Brain-Computer Interfaces: Therapeutics and Basic Science

Aaron Batista
aaron.batista@pitt.edu

30 September 2021

BioEngineering

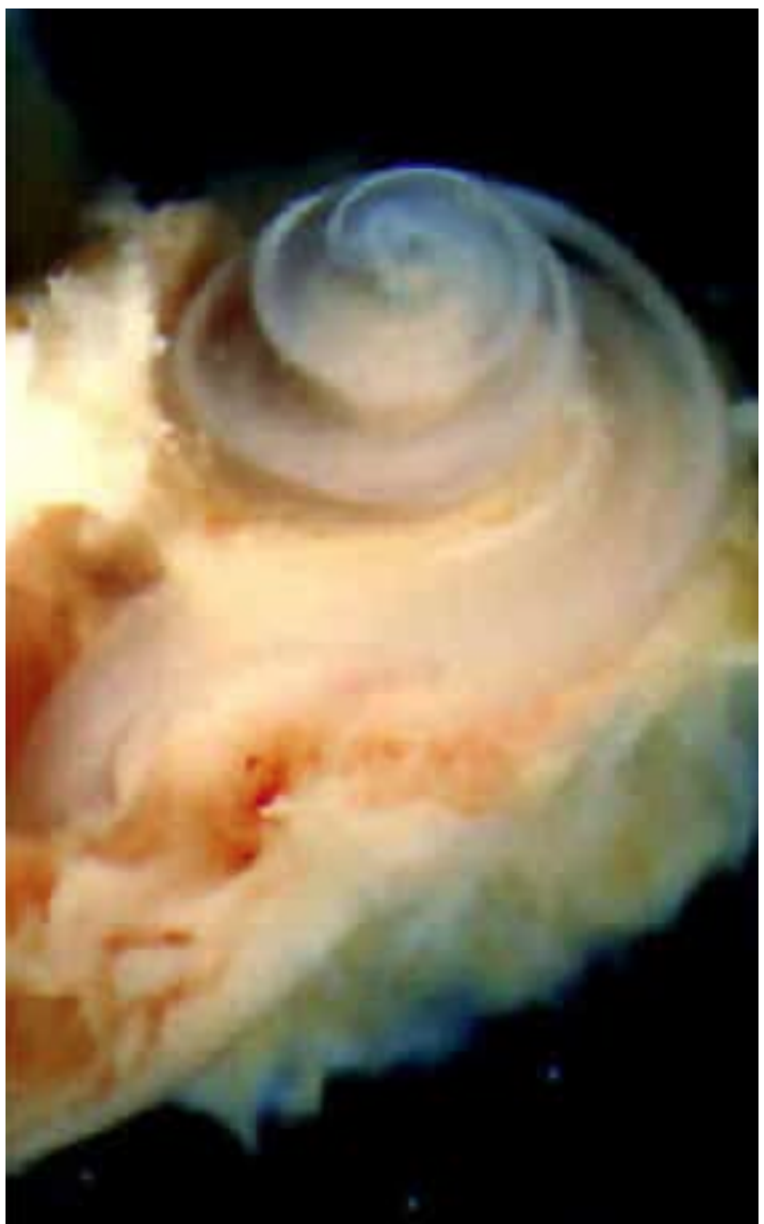
CNBC
Center for the Neural Basis of Cognition

Neural Engineering:

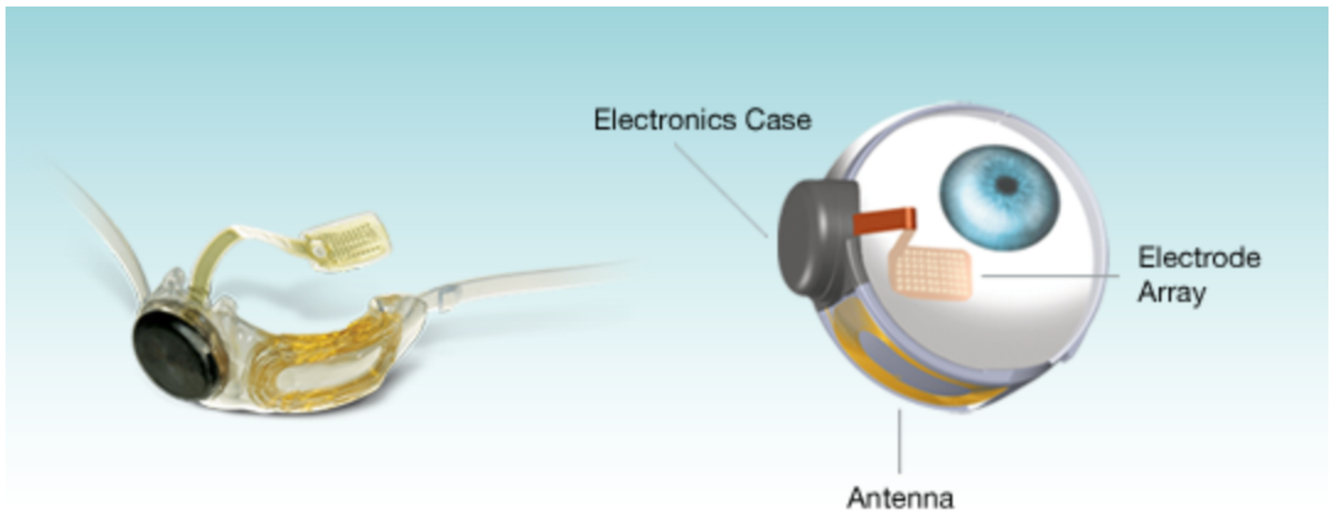
Electrical interventions for restoring sensory, motor, and cognitive function

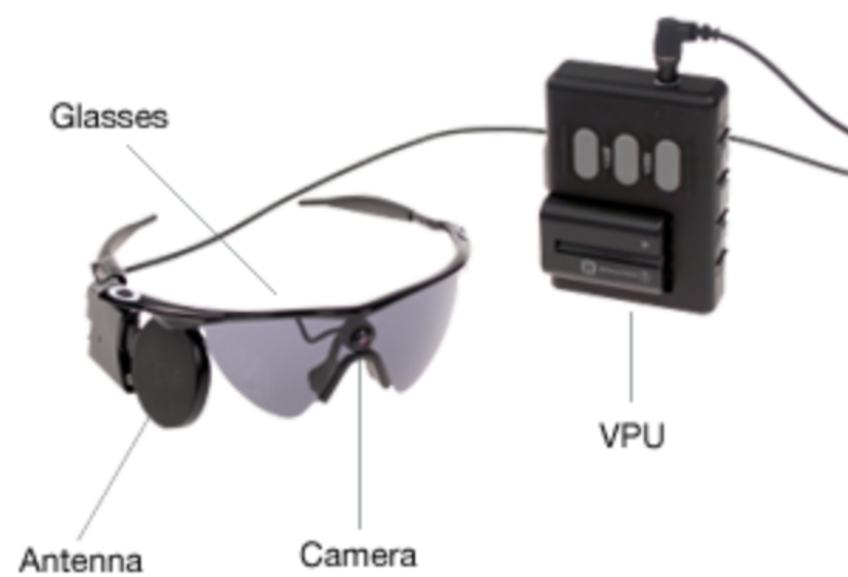
1) Cochlear Implants

Cochlea



2) Retinal Prosthetics





www.2-sight.com

3) Deep Brain Stimulation (DBS) for dystonia and Parkinson's Disease

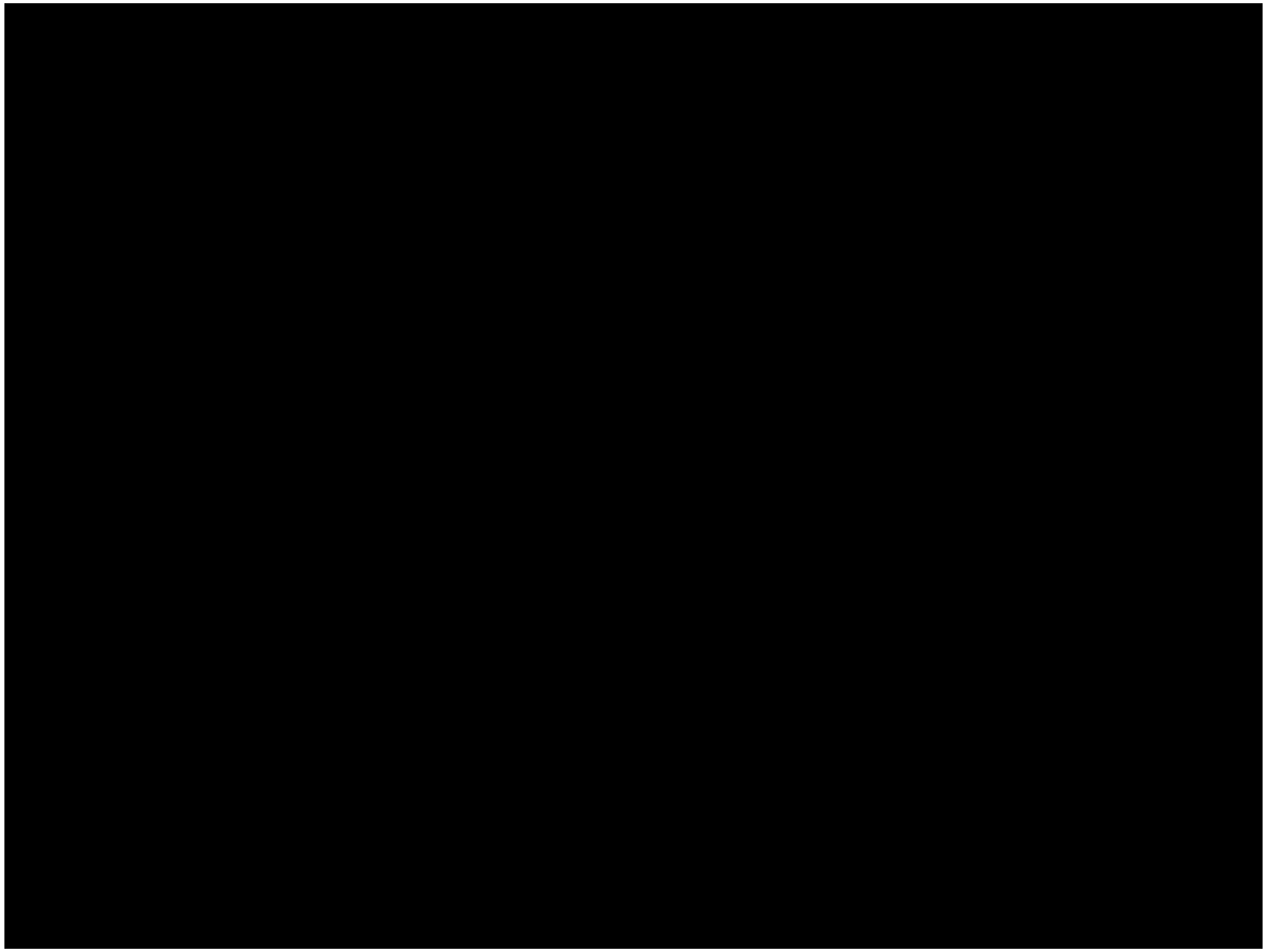
DBS for dystonia

Uploaded on Aug 18, 2010
My name is Kym and I have severe generalized tardive dystonia. I have a DEEP BRAIN STIMULATOR . This is me at my neurologist office showing you what my DBS does for me. I would love to find a cure for this disorder! PLEASE HELP!

https://www.youtube.com/watch?v=M53ov7Fifgo

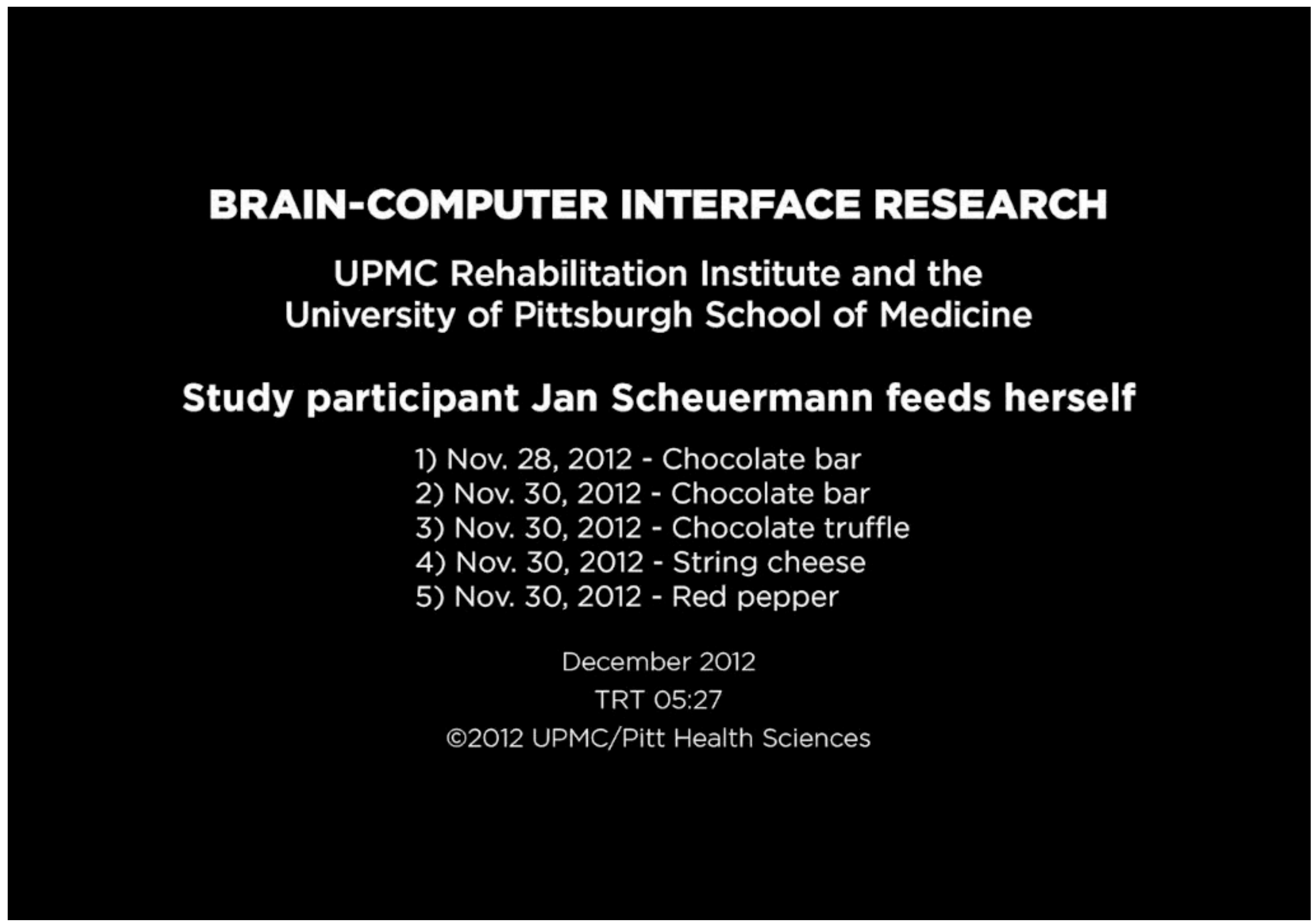
4) Brain-Computer Interfaces (BCI) to restore motor function

BCI to restore motor function



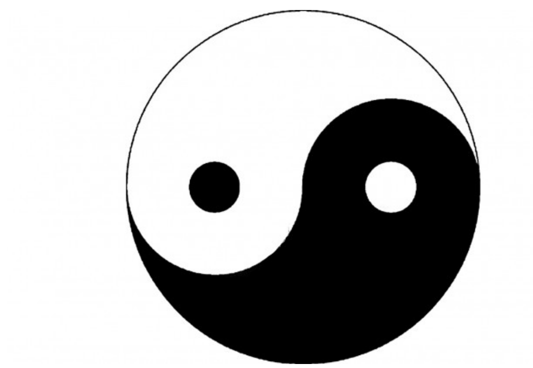
Hochberg et al, 2006

BCI to restore motor function



Collinger et al, 2013

Outline



- Neurophysiology of motor control
- Brain-Computer Interface algorithms
- Population methods in basic neuroscience
- New directions in BCI therapeutics

Outline

- Neurophysiology of motor control
  - Behavior
  - Theories
  - Neurophysiology

Why is reaching difficult?

Why is reaching difficult?

- It takes years to learn.

- Redundant degrees of freedom
  - And yet, movements are stereotyped
  - You must optimize something, but what?

- Muscles are complex
  - Same input, different responses, and it depends on the muscle's position, velocity, and force.

- Spinal reflexes
  - Hierarchical control

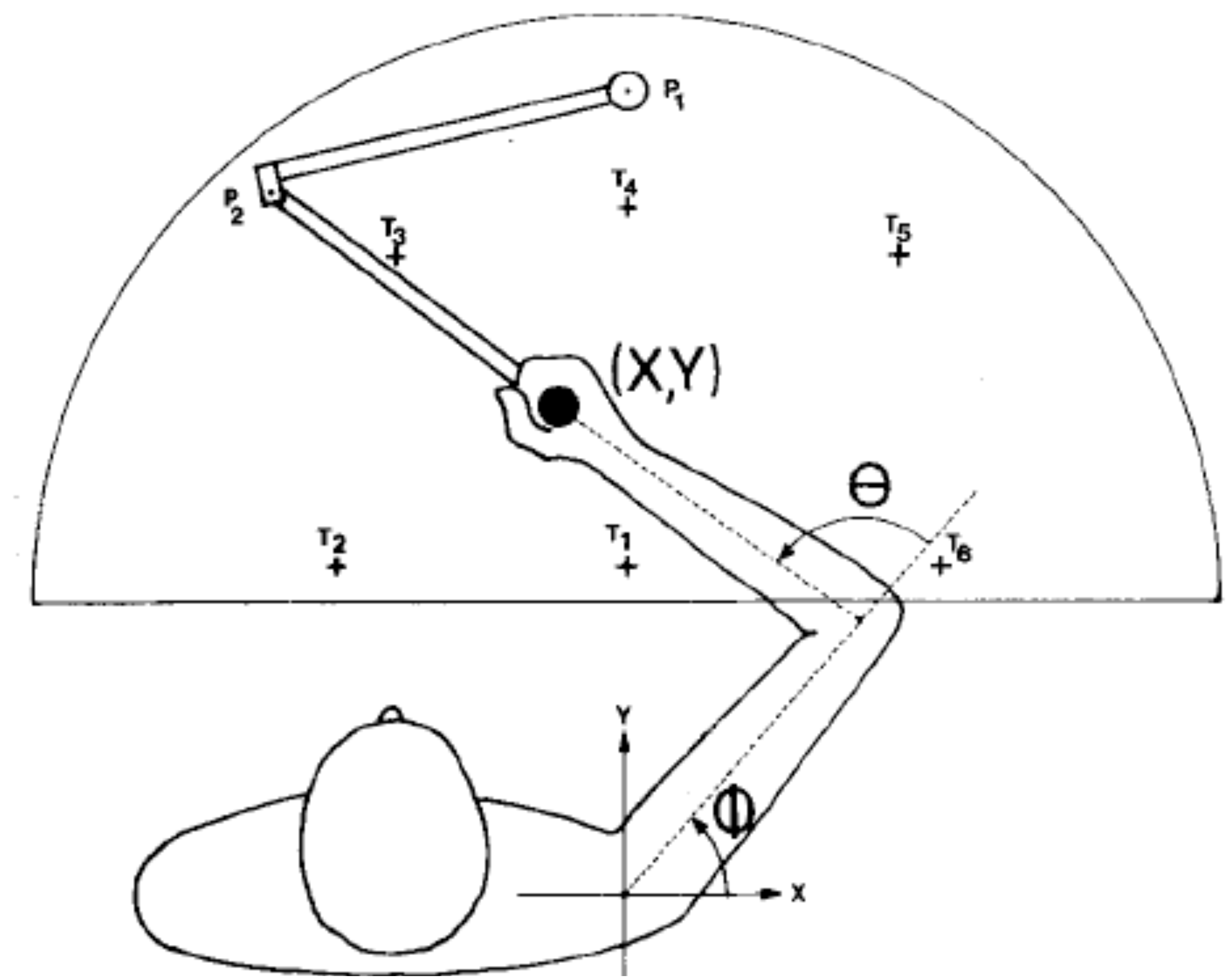
- Inputs and outputs are in different formats.
  - Visual-motor reference frame transformation

- You have to pre-plan movements.
  - You can't just rely on feedback; it's too slow.
  - Nervous system may construct an "internal model" the limbs and the environment.

    [Aaron, prove it to them!]

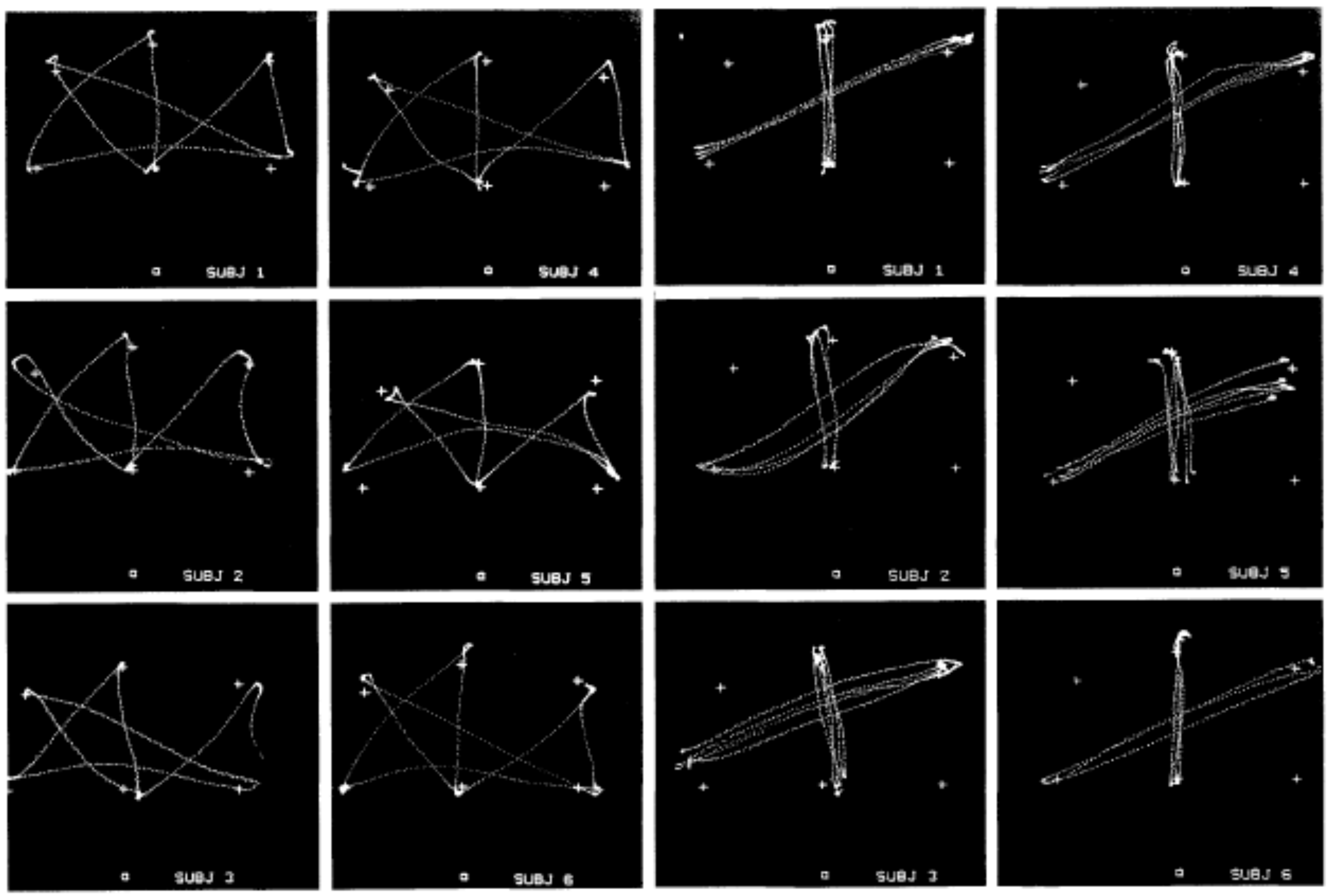
What do we need to explain?

Properties of Arm Movements



Morasso, "Spatial Control of Arm Movements", *Exp Brain Res* 42:223 (1981)

Reaches are straight in visual coordinates…



Morasso, "Spatial Control of Arm Movements", *Exp Brain Res* 42:223 (1981)

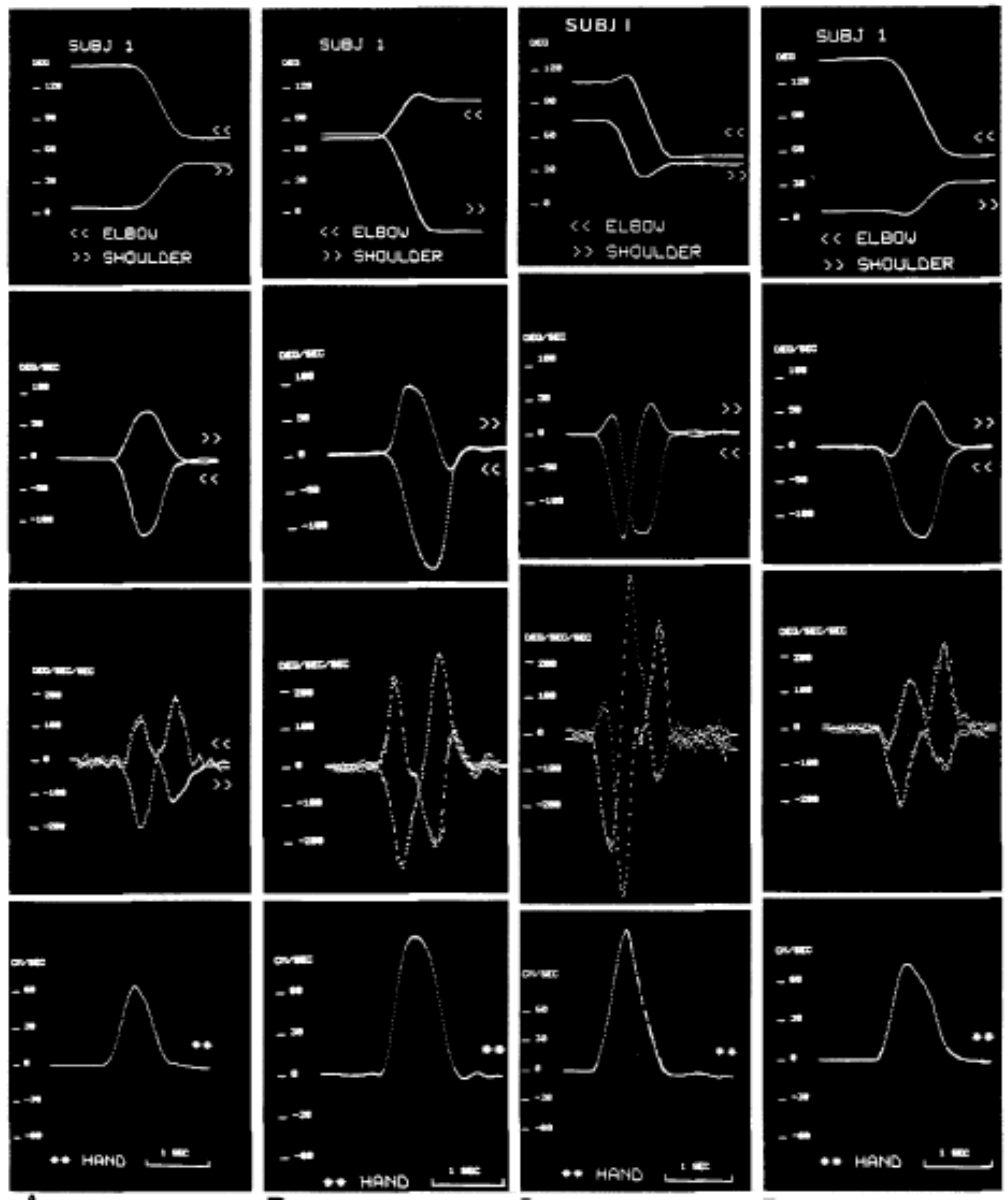
… but very complex in joint coordinates

Joint angles

Joint velocity

Joint accel.

Hand velocity



"Bell-shaped velocity profile"

Morasso, "Spatial Control of Arm Movements", *Exp Brain Res* 42:223 (1981)

Other regularities

- Fitts' Law: speed/accuracy tradeoff
- "2/3 power law": Speed is related to curvature
- Scale and effector invariance (e.g. handwriting)
- Repeated movements are very similar
- Movements are smooth

How do we explain regularities?

