RESEARCH ARTICLE

Tuning Curves for Arm Posture Control in Motor Cortex Are Consistent with Random Connectivity

Hagai Lalazar[1] *, L. F. Abbott[1,2], Eilon Vaadia[3]

1 Center for Theoretical Neuroscience, Columbia University, New York, New York, United States of America, 2 Department of Physiology and Cellular Biophysics, Columbia University, New York, New York, United States of America, 3 Edmond and Lily Safra Center for Brain Sciences, Hebrew University, Jerusalem, Israel

* h.lalazar@columbia.edu

Abstract

Neuronal responses characterized by regular tuning curves are typically assumed to arise from structured synaptic connectivity. However, many responses exhibit both regular and irregular components. To address the relationship between tuning curve properties and underlying circuitry, we analyzed neuronal activity recorded from primary motor cortex (M1) of monkeys performing a 3D arm posture control task and compared the results with a neural network model. Posture control is well suited for examining M1 neuronal tuning because it avoids the dynamic complexity of time-varying movements. As a function of hand position, the neuronal responses have a linear component, as has previously been described, as well as heterogeneous and highly irregular nonlinearities. These nonlinear components involve high spatial frequencies and therefore do not support explicit encoding of movement parameters. Yet both the linear and nonlinear components contribute to the decoding of EMG of major muscles used in the task. Remarkably, despite the presence of a strong linear component, a feedforward neural network model with entirely random connectivity can replicate the data, including both the mean and distributions of the linear and nonlinear components as well as several other features of the neuronal responses. This result shows that smoothness provided by the regularity in the inputs to M1 can impose apparent structure on neural responses, in this case a strong linear (also known as cosine) tuning component, even in the absence of ordered synaptic connectivity.

Author Summary

Relationships between the activity of single neurons and experimental parameters are often characterized by functions called tuning curves. Regular tuning-curve shapes are typically assumed to arise from structure in the synaptic inputs to each neuron. We found that the activities of neurons in primary motor cortex during an arm posture task exhibit both a regular component that fits a well-known tuning curve description, and heterogeneous irregular components that do not. Such complex components are often assumed to

reflect residual noise. However, both the regular and irregular components are needed to optimally decode the commands that guide the muscles used in the task. We then asked what type of input structure was needed to generate neuronal responses with both regular and irregular elements. We constructed and analyzed a mathematical model, based on known physiology of the relevant brain regions that replicates the full spectrum of recorded neuronal responses. Surprisingly, the synaptic connectivity in this model is completely random.

Introduction

The dependence of neuronal responses on stimulus- or movement-related parameters is often characterized by tuning curves. A natural assumption is that a smooth, regular tuning curve reflects structured, orderly input to a neuron. However, neuronal responses inevitably involve a degree of irregularity, even when responses are averaged across trials. Do such irregularities simply reflect noise in the inputs, or might they suggest something more complex such as unstructured input? Here we address this question using data recorded from primary motor cortex (M1) during an arm posture task, augmented by a neural network model of M1 neurons and their inputs.

Interpreting neural activity during arm movements is difficult because motor and sensory activity as well as limb biomechanical variables all change simultaneously. In this study, we focus on the often ignored task of actively maintaining arm posture. Arm posture control is a natural behavior without the dynamic complexity of time-varying movements. It is therefore well suited to address the nature of neuronal tuning curves and their relationship to input. To reveal fine-scale tuning curve structure, we employed a task with 54 different arm postures, consisting of 27 target positions with two forearm rotation angles.

Previous studies of arm posture control [1–3] concluded that neuronal responses in M1 vary as linear functions of hand position, which, by a change of coordinates, is equivalent to cosine tuning. The finding of broad, singly peaked tuning curves in both motor and visual areas led to proposals suggesting that both are generated by similarly structured cortical micro-circuit [4, 5]. The M1 tuning curves we extract have a strong linear component, in agreement with previous studies, but we also find sizable and significant nonlinear elements. We examine the quality of the fit of linear tuning curve models across the entire population of tuned neurons, we analyze the nonlinear components in detail, and we determine how both linear and nonlinear elements contribute to the accuracy of inferring EMGs from M1 neuronal activity. Finally, we construct a neural network model of M1 activity driven by target-position related inputs that accurately replicates the data. Surprisingly, the synaptic connectivity in this model is completely unstructured, in fact, random. This model shows that the presence of a strong regular and, in this case, linear component in the tuning of a population of neurons does not necessarily imply structured connectivity.

Results

During a continuous target-to-target reach and hold task in a virtual reality setup, monkeys maintained static arm posture at one of 27 targets in 3D space (Methods). They were trained to perform this task with their forearm at a pronated or supinated angle (Fig 1). To study posture control, we analyzed single-unit activity during a 200 ms period near the end of the target hold epoch when firing rates are approximately constant (S1B Fig). The responses of 81% (411/510) of the recorded neurons are tuned across the 54 arm postures (27 target positions times 2
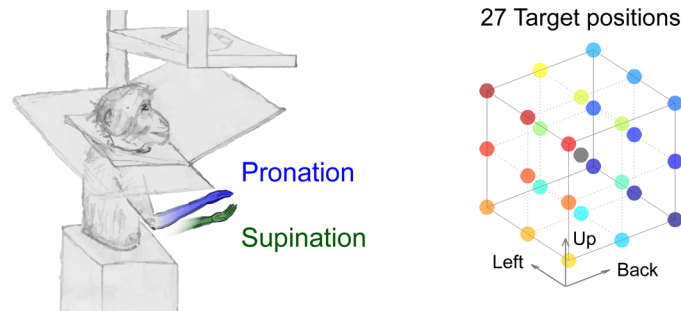
Fig 1. Monkeys controlled a cursor in a standard virtual-reality setup, where they held one of 27 targets in 3D space, with their forearm pronated or supinated.

forearm rotation angles; ANOVA, $p < 0.01$; statistical significance was also verified using the non-parametric Kruskal-Wallis test with identical results). Various tests established that arm-posture tuning did not arise from hand jitter (S1 Text).

Tuning of the neuronal responses

The responses of each neuron, for a particular forearm angle, are summarized by a set of 27 mean firing rates, one for each hand position (Fig 2). To study the relationship between firing rate and hand position, we fit spatial linear tuning curves [2] to the responses of all the tuned neurons (N = 411; Methods), separately for pronation and supination. Neurons with perfectly spatial linear tuning would fire at rates that increase linearly when the hand is held at different positions along a *Preferred Position* (PP) vector and are constant across locations in planes orthogonal to this vector. During pronation, the response of the neuron shown in Fig 2A is well fit by a linear tuning curve (Fig 2C). However, this is one of only 9 neurons with $R^2 > 0.9$ for the fit to linear tuning. Across the population of tuned neurons, the distribution of R^2 values for the spatial linear tuning fit is broad (Fig 3B). We pooled the R^2 distributions from both forearm angles because they are not significantly different (Wilcoxon rank-sum test, $p = 0.56$) and, in fact, there is only a small reduction in goodness of fit if the PP vectors for each neuron are required to be the same for both forearm angles (Methods, Eq 3). The R^2 distribution has a median of 0.52, and its breadth indicates that neurons range from being well fit to poorly fit by a linear tuning curve.

We tested whether the deviations of the data from the linear fit are due to noise by generating artificial data with perfectly spatial linear tuning, using the model parameters obtained by fitting the real data, and including noise extracted from the real data (resampled 1,000 times; Methods). We calculated mean responses for these artificial data and re-fit them to the linear tuning curves. The resulting distribution of R^2 values is skewed towards 1 (Fig 3C), with a median of 0.75, indicating significantly better fits than those for the real data (Wilcoxon rank-sum test, $p < 10^{-46}$). Generating Poisson spike trains with rates given by the generated linear tuning curves over the same number of trials as the real neurons yielded similar results. We also studied whether the observed neuronal responses had linear structure above chance level by generating another artificial dataset constructed from randomly generated responses drawn across the same range of firing rates as the real neurons. Noise traces extracted from the data were added to these responses. When fit to the linear tuning curve, this resulted in significantly worse fits than for the real data, with an R^2 distribution (Fig 3A) skewed towards zero with a median of 0.12. These analyses indicate the linear model is not a complete description of the tuning curves extracted from the data, but that the linear component is significantly larger than what would be expected from a random mapping between hand position and firing rate.
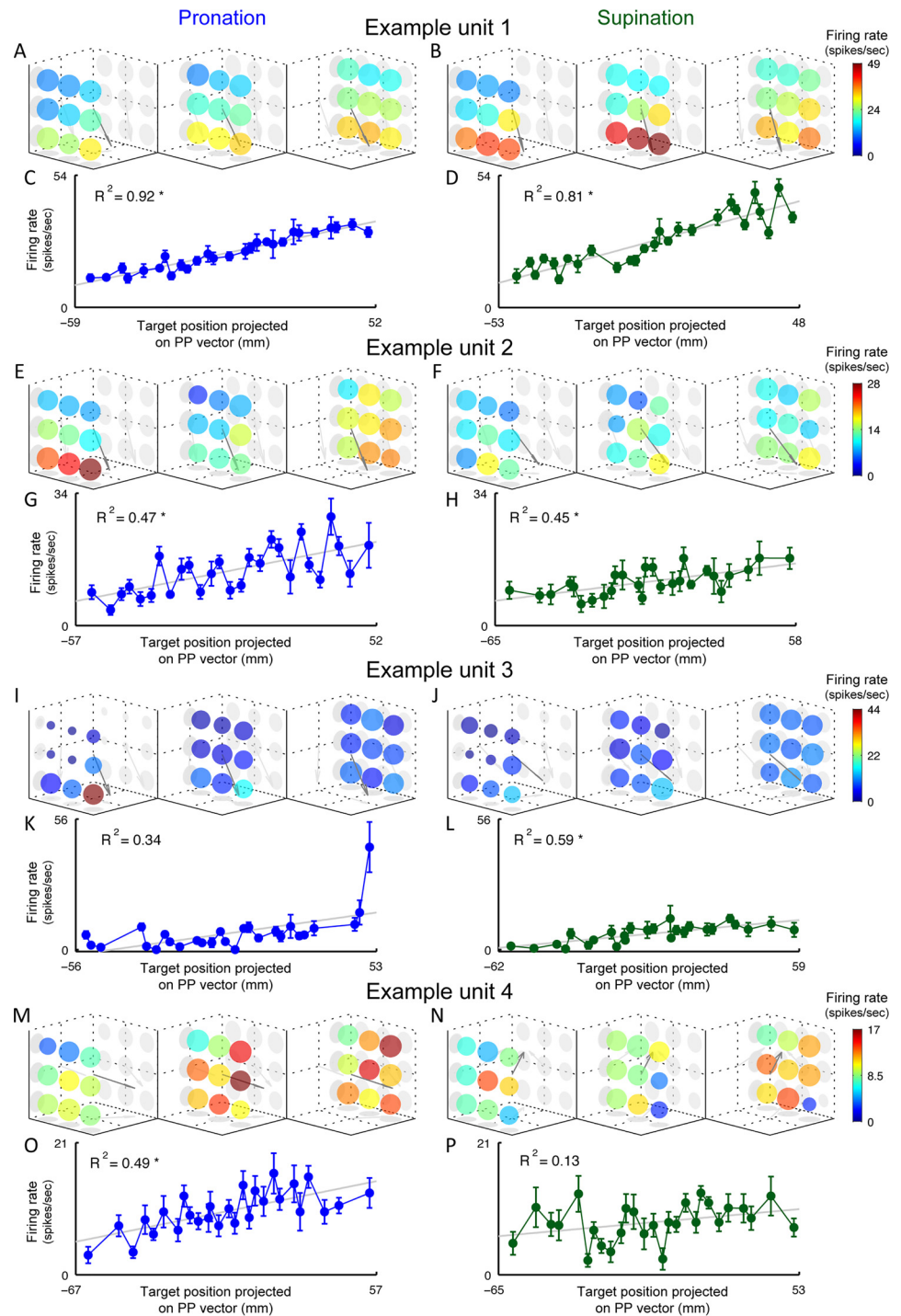
Fig 2. Examples of neuronal response functions. Colored plots show mean firing rates as function of 3D target position. Sphere radius is proportional to the confidence in the mean (SEM normalized by the mean), a smaller radius means lower confidence. The PP vector from the linear tuning-curve is plotted as an arrow. Left panel for pronation, right for supination. Below, mean firing rates ± SEM plotted as function of target positions projected onto PP vector. Gray line shows firing rates expected by linear tuning-curve, and its R^2 (star denotes a significant F-test, $p < 0.01$). **A-D.** Neuron with one of the best fits to the linear tuning-curve in our dataset. Target positions perpendicular to the PP vector have almost equal firing rates. **E-H.** Neuron exhibiting spatial bimodality. **I-L.** Neuron with a spatially local response. **M-P.** Neuron with most common form of nonlinearity with high spatial frequencies (neighboring targets have large differences in firing rates).

