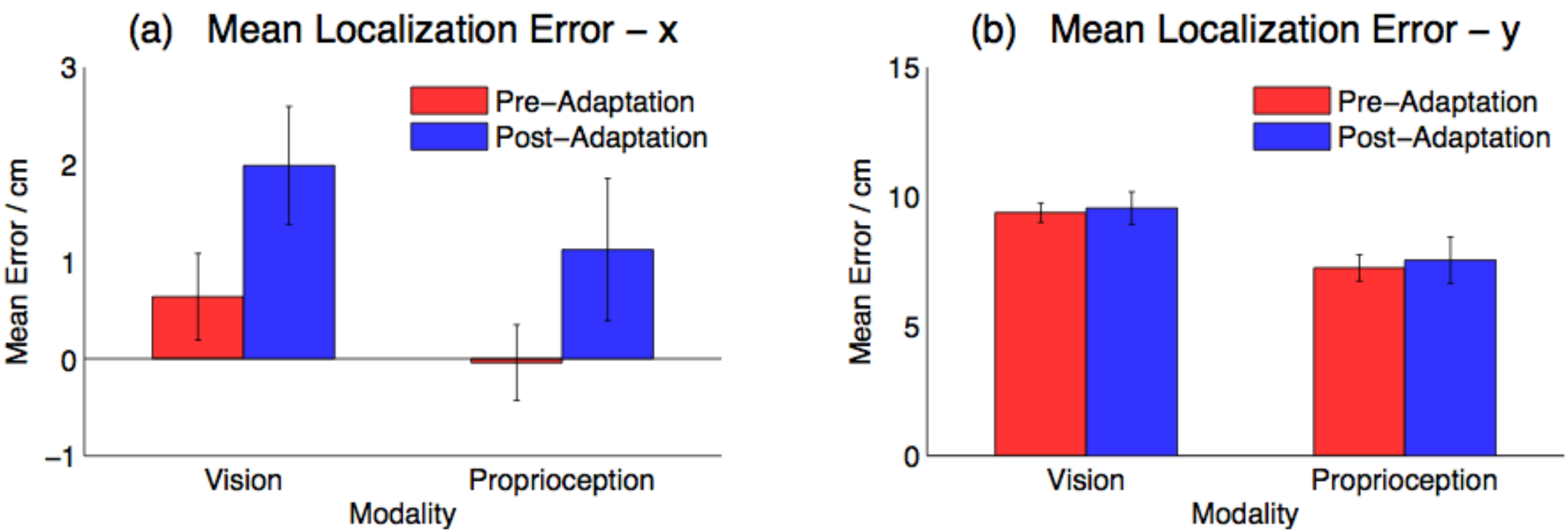
$$R = E\left[(H\epsilon_t)(H\epsilon_t)^T\right] = \begin{pmatrix} \sigma_v^2 + \sigma_u^2 & \sigma_u^2 \\ \sigma_u^2 & \sigma_p^2 + \sigma_u^2 \end{pmatrix}$$

Experimental results



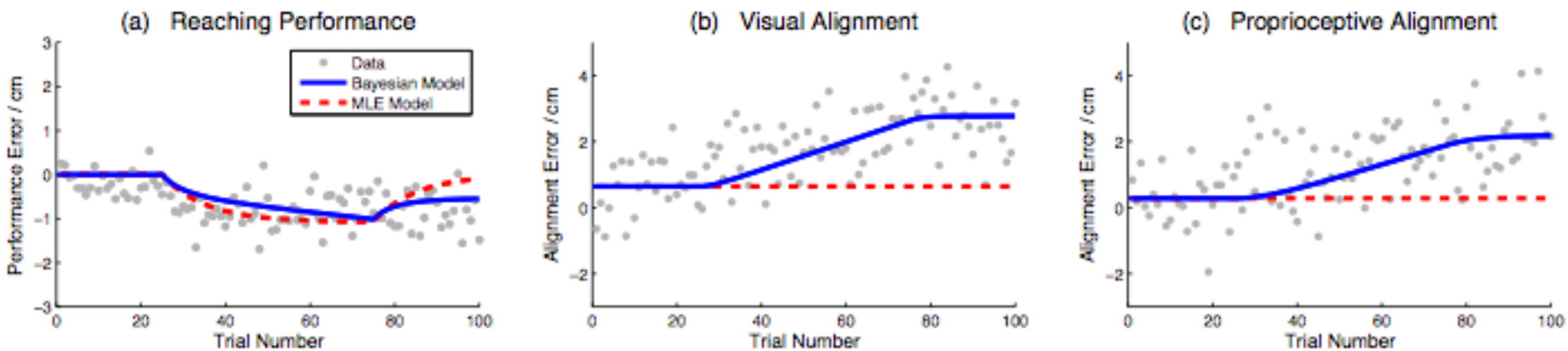






Results contd

Three tasks: Reach to target (right hand),
left hand to visual
left hand to right hand's location



Summing Up so far

- Bayesian models provide a principled language to describe uncertainty, information fusion under uncertainty, and make non-trivially verified predictions about perceptual processing.

- The brain needs to represent priors and likelihoods –
- Simple tools can model complex phenomena