An Early Theory of Motor Control: Minimum Jerk

Smooth movements to minimize jerk

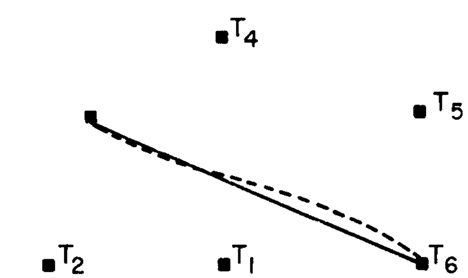
(Circa 1985)

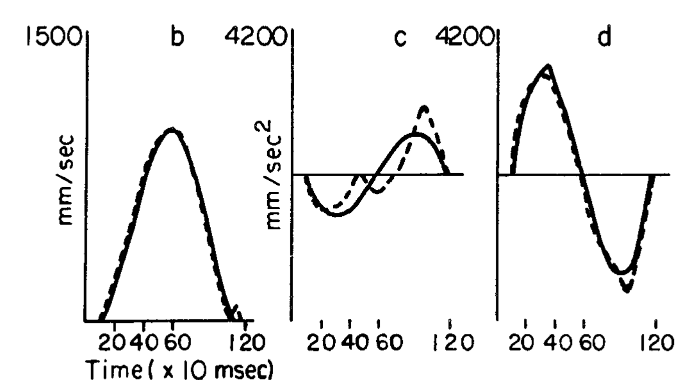
The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model[1]

TAMAR FLASH*,[2] AND NEVILLE HOGAN‡[3]

$$C = \frac{1}{2}\int_0^{t_f}\left(\left(\frac{d^3x}{dt^3}\right)^2 + \left(\frac{d^3y}{dt^3}\right)^2\right)dt$$

Coordination is modeled mathematically by defining an objective function, a measure of performance for any possible movement. The unique trajectory which yields the best performance is determined using dynamic optimization theory. In the work presented here, the objective function is the square of the magnitude of jerk (rate of change of acceleration) of the hand integrated over the entire movement. This is equivalent to assuming that a major goal of motor coordination is the production of the smoothest possible movement of the hand.





Newer Theories of Motor Control

- Signal-Dependent Noise

  (Wolpert and Harris)

- Optimal Feedback Control

  (Todorov and Jordan, and Steve Scott)

- Internal Models

  (Masao Ito, and many others)

Signal-Dependent Noise

Signal-dependent noise determines motor planning

Harris and Wolpert, *Nature* 394:780 (1998)

Smooth, "bell-shaped" reaches are made by a simple model with reasonable assumptions and objectives:

(safe) Assumptions:

- The goal is to make accurate movements.
- Neural control signals are Poisson: the noise scales with the signal.
- Noise accumulates over the duration of the movement.

A simple arm model hits goals under these constraints:

For a given movement duration, the neural command minimizes error.

-or-

For a given error tolerance, the neural command maximizes the speed.

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don’t correct errors that don’t hurt you.*

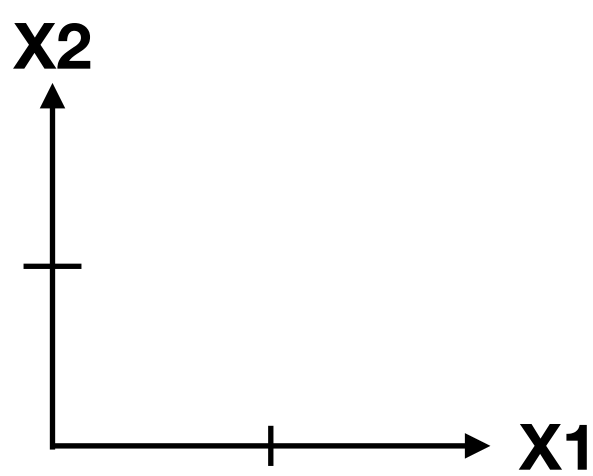
Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don’t correct errors that don’t hurt you.*

Goal: $X1+X2 = 2$

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*

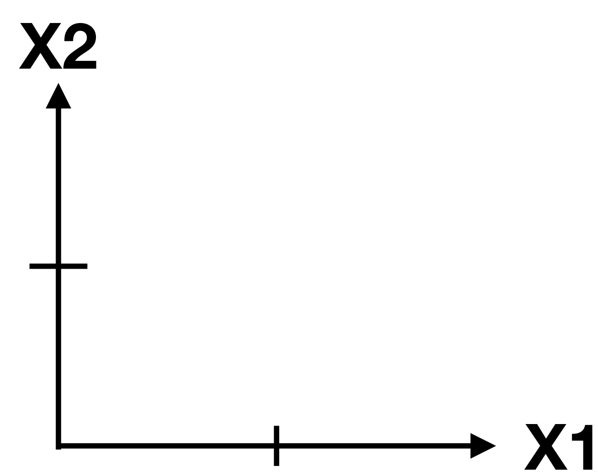


Two dimensional input to one dimensional output

Goal: X1+X2 = 2

Optimal Feedback Control

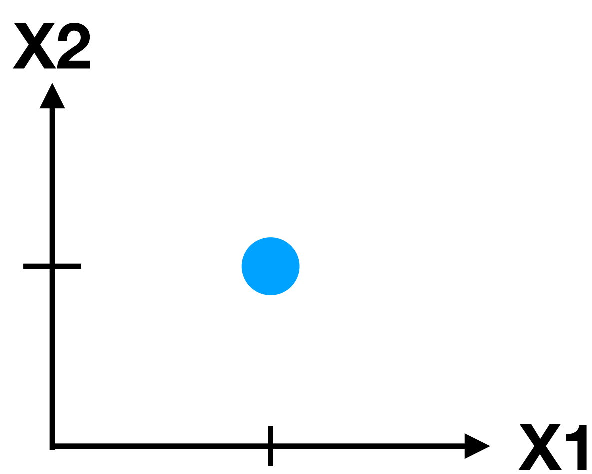
- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*



Goal: X1+X2 = 2

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*

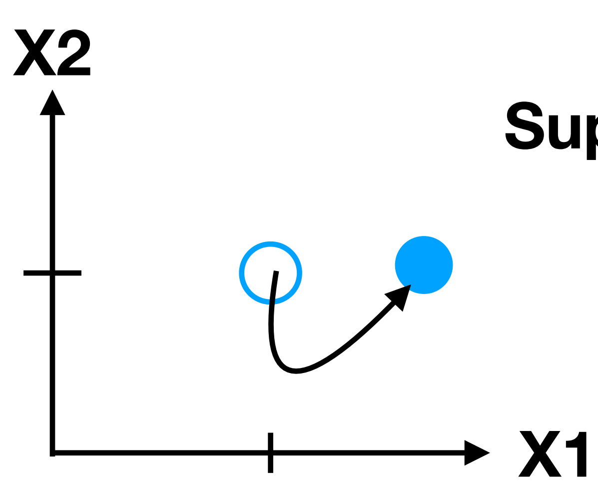


Easy!
X1=1
X2=1
Solved.

Goal: X1+X2 = 2

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don’t correct errors that don’t hurt you.*

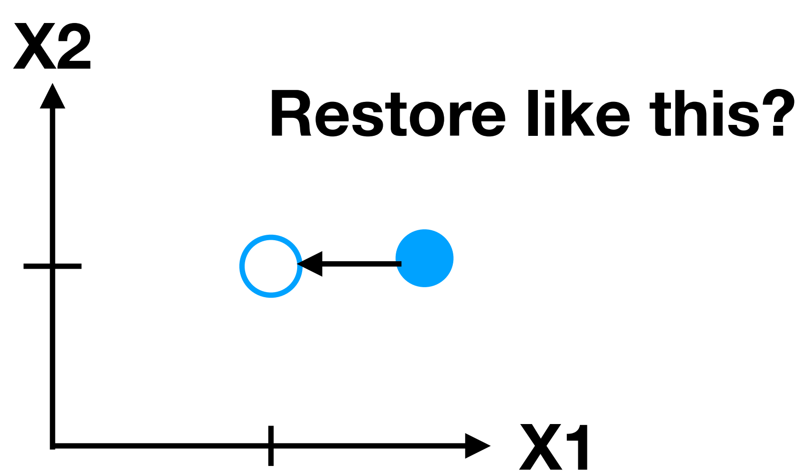


Suppose the system is perturbed.

Goal: X1+X2 = 2

Optimal Feedback Control

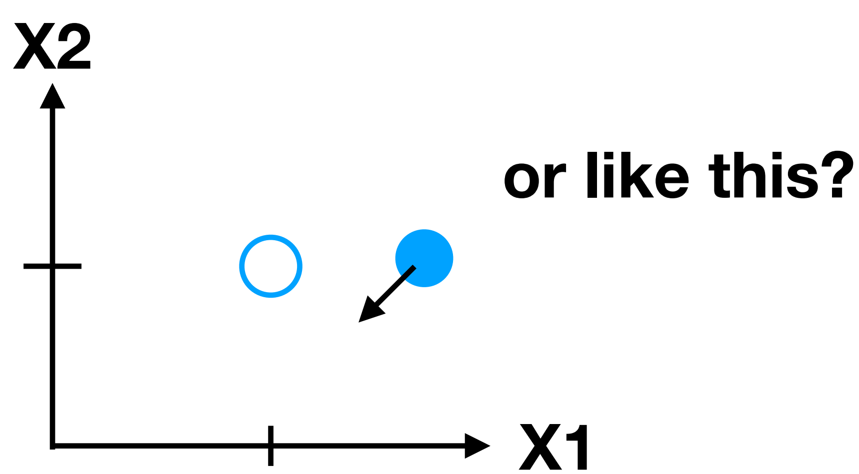
- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*



Goal: X1+X2 = 2

Optimal Feedback Control

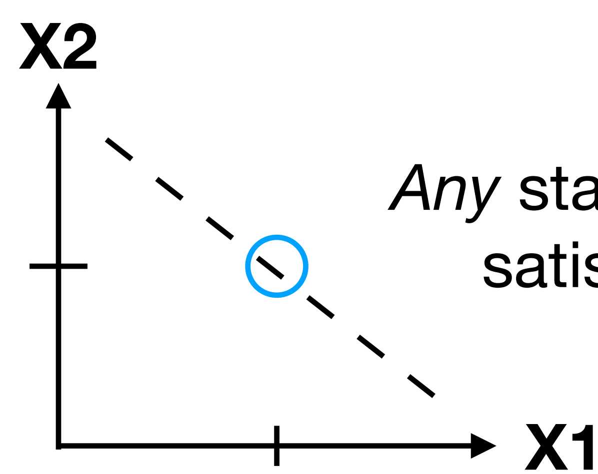
- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*



Goal: X1+X2 = 2

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don’t correct errors that don’t hurt you.*

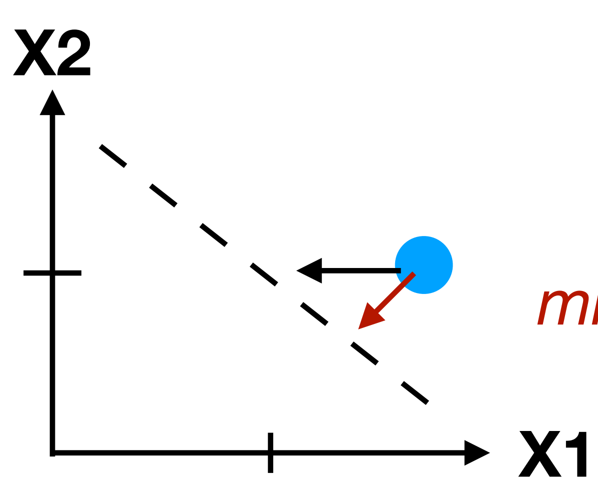


Any state along this line satisfies the goal.

Goal: X1+X2 = 2

Optimal Feedback Control

- A multiple-input, multiple-output system may have redundancies.
- When such a system is noisy, the redundant space can be exploited.
- *Don't correct errors that don't hurt you.*



This one is the *minimum intervention* solution.

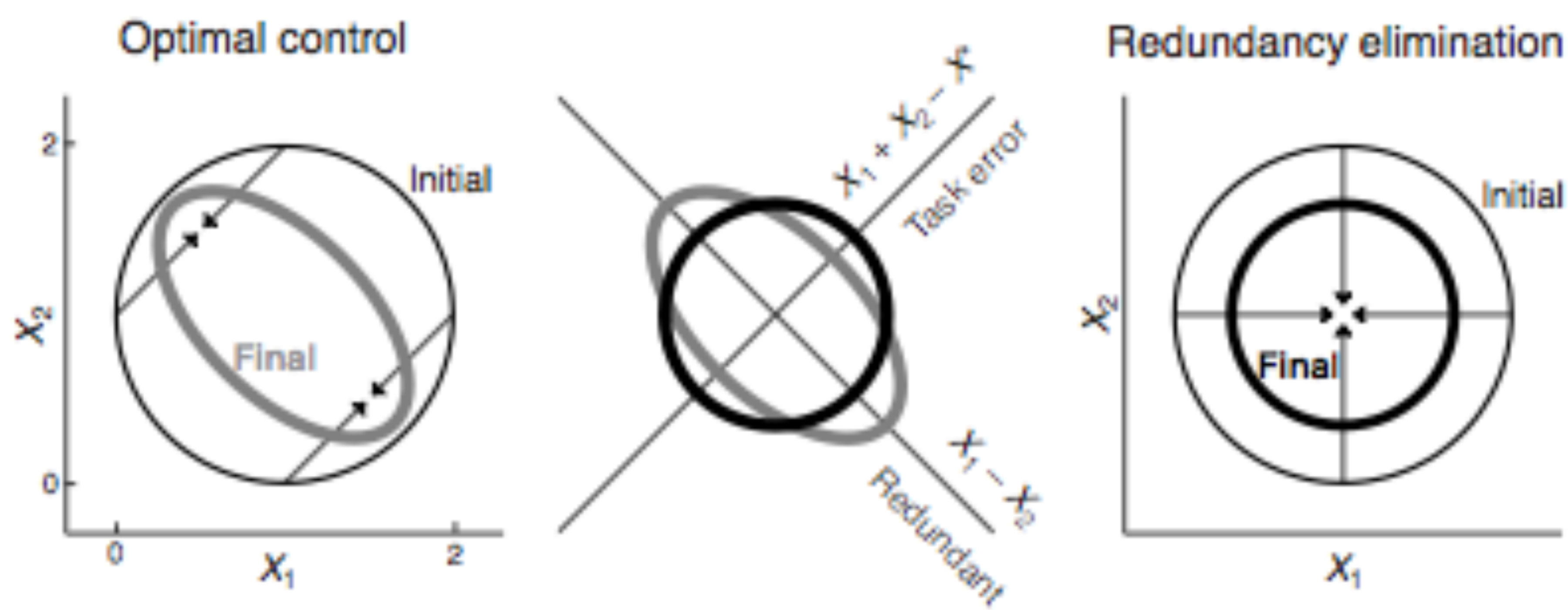
Goal: X1+X2 = 2

Optimal feedback control as a theory of motor coordination

Emanuel Todorov[1] and Michael I. Jordan[2]

"Minimal Intervention Principle"

(Don't correct variability that doesn't matter)

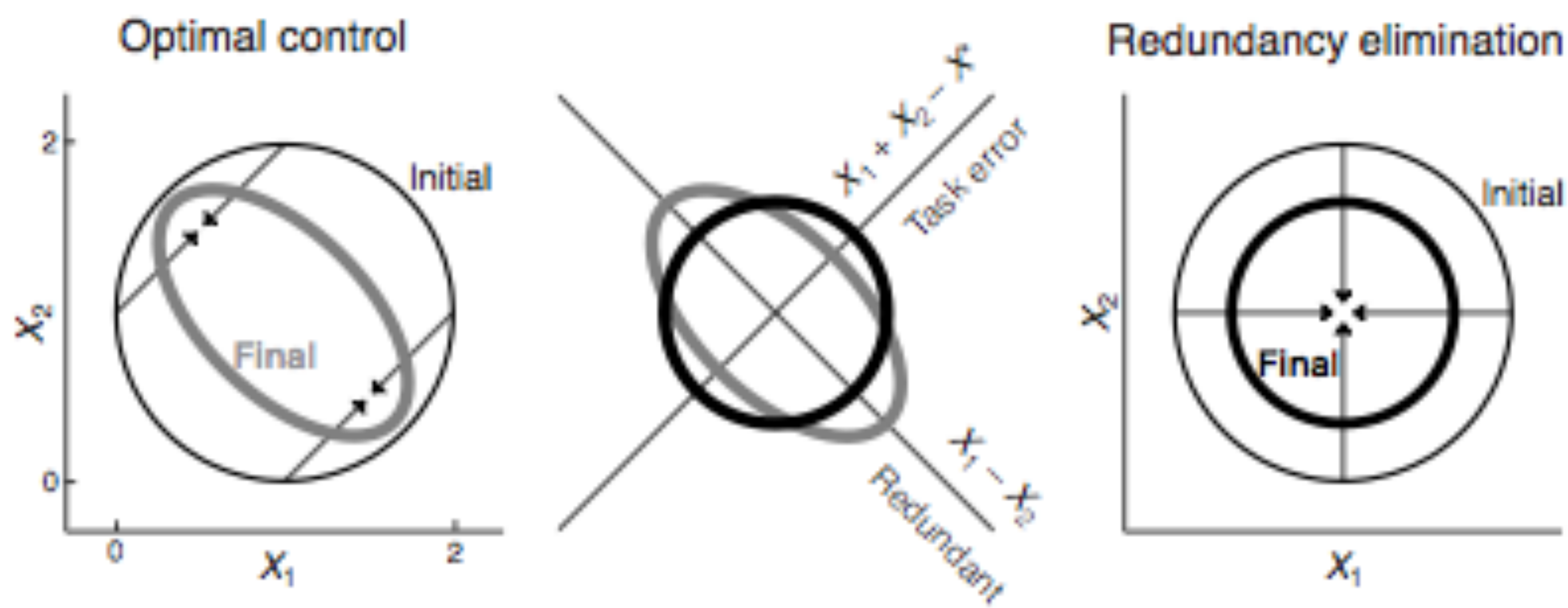


Goal: X1+X2 = 2

Optimal feedback control as a theory of motor coordination

Emanuel Todorov[1] and Michael I. Jordan[2]

"Minimal Intervention Principle"



exploit reduncay

Benefit of OFC:

- Smaller command signals

Hallmarks of an OFC solution:

- Structured variability

Internal Models

- If feedback were fast, you could just start moving, and correct as you go.
- Instead, the brain builds an internal representation of the body, and the environment ("forward model").
- This allows you to plan movements well, and to anticipate the consequences of action.

  [Aaron, did you make them do the demo?]
- Ample behavioral evidence, but still very little physiological demonstrations.
- (For completeness, note that there are also inverse models that convert goals into motor commands. These are "inverse models".)

Theories of motor control:

- Many possible reaches achieve the same goal. This is an *advantage* for the system, but it raises a challenge for us as we try to understand it - *How is a movement selected?*

- Some theories:

  - Reaches are selected to minimize jerk.

  - Reaches try to optimize speed and accuracy (and that results in smoothness.)

  - Variability that can’t hurt performance is ignored.

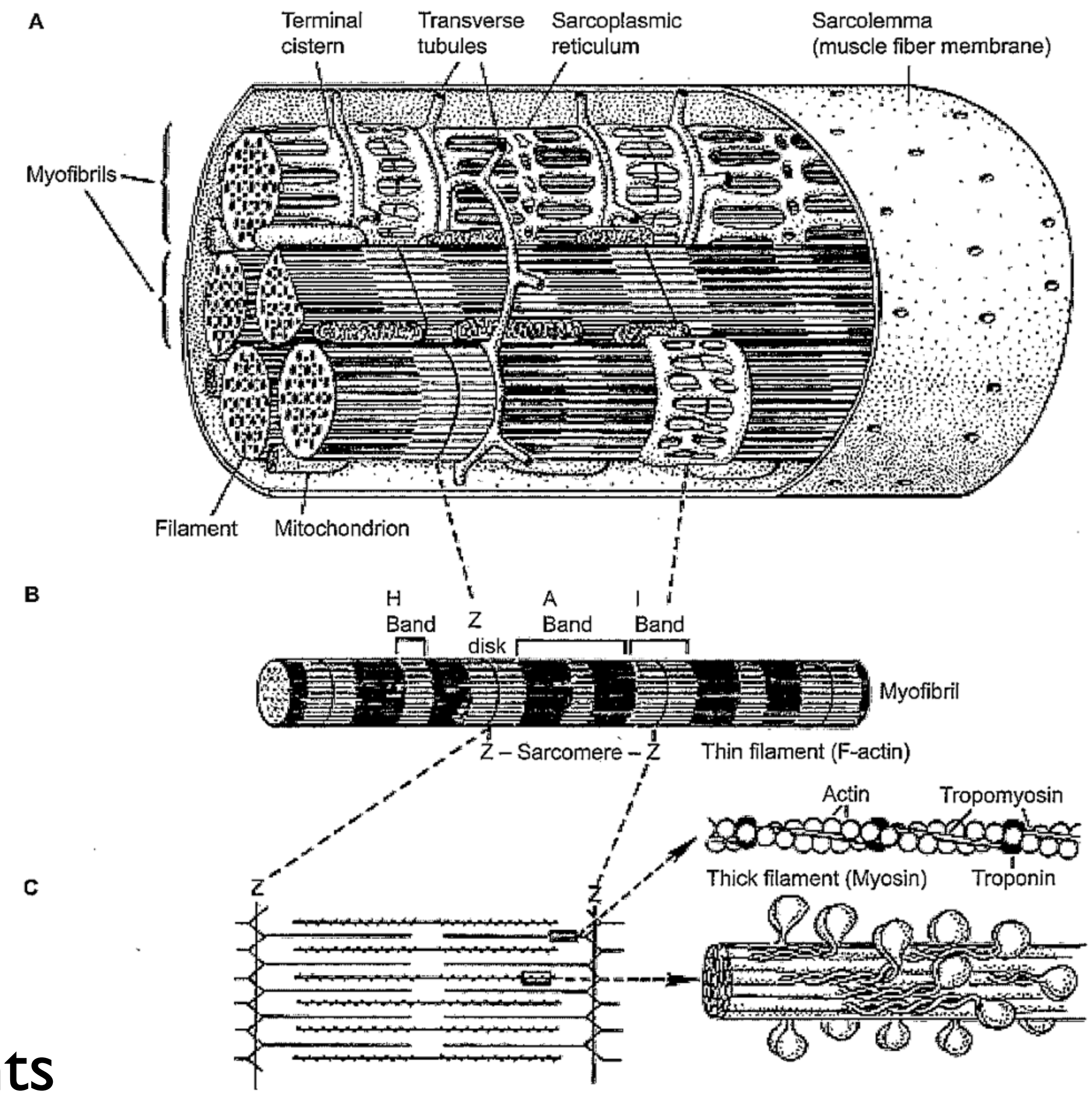
  - Internal models (of the world, of the body) are created, to “try out” many plans before executing one.

Non-ideal aspects of muscles

1) Force depends on length, and velocity
2) Sluggish
3) They only work in one direction

   Antagonist arrangement.

Muscle physiology

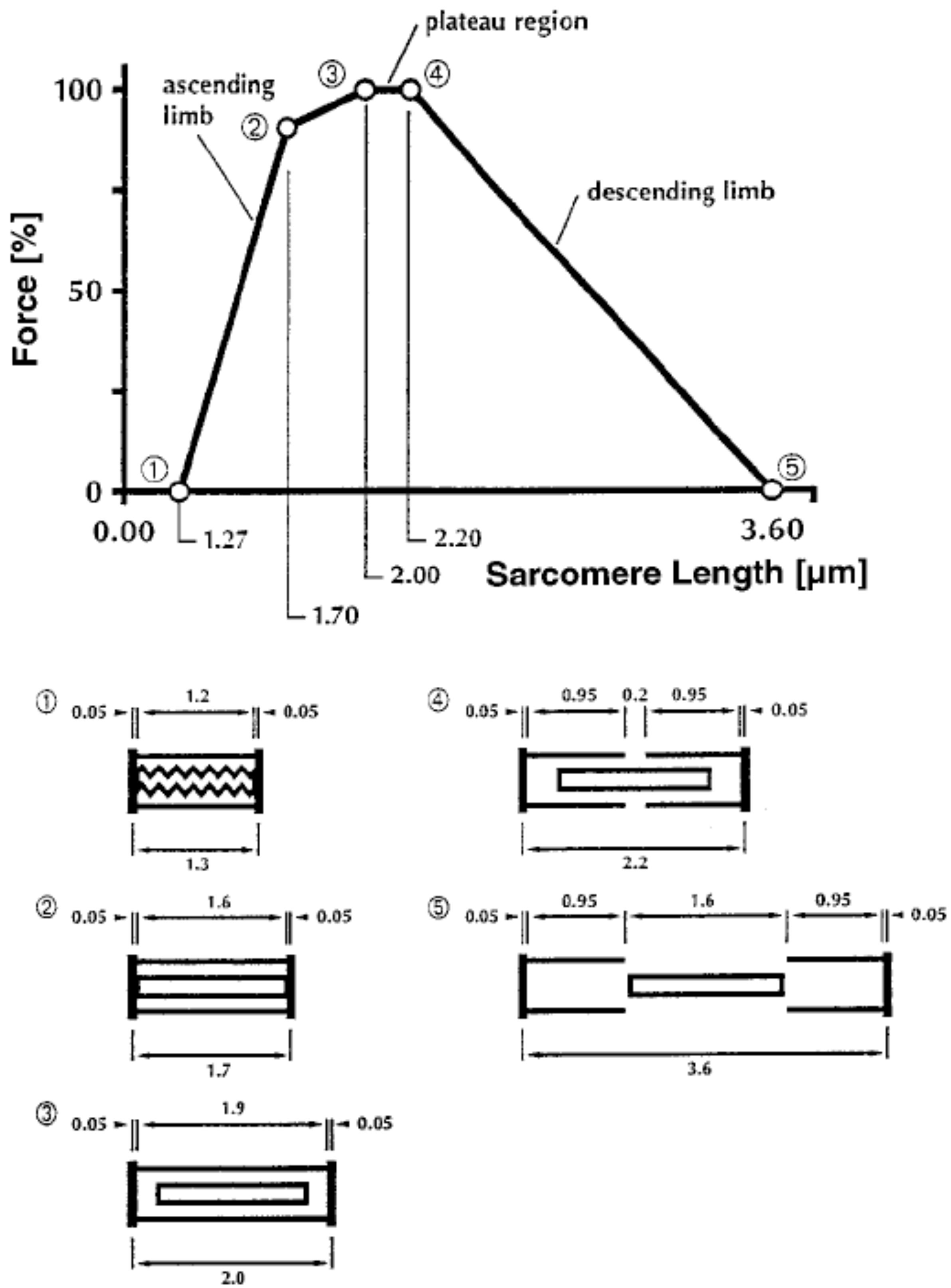


Sliding filaments

Cross bridges

Active properties

Force depends on length

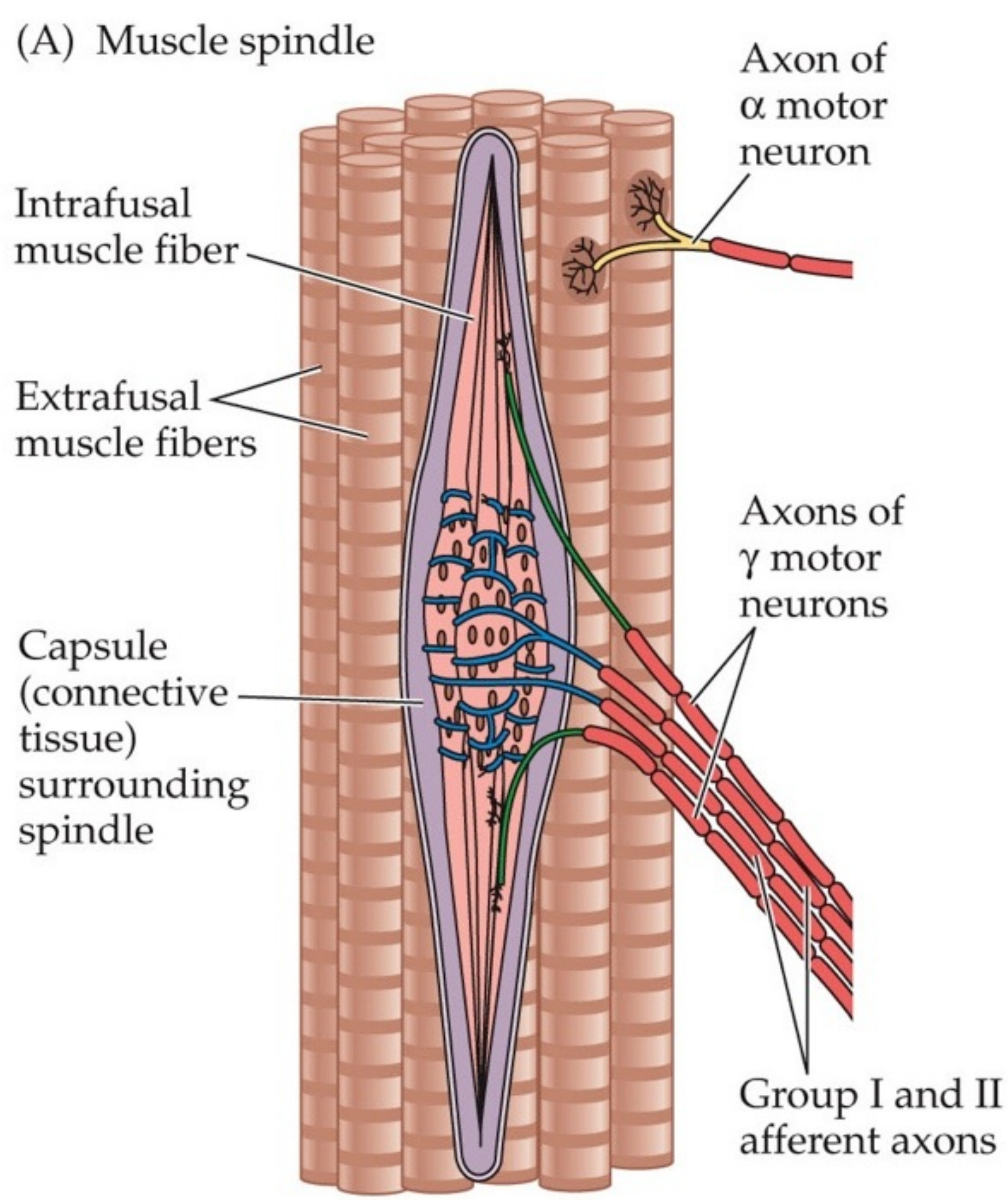


Experiments conducted on single muscle fibers.

