Leigh R. Hochberg et al.

Reach and grasp by people with tetraplegia using a neurally controlled robotic arm

Nature, 17 May 2012

Paper overview

Ilya Kuzovkin

COMPUTATIONAL NEUROSCIENCE
Research group at the University of Tartu

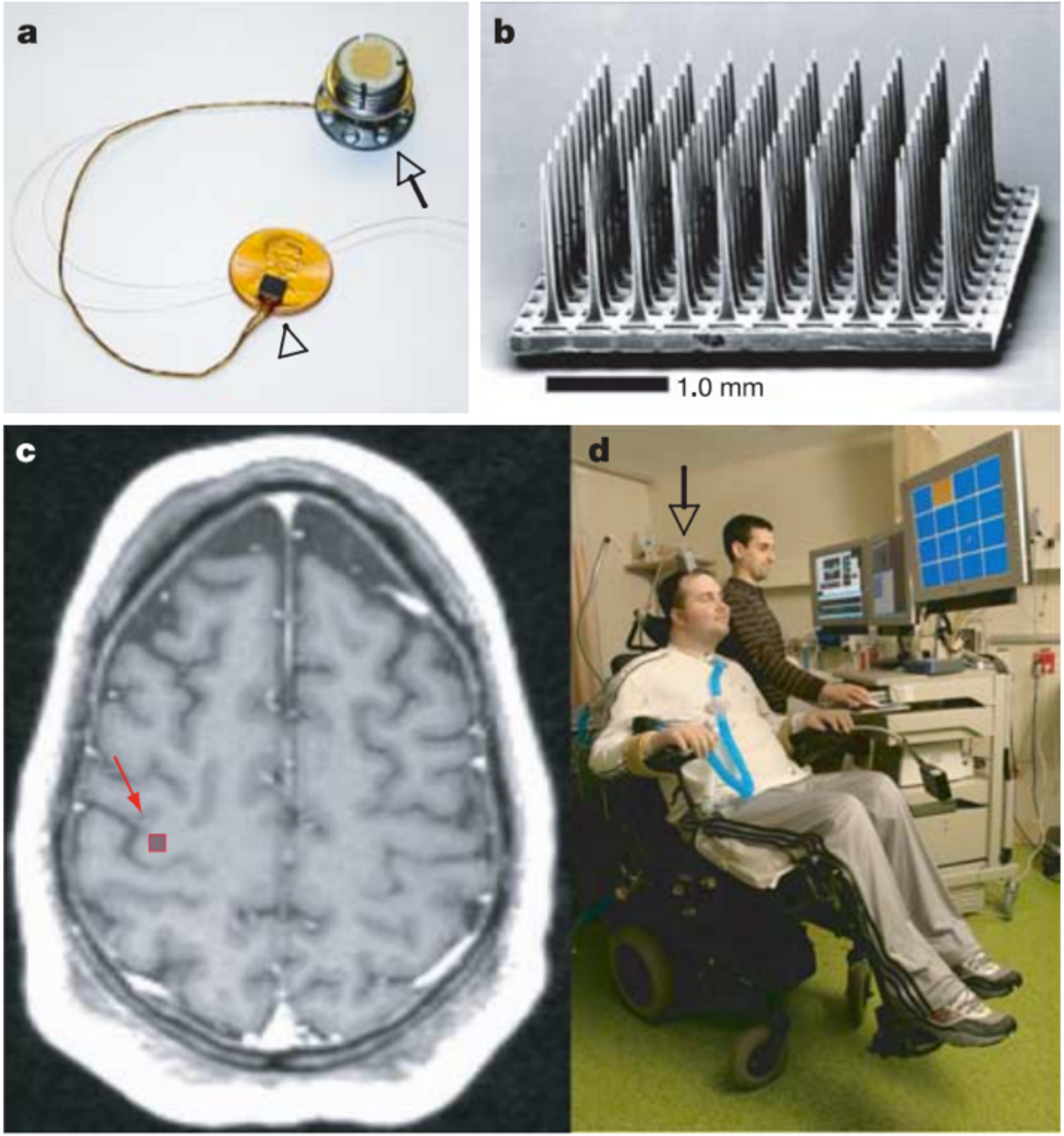
TARTU ÜLIKOOL · UNIVERSITAS TARTUENSIS 1632

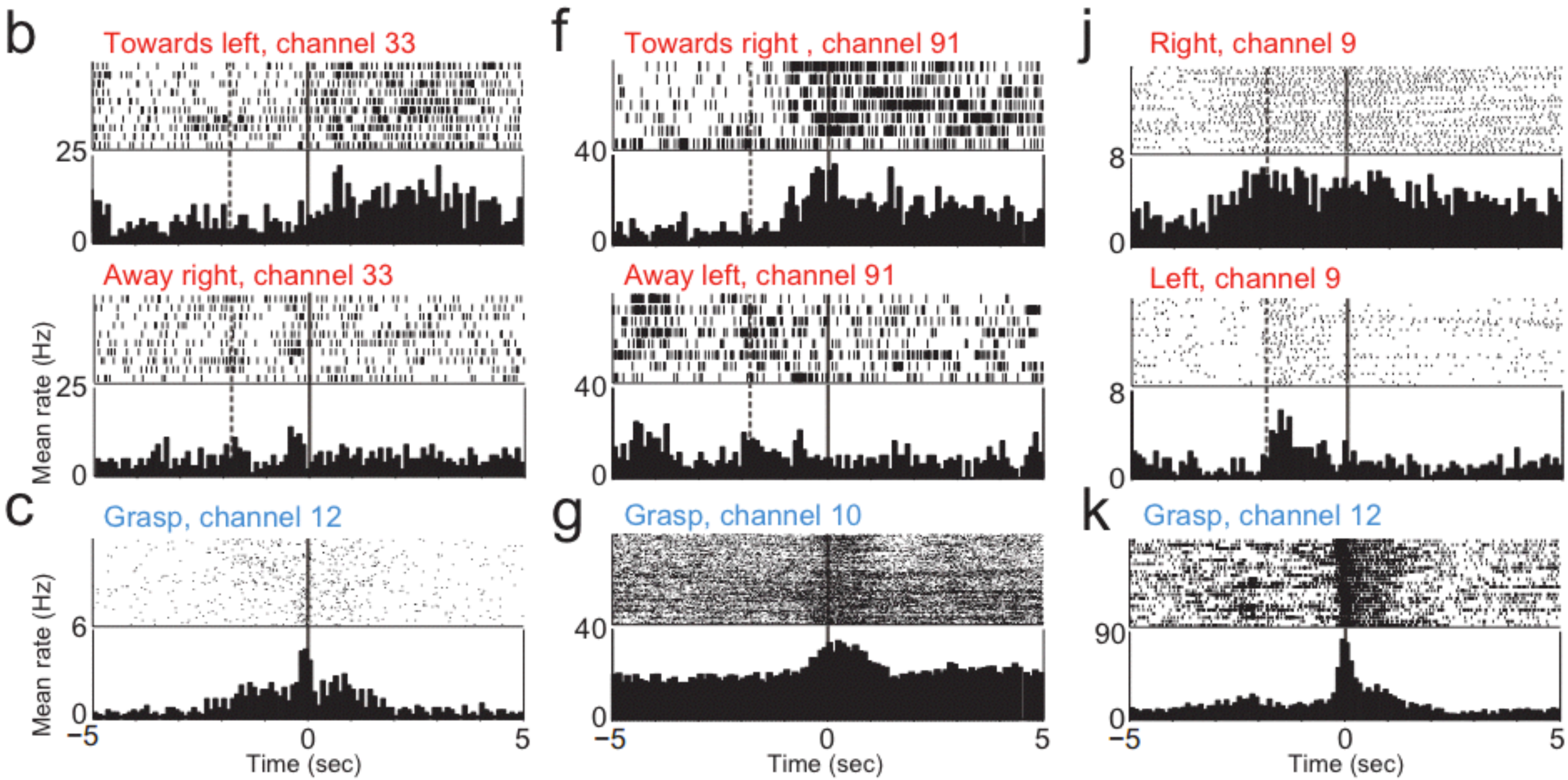
IT Akadeemia toetab Skype™

11 April 2014, Tartu

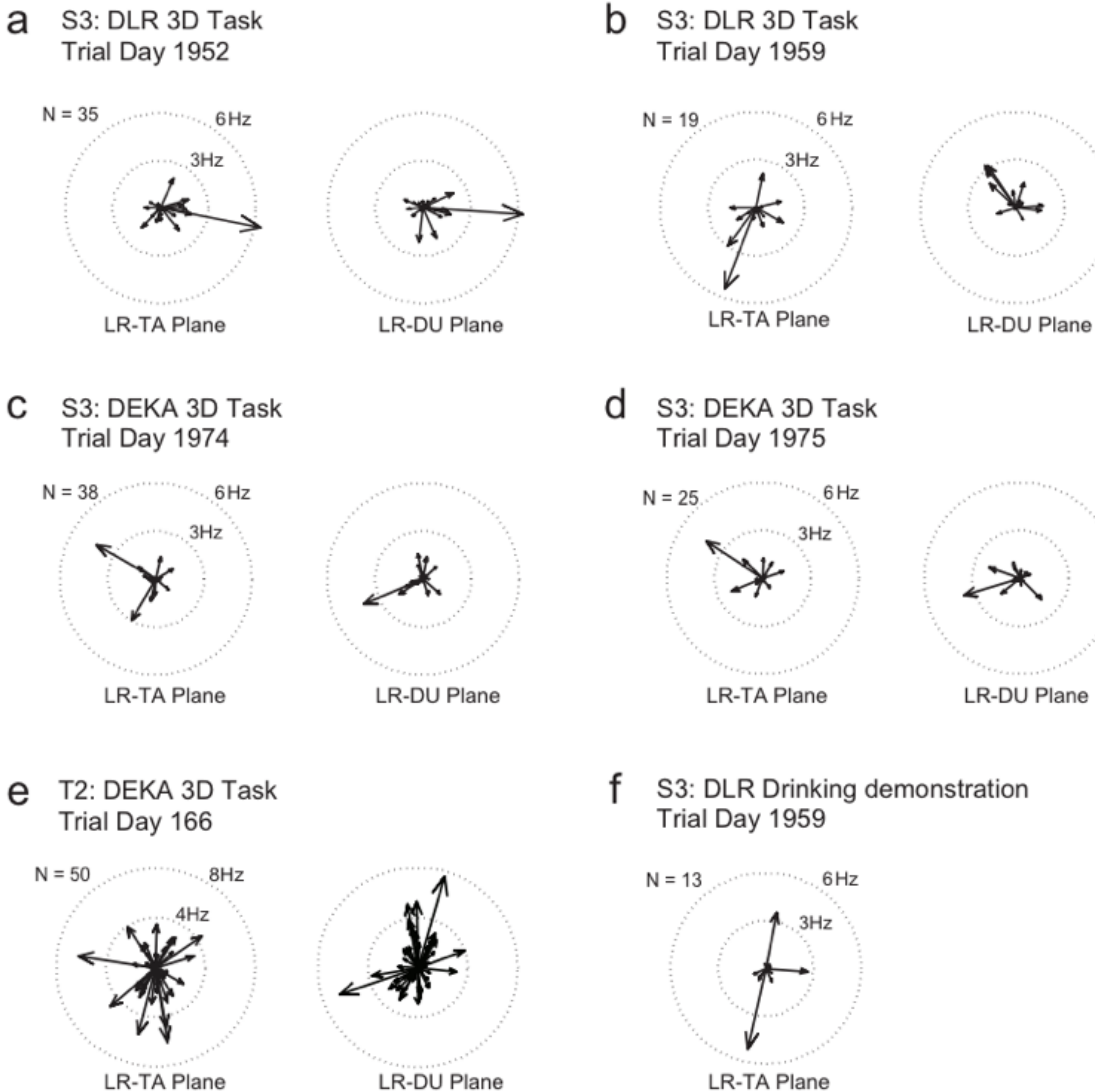
BrainGate Pilot Clinical Trial
Drinking From a Bottle Using a Robotic Arm
Participant S3
Trial Day 1959 / 12 April 2011
Hochberg *et al.*, 2012

