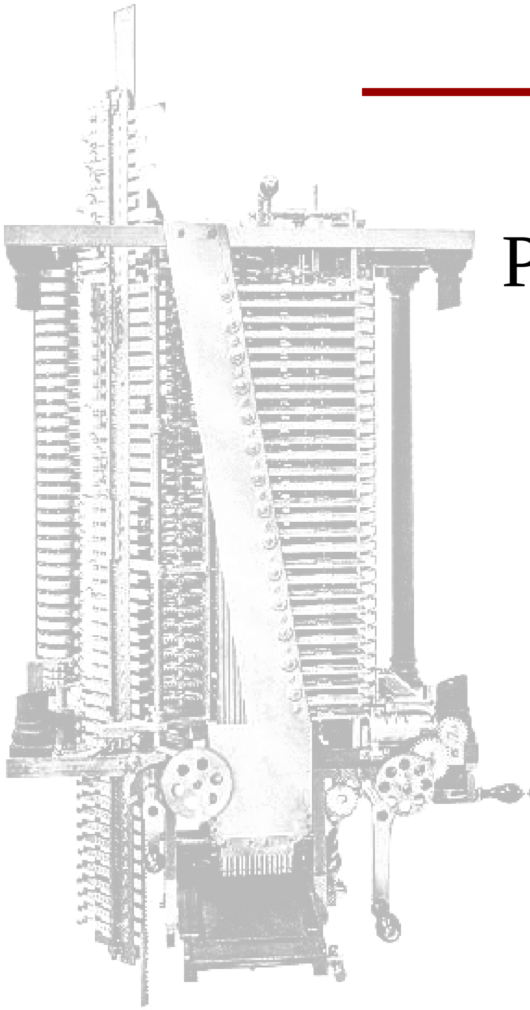
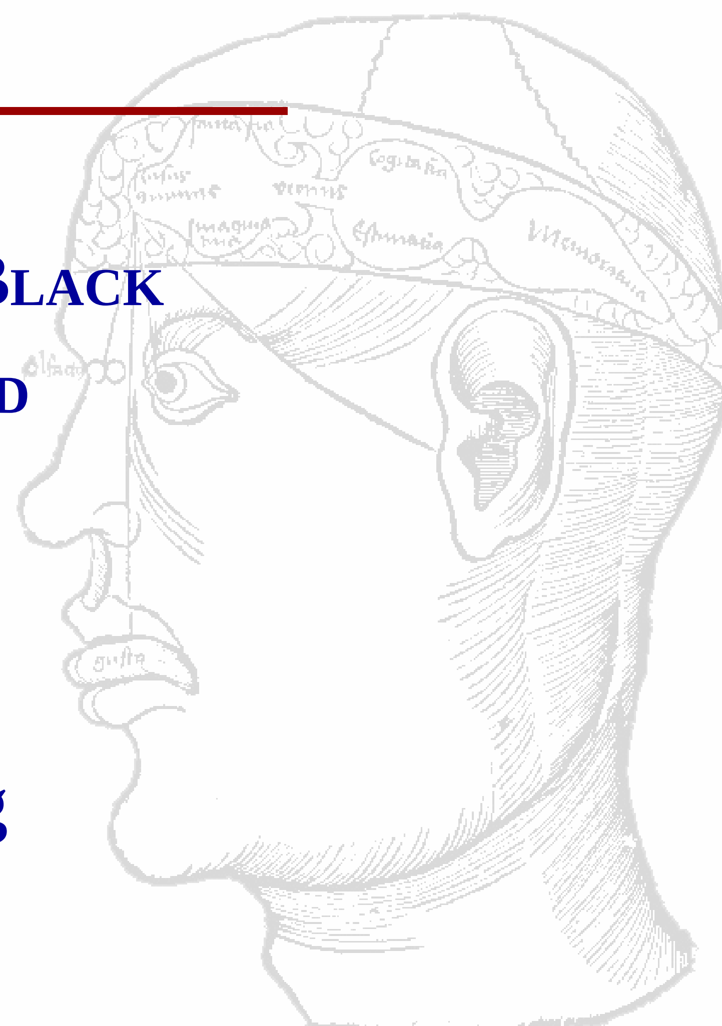
Topics in Brain Computer Interfaces
CS295-7



