A Comparison of Decoding Models for Imagined Motions from Human Motor Cortex

Sung-Phil Kim, Ph.D.[1], John D. Simeral, Ph.D.[2,4], John P. Donoghue, Ph.D.[2,3], Leigh R. Hochberg, M.D., Ph.D.[2,4,5], Gerhard M. Friehs, M.D.[6], Jon A. Mukand, M.D., Ph.D.[7], David Chen, M.D.[8], and Michael J. Black, Ph.D.[1]

[1]Computer Science, Brown Univ, Providence, RI; [2]Neuroscience, Brown Univ, Providence, RI; [3]Cyberkinetics Neurotechnology Systems, Inc., Foxborough, MA; [4]Rehab R&D, Dept. Veterans Affairs, Providence, RI; [5]Massachusetts General Hospital, Harvard Medical School, Boston, MA, [6]Clinical Neurosciences (Neurosurgery), Brown Univ, Providence, RI; [7]Physical Medicine & Rehab., Sargent Rehab. Center, Providence, RI; [8]Spinal Cord Injury, Rehab. Institute of Chicago, Chicago, IL;

256.11

Introduction

Objective:
- Enable reliable neural-motor prosthetic (NMP) cursor control in tetraplegic humans.

Background:
- Initial results from the first trial participant were recently reported (Hochberg, et al., 2006, Nature)
- Encoding model assumed cursor position was a linear function of firing rates.
- Decoding was performed using a linear filter.

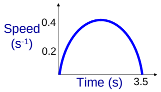
Observations:
- Direction, velocity, and force as well as position are encoded in spatio-temporal patterns of neuronal ensemble activity in monkeys (Paninski et al., 2004).
- In animal studies, a Kalman filter showed improved on-line cursor control when compared with the linear filter (Wu et al., EMBS 2004).

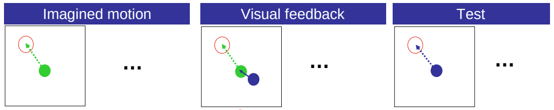
Hypothesis:
- Decoding velocity with the Kalman filter can improve NMP cursor control.

Experimental Design

- FDA Investigational Device Exemption*.
- See poster 256.10 for clinical details.
- Filter building and testing process:

Speed profile of training cursor

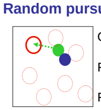




- Linear filter (position) = 4 x 1min open-loop blocks + 4 x 1min closed-loop blocks
- Kalman filter (velocity) = 1 x 1min open-loop blocks + 4 x 1min closed-loop blocks
- Minimum 6~8 movements per block

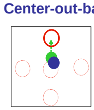
- Training task:

Random pursuit-tracking task



One target at a time
Random target locations
For position decoder

Center-out-back pursuit-tracking task

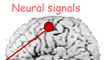


One target at a time
Fixed target locations
For velocity decoder

Decoding Methods

- Direct decoding: directly estimate $E[\vec{x}_t \mid \vec{Z}_{t-j}]$

$$\vec{x}_k = f(\vec{z}_k, \vec{z}_{k-1}, \ldots) \xrightarrow{f(\cdot)\text{ is linear}} \begin{array}{l} x_k = \vec{f}_1^{\,T} \vec{z}_{k:k-d} \\ y_k = \vec{f}_2^{\,T} \vec{z}_{k:k-d} \end{array}$$



$$p(\vec{x}_t \mid \vec{Z}_{t-j}) = p(\vec{x}_t \mid \vec{z}_{t-j}, \ldots, \vec{z}_1)$$

- Generative decoding: recursively estimate $E[\vec{x}_t \mid \vec{Z}_{t-j}]$ using Bayesian model

$$p(\vec{x}_t \mid \vec{Z}_{t-j}) = \kappa \underbrace{p(\vec{z}_{t-j} \mid \vec{x}_t)}_{\text{likelihood}} \underbrace{p(\vec{x}_t \mid \vec{Z}_{t-j-1})}_{\text{prior}}$$

$$p(\vec{x}_t \mid \vec{Z}_{t-j-1}) = \int \underbrace{p(\vec{x}_t \mid \vec{x}_{t-1})}_{\text{temporal prior}} \underbrace{p(\vec{x}_{t-1} \mid \vec{Z}_{t-j-1})}_{\text{posterior at } t-1} d\vec{x}_{t-1}$$

- Kalman filter assumes linear Gaussian likelihoods and priors

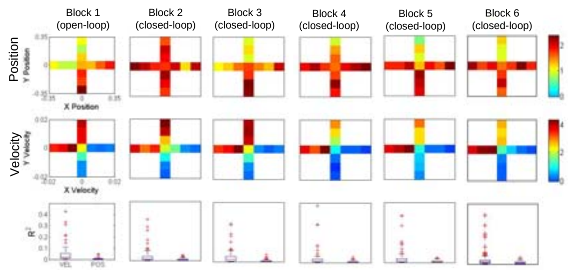
*CAUTION: Investigational device. Limited by Federal (USA) law to investigational use – only being studied in USA.

Training, Tuning and Learning

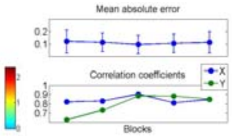
- Tuning of MI cells during training (See Paninski et al. 2004 for methods).