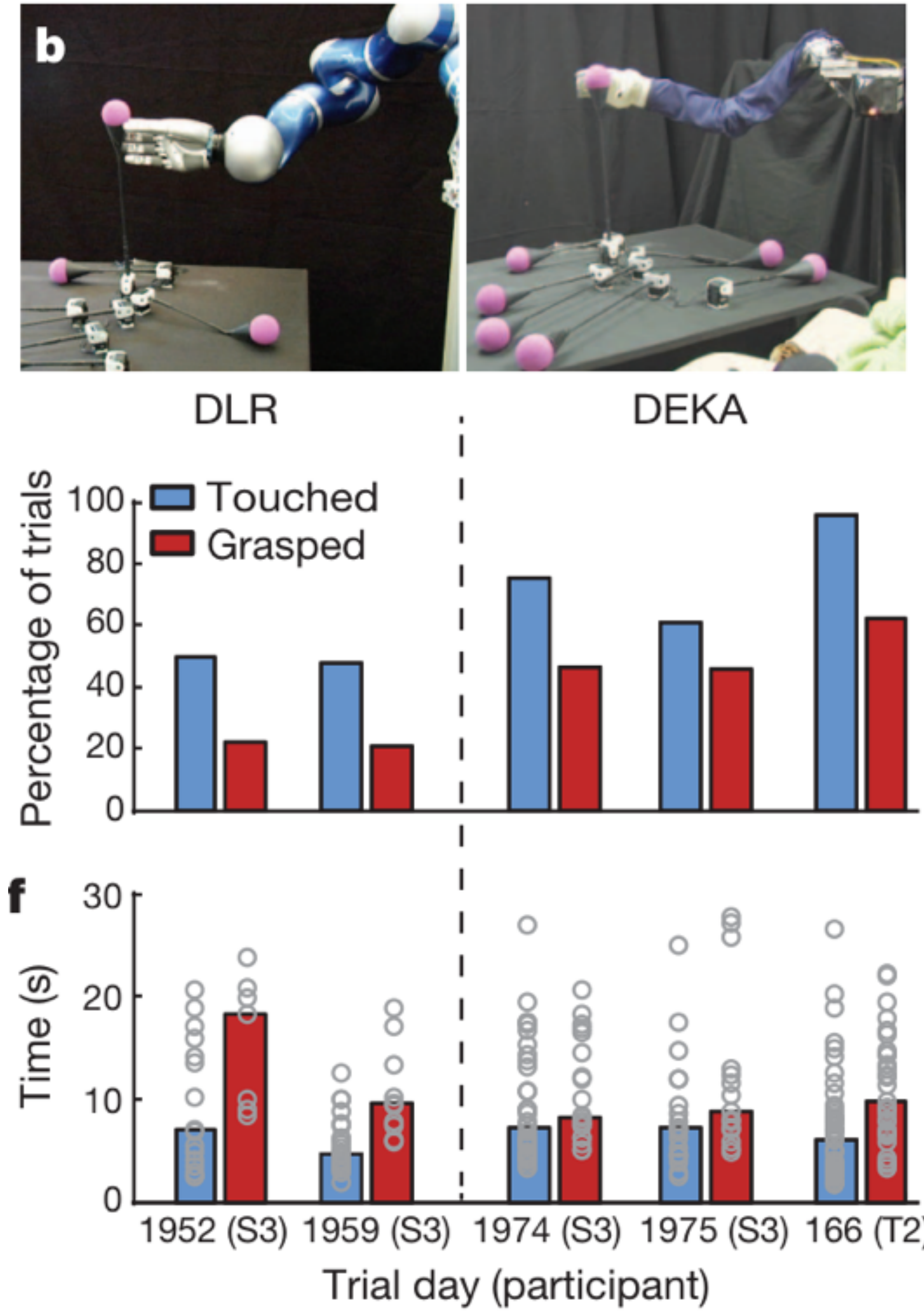
Caution: Investigational Device. Limited by Federal Law to Investigational Use.


a
b
1.0 mm
c
d



b
Towards left, channel 33
Away right, channel 33
c
Grasp, channel 12
f
Towards right , channel 91
Away left, channel 91
g
Grasp, channel 10
j
Right, channel 9
Left, channel 9
k
Grasp, channel 12
Mean rate (Hz)
Time (sec)
25
0
6
40
8
90
−5
5


a
S3: DLR 3D Task
Trial Day 1952
N = 35
6Hz
3Hz
LR-TA Plane
LR-DU Plane
b
S3: DLR 3D Task
Trial Day 1959
N = 19
6Hz
3Hz
LR-TA Plane
LR-DU Plane
c
S3: DEKA 3D Task
Trial Day 1974
N = 38
6Hz
3Hz
LR-TA Plane
LR-DU Plane
d
S3: DEKA 3D Task
Trial Day 1975
N = 25
6Hz
3Hz
LR-TA Plane
LR-DU Plane
e
T2: DEKA 3D Task
Trial Day 166
N = 50
8Hz
4Hz
LR-TA Plane
LR-DU Plane
f
S3: DLR Drinking demonstration
Trial Day 1959
N = 13
6Hz
3Hz
LR-TA Plane



b
DLR
DEKA
Touched
Grasped
Percentage of trials
100
80
60
40
20
0
f
Time (s)
30
20
10
0
1952 (S3)
1959 (S3)
1974 (S3)
1975 (S3)
166 (T2)
Trial day (participant)

adapted to the Kalman filter from refs 37 and 38

desired endpoint velocity was decoded using a Kalman filter[7,8,12,36]. The Kalman filter requires four sets of parameters, two of which were calculated based on the mean-subtracted (and for T2, smoothed with a 0.3 s exponential filter) threshold crossing rate, $\bar{z}$, and the intended direction, d, whereas the other two parameters were hard coded. The first parameter was the directional tuning, H, calculated as

etc…

adapted to the Kalman filter from refs 37 and 38

desired endpoint velocity was decoded using a Kalman filter[7,8,12,36]. The Kalman filter requ … the mean-sub … hold crossing r … eters were hard coded. The first parameter was the directional tuning, H, calculated as

How it works?

etc…

2012	Reach and grasp by people with tetraplegia using a neurally controlled robotic arm
2011	Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems
	Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array
	Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia
2010	Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia
2006	Neuronal ensemble control of prosthetic devices by a human with tetraplegia
	Bayesian Population Decoding of Motor Cortical Activity using a Kalman Filter

2012	Reach and grasp by people with tetraplegia using a neurally controlled robotic arm
2011	Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems
	Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array
	Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia
2010	Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia
2006	Neuronal ensemble control of prosthetic devices by a human with tetraplegia
	Bayesian Population Decoding of Motor Cortical Activity using a Kalman Filter

"… uses Bayesian inference techniques to estimate hand motion from the firing rates of multiple neurons."

"Decoding was performed using a Kalman filter which gives an efficient recursive method for Bayesian inference …"

*"… uses **Bayesian inference** techniques to estimate hand motion from the firing rates of multiple neurons."*

*"… uses **Bayesian inference** techniques to estimate hand motion from the firing rates of multiple neurons."*

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

*"… uses **Bayesian inference** techniques to estimate hand motion from the firing rates of multiple neurons."*

Posterior probability

Likelihood

Prior probability

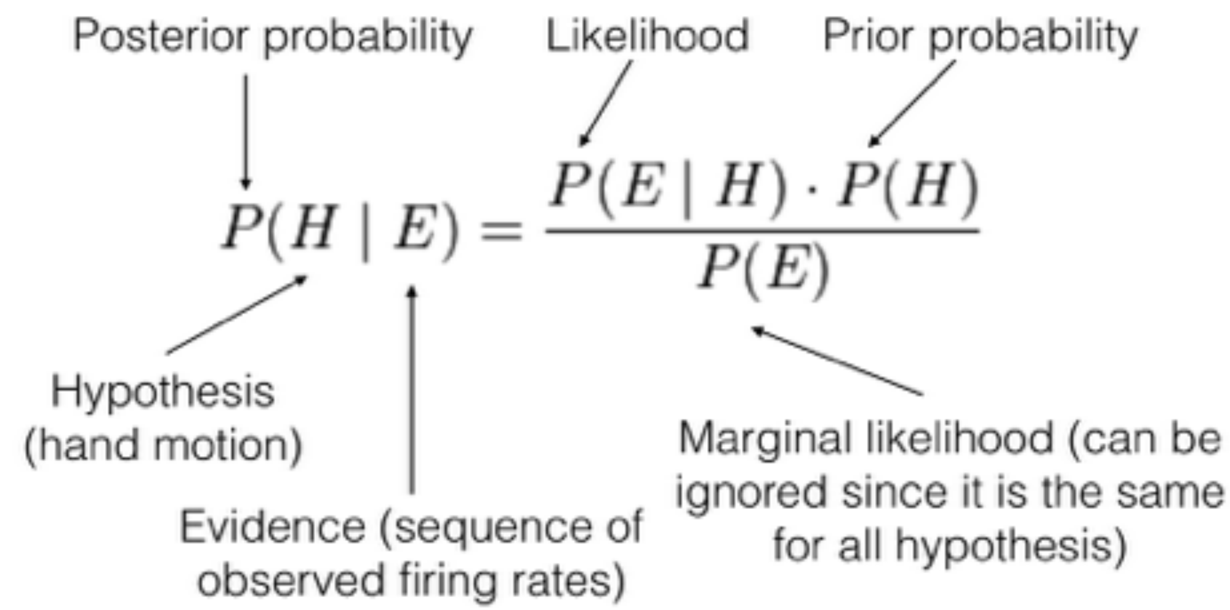
$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

Hypothesis (hand motion)

Evidence (sequence of observed firing rates)