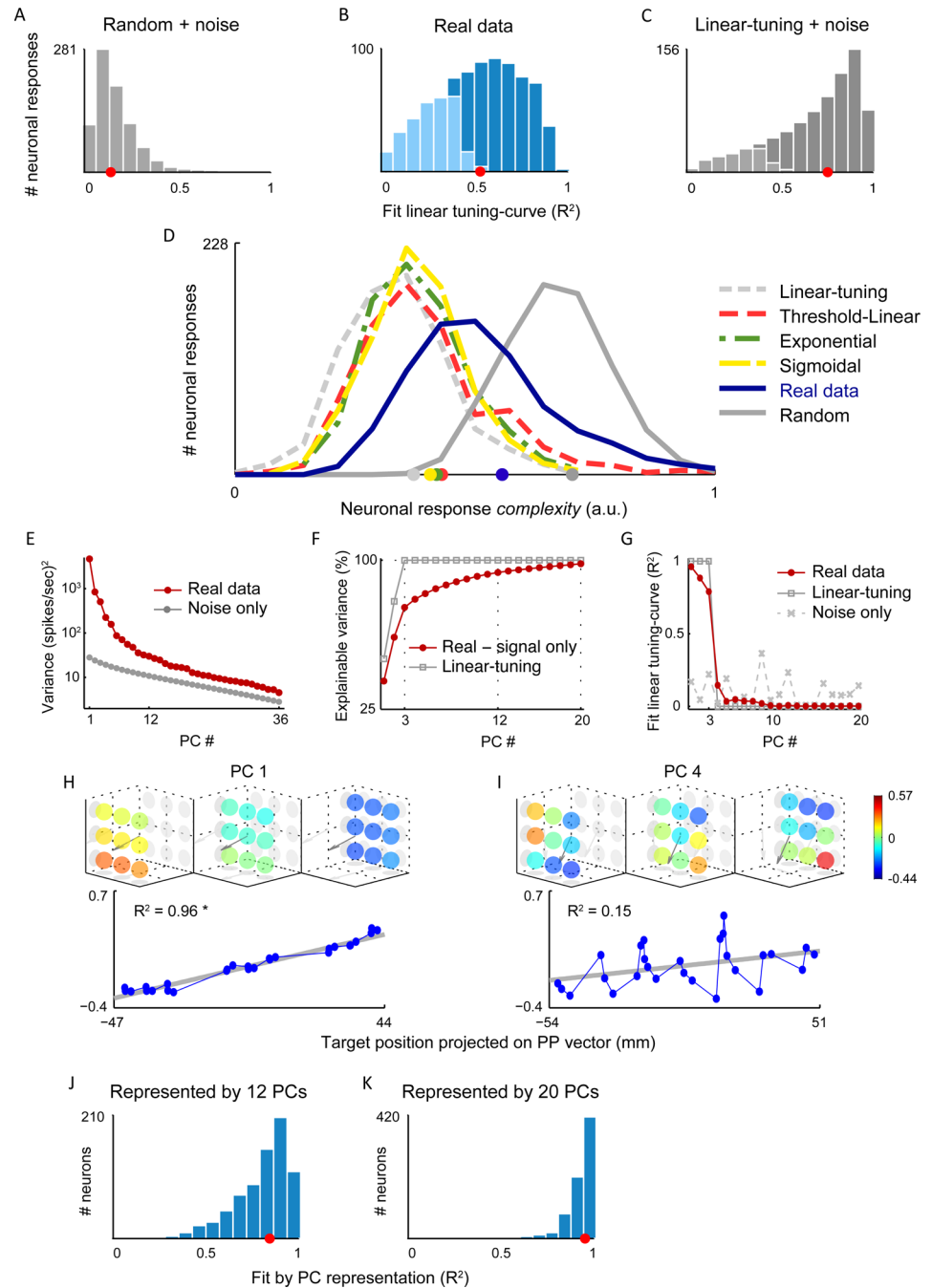doi:10.1371/journal.pcbi.1004910.g002

Fig 3. A. Distribution of R^2 values for fit of linear tuning-curve to uniformly random artificial responses (median = 0.15), with addition of noise traces from real data (repeated 1,000 times). Almost all fits were not significant (F-Test, $p < 0.01$), denoted by light gray bars. **B.** Same as A but for real data, including the responses for all tuned neurons ($N = 411$) for both forearm rotation angles, significant fits (F-Test, $p < 0.01$) are denoted by dark blue (median = 0.52). **C.** Same for the artificial spatial linear control dataset. Distribution (median = 0.76) significantly different from that for real data (Wilcoxon rank-sum test, $p < 10^{-27}$). **D.** Distribution of *complexity measure* across (normalized) response functions. Blue for the real neuronal responses, other colors for matched spatial linear, smooth nonlinear, and uniformly random control datasets, all with added real noise traces. Means (plotted on x-axis) significantly different for controls than for real data (*t*-test, $p < 10^{-40}$ for the closest distribution). **E.** PC variances of the response functions across 54 conditions, and PC variances of noise traces sampled from the data (resampled 1,000). Variance explained by first 12 PCs of data is above variance of first noise PC, and variance of first 36 PCs of data is above the mean noise variance. The 99% confidence interval of the mean of the variance of the noise traces is indicated by the

thickness of the gray line, and is smaller than the round markers. **F.** PCA of response function of both forearm angles (27 conditions each). Percent cumulative *explainable variance* as function of PC number, computed by subtracting the noise variance from total variance of the real data. In gray, the variance of purely spatial linear control (without noise). **G.** R^2 values for fit of PCs to linear tuning-curve, for real data (red), purely spatial linear (gray and squares), or real noise traces (dashed gray and x's). For real data and linear control only first 3 PCs have a significant fit to linear tuning-curve. Mean R^2 for fit of each of noise PCs, across the 1,000 resamples is plotted, showing chance level for fitting noise to linear tuning-curve. **H.** 1st PC vector of real data plotted in the same format as the neuronal responses (Fig 2A and 2C); here color denotes the PC coefficient. This PC vector is the most spatially linear and represents one dimension of the spatial linear component. **I.** Same for 4th PC vector. This PC vector is highly nonlinear and is part of basis representing the nonlinear structure of responses. (See the first 20 PC vectors in S3 Fig). **J.** R^2 distribution for the "model" formed by representing each response function by the first 12 PC vectors (median = 0.84). **K.** Same as H when using the first 20 PC vectors (median = 0.95).

The neuronal responses deviate from spatial linearity in myriad ways. Some are bimodal, with high firing rates in two regions of space and low firing rates for the targets in between (Fig 2E). Other neurons have spatially localized tuning, responding to only one or two adjacent target locations (Fig 2I). Still others display more complex nonlinear spatial patterns, either on top of a linear component (Fig 2M) or without one (Fig 2N). Strikingly, most of the responses exhibit high spatial frequencies, with large differences in firing rates between neighboring targets in a seemingly unsystematic manner. This causes a zigzag pattern in the projected responses (Fig 2D, 2G, 2O and 2P). Such "salt and pepper" tuning does not support the explicit encoding of motor parameters (hand position, joint angles, etc.) by single neurons, because these should vary systematically with changes in hand position.

We quantified the spatial irregularity of each neuron's responses using a *complexity measure*, based on the distribution of firing rate differences between all pairs of neighboring targets (Methods). The resulting distribution of spatial complexities across the real neurons (Fig 3D) covers values that are significantly larger than for linear and "regular" nonlinear (threshold-linear, exponential, sigmoidal; with added noise as described above) artificial datasets, yet significantly smaller than for an artificial dataset constructed from random responses. Stated another way, the real responses contain more power at high spatial frequencies than conventional parametric linear and nonlinear tuning curves, but not as much as expected for completely random tuning (Fig 3D).

Because the neuronal responses are not well described by standard parametric tuning curves (e.g. threshold-linear, exponential, sigmoidal, etc.), we used principal component analysis (PCA) to quantify their shapes non-parametrically. We also performed PCA on noise traces resampled from our data to implement a procedure developed by Machens et al. [6] for separating signal from noise (Methods). The first 12 PCs of the full data each account for more variance than the 1st PC of the noise (Fig 3E; 99% confidence interval, bootstrap). Using this strict threshold, we find that the responses occupy 12 of the possible 54 dimensions, with the rest assumed to be noise. If, instead, we take the mean of the noise variance across all of its PCs as the noise threshold, the responses are 36 dimensional. These results do not depend on whether or not the data is normalized (Methods). We also performed PCA on the separate pronation and supination data sets, obtaining in each case 12 or 20 dimensions above the maximal or mean noise thresholds, respectively. Regardless of where in the range from 12–36 the exact dimensionality lies, it is much higher than the 3 dimensions expected for linear tuning, and it is presumably sufficiently high to allow these neurons to control the arm muscles required for this task.

To study the signal variance in the data, we subtracted away the noise variance and compared the resulting cumulative *explainable variance* to that of the artificial linear dataset, which is described by only 3 PCs (Fig 3F). The difference between these curves reveals the

contribution of the heterogeneous nonlinear components. The first 3 PCs of the real responses roughly describe the linear component described above (F-Test, $p < 0.01$, Fig 3G, Fig 3H). The 4th and higher PCs (Fig 3I) provide a basis for the heterogeneous nonlinear structure found in the responses (S3 Fig).

How much better can a general model describe the neuronal responses than the linear tuning curves? To answer this question we used the non-noise PCs to construct nonlinear "models" of the neuronal responses and compared these with the data. Using only 12 PCs, results in a dramatically improved description relative to the linear fit, with 93% of neuronal responses having an $R^2 > 0.52$ (the median value for the linear tuning curve fit) and a median R^2 of 0.84 (Fig 3J). Using 20 PCs produces an excellent fit for every neuron (median = 0.95; Fig 3K).

Decoding EMG

To investigate the relationship between position-dependent tuning of M1 neurons and arm muscle activity, we studied how the EMG of the primary muscles used in the task could be decoded from M1 activity. To obtain sufficiently accurate measurements of EMG activity, we used the mean EMG (averaged over a time window, trials, and 6 days; Methods) recorded at 18 arm postures (9 targets times 2 forearm rotation angles; Fig 4A) rather than the full 27. Like the neuronal activities, the EMG of all 5 muscles had a significant nonlinear component, with 3 of the muscles significantly fit by a linear tuning curve as well (F-test, $p < 0.01$). To reconstruct these EMGs from M1 activity, we used single-trial neuronal firing rates combined from neurons recorded across different days (pseudo-simultaneous population activity; Methods). This is a reasonable strategy because the EMGs were averaged over trials and days as well. However we expect this to limit our ability to decode the EMG because using the mean EMG and pseudo-population activity breaks single-trial correlations.

