Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Optimal Target Placement for Neural Communication Prostheses

John P. Cunningham*, Byron M. Yu*, Krishna V. Shenoy*†
*Department of Electrical Engineering, †Neurosciences Program,
Stanford University, Stanford, CA 94305, USA

***Abstract*— Neural prosthetic systems have been designed to estimate continuous reach trajectories as well as discrete reach targets. In the latter case, reach targets are typically decoded from neural activity during an instructed delay period, before the reach begins. We have recently characterized the decoding speed and accuracy achievable by such a system. The results were obtained using canonical target layouts, independent of the tuning properties of the neurons available. Here we seek to increase decode accuracy by judiciously selecting the locations of the reach targets based on the characteristics of the neural population at hand. We present an optimal target placement algorithm that approximately maximizes decode accuracy with respect to target locations. Using maximum likelihood decoding, the optimal target placement algorithm yielded up to 11 and 12% improvement for two and sixteen targets, respectively. For four and eight targets, gains were more modest (5 and 3%, respectively) as the target layouts found by the algorithm closely resembled the canonical layouts. Thus, the algorithm can serve not only to find target layouts that outperform canonical layouts, but it can also confirm or help select among multiple canonical layouts. These results indicate that the optimal target placement algorithm is a valuable tool for designing high-performance prosthetic systems.**

***Index Terms*— Brain-machine interface, neural coding and decoding, motor control, pre-motor cortex, Kullback-Leibler divergence.**

I. Introduction

Most current neural prostheses decode neural activity into continuous commands, which guide either a smoothly moving on-screen cursor or robotic arm to choose a visual reach target of interest. When the target represents a discrete action, such as selecting a key on a keyboard, the prosthesis is serving as a communication prosthesis. Some communication prostheses estimate just the intended target, which has certain advantages previously discussed [1], [2]; here it was shown that intended reach targets can be reliably decoded from the motor *plan*. In the experimental paradigm, a subject is trained to make center-out, delayed reaches to one of a discrete number of visual targets presented on a fronto-parallel screen (Fig. 1). Using neural activity recorded from dorsal pre-motor (PMd) cortex before the onset of movement, during the *instructed delay period*, maximum likelihood (ML) decoding algorithms can choose the intended reach target with high speed and accuracy [1].

This work was supported by the Michael Flynn Stanford Graduate Fellowship (J.P.C.), NDSEG and NSF Graduate Research Fellowships (B.M.Y.), and the following awards to K.V.S.: Christopher Reeve Paralysis Foundation, NIH-CRCNS-R01, NSF Center for Neuromorphic Systems Engineering at Caltech, ONR, Whitaker Foundation, Center for Integrated Systems at Stanford, Sloan Foundation, and Burroughs Wellcome Fund Career Award in the Biomedical Sciences. Please address correspondence to jcunnin@stanford.edu.

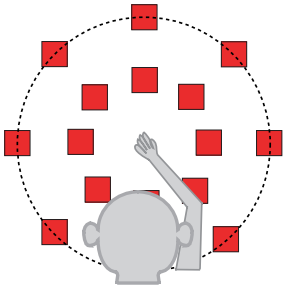


Fig. 1. Sixteen target paradigm. Red squares show placement of targets in canonical ring topology, evenly spaced around two rings of eight targets each. Dotted line indicates the workspace bound.

There is a great deal of interest in improving the decode performance of these systems. Here we address the problem of target placement. Since a communication prosthesis consists of a keyboard or some other user interface, the key or target layout can be physically configured as the system designer sees fit. Conventional communication prostheses place a number of targets (typically two to sixteen) evenly spaced around one or two rings, the radius of which is determined by the subject's reach extent (Fig. 1). This canonical target layout, known as the *ring topology*, reflects the observation that neural activity is more strongly modulated by reach direction than reach extent [3], [4]. However, if we understand the tuning properties of the particular neurons from which we are recording, we can exploit this prior knowledge to place targets in a configuration that will yield lower decode error. Thus, our goal here is to increase decode accuracy by placing targets optimally; to our knowledge, this problem has not yet been investigated. In this study, we use an approach based on Kullback-Leibler divergence to provide a constellation of optimal target placements. We describe the basic approach and then compare the decode performance of the optimal constellation with canonical ring topologies.

