Real-Time Decision Fusion for Multimodal Neural Prosthetic Devices

James Robert White[1]*, Todd Levy[2], William Bishop[2], James D. Beaty[2]

[1] Applied Mathematics and Scientific Computation Program, University of Maryland – College Park, College Park, Maryland, United States of America, [2] The Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland, United States of America

Abstract

Background: The field of neural prosthetics aims to develop prosthetic limbs with a brain-computer interface (BCI) through which neural activity is decoded into movements. A natural extension of current research is the incorporation of neural activity from multiple modalities to more accurately estimate the user's intent. The challenge remains how to appropriately combine this information in real-time for a neural prosthetic device.

Methodology/Principal Findings: Here we propose a framework based on *decision fusion*, i.e., fusing predictions from several single-modality decoders to produce a more accurate device state estimate. We examine two algorithms for continuous variable decision fusion: the Kalman filter and artificial neural networks (ANNs). Using simulated cortical neural spike signals, we implemented several successful individual neural decoding algorithms, and tested the capabilities of each fusion method in the context of decoding 2-dimensional endpoint trajectories of a neural prosthetic arm. Extensively testing these methods on random trajectories, we find that on average both the Kalman filter and ANNs successfully fuse the individual decoder estimates to produce more accurate predictions.

Conclusions: Our results reveal that a fusion-based approach has the potential to improve prediction accuracy over individual decoders of varying quality, and we hope that this work will encourage multimodal neural prosthetics experiments in the future.

Citation: White JR, Levy T, Bishop W, Beaty JD (2010) Real-Time Decision Fusion for Multimodal Neural Prosthetic Devices. PLoS ONE 5(3): e9493. doi:10.1371/journal.pone.0009493

Editor: Joel M. Schnur, George Mason University, United States of America

Received November 23, 2009; **Accepted** February 10, 2010; **Published** March 2, 2010

Funding: This work was funded by the Defense Advanced Research Projects Agency (DARPA) Revolutionizing Prosthetics program, contract N66001-06-C-8005. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: whitej@umd.edu

Introduction

Each year ~150,000 people in the United States undergo an arm or leg amputation [1]. An estimated 1.7 million amputees live in the United States [2] and millions more throughout the world. Reasons for limb loss range from physical trauma to infection to diseases such as diabetes and cancer. Regardless of the cause, the loss of a limb dramatically affects a person's life, making many simple tasks unbearably difficult. Over the past decade, prosthetic limbs have been developed to incorporate electrical signals from indirect muscles for user control – this is known as conventional prosthetic control. The emerging field of neural prosthetics goes further, interpreting the neural activity of the user for more intuitive control of prosthetic devices.

The problem of translating neural activity into direct movements is known as *neural decoding*. Types of recorded neural activity that can be decoded include cortical single-neuron action potentials (spikes) [3,4], local field potentials (LFPs) [5,6,7], and activity on the surface of the brain via electrocorticography (ECoG) [8,9,10,11,12], electromyography (EMG) [13], or electroencephalography (EEG) [14,15,16]. Each of these modalities offers particular advantages and limitations. For example, the surface-based EEG and ECoG recording platforms are relatively non-invasive, but provide poor spatial resolution (millimeters or centimeters). In contrast, spike signals provide accurate firing rates of single neurons, but this modality is highly invasive and prone to electrode failure [17,18,19]. While spike decoding is useful for predicting prosthetic endpoint trajectories, recent studies have demonstrated that modalities with less resolution are superior at encoding more general movement regimes [20,21].

Each modality involves specific hardware (e.g. electrodes), and analysis of these signals requires algorithms carefully designed to predict the user's intent given the characteristics of the signal (e.g. signal-to-noise ratio, noise distributions, dependencies). Neural decoding algorithms generate a *state estimate* as either a discrete classification (e.g. a gating classifier results in a decision for movement or no movement [22]) or a prediction of continuous variables (e.g. three-dimensional position and velocity estimates for the endpoint of a limb [23]). Moreover, some algorithms calculate confidence regions for state estimates, thereby providing additional information for the robotic controls interface.

Decoding of individual neural modalities is a consistently improving field with many robust methodologies. However, due to the limitations of current recording technologies, more advanced prosthetic limbs will require multiple neural signals with varying information content in order to achieve full functionality. A major computational challenge is to analyze all signals simultaneously to provide the best estimate of the user's desired movement.

Here we present a framework for combining information from multiple modalities to more accurately decode user intent for a prosthetic device. There are two solution paradigms for this problem: *data* fusion and *decision* fusion. Data fusion (low-level fusion) merges several raw signals prior to analysis, while decision fusion (high-level fusion) acts as a post-processor to merge the results of individual data analyses. Fusion frameworks have been shown to improve prediction accuracy in a wide range of fields including biometric identity confirmation [24,25,26], surface-to-air defense [27], robot navigation [28,29,30,31], image segmentation [32], and diagnosis of disease [33,34].

Though data fusion allows for all information to be assessed at once by a single algorithm, current hardware architectures for neural prostheses are parallelized with multiple recording platforms and processors, inherently advocating parallelized decoding prior to a final state prediction. As most decoding algorithms are optimized for specific modalities, we employ techniques for *decision fusion*, where we incorporate the estimates from each individual decoder into a single device state estimate.

In this report, we examine two algorithms for decision fusion of continuous variables: the Kalman filter and artificial neural networks (ANNs). We implemented three of the most successful individual neural decoding algorithms with simulated cortical neural spike data to test the capabilities of each fusion method. Through these simulations, we reveal the advantages and limitations of these approaches. Our methodology provides a flexible framework for fusing state estimates from decoding algorithms with different properties and hopefully will encourage multimodal experiments for improved control of sophisticated neural prosthetic devices.

Materials and Methods

The Kalman Filter for Decision Fusion

We first formulate decision fusion in terms of Bayesian statistical inference. For our purposes, measurements are predictions from the individual decoders, and the system state is the 2-dimensional velocity vector of the prosthetic endpoint. Given the history of all measurements up to timestep k, $z_1,...,z_k$, we seek to find the most likely state of the system, x_k, which is equivalent to the mode of the *posterior* probability distribution:

$$p(x_k|z_k,...,z_1).$$

The Kalman filter is a well-known recursive Bayesian algorithm for solving this problem. This algorithm efficiently solves for the mode of the system posterior at time k given the set of all measurements of the system through time k. The Kalman filter first assumes a linear-Gaussian relationship between the current state of the system and the state at the previous timestep:

$$x_k = A_k x_{k-1} + w_k,$$

A_k is a coefficient matrix, and w_k is a Gaussian error term with mean 0 and covariance matrix W_k. The Kalman filter further assumes a linear-Gaussian relationship between the measurements and the state of the system at each timestep:

$$z_k = H_k x_k + q_k,$$

H_k is a coefficient matrix, and q_k is a Gaussian error term with mean 0 and covariance Q_k. Under these assumptions the Kalman filter provides an "optimal" estimate of the state posterior minimizing the mean-squared error.