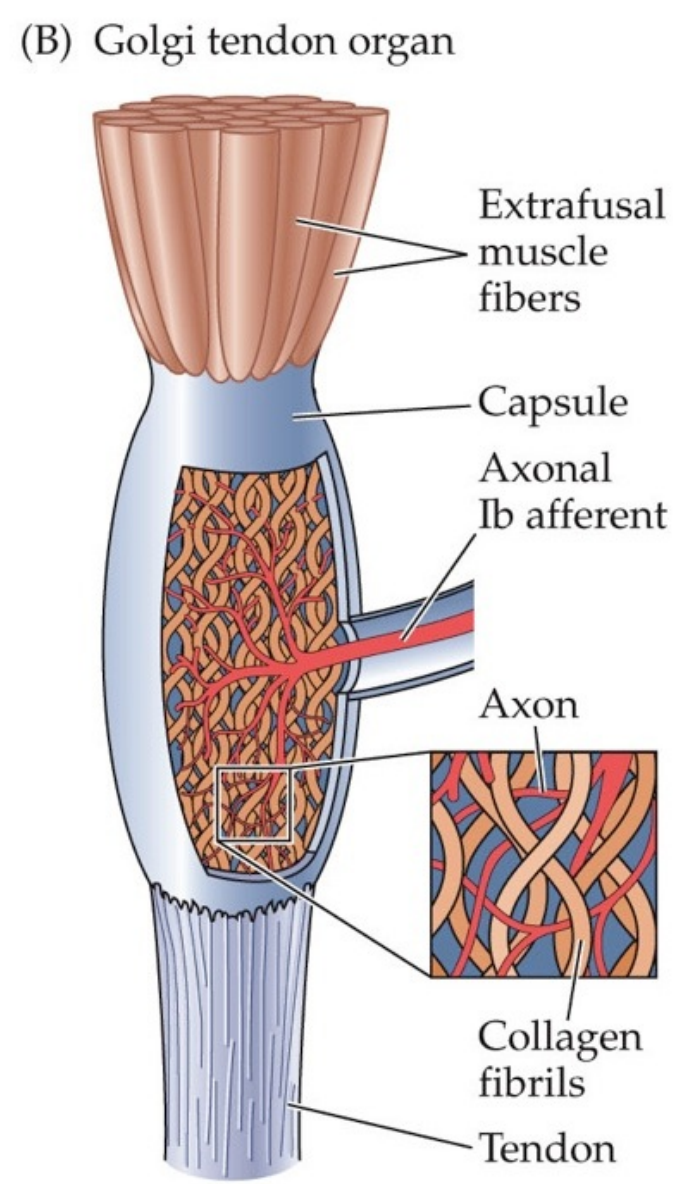
Proprioceptors

Muscle Spindle

Golgi tendon organ



NEUROSCIENCE, Fourth Edition, Figure 9.7

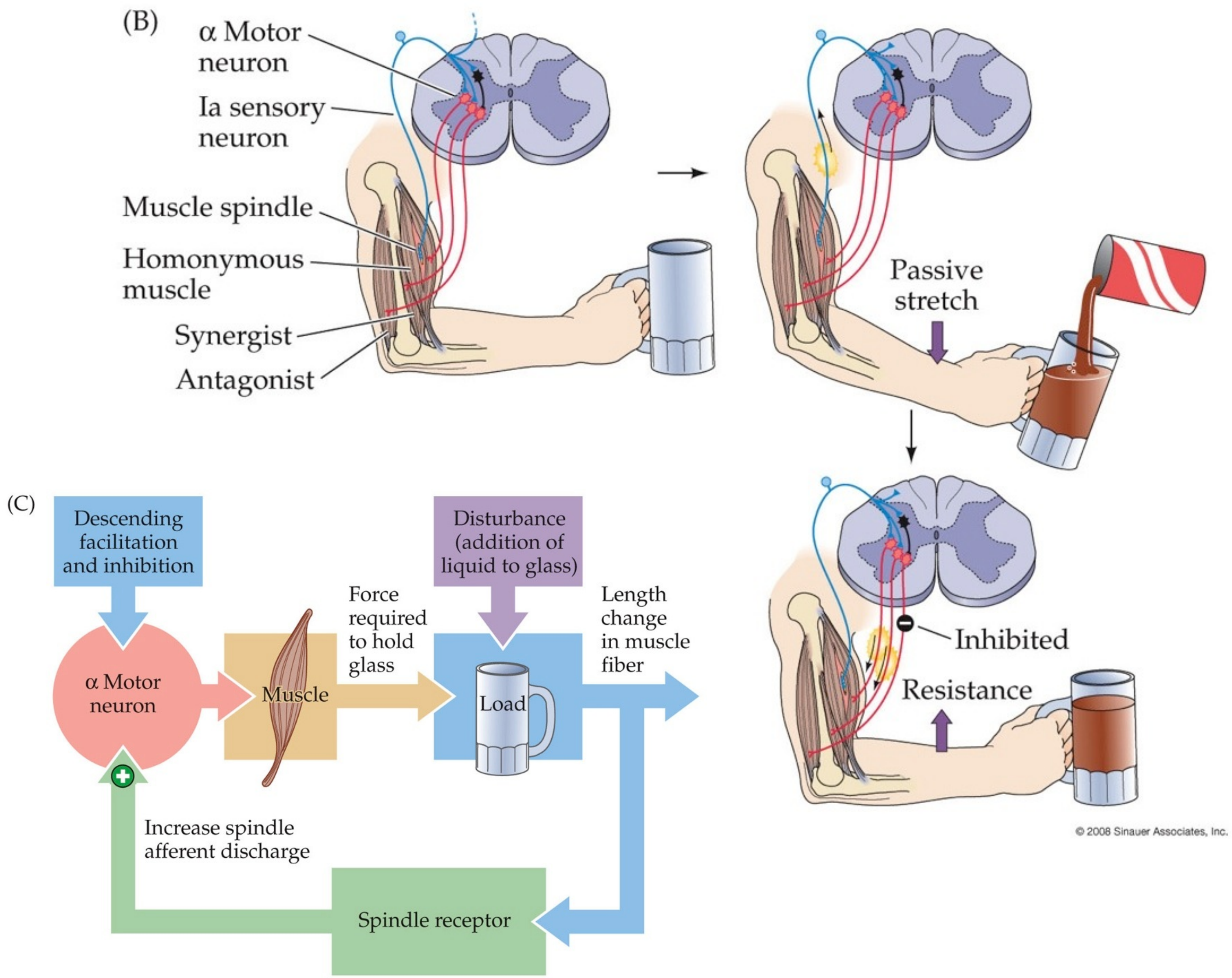


In series with muscle fibers

In parallel with muscle fibers

Feedback loops
Spindles regulate posture

"Myotactic reflex"



NEUROSCIENCE, Fourth Edition, Figure 16.10 (Part 3)

Today's Menu

- Neurophysiology of motor control
  - Behavior
  - Theories
  - Neurophysiology

Given all this complexity, is the goal of the primary motor cortex easier, or harder?

Given all this complexity, is the goal of the primary motor cortex easier, or harder?

How many of the details of arm movements are reflected in the activity of M1?
(and, *which* details are evident there?)

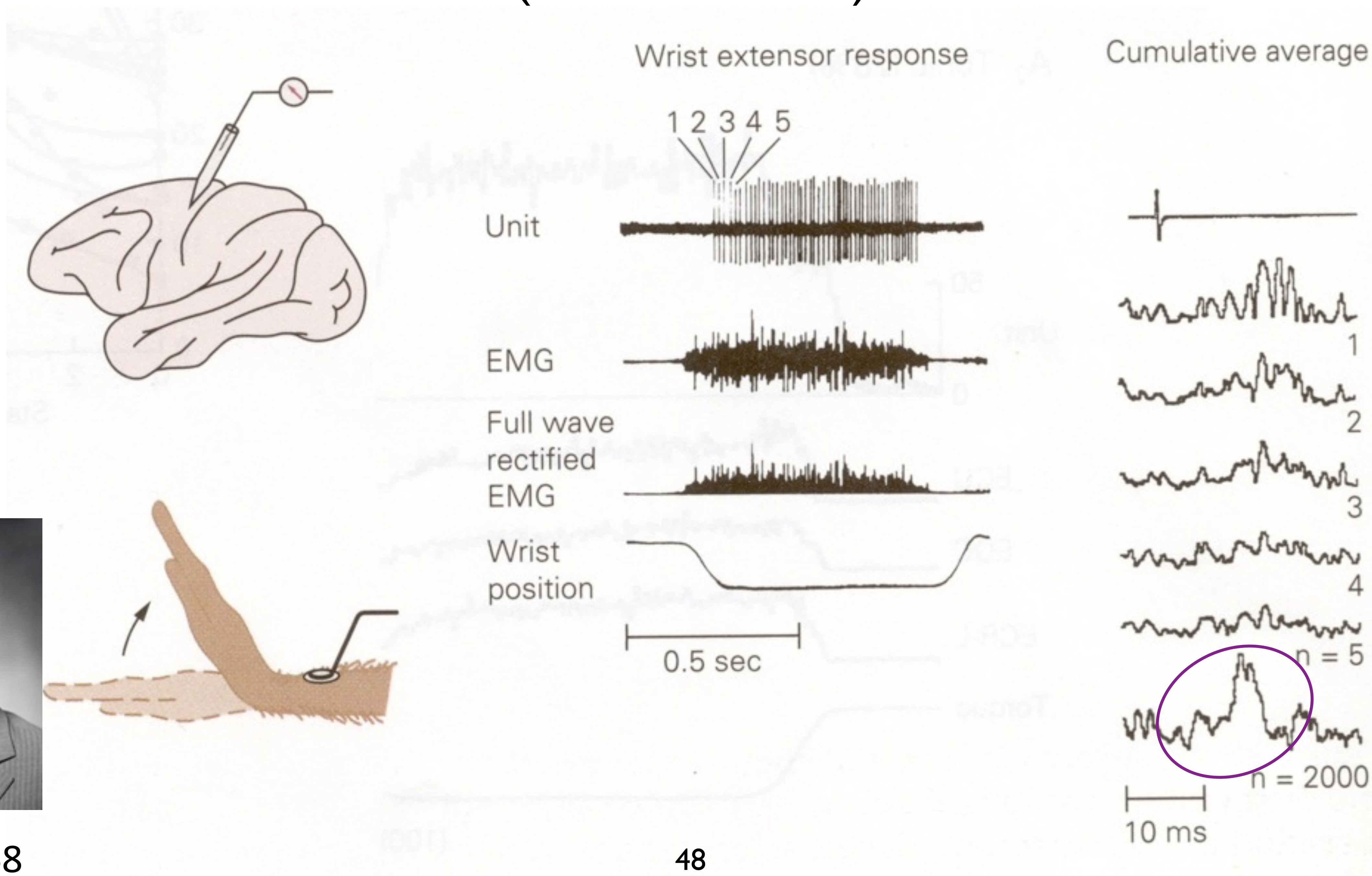
Does M1 generate commands for…

Muscles or Movements?

How can we approach this problem?

M1 codes *Muscles*: Evidence

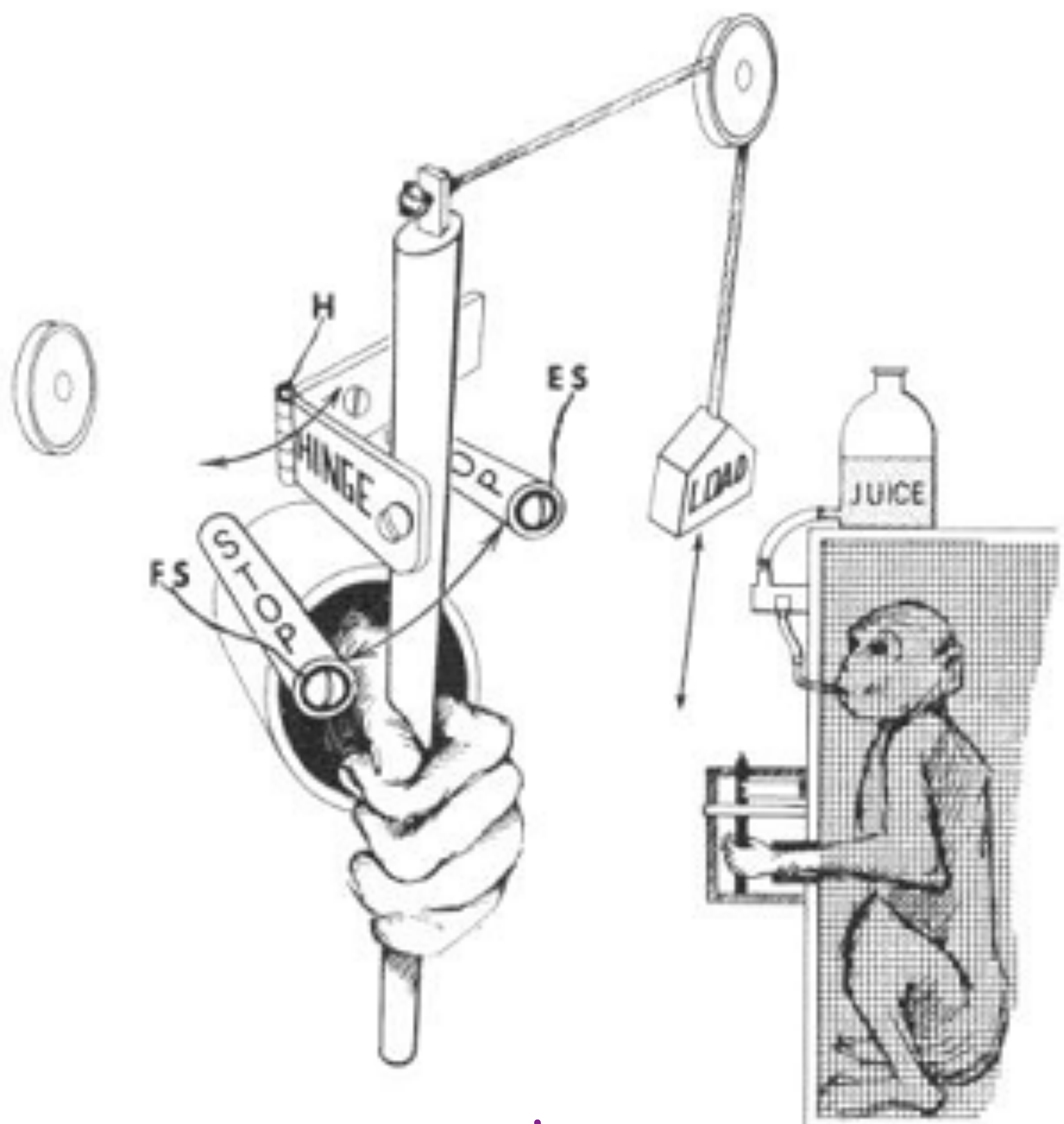
Method: Spike-triggered averaging to identify neurons that project directly to the spinal cord (Evarts, 1968)