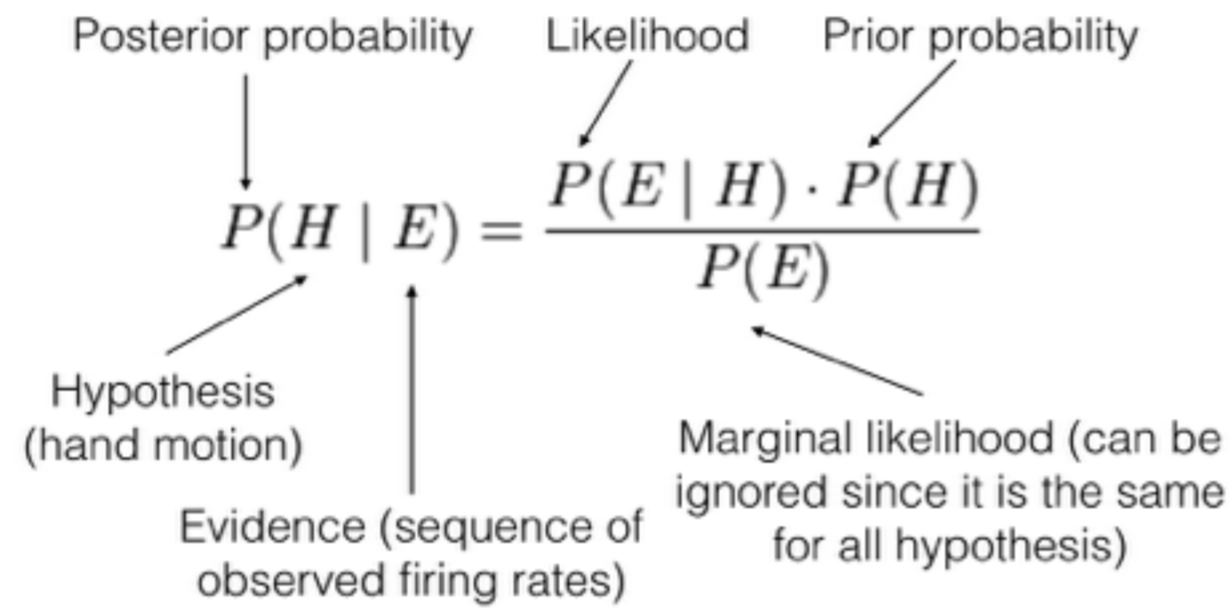
Marginal likelihood (can be ignored since it is the same for all hypothesis)

"… uses Bayesian inference techniques to ***estimate*** *hand motion from the firing rates of multiple neurons."*



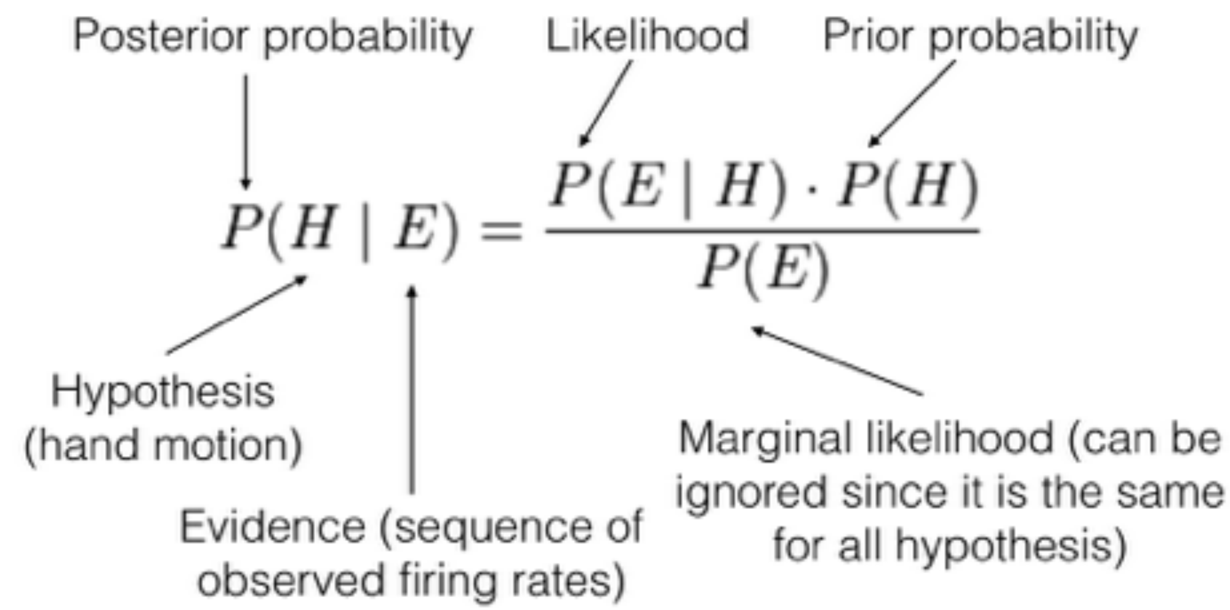
"… uses Bayesian inference techniques to ***estimate*** *hand motion from the firing rates of multiple neurons."*

*"**Likelihood** term models the probability of firing rates given a particular hand motion"*



*"… uses Bayesian inference techniques to **estimate** hand motion from the firing rates of multiple neurons."*

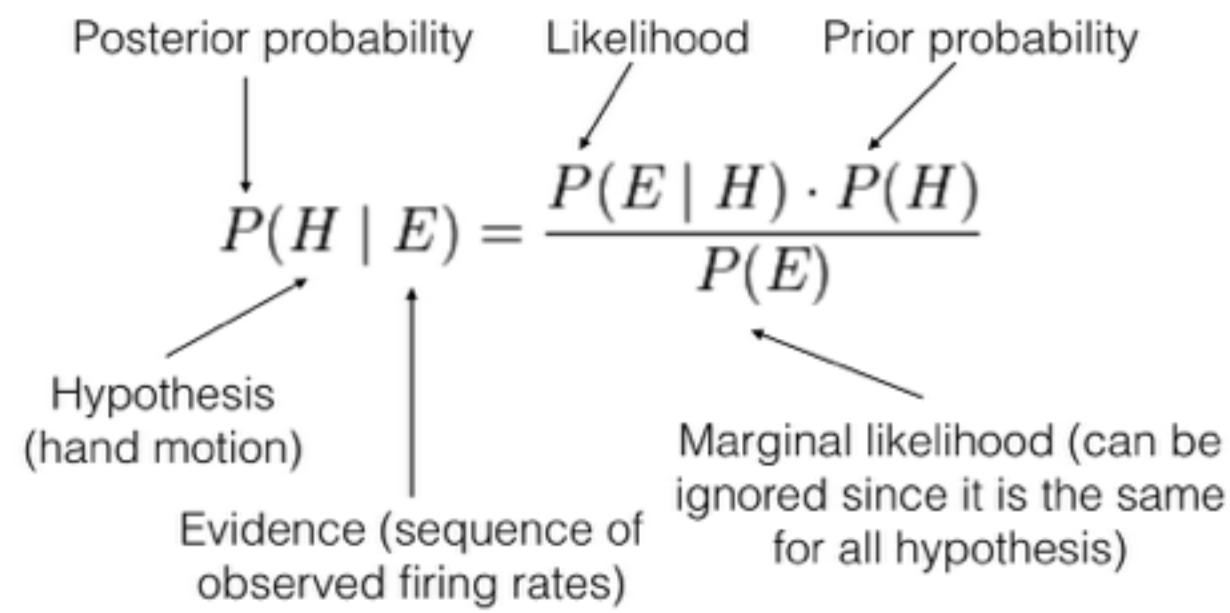
*"**Likelihood** term models the probability of firing rates given a particular hand motion"*



*"**linear Gaussian model** could be used to approximate this likelihood and could be readily **learned from** a small amount of **training data**"*

"… uses Bayesian inference techniques to ***estimate*** *hand motion from the firing rates of multiple neurons."*

*"****Likelihood*** *term models the probability of firing rates given a particular hand motion"*



*"****linear Gaussian model*** *could be used to approximate this likelihood and could be readily* ***learned from*** *a small amount of* ***training data****"*

"The ***prior*** *term defines a probabilistic* ***model of hand kinematics*** *and was also taken to be a linear Gaussian model."*

Neural Coding

Neural Coding of Hand Kinematics

Georgopoulos *et al*. (1982) observed that the firing rate of cells in MI were approximated by a cosine "tuning function." In particular the firing rate, z_k, of a cell at some time t_k is related to movement *direction*, θ_k, by

$$z_k = h_0 + h_p cos(\theta_k - \theta_p) \tag{1}$$

where θ_p is the cell's "preferred" direction (i.e. direction of maximal response) and h_0 and h_p are scalar constants. Equivalently, this can be expressed as

$$z_k = h_0 + h_1 sin(\theta_k) + h_2 cos(\theta_k), \tag{2}$$

where h_1, h_2 are scalar parameters that can be fit to training data.

Neural Coding of Hand Kinematics

Moran and Schwartz (Moran and Schwartz, 1999b) extended the above model to include the speed, $||v||$, of the hand

$$z_k = h_0 + ||v||(h_x sin(\theta_k) + h_y cos(\theta_k)). \quad (3)$$

This is equivalent to modeling the firing rate as a linear function of velocity in the x and y coordinates:

$$z_k = h_0 + h_x v_{x,k} + h_y v_{y,k}, \quad (4)$$

where $v_{x,k}$ and $v_{y,k}$ represent the hand velocity in the x and y directions respectively at time t_k.

Neural Coding of Hand Kinematics

A similar linear model was proposed to relate firing rate and hand position (Kettner et al., 1988):

$$z_k = f_0 + f_x x_k + f_y y_k, \qquad (5)$$

where x_k and y_k represent hand position and f_0, f_x, f_y are the linear coefficients which are fit to training data.

Neural Coding of Hand Kinematics

A similar linear model was proposed to relate firing rate and hand position (Kettner et al., 1988):

$$z_k = f_0 + f_x x_k + f_y y_k, \tag{5}$$

where x_k and y_k represent hand position and f_0, f_x, f_y are the linear coefficients which are fit to training data.

Experiment 1: 23/25 neurons are correctly described by equations (4) and (5)

Experiment 2: 39/42 neurons correctly described by (4) and (5)

Neural Coding of Hand Kinematics

A similar linear model was proposed to relate firing rate and hand position (Kettner et al., 1988):

$$z_k = f_0 + f_x x_k + f_y y_k, \quad (5)$$

where x_k ar … ts which

are fit to tra …

The **relationship** between the **kinematics** of the **arm** and the behavior of the **neurons** is **strong**

described by equations (4) and (5)

Experiment 2: 39/42 neurons correctly described by (4) and (5)

Learning the model

Detour: Multivariate normal distribution

The multivariate normal distribution of a k-dimensional random vector $\mathbf{x} = [X_1, X_2, \ldots, X_k]$

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma),$$

or to make it explicitly known that X is k-dimensional,

$$\mathbf{x} \sim \mathcal{N}_k(\boldsymbol{\mu}, \Sigma).$$

with k-dimensional mean vector

$$\boldsymbol{\mu} = [\mathrm{E}[X_1], \mathrm{E}[X_2], \ldots, \mathrm{E}[X_k]]$$

and k x k covariance matrix

$$\Sigma = [\mathrm{Cov}[X_i, X_j]], i = 1, 2, \ldots, k; j = 1, 2, \ldots, k$$

Every linear combination of its components $Y = a_1X_1 + \ldots + a_kX_k$ is normally distributed. That is, for any constant vector $\mathbf{a} \in \mathbf{R}^k$, the random variable $Y = \mathbf{a'x}$ has a univariate normal distribution.