To simplify the model, we assume A_k, W_k, H_k and Q_k are time-invariant, and so closed-form maximum joint probability solutions exist for each matrix [35]:

$$A = \left(\sum_{i=2}^{M} x_i x_{i-1}^T\right)\left(\sum_{i=2}^{M} x_{i-1} x_{i-1}^T\right)^{-1}$$

$$W = \frac{1}{M-1}\left(\sum_{i=2}^{M} x_i x_i^T - A\sum_{i=2}^{M} x_{i-1} x_i^T\right)$$

$$H = \left(\sum_{i=1}^{M} z_i x_i^T\right)\left(\sum_{i=1}^{M} x_i x_i^T\right)^{-1}$$

$$Q = \frac{1}{M}\left(\sum_{i=1}^{M} z_i z_i^T - H\sum_{i=1}^{M} x_i z_i^T\right).$$

See [36] for an excellent review of Kalman filter theory.

Artificial Neural Networks for Decision Fusion

Artificial neural networks have also been used as a method for fusing decisions from supervised classifiers and data from multiple sensors. An ANN is a mathematical model composed of simulated neuron *units* and links between units. Each unit has a corresponding activation function, ξ, that accepts a weighted sum of input values and outputs a net activation value. Activation functions may be piecewise constant, linear, or nonlinear. The general form of the net activation value for unit j is:

$$n_j = \xi_j\left(w_{0i} + \sum_{i=1}^{g} y_i w_{ij}\right),$$

where ξ_j is the activation function of the j^{th} unit, y_i is the net activation from unit i, and w_{ij} is the weight from unit i into unit j (see Figure 1).

We implemented feed-forward ANNs with either one or two hidden layers. At each timestep, the state estimates of each individual decoder are provided to the input units, while the output layer produces a fused estimate of the x and y velocities. The activation functions for all hidden units are tansigmoid, and the output layer uses linear functions. To train each ANN, we employed the scaled conjugate gradient method for learning the neuron weights and the mean squared error as a criterion function. We additionally optimized the number of hidden units by searching the space of all permutations ranging from one to 12 hidden units in the first layer, and zero to 11 hidden units in the second layer. Thus, 144 ANNs were examined to find an optimal selection of hidden units within each layer.

Simulated Neural Data

Similar to Moran and Schwartz [37] and Wu *et al.* [35] we model neuron spiking activity according to a cosine-tuning function relating the "preferred direction" of each neuron to the direction and velocity of an endpoint. Thus, the firing rate of a neuron at time t follows a Poisson distribution with mean z_t:

$$z_t = a_0 + \|v_t\| a_p \cos(\theta_t - \theta_p),$$

where θ_p is the preferred direction of the neuron, and θ_t and v_t are the angle and velocity of the movement, respectively. All experiments modeled 50 input neurons. Simulated neurons were randomly assigned preferred directions (within range $[-\pi, \pi]$), and parameters a_0 and a_p varied for each experiment.

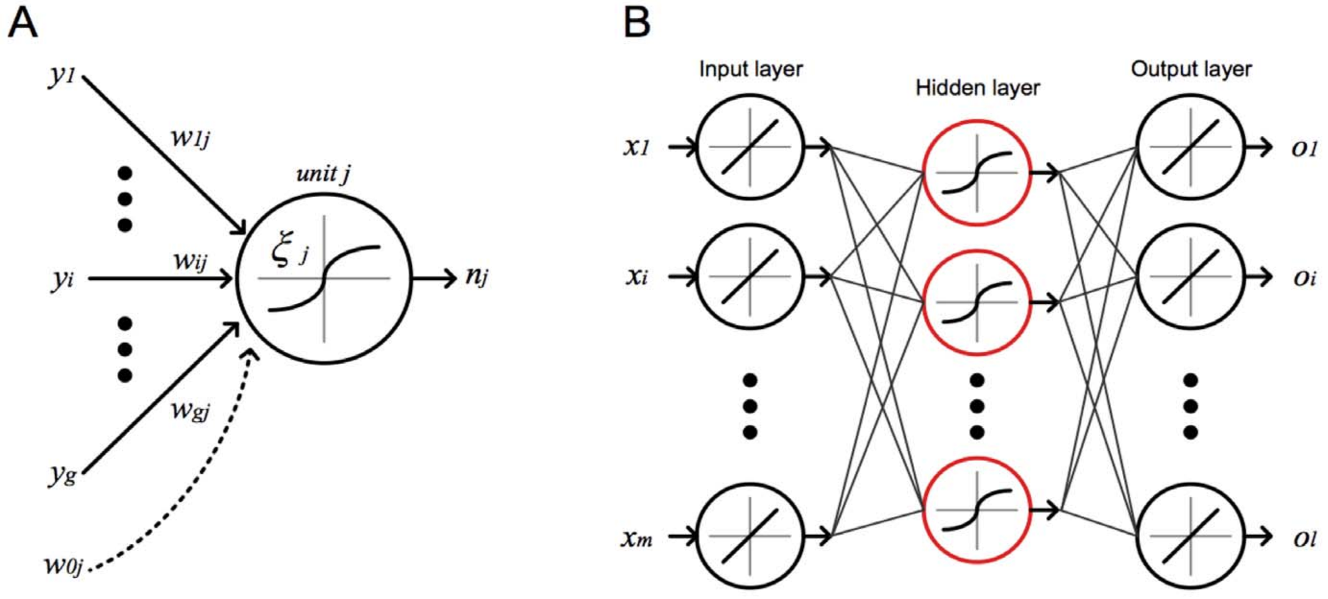

Figure 1. Conceptual design of an artificial neural network. (**A**) Each individual unit in the network accepts a weighted sum of input values, producing a single net activation value, n_j. (**B**) A three-layer network topology. This topology is *feed-forward* and *fully-connected*, that is, each unit links to all units in the layer directly after it.
doi:10.1371/journal.pone.0009493.g001

Individual Decoder Algorithms

Kalman filter. The Kalman filter framework as a single neural decoder was very similar to that of the fusion implementation. The individual Kalman filter modeled the relationship between neural spikes and the state of the device as a linear Gaussian process. The dimensionality of this observation model was larger than the observation model used for the fusion Kalman filter.