We performed decoding using 1,000 different random cross-validation splits of the data into training and test sets (C.V. repetitions) each consisting of 18 test trials (one per arm posture). We obtained the best performance in cross-validation tests by using *optimal subset selection* [7] in which the linear decoder is constructed with a LASSO algorithm [8]. This selects the optimal subpopulation of neurons for decoding each muscle, rather than using all of the neurons to decode all of the EMGs (Methods). This has the additional benefit of providing information about how the signals that affect different muscles are distributed across the M1 population (S2 Text).

We first determined how well the EMGs could be decoded using the full neuronal responses (see Methods). The decoding predictions using the neural activity capture the nonlinear structure in the EMG even on a *single-trial* basis, outperforming the fit of the EMG to a linear tuning curve, which is based on the *mean* EMG values. The latter is the best that could be accomplished from the neurons if they had purely spatially linear responses and no noise. We assessed decoding performance using the full responses and also different response components separately, by dividing the data into distinct spatially linear and nonlinear components. To make this split, each single-trial firing rate was expressed as the sum of its underlying mean linear component (the value predicted by the linear tuning curve), the mean nonlinear component (the total mean firing rate minus the linear component), and a noise fluctuation for that trial (Methods; S4D Fig). We then constructed spatially linear and nonlinear datasets by keeping either the linear component and the noise or the nonlinear component and the noise, and we repeated our decoding procedure twice, using one or the other of these datasets.

Predicted EMG for the test trials of one C.V. repetition are compared with data for each muscle in Fig 4A. Using the full data, the median correlation coefficient between the decoded and actual EMG signals across the 1,000 C.V. repetitions is 0.81 (Fig 4B), and decoding errors
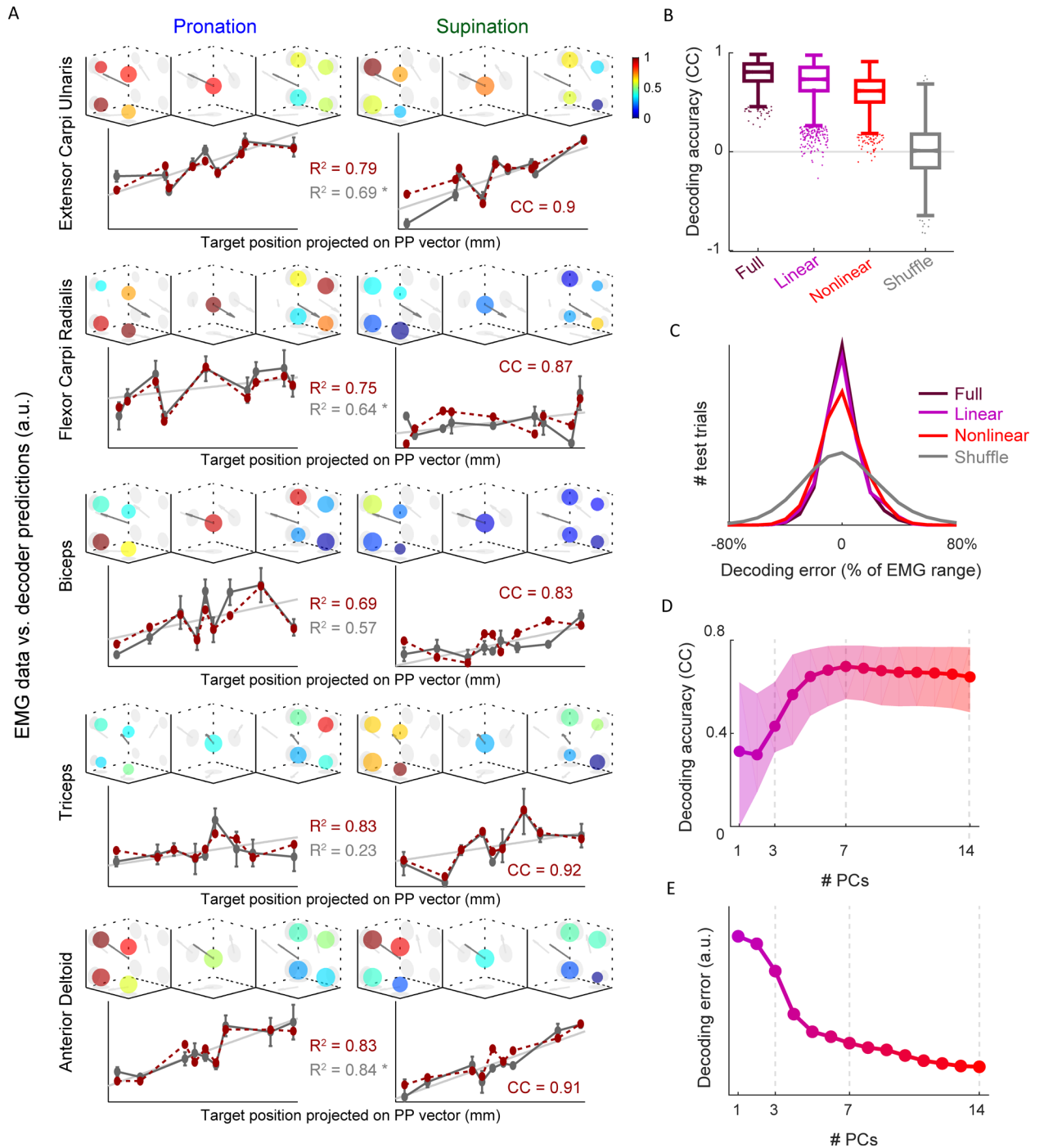
Fig 4. A. EMG as function of arm posture in same format as the neuronal responses (Fig 2). EMGs were fit to the extended linear tuning-curve (Methods, Eq 3), the PP vector in 3D plots, gray line in 2D plots, and R^2 in gray, reflect this fit. Dotted red line shows EMG values predicted by linear decoder on held-out test data from one repetition of cross-validation, with its R^2 and correlation coefficient in red. **B.** Distributions of decoding correlation coefficients between predicted and real EMG values for the test trials of each cross-validation, over all muscles. Box plot shows median, 25% and 75% percentiles, whiskers cover ~ 99% of distribution, dots are outliers. Decomposition of responses into linear and nonlinear datasets described in text (and S3D Fig). **C.** Distributions of decoding errors (real EMG minus predicted) for all test trials and all muscles. The Linear and Full datasets' curves partially overlap. **D.** Decoding correlation coefficients, like in B., as a function of number of PCs that data is projected on to. PCs were learned on training data only, and normalized and trial-averaged before PCA. **E.** Same as D., for S.D. of decoding error.

doi:10.1371/journal.pcbi.1004910.g004

are distributed around zero with a standard deviation of 13% of the min/max range of the EMG (Fig 4C). The decoding accuracy using only the linear component (median CC = 0.73, S. D. decoding error = 14%; Fig 4B and 4C) is somewhat worse than decoding using the full data, and decoding using only the nonlinear components is slightly worse still (median CC = 0.62, S. D. decoding error = 16%; Fig 4B and 4C). Chance level, as quantified by a shuffle control, resulted in a median correlation coefficient of 0.01, and decoding errors with a standard deviation of 29% (Fig 4B and 4C). Thus, the EMG could be decoded far above chance level using either the linear or nonlinear datasets, and the nonlinear component contributed to the accuracy of the full decoding.

In a second approach, we decoded the EMG using increasingly complex nonlinear datasets by projecting the data onto increasing numbers of PCs of the neuronal responses (as described previously). As PCs representing nonlinear structure are added, the decoding accuracy improves, achieving the maximal (median) correlation coefficient with 7 PCs (Fig 4D) and the minimal (standard deviation) decoding error with 14 PCs (Fig 4E). These results confirm that nonlinear tuning contributed significantly to the information about muscle activity contained in the recorded M1 responses.

Non-separable interaction between hand position and forearm angle

We now address the impact of forearm angle on hand-position tuning, starting by looking for two simple effects, an additive shift and a multiplicative gain change between the two forearm angles. To examine a possible linear shift, we computed the baseline firing rate for each neuron, defined as the lowest firing rate across targets. The distribution of baseline firing rate differences (S.D. = 4.3 spike/s) was significantly greater than what would be expected purely from noise (bootstrap control; S.D. = 1.4 spikes/s; F-test for variances, $p < 10^{-48}$), indicating a significant additive shift between the firing rates for the two forearm angles. To look for gain changes, we examined the range of the firing rates across all targets for the two forearm rotation angles. Again, the distribution of gain differences (S.D. = 8.7 spikes/s) was significantly broader than for the bootstrap control (S.D. = 3.3 spikes/s, F-test for variances, $p < 10^{-31}$), indicating a gain change.

If baseline shifts and gain changes were the only impact of forearm rotation, the correlation coefficient between the firing rates of any neuron across the two forearm angles would differ from 1 only due to noise effects (Methods). Instead, we found that the distribution of correlation coefficients between responses for pronation, and supination is broad (Fig 5B, median = 0.54) and very different from the distribution expected solely from noise (Fig 5C, median = 0.93, bootstrap of the medians, $p < 10^{-64}$). On the other hand, most of these correlation coefficients are positive, and their distribution is not at all like that for correlation
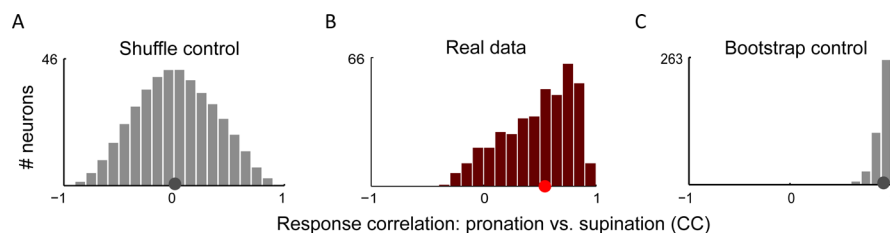
Fig 5. **A.** Distribution of correlation coefficients between pronation and supination for shuffle control, created by computing correlation coefficient between each pronation response function and supination response of 1,000 other neurons (with replacement), median = 0. **B.** Same for responses of real (tuned) neurons, median = 0.54. **C.** Same, for a bootstrap control, median = 0.93, drastically different than for real data (Bootstrap of the medians, $p < 10^{-64}$).

doi:10.1371/journal.pcbi.1004910.g005

coefficients between random pairs of responses ([Fig 5A](#)). Thus, the *shapes* of the hand-position tuning curves change as a function of forearm angle beyond a baseline shift and gain change, but not as much as if the tunings in these two cases were unrelated. This suggests that the independent task variables (the joint angles constrained by target position, and the forearm rotation angle) are *non-separable* at the level of M1, and thus they appear to have undergone nonlinear mixing.

A random feedforward model replicates the data

In order to further understand the data, we searched for a minimalistic neural network model with biologically realistic constraints that could give rise to the empirical results. The nonlinear component of the responses we have studied appears quite random, but the presence of a strong linear component would seem, at first sight, to rule out a model based purely on random target-related inputs to M1 neurons. There is, however, a complication because physiological constraints impose a degree of smoothness on the tuning curves, and a random set of responses might exhibit a degree of regularity, including linearity, purely because of smoothness. To address the degree to which smoothness constraints impose a linear component on responses to random inputs, we constructed a model in which M1 neurons are driven by inputs conveying information about target location.

In the model, target position is represented by a population of neurons that have Gaussian receptive fields (one dimension of which is illustrated in [Fig 6A](#)) centered on particular 3-dimensional preferred target locations. This is consistent with the properties of parietal reach area neurons [9–11], but this population could also correspond to neurons in premotor cortex. The model input neurons have receptive fields with different preferred target locations and, at first, they have the same widths. Later, we extend the model to include more realistic heterogeneity in input receptive field widths.

Each target generates a Gaussian "bump" of activity across the population of input neurons. Each M1 neuron is connected to a random selection of 10% of the input neurons. As a result, each M1 neuron received about 330 inputs that are active somewhere in the workspace and about 50 active inputs at each target location. These inputs are multiplied by random synaptic weights ([Methods](#)) and the result is summed to produce the total input for each neuron. The response of each model M1 neuron is then computed by passing its total input through a threshold-linear firing-rate function. The threshold for this response function depends on a threshold parameter that, in this initial model, is set to the same value for all M1 neurons to match the mean *coding level* of the real neurons (0.85). The coding level is the fraction of conditions that cause a neuron to respond either at a level significantly different from 0 ($p < 0.01$, t-test, Bonferonni corrected) or ≥ 5 spikes/s.