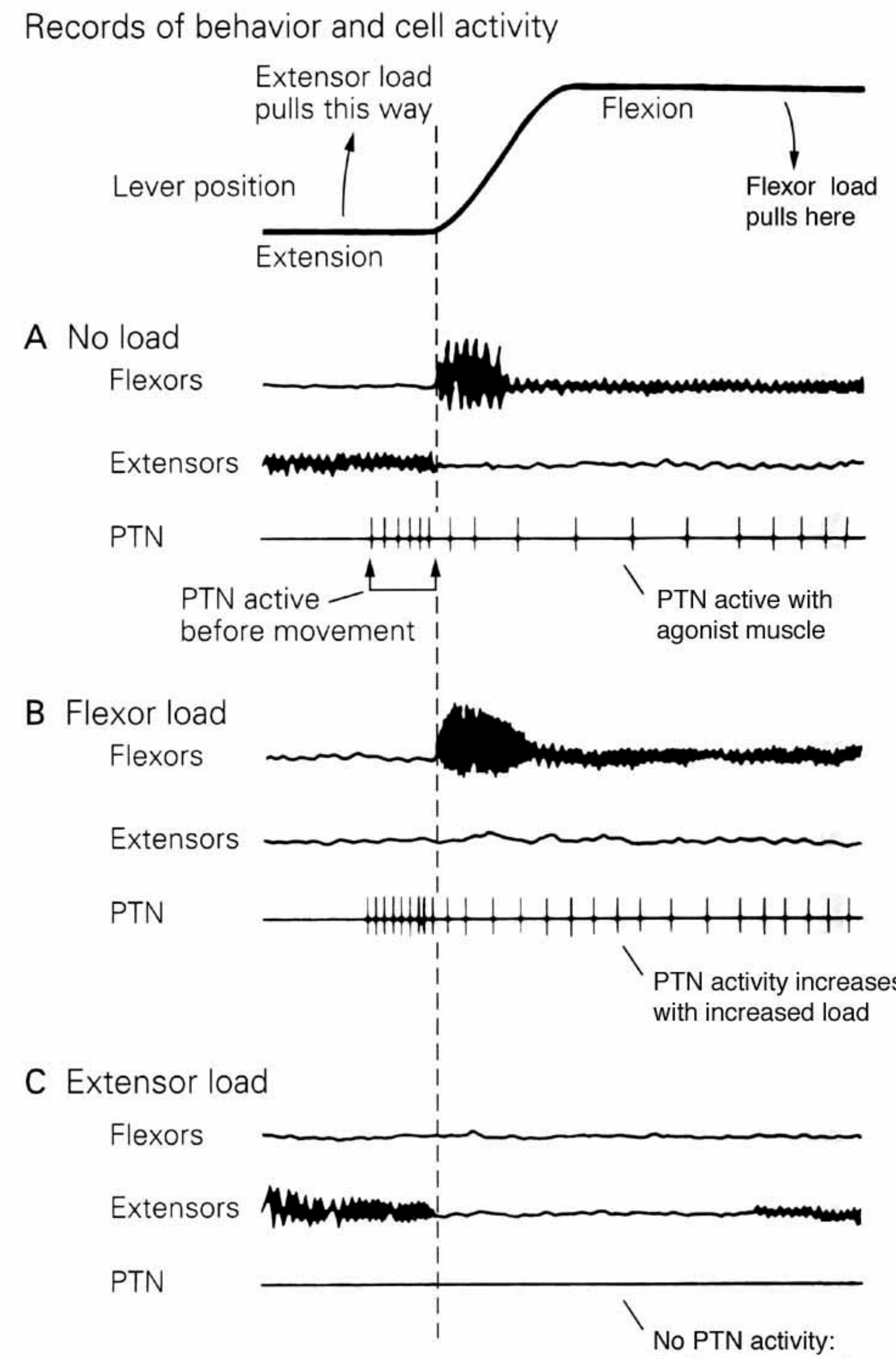
EMG

Evarts 1968

M1 codes *Muscles*: Evidence



w/ or w/out load

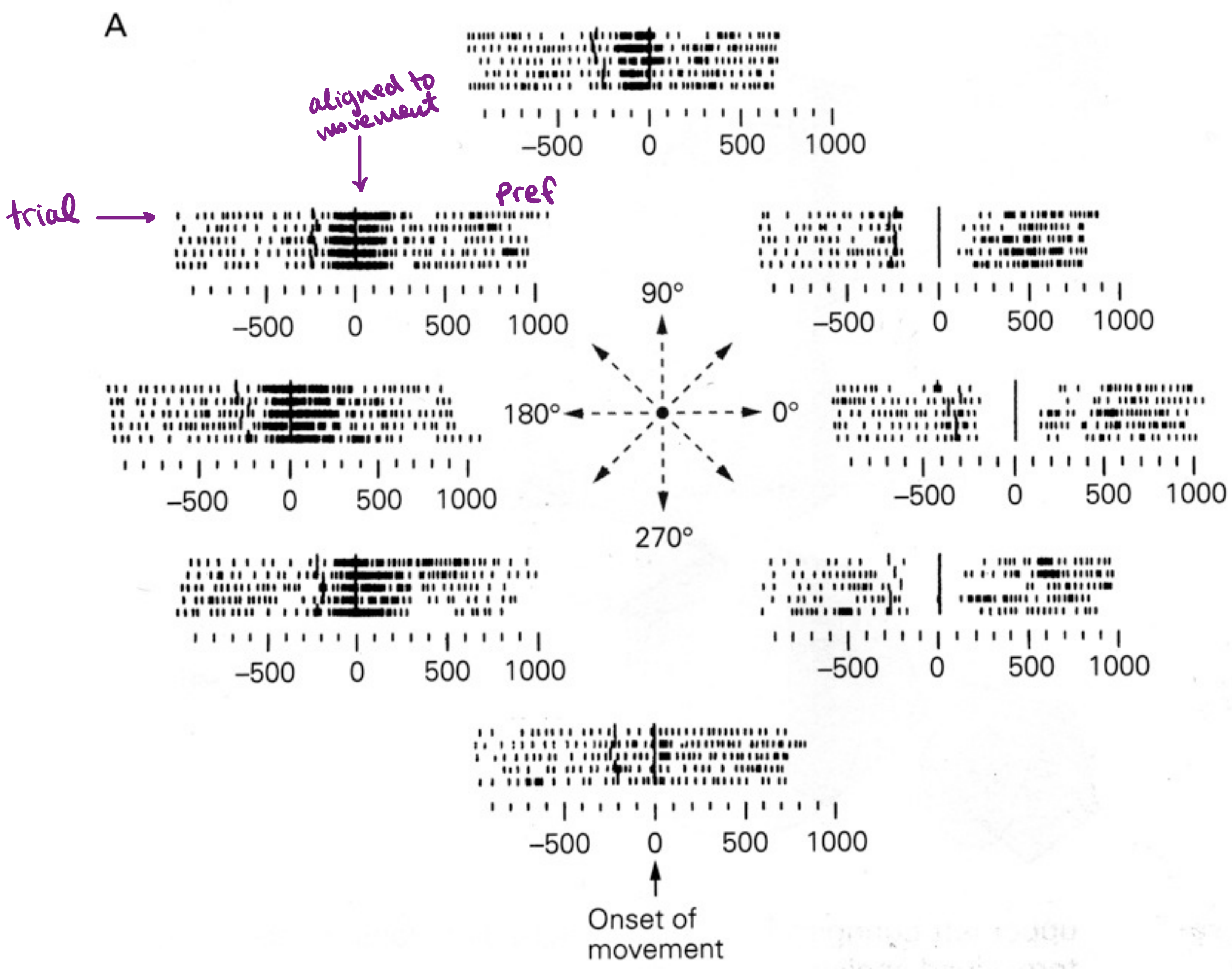


Kandel, Schwartz and Jessell, 4th Ed.
Figure 38-10B

Evarts 1968

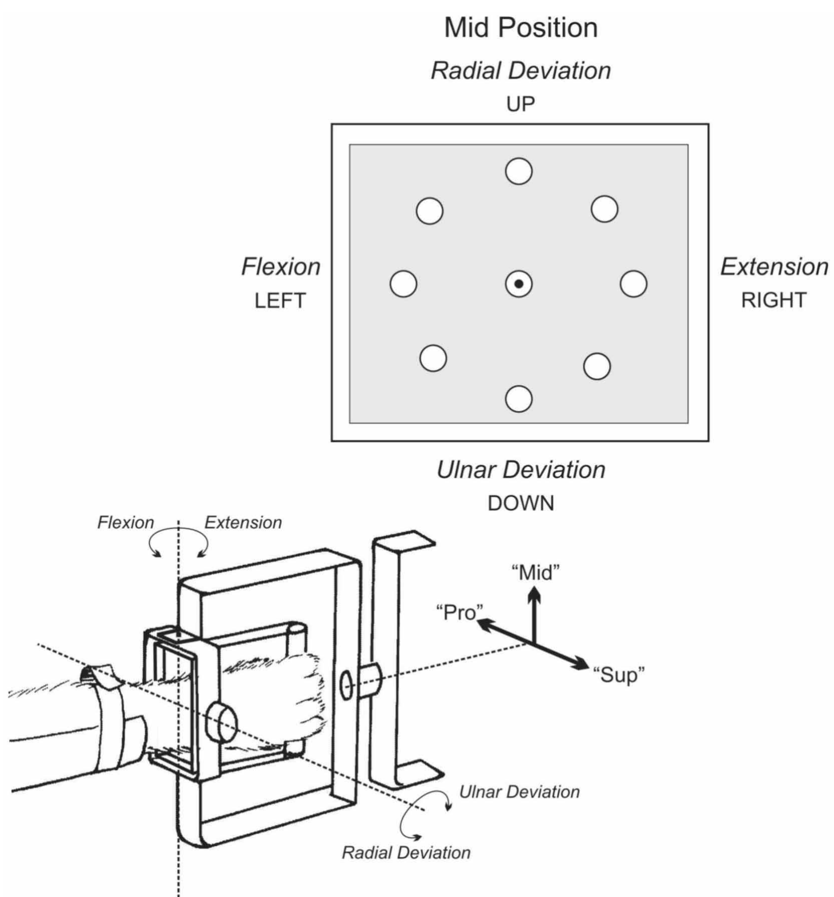
M1 codes *Movements*: Evidence

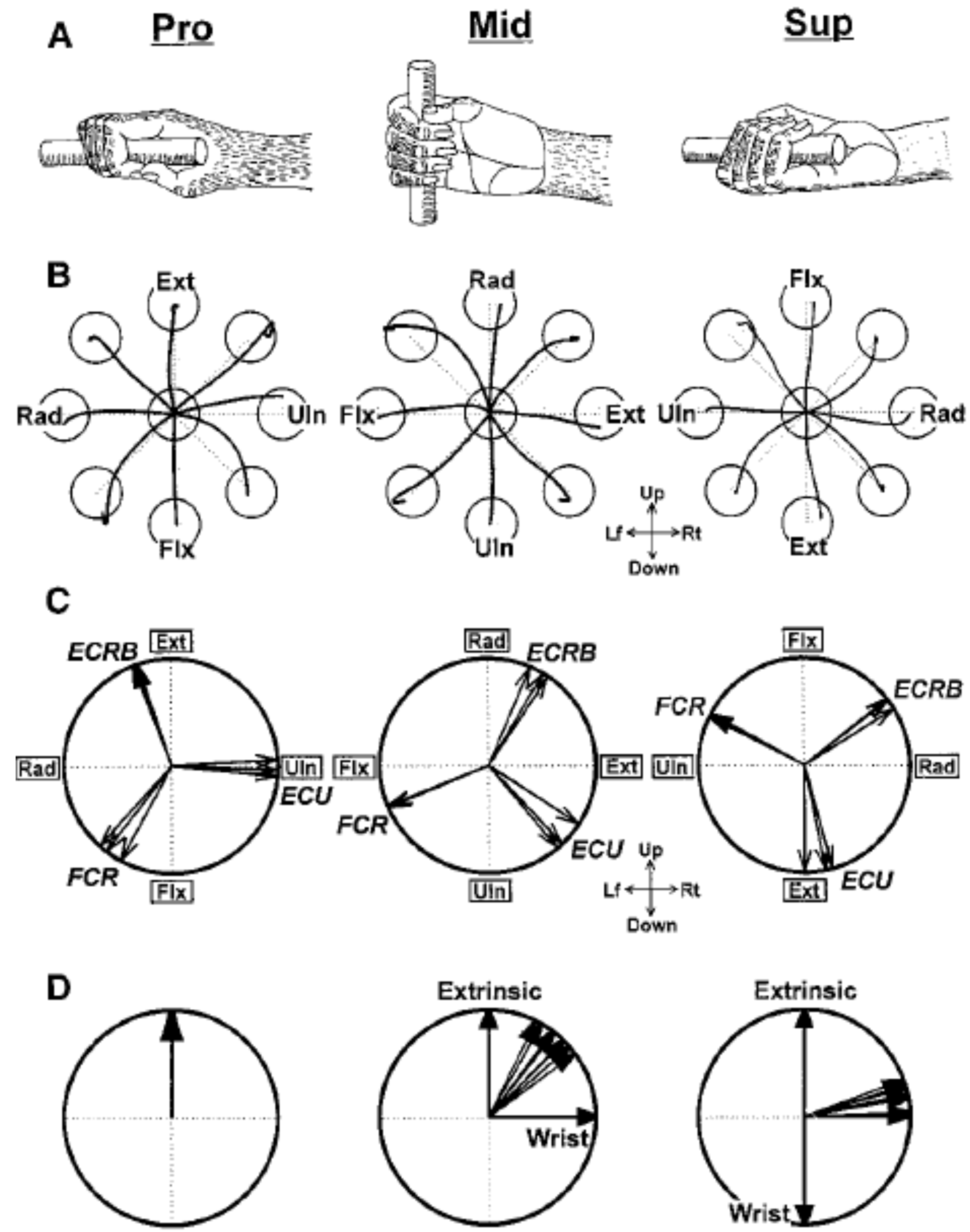
Center-Out reach task



Georgopoulos, Schwartz, Kettner, 1986

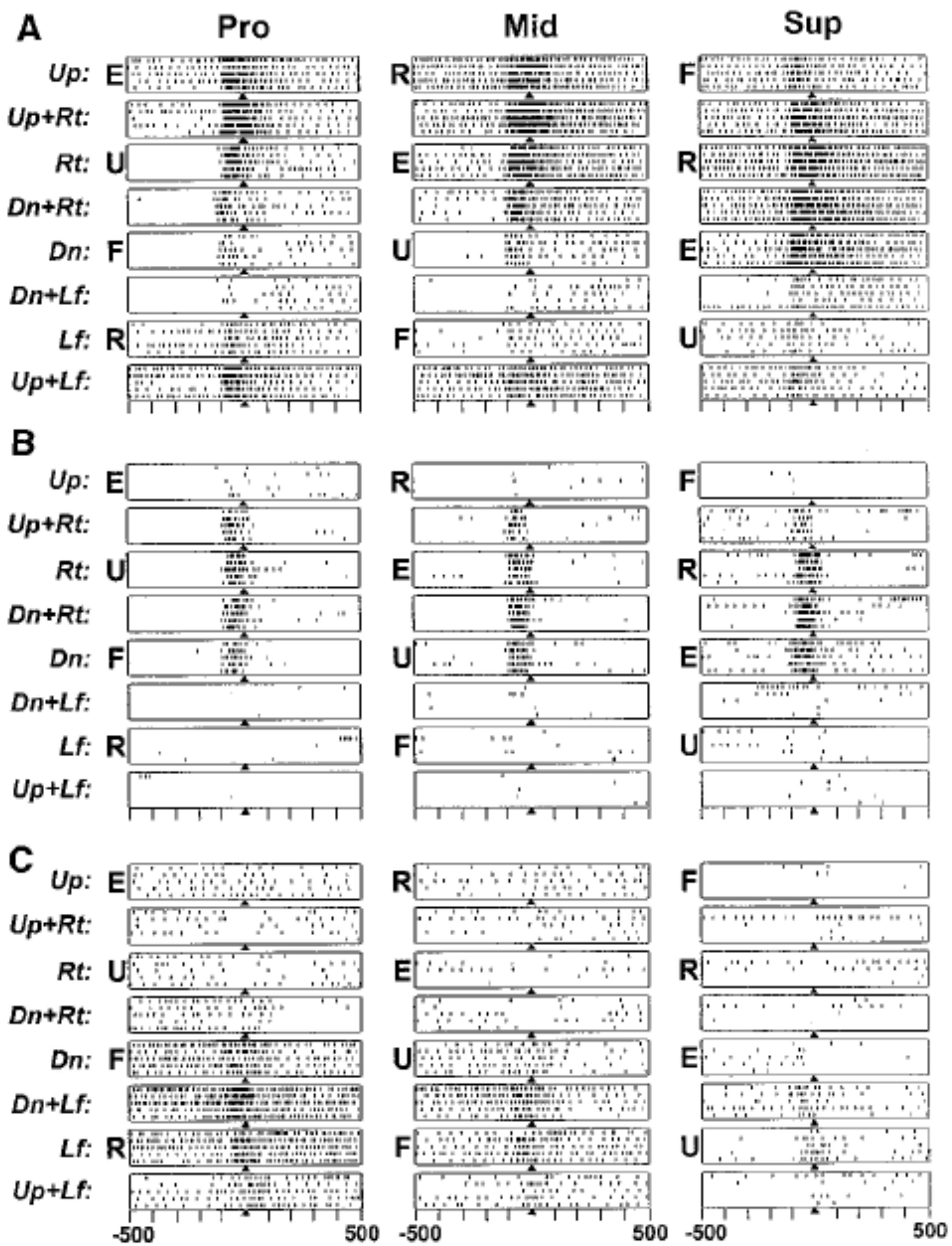
M1 codes *Both*: Evidence





Kakei, Hoffman, Strick 1999

Example neurons



"Muscle-like" neuron

Wrist orientation invariant

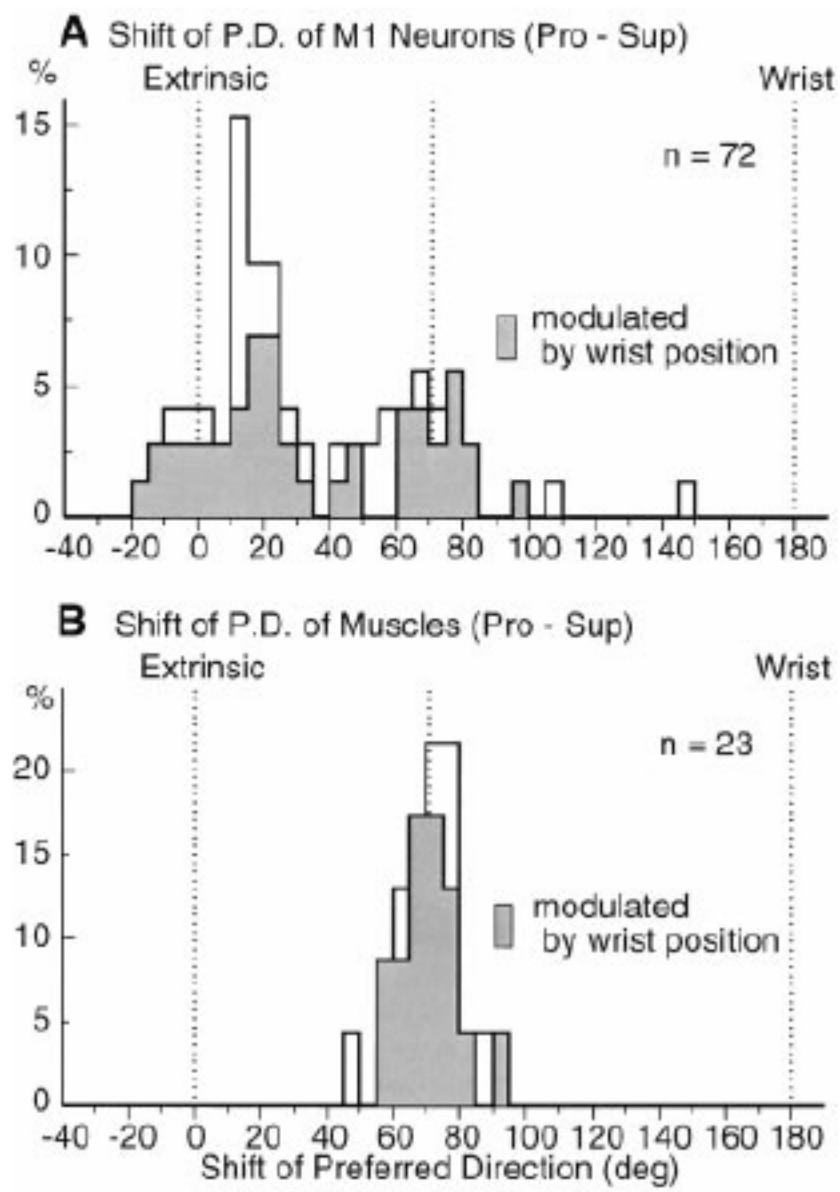
"Extrinsic-like" neuron

"Extrinsic-like" neuron, modulated by posture

Direction invariant

Kakei, Hoffman, Strick 1999

The population of individual neurons

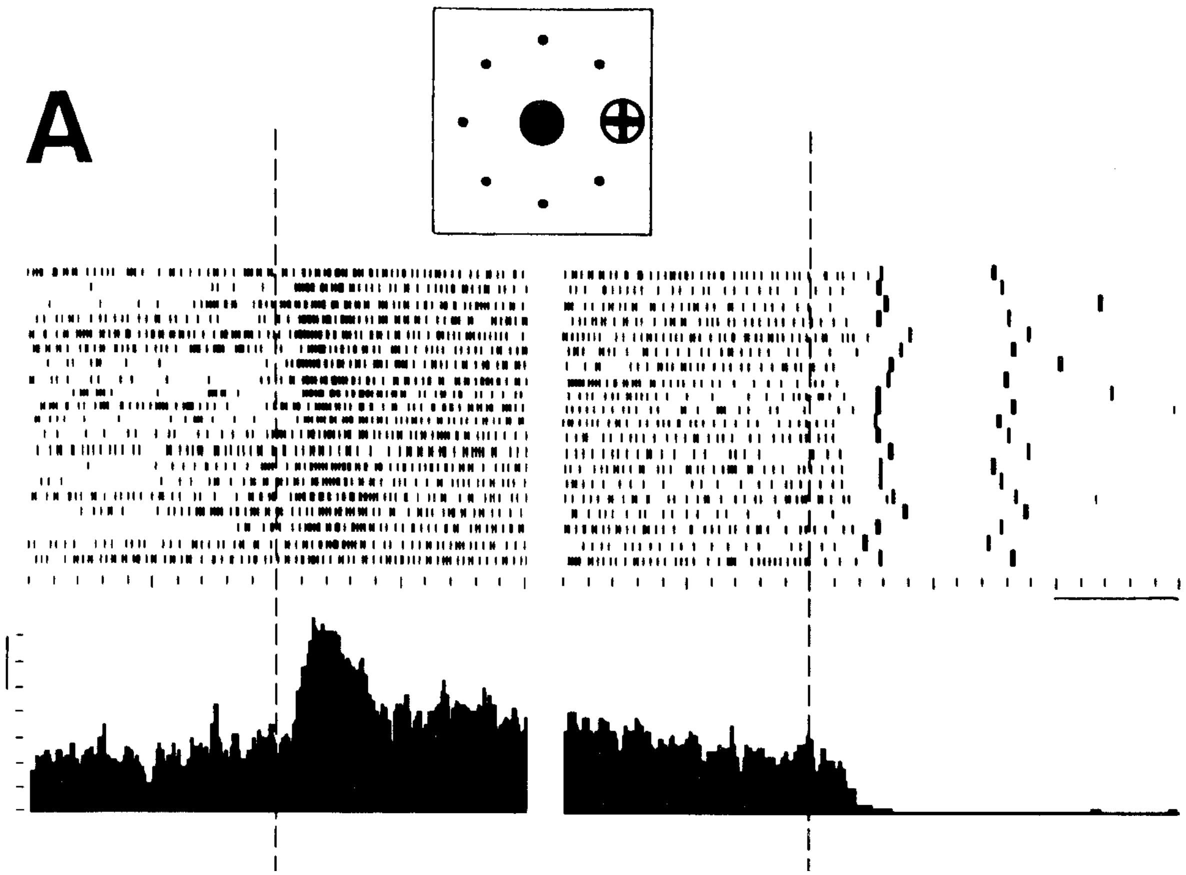


Neurons

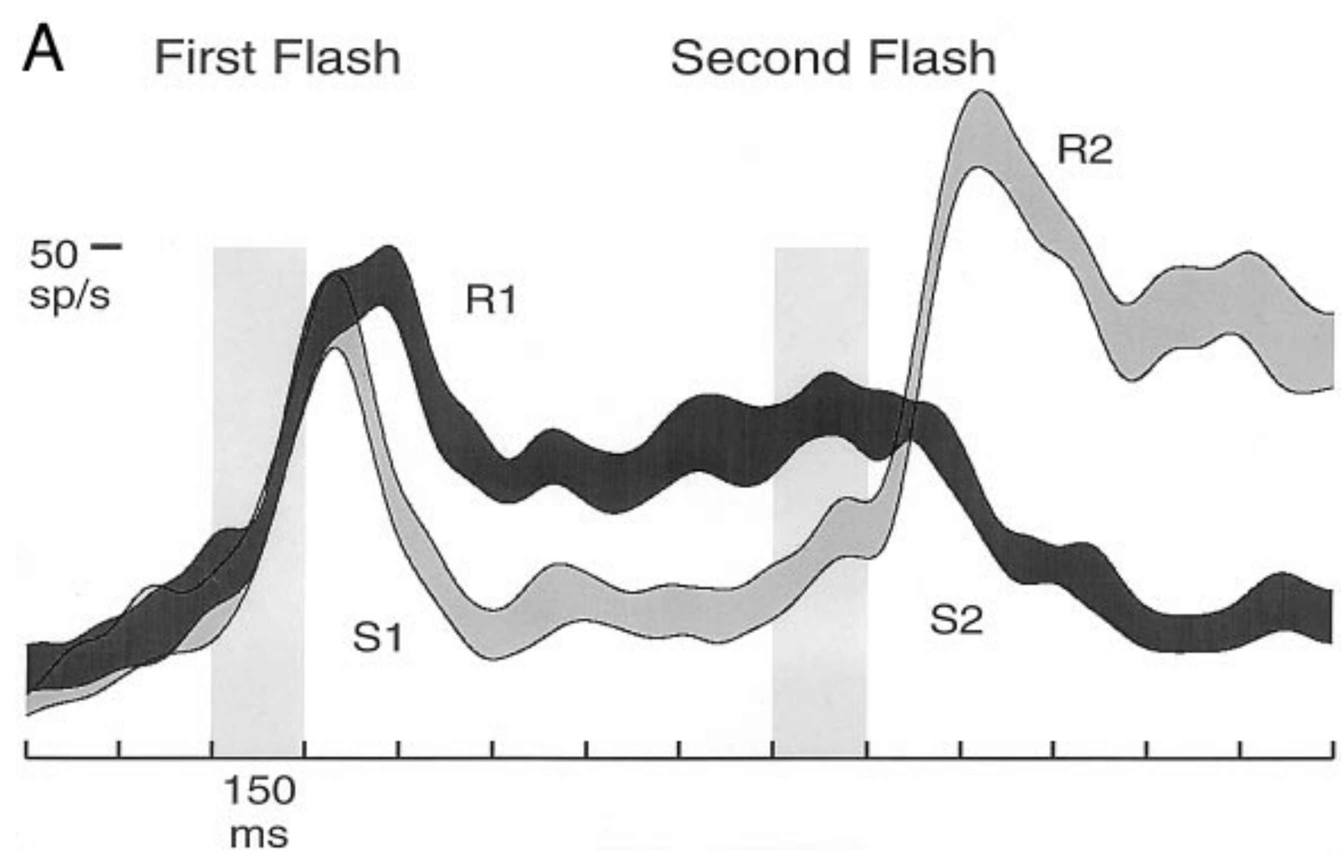
Muscles

Kakei, Hoffman, Strick 1999

Neurons are variable…



neurons are noisy
but their noise
correlates overtime

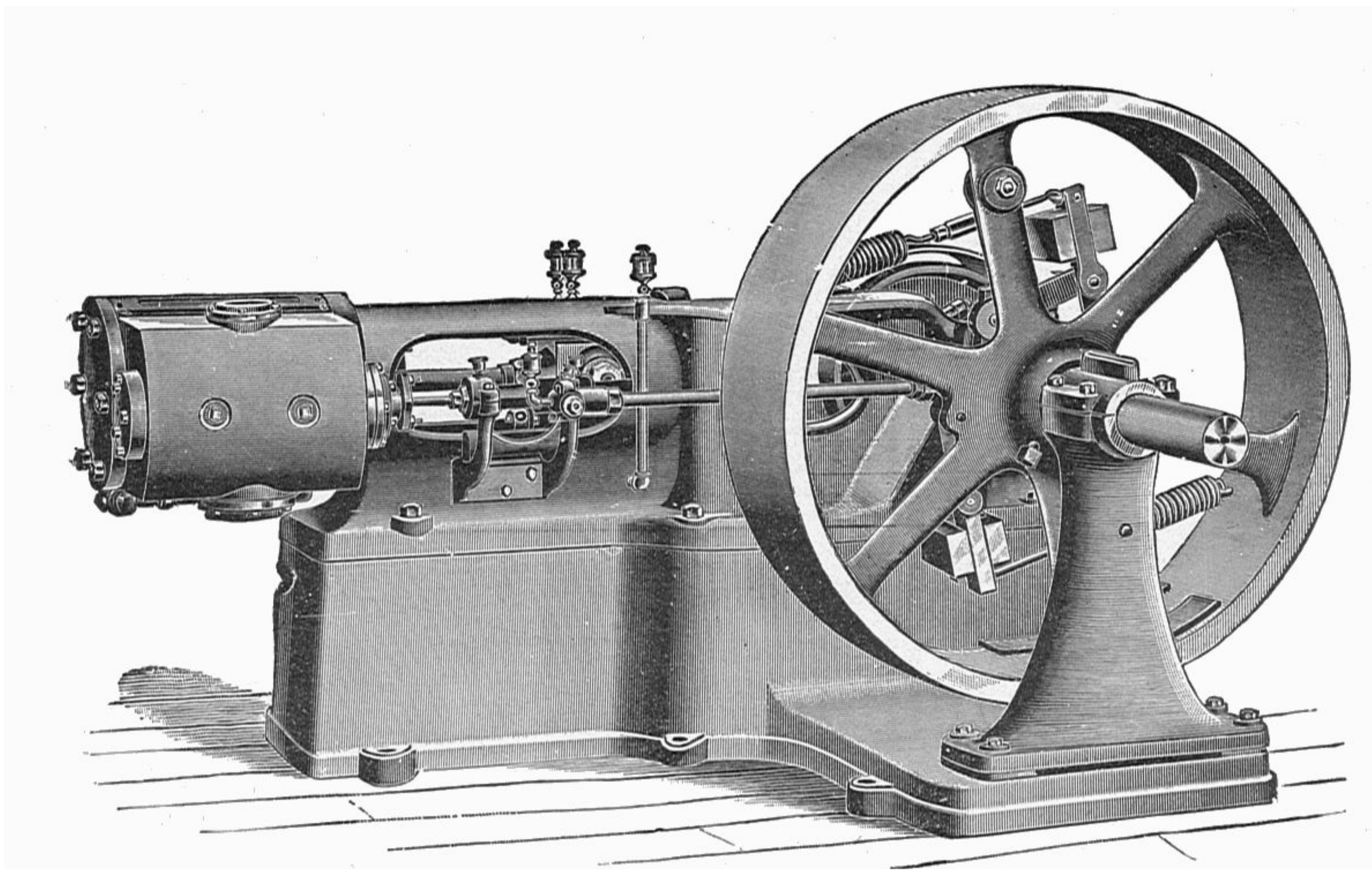


“One scientist’s signal is another scientist’s noise.”

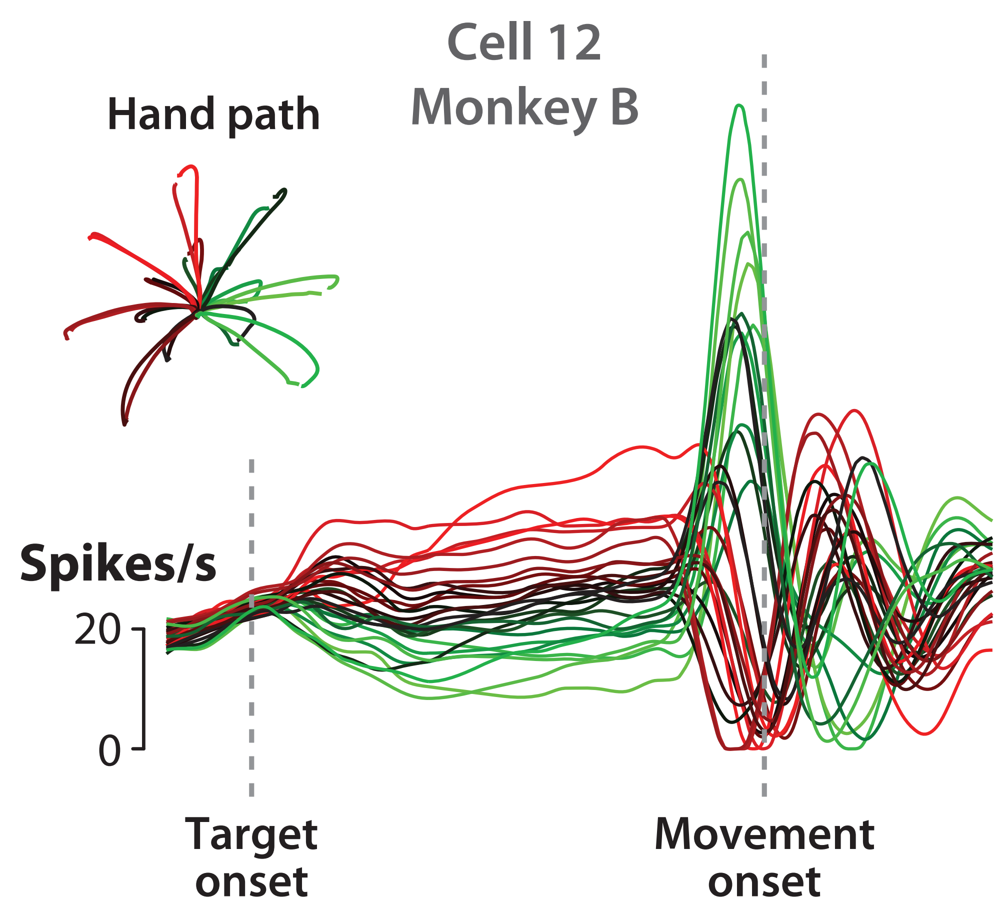
Does M1 generate commands for…

Muscles or Movements?

Or, is there another way to look at the problem?



M1 codes *dynamics*

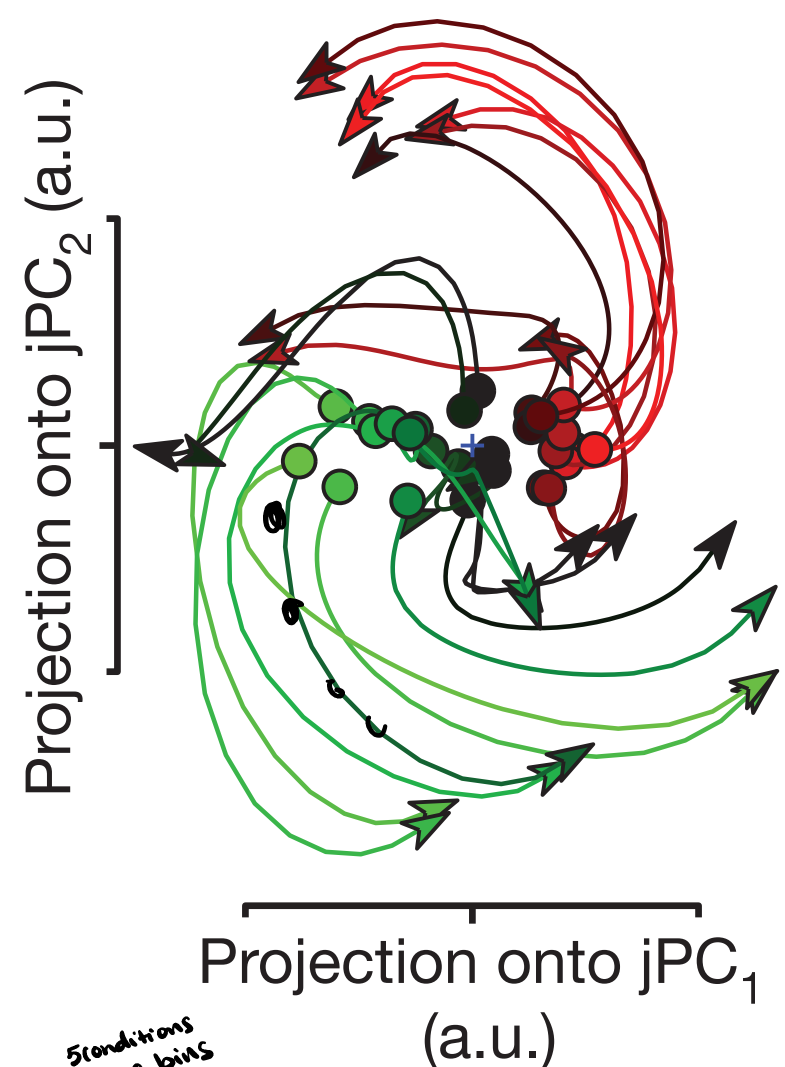


Churchland and Shenoy and colleagues, 2010s

M1 codes *dynamics*

50 neurons
100 trials
5 conditions
10 timebins

What dimensions of input array to get this output using PCA?



OR

50x50

c1t1 … c1t10 c2t1 … c2t10 … c5t10

n1
n2
n3
⋮

neurons

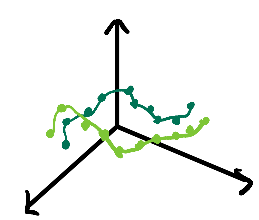
50 x 1 ← 1PC

is this the projection vector?

project data onto this projection vector

How?

50 x 50



Project C1t1 onto 50 x 1

project C1t2 onto 50x1
⋮

Trial average

50 neurons
5 conditions
10 timebins

? →

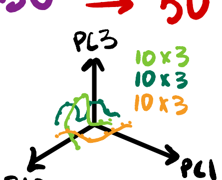
input array

n_1 n_2 n_3 …

c1t1
⋮
c1t10
c2t1
⋮
c2t10
⋮
c5t10

5 conditions
10 time bins

50 x 50 → 50 x 3 ← PCs (neurons reduced)
?



Output

50 x 1 → PC1
50 x 1 → PC2
50 x 1 → PC3

50 x10 $\xrightarrow{PC1}$ 1 x10
50 x10 $\xrightarrow{PC2}$ 1 x10
50 x10 $\xrightarrow{PC3}$ 1 x10

3x10 one condition
⋮
50 x10

C1t1 → 10 x 50
time bins, neurons
↓
10 x 3
⋮
C5t10 → 10 x 50
↓
10 x 3

c1 3x10 c2 3x10
50 →

Churchland and Shenoy and colleagues, 2010s

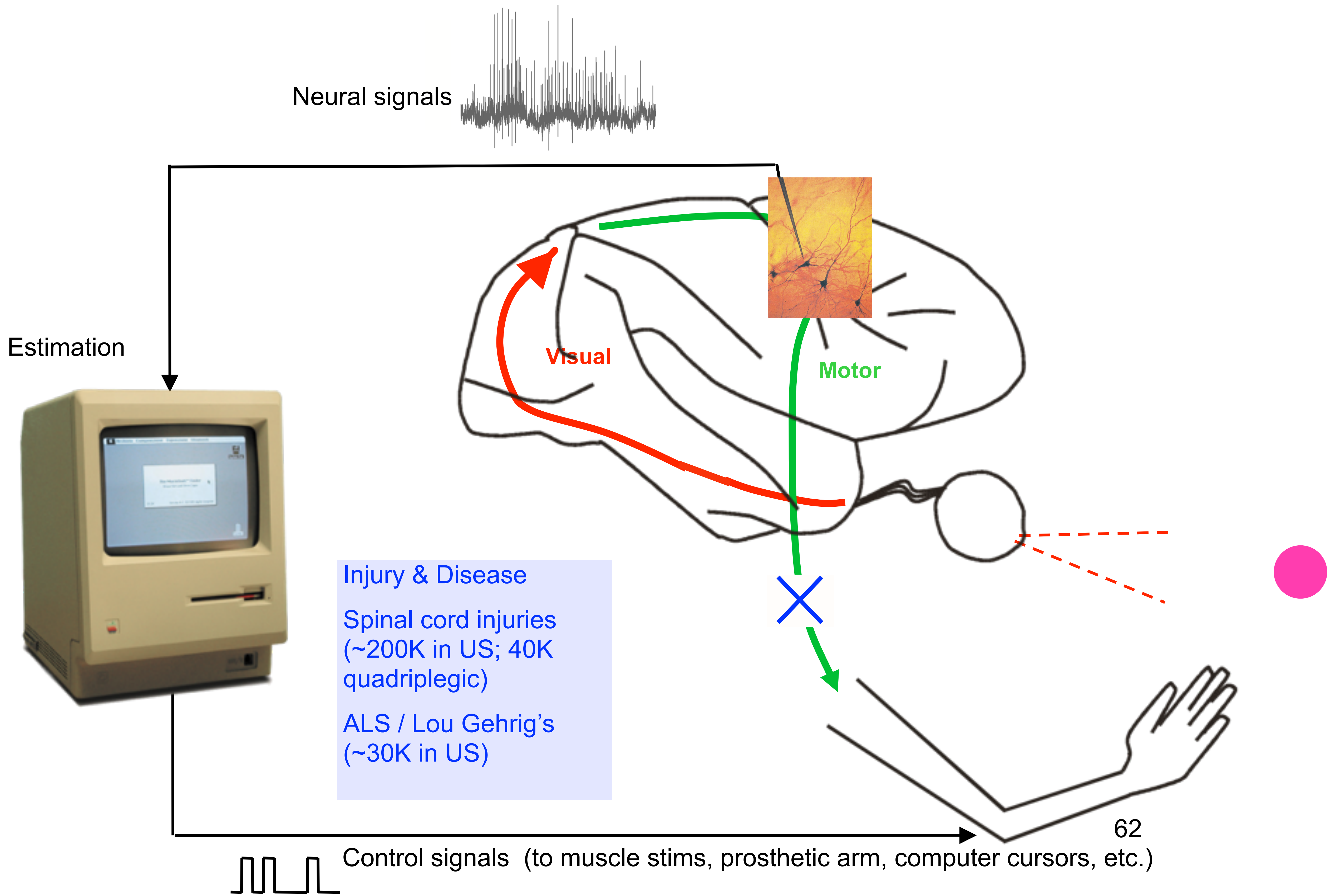
Step 1:

Throw away the mean response.