A variant of the population vector algorithm. We employed a model similar to the population vector algorithm (PVA) described in Moran and Schwartz [37] to decode the intended endpoint velocities. The equation used to generate our simulated neural data is described above, and the population vector algorithm utilizes the following model:

$$D(t-\tau)-b_0=\left\|\vec{V}(t)\right\|\left(b_n+b_y\sin[\theta(t)]+b_x\cos[\theta(t)]\right)$$

In PVA, τ, b_0, b_n, b_y, and b_x must be estimated before determining θ and $\| V(t)\|$ whereas in our model we only needed to estimate a_0, a_p, and θ_p. We can estimate these parameters using an iterative Taylor series approximation. As long as there are more neurons than the number of parameters (in this case 3), we can then estimate the angle and speed, or equivalently, the x and y components of the velocity.

Optimal linear decoder. The linear filters constructed for decoding used sliding windows of length four timepoints to form a response matrix of neuron firing rates. To train each filter, we performed a multiple regression of the x and y velocities over a response matrix spanning the entire training set:

$$f=\left(R^T R\right)^{-1} R^T v,$$

where f is the linear filter, R is the response matrix, and v is a vector containing the x or y velocities. For any response matrix, R, the linear prediction is:

$$u=R\bullet f$$

Note that for this filter, there exists a delay the same length as the window size, and we translated each decoded trajectory accordingly.

Decision Fusion Evaluation

Evaluation trials were designed to compare the accuracy of individual decoder predictions to "fused" results obtained from the Kalman filter and ANNs. Below we describe the three major components of each experiment: (i) individual decoder training, (ii) fusion decoder training, and (iii) final testing. See Figure 2 for a graphical description.

Individual decoder training. Each single decoder (PVA, Kalman filter, and optimal linear decoder) was trained on an identical dataset composed of 50 simulated neuron spike observations with a corresponding endpoint path. Trials associated with high-quality and poor-quality decoders used training datasets with 3,000 and 1,500 time-steps, respectively.

Fusion decoder training. When training the decision fusion algorithms, a set of predictions for each individual decoder is required. One could simply let the single decoders make predictions based on the initial training dataset, but this could lead to overfitting and poor performance on new data. To avoid this, a second dataset for *fusion training* was generated separately for the decision fusion algorithms. This dataset uses the same 50 simulated neurons, but for a different endpoint trajectory of 10,000 timesteps. Trained individual decoders were used to predict the two-dimensional endpoint velocity of the limb based on the fusion training dataset. At each timepoint, the predictions (v_x, v_y) were formed into an observation vector, (3 individual decoders ×2 velocity components = 6 components to each observation vector). The set of all observation vectors were used as a training set for the fusion Kalman filter and ANNs. To prevent overfitting the ANNs, a secondary *ANN fusion validation* dataset for a limited trajectory (3,000 timesteps) was employed in the same manner as the fusion training dataset.

Final testing. After training the fusion and individual decoders, a set of trajectories and corresponding spike signals

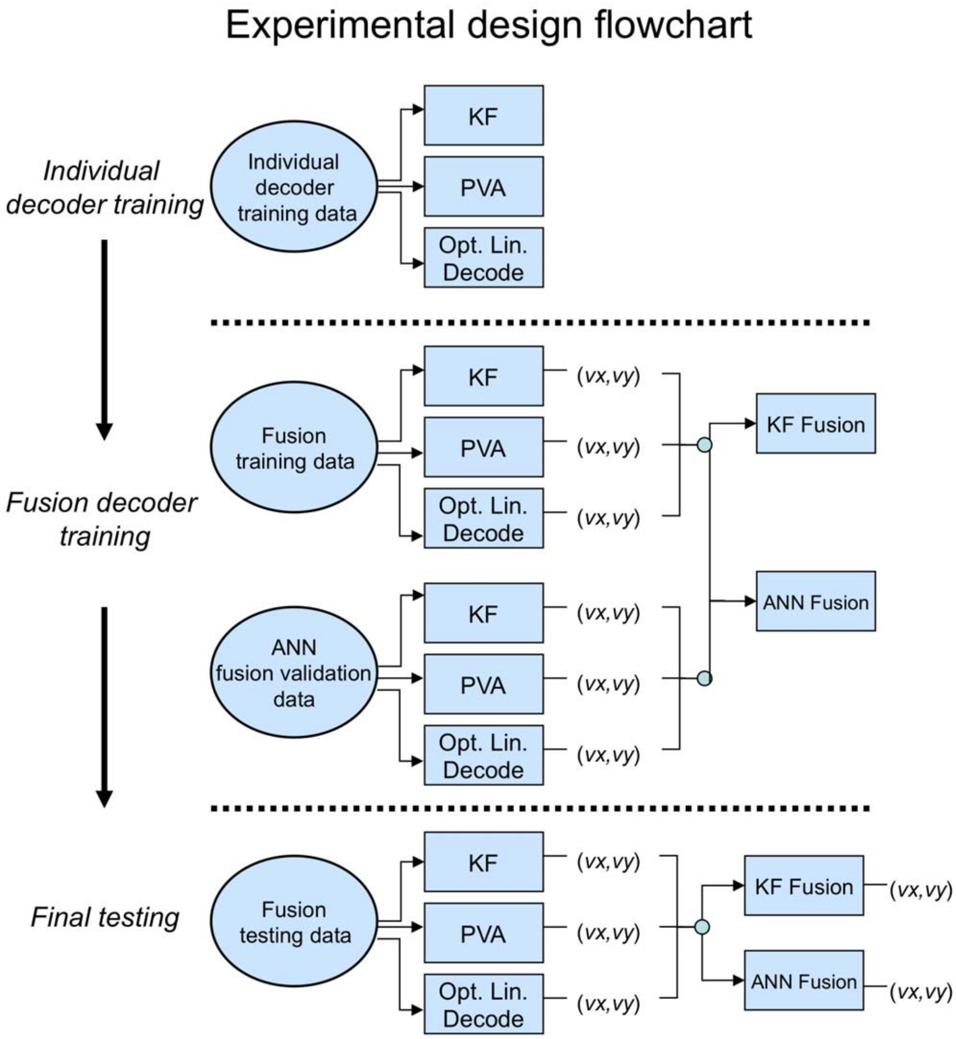

Figure 2. Experimental design for fusion trials. Flowchart describing fusion of Kalman filter (KF), PVA, and the optimal linear decodes using the Kalman filter and ANNs. Experimental trials contained three major phases: (i) individual decoder training, (ii) fusion decoder training, and (iii) final testing. In each experiment, individual decoders were first trained using the same simulated spike count data. Next, fusion decoders were trained on the individual decoders' outputs (predicted velocity components in *x* and *y* dimensions) for a separate **fusion training** dataset. An additional **validation** dataset was employed to prevent overtraining of ANNs. In final testing, trained individual decoders were used to predict the 2-d velocities, which were then compiled as input for fusion decoders. Endpoint velocity predictions from all decoders were then compared for accuracy. See Methods for details of the evaluation methodology.
doi:10.1371/journal.pone.0009493.g002

were generated for testing. Each trajectory represented 3,000 timesteps. For each trial, cortical spikes counts were input to individual decoders, which output predictions for *x* and *y* velocity estimates. Endpoint velocity predictions were then compiled into observation vectors and fed to the fusion algorithms for final predictions. Predictions from the individual decoders and the fusion methods were finally compared to the true endpoint velocities using root mean squared error.