II. Spike Count Model and Decoding

We define a constellation of targets χ as a collection of M targets placed at positions $\mathbf{x}_m \in \Re^2$ for $m \in \{1, \ldots, M\}$ (*e.g.* M=16 in Fig. 1). To model spike counts during the instructed delay period, we begin with a basic cosine tuning model [5]. Since firing rate must be non-negative, we use an exponential link function, which gives the per neural unit firing rate model

$$f_k(\mathbf{x}_m) = e^{\mathbf{c}_k^T \mathbf{x}_m + d_k} \quad (1)$$

where $f_k(\mathbf{x}_m)$ is the firing rate for unit $k \in \{1,\ldots,K\}$ during a fixed time window of duration Δ. Under this model, $\mathbf{c}_k \in \Re^2$ and $d_k \in \Re$ over all K neural units specify the neural population completely. We call $C \in \Re^{2\times K}$ the matrix with columns $\mathbf{c}_k$ and $\mathbf{d} \in \Re^{K\times 1}$ the vector of elements d_k. Our observed output is then a conditionally independent Poisson realization of these firing rates, which we write

$$y_k \mid \mathbf{x}_m \sim \text{Poisson}(f_k(\mathbf{x}_m)\Delta) \;\; \forall \; k = 1\ldots K \tag{2}$$

where y_k is the kth element of $\mathbf{y} \in \Re^{K\times 1}$, a vector of non-negative integer spike counts.

According to this generative model, on a given trial, the presented target $\mathbf{x}_{m^*}$ is chosen by the experimenter, where $m^* \in \{1,\ldots,M\}$. The observed spike counts $\mathbf{y}$ are then conditionally distributed according to (2), where $m = m^*$. From $\mathbf{y}$, we decode the presented target using maximum likelihood

$$\begin{aligned} \widehat{m} &= \underset{m}{\text{argmax}}\, p(\mathbf{x}_m \mid \mathbf{y}) && (3)\\ &= \underset{m}{\text{argmax}}\, \frac{p(\mathbf{y} \mid \mathbf{x}_m)p(\mathbf{x}_m)}{p(\mathbf{y})} && (4)\\ &= \underset{m}{\text{argmax}}\, p(\mathbf{y} \mid \mathbf{x}_m) && (5) \end{aligned}$$

where $\widehat{m}$ is the index of the estimated reach target $\mathbf{x}_{\widehat{m}}$. (4) is obtained using Bayes' rule; (5) is a result of all reach directions being equally likely, and $p(\mathbf{y})$ not being dependent on $\mathbf{x}_m$. If the modelling assumptions are satisfied, this decode rule will minimize the total error probability

$$P_{\text{error}} = \sum_{m=1}^{M} P(\{\mathbf{x}_{\widehat{m}} \neq \mathbf{x}_m\} \mid \{\mathbf{x}_{m^*} = \mathbf{x}_m\}) \; . \tag{6}$$

III. Optimal Target Placement Algorithm

A. General Problem

To optimize with respect to decode accuracy, we seek the constellation $\boldsymbol{\chi}$ that minimizes the overall probability of decode error (6). The general problem, well known in communications literature (see *e.g.*, [6]), is currently considered intractable. As a result, it is common to instead minimize the worst pairwise error probability [7]. Thus, with the Poisson noise in our problem (see (2)), we choose to solve the minimax problem

$$\boldsymbol{\chi}_{otp} = \underset{\boldsymbol{\chi}}{\text{argmin}}\left(\max_{m'\neq m} P\big(\{\mathbf{x}_{\widehat{m}} = \mathbf{x}_{m'}\} \mid \{\mathbf{x}_{m^*} = \mathbf{x}_m\}\big)\right) \tag{7}$$

over all trials. In other words, find the optimal target placement (OTP) constellation $\boldsymbol{\chi}_{otp}$ that minimizes the worst pairwise decode error between any two targets.