Once the coding level is set, this first-round model has only a single free parameter, the input tuning curve width (our model fits are not sensitive to other features of the model, such as the total number of input-layer neurons, and the number and strength of synapses per M1 neuron). This width determines the smoothness of the M1 responses and this, in turn, controls the tradeoff between spatial-linearity and complexity ([Fig 6B](#)). Wider input receptive fields lead to increased smoothness and spatial linearity, whereas narrower fields increase spatial complexity. For a receptive field width corresponding to a visual angle of 12°, the model M1 responses have the same mean complexity measure as the real neurons ([Fig 3D](#)). Interestingly, this width is consistent with the values reported for parietal cortex neurons [9, 12].

Surprisingly, this simple random model does a good job of matching both the linear and nonlinear components of the real neurons, ([S5 Fig](#), compare to [Fig 2](#)). Even though we only used the mean complexity and mean coding level of the data to set the 2 parameters of the
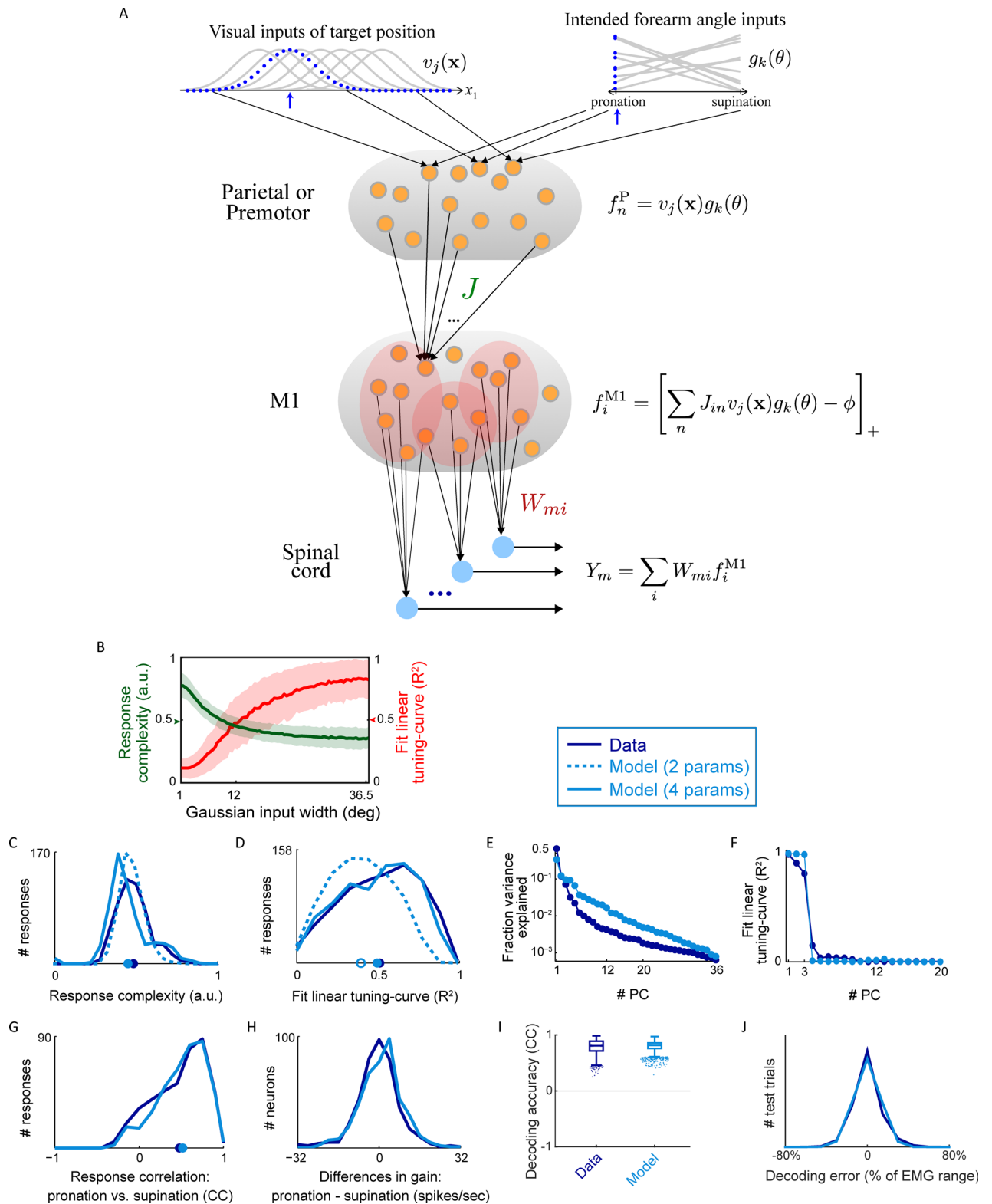
Fig 6. A. Feedforward neural network model for arm posture control. Visual inputs conveying target position are represented by array of 3D Gaussian units (here illustrated in 1D). For given target position (blue arrow) population response is a Gaussian bump (blue dots). Inputs representing intended forearm rotation angle are carried by second population of input neurons, each having a random preference for either pronation or supination, the population of responses for a pronation condition is shown in blue. The shape of their tuning curves

for intermediate angles were not sampled in our task; they are drawn as lines only for illustration. Presynaptic parietal or premotor responses are modeled as a multiplicative interaction of the two input streams from a subset of "nearby" inputs from each (to maintain tuning). Each M1 response function receives a random set of 10,000 presynaptic inputs, but only a small fraction of these are active at any target. These are mixed through random connectivity J and passed through a threshold-linear nonlinearity. The EMG for each muscle is obtained from a linear combination of a subpopulation of such M1 neurons. These synaptic weights are the only ones in the model that require adjustment. **B.** Mean ± S.D. of response function complexity (left vertical axis) and R^2 of fit to the spatial linear tuning-curve (right vertical axis) of model-generated responses (2-parameter model), as a function of the Gaussian width parameter for the visual inputs. This single parameter accounts for the higher spatial-linearity (wider Gaussian inputs) or higher spatial complexity (narrower Gaussian inputs) found in different response functions in the population. Arrows mark the mean values for the real data. **C.** Distribution of response function complexity for real data (blue, re-plotted from Fig 3D, mean = 0.49), and for the model-generated responses for the 2-parameter model (teal dashed line, mean = 0.49) and for 4-parameter model (solid line, mean = 0.46). Although only the mean of the data distribution was used for fitting, the model reproduces the entire shape of the data distribution well.
D. Distributions of R^2 values for fits to the linear tuning-curve for real data (mean = 0.5, S.D. = 0.22; re-plotted from Fig 3B), 2-parameter model (mean = 0.4, open circle, S.D. = 0.19), and 4-parameter model (mean = 0.49, S.D. = 0.23). **E.** Fraction of total variance explained of the PCA of the (54 conditions) real (adapted from Fig 3E) and model generated response functions. **F.** R^2 values for fits of PCs to linear tuning-curve for the data- (Fig 3G) and the model-generated responses. **G.** Distribution of correlation coefficients between the pronation and supination responses for each neuron for the data (Fig 5B; mean = 0.48, S.D. = 0.31) and the model (mean = 0.51, S.D. = 0.28). This result involves no additional parameter fitting. **H.** Distributions of differences in gain for each neuron across forearm angles for the data (S.D. = 8.7 spikes/s) and the model (S.D. = 8.4 spikes/s). We scaled response of each model neuron for both forearm angles to match the range of firing rates of a randomly chosen real neuron without using any forearm angle information in this scaling. **I.** EMG decoding correlation coefficients using the real data (Fig 4B) and the model response functions. **J.** Same for the distributions of decoding errors (Fig 4C).

doi:10.1371/journal.pcbi.1004910.g006

model (the input tuning-curve width and the M1 threshold), the entire complexity measure distribution for the model responses is very similar to that of the real neurons (Fig 6C). In addition, the distribution of R^2 values for the fit to the linear tuning curve is similarly broad (Fig 6D). Thus a feedforward model receiving input with structured tuning through random synapses can account for both the spatially linear and nonlinear components without any explicit tuning of its synaptic connections (i.e. with purely random connectivity).

The 15% undershoot in the standard deviation of the linear tuning curve R^2 distribution (Fig 6D) for the two-parameter model can be improved by extending the model slightly. In the extended model, each M1 neuron is assigned a threshold to match the coding level of one of the real neurons (randomly chosen), and the input receptive field widths are drawn independently from a uniform distribution parameterized by a mean and range (Methods). We chose the mean and range of the tuning curve width distribution to obtain the best match between the model and data for the mean and standard deviation of the linear tuning curve R^2 distribution (Fig 6D).

The neuronal responses of the enhanced model match the real data extremely well (Fig 6C–6J). The fraction of variance explained by the PCs of the model data matches the general shape seen for the real data, although the nonlinear components of the model are represented slightly more uniformly across its PCs (Fig 6E). This shows that the model responses capture approximately the same dimensionality as the real data. In addition, the first 3 PCs of the model neurons capture the spatial linear component as they do for the real data (Fig 6F). We also checked whether the M1 model could reproduce the recorded EMG activity. To do this, we added a set of linear readouts to the model (Fig 6A, bottom) and adjusted the weights of these readouts to fit the EMGs, using the decoding scheme used for the real neurons (described above). Relative to the small number of muscles, the neurons form an overcomplete basis. As a result, using the noise-free model neurons decodes the EMGs almost perfectly. For a fair comparison to the performance of the real data, we added noise samples to the model neurons, resampled from the real data (Methods). Decoding the EMGs using the model neurons with noise produced decoding correlation coefficients (median CC = 0.82 for the model vs. 0.81 for the data; Fig 6I) and decoding errors (S.D. = 12% of EMG range vs. 13%, respectively; Fig 6J) comparable to that of the real neurons. These decoding results provide additional confirmation that the model M1 neurons capture the dimensionality and frequency content of the real neurons.

To model the effects of forearm rotation angle, we let the input neurons that drive the model M1 neurons depend on forearm angle as well as on target location (Fig 6A, top right). The forearm angle dependence involves multiplying the Gaussian function of target location for each neuron by a factor, chosen independently for each input neuron, that takes a different value for pronation or supination, with the values chosen randomly (tuning of the forearm factor could have any shape for intermediate values, since these were not sampled in our task). Thus, the forearm angle inputs act as a gain modulation of the visual inputs consistent with responses found in premotor cortex [13] and eye-position gain field modulation of visual responses in parietal cortex [12, 14]. With forearm angle included in this way, the correlations between the model neuronal responses for pronation and supination are extremely similar to those for the real neurons (98% similar in the mean and S.D. of the two distributions; Fig 6G), without any additional parameters or fitting. The distributions of differences in gain across forearm angles are also very similar (S.D. for the data = 8.7 versus for the model = 8.4 spikes/sec, Fig 6H). These results are a consequence of the random nonlinear interaction of visual target and forearm angle signals at the input stage of the model, suggesting a mechanism that may underlie the non-separable mixing in M1 discussed above.

Discussion

Cosine tuning has played a dominant role in discussion of the tuning of M1 responses during reaching movements [15–18] but, due to many conflicting findings, this approach has fallen out of vogue [19–23]. During arm posture control, we found that M1 neuronal responses show a heterogeneity of irregular nonlinear responses that includes high spatial frequencies (Fig 2), in addition to the linear (cosine-tuned) component described previously [1–3]. Due to this complex nonlinear component, it is unlikely that any reasonable coordinate choice could lead to a simple parametric description of these responses. This is in contrast to the long-standing tradition of using parametric tuning curves and encoding models to describe neuronal responses. Despite its irregularity, the nonlinear response component contributes to decoding the EMG of the major muscles used during arm posture control (Fig 4).