Random Trajectory Generation

We generated random trajectories in 2-dimensional position space according to the following model:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \delta_1\alpha_1 \cos\left(\frac{t}{\alpha_2}\right) + \delta_2\alpha_3 \sin\left(\frac{t}{\alpha_4}\right) + \delta_3\alpha_5 \\ \delta_4\beta_1 \cos\left(\frac{t}{\beta_2}\right) + \delta_5\beta_3 \sin\left(\frac{t}{\beta_4}\right) + \delta_6\beta_5 \end{bmatrix}.$$

The parameters of the model for each trajectory were chosen by sampling from the following statistical distributions:

$$\delta_i = \begin{cases} 1 & \text{with probability } 0.8 \\ 0 & \text{with probability } 0.2 \end{cases}$$

$$(\alpha_1,\alpha_3,\beta_1,\beta_3) = \mathit{Uniform}\ (0.5,\, 2)$$

$$(\alpha_2,\alpha_4,\beta_2,\beta_4) = \mathit{Uniform}\ (100,\, 2000)$$

$$(\alpha_5,\beta_5) = \mathit{Uniform}\ (0,\, .1)$$

The space of possible trajectories spanned both nonlinear and linear relationships.

Results

We present the fusion problem in the context of estimating the endpoint velocity of a prosthetic arm using several different

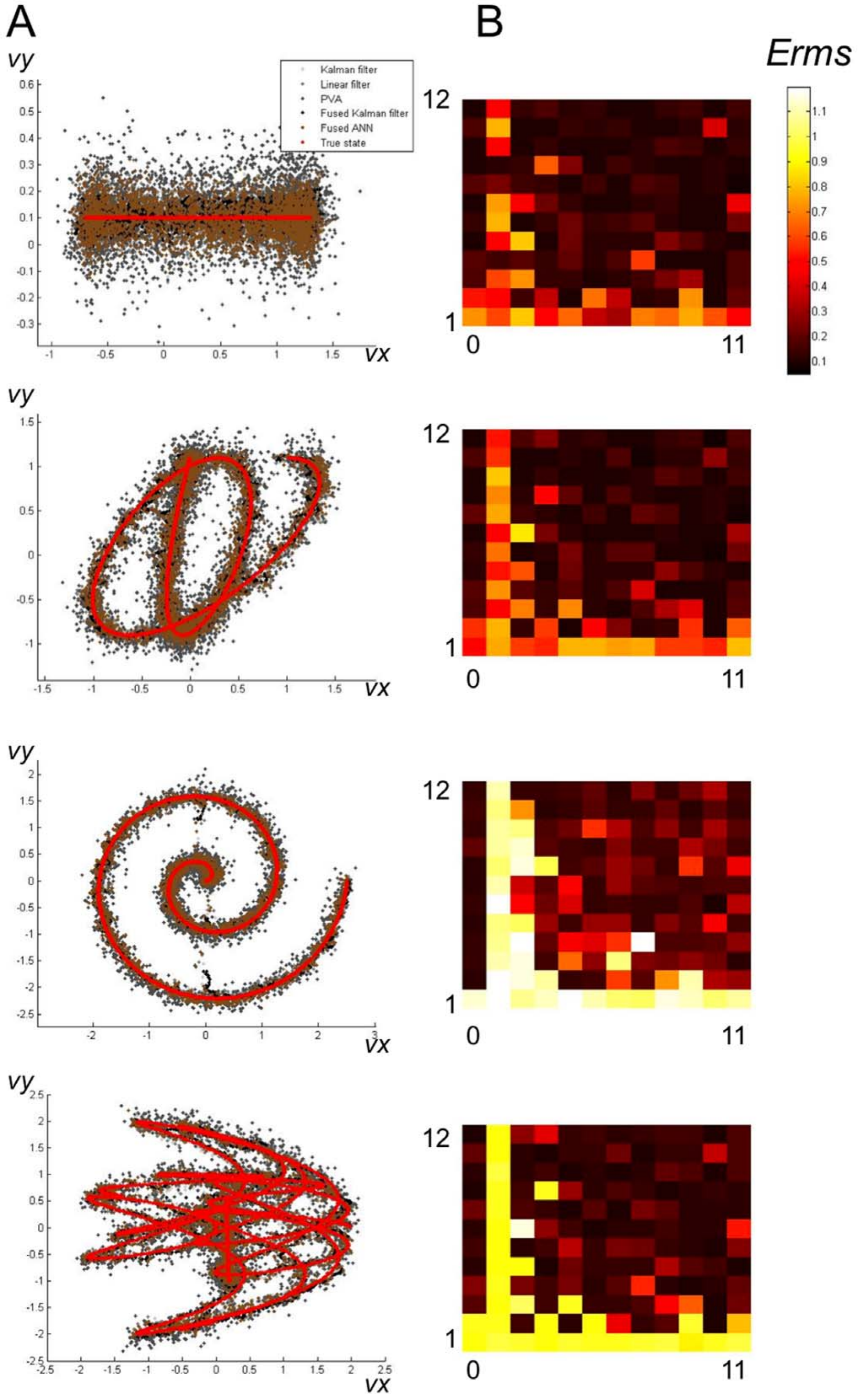

Figure 3. Initial testing of fusion decoders. (**A**) Decoded velocity trajectories for four trials. The true velocities are shown in red. The fused ANN and fused Kalman filter decodes are shown in brown and black, respectively. Individual decoders are plotted in varying shades of grey. (**B**) E_{rms} of 144 neural networks for four trial decodes. We examined a range of single and double hidden-layer networks to optimize the fusion results. Rows correspond to 1st-layer sizes, while columns are 2^{nd}-layer sizes. Note the first column in each matrix corresponds to all single hidden-layer networks. Interestingly, many single hidden-layer networks outperform more complex networks, indicating the dynamic accuracies of different neural network topologies. Table 2 displays the corresponding E_{rms} values for each decoder.
doi:10.1371/journal.pone.0009493.g003

Table 1. Artificial neural networks with lowest E_{rms} for each trial.