B. Pairwise Error Using Kullback-Leibler Divergence

For Poisson distributions, there is no closed-form expression for the probability of error between two parameterized distributions [8]. This problem arises when trying to calculate the probability of error between two distributions parameterized by different target locations $\mathbf{x}_m$. In lieu of a closed-form expression, Kullback-Leibler (KL) divergence has been used as a close proxy to pairwise error probability [7], [9], [10]. We motivate this relationship by returning to the two-target case of the ML decode rule (5) and rewriting it as

$$\mathbf{x}_{\widehat{m}} = \begin{cases} \mathbf{x}_m & \text{if } \frac{p(\mathbf{y}|\mathbf{x}_m)}{p(\mathbf{y}|\mathbf{x}_{m'})} \geq 1, \\ \mathbf{x}_{m'} & \text{otherwise.} \end{cases} \tag{8}$$

Assuming a trial is generated under $\mathbf{x}_m$, we want to maximize this likelihood ratio over all possible instances of $\mathbf{y}$. Doing so will provide the maximum distinguishability between the distributions and will miminize the chance that the likelihood ratio will be less than 1 (an error). We can equivalently maximize the logarithm of this likelihood ratio, and, taking the expectation, we have the KL divergence

$$KL(\mathbf{x}_m \parallel \mathbf{x}_{m'}) = E_{\mathbf{y}|\mathbf{x}_m}\left[\log\frac{p(\mathbf{y} \mid \mathbf{x}_m)}{p(\mathbf{y} \mid \mathbf{x}_{m'})}\right] \tag{9}$$

where the expectation is taken with respect to $\mathbf{y} \mid \mathbf{x}_m$. Under the Poisson output distribution we assume in (2), KL divergence can be calculated exactly:

$$\begin{aligned} &KL(\mathbf{x}_m \parallel \mathbf{x}_{m'}) = \\ &\Delta\sum_{k=1}^{K}\left(f_k(\mathbf{x}_{m'}) - f_k(\mathbf{x}_m) + f_k(\mathbf{x}_m)\log\frac{f_k(\mathbf{x}_m)}{f_k(\mathbf{x}_{m'})}\right). \end{aligned} \tag{10}$$

Note that this form is independent of the form of (1), allowing OTP to easily generalize for other, more complex firing rate models. In summary, we have the relationship

$$\begin{aligned} &\underset{\mathbf{x}_m,\mathbf{x}_{m'}}{\text{argmax}}\, KL(\mathbf{x}_m \parallel \mathbf{x}_{m'}) \approx \\ &\qquad \underset{\mathbf{x}_m,\mathbf{x}_{m'}}{\text{argmin}}\, P\big(\{\mathbf{x}_{\widehat{m}} = \mathbf{x}_{m'}\} \mid \{\mathbf{x}_{m^*} = \mathbf{x}_m\}\big). \end{aligned} \tag{11}$$

Finding ($\mathbf{x}_m$, $\mathbf{x}_{m'}$) that maximizes the KL divergence from $\mathbf{x}_m$ to $\mathbf{x}_{m'}$ is roughly equivalent to finding that which minimizes the probability of decoding $\mathbf{x}_{m'}$ when $\mathbf{x}_m$ was presented. For symmetric noise distributions, this relationship holds with equality [7]. For asymmetric distributions, such as the Poisson distribution used here, our simulations show that the relationship between error probability and KL divergence is nearly monotonic. While the Chernoff bound [11] proves an upper bound on error probability with respect to KL divergence in hypothesis testing, the bound has been found to be loose [9]. Though not a provable bound (upper or lower) on error probability, KL does provide a close proxy; supporting arguments in addition to that given above can be found in [9], [10].

