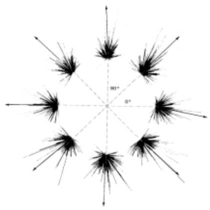
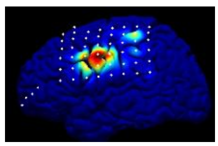
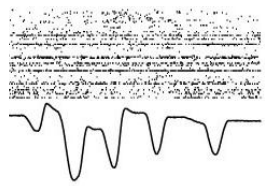
CSE 599E

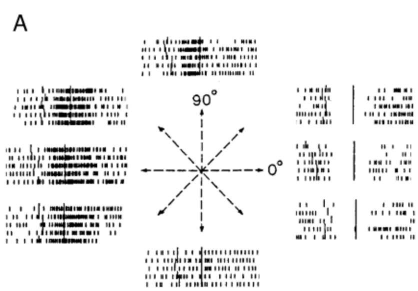
Review of Neural Decoding Techniques



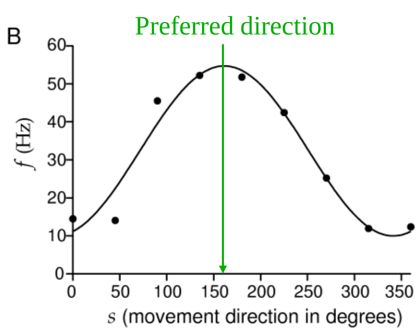




Tuning Curve of a Neuron in Primary Motor Cortex (M1)

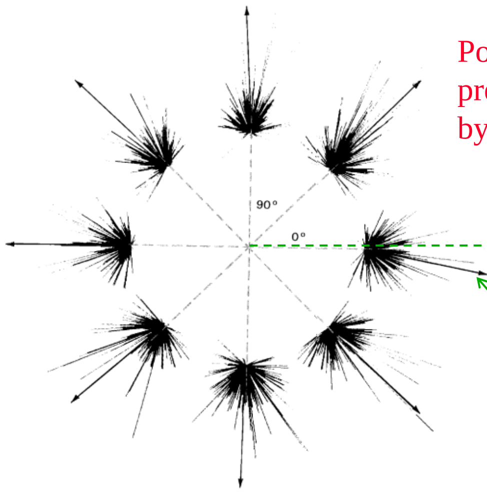


Spike trains as a function of hand reaching direction



Cosine Tuning Curve

Population Vector Method for Decoding Movement Direction



Population vector = sum of preferred directions weighted by their firing rates

Actual arm movement direction

Population vector (decoded movement direction)

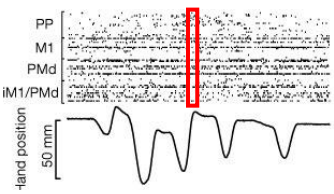
$$\hat{\mathbf{d}} = \sum_i \mathbf{p}_i \left(\frac{r - r_0}{r_{\max}}\right)_i$$

Decoding problem

Input: **x** [vector of firing rates from multiple neurons]

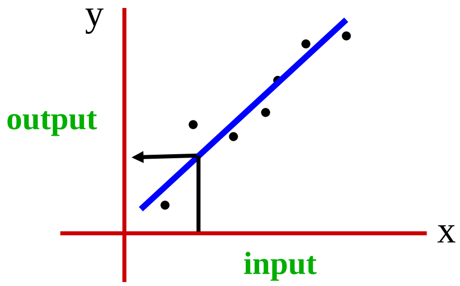
Output: y [Hand position, etc.]

Input vector



Output (hand position)

Linear filter method (regression)



x	y
1	3.1
2	6.4
3.1	8.9
0.9	2

Assumption: Output is a *linear function of input*, i.e.,

$$\mathbf{y_i = wx_i + noise}$$

where noise is independent, Gaussian, unknown fixed variance

Find linear filter **w** by minimizing: $\boldsymbol{\Sigma_i (y_i - wx_i)^2}$

When **x** is a vector…

Suppose input $\mathbf{x}_i$ is n-element vector of firing rates for a neuron for current and past n-1 time steps: $y_i = \mathbf{w}^T\mathbf{x}_i$ + noise

Write the *m* inputs as rows of matrix and outputs y_i as vector:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Then, **Y = Xw + noise**

Find **w** by minimizing $\|\mathbf{Y} - \mathbf{Xw}\|^2$ over all data:

Linear Filter $\mathbf{w = (X^TX)^{-1}X^TY}$

Kalman Filter

Running Average Example (on board)

Kalman Filtering

- Measurement linearly related to state: $y_t = Bx_t + m_t$
- State linearly evolves according to: $x_t = Ax_{t-1} + n_t$
- n_t and m_t are zero-mean Gaussian noise with covariance matrices Q and R respectively
- Bayesian filtering:

$$P(x_t \mid y_1,\ldots,y_t) = \alpha\, P(y_t \mid x_t)\, P(x_t \mid y_1,\ldots,y_{t-1})$$
$$= \alpha\, P(y_t \mid x_t) \int_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_1,\ldots,y_{t-1}) dx_{t-1}$$

- Because everything is Gaussian and linear, posterior is also Gaussian: $P(x_t \mid y_1,\ldots,y_t) = N(\hat{x}_t, S_t)$

Kalman Filter Equations

- **Prediction** from one time step to next:

$\bar{x}_t = A\hat{x}_{t-1}$ Mean

$M_t = AS_{t-1}A^T + Q$ Covariance

- **Correction** upon measuring y_t:

$\hat{x}_t = \bar{x}_t + K_t(y_t - B\bar{x}_t)$ Mean

$S_t = (B^T R^{-1} B + M_t^{-1})^{-1}$ Covariance

- "Kalman" gain $K_t = S_t B^T R^{-1}$ dictates how much weight to give to prediction error $(y_t - B\bar{x}_t)$ versus prediction $\bar{x}_t$

Kalman Filter: Prediction and Correction Cycle