Trial	1	2	3	4
No. units in 1st hidden layer	8	11	12	6
No. units in 2nd hidden layer	11	7	10	10
E_{rms}± s.e.	0.085±0.002	0.083±0.002	0.097±0.003	0.103±0.004

Searching the space of possible topologies seen in Figure 3B, the most accurate decoding ANNs had different topologies for each trial.
doi:10.1371/journal.pone.0009493.t001

decoding algorithms of varying accuracy. Decoding studies often focus on endpoint trajectories, leaving the controls of the limb to determine optimal joint positions and velocities by inverse kinematics.

Simulated Fusion Trials

To investigate these fusion methods, we simulated neural spike data and implemented the following algorithms for spike decoding: standard Kalman filter [35,38,39], optimal linear filter [40,41], and a variant of the population vector algorithm (PVA) [23,42,43,44]. The optimal linear filter uses a sliding window to look back in time to estimate the current state of the arm using a multidimensional linear regression. A separate linear filter is developed for each variable of interest (in our case, x and y velocities). The population vector algorithm predicts velocity and direction using the "preferred direction" of each neuron in conjunction with a model relating neural activity to speed and direction of movement. We simulate single-neuron spike firing rates as a function of the velocity and direction of the limb in x and y coordinates. All simulated neuron firing rates were perfectly cosine-tuned and included Poisson noise (see Methods for detailed descriptions of decoders and simulated firing rates).

Initial Testing of Fusion Algorithms

Testing the fusion algorithms first required training each individual decoder. Each trained algorithm was then used to decode a fusion training dataset and a separate fusion validation dataset for training the artificial neural network. The use of a validation dataset prevents overtraining of the ANN. The outputs of the trained algorithms (in our case x and y velocities) served as inputs to train the fusion algorithms (Figure 2). All trained algorithms decoded velocities for four testing datasets. The four test sets were generated independently from previous training and validation data, and tested a range of trajectories from simple to complex.

We measure the accuracy of the decoded trajectories in terms of the root mean squared error (E_{rms}) in velocity space. If $\left(v_{xk}, v_{yk}\right)$ is the true velocity and $\left(\hat{v}_{xk}, \hat{v}_{yk}\right)$ is the estimate $(k=1,\ldots,N)$, then:

$$E_{rms}=\left[\frac{1}{N}\sum_{k=1}^{N}\left(\left(v_{xk}-\hat{v}_{xk}\right)^2+\left(v_{yk}-\hat{v}_{yk}\right)^2\right)\right]^{1/2}.$$

Figure 3B displays ANN E_{rms} results of optimizing the number of neurons in each hidden layer for each of four trials. Note that the first column of cells in each matrix corresponds to a single hidden-layer network. We observe that neural networks with a single unit in the first or second hidden layer perform poorly. We also see that the single hidden-layer networks typically perform just as well as many of the double hidden-layer networks. This experiment reveals the dynamic nature of network accuracy depending on the topology employed. Indeed more complex networks do not necessarily provide the best performance. A notable example is the double hidden-layer network with nine and three units in the first and second hidden layers, respectively. The E_{rms} for this network is relatively high (compared to its immediate neighbors) for trials 1 through 3, but this disappears for trial 4. ANN topologies with the lowest E_{rms} were all different for each trial (Table 1). This suggests that optimizing the number of neurons is data dependent and no one topology will always result in the best performance.

The final decoded trajectories are presented in Figure 3A. For each trial, the best performing ANN is plotted in brown. True velocities are plotted in red. **Table 2** shows the E_{rms} for all individual decoders and fusion algorithms. In three out of four trials, the Kalman filter fusion resulted in the most accurate decode. In the remaining trial, the fused ANN decoded velocities had the lowest E_{rms}. In all four trials, at least one fusion algorithm outperformed all three individual decoders. Furthermore, across individual decoders, no single method was consistently superior.

Variable Decoding Accuracies

The accuracy of neural decoders depends not only on the sophistication of the decoding algorithms but also on the physical recording locations and the nature of the signals. A few millimeters of discrepancy in electrode placement can dramatically impact decoding accuracy [20]. Thus, in devices with multimodal recording, no one decoded modality is likely to provide superior performance over others for the full spectrum of functionality.

To address this scenario, we subsequently tested the ability of our fusion algorithms to handle poor quality decoding. Generating a simulated neural training set lacking sufficient complexity and size, we retrained the individual decoders resulting in unacceptable decoding accuracy. We ran four decoding trials, comparing the fusion outputs to the single decoders. In Figure 4 and **Table 3**, we observe the poor performance of the Kalman filter and optimal linear filter decoders. Despite the high error associated with each single decoder,

Table 2. E_{rms} ± standard error (s.e.) for four trials.

Trial	1	2	3	4
Kalman filter	0.073±0.001	0.069±0.001	0.126±0.004	0.090±0.004
Linear filter	0.093±0.002	0.090±0.002	0.102±0.003	0.107±0.005
Population vector	0.174±0.003	0.172±0.003	0.179±0.003	0.203±0.011
Kalman fusion	**0.059±0.001**	**0.062±0.001**	0.119±0.004	**0.066±0.002**
ANN fusion	0.0850±0.002	0.083±0.002	**0.097±0.003**	0.103±0.004

Bold elements in tables have the lowest E_{rms} for the trial. In all four trials, the fusion algorithms had more accurate results than at least two of the three individual decoders. In trials 1, 2, and 4, the Kalman fusion method produced the lowest E_{rms}. In trial 3, the fused ANN decisions were the most accurate.
doi:10.1371/journal.pone.0009493.t002