Detour: Multivariate normal distribution

The multivariate normal distribution of a *k*-dimensional random vector $\mathbf{x} = [X_1, X_2, \ldots, X_k]$

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu},$$

or to make it explicit

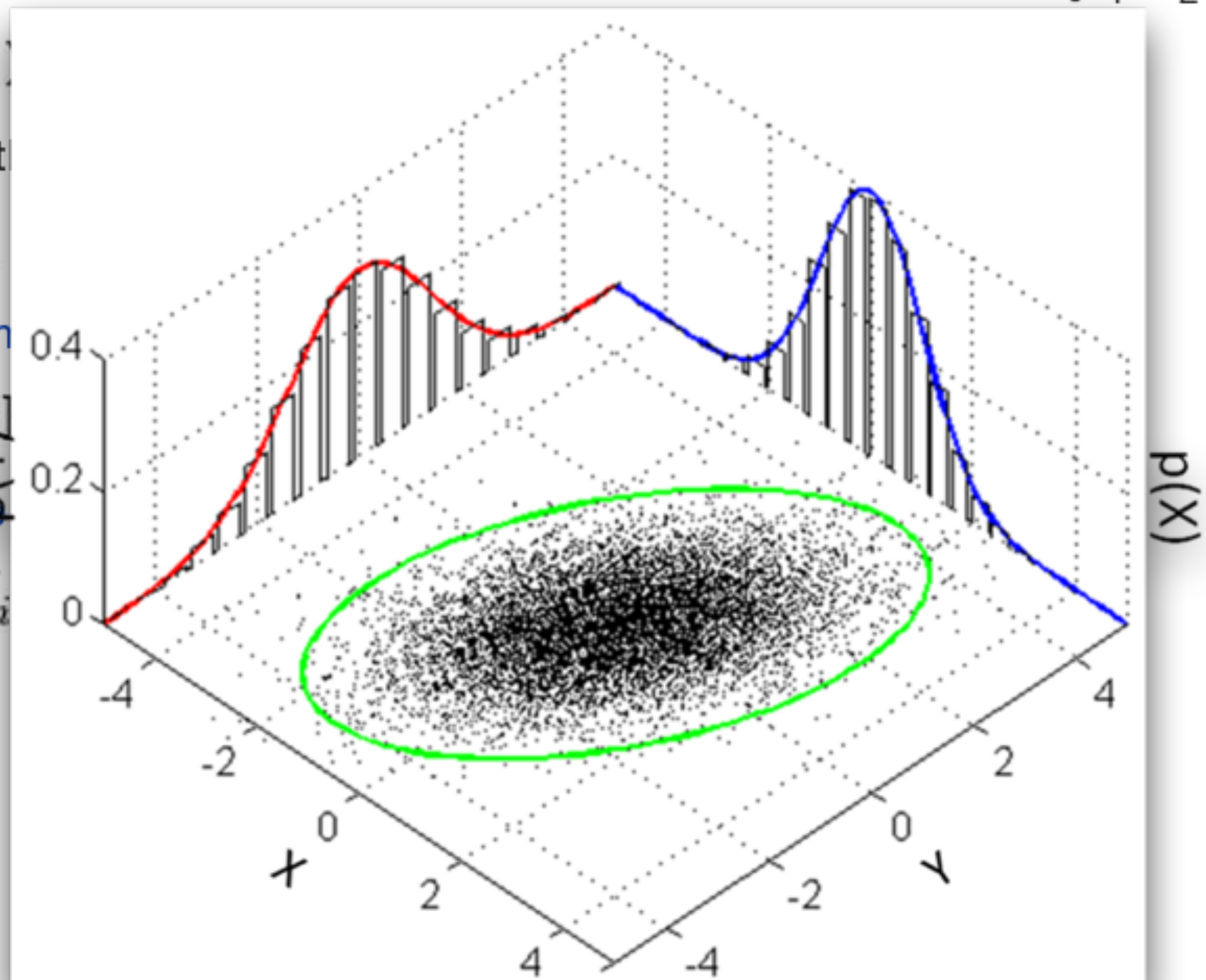
$$\mathbf{x} \sim \mathcal{N}_k(\boldsymbol{\mu},$$

with *k*-dimensional n

$$\boldsymbol{\mu} = [\mathrm{E}[X_1],$$

and *k x k* covariance

$$\boldsymbol{\Sigma} = [\mathrm{Cov}[X_i$$



Every linear combination of its components $Y = a_1X_1 + \ldots + a_kX_k$ is normally distributed. That is, for any constant vector $\mathbf{a} \in \mathbf{R}^k$, the random variable $Y = \mathbf{a'x}$ has a univariate normal distribution.

Detour: Multivariate normal distribution

The multivariate normal distribution of a k-dimensional random vector $\mathbf{x} = [X_1, X_2, \ldots, X_k]$

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu},$$

or to make it explicit

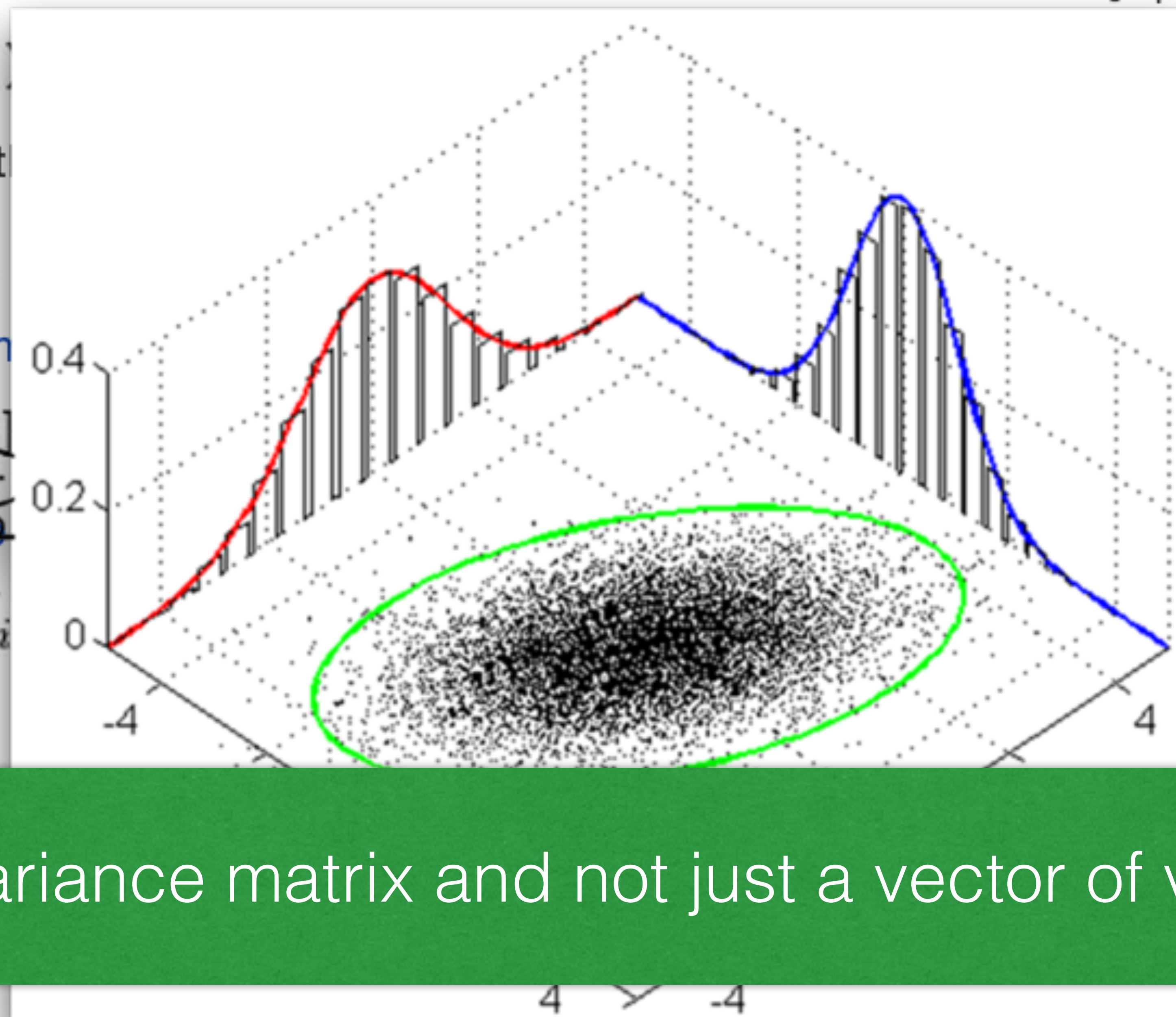
$$\mathbf{x} \sim \mathcal{N}_k(\boldsymbol{\mu},$$

with k-dimensional m

$$\boldsymbol{\mu} = [\mathrm{E}[X_1],$$

and k x k covariance

$$\boldsymbol{\Sigma} = [\mathrm{Cov}[X_i$$



Why covariance matrix and not just a vector of variances?

Every linear combination of its components $Y = a_1X_1 + \ldots + a_kX_k$ is normally distributed. That is, for any constant vector $\mathbf{a} \in \mathbf{R}^k$, the random variable $Y = \mathbf{a'x}$ has a univariate normal distribution.

Definitions

We then define the system *state* to be a six-dimensional vector $\mathbf{x}_k = [x, y, v_x, v_y, a_x, a_y]_k^T$ representing the x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration of the hand at time $t_k = k\Delta t$ where $\Delta t = 70ms$

Definitions

We then define the system *state* to be a six-dimensional vector $\mathbf{x}_k = [x, y, v_x, v_y, a_x, a_y]_k^T$ representing the x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration of the hand at time $t_k = k\Delta t$ where $\Delta t = 70ms$

let observations $\mathbf{z}_k \in \mathbb{R}^C$ represent a $C \times 1$ vector containing the firing rates at time t_k for C observed neurons within a Δt time interval. Let $\mathbf{Z}_k = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k]^T$ represent the history of measurements up to time bin k.