Professor: **MICHAEL BLACK**

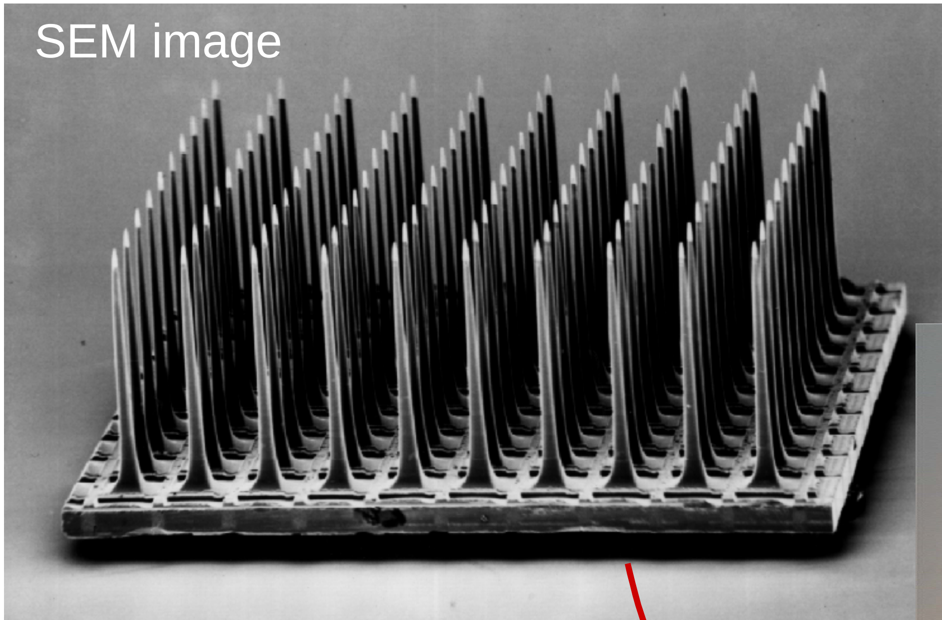
TA: **FRANK WOOD**



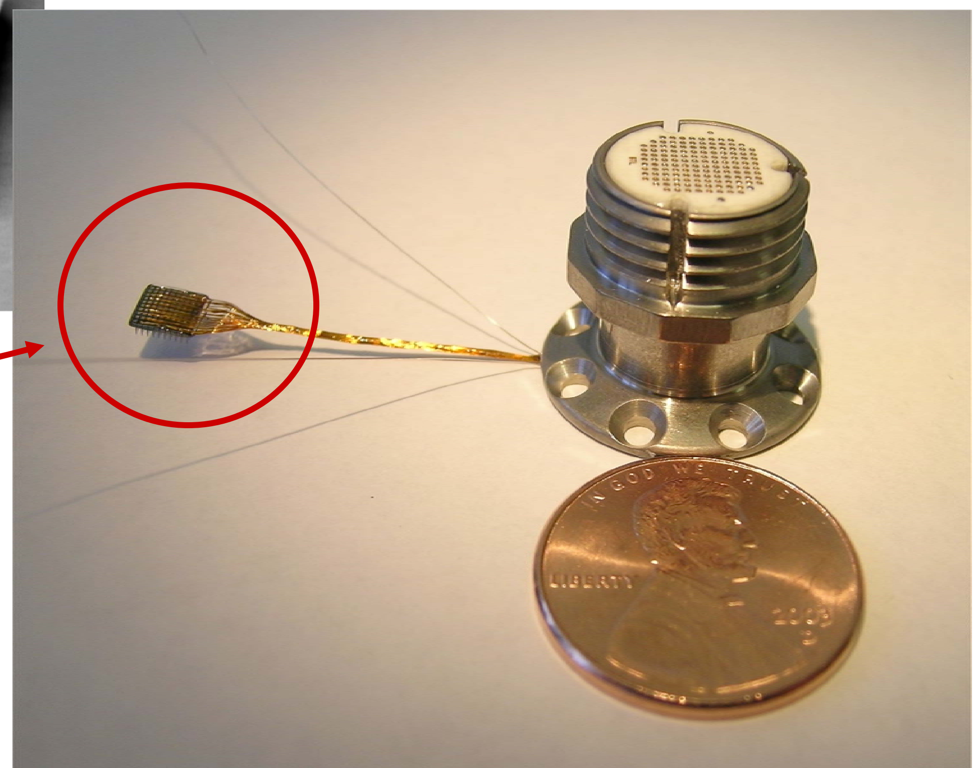
Spring 2005

Linear Filtering

Show & Tell



Extra-cellular recording

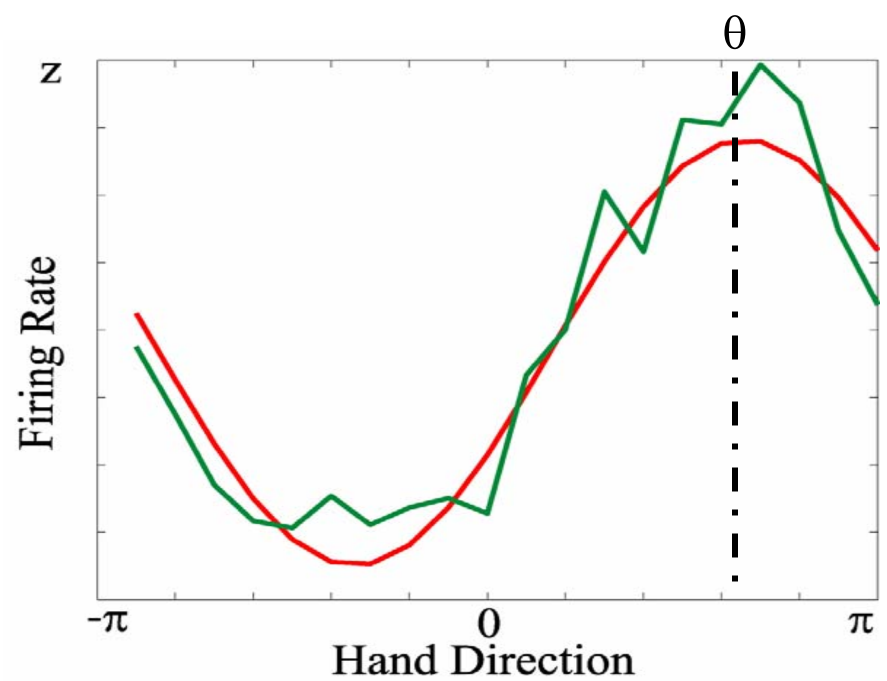


100 “ideal” microelectrodes
10x10 grid,
4x4 mm platform
1 or 1.5 mm long, Si shafts,
Pt coated tips
Glass separation
Parylene insulation coating

Encoding

Georgopoulos et al ('82): (***cosine tuning*** of single cells)

$$z_k = h_0 + h\cos(\theta_k - \theta)$$
$$= h_0 + h_x \cos(\theta_k) + h_y \sin(\theta_k)$$



z_k = firing **rate**, θ_k = hand **direction**

How do we compute the linear coefficients?

Encoding

$$z_k = h_0 + h_x \cos(\theta_k) + h_y \sin(\theta_k)$$

$$z_{k-1} = h_0 + h_x \cos(\theta_{k-1}) + h_y \sin(\theta_{k-1})$$

$$z_{k-2} = h_0 + h_x \cos(\theta_{k-2}) + h_y \sin(\theta_{k-2})$$

$$\vdots$$

k equations in 3 unknowns.

Linear system of equations.

Encoding

$$z_k = h_0 + h_x \cos(\theta_k) + h_y \sin(\theta_k)$$

$$z_{k-1} = h_0 + h_x \cos(\theta_{k-1}) + h_y \sin(\theta_{k-1})$$

$$z_{k-2} = h_0 + h_x \cos(\theta_{k-2}) + h_y \sin(\theta_{k-2})$$

$$\vdots$$

$$\begin{bmatrix} z_k \\ z_{k-1} \\ z_{k-2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & \cos(\theta_k) & \sin(\theta_k) \\ 1 & \cos(\theta_{k-1}) & \sin(\theta_{k-1}) \\ 1 & \cos(\theta_{k-2}) & \sin(\theta_{k-2}) \\ & \vdots & \end{bmatrix} \begin{bmatrix} h_0 \\ h_x \\ h_y \end{bmatrix}$$

Encoding

$$\mathbf{z} = \mathbf{A}\mathbf{h}$$

Solve for $\mathbf{h}$

Recall A^{-1} exists only when A is square and full rank.

$$\begin{bmatrix} z_k \\ z_{k-1} \\ z_{k-2} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & \cos(\theta_k) & \sin(\theta_k) \\ 1 & \cos(\theta_{k-1}) & \sin(\theta_{k-1}) \\ 1 & \cos(\theta_{k-2}) & \sin(\theta_{k-2}) \\ & \vdots & \end{bmatrix} \begin{bmatrix} h_0 \\ h_x \\ h_y \end{bmatrix}$$