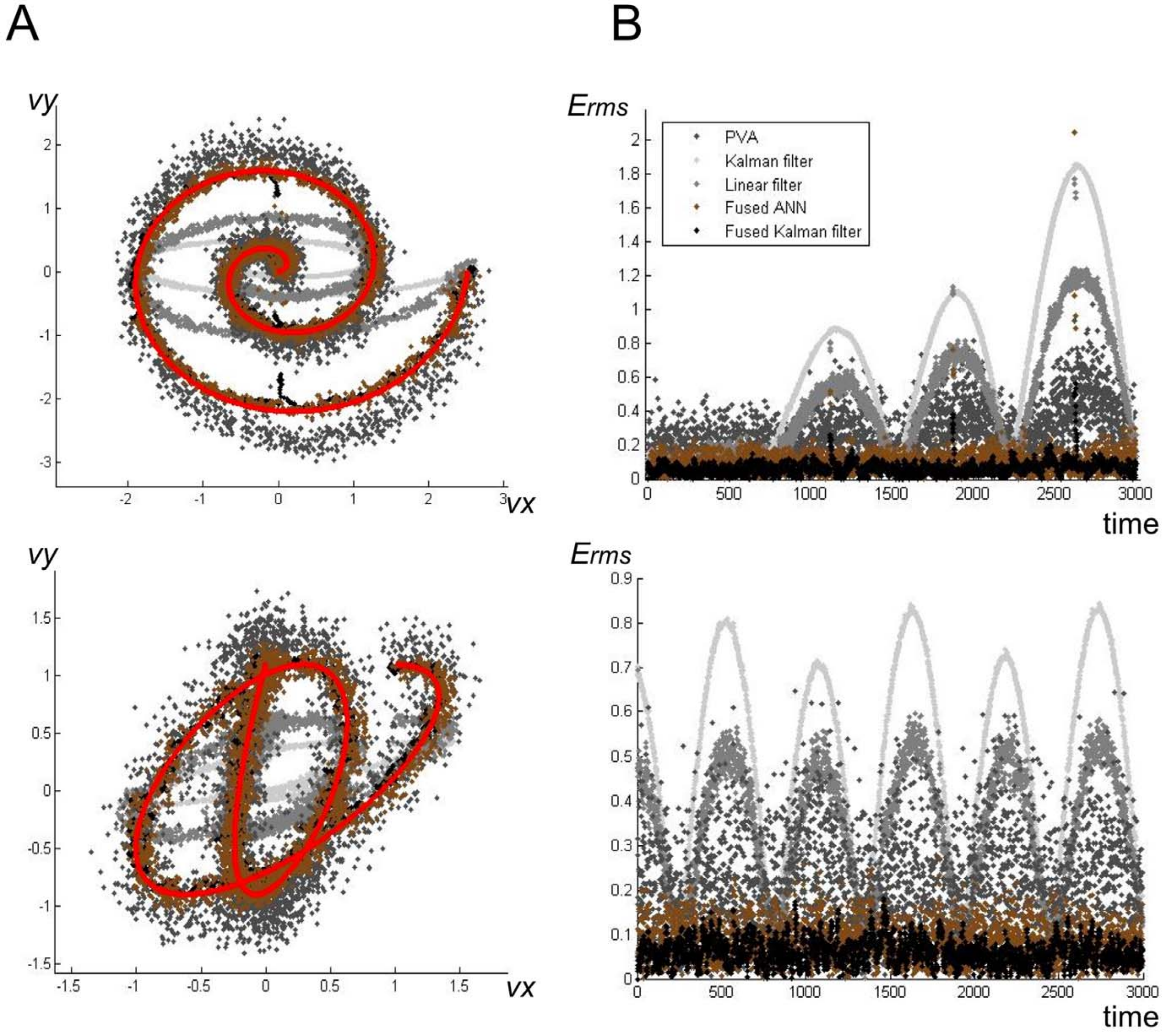

Figure 4. Fusion results of using potentially poor quality decoders. These two sets correspond to trials 2 and 3 in Table 3. (A) Example trials showing individual and fusion decodes. True velocities are shown in red. The fused ANN and fused Kalman filter decodes are shown in brown and black, respectively. Individual decoders are plotted in varying shades of grey. (B) Corresponding pointwise root mean squared error of decodes over time. Note that time is unitless in these simulations. Though the decoders have variable accuracy over time, the fusion algorithms maintain acceptable decoding accuracy throughout the entire trials.
doi:10.1371/journal.pone.0009493.g004

the fusion algorithms successfully produce highly accurate decodes, significantly improving over all three individual decoders. Note that we again optimized the ANN topologies for each trial similarly to the previous experiment. In Figure 4B, we analyzed the decoding accuracy of each algorithm over time for trials 2 and 3. While the error for the individual decoders varies over time, the fusion algorithms effectively assessed the individual decoders' weaknesses, and resulted in lower E_{rms} throughout the entire trials.

To determine if the improvement of the fusion algorithms was statistically significant, we generated 468 additional randomized trajectories (selected from a large space of smooth realistic movements, see Methods) and corresponding simulated neural spike datasets. For each trial, we employed only a single ANN topology, because searching a space of topologies is not feasible for real-time decoding. The selected ANN used a single hidden-layer with six hidden units, the same as the number of input nodes. The fusion Kalman filter resulted in significantly lower E_{rms} than all three individual decoders, ($\boldsymbol{p<1e\text{-}150}$ in all cases, one-tailed paired T-test) (see Figure 5). The ANN fusion method was not as successful, though still produced significantly more accurate decodes than the Kalman filter and linear filter single decoders, ($\boldsymbol{p<1e\text{-}44}$ for both comparison, one-tailed paired T-test). Our PVA variant resulted in significantly more accurate decodes than ANN fusion ($\boldsymbol{p<1e\text{-}42}$, one-tailed paired T-test). Since it is not reasonable to find an optimal ANN topology in real time, the Kalman filter has a major advantage over the ANN as a fusion method. However, if a topology could be found in training that performed well overall, then the ANN would provide a computationally efficient method for decision fusion.

Table 3. E_{rms} ± standard error (s.e.) for four trials with variable decoding accuracies.

Trial	1	2	3	4
Kalman filter	0.557±0.009	0.828±0.017	0.549±0.009	0.898±0.017
Linear filter	0.371±0.006	0.550±0.012	0.365±0.006	0.523±0.010
Population vector	0.248±0.005	0.318±0.013	0.238±0.004	0.285±0.005
Kalman fusion	0.108±0.002	**0.079±0.012**	**0.063±0.001**	0.198±0.003
ANN fusion	**0.098±0.002**	0.116±0.005	0.091±0.002	**0.121±0.002**

Bold elements in tables have the lowest E_{rms} for the trial. In all four trials, the fusion algorithms had more accurate results than **all** three individual decoders. Trials 2 and 3 are displayed in Figure 4.
doi:10.1371/journal.pone.0009493.t003

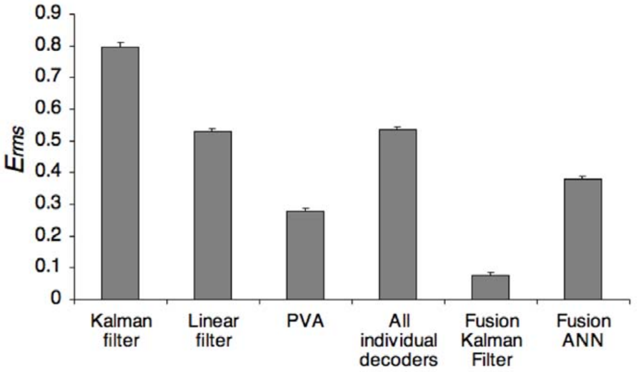

Figure 5. Results of decoders on 468 random trajectories (E_{rms} mean ± s.e.). The improvement of fusion algorithms over the combined individual decoders was statistically significant (**p<1e-29** in both cases, two-tailed Welch's T-test). While the fusion Kalman filter produced the significantly more accurate outputs than the individual decoders, the ANN limited to a single topology did not perform as well, illustrating an advantage of the Kalman filter as a fusion method.
doi:10.1371/journal.pone.0009493.g005

Discussion

We have described a framework for fusing decisions in the context of multimodal prosthetic devices. Investigating the Kalman filter and ANNs, we have shown that each fusion method is capable of producing accurate fusion decodes and can adapt to decodes of varying quality over time.