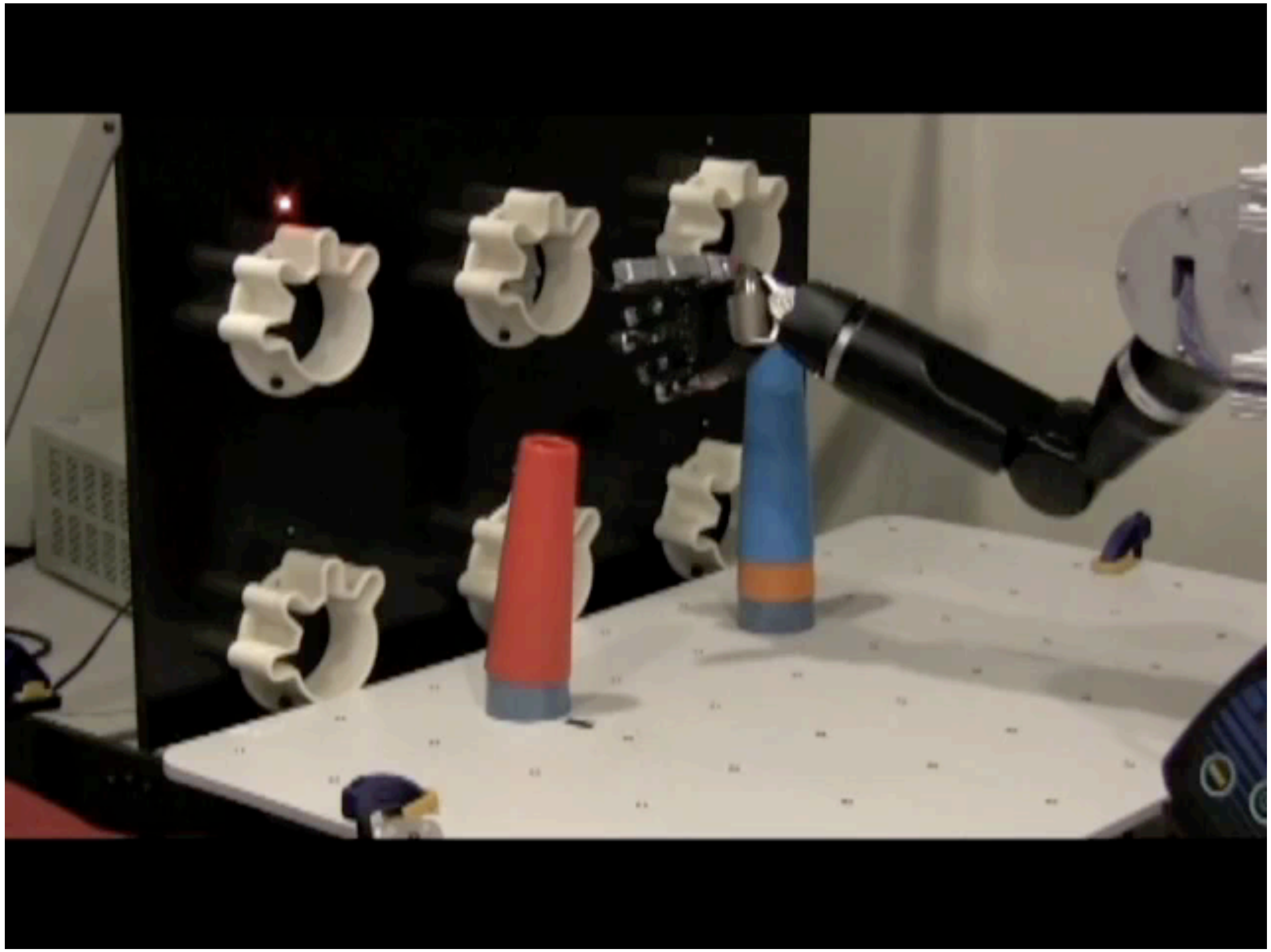
Outline

- Neurophysiology of motor control
- **Brain-Computer Interface algorithms**
- Population methods in basic neuroscience
- New directions in BCI therapeutics

Neural Prosthetics

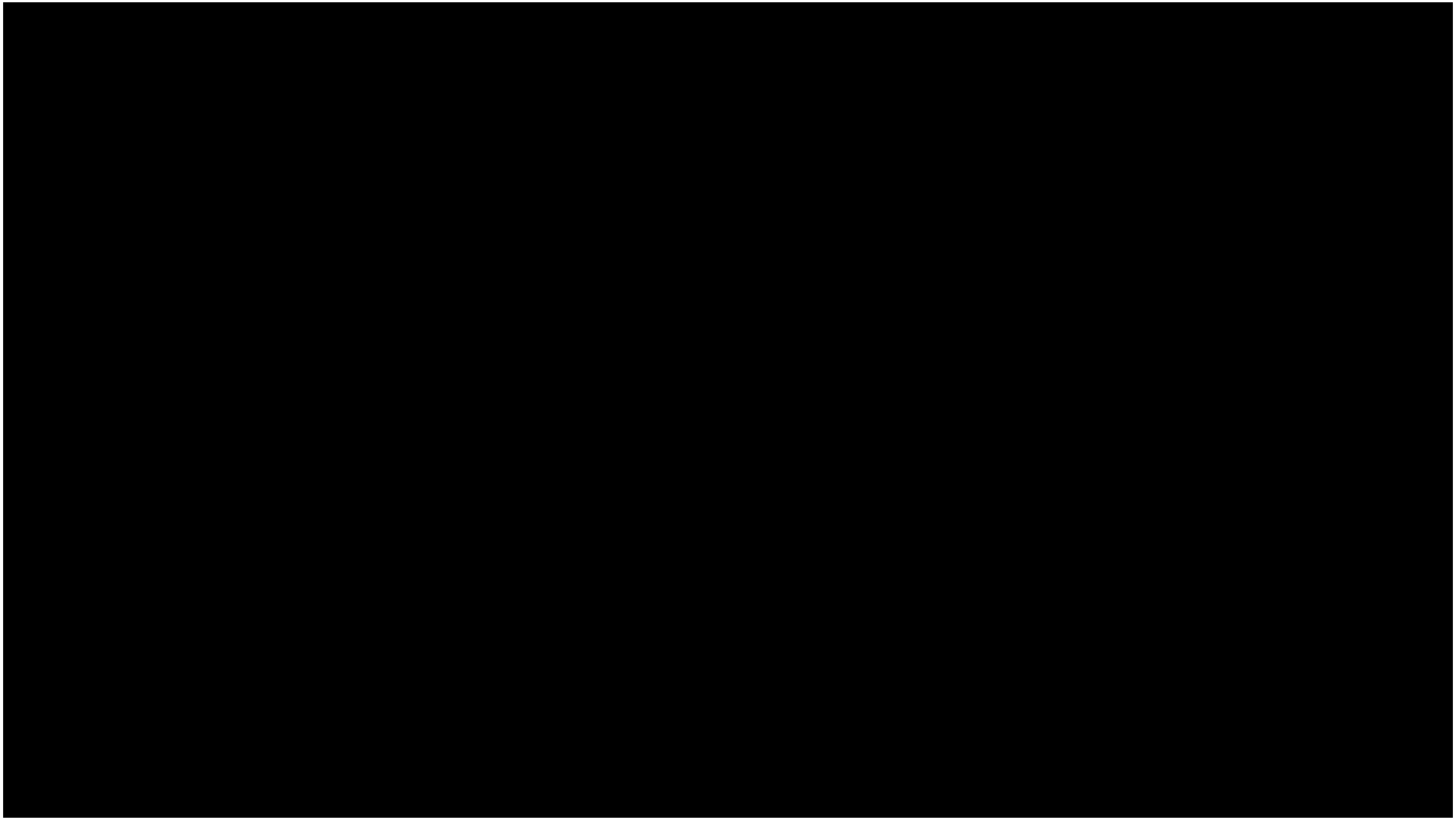


Neural Prosthetics



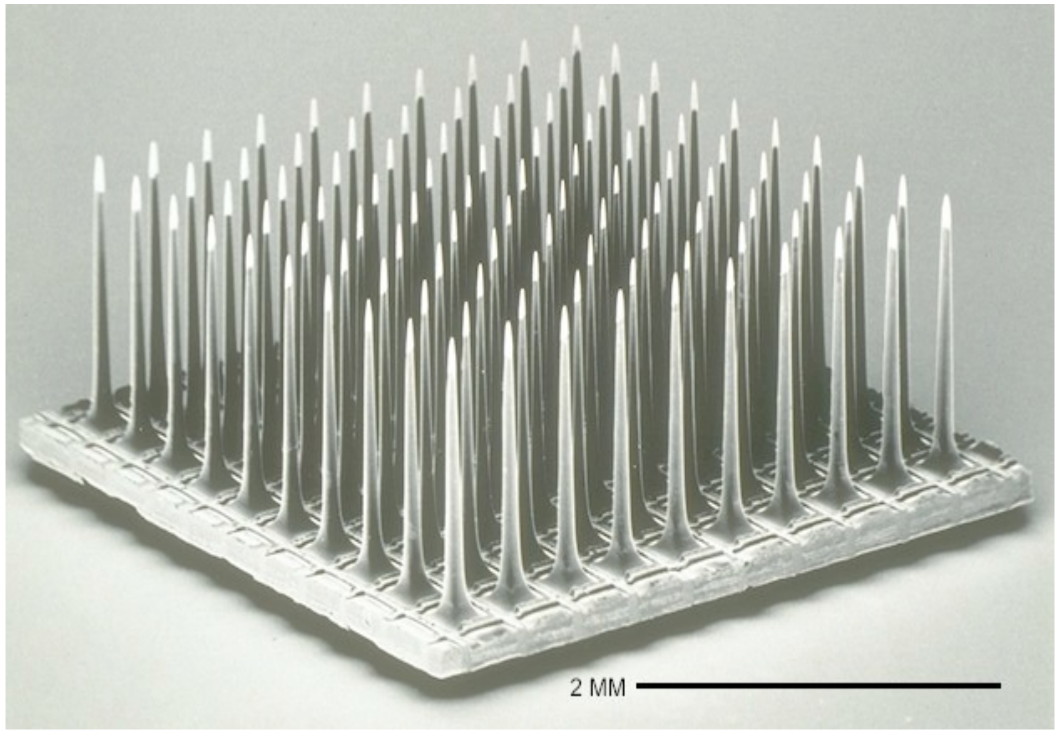
Collinger et al., *The Lancet*, 2013

Neural Prosthetics

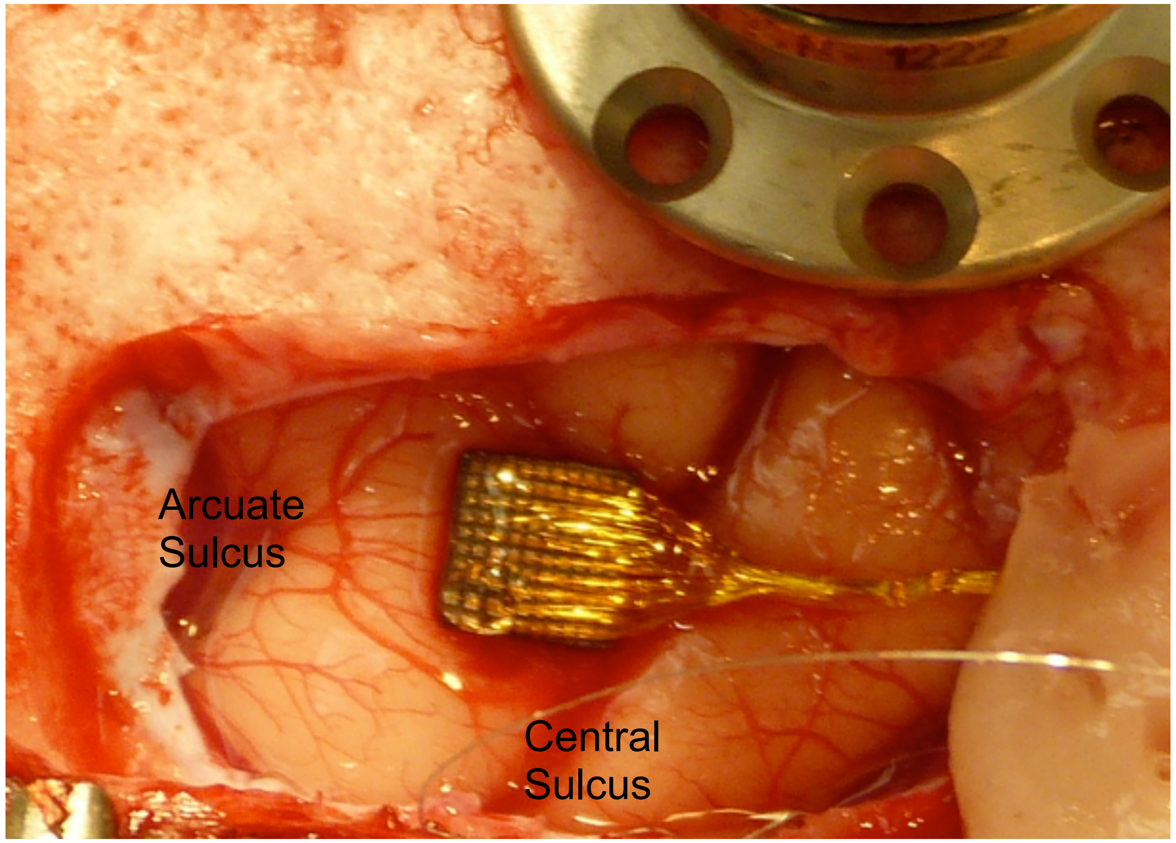


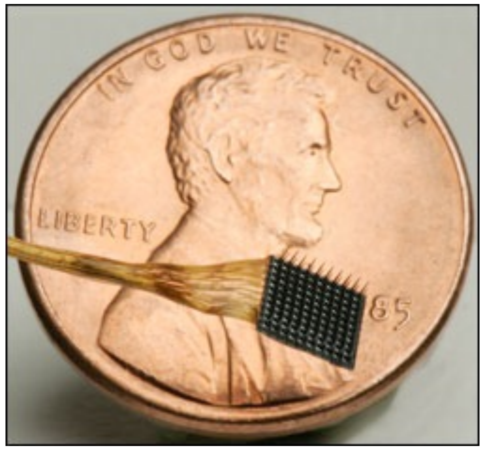
Velliste et al., *Nature* 2008

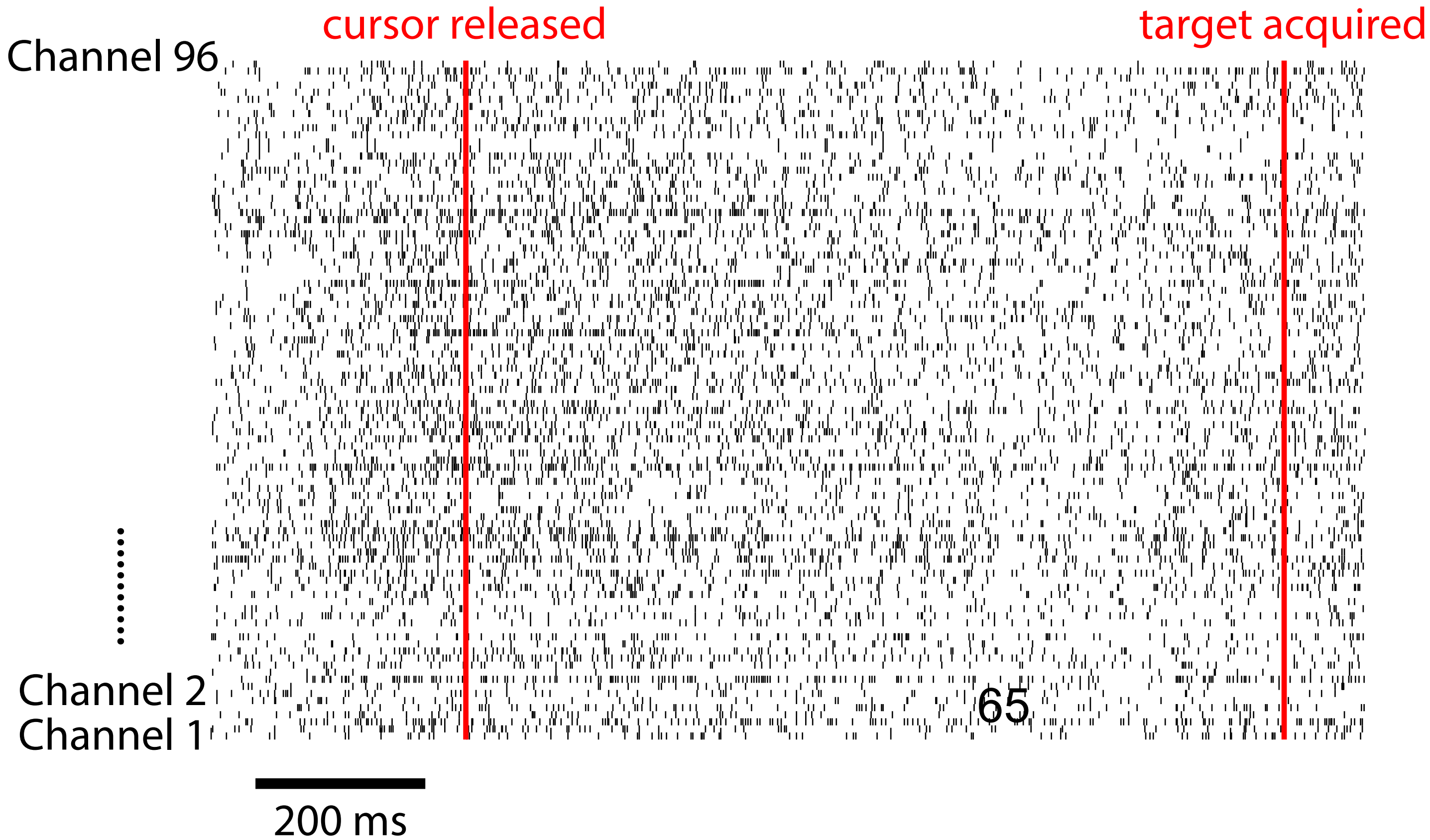
Multielectrode Array Recordings



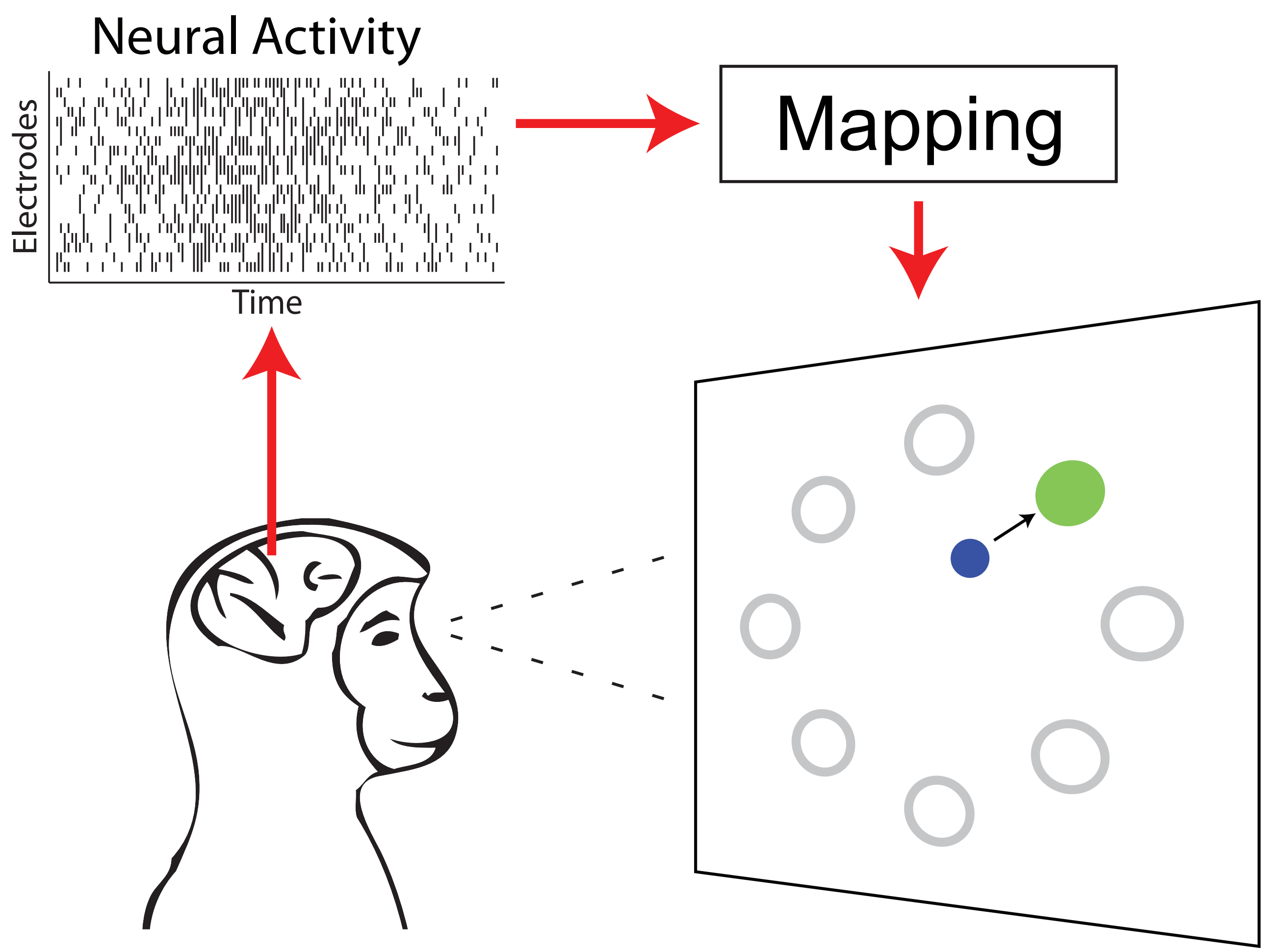
Blackrock Technologies







Using a BCI

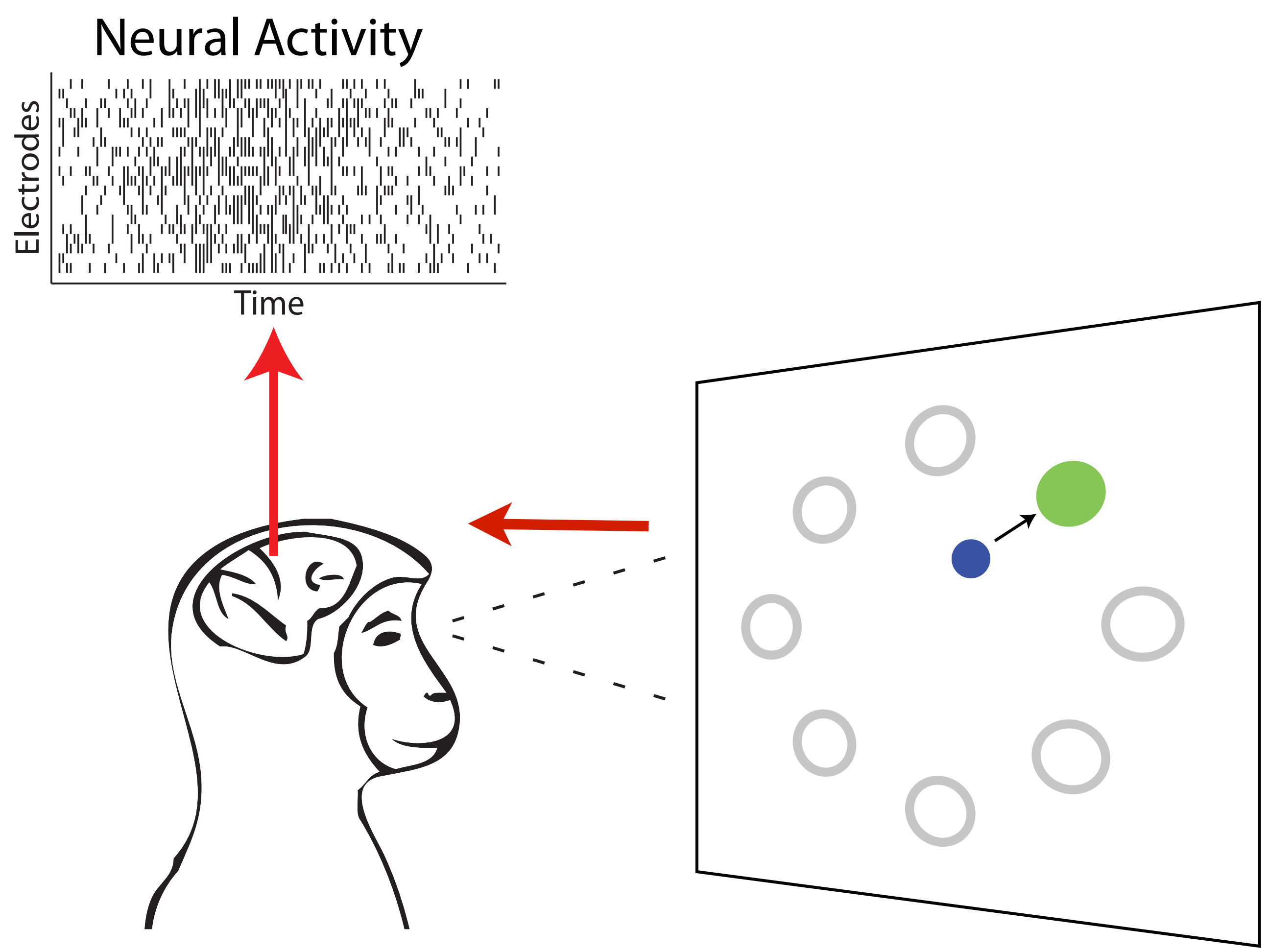


Designing a BCI

<u>Calibrate the BCI:</u>

- *We* move the cursor right or left.
- Neurons are active.
- Map those activity patterns to cursor kinematics.

"Observation-based" calibration



Designing a BCI

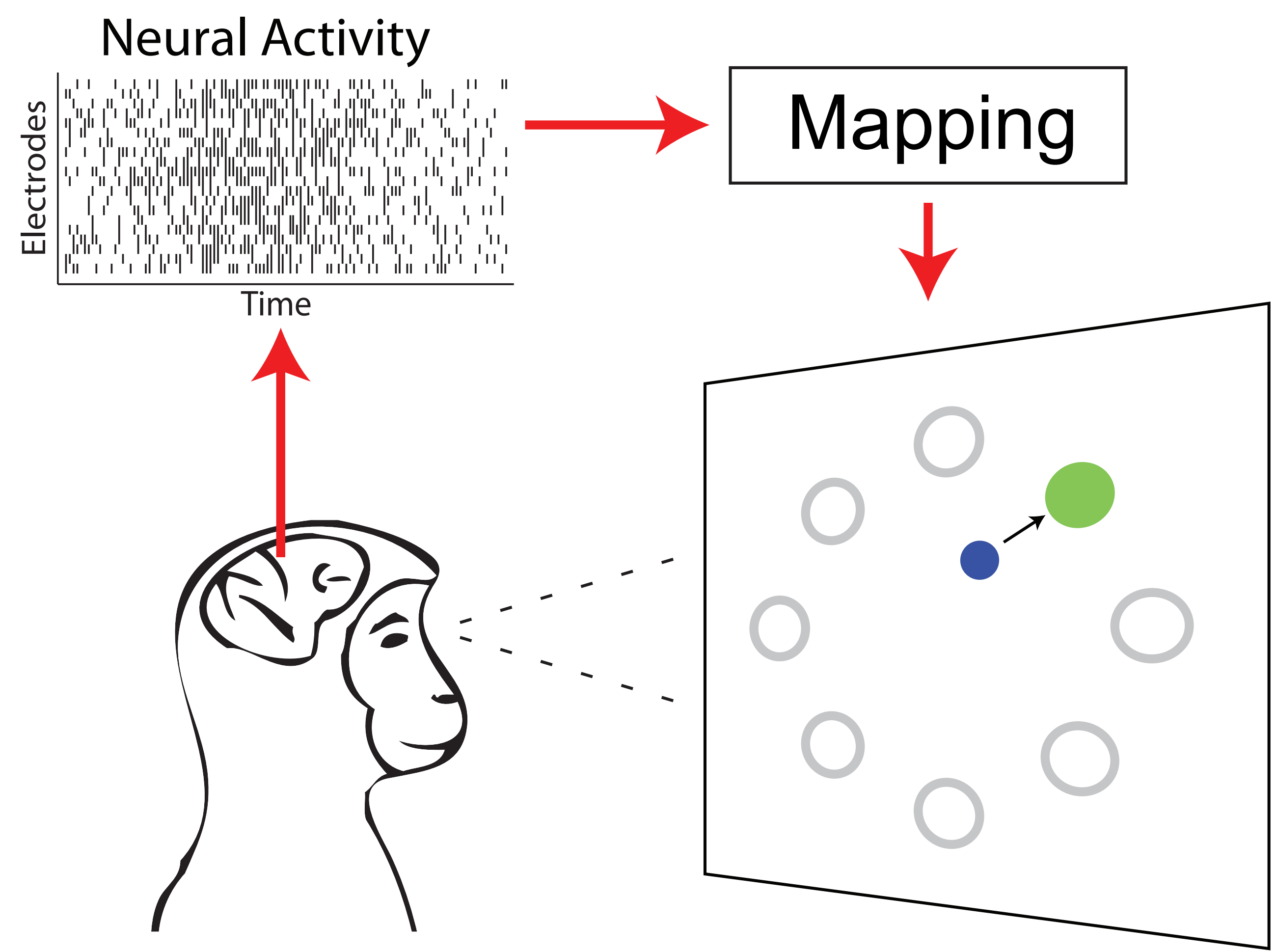
<u>Calibrate the BCI:</u>

- *We* move the cursor right or left.
- Neurons are active.
- Map those activity patterns to cursor kinematics.