C. Formulating a Sequential Quadratic Program

With this approximation, we can rewrite the minimax problem in (7) as

$$\boldsymbol{\chi}_{otp} = \underset{\boldsymbol{\chi}}{\text{argmax}}\left(\min_{m\neq m'} KL(\mathbf{x}_m \parallel \mathbf{x}_{m'})\right) \tag{12}$$

An algorithm solving this problem will push the $\mathbf{x}_m$ as far apart as possible from each other in terms of KL divergence.

The limitation of how far a subject's arm can reach is captured by a constraint γ on the Euclidean distance of $\mathbf{x}_m$ from the center of the workspace screen. We can rewrite this as a standard optimization problem in the variable $\boldsymbol{\chi}$

$$\begin{aligned} \max_{\boldsymbol{\chi},t} \quad & t^2 \\ \text{subject to} \quad & KL(\mathbf{x}_m \parallel \mathbf{x}_{m'}) \geq t^2 \quad \forall\ m \neq m' \\ & \|\mathbf{x}_m\| \leq \gamma \quad \forall\ m = 1 \ldots M \end{aligned} \tag{13}$$

Thus we have formulated the optimal target placement problem as a constrained (non-convex) optimization problem. We applied sequential quadratic programming (SQP) [7], [12] to convert (13) into a series of constrained quadratic programs, each of which locally approximates (13) at the current estimates of $\boldsymbol{\chi}$ and t. SQP does not guarantee convergence to the global optimum; the local optimum it finds depends on the choice of seed constellation $\boldsymbol{\chi}_0$. Thus, we solved the SQP multiple times (eight to thirty-two, depending on the number of targets in the constellation) starting at randomly chosen $\boldsymbol{\chi}_0$. The feasible solution corresponding to the largest objective t^2 was designated the optimal constellation $\boldsymbol{\chi}_{otp}$.

IV. Results

We used electrode arrays to record simultaneously from 189 neural units in PMd (experiment `H20041119`). The data was recorded on a sixteen target task, where the targets were oriented in two aligned rings. Spike counts were taken in bins of Δ=400 ms starting 150 ms after target onset. We fit C, $\mathbf{d}$ from real neural data by maximizing the data likelihood (2) taken across all 1008 trials. This problem is convex and can be readily solved using Newton's Method. The distribution of the fitted $\mathbf{c}_k$ is shown in Fig. 2.

For small numbers of targets and neural units, we can make a reasonable prediction about where the optimal targets should be placed. In the simplest case, we seek to place two targets optimally with only one neural unit. Given the preferred direction of the unit $\mathbf{c}_1$, the targets should be placed as far apart as possible (on the workspace bound) along the axis defined by $\mathbf{c}_1$. In this configuration, the presentation of one target elicits maximal firing, while the other target gives miminal firing. Indeed, the SQP approach to optimal target placement yields this result. Extending beyond this trivial case, the utility of OTP becomes apparent when looking at larger neural populations and larger numbers of targets.

With a population of neural units that are fairly uniform in their preferred directions and tuning strength, we imagine that the placement of four or eight targets will reduce effectively to a geometric problem, and placing the targets evenly around a ring will produce a near optimal result. We will validate this intuition below. When the number of targets grows larger, intuition breaks down: for example, with sixteen targets, should they be placed evenly around the workspace bound? Should they be placed in two rings; if so, how many targets in each ring? OTP gives answers to these questions. An example is shown in Fig. 2, where OTP returns a constellation (blue circles) with eleven targets