Encoding

$$\mathbf{z} = \mathbf{A}\mathbf{h}$$

Solve for $\mathbf{h}$

$$\mathbf{A}^T\mathbf{z} = \mathbf{A}^T\mathbf{A}\mathbf{h}$$

Square

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{z} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{A})\mathbf{h}$$

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{z} = \mathbf{h}$$

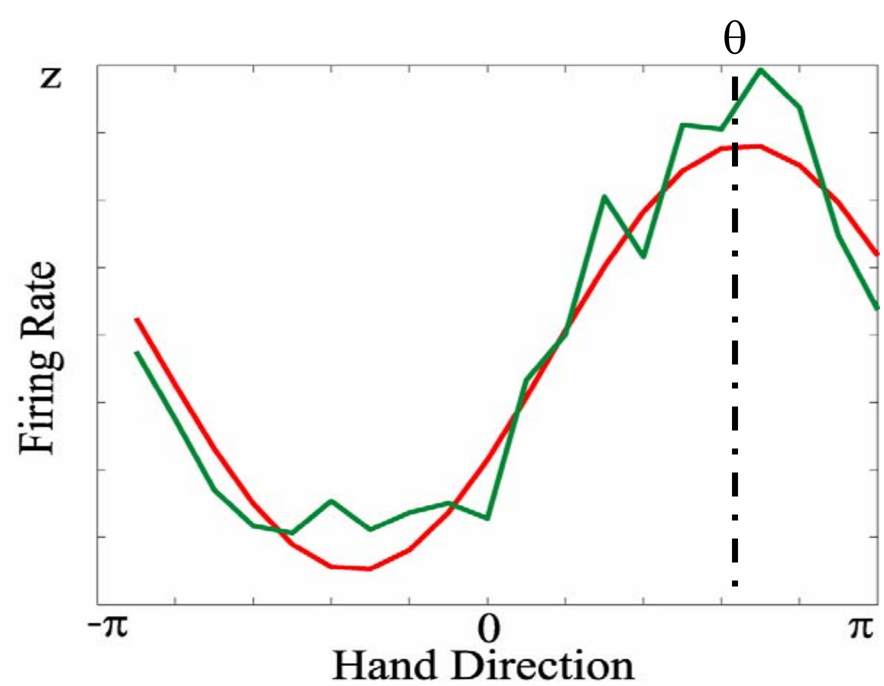
$$\mathbf{A}^+\mathbf{z} = \mathbf{h}$$

Pseudo inverse

Encoding

Georgopoulos et al ('82): (***cosine tuning*** of single cells)

$$z_k = h_0 + h\cos(\theta_k - \theta)$$
$$= h_0 + h_x \cos(\theta_k) + h_y \sin(\theta_k)$$



z_k = firing **rate**, θ_k = hand **direction**

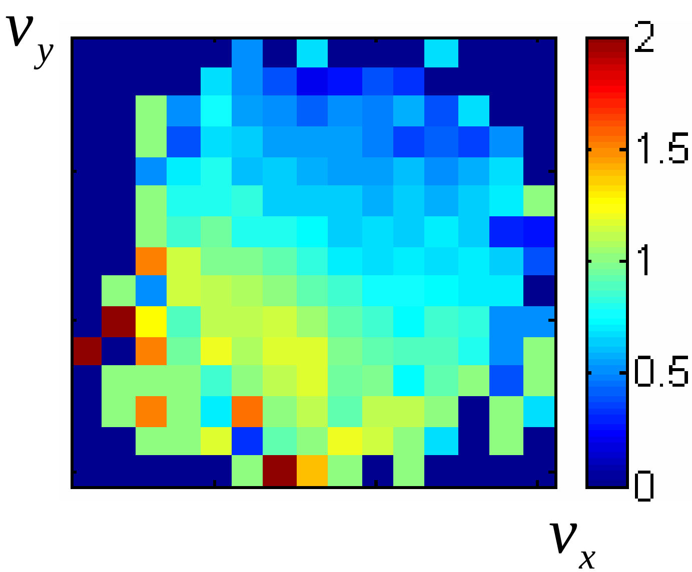
This is not the whole story!

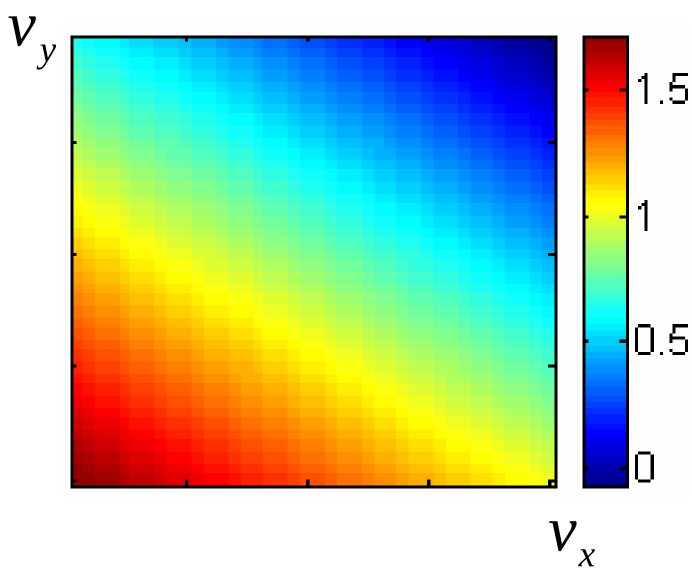
Encoding

Moran & Schwartz ('99):

$$z_k = h_0 + speed\,(h_x \cos(\theta_k) + h_y \sin(\theta_k))$$

$$= h_0 + h_x v_{x_k} + h_y v_{y_k} \quad \text{(}\textbf{\textit{Linear}}\text{ in velocity)}$$