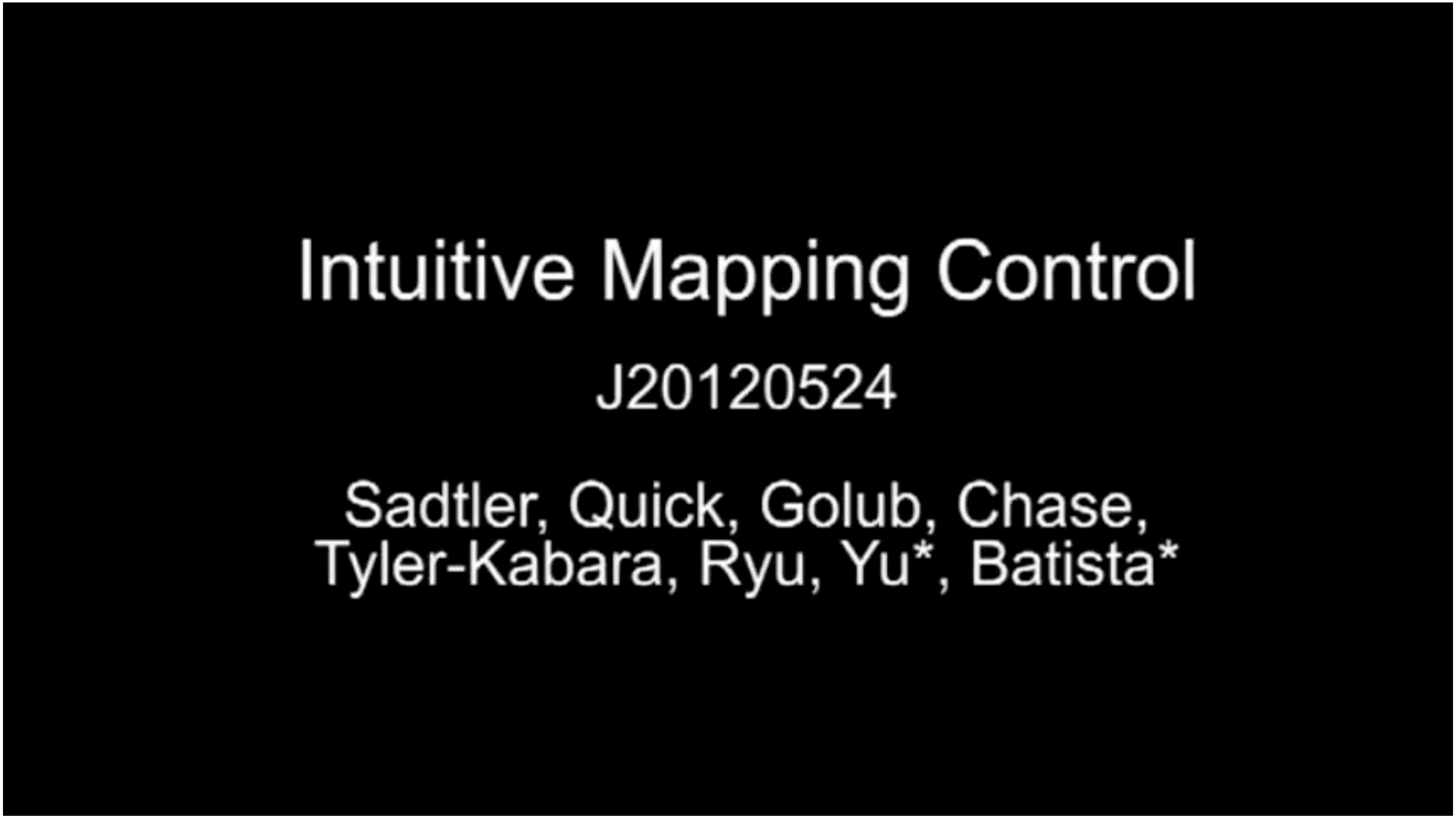
"Observation-based" calibration

<u>Control the BCI:</u>

- Generate those activity patterns to move the cursor.



BCI Performance



Three decode algorithms

1) Population vector

2) Linear filter

3) Kalman filter

Three decode algorithms

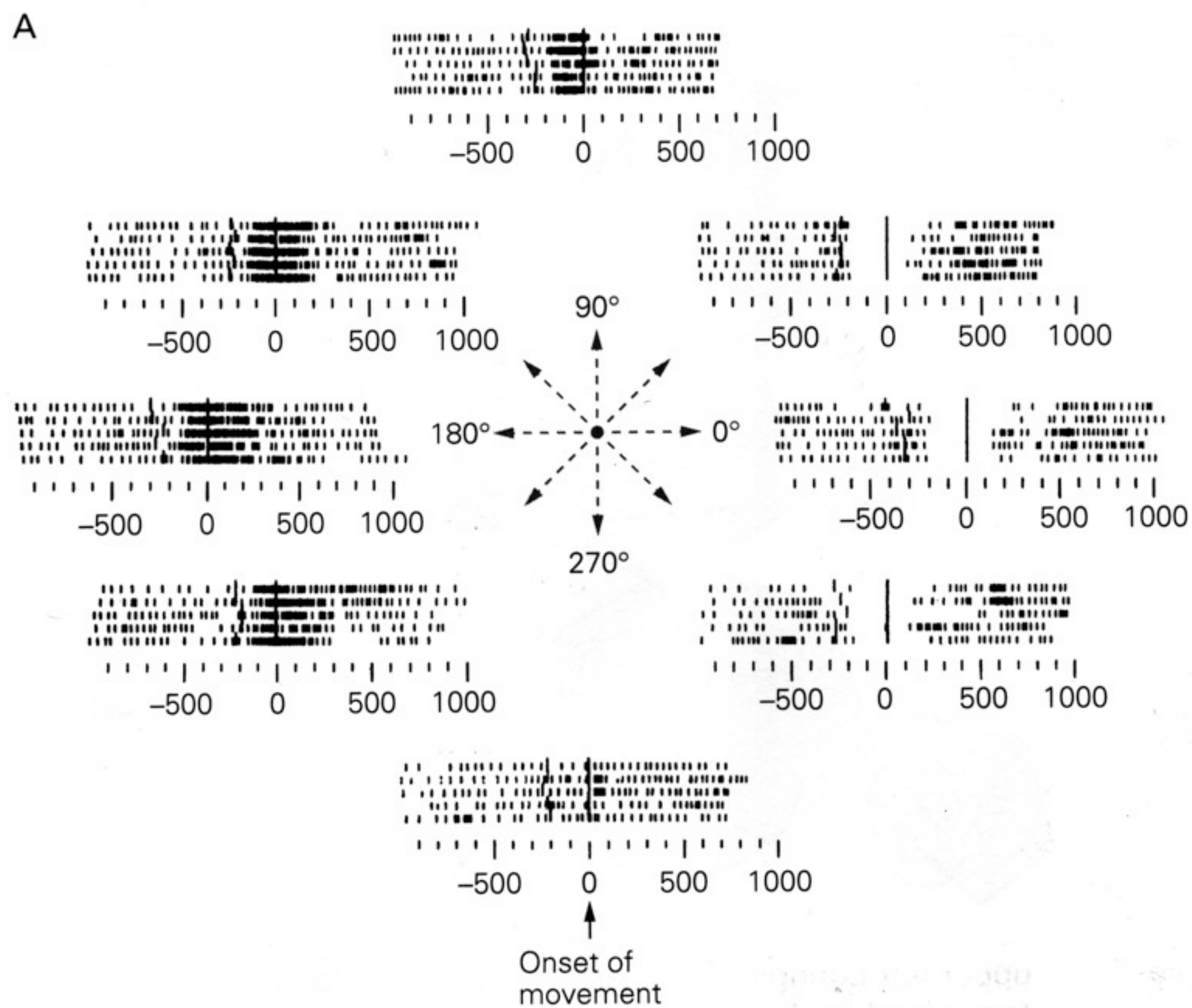
1) Population vector

2) Linear filter

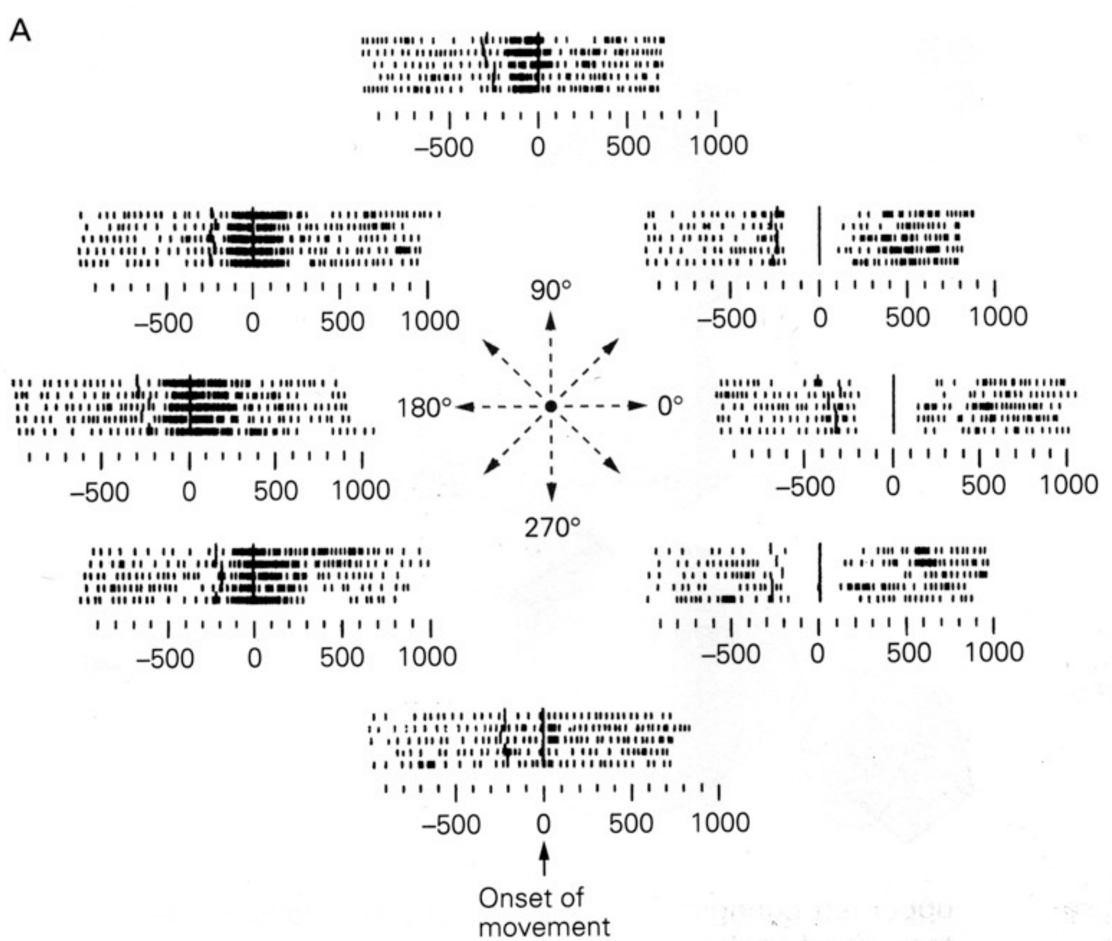
3) Kalman filter

1) Population Vector Algorithm

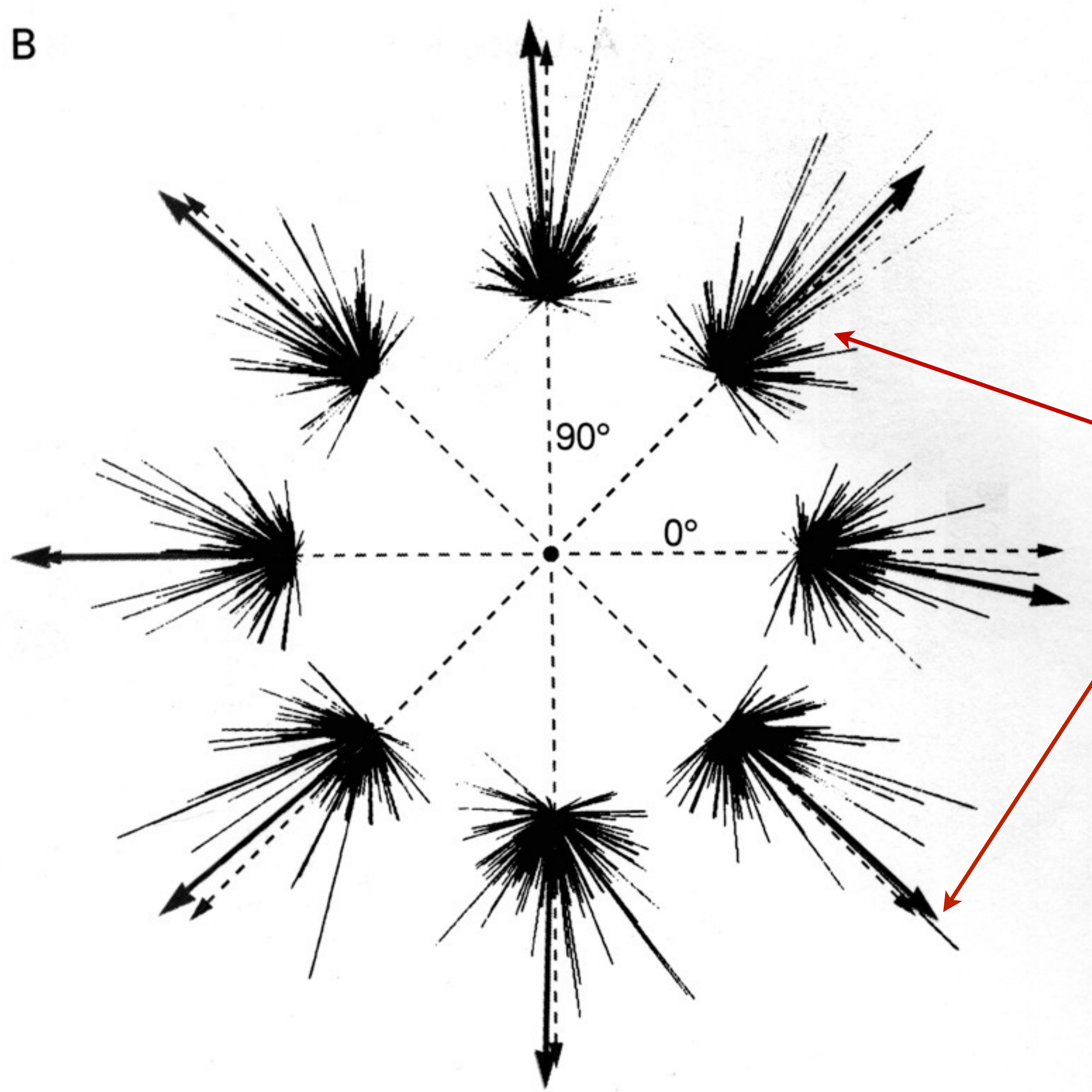
Center-Out reach task



Georgopoulos et al., 1982

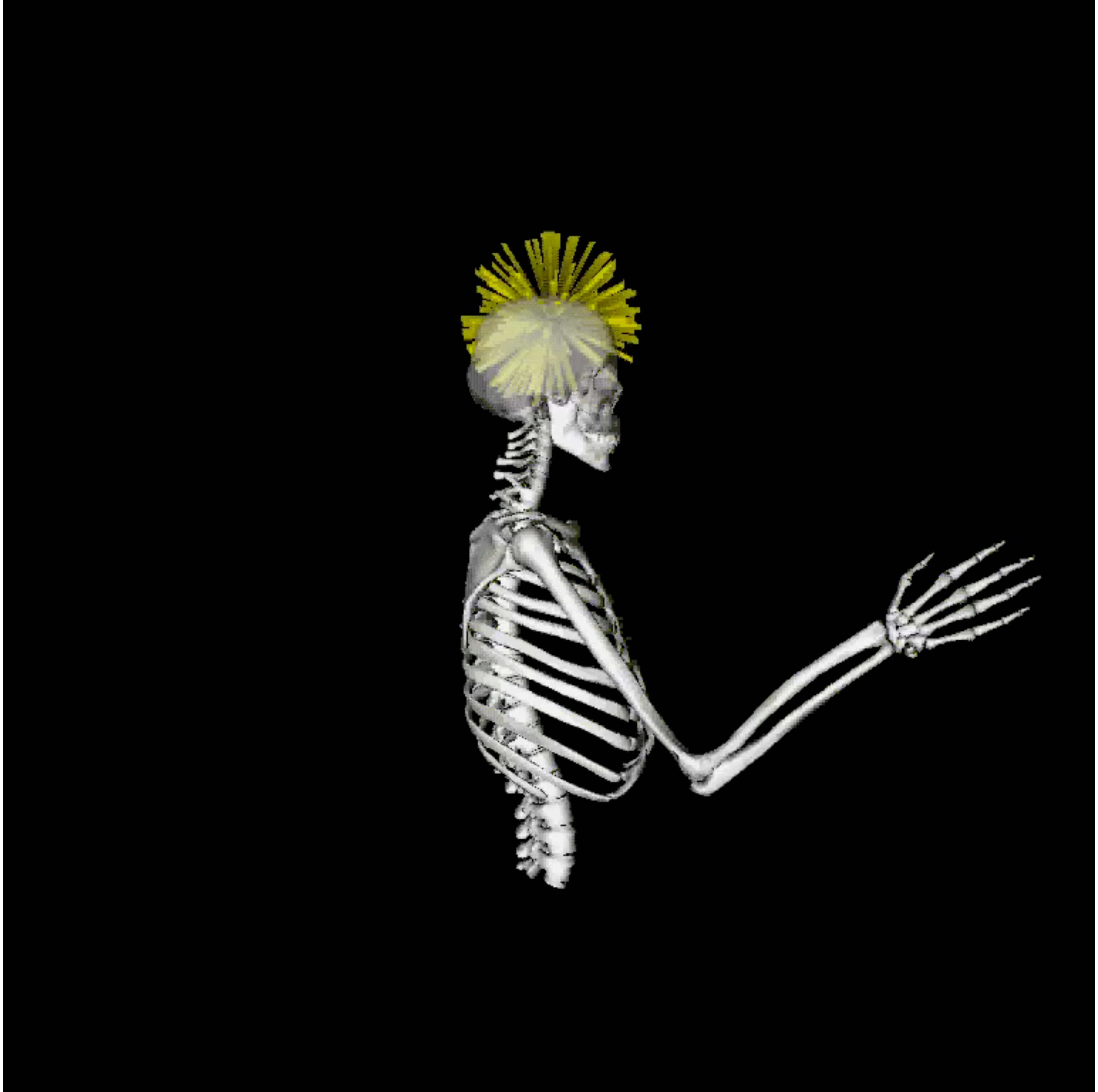


- Neurons are broadly tuned
- Many neurons active during each reach
- "Population code" for movement direction



Population Vector:

1. Fit each neuron with a cosine.
2. Estimate the preferred direction.
3. Weight each preferred direction by the cell's firing rate.
4. Sum all the weighted vectors.



Three decode algorithms

1) Population vector

2) Linear filter

3) Kalman filter

2) Linear filter

$$\vec{y}(t) = \vec{b} + \sum_{u=-m}^{n} \vec{a}(u)\vec{x}(t-u) + \vec{\epsilon}(t)$$

y: external observable
x: neural activity
u: time lag
a: weights
b: intercepts
e: error

It's just

$$y = mx + b + \epsilon$$

y: external observable
x: neural activity
m,b: parameters we must find

Testing (Decoding)

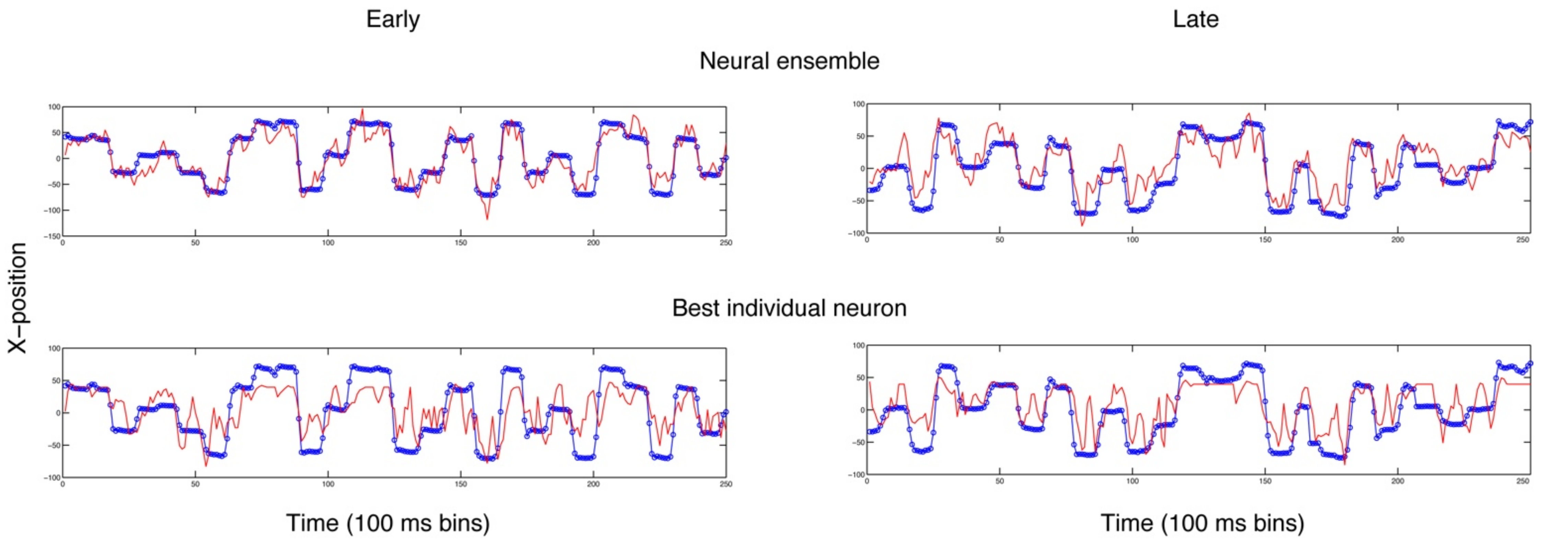
$$\boxed{Y = XA}$$

$$\boxed{A = (X^T X)^{-1} X^T Y}$$

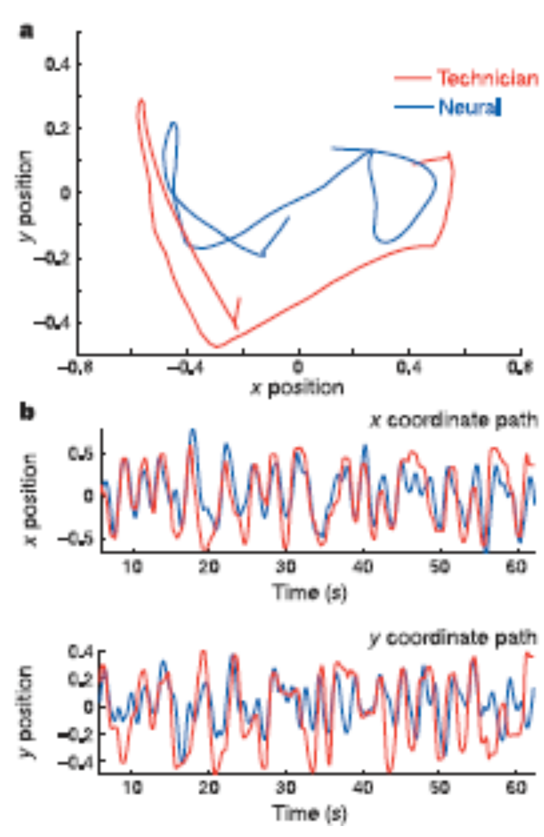
Training

Intuition: each spike contributes a little pulse of movement. Add those all together.

2) Linear filter



Chestek et al., 2008



Hochberg et al., 2006

Three decode algorithms

1) Population vector

2) Linear filter

3) Kalman filter

4) Kalman Filter (continuous Bayes estimator)

4) Kalman Filter (continuous Bayes estimator)

State Equation:

$y(t) = f(y(t-1))$

Observation Equation:

$x(t) = g(y(t))$

f is the state transition function,
g is the observation function,
y is the state of the system (e.g. position, velocity, etc.)
x is the measurements you made.

4) Kalman Filter (continuous Bayes estimator)

State Equation:

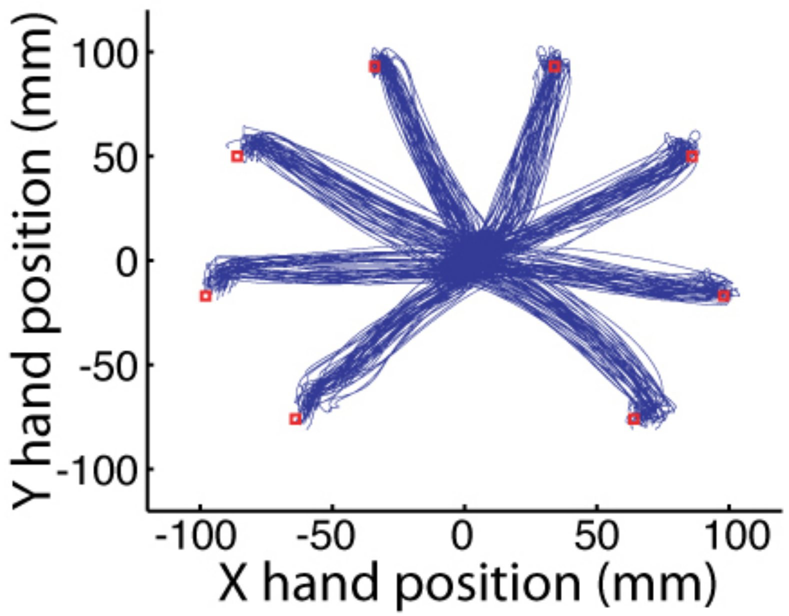
$$y(t) = f(y(t-1)) + \text{noise}$$

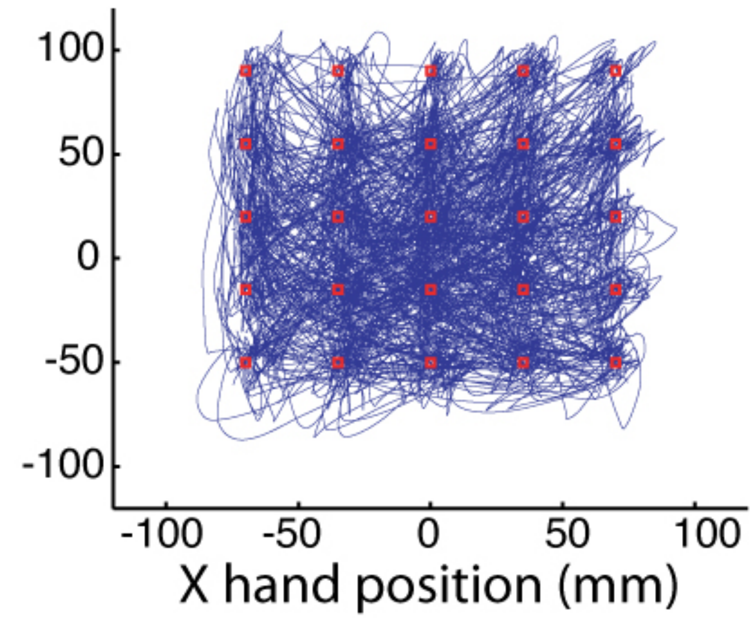
Observation Equation:

$$x(t) = g(y(t)) + \text{noise}$$

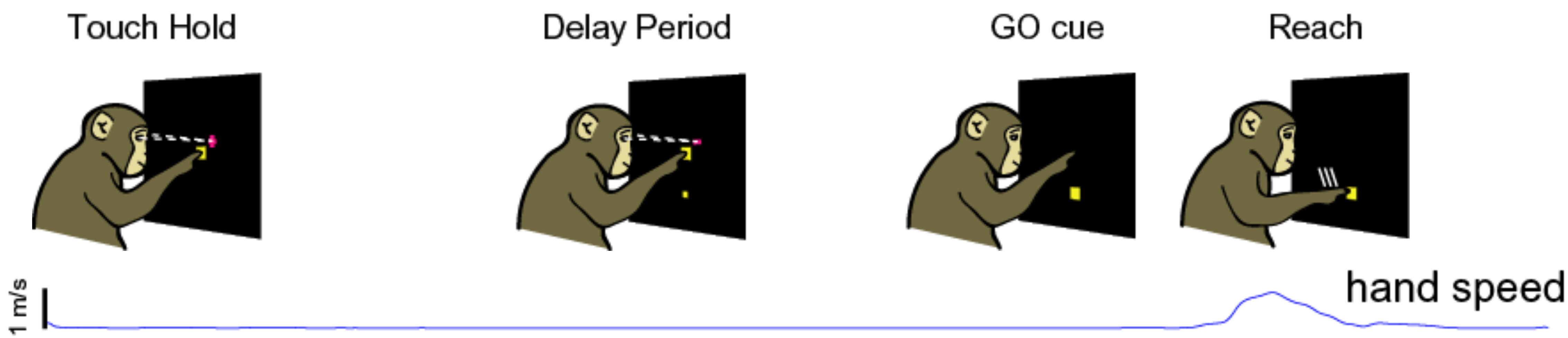
f is the state transition function,
g is the observation function,
y is the state of the system (e.g. position, velocity, etc.)
x is the measurements you made.