Parameters of the model

Parameters of the model

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k,$$

$$\mathbf{q}_k \sim N(0, \mathbf{Q}_k), \mathbf{Q}_k \in \mathbb{R}^{C \times C}$$

H is the relation between the firing rates of each of the neurons and states of the arm

Q is covariance matrix of the noise

Parameters of the model

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k,$$
$$\mathbf{q}_k \sim N(0, \mathbf{Q}_k), \mathbf{Q}_k \in \mathbb{R}^{C \times C}$$

H is the relation between the firing rates of each of the neurons and states of the arm
Q is covariance matrix of the noise

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k,$$
$$\mathbf{w}_k \sim N(0, \mathbf{W}_k), \mathbf{W}_k \in \mathbb{R}^{6 \times 6}$$

A is the relation between the state at time k+1 and the state at time k
W is covariance matrix of the noise

Parameters of the model

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k,$$
$$\mathbf{q}_k \sim N(0, \mathbf{Q}_k), \mathbf{Q}_k \in \mathbb{R}^{C \times C}$$

H is the relation between the firing rates of each of the neurons and states of the arm

Q is covariance matrix of the noise

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k,$$
$$\mathbf{w}_k \sim N(0, \mathbf{W}_k), \mathbf{W}_k \in \mathbb{R}^{6 \times 6}$$

A is the relation between the state at time k+1 and the state at time k

W is covariance matrix of the noise

Matrices **A**, **H, Q, W** is what we want to learn from the training data

The Learning

we can estimate all the constant parameters $\mathbf{A}, \mathbf{H}, \mathbf{W}, \mathbf{Q}$ from training data by maximizing the joint probability $p(\mathbf{X}_M, \mathbf{Z}_M)$, where both the hand kinematics $\mathbf{X}_M = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]^T$ and the firing rates $\mathbf{Z}_M$ are known for the M time instants in the training set.

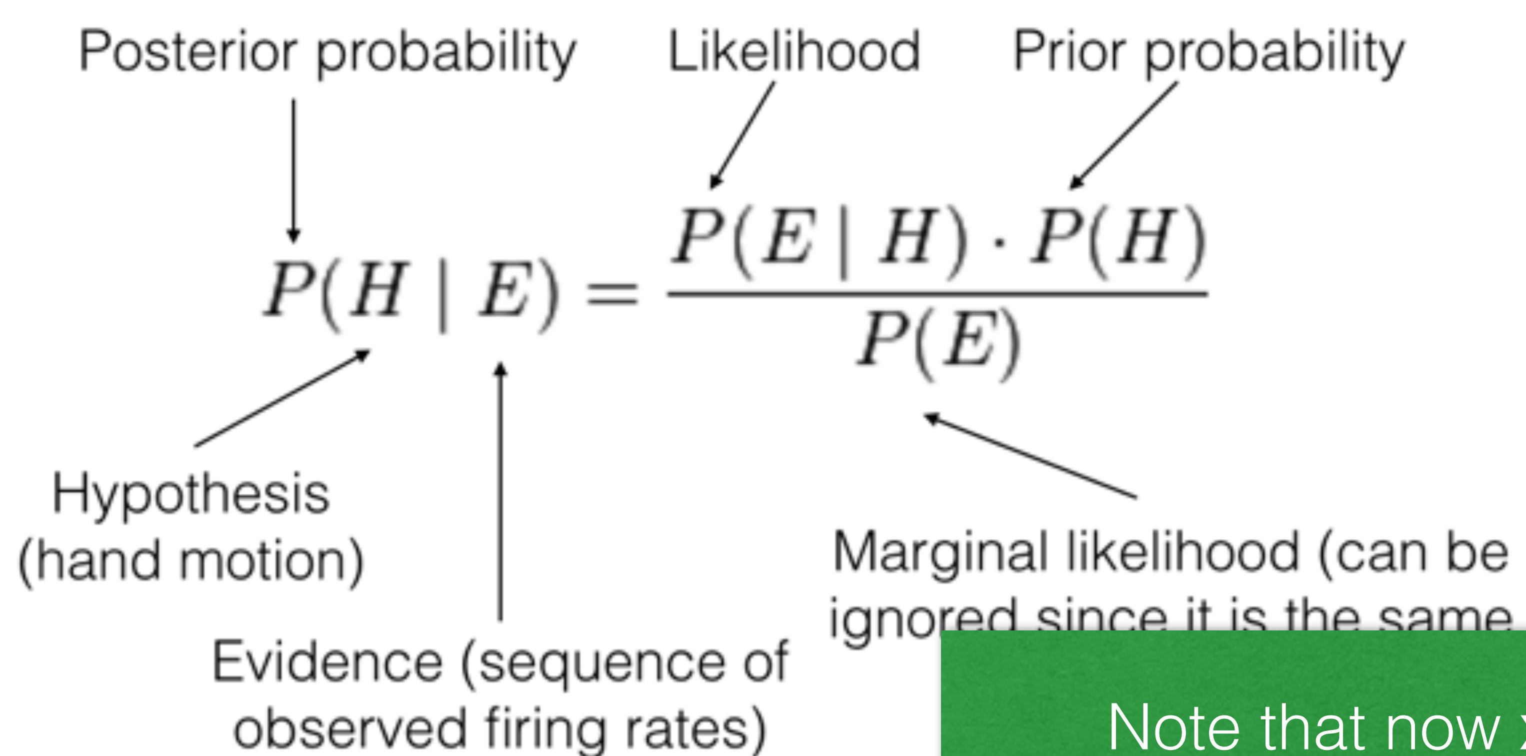
$$p(\mathbf{X}_M, \mathbf{Z}_M) = [p(\mathbf{x}_1) \prod_{k=2}^{M} p(\mathbf{x}_k | \mathbf{x}_{k-1})][\prod_{k=1}^{M} p(\mathbf{z}_k | \mathbf{x}_k)].$$

This is simply the product of the prior hand state $p(\mathbf{x}_1)$ at the first time instant, the probability of each new state conditioned on the previous one, and the likelihood at each time instant. Given training data we maximize this probability with respect to $\mathbf{A}, \mathbf{H}, \mathbf{W}, \mathbf{Q}$

Decoding

“Decoding was performed using a Kalman filter which gives an efficient recursive method for **Bayesian inference** *…”*

We pose the problem of estimating the system state as one of Bayesian inference. Let $p(\mathbf{x}_k|\mathbf{Z}_k)$ be the *a posteriori* probability of the system state conditioned on the measurements.



Note that now **x** and **z** and everything else refer to the test data

"Decoding was performed using a Kalman filter which gives an efficient recursive method for Bayesian inference *…"*

Then, using Bayes Theorem and simple algebra we can write the posterior probability

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa\, p(\mathbf{z}_k|\mathbf{x}_k) \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}, \quad (8)$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is called the *likelihood* of the state, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is a *temporal prior* that models how the state evolves from one time instant to the next, and $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ is simply the posterior probability at the previous time instant. The term κ is a normalizing term, independent of $\mathbf{x}_k$, which ensures that posterior integrates to 1.

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

Posterior probability

Likelihood

Prior probability

Hypothesis (hand motion)

Evidence (sequence of observed firing rates)

Marginal likelihood (can be ignored since it is the same for all hypothesis)

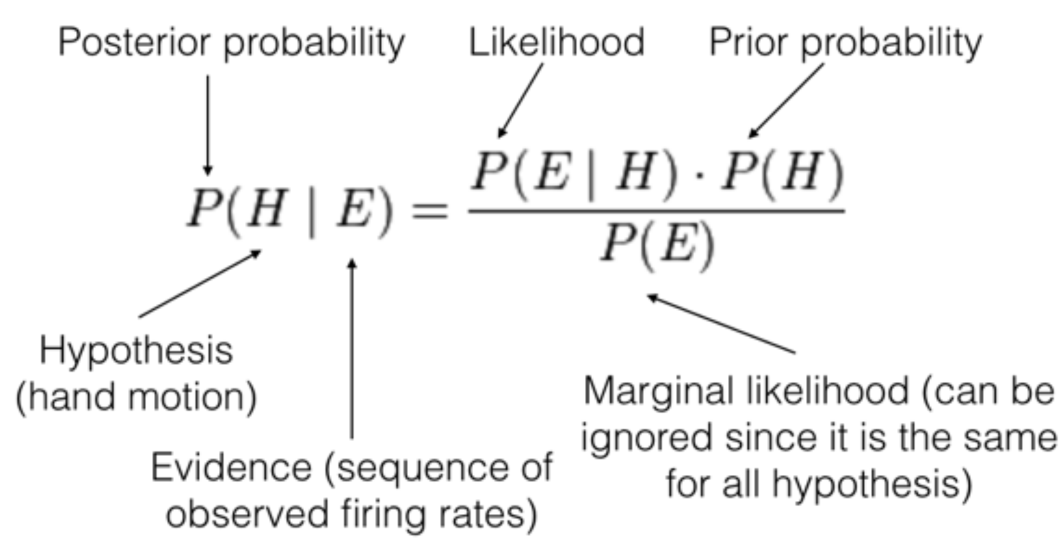
“Decoding was performed using a Kalman filter which gives an efficient recursive method for ***Bayesian inference*** *…”*

Then, using Bayes Theorem and simple algebra we can write the posterior probability

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa\, p(\mathbf{z}_k|\mathbf{x}_k) \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}, \quad (8)$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is called the *likeliho* … of the state, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is a *temporal prior* that models how the sta … time instant to the next, and $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ is simply the posterio … ous time instant. The term κ is a normalizing term, independent of $\mathbf{x}_k$, … res that posterior integrates to 1.

The probability that the hand can move in the way it did



"Decoding was performed using a Kalman filter which gives an efficient recursive method for ***Bayesian inference*** *…"*

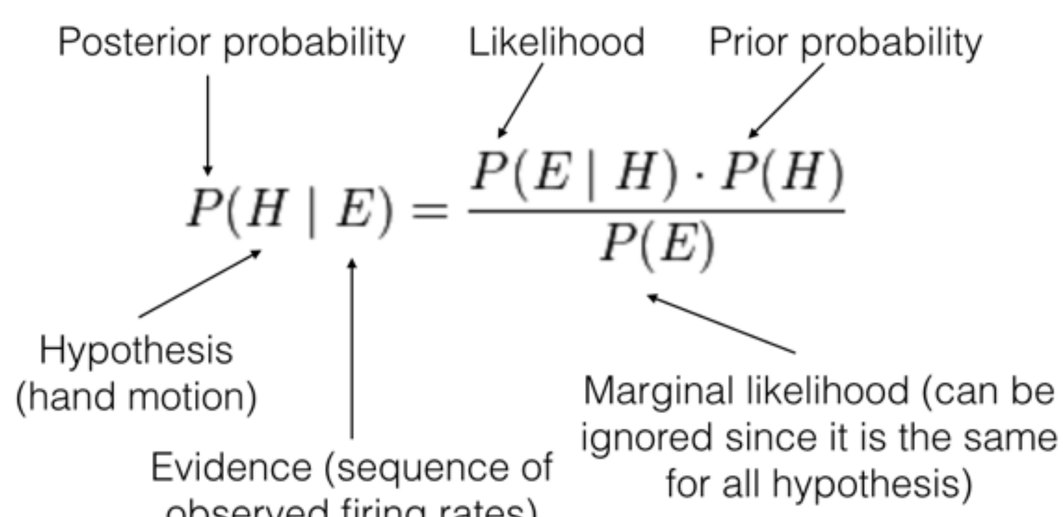
Then, using Bayes Theorem and simple algebra we can write the posterior probability

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa\, p(\mathbf{z}_k|\mathbf{x}_k) \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1}, \quad (8)$$