While our expertise is targeted towards neural decoding for prosthetic limb movement, this approach may be generalized to the larger field of brain-machine interfaces (BMIs) to help improve communication for patients suffering from severe paralysis, locked-in syndrome, and other neurological injuries. Recent BMI studies have demonstrated success in providing some level of communication for subjects [41,45], though to our knowledge, none have employed a fusion framework for decoding. As hardware platforms for neural recording continue to advance, so too will our opportunities for fusing multiple signals with distinct characteristics.

The computational expense of a fusion step in a neural prosthetic device is of notable importance. Each of the methods examined in this study is capable of running in real-time on a single processor, which is likely to be the hardware implementation of such a framework. Furthermore, the computational cost of individual modality decoders is increasing considerably, with many suggesting parallel processing implementations [46,47]. The efficiency of these fusion algorithms could be improved by reducing the dimensionality of the data using feature selection or principal component analysis [22].

Progress in neural recording technologies may eventually lead to opportunities for data fusion, where a single decoder is used on all modalities simultaneously. Our choice to employ decision fusion in this study was in large part due to the current capabilities of neural prostheses and those in development, making our findings timely.

Our results must be qualified because of the artificial nature of our cortical spike data. Though our analysis is based on simulated neural activity, we sought to capture the fundamental features of spike data including: a realistic number of monitored neurons, randomized preferred directions, and firing rates exhibiting Poisson noise. Our simulated neurons are indeed close to ideal, but we have shown the significant improvement decision fusion can provide when fusing predictions from decoders of variable accuracy – a result independent of the simulated data itself. Currently, no continuous real-time multimodal neural data recordings are available, but several are in production, and the community has shown an evident interest in this direction [48,49]. We plan to perform a rigorous off-line evaluation of decision fusion and data fusion methodologies using real multimodal neural data in future work.

An ideal neural prosthesis will be fully autonomous, capable of independently retraining and adapting to different human conditions and mechanical failure. Electrode loss is arguably the most important limiting factor for neural prostheses proliferation [17,18,19], and multiple craniotomies are not a practical solution. As a corollary, an autonomous prosthetic arm will need to detect recording anomalies and adjust appropriately. If individual decoders do not address this issue, any fusion technique is susceptible to electrode loss. However, some fusion methods are easily modified to adapt to this problem. The Kalman filter and other methods may be formulated such that poor quality decoders can be isolated and removed from the prediction without retraining, while the ANNs would be significantly more problematic. We hope to extend these methods to provide better autonomy in the future.

Neural prosthetics is a swiftly evolving field with ambitious goals. Restoring the functionality of a limb for an individual will require innovative technology and robust computational methods to rapidly and accurately assess user intent.

Acknowledgments

We are grateful to Matthew Para, Francesco Tenore, Vikram Aggarwal (Johns Hopkins U.), and Cevat Ustun (Caltech) for helpful conversations regarding this project. We also thank Dan Mendat (Rutgers), David Huberdeau (Johns Hopkins U.), John Kegelman (Johns Hopkins U.) and Justin Bartley for their technical contributions in the Revolutionizing Prosthetics 2009 team at the Johns Hopkins University Applied Physics Lab.

Author Contributions

Conceived and designed the experiments: JRW TL. Performed the experiments: JRW TL. Analyzed the data: JRW. Contributed reagents/materials/analysis tools: JRW. Wrote the paper: JRW TL WEB JDB.

References

1. Dillingham TR, Pezzin LE, MacKenzie EJ (2002) Limb amputation and limb deficiency: epidemiology and recent trends in the United States. South Med J 95: 875–883.
2. Ziegler-Graham K, MacKenzie EJ, Ephraim PL, Travison TG, Brookmeyer R (2008) Estimating the prevalence of limb loss in the United States: 2005 to 2050. Arch Phys Med Rehabil 89: 422–429.