4) Kalman Filter (continuous Bayes estimator)





We know the statistics of hand movements.
Why not use it?



$$\text{arm state } \mathbf{x}_t = \begin{bmatrix} \text{position} \\ \text{velocity} \\ \text{acceleration} \end{bmatrix}$$

Dynamical model

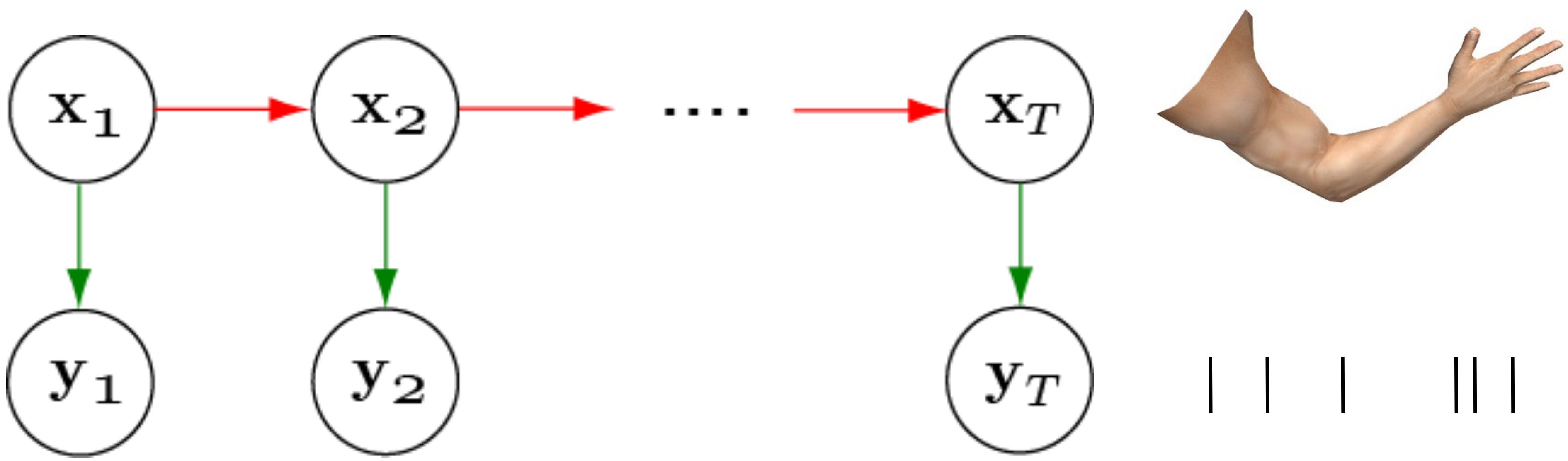
Trajectory model

$$\mathbf{x}_t \mid \mathbf{x}_{t-1} \sim \mathcal{N}\left(A\mathbf{x}_{t-1},\ Q\right)$$

Dynamical model

Trajectory model

Observation model



$$\mathbf{y}_t \mid \mathbf{x}_t \sim \mathcal{N}\left(C\mathbf{x}_t,\ R\right)$$

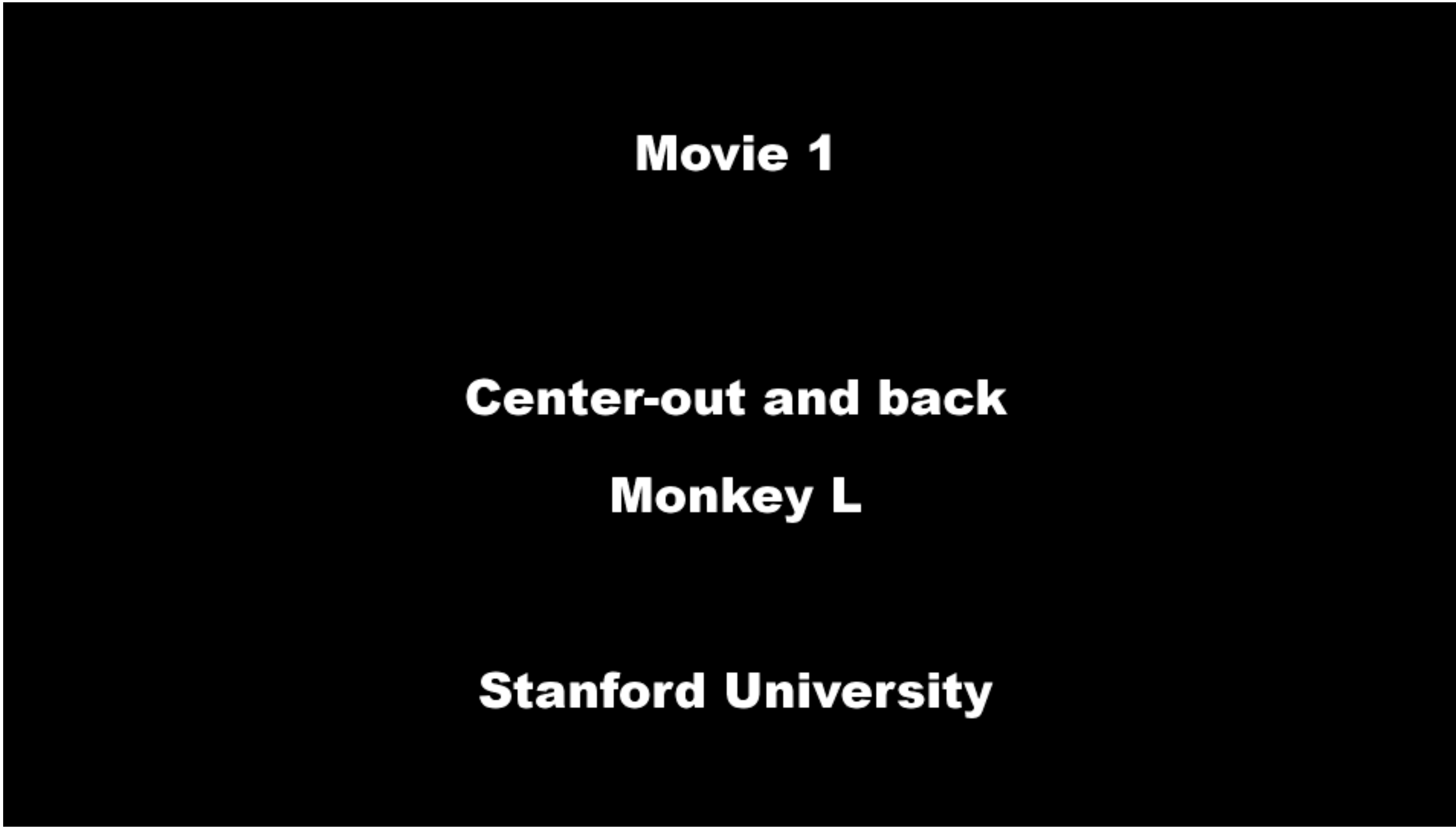
Kalman Filter

Using the trajectory model and observation model, we compute at each timepoint:

Arm state estimate

$$\hat{x}_t = M_1 x_{t-1} + M_2 y_t$$

- M1 and M2 depend on the noise in the state update, Q, and the noise in the observation estimate R.
- The *Kalman gain* captures the tradeoff between them.

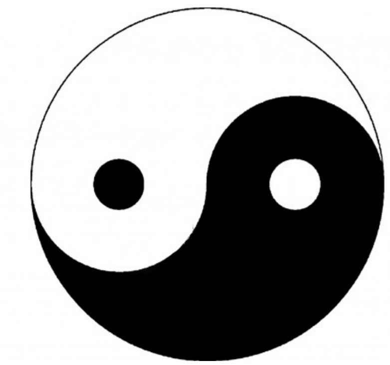


Gilja, Shenoy, et al. 2012

So, which is the right algorithm?

- It depends (at least somewhat) on what you think M1 does naturally.
    - Although all of these decode *velocity*.
- Ancillary factors might skew the results:
    - Amount of experience
    - Implementation details (number of parameters, quality of recordings, etc.)
- Learning probably erases most of the differences between them in performance.
- Next-Gen algorithms come out all the time.
    - These days, of course, it's neural networks.
-

Outline

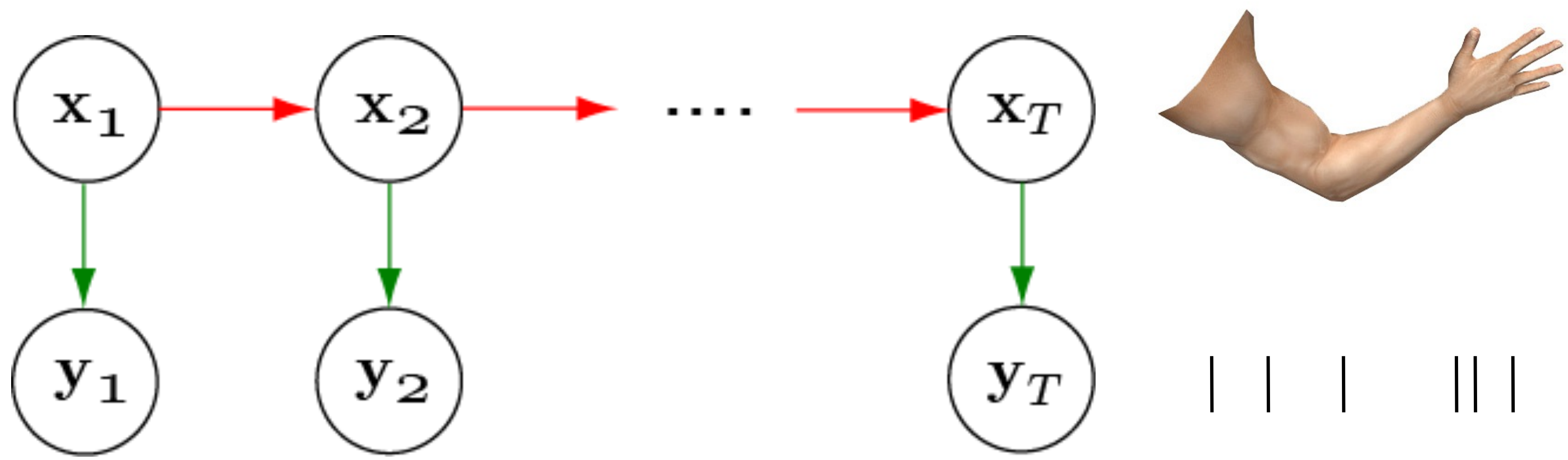


- Neurophysiology of motor control
- Brain-Computer Interface algorithms
- **Population methods in basic neuroscience**
- New directions in BCI therapeutics

Dynamical model

Trajectory model

Observation model



$$\mathbf{y}_t \mid \mathbf{x}_t \sim \mathcal{N}\left(C\mathbf{x}_t,\, R\right)$$

Latent-Variable modeling

Motivating question: What does the activity of a neuron really tell us?

Latent-Variable modeling

Motivating question: What does the activity of a neuron really tell us?

The activity of a neuron is governed by its inputs, and it contributes to shaping the activity of its target neurons.

It may be the combined activity of lots of neurons that is important for neural information processing. (One supporting insight: neurons are arranged into networks.)

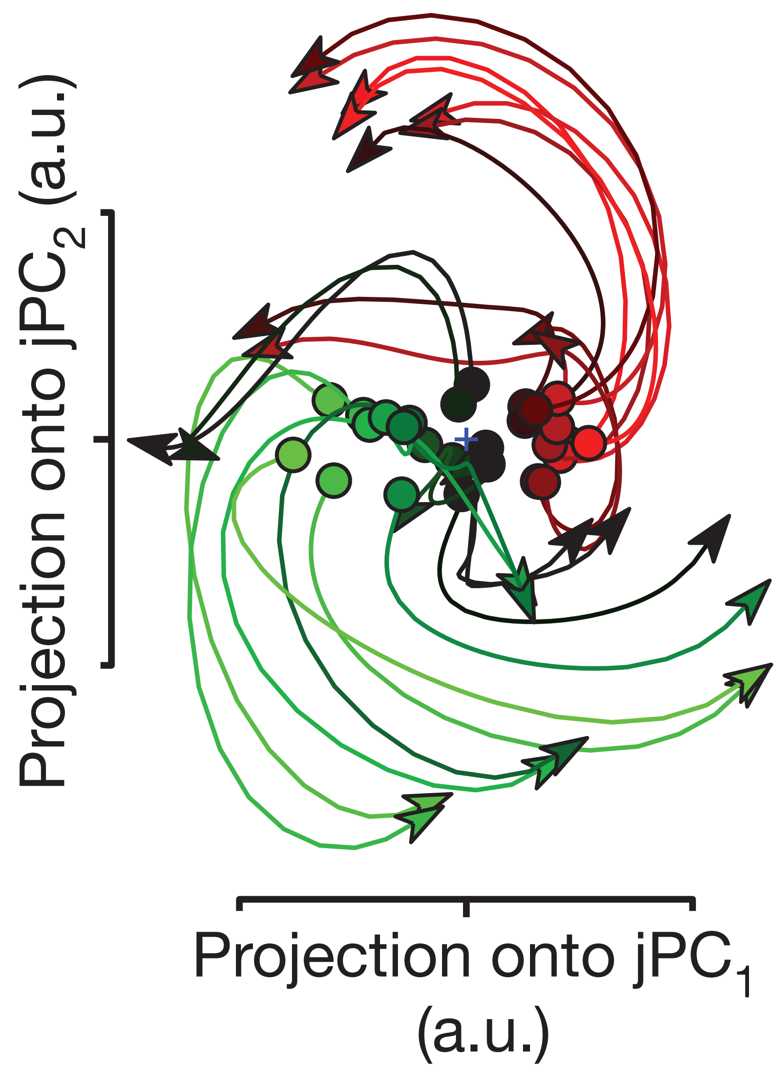
Dimensionality reduction can reveal the low-dimensional structure that's present among a population of neurons. (Patterns of correlation)

We can interpret this as the *latent variables*, and we can study them as if they are the information-carrying signal in the brain.

Geometrical and topological concepts offer powerful predictions.

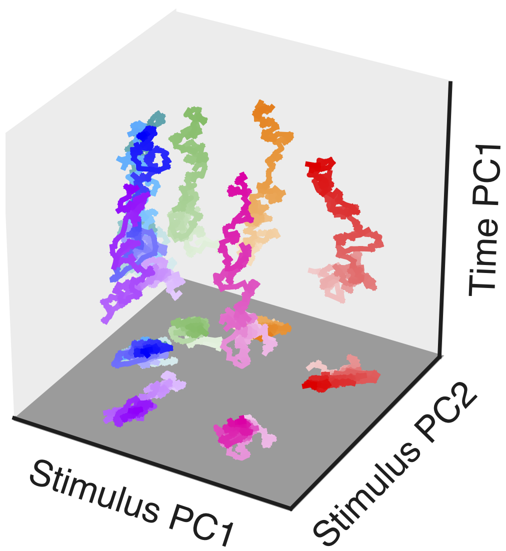
we’ve already seen this…

M1 codes *dynamics*

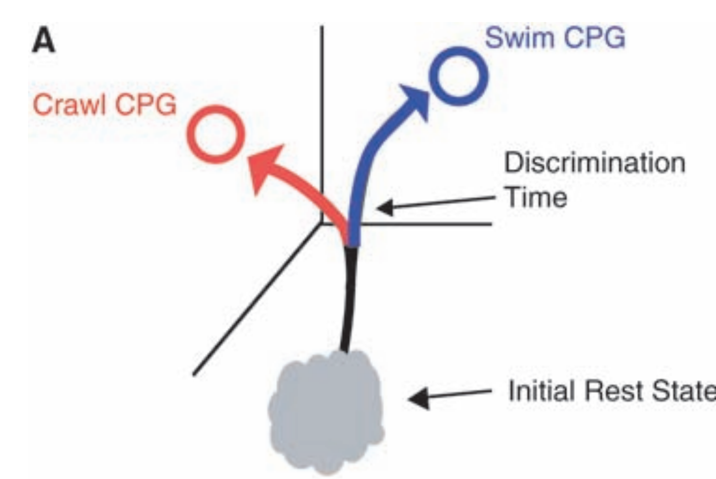


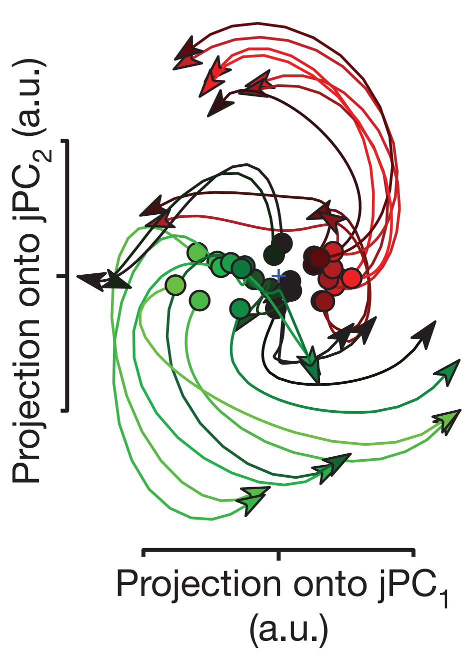
Churchland and Shenoy and colleagues, 2010s

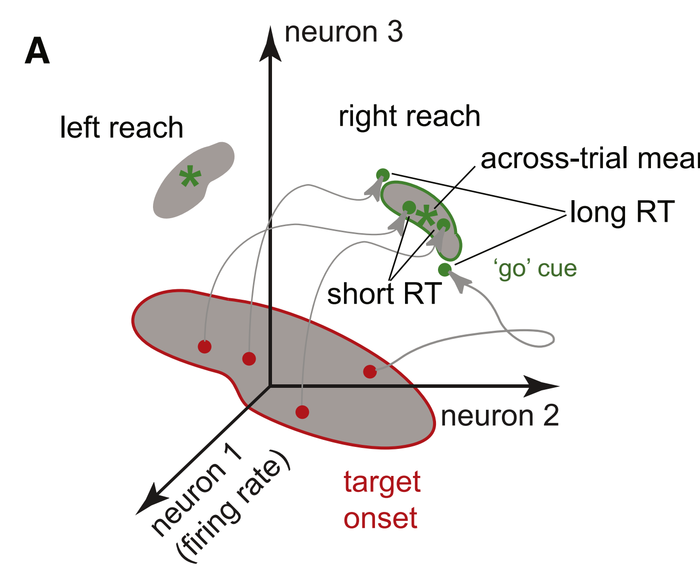
Low-Dimensional Structure is Everywhere

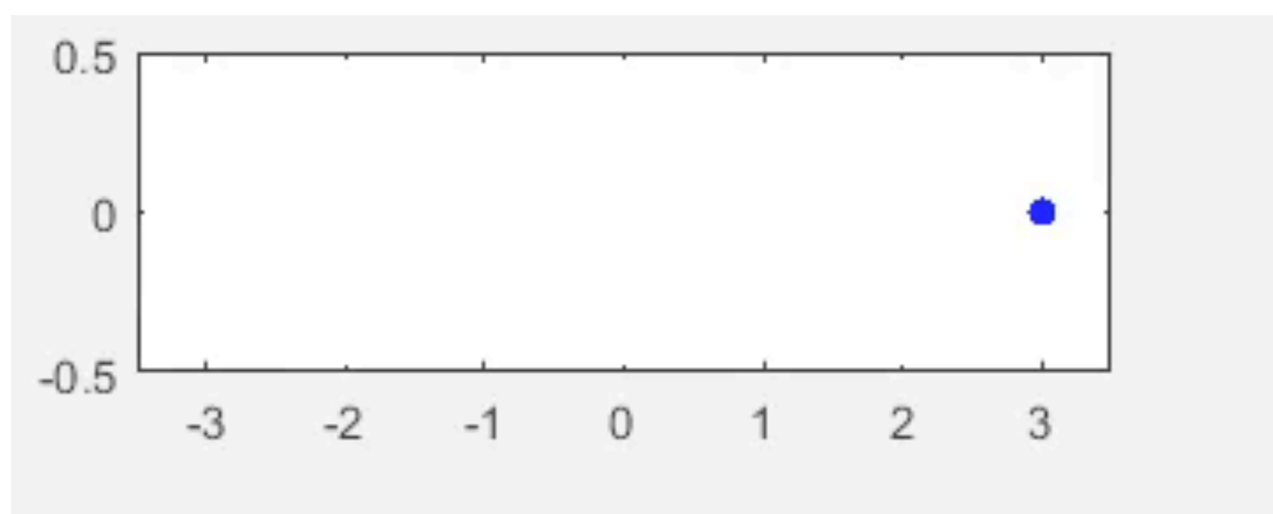


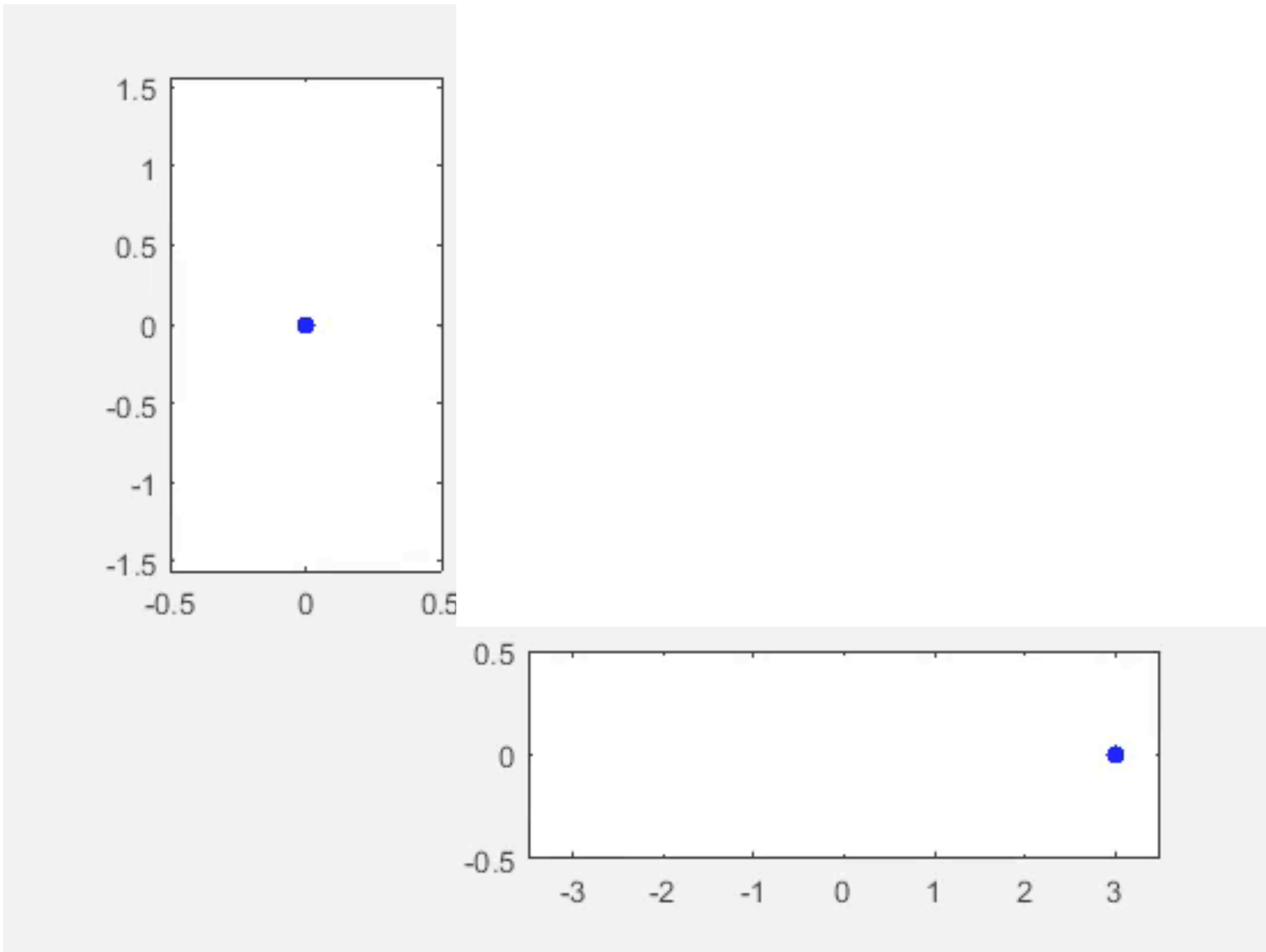
Briggman, Abarbanel, Kristan 2005
Broome, Jayaraman, Gilles Laurent, 2006
Machens, Romo, Brody 2010
Afshar, Yu,… Sahani, Shenoy 2011
Churchland, Cunningham, …. Shenoy 2012
Kaufman, … Shenoy 2014
Mante, Sussillo, Shenoy, Newsome 2014
Law and Schieber 2014
Murray, … Constantinidis, XJ Wang 2017
Remington, … Jazayeri 2018
Ni, Ruff, … Marlene Cohen 2018
Perich, Gallego, Lee Miller 2018
…

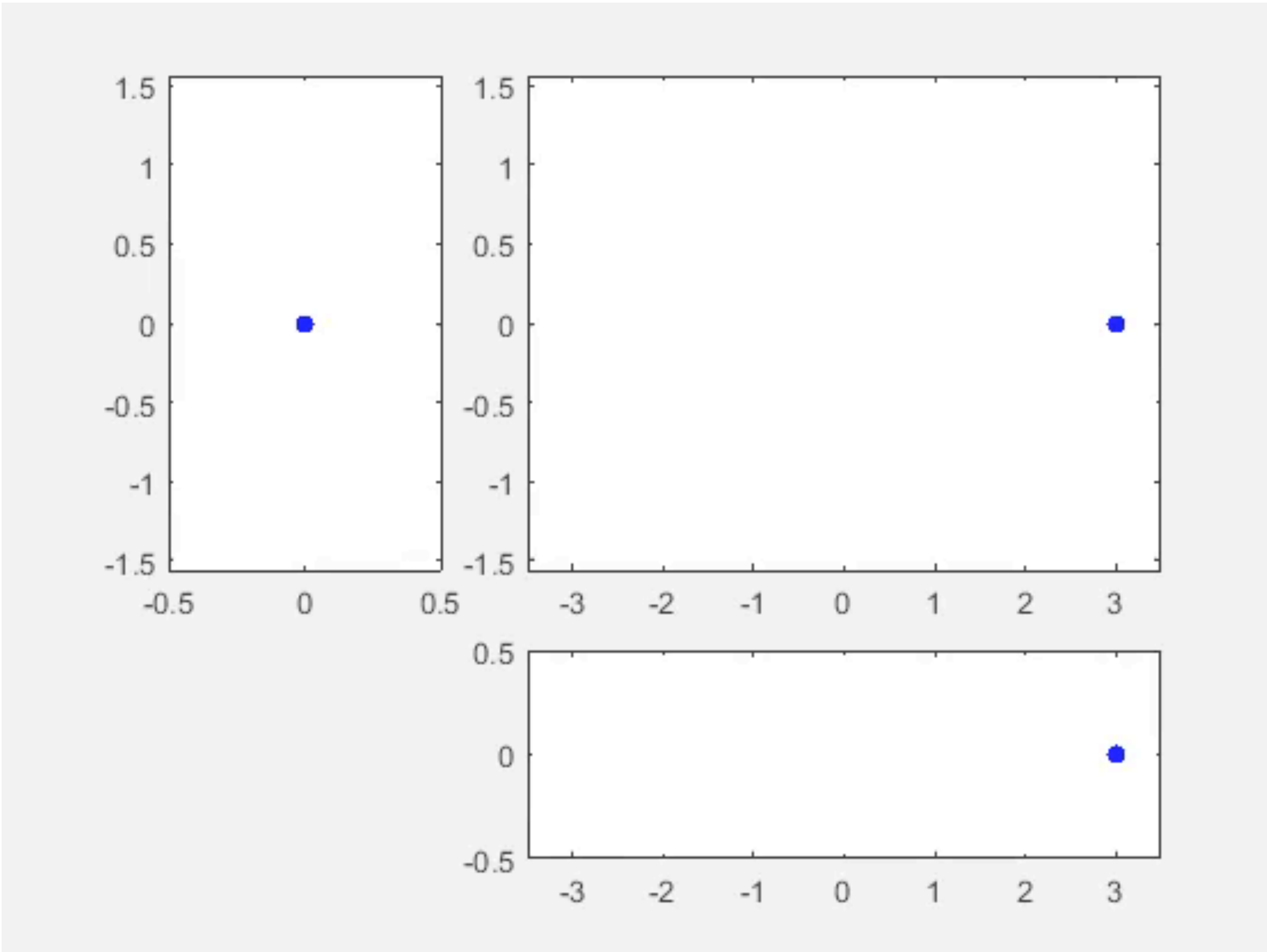







0.5
0
-0.5
-3
-2
-1
0
1
2
3

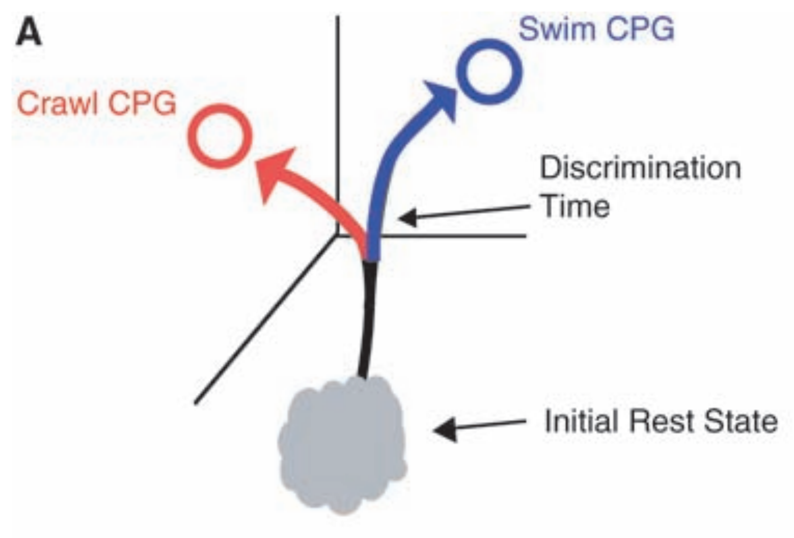

1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5
0.5
0
-0.5
-3
-2
-1
0
1
2
3