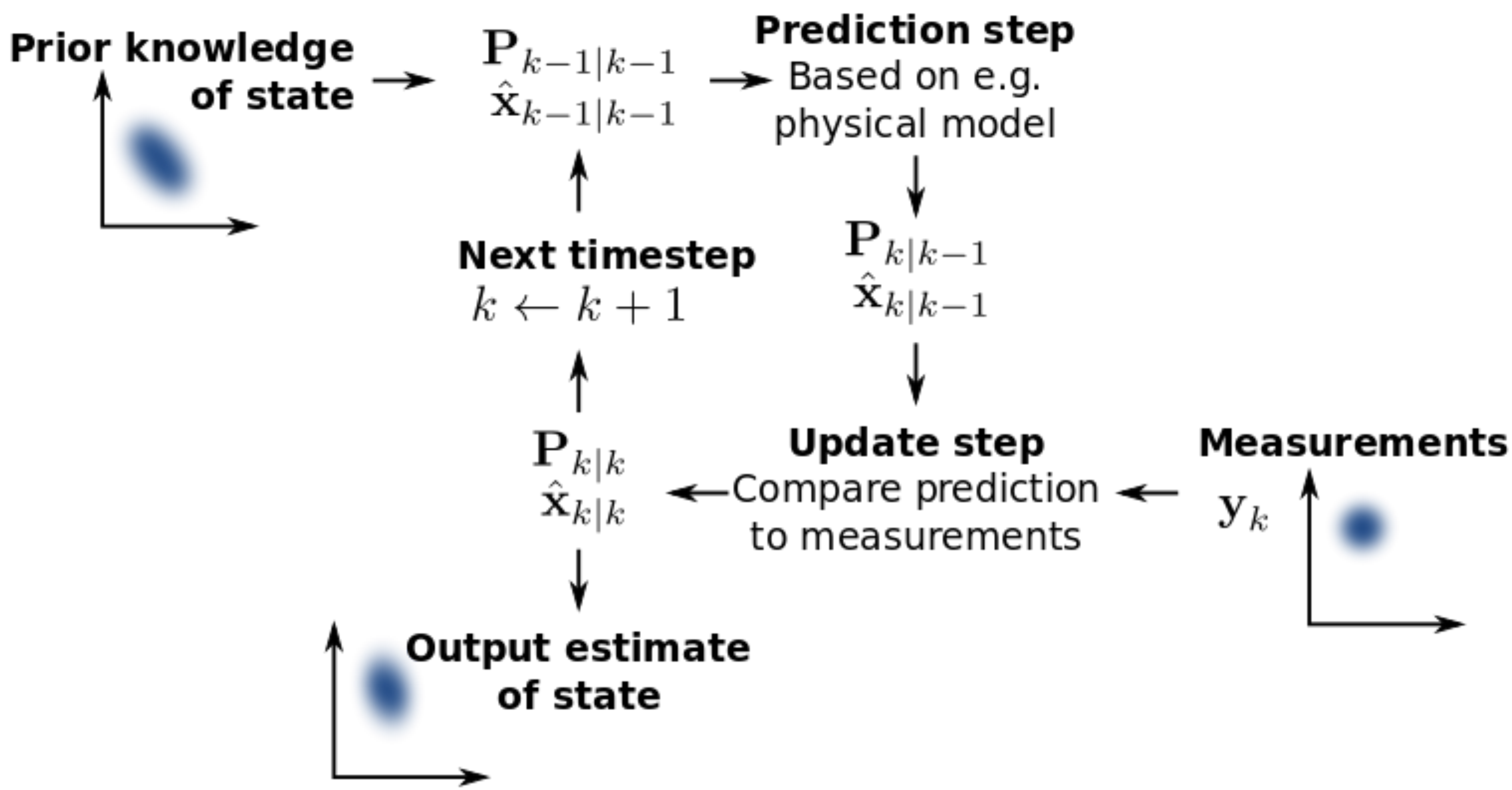
where $p(\mathbf{z}_k|\mathbf{x}_k)$ is called the *likeliho* … of the state, $p(\mathbf{x}_k|$ …) i … *…ral prior* that models how the sta … time instant to … $_{-1})$ is simply the posterio … ous time insta … lizing term, independent of $\mathbf{x}_k$, … es that posterior integra …

The probability that the hand can move in the way it did

The probability that hand can end up in the state where it was in time k-1



"Decoding was performed using a ***Kalman filter*** *which gives an efficient recursive method for Bayesian inference …"*



"… the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state." (Wikipedia)

“Decoding was performed using a Kalman filter which gives an efficient ***recursive method*** *for Bayesian inference …”*

"Decoding was performed using a Kalman filter which gives an efficient ***recursive method*** *for Bayesian inference …"*

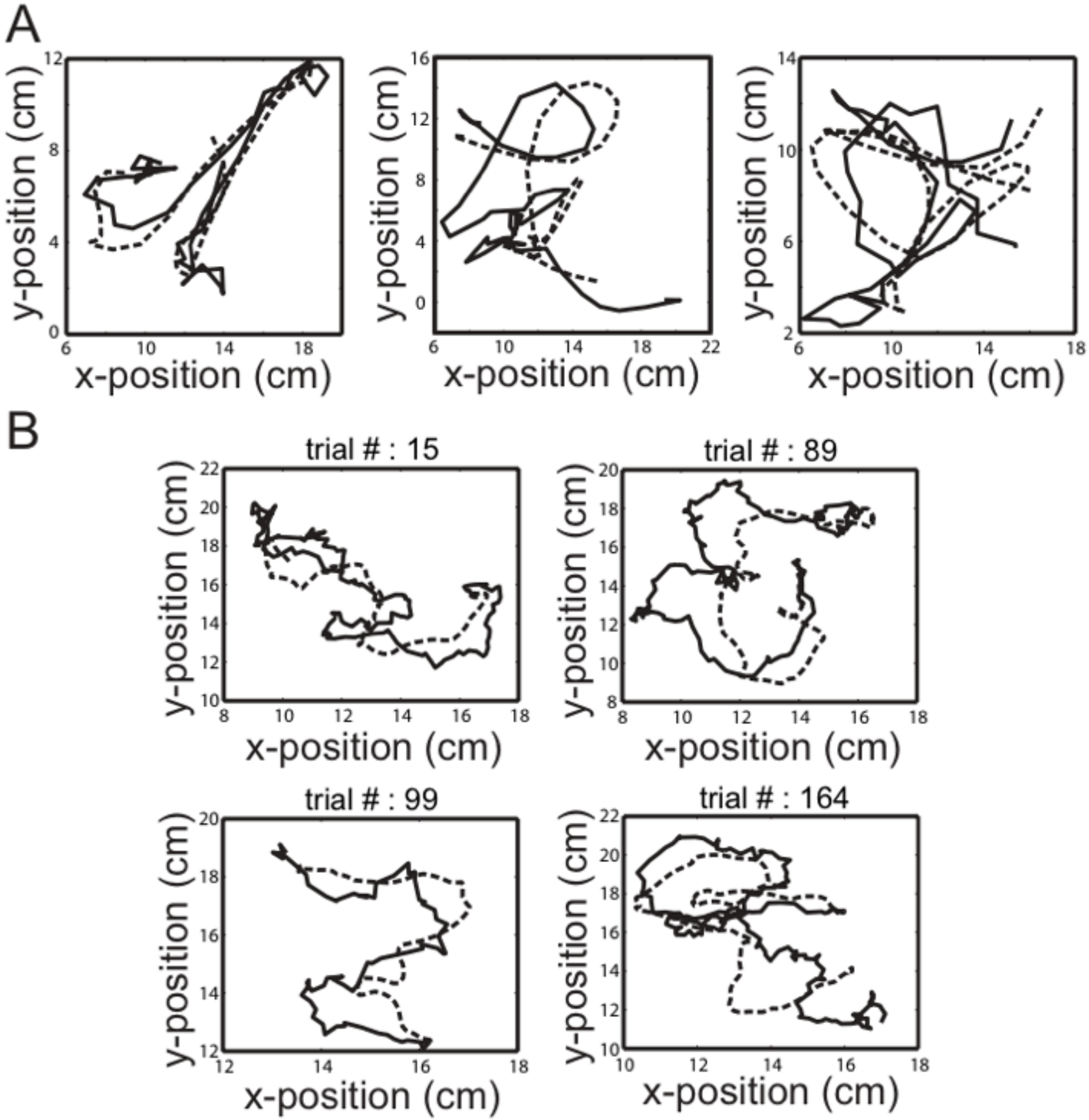
Decoding then involves estimating the posterior probability, $p(\mathbf{x}_k|\mathbf{Z}_k)$, at each time instant, which we can do recursively using Eq. (8).

"Decoding was performed using a Kalman filter which gives an efficient ***recursive method*** *for Bayesian inference …"*

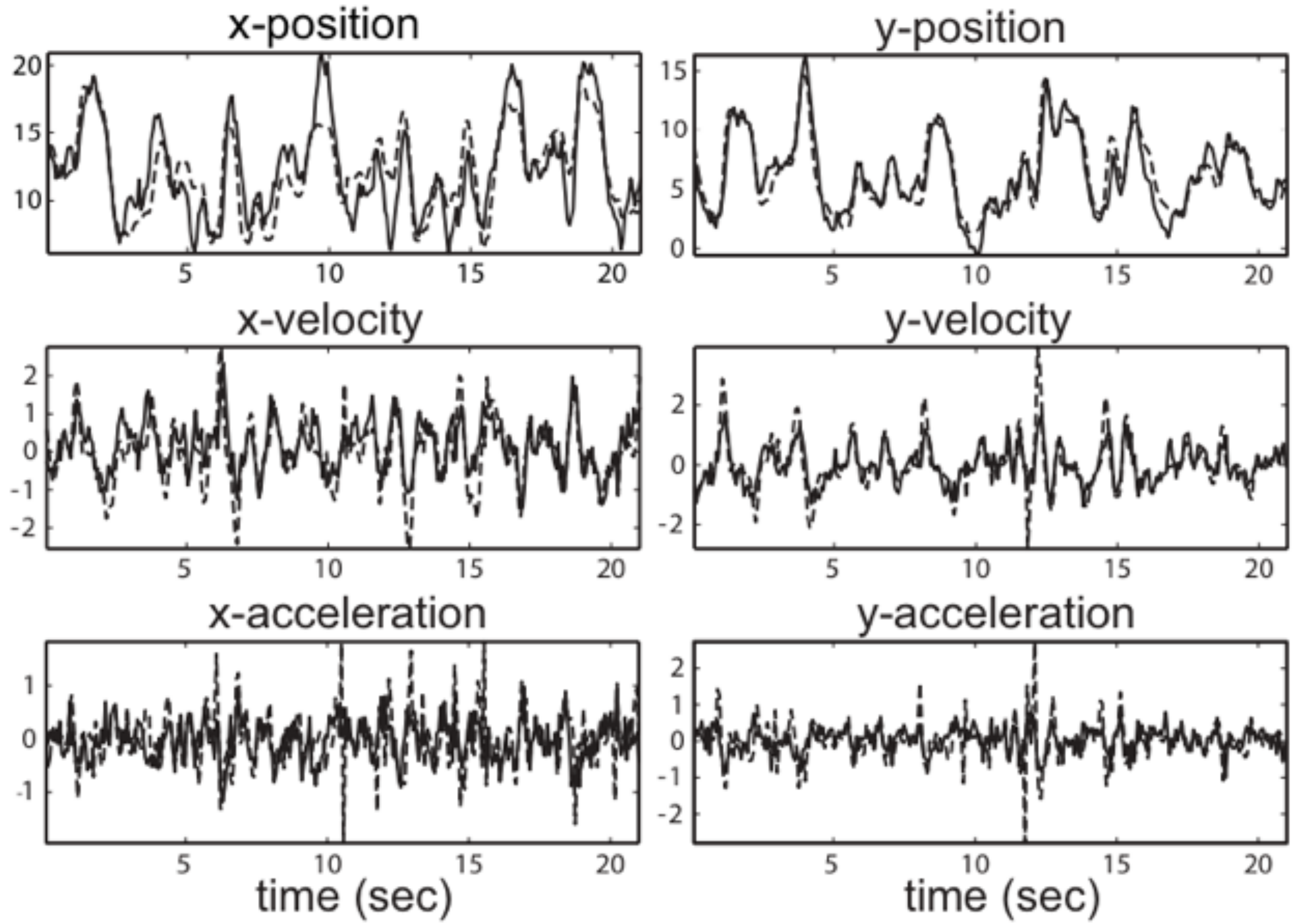
Decoding then involves estimating the posterior probability, $p(\mathbf{x}_k|\mathbf{Z}_k)$, at each time instant, which we can do recursively using Eq. (8).