3. Nawrot MP, Boucsein C, Rodriguez Molina V, Riehle A, Aertsen A, et al. (2008) Measurement of variability dynamics in cortical spike trains. J Neurosci Methods 169: 374–390.
4. Banerjee A, Series P, Pouget A (2008) Dynamical constraints on using precise spike timing to compute in recurrent cortical networks. Neural Comput 20: 974–993.
5. Scherberger H, Jarvis MR, Andersen RA (2005) Cortical local field potential encodes movement intentions in the posterior parietal cortex. Neuron 46: 347–354.
6. Ray S, Hsiao SS, Crone NE, Franaszczuk PJ, Niebur E (2008) Effect of stimulus intensity on the spike-local field potential relationship in the secondary somatosensory cortex. J Neurosci 28: 7334–7343.
7. Rossi L, Foffani G, Marceglia S, Bracchi F, Barbieri S, et al. (2007) An electronic device for artefact suppression in human local field potential recordings during deep brain stimulation. J Neural Eng 4: 96–106.
8. Kim J, Wilson JA, Williams JC (2007) A cortical recording platform utilizing microECoG electrode arrays. Conf Proc IEEE Eng Med Biol Soc 2007: 5353–5357.
9. Pistohl T, Ball T, Schulze-Bonhage A, Aertsen A, Mehring C (2008) Prediction of arm movement trajectories from ECoG-recordings in humans. J Neurosci Methods 167: 105–114.
10. Sanchez JC, Gunduz A, Carney PR, Principe JC (2008) Extraction and localization of mesoscopic motor control signals for human ECoG neuroprosthetics. J Neurosci Methods 167: 63–81.
11. Schalk G, Kubanek J, Miller KJ, Anderson NR, Leuthardt EC, et al. (2007) Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. J Neural Eng 4: 264–275.
12. Shenoy P, Miller KJ, Ojemann JG, Rao RP (2008) Generalized features for electrocorticographic BCIs. IEEE Trans Biomed Eng 55: 273–280.
13. Nagata K, Ando K, Nakano S, Nakajima H, Yamada M, et al. (2006) Development of the human interface equipment based on surface EMG employing channel selection method. Conf Proc IEEE Eng Med Biol Soc 1: 6193–6196.
14. Wu W, Gao X, Hong B, Gao S (2008) Classifying single-trial EEG during motor imagery by iterative spatio-spectral patterns learning (ISSPL). IEEE Trans Biomed Eng 55: 1733–1743.
15. Morash V, Bai O, Furlani S, Lin P, Hallett M (2008) Classifying EEG signals preceding right hand, left hand, tongue, and right foot movements and motor imageries. Clin Neurophysiol 119: 2570–2578.
16. Gladwin TE, t Hart BM, de Jong R (2008) Dissociations between motor-related EEG measures in a cued movement sequence task. Cortex 44: 521–536.
17. Biran R, Martin DC, Tresco PA (2005) Neuronal cell loss accompanies the brain tissue response to chronically implanted silicon microelectrode arrays. Exp Neurol 195: 115–126.
18. Griffith RW, Humphrey DR (2006) Long-term gliosis around chronically implanted platinum electrodes in the Rhesus macaque motor cortex. Neurosci Lett 406: 81–86.
19. Polikov VS, Tresco PA, Reichert WM (2005) Response of brain tissue to chronically implanted neural electrodes. J Neurosci Methods 148: 1–18.
20. Andersen RA, Burdick JW, Musallam S, Scherberger H, Pesaran B, et al. (2004) Recording advances for neural prosthetics. Conf Proc IEEE Eng Med Biol Soc 7: 5352–5355.
21. Andersen RA, Musallam S, Pesaran B (2004) Selecting the signals for a brain-machine interface. Curr Opin Neurobiol 14: 720–726.
22. Aggarwal V, Acharya S, Tenore F, Shin HC, Etienne-Cummings R, et al. (2008) Asynchronous decoding of dexterous finger movements using M1 neurons. IEEE Trans Neural Syst Rehabil Eng 16: 3–14.
23. Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB (2008) Cortical control of a prosthetic arm for self-feeding. Nature 453: 1098–1101.
24. Ben-Yacoub S, Abdeljaoued Y, Mayoraz E (1999) Fusion of face and speech data for person identity verification. IEEE Trans Neural Netw 10: 1065–1074.
25. Monwar MM, Gavrilova ML (2009) Multimodal Biometric System Using Rank-Level Fusion Approach. IEEE Trans Syst Man Cybern B Cybern 39: 867–878.
26. Ribaric S, Fratric I (2005) A biometric identification system based on eigenpalm and eigenfinger features. IEEE Trans Pattern Anal Mach Intell 27: 1698–1709.
27. Zhou DY, Zhou F (2008) Data fusion control and guidance of surface-to-air missile under the complex circumstance based on neural-net technology. Journal of Systems Engineering and Electronics 19: 996–1002.
28. Kyriakoulis N, Gasteratos A, Amanatiadis A (2006) Comparison of data fusion techniques for robot navigation. Advances in Artificial Intelligence, Proceedings 3955: 547–550.
29. Lima PU (2007) A Bayesian approach to sensor fusion in autonomous sensor and robot networks. Ieee Instrumentation & Measurement Magazine 10: 22–27.
30. Lin PC, Komsuoglu H, Koditschek DE (2006) Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits. IEEE Transactions on Robotics 22: 932–943.
31. Zhang H, Chen HP, Xi N (2006) Automated robot programming based on sensor fusion. Industrial Robot-an International Journal 33: 451–459.
32. Rohlfing T, Russakoff DB, Brandt R, Menzel R, Maurer CRJ (2004) Performance-based multi-classifier decision fusion for atlas-based segmentation of biomedical images. IEEE International Symposium on Biomedical Imaging: Nano to Macro. pp 404–407.
33. Ahiskali M, Green D, Kounios J, Clark CM, Polikar R (2009) ERP based decision fusion for AD diagnosis across cohorts. Conf Proc IEEE Eng Med Biol Soc 1: 2494–2497.
34. Prasad S, Bruce LM, Ball JE (2008) A multi-classifier and decision fusion framework for robust classification of mammographic masses. Conf Proc IEEE Eng Med Biol Soc 2008: 3048–3051.
35. Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ (2006) Bayesian population decoding of motor cortical activity using a Kalman filter. Neural Comput 18: 80–118.
36. Welch G, Bishop G (2006) An Introduction to the Kalman Filter. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf. Accessed 2008 August 14.
37. Moran DW, Schwartz AB (1999) Motor cortical representation of speed and direction during reaching. J Neurophysiol 82: 2676–2692.
38. Wu W, Black MJ, Mumford D, Gao Y, Bienenstock E, et al. (2004) Modeling and decoding motor cortical activity using a switching Kalman filter. IEEE Trans Biomed Eng 51: 933–942.
39. Wu W, Shaikhouni A, Donoghue JP, Black MJ (2004) Closed-loop neural control of cursor motion using a Kalman filter. Conf Proc IEEE Eng Med Biol Soc 6: 4126–4129.
40. Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, et al. (2003) Learning to control a brain-machine interface for reaching and grasping by primates. PLoS Biol 1: E42.
41. Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, et al. (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442: 164–171.
42. Georgopoulos AP, Kettner RE, Schwartz AB (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. J Neurosci 8: 2928–2937.
43. Kettner RE, Schwartz AB, Georgopoulos AP (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. III. Positional gradients and population coding of movement direction from various movement origins. J Neurosci 8: 2938–2947.
44. Schwartz AB, Kettner RE, Georgopoulos AP (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. J Neurosci 8: 2913–2927.
45. Guenther FH, Brumberg JS, Wright EJ, Nieto-Castanon A, Tourville JA, et al. (2009) A wireless brain-machine interface for real-time speech synthesis. PLoS One 4: e8218.
46. Srinivasan L, Eden UT, Mitter SK, Brown EN (2007) General-purpose filter design for neural prosthetic devices. J Neurophysiol 98: 2456–2475.
47. Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, et al. (2007) Mixture of trajectory models for neural decoding of goal-directed movements. J Neurophysiol 97: 3763–3780.
48. Wilson JA, Felton EA, Garell PC, Schalk G, Williams JC (2006) ECoG factors underlying multimodal control of a brain-computer interface. IEEE Trans Neural Syst Rehabil Eng 14: 246–250.
49. Cunningham JP, Yu BM, Gilja V, Ryu SI, Shenoy KV (2008) Toward optimal target placement for neural prosthetic devices. J Neurophysiol 100: 3445–3457.