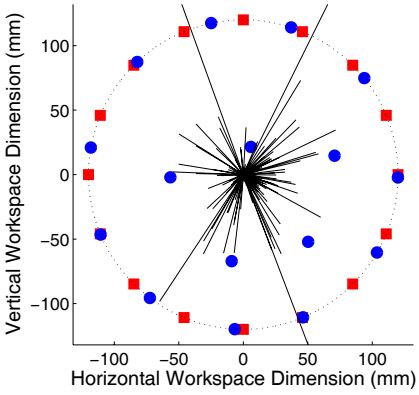


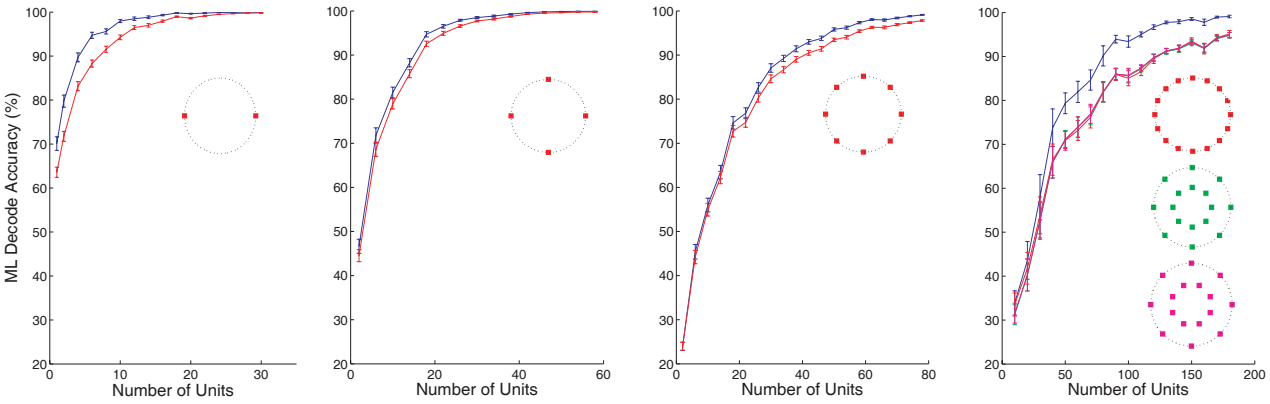
Fig. 2. Sixteen target placement example. Blue circles: OTP solution; red squares: canonical ring topology. Black lines show $\mathbf{c}_k$ (preferred direction and tuning strength; longer indicates sharper directional tuning) for each of 189 neural units. Workspace bound shown as a dotted line (γ=120 mm).

spaced roughly evenly around the workspace bound, and with five targets placed elsewhere in the workspace.

Having placed targets optimally, we investigated the performance of OTP vs. canonical ring topologies. We randomly drew a set of K units from the data set discussed in Section II, and we ran OTP to find the constellation $\boldsymbol{\chi}_{otp}$. We generated spike counts for 1000 trials to each of M targets according to (2). We then decoded using (5) and computed error using (6). We also ran 1000 trials for the canonical ring topology. To get a true average performance for the ring topology, we rotated the preferred directions during the trials and averaged the decode performance. Doing so introduces a rotation invariance to the ring topology, preventing the ring topology from being punished by a disadvantageous orientation with respect to a unit's preferred direction. This whole procedure was repeated 100 times (10 times for the sixteen target case, due to computational limitations) at each choice of K to create an average performance comparison.

These results are shown in Fig. 3 for two, four, eight, and sixteen targets. In Fig. 3(a), for two targets, OTP provides up to 11% improvement in decode accuracy (from 72% to 80% with K=2). As K grows and decode accuracy saturates to the performance ceiling of 100%, we expect the ring topology to approach the performance of OTP, and indeed we see this effect. In both the four and eight target cases, there is less improvement above the ring topology. This is not surprising: the OTP layouts closely resemble canonical ring topologies, so we would not expect a substantial performance difference.

At larger unit counts, we can again see substantial improvements offered by OTP. Fig. 3(d) illustrates the performance of the optimal configuration in the sixteen target case. We compare OTP to three canonical ring topologies: sixteen targets evenly spaced around the workspace bound, two radially aligned rings of eight targets each, and two radially staggered rings of eight targets each. Most OTP constellations seen in the sixteen target case (for different values of K and different sets of units drawn at random) place five interior targets and eleven on the workspace bound, as shown in Fig. 2. For fifty units, OTP yields 12% average



(a) Two Targets. (b) Four Targets. (c) Eight Targets. (d) Sixteen Targets.