It is natural to assume that regular neuronal responses imply structured (and in network models, learned) input synaptic connectivity [24–26]. However, we found that random connectivity could reproduce the data (Fig 6C–6J), with regularity, in particular linearity, arising from smoothness. This smoothness stems from assumptions about the nature of the inputs to M1, assumptions that are consistent with the known physiology of the target-position coding regions [9, 12]. Importantly, this smoothness was not imposed by limits on the resolution provided by the task (as an extreme example, a task with only two targets would unavoidably produce linear tuning) or by the analyses we performed [27, 28]. Despite its simplicity and small number of adjustable parameters, the model accounts for not only the linear (Fig 2A) and nonlinear (Fig 3D) components, but also their distributions across the neuronal population (Fig 6). This is a rare example in which both the regularity *and* the diversity of a population of neural responses have been replicated by a neural network model. More generally, the model provides support for the idea that some neural circuits use random connectivity to generate rich and high-dimensional representations, and produce their outputs by tuning only their output synaptic weights [29–32].

Firing rates were constant during posture control and feedback did not have a measurable effect on the neuronal responses (S1 Text). This implies that the population has either relaxed into fixed points or is dominated by the (target position and intended forearm angle) inputs external to M1. Either way, any internal recurrent inputs from other M1 neurons cannot generate the tuning to arm posture, which is why we focused on a feedforward network architecture.

Recurrent dynamics are likely to play an important role in how M1 generates time-varying movements, such as reaching [33, 34]. Such recurrent connections are unlikely to be random and may require synaptic modification [35].

A number of influential theoretical models of M1 function have taken a normative engineering-inspired approach [36–40]. By combining the equations of motion of a model arm with putative population M1 responses with features such as cosine tuning, these models can account for many psychophysical features of arm movements. The normative approach has helped clarify the relationships between the arm's biomechanics and resulting movement variables, and has highlighted the caution necessary in interpreting the correlations between movement variables and neuronal responses. However, as has long been acknowledged [36], due to the considerable dimensionality reduction from M1 to muscles, biomechanical properties do not generally suffice to constrain models of cortical computation. The model we have presented here does not attempt to be a general model of M1 function, but it does incorporate representations in the inputs to M1 that are consistent with known physiology and neuronal properties that are biologically realistic. We believe that understanding M1 computation will require models that integrate *neural* constraints, in addition to constraints imposed by the periphery.

The idea that sensory inputs are mapped to motor commands through a series of sequential coordinate transformations has received a lot of attention [41–45]. In this view, sensory representations lead to kinematic representations and only finally to joint torques and muscle activity. However, many daily actions require combinations of task demands, involving hand position, arm posture, endpoint and segmental forces. Therefore, intended endpoint forces, for example, must sometimes be represented by the inputs to M1 and not only by its outputs. In general, the sensorimotor system must be able to process parallel sensory and intentional signals, each ultimately shaping the activity of the same muscles. Our experiment involved demands on both hand position and forearm rotation angle, a degree of freedom that does not affect hand position. The neural responses show an interaction between these two independent inputs that is non-separable (i.e. cannot be decomposed into a sum or product of functions; Fig 5). This suggests that these signals are already mixed nonlinearly before the level of M1. Most encoding models introduced in the M1 literature, whether linear [46–48] or nonlinear [49, 50], produce separable functions. In this view, M1 responses are an example of *nonlinear mixed-selectivity* [32], representations that have been shown to have considerable computational power in tasks requiring nonlinear combinations of multiple parameters [51]. We incorporated this finding into our model by basing the inputs to the model M1 neurons on the multimodal visual and arm-posture-dependent receptive fields found in premotor cortex [13, 52–56] and posterior parietal cortex [57–62]. The model places the idea that such multimodal activity reflects the multiplexing of several sensorimotor parameters [63, 64] within the context of a class of basis-function network models [65–67]. In this view, the multimodal responses found in parietal and premotor cortex may be evidence for the nonlinear mixing of parallel task variables, in addition to sequential sensorimotor transformations. Because of this basic feature of its construction, our model predicts that during experiments with additional concurrent task demands, M1 responses should show non-separable interactions across all task dimensions. This can be tested experimentally, by, for example, repeating our experiment with two different load conditions at the hand, in addition to varying hand position, and forearm rotation.

Posture control could, in principal, be implemented by a simple "on-off" cortical signal that causes ongoing spinal feedback to maintain current muscle activation levels. Even a transiently tuned cortical motor command could activate ongoing spinal control [68]. In contrast to these possibilities, we found that most M1 neurons are tuned even 1–1.8 seconds after a target is reached and posture is held constant. By using a powerful decoding algorithm, some of the hand jitter during the target hold epoch could be predicted [69]. However, having found no

correlations between firing rates and hand jitter suggests that such signals are weak and that the ongoing M1 responses primarily represents a feedforward (intentional) motor command, with ongoing corrections executed by sub-cortical feedback loops. This is in contrast to the strong and fast feedback corrections seen in M1 in response to large unexpected perturbations of the arm [70].

The muscle activations required for a motor task are typically nonlinear functions of task parameters. In principle, M1 neurons could encode task parameters linearly and only down-stream spinal cord circuits would generate the required nonlinearities. Alternatively, the required nonlinearities may already be present in M1 responses, which could then be read out in a purely linear manner by spinal cord circuits (an idea we incorporated into our model; Fig 6A). Obviously, these are two extremes, and the truth probably lies between them. Neverthe-less, our data show that the nonlinearities needed for correct muscle activation are already present in the M1 responses, and these can be read out linearly to produce the recorded EMGs (Fig 4). In a series of studies, Miller and colleagues found an invariant relationship between M1 and EMG activity by decoding EMGs in a variety of reaching and grasping tasks [71–73]. In our study, by first studying the statistical structure of the population of response functions (Fig 3) and then combining analyses of encoding with decoding (Fig 4B–4E), we showed that the full structure of the M1 responses, both the spatial-linear component and heterogeneous non-linear components, contribute to creating the EMG. This diversity provides a basis with high-dimensionality (Fig 3E), providing enough dimensions in neural space to span the space of independently controllable muscles of the arm. By using the subset selection algorithm, we found that specific subpopulations of M1 neurons more optimally control individual muscles, or groups of muscles. These subpopulations partially overlap (S4A–S4C Fig), consistent with the gross topography found in M1 [74].

Neural representations involve a compromise between two incompatible demands: accuracy and noise tolerance. M1 arm-posture related responses, situated between the noise tolerant prop-erties of regular (linear) tuning curves and the high degree of accuracy provided by irregular tun-ing, provide an example of such a compromise. An interesting prediction of the model, related to this observation, involves the increased motor accuracy associated with foveating a target. The presentation of a visual target initiates a saccade that brings it to the fovea [75], improving target localization [76]. By what mechanism does the resulting increased in visual acuity benefit motor execution? The dependence of M1 response complexity on the width of the target-position inputs in our model (Fig 6B) suggests an answer. Because foveal receptive fields are narrower [10], fixat-ing the target should bias the smoothness-randomness tradeoff of M1 neurons towards random-ness, increasing response complexity with higher spatial frequencies and resulting in more accurate motor commands. This prediction can be tested experimentally by comparing M1 neu-rons while arm posture is maintained at a visual target viewed peripherally versus foveally.

Methods

Two monkeys (Macaca fascicularis; PK—4.3 kg male, and BR—3.4 kg female) were trained in our behavioral task, in a standard 3D virtual reality setup (Fig 1). They controlled a cursor by active LED markers attached to their right hand, measured by a motion capture system (Phoe-nix Technologies, sampled at 100 Hz), while their left arm was comfortably restrained. An opaque panel concealed their arms from view.

Animal care was in accordance with the National Institutes of Health guidelines and was approved and supervised by the Hebrew University committee on animal experiments. The animals were housed in groups of four in an open playroom that included trees, toys, and enrichment. Surgery involved the use of standard anesthetics (Kamacaine, Medetomidine,

Ketamine, and Isoflurane), antibiotics (Cefazoline), and analgesics (Carprofen) that were administered pre-, during, and post-operatively as needed. Throughout the experiments, the animals were carefully monitored 7 days a week (i.e. including rest days), and after conclusion of the experiments were retired to a primate shelter park.

Neural & EMG recording

After training, a chronic 96 microelectrodes array with 1.5 mm electrodes (Blackrock Microsystems) was implanted in the arm area of contralateral rostral M1 (S1A Fig) using the surgery protocol described elsewhere [77]. Single-unit action potentials were sampled at 30 kHz and sorted using an automatic adaptive spike-sorting algorithm with human adjustment (Blackrock Microsystems). Spike waveforms were inspected visually offline and, due to the large neuronal yield, only very well isolated single units were included. A firing-rate stability analysis was performed in a semi-supervised manner using a Hidden-Markov Model that segmented even subtle changes in firing rate. Only neurons with firing rate that was stable throughout the recording session and had at least 1 spike/minute were included in the analysis. Therefore our inclusion criteria was based only on recording quality, and introduced no task specific selection bias. We included 378 neurons from monkey PK, 132 neurons from monkey BR. After verifying that the dataset from each monkey produces equivalent results (S2 Fig), we report all 510 neurons together.

Electromyography (EMG) was recorded on 6 days from monkey PK, from the Anterior Deltoid, Biceps, Triceps, Flexor Carpi Radialis, Extensor Carpi Ulnaris, using double-differential surface electrodes with pre-amplifiers x 20 (Motion Lab Systems). The raw EMG signals were down-sampled, rectified, and root-mean squared (20 ms window), and normalized per muscle for each day.

Behavioral task

Monkeys made continuous, instructed-delay, target-to-target reaching movements between 27 targets in 3D space. On each trial, after reaching the target and receiving a drop of food reward (275 ms later), the monkeys had to maintain the static position of their hand at the target and their current forearm angle for another 750–1,500 ms (*target hold epoch*). After successfully reaching the target, any deviation from the target would cause the trial to fail and the screen to go blank for 1–2 seconds. This target hold epoch was before any of the sensory cues of the upcoming trial, including the next target, had appeared. The monkeys were trained to perform the task with their forearm in either the pronated (palm-down) or supinated (palm-up) postures (Fig 1), and these were instructed in alternating blocks of 25 trials. The cursor's graphics were the same in both forearm angles, and in this way we dissociated the kinematics of the cursor from the biomechanical state of the arm. The 27 targets were arranged in a virtual cube (3 x 3 x 3), where the distance between the 2 furthest targets was 12.1 cm. This distance was a significant portion of the full range of motion for these monkeys, as the lengths of their upper limbs were 23.3, 21 cm from shoulder to wrist for monkeys PK and BR, respectively. Target radii were 1.7 cm for monkey PK (and 2 cm on some control days), and 2.5 cm for monkey BR. Data was analyzed from 38 recording days for monkey PK and 10 recording days for monkey BR. On nine control days of monkey PK's experiment and when recording the EMG, only 9 targets (the corners and center of the virtual cube) were presented.

Data analysis

We analyzed the data during a 200 ms period in the target hold epoch, which was 1,025–1,775 ms after reaching the target (750–1,500 ms after the reward), and 50 ms before the first instruction of the next trial (S1B Fig). The time ranges are due to an experimental design with

randomly variable intervals. Both firing rates and hand position were most stable during this period. Firing rates were calculated by averaging over this 200 ms time window and mean firing rates over trial repetitions to the same *condition* (target position and forearm rotation angle). There was a mean of 14 trial repetitions, per condition, per day, giving a sizeable sample to estimate the underlying firing rates.