To achieve a real-time decoding algorithm, we assume that the likelihood and the prior are both Gaussian, and this leads to a closed-form recursive solution for estimating the posterior (which is also Gaussian); this is known as the Kalman filter (Kalman, 1960; Gelb, 1974; Welch and Bishop, 2001).

Results

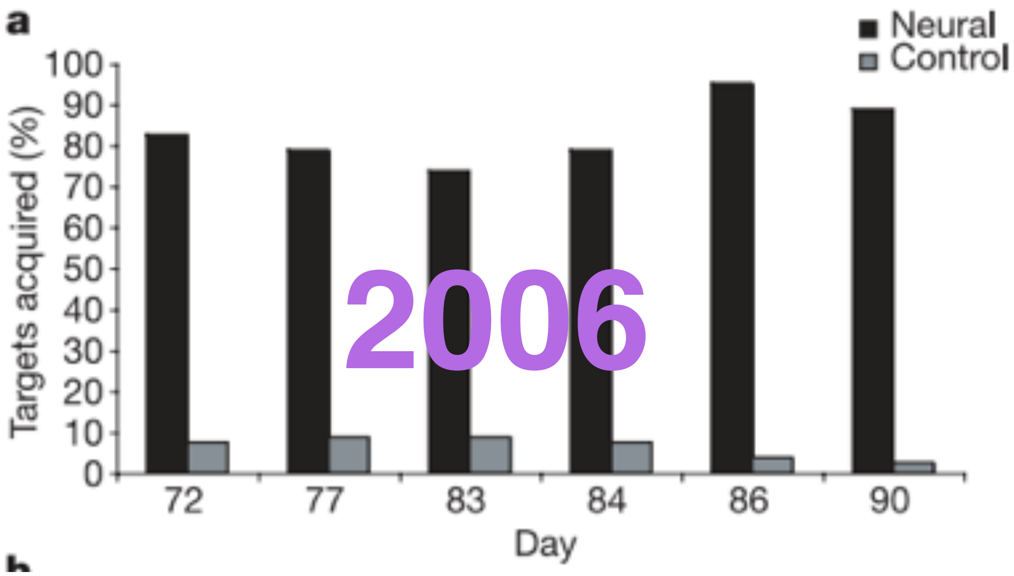


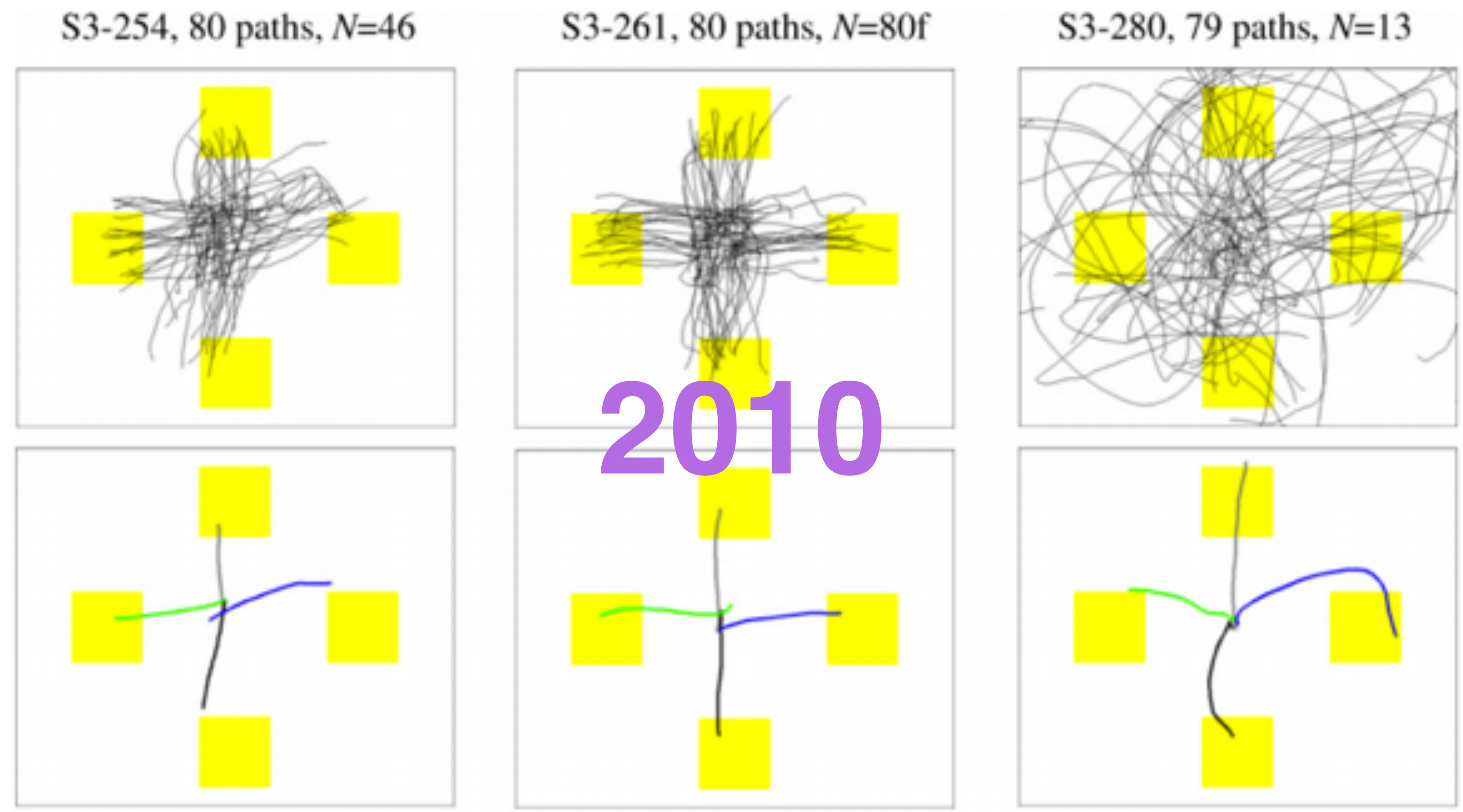
A
y-position (cm)
x-position (cm)
B
trial # : 15
trial # : 89
trial # : 99
trial # : 164