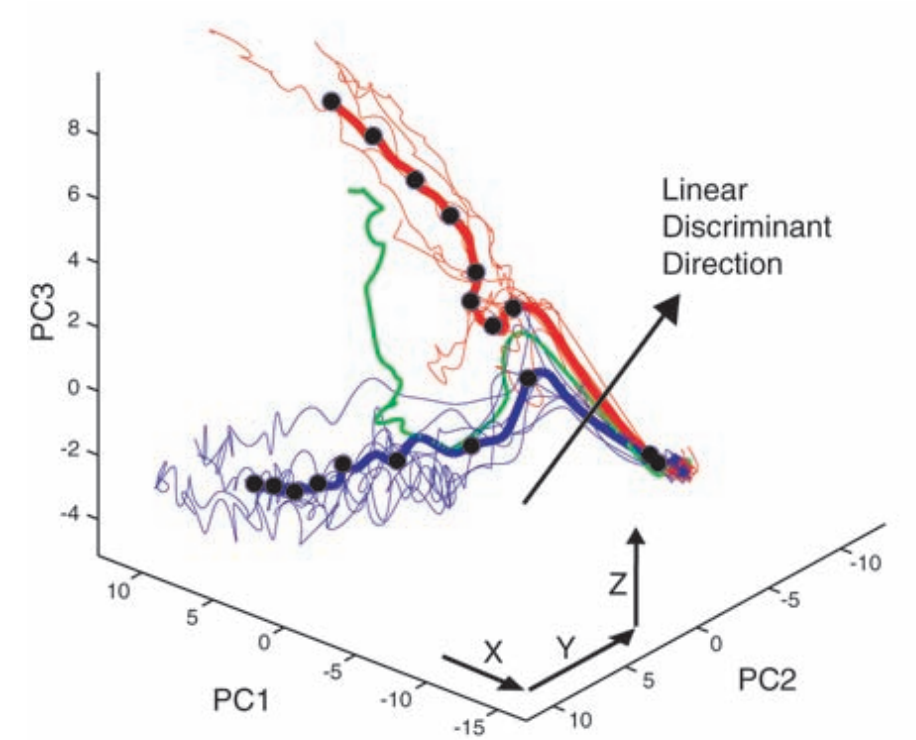

1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5
1.5
1
0.5
0
-0.5
-1
-1.5
-3
-2
-1
0
1
2
3
0.5
0
-0.5
-3
-2
-1
0
1
2
3

Optical Imaging of Neuronal Populations During Decision-Making

K. L. Briggman,[1] H. D. I. Abarbanel,[2,3] W. B. Kristan Jr.[1*]

11 FEBRUARY 2005 VOL 307 SCIENCE



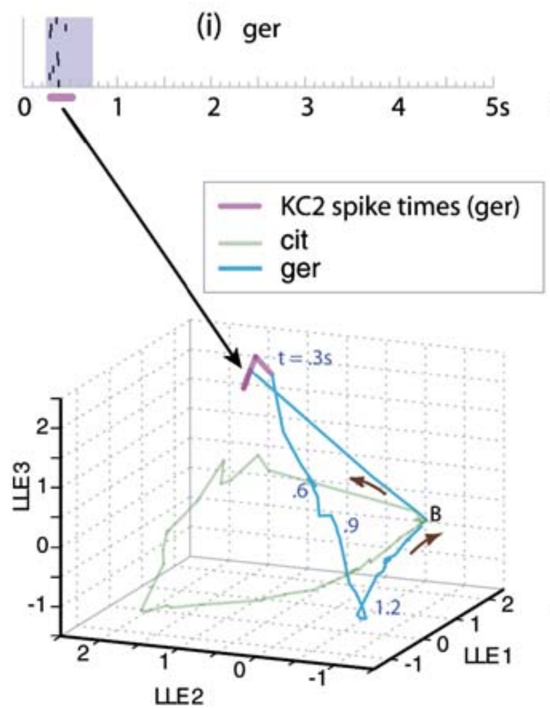


Neuron *51*, 467–482, August 17, 2006 ©2006 Elsevier Inc. DOI 10.1016/j.neuron.2006.07.018

Encoding and Decoding of Overlapping Odor Sequences

Bede M. Broome,[1,2] Vivek Jayaraman,[1,2]
and Gilles Laurent[1,*]

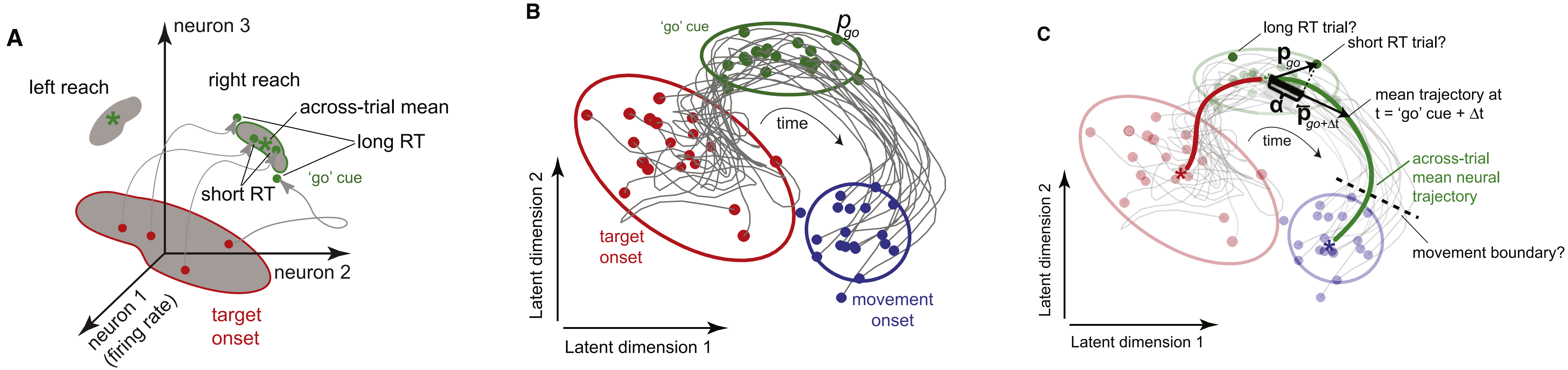
Projection neurons fire broadly, but their target, kenyon cells, fire sparsely.



Single-Trial Neural Correlates of Arm Movement Preparation

Afsheen Afshar,[1,2] Gopal Santhanam,[1] Byron M. Yu,[1,5] Stephen I. Ryu,[1,6] Maneesh Sahani,[1,5,7]
and Krishna V. Shenoy[1,3,4,7,*]

Neuron 2011



What we covered today:

- Neural Engineering, including BCI
- Arm movement control: Behavior and Theories
- What M1 does (still a work in progress.)
- The <u>population revolution</u> is underway.

Outline

- Neurophysiology of motor control
- Brain-Computer Interface algorithms
- Population methods in basic neuroscience
- New directions in BCI therapeutics

Brain-Computer Interfaces: Therapeutics and Basic Science

Aaron Batista
aaron.batista@pitt.edu

30 September 2021

Bio Engineering

CNBC
Center for the Neural Basis of Cognition

Additional Slides

The Reference Frame paradigm

What are individual neurons “tuned for”?

The eye-to-hand reference frame transformation consists of:

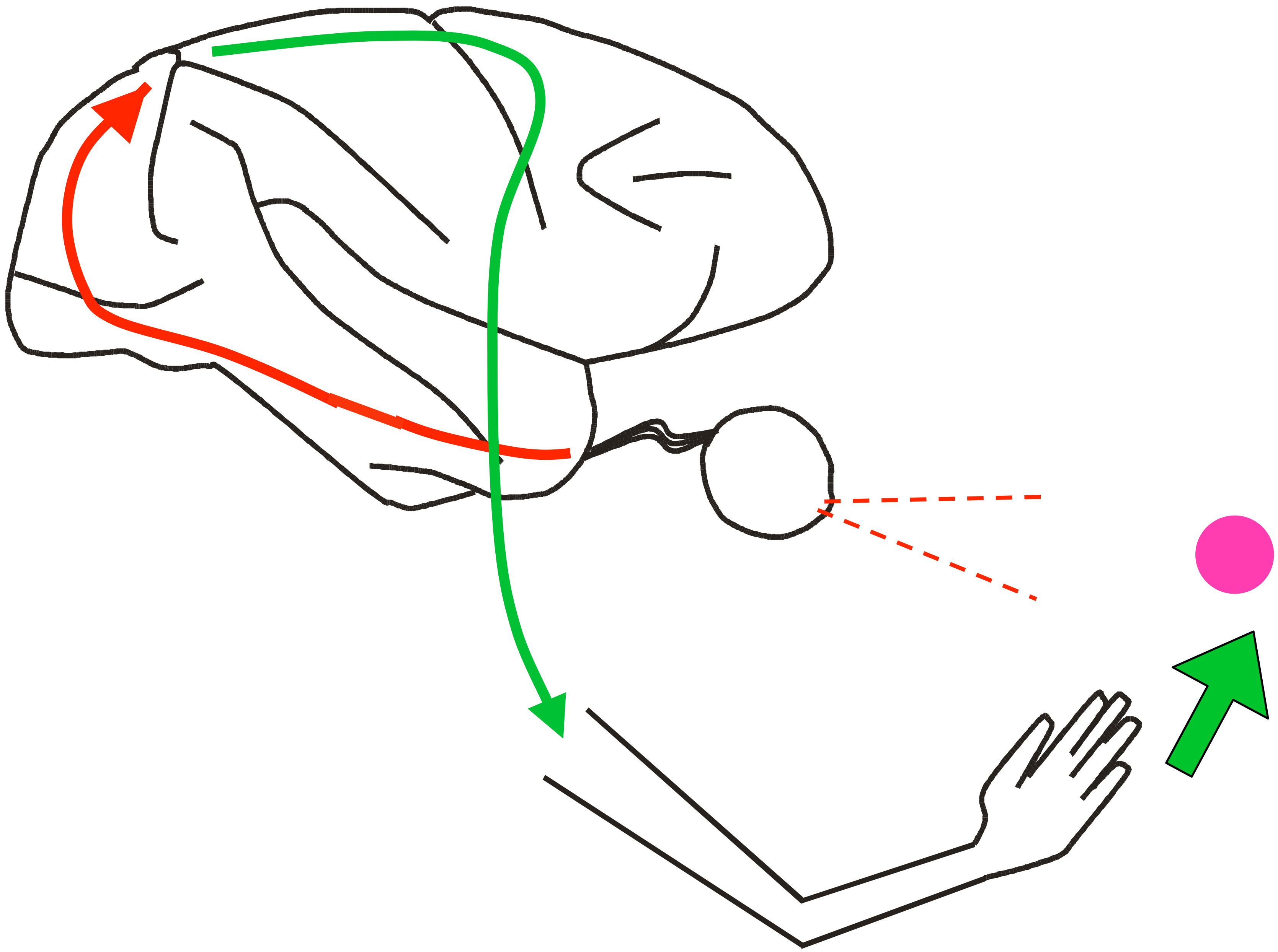
- visual receptive fields…
- combine with proprioception…
- to yield limb-centered tuning…
- which drives the muscles!

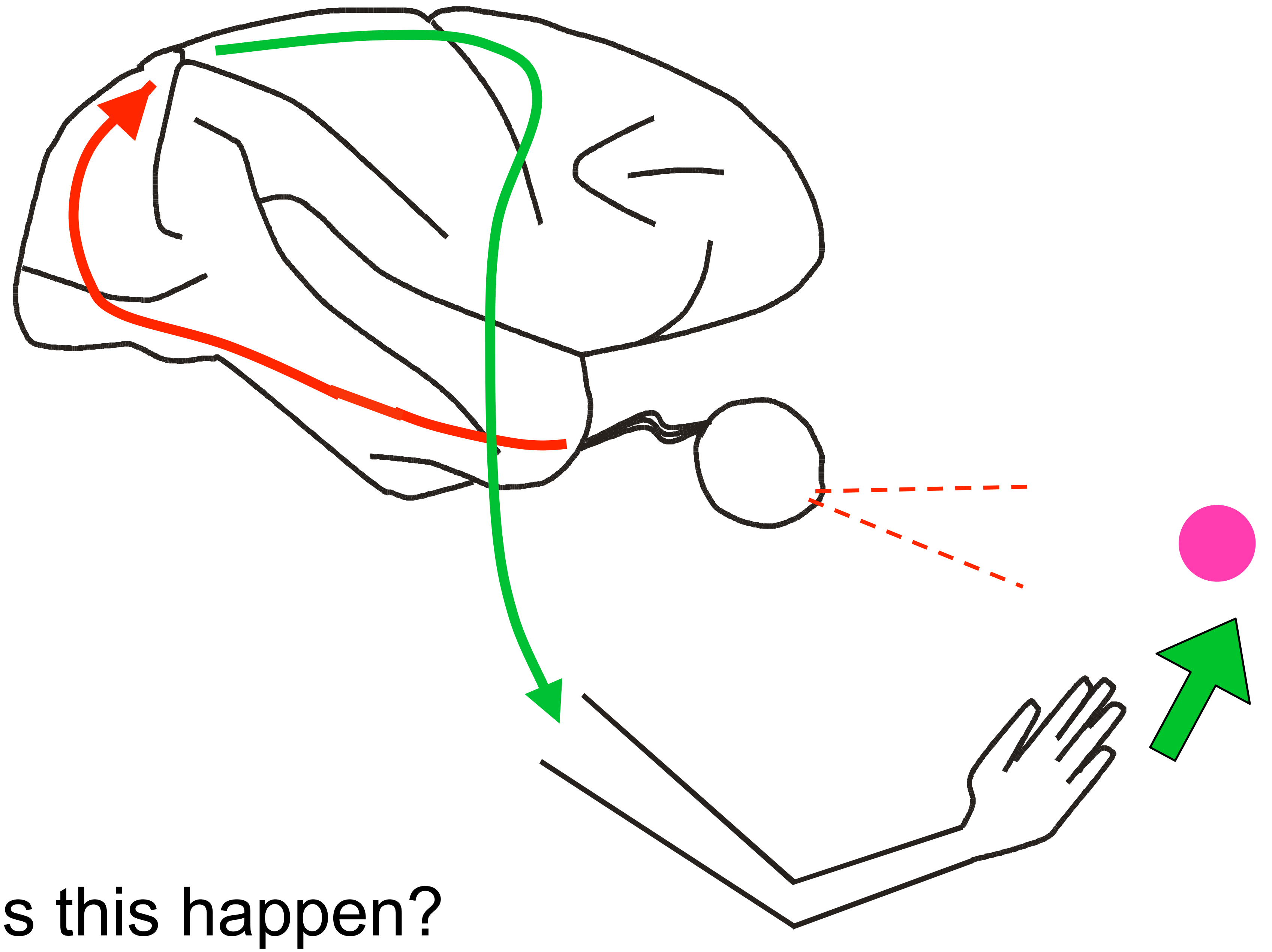
Goal-Directed Reaching Requires a Reference Frame Transformation

Goal-Directed Reaching Requires a Reference Frame Transformation

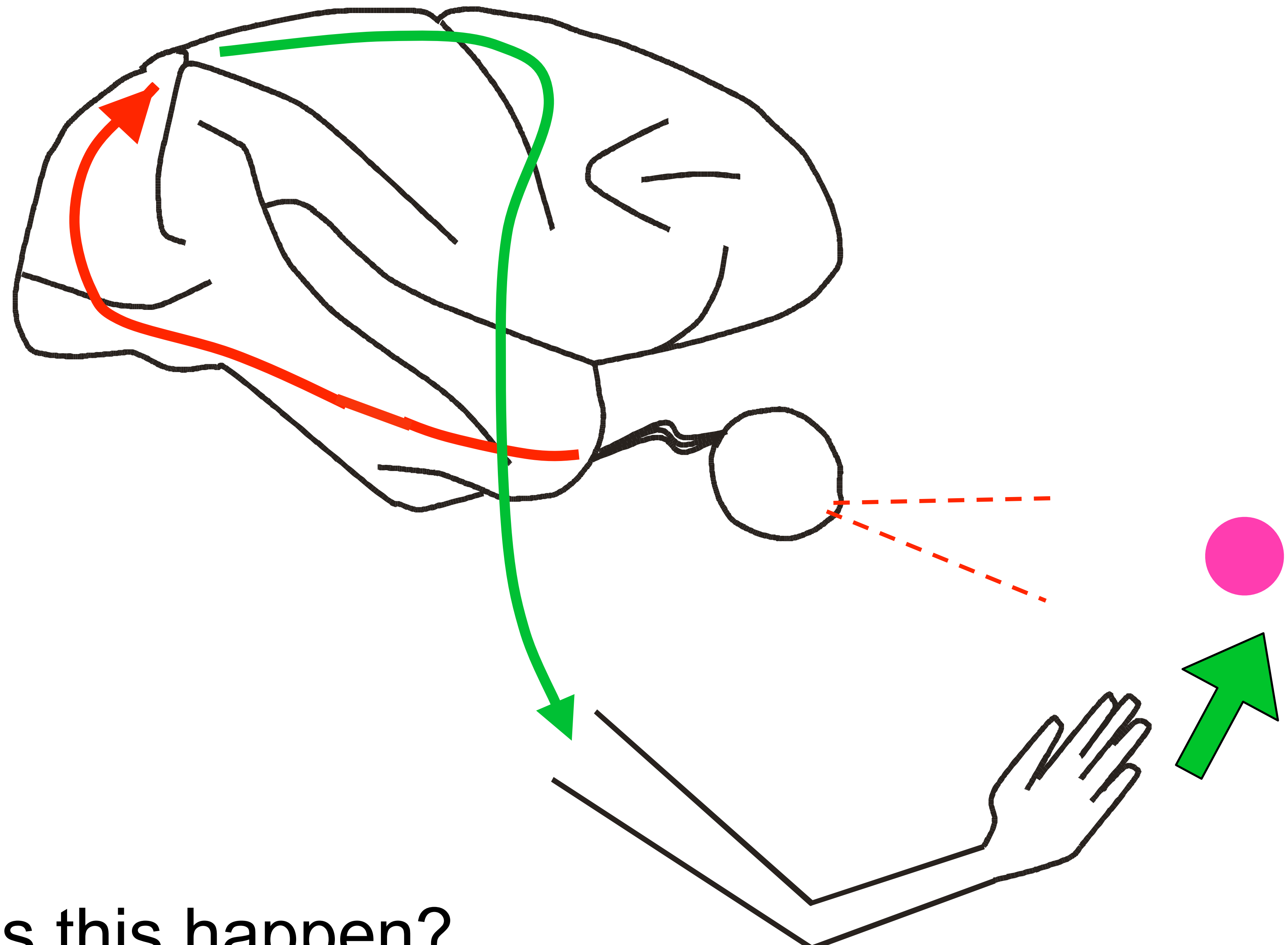
Goal-Directed Reaching Requires a Reference Frame Transformation

Goal-Directed Reaching Requires a Reference Frame Transformation



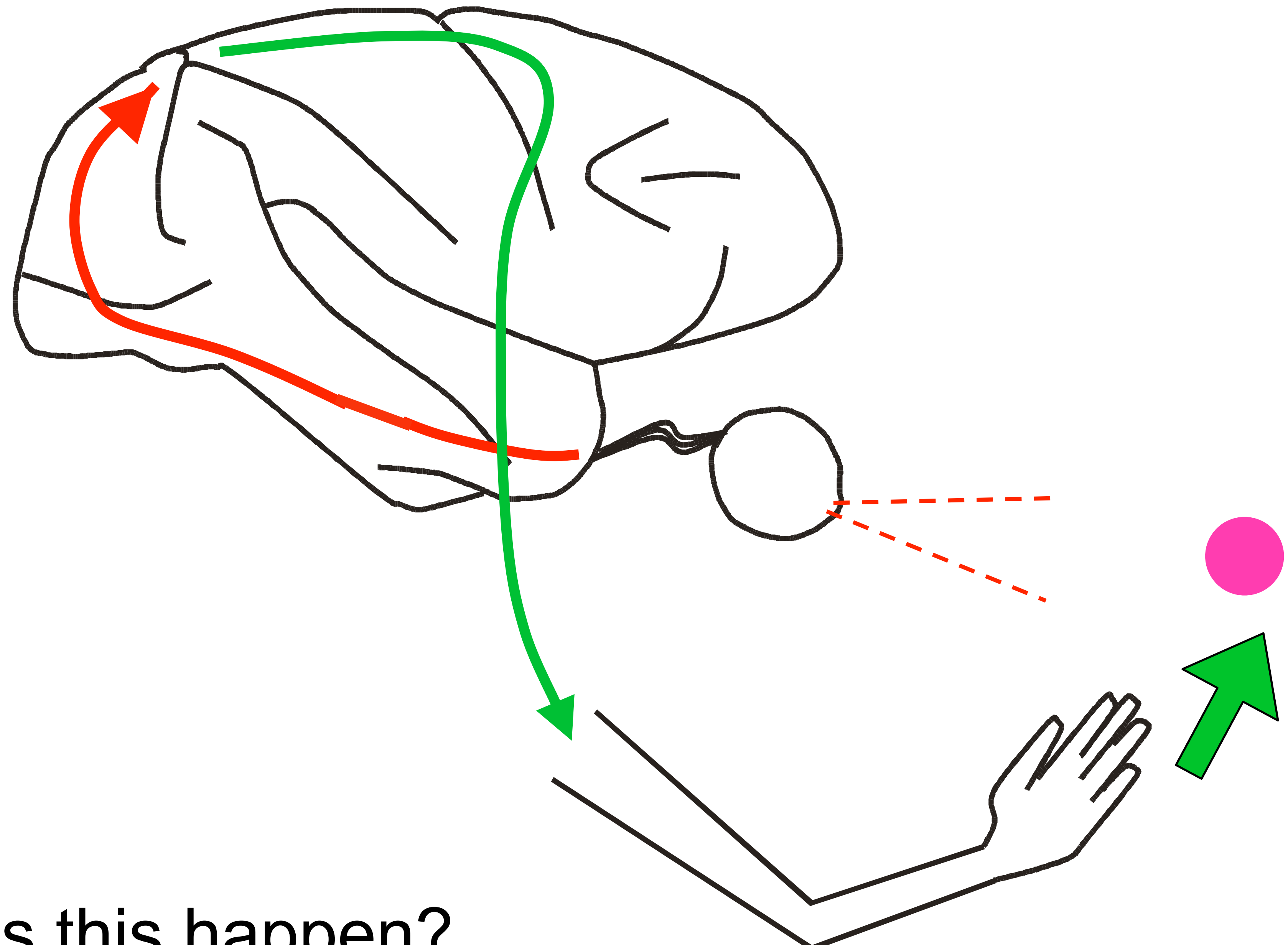


How does this happen?



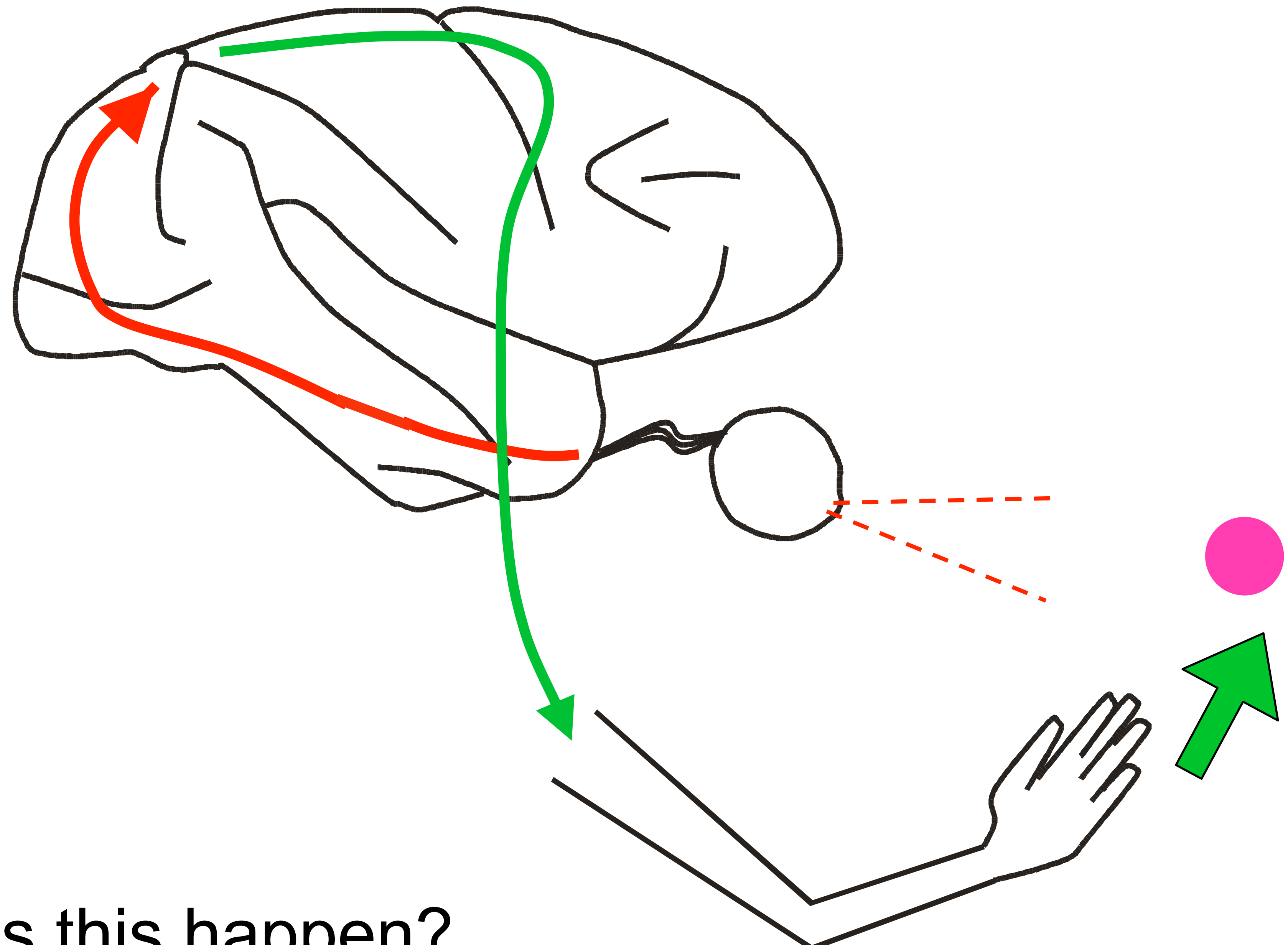
How does this happen?

Retina → LGN → V1 → PPC → PMd → M1 → Spine → Muscles



How does this happen?

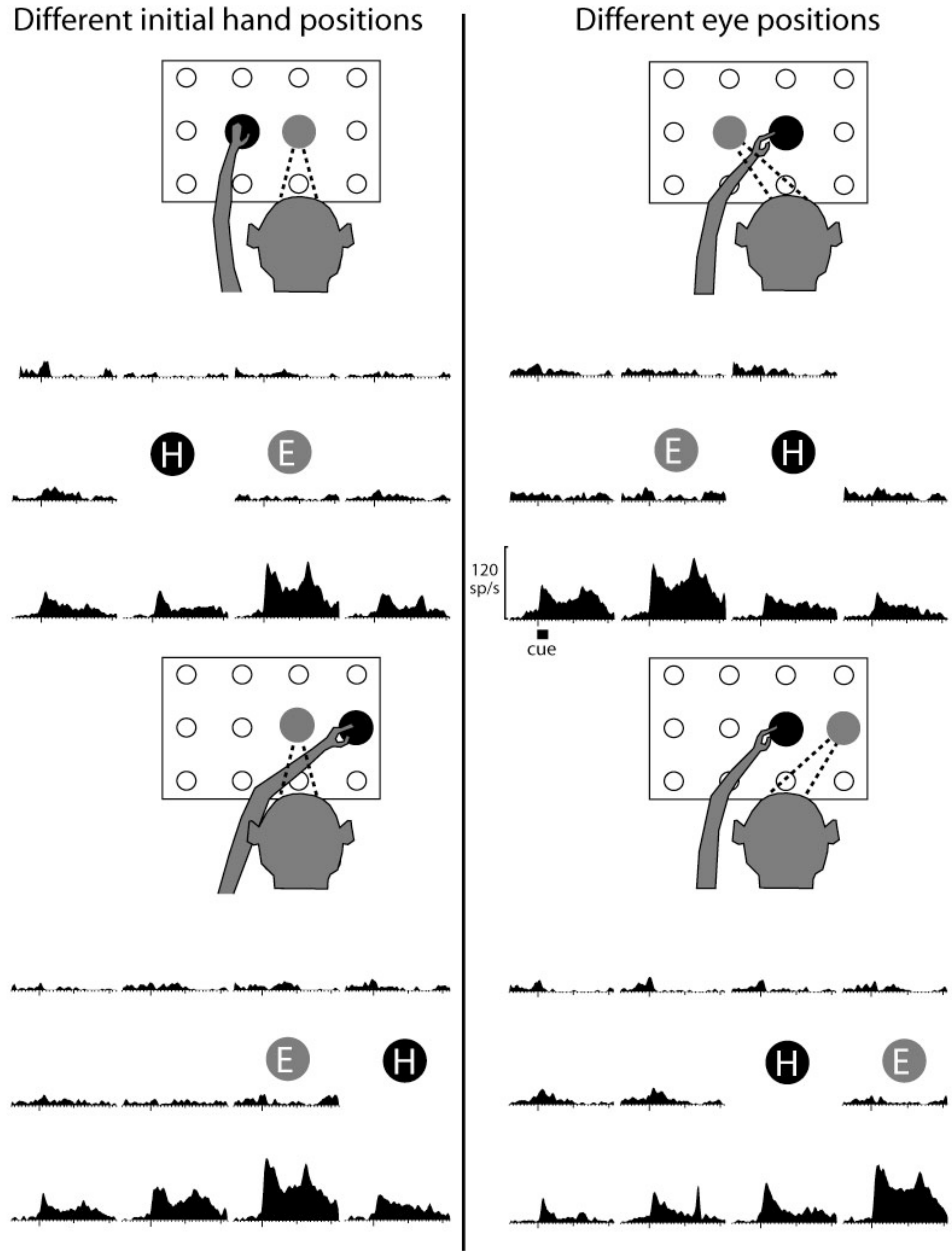
Retina → LGN → V1 → PPC → PMd → M1 → Spine → Muscles



How does this happen?

Retina → LGN → V1 → PPC → PMd → M1 → Spine → Muscles

Visual information → Muscular contractions


Different initial hand positions
Different eye positions
H
E
120
sp/s
cue

The Reference Frame paradigm

What are individual neurons “tuned for”?

The eye-to-hand reference frame transformation consists of:

- visual receptive fields…
- combine with proprioception…
- to yield limb-centered tuning…
- which drives the muscles!

This is a feedforward, single-neuron, mean-firing-rate view of the process.