Spatial linear tuning. To measure the linear component of the spatial structure of each neuronal response, we fit them to the linear tuning curve described previously [2] using multiple linear regression:

$$\overline{r}^{linear}(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 \tag{1}$$

where the mean firing rate of each neuron, $\overline{r}^{linear}(\mathbf{x})$, at the mean hand position (binned by targets) $\mathbf{x} = (x_1, x_2, x_3)$ is a linear function of the position, with regression coefficients a_0, a_1, a_2, a_3. If we define the *Preferred Position Vector* as $\mathbf{PP} \equiv (a_1, a_2, a_3)$, we can write this equation equivalently in polar coordinates as:

$$\overline{r}^{linear}(\mathbf{x}) = a_0 + \|\mathbf{PP}\| \cdot \|\mathbf{x}\| \cdot \cos \varphi_{\mathbf{PP},\mathbf{x}} \tag{2}$$

where $\varphi_{\mathbf{PP},\mathbf{x}}$ is the angle (in radians) between the mean hand position and the $\mathbf{PP}$ vector. The $\mathbf{PP}$ vector points in the direction in space along which the firing rate increases linearly, and its norm, $\|\mathbf{PP}\|$, is the slope of this linear modulation. According to this model, the firing rates at hand positions in each plane orthogonal to this vector should be equal.

To quantify the linear dependence of the firing rates on the full arm posture (i.e. forearm rotation angle in addition to hand position), we additionally fit each neuron to two other equations. An *Extended* spatial linear tuning curve fits all 54 conditions with one $\mathbf{PP}$ vector but allows different baseline firing rates for each of the two forearm angles,

$$\overline{r}^{extended}(\mathbf{x}) = a_0 + \|\mathbf{PP}\| \cdot \|\mathbf{x}\| \cdot \cos \varphi_{\mathbf{PP},\mathbf{x}} + a_4 \theta_f \tag{3}$$

where θ_f is the angle of forearm rotation. A third equation accounts for the forearm rotation angle exerting a multiplicative, instead of an additive, effect,

$$\overline{r}^{multiplicative}(\mathbf{x}) = a_0 + g(\theta_f) \cdot \|\mathbf{PP}\| \cdot \|\mathbf{x}\| \cdot \cos \varphi_{\mathbf{PP},\mathbf{x}} \tag{4}$$

where g is defined to be 1 when $\theta_f = p$ and a value determined by fitting when $\theta_f = s$, for pronation and supination, respectively.

Extracting and quantifying hand jitter. We detected hand jitter by finding instantaneous hand speeds that exceeded the median of the distribution of maximal jitter speeds per trial (2.48 cm/s), and then found the beginning of each hand jitter using an algorithm developed by Amos Arieli [78]. Each trial could have between zero to several jitters. Different choices of the threshold, which slightly changed the number of jitters included in this analysis, had no significant effect on the results. Mean 3D directions were calculated from the hand jitter segments using orthogonal regression and were distributed in all directions in 3D space. Using multiple linear regression, we fit the instantaneous firing rates to the velocity using the multiplicative velocity tuning curve [48]:

$$r(t - \tau) = b_0 + b_1 \cdot \|\dot{\mathbf{x}}\| + b_2 \cdot \|\dot{\mathbf{x}}\| \cdot \mathbf{PD}' \cdot \mathbf{x} \tag{5}$$

where the mean firing rate, r, with a lead/lag of τ ms to the beginning of the hand jitter, is a linear function of the hand's speed, $\|\dot{\mathbf{x}}\|$, and direction of movement, $\mathbf{x}$ (in Cartesian units), and $'$ denotes transpose. $\mathbf{PD}$ is the *preferred direction*—the normalized vector of the direction-specific regression coefficients—and b_2 is its norm. For each neuron we calculated the firing rates in 100 ms windows, repeating the fitting procedure from neuronal activity leading behavior by

300 ms, to lagging behavior by 100 ms, in 10 ms increments. This was done for each forearm angle separately, and we report the maximal R^2 for each neuron across all temporal lags. We repeated this analysis with either the full firing rates or the firing rate fluctuations (i.e. minus the mean rate for that condition).

Principal component analysis of the neuronal responses. Principal component analysis (PCA) [79] was applied to the responses of the significantly tuned neurons for the full 54 conditions response functions, and then again for each of the forearm rotation angles separately (27 conditions). We calculated PCA for 376 neurons, after excluding 35 tuned neurons from control days with only 9 targets so as not to introduce a bias at the targets they were not sampled.

We used the method described by Machens [6] to calculate an upper bound on the variance of the residual noise by calculating the PCA for *noise trace* estimates sampled directly from the data. Briefly, let the single-trial firing rate of the i-th neuron in the c-th condition for the k-th trial be denoted $r_i(k, c)$. After averaging over the trials for each condition, its response function is $\bar{r}_i(c)$, where the bar denotes the average over trials. The residual noise estimates were created by sampling two different single-trial firing rates randomly for each condition, subtracting them, and normalizing:

$$\bar{\eta}_i(c) = \frac{1}{\sqrt{2K}}(r_i(k, c) - r_i(l, c)) \qquad , k \neq l \qquad (6)$$

where there are K trials for this condition, and the bar denotes that this is an estimate of the average noise trace (over trials) for that neuron and condition. Subtracting two single-trials removes the shared underlying signal component, leaving a noise fluctuation that is from the same noise distribution. $\bar{\eta}_i(c)$ can be conceived as the residual noise *trace* remaining from averaging over a finite number of trials. Clearly as $K \to \infty$, $\bar{\eta}_i(c) \to 0$. We then calculated the PCA separately for this dataset of noise traces, which produces an upper bound on the variance explained by the noise in the data. For the complete derivation of this method see the supplementary materials of [6]. We repeated this procedure 1,000 times (each repetition choosing a pair of single trials $k \neq l$ for each neuron and each condition) to calculate bootstrap estimates of the 99% confidence intervals for the variance of the noise.

To control for the potentially disproportionate effect of neurons with large firing rate ranges, we repeated the analysis by first normalizing each response function (mean subtraction and normalizing their range to 1). Only significantly tuned neurons were used for the PCA analysis to avoid magnifying neuronal noise by normalization. To control for neurons with high firing-rate variance, alternatively, we pre-processed the single-trial firing rates with the variance-stabilizing transformation for the Poisson distribution [80], which replaces $r_i(k, c)$ with its square root $\sqrt{r_i(k, c)}$. The un-normalized data and both of these normalizations produced essentially the same results.

Complexity measure. In order to measure the complexity of each neuron's response function non-parametrically, we calculated the discrete derivative (Lipschitz continuity) between the mean firing-rate at each target, $\bar{r}(\mathbf{x}_k)$, and at each of its closest targets, $\bar{r}(\mathbf{x}_l)$. We defined the *complexity measure* for that i-th response function, as the standard deviation over these values:

$$\text{complexity}(i) = \text{stdev}\left\{ \frac{|\bar{r}(\mathbf{x}_k) - \bar{r}(\mathbf{x}_l)|}{\sqrt{\sum(\mathbf{x}_k - \mathbf{x}_l)^2}} \,|\, \forall \, k, l, \text{s.t.} \sqrt{\sum(\mathbf{x}_k - \mathbf{x}_l)^2} = D_{min} \right\} \qquad (7)$$

where, k and l are pairs of targets with the minimal distance D_{min}. We used the normalized responses (with range 1, described above), in order to compare neurons with different firing

rates. We compared the distribution of this complexity measure across the real neuronal responses to a purely spatial-linear control dataset (firing rates generated from the fitted linear tuning curve parameters, of each neuron) and several nonlinear controls, which passed the spatial-linear response through an additional nonlinearity (threshold-linear, exponential, or sigmoidal). We matched the range of firing rates of these controls to those of the real neurons and added noise traces (as described above). The complexity measure was calculated for 1,000 resamples of the noise traces, and the mean distribution for each control was reported.

Joint angle tuning curves. The five arm joint angles relevant to this task, shoulder abduction/adduction, shoulder flexion/extension, shoulder rotation, elbow flexion/rotation and forearm rotation, (both monkeys kept their wrist in a stereotyped posture throughout the task) can be divided into 4 that are determined by hand position, which we denote by the vector $\boldsymbol{\theta}(\mathbf{x})$, and the forearm rotation angle θ_f. While the monkeys did not have a single stereotyped elbow swivel angle across the entire workspace, elbow swivel was consistent as a function of the target position held, and hence is considered a member of $\boldsymbol{\theta}(\mathbf{x})$.

We first consider tuning curves in which the average firing rates can be expressed as

$$\overline{r}(\mathbf{x}, \theta_\mathrm{f}) = h(\boldsymbol{\theta}(\mathbf{x})) + g(\theta_\mathrm{f}) \tag{8}$$

where h and g are arbitrary functions. The difference between the response functions (based on hand position) across the 2 forearm angles is

$$\overline{r}(\mathbf{x}, \theta_\mathrm{f} = p) - \overline{r}(\mathbf{x}, \theta_\mathrm{f} = s) = g(\theta_\mathrm{f} = p) - g(\theta_\mathrm{f} = s) \equiv C$$

where p and s stand for pronation and supination, and C is constant that does not depend on hand position. The correlation coefficient between each neuron's pair of responses should only deviate from 1, due to noise because

$$\mathrm{corr}(\overline{r}(\mathbf{x}, \theta_\mathrm{f} = s), \ \overline{r}(\mathbf{x}, \theta_\mathrm{f} = s) + C) = 1 \tag{9}$$

If instead we assume a multiplicative interaction for the encoding of the forearm rotation with the other joint angles,

$$\overline{r}(\mathbf{x}, \theta_\mathrm{f}) = h(\boldsymbol{\theta}(\mathbf{x})) \cdot g(\theta_\mathrm{f}) \tag{10}$$

where h and g are again arbitrary. In this case, the response functions (based on hand position) will be scaled by a gain as a function of forearm angle,

$$\overline{r}(\mathbf{x}, \theta_\mathrm{f} = p) - \overline{r}(\mathbf{x}, \theta_\mathrm{f} = s) = h(\boldsymbol{\theta}(\mathbf{x})) \cdot (g(\theta_\mathrm{f} = p) - g(\theta_\mathrm{f} = s))$$

and again there is no change in response function shape,

$$\mathrm{corr}(h(\boldsymbol{\theta}(\mathbf{x})) \cdot g(\theta_\mathrm{f} = p), \ h(\boldsymbol{\theta}(\mathbf{x})) \cdot g(\theta_\mathrm{f} = s)) = 1 \tag{11}$$

These equations simply make explicit the repercussions of the fact that forearm rotation is independent of hand position and therefore independent of the joint angles that are constrained by it. The extended and multiplicative spatial linear tuning curves (Eqs 3 and 4) are non-parametric versions of each of these types of models (Eqs 8 and 10, respectively).

Decoding EMG. Using single-trial neural activity, we decoded the mean EMG of the five muscles we recorded (Fig 4A). The EMG was decoded in 18 conditions (9 targets per forearm rotation angle) by using the neural activity of $N = 131$ neurons that had at least 15 repetitions for each of these specific conditions. Pseudo-simultaneous population activity was created by concatenating randomly sampled single-trials from each neuron from the same condition. That is, to create one vector of pseudo-simultaneous population activity for the condition c, each neuron contributed a single-trial firing rate of one trial from that condition. Some of

those firing rates may have been recorded simultaneously by chance, but on average most were not, effectively averaging out any single-trial ("noise") correlations. This is a reasonable choice because we decoded *mean* EMG that averaged out single-trial correlations too. Decoding was carried out in a cross-validated scheme: for each repetition ("fold") of cross-validation, 14/15th of the data ($T = 252$ trials) were randomly chosen for training and the remaining 1/15th ($T = 18$ trials) for testing. This was done in a balanced design, such that each test dataset included exactly one repetition for each condition. Cross-validation was repeated 1,000 times.