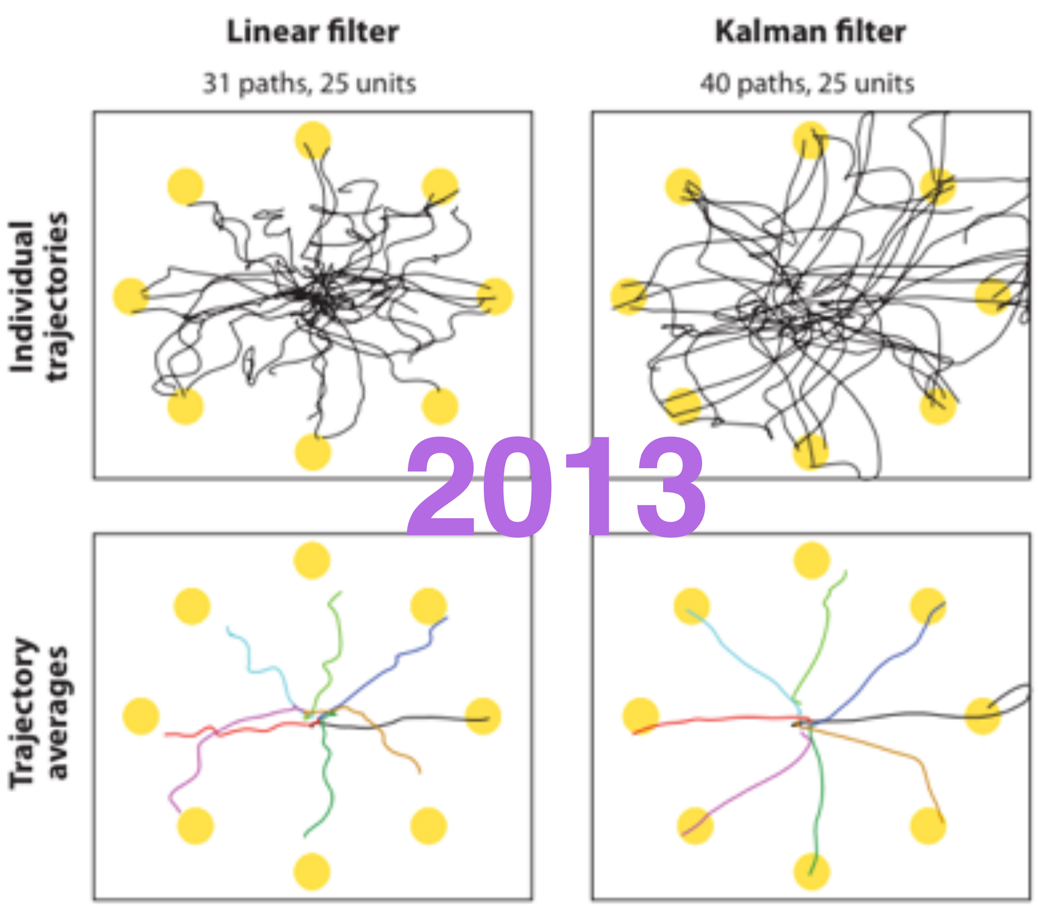


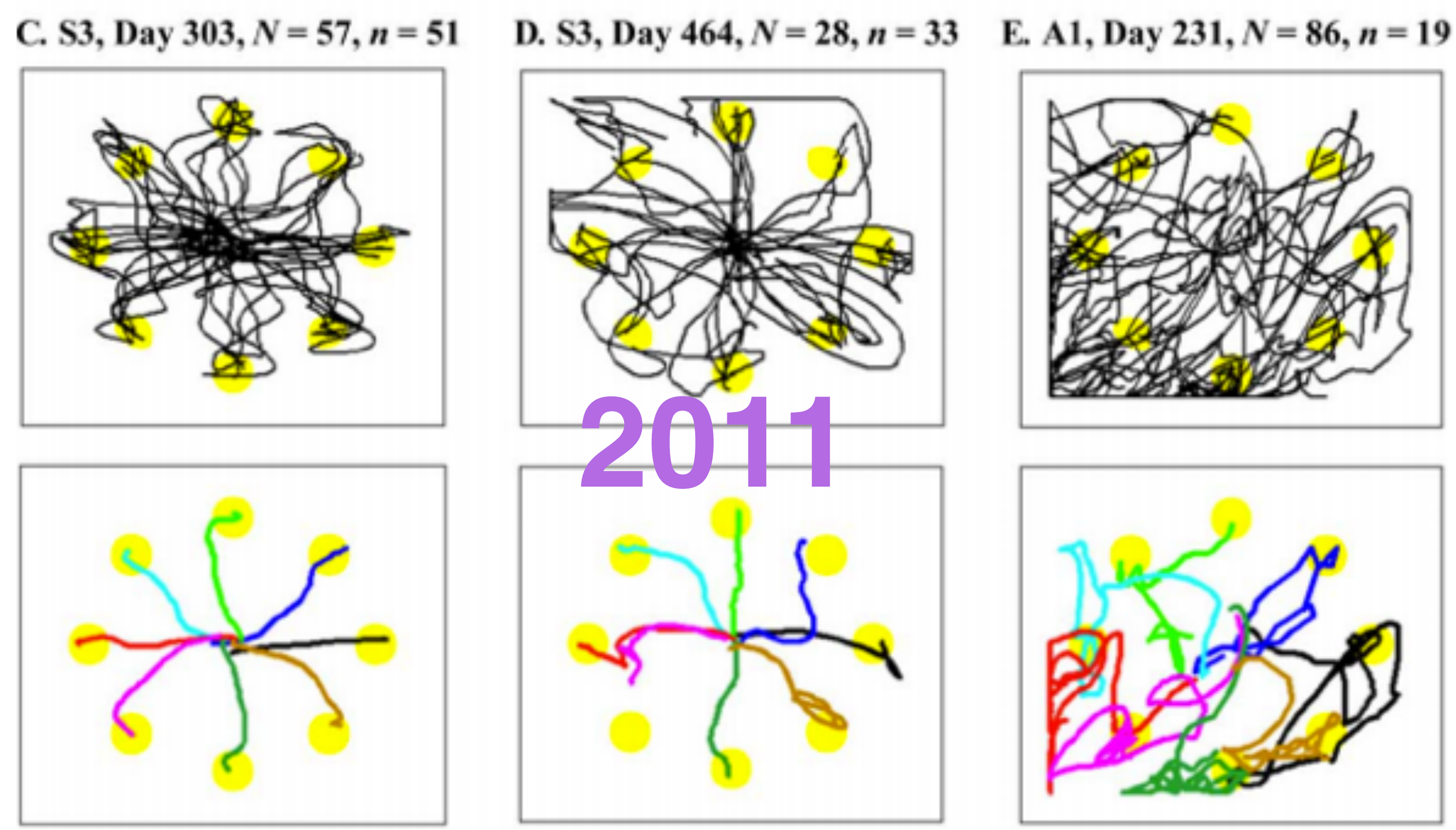
Method	$CC\ (x, y)$	$MSE\ (cm^2)$
Kalman (0ms lag)	$(0.78, 0.91)$	7.01
Kalman (70ms lag)	$(0.80, 0.93)$	6.25
Kalman (140ms lag)	**$(0.82, 0.93)$**	**5.87**
Kalman (210ms lag)	$(0.81, 0.89)$	6.80
Kalman (280ms lag)	$(0.76, 0.82)$	8.81
Kalman (non-uni lag, init 1)	**$(0.84, 0.93)$**	**4.75**
Kalman (non-uni lag, init 2)	$(0.84, 0.93)$	4.77

2012	Reach and grasp by people with tetraplegia using a neurally controlled robotic arm
2011	Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems
	Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array
	Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia
2010	Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia
2006	Neuronal ensemble control of prosthetic devices by a human with tetraplegia
	Bayesian Population Decoding of Motor Cortical Activity using a Kalman Filter


a
Neural
Control
Targets acquired (%)
100
90
80
70
60
50
40
30
20
10
0
72
77
83
84
86
90
Day
2006

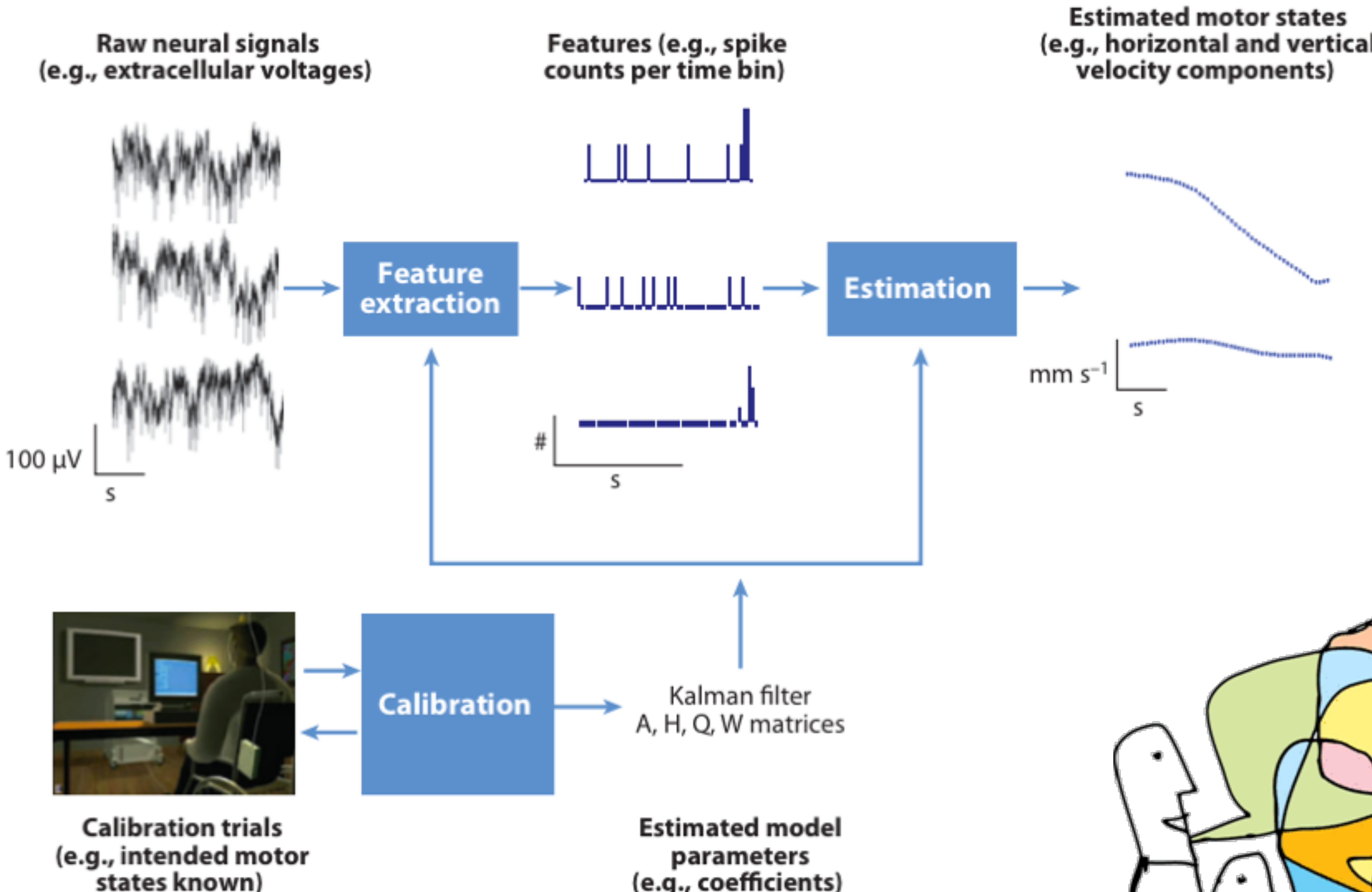

S3-254, 80 paths, N=46
S3-261, 80 paths, N=80f
S3-280, 79 paths, N=13
2010

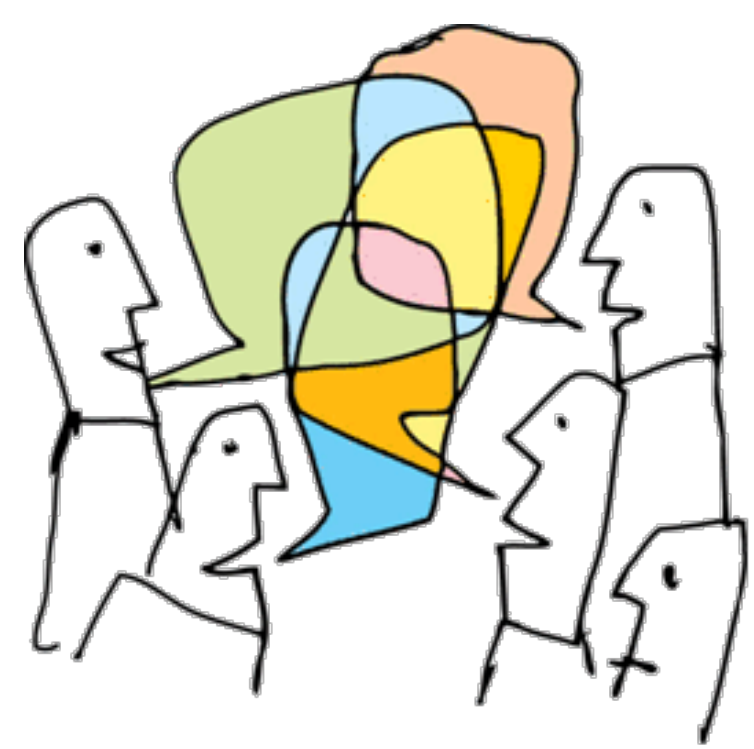

Linear filter
31 paths, 25 units
Kalman filter
40 paths, 25 units
Individual trajectories
Trajectory averages
2013


C. S3, Day 303, N = 57, n = 51
D. S3, Day 464, N = 28, n = 33
E. A1, Day 231, N = 86, n = 19
2011

The **steady-state Kalman filter** significantly **increases** the computational **efficiency** for even relatively simple neural spiking data sets from a human NIS. <…> The **decoding complexity is reduced dramatically** by the SSKF, resulting in approximately seven-fold reduction in the execution time for decoding a typical neuronal firing rate signal.

Summary





http://braingate2.org/publications.asp