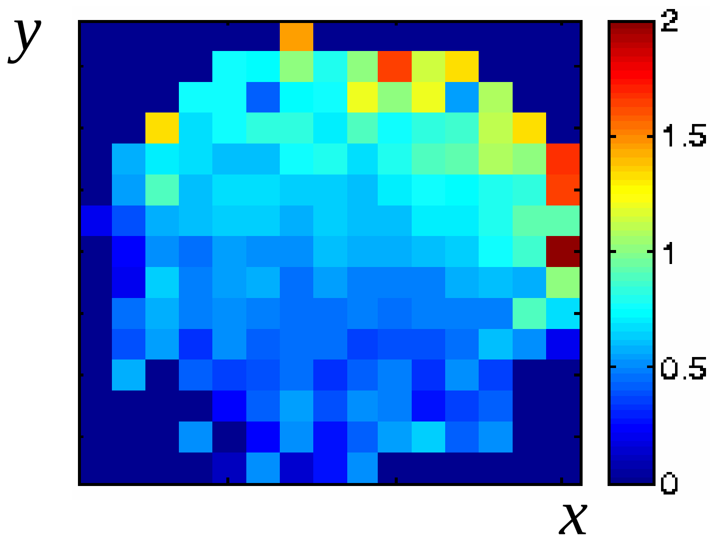


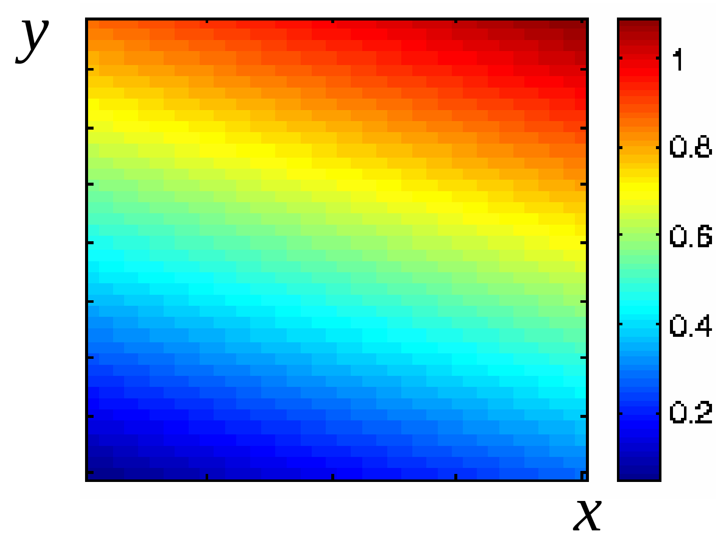


Encoding

Kettner et al ('88):

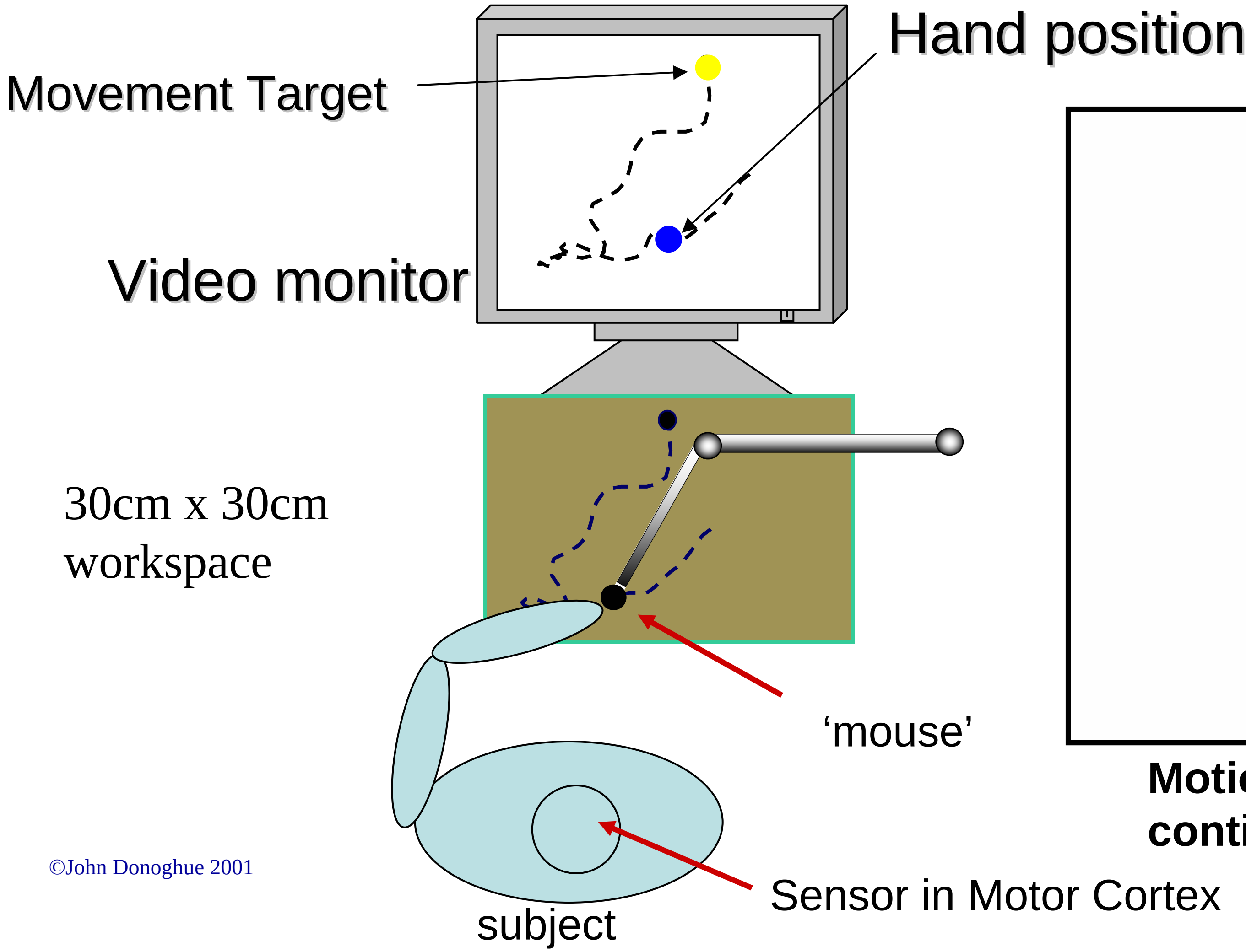
$$z_k = b_0 + b_x x_k + b_y y_k$$ (***Linear*** in position)

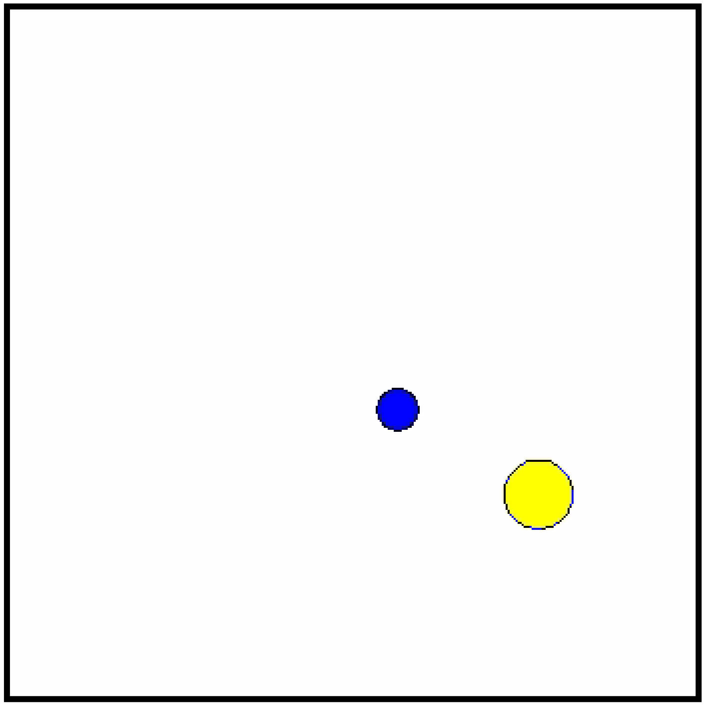




Flament et al ('88): Firing rate is also related to hand acceleration

Behavior and Neural Firing





Motions are 2D, fast, continuous, and "natural".