The linear decoder is an N x 5 matrix **W** of weights such that

$$\mathbf{Y} = \mathbf{R}' \cdot \mathbf{W} \tag{12}$$

where **R** is the N x T matrix with each column equal to the population activity for one pseudo-simultaneous trial, **R'** is its transpose, **Y** is the T x 5 matrix of the mean EMG activity of the 5 muscles. We initially estimated the linear decoders using the *Moore-Penrose Pseudoinverse* (since our case is under-determined):

$$\mathbf{W} = \text{pinv}(\mathbf{R}') \cdot \mathbf{Y} \tag{13}$$

An approach that uses all the neurons for decoding can produce suboptimal performance on cross-validated data. We achieved improved decoding performance by using the LASSO algorithm [8], which selects an optimal subset of neurons. This learning algorithm is a regularized version of least squares estimation, that minimizes the decoder's L_1-norm in addition to the sum of squared residuals, leaving a smaller number of regressors with nonzero weights. The LASSO estimate is found by solving the following (quadratic programming) optimization problem:

$$W = \text{argmin}_W \left\{ \frac{1}{2} \sum_{t=1}^{T} \left(y_t - \sum_{i=1}^{N} r_{ti} W_i \right)^2 + \lambda \sum_{i=1}^{N} |W_i| \right\} \tag{14}$$

This algorithm was repeated for the EMG of each muscle separately, allowing a distinct optimal subpopulation for each muscle. y_t is the EMG activity of the given muscle on trial t, r_{ti} is the firing rate of i-th neuron on that trial, and W is now a weight vector. The meta-parameter λ weights the relative importance of decoding accuracy (left term) versus the regularization (right term) and was selected as the value that minimized the mean squared error on a 5-fold cross-validation of the training data alone. The neural data was mean-centered, where the means were learned only from the training data and likewise for the intercept.

Decoding performance was quantified by the distribution of prediction errors: real EMG minus predicted value, for each of the 18 test trials, each repetition of cross-validation, and for each muscle. In addition, we computed the (Pearson) correlation coefficient between the series of EMG values for the 18 test trials and their respective prediction values. The distribution of correlation coefficients is over the repetitions of cross-validation and muscles.

Decomposition of single-trials into linear vs. nonlinear components. To compare the decoding performance of the nonlinear and the linear components separately, we decomposed the neural activity into 2 distinct datasets. Each single-trial firing rate, $r_i(k, c)$, for the i-th neuron on the k-th trial in condition c, was decomposed into a sum of its underlying mean spatial-linear component (the value predicted by the tuning curve at that hand position), $\overline{r}_i^{linear}(c)$, the mean nonlinear component (the total mean firing rate minus the spatial-linear component), $\overline{r}_i^{nonlinear}(c)$, and that trial's noise fluctuation, $\eta_i(k, c)$, as defined above:

$$r_i(k, c) = \overline{r}_i^{linear}(c) + \overline{r}_i^{nonlinear}(c) + \eta_i(k, c)$$

This is illustrated in S4D Fig for one example neuron's response. To create the dataset comprised of purely spatial-linear components plus single-trial noise, we subtracted from each

single-trial firing rate the mean nonlinear component for that condition (S4D Fig, bottom panel)

$$\overline{r}_i^{linear}(c) + \eta_i(k, c) = r_i(k, c) - \overline{r}_i^{nonlinear}(c) \tag{15}$$

This is identical to taking the spatial-linear tuning curve's predicted value for that condition and adding the trial's noise. Equivalently, subtracting from each single-trial firing rate its respective spatial-linear component, creates a dataset of purely nonlinear components plus single-trial noise (S4D Fig, top panel)

$$\overline{r}_i^{nonlinear}(c) + \eta_i(k, c) = r_i(k, c) - \overline{r}_i^{linear}(c) \tag{16}$$

This is identical to taking the residuals (from the fit to the spatial-linear tuning curve) and adding that trial's noise fluctuation. We repeated the decoding procedure (described above) twice, using each of these datasets as inputs separately. Our decomposition relies on identifying the responses with a significant fit to the spatial-linear component, we therefore repeated this analysis with either $p < 0.01$ or $p < 0.05$ thresholds.

Feedforward neural network model. To discover what are minimal biologically realistic constraints that could give rise to our data, we built and analyzed a feedforward neural network model (Fig 6A). We assumed that visual information about target position is represented by an array of 3D Gaussian input units, consistent with the known receptive field properties of parietal reach area neurons [9–11]. The response of the j-th visual input unit at target position $\mathbf{x}$ is given by

$$v_j(\mathbf{x}) = a_0 \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)' \cdot \boldsymbol{\Sigma}_j^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}_j)\right) \tag{17}$$

where $\boldsymbol{\mu}_j$ is the center of this unit's receptive field, a_0 is a normalization that was set such that the population response at each target would sum to 1, and $\boldsymbol{\Sigma}_j$ is a diagonal matrix of variances, $\boldsymbol{\sigma}_j^2$. While in reality there may be neurons with non-spherical receptive fields, we started with this assumption to see what minimal structure is necessary to account for the diversity of M1 responses. In such an array target position is encoded by neuron identity, and each target results in a population response that is also a Gaussian bump. Since visual acuity is much higher than motor precision, we simulated 1,000,000 visual input units (which tiled space 100 x 100 x 100) to avoid any discretization artifacts. We had their centers, $\boldsymbol{\mu}_j$, extend 3 times past the furthest target positions used in our task (in both directions and along each spatial dimension), to avoid boundary effects.

Intended forearm rotation angles θ were modeled as an additional set of input units, $g_k(\theta)$, each randomly preferring pronation or supination and each with a uniformly random slope, such that for a pronation preferring neuron for example

$$g_k(\theta) \sim \begin{cases} U[1, 2], & \text{if } \theta = \text{pronation} \\ U[0, 1], & \text{if } \theta = \text{supination} \end{cases} \tag{18}$$

where U denotes a uniform distribution over the indicated range. The distributions are reversed for a supination preferring neuron. Since we only sampled forearm rotation at two points, we did not make any assumptions about these neurons' tuning for intermediate values of θ (although we depicted them graphically as lines in Fig 6A). Alternatively, we modeled intended forearm rotation angle inputs by a 1D array of Gaussians as well, which produced similar results. The presynaptic inputs into M1, f_n^P, were modeled as (premotor or parietal)

neurons that multiplicatively mixed the two input streams.

$$f_n^P(\mathbf{x}, \theta) = \nu_j(\mathbf{x})g_k(\theta) \tag{19}$$

This construction assures that the neurons exhibit the tuning that depends on both visual targets and arm posture found in many premotor [13, 52–56], and posterior parietal areas [57–62].

Finally, our model M1 neurons were modeled as each receiving a random combination of 10,000 such inputs mixed through random connectivity, and passed through a threshold-linear nonlinearity

$$f_i^{M1}(\mathbf{x}, \theta) = \left[\sum_n J_{in} f_n^P(\mathbf{x}, \theta) - \phi_i\right]_+ \tag{20}$$

where J_{in} is the random connection between the n-th presynaptic neuron and the i-th M1 neuron, ϕ_i is its threshold, and the square brackets denote the threshold-linear function. Each element J_{in} is chosen independently and randomly from a uniform distribution between 0 and 1. Because the model tries to capture tuning curve shape and not absolute firing rates, only the shape of the distribution of the random connections described by J matters, not their magnitude.

In the first model we used (Fig 6C and 6D, S5 Fig), the input tuning widths and threshold were all set the same values, so the model depends on only 2 parameters. The threshold ϕ was set to insure that the coding level of the model matched that of the real neurons (0.85). The coding level is defined as the fraction of conditions that causes a neuron to respond, which we took as being either significantly different than 0 spikes/s ($p < 0.01$, t-test, Bonferonni corrected), or at least 5 spikes/s. The visual Gaussian width σ was chosen (visual angle of 12°) as to produce response functions with a mean complexity measure that fit the value for the real data.

In the enhanced version of our model (Fig 6C–6J), the threshold for each model neuron was chosen independently so that its responses matched the coding level of a randomly assigned real neuron. In addition, the visual tuning-curve widths were chosen randomly independently for each input neuron from a uniform distribution with a mean σ_M with a range σ_R

$$\sigma \sim U\left[\sigma_M - \frac{\sigma_R}{2}, \, \sigma_M + \frac{\sigma_R}{2}\right] \tag{21}$$

σ_M and σ_R were fit by grid search ($\sigma_M \in [20,120]$, $\sigma_R \in [0,200]$) and nonlinear optimization over the multi-objective optimization problem of trying to match the real neurons' mean complexity measure and the mean (0.5) and standard deviation (0.22) of their R^2 distribution for the linear tuning curve. There is a continuous range of values for which this version of the model is near optimal, and we simply chose one set of values for producing Fig 6C–6J.

In order to use the model neurons to decode the EMG, we followed the same decoding scheme used for the real neuronal responses, as described above. Since the model generates mean responses, to generate single-trials we added noise as follows. First, we created *noise samples* by subtracting randomly chosen pairs of single-trial firing rates, for each condition, from each real neuron:

$$\eta_i(c) = r_i(k, c) - r_i(l, c) \quad , k \neq l \tag{22}$$

This is similar to the noise traces defined above but without the normalization (Eq 6). These noise samples were resampled from the data to match the number of trials per condition of the real dataset. Next, we created a lookup table of these noise samples across all the real neurons, where each row contains all the noise traces associated with the original mean firing rate, $\bar{r}_i(c)$, from which it was generated. Finally, single-trials firing rates were generated for each model

neuron by adding noise samples extracted from the lookup table, by choosing the firing rate closest to the model neuron's mean firing rate, at that condition. We chose to assign noise samples from the real to model neurons by using the mean rates as the lookup in order to capture the signal dependence of neuronal variability.

Supporting Information

S1 Fig. A. Chronic electrode array after insertion (for monkey BR; same location was used for monkey PK). C.S. = Central sulcus, A.S. = Arcuate sulcus. **B.** Mean ± S.D. of hand speed during the reaching movements to the target and the target-hold epochs, for each forearm posture. The 200 ms window analyzed in this study is denoted in red. Time is relative to task events; MB = Movement Beginning, and FC = Forearm rotation angle Cue (first sensory cue of next trial). Wide gray bar overlays the period where trials had random length intervals. Note that the hand speed is extremely low during target hold, yet not identically zero, because the hand was held freely in space. The larger variance before movement beginning is during the reaction time, and the larger variance during movement is due to averaging over reaches of different amplitudes. **C.** Mean ± S.D. of maximal hand jitter speed per trial, as a function of target position, and for each forearm posture. The hand jitter was uniform across targets (Multiple comparisons, using ANOVA, $p > 0.01$). **D.** Distribution of R^2 values for fitting single trial firing rate fluctuations to hand jitter using the velocity tuning-curve, median = 0.12. Values with a significant fit (F-Test, $p < 0.01$) are in dark gray, while the rest are in light gray. Fitting to all tuned neurons ($N = 411$) and the optimal R^2 for each neuron across all leads/lags (of firing rate to behavior) and forearm angles is presented. **E.** Same as D for control that shuffled single-trial firing rate fluctuations relative to their respective jitter movements, median = 0.11.
(TIF)

S2 Fig. A. Distributions of R^2 values for fit of linear tuning-curve for all tuned neurons (same as Fig 3B), separately from each monkey (means = 0.54 and 0.5, for monkeys BR and PK, respectively). **B.** Distributions of *complexity measure* of the (normalized) response functions (same as Fig 3D), separately for each monkey (means = 0.5 and 0.49, for monkeys BR and PK, respectively).
(TIF)

S3 Fig. The first 20 PCs of the 54-condition PCA (Fig 3E), in the format of the neuronal responses (Fig 2).
(TIF)

S4 Fig. A. Percent of neurons selected for decoding each muscle, with distributions over cross-validation repetitions. Medians = 49%, 43%, 43%, 48%, 44%, for the forearm extensor, flexor, biceps, triceps, and deltoid, respectively (grand total median = 44%). **B.** Same as A., for percent of neurons selected for decoding any single muscle, any pair of muscles, etc., in each cross-validation repetition. Medians = 30% selected for any single, 27% for any pair, 24% for triples, 15% for quadruples, and 5% of neurons selected for decoding all 5 muscles. **C.** Matrix of mean correlations between decoders for each pair of muscles (only significant correlation coefficients were used, $p < 0.01$, Bonferroni corrected), averaged over cross-validation repetitions. This matrix can be viewed as an effective connectivity of the output projections of M1 to the muscles, during arm posture control. **D.** Illustration of decomposition of single trials into spatial linear and nonlinear components. *Left*, an example response function in blue in the 2D format (same as Fig 2C) but with mean firing rates drawn as residuals of spatial linear tuning-curve. Orange dots are randomly selected single-trials. *Top left*, magnifies one condition (*c*) showing the decomposition of a single trial (*t*) into the sum of: (i) mean spatial linear component, (ii)

mean nonlinear component, (iii) single-trial noise fluctuation. *Top right*, example of the resulting purely nonlinear component with single-trial noise; note that there is no remaining linear component. *Bottom left*, same for the resulting purely spatial-linear component with single-trial noise.
(TIF)

S5 Fig. 4 example response functions generated by the 2-parameter version of the neural network model following the same format as Fig 2. Examples were chosen to highlight the spatial-linearity and various forms of nonlinearity seen in Fig 2.
(TIF)

S1 Text. Tuning is not the result of hand jitter.
(PDF)

S2 Text. Subpopulations for decoding EMG.
(PDF)

Acknowledgments

Author Contributions

Conceived and designed the experiments: HL EV. Performed the experiments: HL EV. Analyzed the data: HL LFA. Wrote the paper: HL LFA.