Fig. 3. Comparison of Performance: Optimal target placements vs ring topologies. Blue and red lines show performance (mean $\pm$ s.e.m.) under OTP and a single ring topology, respectively. In (d), green and magenta lines show performance (mean $\pm$ s.e.m.) under the aligned and staggered double ring topologies, respectively (red, green, and magenta curves are highly overlapped). Insets show different ring topologies tested.

improvement over the canonical ring topologies.

V. Discussion

Across the range of unit and target counts tested, OTP outperforms each canonical ring topology, with performance gains of up to 12%. The results shown here are subject to three approximations, which we summarize here: *i)* in (7), we solve a minimax problem instead of a total probability of error problem; *ii)* we use KL divergence as a proxy for pairwise error probability in (11); and *iii)* we optimize a non-convex problem in (13) via a sequence of local quadratic approximations. Each of these approximations is necessary to put the problem in a tractable form and enables us to answer this previously unanswerable question. Although the impact of these approximations has yet to be fully characterized, their use allows us to achieve performance gains (cf. Fig. 3) that would not otherwise be possible.

While the parameters of the firing rate model (1) were fit to real neural data, the spike counts used to measure performance in Fig. 3 were generated from the model in (2). This step is necessary because we did not have real neural data from all possible target locations in the two-dimensional workspace. The performance for real neural data will depend on *i)* how well the firing rate model fits the neural data collected, and *ii)* how well the model generalizes to other target locations for which we have no neural data. Future neurophysiology experiments should assess the appropriateness of the firing rate model used, as well as the performance of OTP in a real experimental setting.

VI. Conclusion

The performance of neural communication prostheses can be improved by optimally placing the reach targets. We showed that we can find this optimal target constellation by converting what is believed to be an intractable problem into a tractable form. For four and eight targets, OTP layouts closely resembled canonical layouts, thus validating the canonical topology used in [1]. Also, we realized substantial decode performance improvements for two and sixteen target configurations across a wide range of unit counts.

Acknowledgments

We thank Maneesh Sahani, Gopal Santhanam, and George Gemelos for valuable technical discussions; and Stephen Ryu, Gopal Santhanam, and Afsheen Afshar for the neural data used to fit the firing rate model.

References

[1] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain computer interface," *Nature, in press*, 2006.

[2] N. Hatsopoulos, J. Joshi, and J. G. O'Leary, "Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles," *J Neurophysiol*, vol. 92, pp. 1165–1174, 2004.

[3] A. Riehle and J. Requin, "Monkey primary motor and premotor cortex: single-cell activity related to prior information about direction and extent of an intended movement," *J Neurophysiol*, vol. 61, pp. 534–549, 1989.

[4] J. Messier and J. F. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J Neurophysiol*, vol. 84, pp. 152–165, 2000.

[5] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex," *J Neurosci*, vol. 2, pp. 1527–1537, 1982.

[6] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice Hall, New Jersey, 1994.

[7] M. S. Gockenbach and A. J. Kearsley, "Optimal signal sets for non-Gaussian detectors," *SIAM Journal of Optimization*, vol. 9, no. 2, pp. 316–326, 1999.

[8] S. Verdu, "Asymptotic error probability of binary hypothesis testing for Poisson point-process observations," *IEEE Transactions on Information Theory*, vol. IT-32, no. 1, pp. 113–115, 1986.

[9] D. H. Johnson, C. M. Gruner, K. Baggerly, and C. Seshagiri, "Information-theoretic analysis of neural coding," *Journal of Computational Neuroscience*, vol. 10, pp. 47–69, 2001.

[10] D. H. Johnson and G. C. Orsak, "Relation of signal set choice to the performance of optimal non-Gaussian detectors," *IEEE Transactions on Communications*, vol. 41, no. 9, pp. 1319–1328, 1993.

[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[12] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, pp. 1–52, 1996.