Data

Firing rates of *n* cells at time *k*

$$\mathbf{z}_k = \begin{bmatrix} z_{1,k} \\ z_{2,k} \\ \vdots \\ z_{n,k} \end{bmatrix}$$

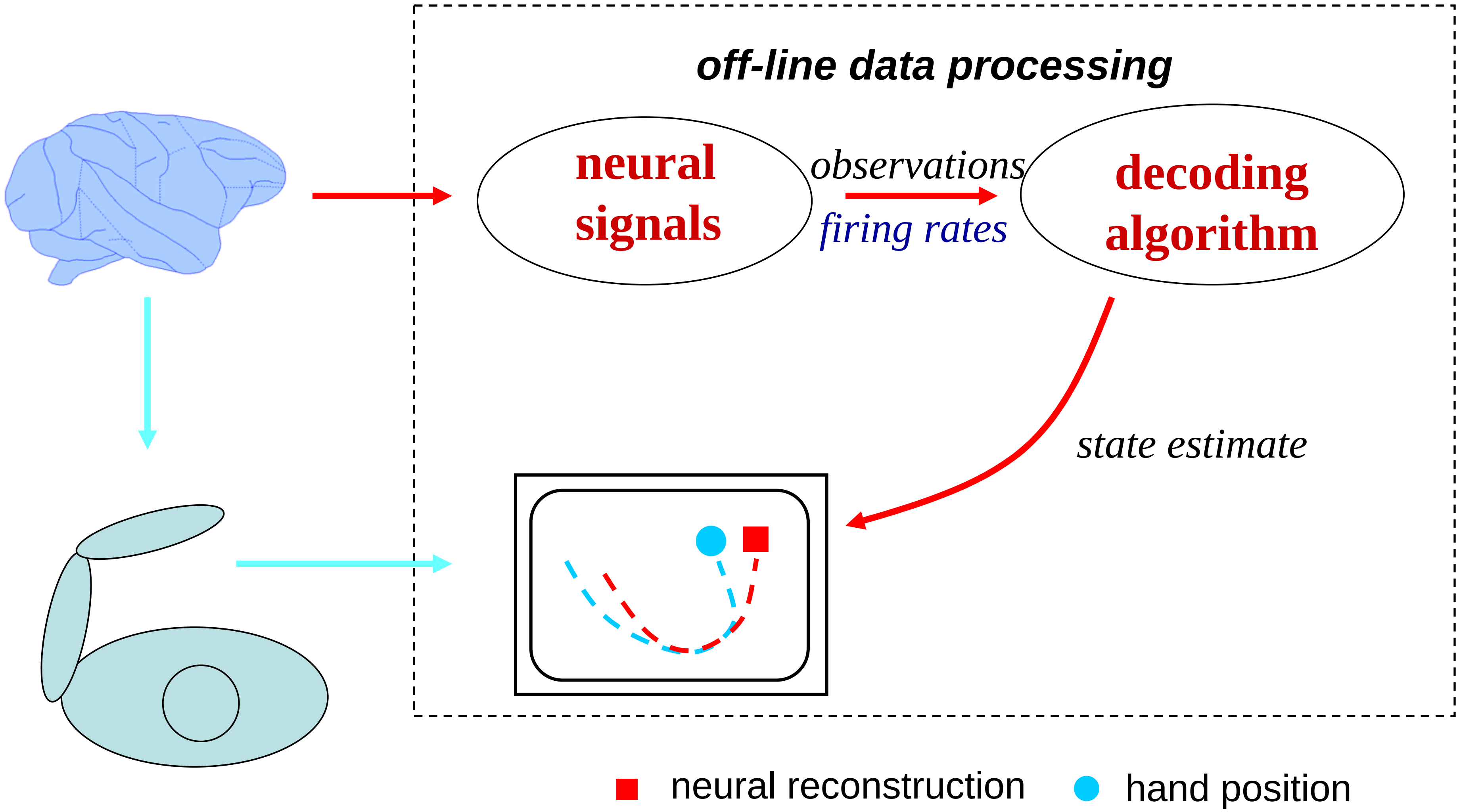
Hand position at time *k*

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

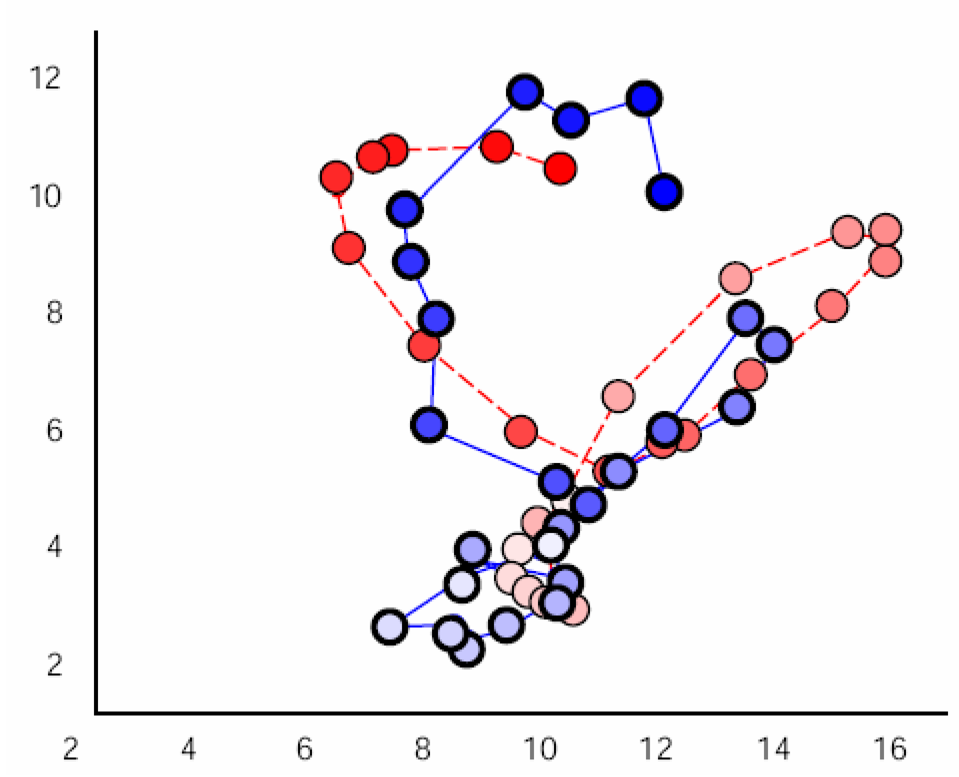
Firing rates computed in 50 or 70msec time bins.