Kalman filter (velocity) trained with center-out-back pursuit-tracking task

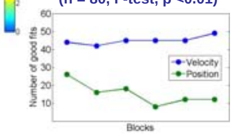
Kinematic tuning for one unit during training



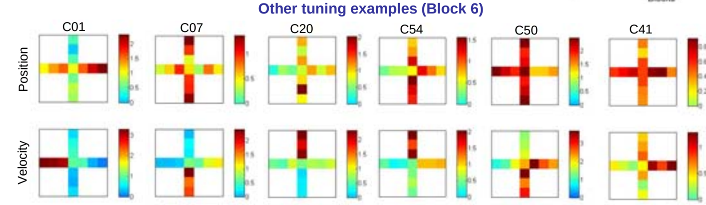
Accuracy during training



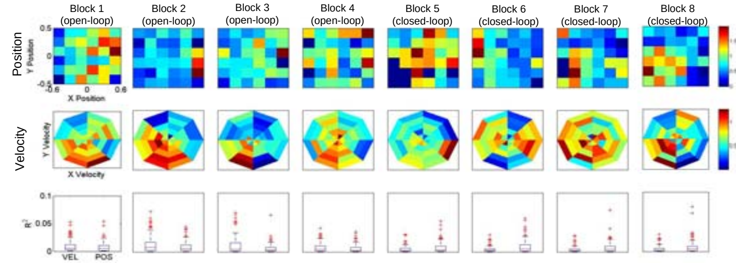
Number of tuned units (n = 80, F-test, p <0.01)



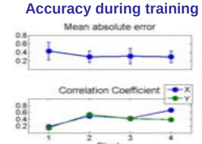
Other tuning examples (Block 6)



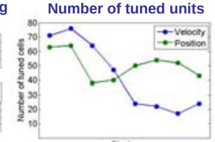
Linear filter (position) trained with random pursuit-tracking task



Accuracy during training

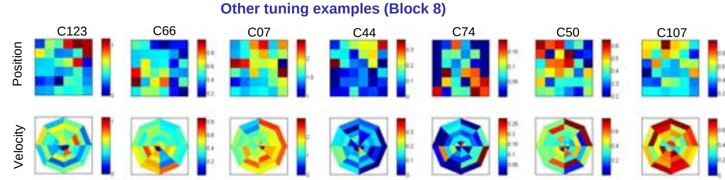


Number of tuned units



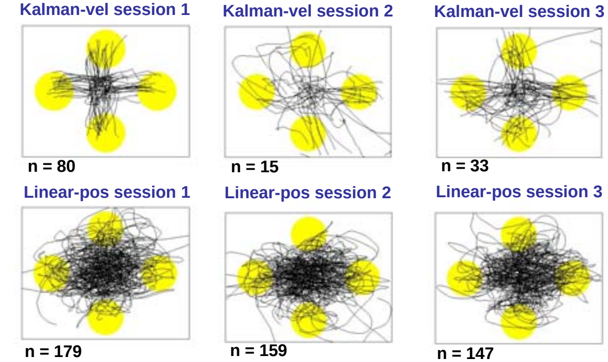
(n = 147, p <0.01)

Other tuning examples (Block 8)



Performance

Trajectory of the neural cursor for radial tasks



Target acquisition performance

Filter Type		# Units	Radial Task: Up	Down	Right	Left	Avg
Linear position filter	Session 1	179	85%	50%	80%	80%	78%
	Session 2	159	45%	60%	85%	100%	73%
	Session 3	147	95%	60%	100%	85%	85%
Kalman velocity filter	Session 1	80	100%	95%	100%	90%	96%
	Session 2	15	90%	58%	100%	90%	84%
	Session 3	33	95%	88%	100%	100%	96%

Conclusions

- Velocity decoding model provides more accurate cursor control than position decoding model (from 79% to 92% for radial target acquisition task).
- Velocity decoding model yields accurate control with fewer neuronal units (from ~150 units to ~50 units).
- Velocity decoding model can be trained faster than position decoding model.
- Training paradigm should match kinematic state (position or velocity).
- Preliminary studies show kinematic state used in training affects tuning of human MI cells.
- Current study
  - Effect of decoding algorithms:
    e.g. Linear velocity filter vs. Kalman position filter.
  - Decoding "click" state along with kinematics (see movie)
  - Performance evaluation: e.g. bit rate based on Fitts' law.

References:

Paninski, L., Fellows, M., Hatsopoulos, N., and Donoghue, J. P. (2004). "Spatiotemporal tuning of motor cortical neurons for hand position and velocity." J. Neurophys., 91, 515–532.

Wu, W., Shaikhouni, A., Donoghue, J. P., Black, M. J., "Closed-loop neural control of cursor motion using a Kalman filter," Proc. EMBS, pp. 4126-4129, Sept. 2004.

Support gratefully acknowledged: NIH/NINDS, VA, ONR, DARPA, Cyberkinetics Neurotechnology Systems, Inc.
Disclosures: LH: Clinical trial support, Cyberkinetics (CKI); JM: Consultant, CKI; GF: Consultant, stock holdings, CKI; JD: Chief Scientific Officer, salary, stock holdings, director, CKI.