Training data 1-2 minutes of data (~1200 time bins).

Off-line Reconstruction



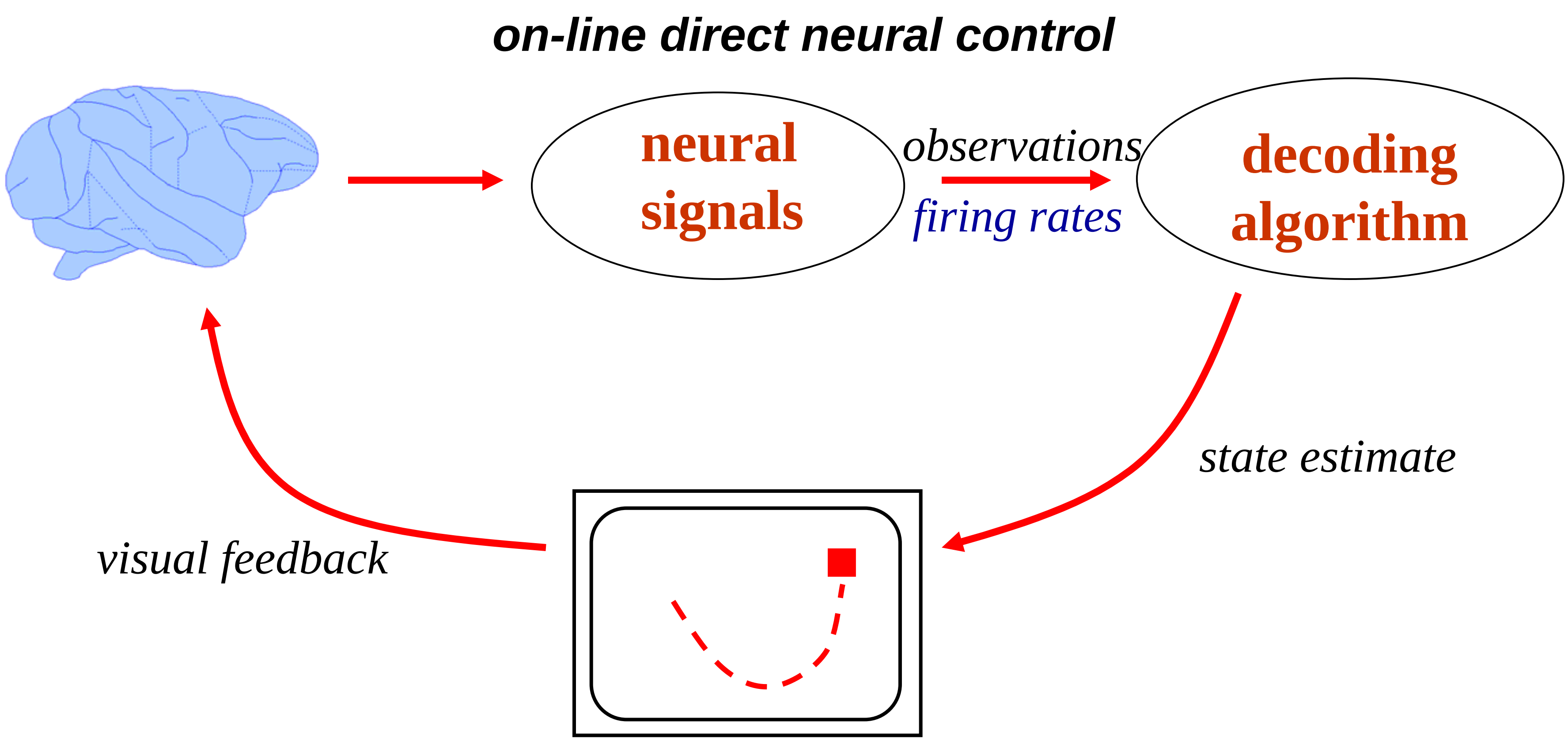
Off-line Reconstruction



True

Reconstruction

Closed-loop Neural Control



- Cursor under neural control

Linear Regression

We want to reconstruct hand position from firing rates.
Let's assume a linear relationship:

$$x_k = \mathbf{f}_x^T \mathbf{z}_k + b_x$$

$$y_k = \mathbf{f}_y^T \mathbf{z}_k + b_y$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} \mathbf{f}_x^T \\ \mathbf{f}_y^T \end{bmatrix} \mathbf{z}_k + \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

Linear Regression

We want to reconstruct hand position from firing rates.
Let's assume a linear relationship:

$$x_k = \mathbf{f}_x^T \mathbf{z}_k + b_x$$

$$x_k = \begin{bmatrix} f_{x,1} \cdots f_{x,n} \end{bmatrix} \begin{bmatrix} z_{k,1} \\ \vdots \\ z_{k,n} \end{bmatrix} + b_x = \begin{bmatrix} b_x & f_{x,1} \cdots f_{x,n} \end{bmatrix} \begin{bmatrix} 1 \\ z_{k,1} \\ \vdots \\ z_{k,n} \end{bmatrix}$$

Linear Regression

In 50msec a cell only fires a few times.
Looking at a small population of cells (a few dozen say) will give a very noisy estimate of hand position.

$$x_k = \mathbf{f}_x^T \mathbf{z}_k$$

$$y_k = \mathbf{f}_y^T \mathbf{z}_k$$

Let's assume that hand position is related to the firing rates over the last second of firing data.

$$x_k = \mathbf{f}_x^T [1 \; \mathbf{z}_k^T \; \mathbf{z}_{k-1}^T \cdots \mathbf{z}_{k-20}^T]^T$$

$$y_k = \mathbf{f}_y^T [1 \; \mathbf{z}_k^T \; \mathbf{z}_{k-1}^T \cdots \mathbf{z}_{k-20}^T]^T$$

Linear Regression

$$x_k = \mathbf{f}_x^T [1 \; \mathbf{z}_k^T \; \mathbf{z}_{k-1}^T \cdots \mathbf{z}_{k-20}^T]^T$$

$$y_k = \mathbf{f}_y^T [1 \; \mathbf{z}_k^T \; \mathbf{z}_{k-1}^T \cdots \mathbf{z}_{k-20}^T]^T$$

$$\begin{bmatrix} 1 \\ z_{1,k} \\ z_{2,k} \\ \vdots \\ z_{n,k} \\ z_{1,k-1} \\ \vdots \\ z_{n,k-1} \\ \vdots \\ z_{n,k-20} \end{bmatrix}$$

"Filter":

$$\lfloor b_y \quad f_{y,1} \quad f_{y,2} \quad \cdots \quad f_{y,20n} \rfloor$$

One equation in $20n+1$ unknowns.

Linear Regression

Solve for $\mathbf{f}_x$

$$x_k = [1\ \mathbf{z}_k^T\ \mathbf{z}_{k-1}^T \cdots \mathbf{z}_{k-20}^T]\mathbf{f}_x$$

$$x_{k-1} = [1\ \mathbf{z}_{k-1}^T\ \mathbf{z}_{k-2}^T \cdots \mathbf{z}_{k-21}^T]\mathbf{f}_x$$

$$x_{k-2} = [1\ \mathbf{z}_{k-2}^T\ \mathbf{z}_{k-3}^T \cdots \mathbf{z}_{k-22}^T]\mathbf{f}_x$$

$$\vdots$$

$$\mathbf{x} = \mathbf{Z}\mathbf{f}_x$$

Linear Regression

Solving for $\mathbf{f}_x$

$$\mathbf{x} = \mathbf{Z}\mathbf{f}_x$$

$$\mathbf{f}_x = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{x}$$

$$\mathbf{f}_x = \text{pinv}(\mathbf{Z})\mathbf{x}$$

Least Squares

$$\mathbf{Z}\mathbf{f} = \mathbf{x}$$

We want the solution for $\mathbf{f}$ that minimizes the error; e.g.:

$$\min_{\mathbf{f}} \left\| \mathbf{Z}\mathbf{f} - \mathbf{x} \right\|^2$$

Least Squares

$$\|\mathbf{Zf}-\mathbf{x}\|^2 = (\mathbf{Zf}-\mathbf{x})^T(\mathbf{Zf}-\mathbf{x})$$

$$= \sum_k (\mathbf{Z}_k\mathbf{f} - x_k)^2$$

$$= \sum_k (b + f_1 z_{k,1} + \cdots + f_{20n} z_{k-20,n} - x_k)^2$$

Claim: $\mathbf{f} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{x}$ minimizes the square error

Least Squares

Minimize over $\mathbf{f}$

$$(\mathbf{Zf}-\mathbf{x})^T(\mathbf{Zf}-\mathbf{x})$$

$$=\mathbf{f}^T\mathbf{Z}^T\mathbf{Zf}-\mathbf{f}^T\mathbf{Z}^T\mathbf{x}-\mathbf{x}^T\mathbf{Zf}+\mathbf{x}^T\mathbf{x}$$

We can ignore this since it does now involve $\mathbf{f}$

$$=\mathbf{f}^T\mathbf{Z}^T\mathbf{Zf}-2\mathbf{f}^T\mathbf{Z}^T\mathbf{x}$$

Least Squares

Let

$$P(\mathbf{f}) = \frac{1}{2}\mathbf{f}^T\mathbf{Z}^T\mathbf{Z}\mathbf{f} - \mathbf{f}^T\mathbf{Z}^T\mathbf{x}$$

This is a quadratic equation $\frac{1}{2}\mathbf{f}^T\mathbf{C}\mathbf{f} - \mathbf{f}^T\mathbf{d}$

How do we solve for **f**?

Least Squares

Answer is always the same:

Differentiate wrt **f**, set it equal to zero, and solve the linear equations.

$$\frac{dP}{d\mathbf{f}} = \mathbf{C}\mathbf{f} - \mathbf{d} = 0$$

$$\mathbf{f} = \mathbf{C}^{-1}\mathbf{d}$$

Substituting: $\mathbf{f} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{x}$

Linear Regression

Fit $\mathbf{f}_x$ and $\mathbf{f}_y$ given training data.

Decoding is now simple.

Given a vector of observed firing activity

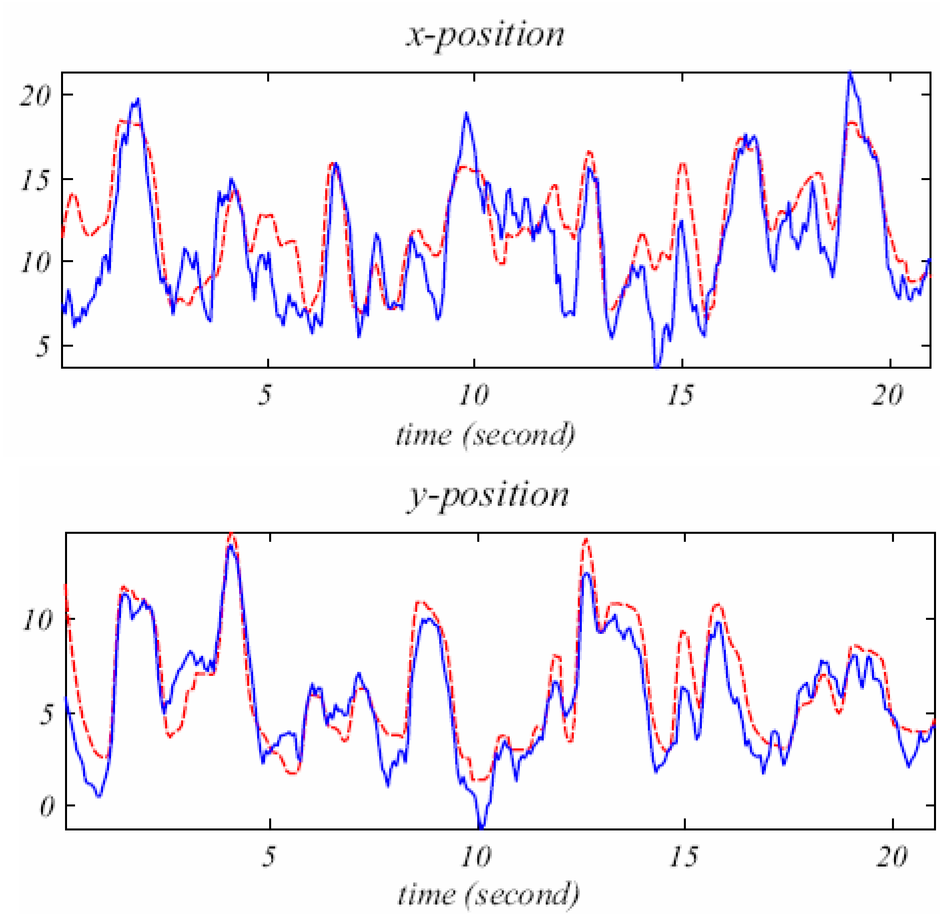
$$\mathbf{Z}_k^T = \begin{bmatrix} 1 & z_{1,k} & \cdots & z_{n,k} & z_{1,k-1} & \cdots & z_{n,k-1} & \cdots & z_{n,k-20} \end{bmatrix}$$

"project" onto the filter:

$$x_k = \mathbf{Z}_k^T \mathbf{f}_x$$

$$y_k = \mathbf{Z}_k^T \mathbf{f}_y$$

Decoding (off-line)



True

Reconstruction

Evaluating Accuracy

let $(\hat{x}_k, \hat{y}_k)$ be the estimate of the true position (x_k, y_k)

Correlation coefficient

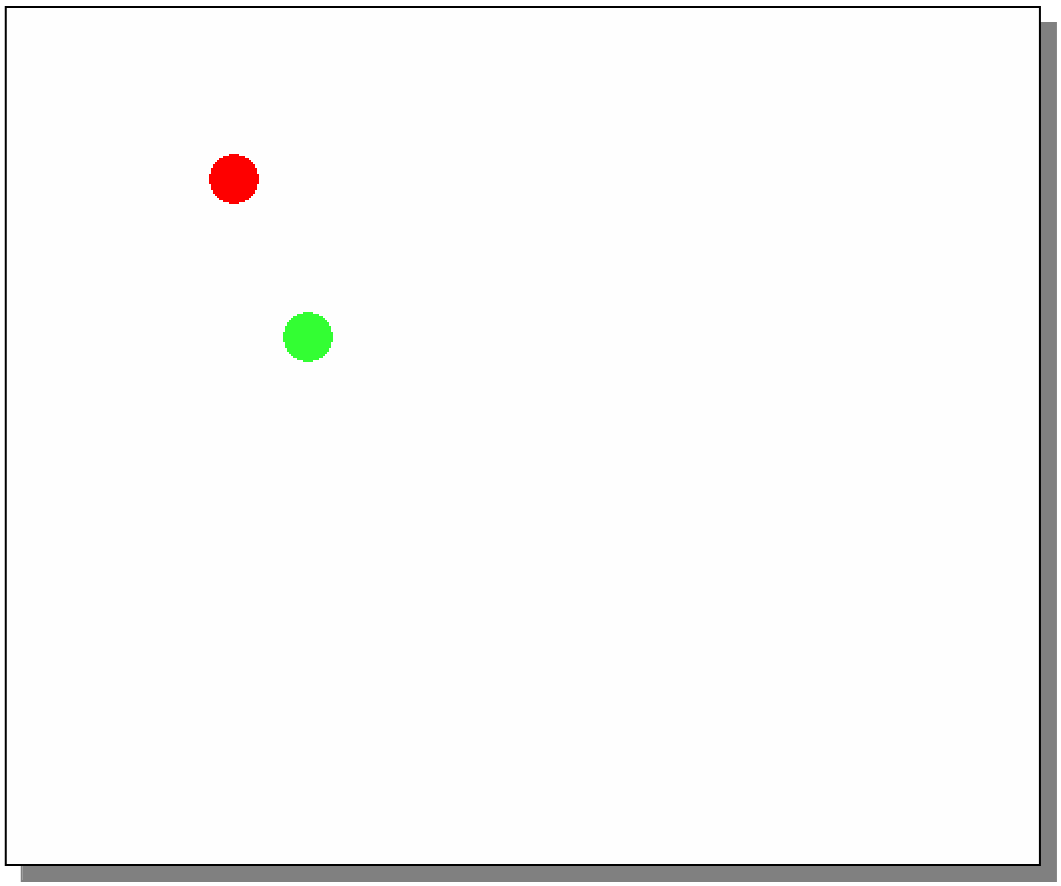
mean position

$$cc_x = \frac{\sum_k (x_k - \bar{x})(\hat{x}_k - \bar{\hat{x}})}{\sqrt{\sum_k (x_k - \bar{x})^2 \sum_k (\hat{x}_k - \bar{\hat{x}})^2}}$$

Mean squared error

$$MSE = \frac{1}{T} \sum_{k=1}^{T} ((x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2)$$

CLOSED LOOP NEURAL CONTROL



Target

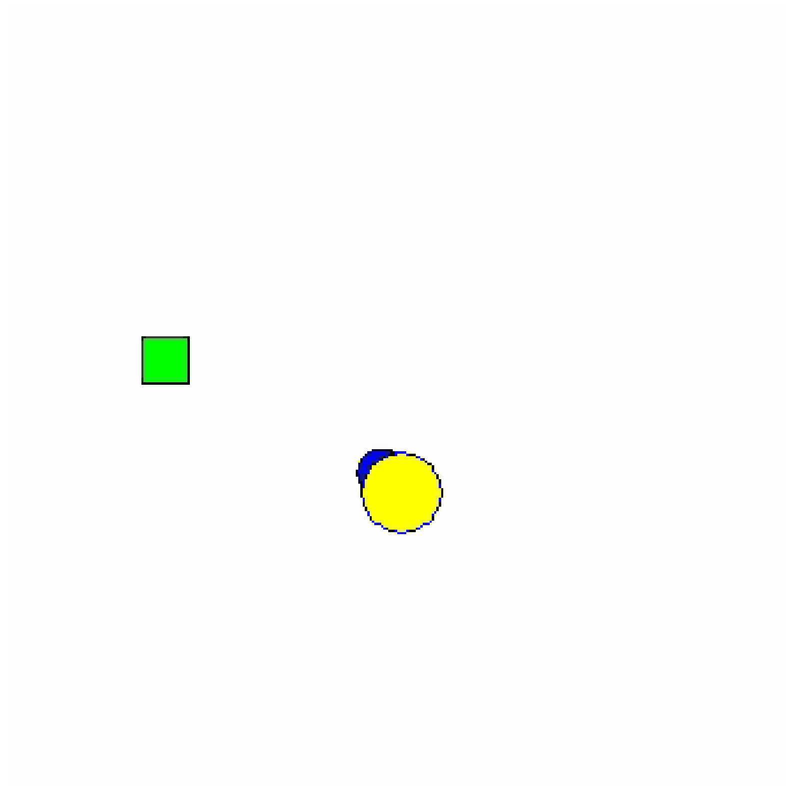
Neural control

Linear filters built on-line.

Heavily smoothed.

Mijail Serruya

Decoding



- Target
- True hand position
- Linear filter reconstruction

69 cells, bin size = 70ms
History used = 10 bins
Lowpass filter length = 8