References

1. Georgopoulos AP, Caminiti R, Kalaska JF. Static spatial effects in motor cortex and area 5: quantitative relations in a two-dimensional space. Experimental Brain Research. 1984; 54(3):446–54. PMID: 6723864

2. Kettner RE, Schwartz AB, Georgopoulos AP. 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. JNeurosci. 1988; 8(8):2938–47.

3. Wang W, Chan SS, Heldman DA, Moran DW. Motor Cortical Representation of Position and Velocity During Reaching. Journal of Neurophysiology. 2007; 97(6):4258–70. doi: 10.1152/jn.01180.2006 PMID: 17392416

4. Paz R, Wise SP, Vaadia E. Viewing and doing: similar cortical mechanisms for perceptual and motor learning. Trends in Neural Science. 2004; 27(8):496–503.

5. Poggio T, Bizzi E. Generalization in vision and motor control. Nature. 2004; 431(7010):768–74. PMID: 15483597

6. Machens CK, Romo R, Brody CD. Functional, But Not Anatomical, Separation of "What" and "When" in Prefrontal Cortex. The Journal of Neuroscience. 2010; 30(1):350–60. doi: 10.1523/jneurosci.3276-09. 2010 PMID: 20053916

7. Friedman J, Hastie T, Tibshirani R. The Elements of Statistical Learning: Data Mining, Inference, and Prediction2009.

8. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological). 1996:267–88.

9. Gnadt JW, Breznen B. Statistical analysis of the information content in the activity of cortical neurons. Vision research. 1996; 36(21):3525–37. PMID: 8977019

10. Hamed SB, Duhamel JR, Bremmer F, Graf W. Representation of the visual field in the lateral intraparietal area of macaque monkeys: a quantitative receptive field analysis. Experimental Brain Research. 2001; 140(2):127–44. PMID: 11521146

11. Gnadt JW, Mays LE. Neurons in monkey parietal area LIP are tuned for eye-movement parameters in three-dimensional space. Journal of neurophysiology. 1995; 73(1):280–97. PMID: 7714572

12. Andersen R, Essick G, Siegel R. Encoding of spatial location by posterior parietal neurons. Science. 1985; 230(4724):456–8. doi: 10.1126/science.4048942 PMID: 4048942

13. Graziano MSA, Yap GS, Gross CG. Coding of visual space by premotor neurons. Science. 1994; 266 (5187):1054–7. PMID: 7973661

14. Pouget A, Sejnowski TJ. Spatial Transformations in the Parietal Cortex Using Basis Functions. Journal of Cognitive Neuroscience. 1997; 9(2):222–37. doi: 10.1162/jocn.1997.9.2.222 PMID: 23962013

15. Kakei S, Hoffman DS, Strick PL. Muscle and movement representations in the primary motor cortex. Science. 1999; 285(5436):2136–9. PMID: 10497133

16. Georgopoulos AP, Ashe J. One motor cortex, two different views. NatNeurosci. 2000; 3(10):963–5.

17. Scott SH. Reply to 'One motor cortex, two different views'. NatNeurosci. 2000; 3(10):964–5.

18. Sergio LE, Hamel-Paquet C, Kalaska JF. Motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks. The Journal of Neurophysiology. 2005; 94(4):2353–78. PMID: 15888522

19. Churchland MM, Shenoy KV. Temporal Complexity and Heterogeneity of Single-Neuron Activity in Premotor and Motor Cortex. Journal of Neurophysiology. 2007; 97(6):4235–57. doi: 10.1152/jn.00095. 2007 PMID: 17376854

20. Fetz EE. Are movement parameters recognizably coded in the activity of single neurons? Behavioral and Brain Sciences. 1992; 15(Special Issue 04):679–90. doi: 10.1017/S0140525X00072599

21. Wu W, Hatsopoulos N. Evidence against a single coordinate system representation in the motor cortex. Experimental Brain Research. 2006; 175(2):197–210. doi: 10.1007/s00221-006-0556-x PMID: 16775704

22. Aflalo TN, Graziano MSA. Relationship between Unconstrained Arm Movements and Single-Neuron Firing in the Macaque Motor Cortex. The Journal of Neuroscience. 2007; 27(11):2760–80. doi: 10. 1523/jneurosci.3147-06.2007 PMID: 17360898

23. Morrow MM, Jordan LR, Miller LE. Direct Comparison of the Task-Dependent Discharge of M1 in Hand Space and Muscle Space. Journal of Neurophysiology. 2007; 97(2):1786–98. doi: 10.1152/jn.00150. 2006 PMID: 17122326

24. Pagan M, Simoncelli EP, Rust NC. A generalization of the Energy model explains the transformation from IT to Perirhinal cortex. Journal of Vision. 2014; 14(15):78–.

25. Lukashin AV. A learned neural network that simulates properties of the neuronal population vector. Biological Cybernetics. 1990; 63(5):377–82. doi: 10.1007/bf00202754

26. Lau B, Stanley GB, Dan Y. Computational subunits of visual cortical neurons revealed by artificial neural networks. Proceedings of the National Academy of Sciences. 2002; 99(13):8974–9.

27. Sanger TD. Theoretical Considerations for the Analysis of Population Coding in Motor Cortex. Neural Comput. 1994; 6:12–21.

28. Mussa-Ivaldi FA. Do neurons in the motor cortex encode movement direction? An alternative hypothesis. NeurosciLett. 1988; 91(1):106–11.

29. Babadi B, Sompolinsky H. Sparseness and Expansion in Sensory Representations. Neuron. 2014; 83 (5):1213–26. doi: http://dx.doi.org/10.1016/j.neuron.2014.07.035 PMID: 25155954

30. Caron SJ, Ruta V, Abbott L, Axel R. Random convergence of olfactory inputs in the Drosophila mushroom body. Nature. 2013; 497(7447):113–7. doi: 10.1038/nature12063 PMID: 23615618

31. Sussillo D, Abbott LF. Generating Coherent Patterns of Activity from Chaotic Neural Networks. Neuron. 2009; 63(4):544–57. doi: 10.1016/j.neuron.2009.07.018 PMID: 19709635

32. Barak O, Rigotti M, Fusi S. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. The Journal of Neuroscience. 2013; 33(9):3844–56. doi: 10.1523/ JNEUROSCI.2753-12.2013 PMID: 23447596

33. Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, et al. Neural population dynamics during reaching. Nature. 2012;advance online publication. http://www.nature.com/ nature/journal/vaop/ncurrent/abs/nature11129.html—supplementary-information.

34. Rokni U, Sompolinsky H. How the Brain Generates Movement. Neural Computation. 2011; 24(2):289–331. doi: 10.1162/NECO_a_00223 PMID: 22023199

35. Sussillo D, Churchland MM, Kaufman MT, Shenoy KV. A neural network that finds a naturalistic solution for the production of muscle activity. Nature neuroscience. 2015; 18(7):1025–33. doi: 10.1038/nn. 4042 PMID: 26075643

36.	Todorov E. Direct cortical control of muscle activation in voluntary arm movements: a model. NatNeurosci. 2000; 3(4):391–8.

37.	Trainin E, Meir R, Karniel A. Explaining Patterns of Neural Activity in the Primary Motor Cortex Using Spinal Cord and Limb Biomechanics Models. Journal of Neurophysiology. 2007; 97(5):3736–50. doi: 10.1152/jn.01064.2006 PMID: 17360816

38.	Lillicrap TP, Scott SH. Preference Distributions of Primary Motor Cortex Neurons Reflect Control Solutions Optimized for Limb Biomechanics. Neuron. 2013; 77(1):168–79. doi: http://dx.doi.org/10.1016/j.neuron.2012.10.041 PMID: 23312524

39.	Guigon E, Baraduc P, Desmurget M. Coding of movement- and force-related information in primate primary motor cortex: a computational approach. European Journal of Neuroscience. 2007; 26(1):250–60. doi: 10.1111/j.1460-9568.2007.05634.x PMID: 17573920

40.	Bullock D, Cisek P, Grossberg S. Cortical networks for control of voluntary arm movements under variable force conditions. CerebCortex. 1998; 8(1):48–62.

41.	Kawato M, Wolpert D. Internal models for motor control. NovartisFoundSymp. 1998; 218:291–304.

42.	Jordan MI. Computational motor control. In: Gazzaniga MS, editor. The cognitive neurosciences. Cambridge, MA: MIT Press; 1995. p. 597–609.

43.	Kalaska JF, Crammond DJ. Cerebral Cortical Mechanisms of Reaching Movements. Science. 1992; 255:1517–23. PMID: 1549781

44.	Shadmehr R, Wise SP. The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning. Cambridge: MIT Press 2004.

45.	Lalazar H, Vaadia E. Neural basis of sensorimotor learning: modifying internal models. Current Opinion in Neurobiology. 2008; 18(6):573–81. doi: 10.1016/j.conb.2008.11.003 PMID: 19054663

46.	Schwartz AB, Kettner RE, Georgopoulos AP. Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. relations between single cell discharge and direction of movement. JNeurosci. 1988; 8(8):2913–27.

47.	Caminiti R, Johnson PB, Urbano A. Making Arm Movements Within Different Parts of Space: Dynamic Aspects in the Primate Motor Cortex. JNeurosci. 1990; 10:2039–58.

48.	Moran DW, Schwartz AB. Motor cortical representation of speed and direction during reaching. The Journal of Neurophysiology. 1999; 82(5):2676–92. PMID: 10561437

49.	Paninski L, Shoham S, Fellows MR, Hatsopoulos NG, Donoghue JP. Superlinear population encoding of dynamic hand trajectory in primary motor cortex. JNeurosci. 2004; 24(39):8551–61.

50.	Hatsopoulos NG, Xu Q, Amit Y. Encoding of Movement Fragments in the Motor Cortex. The Journal of Neuroscience. 2007; 27(19):5105–14. doi: 10.1523/jneurosci.3570-06.2007 PMID: 17494696

51.	Rigotti M, Barak O, Warden MR, Wang X-J, Daw ND, Miller EK, et al. The importance of mixed selectivity in complex cognitive tasks. Nature. 2013;advance online publication. doi: 10.1038/nature12160 http://www.nature.com/nature/journal/vaop/ncurrent/abs/nature12160.html—supplementary-information.

52.	Graziano MSA, Hu XT, Gross CG. Visuospatial properties of ventral premotor cortex. Journal of Neurophysiology. 1997; 77(5):2268–92. PMID: 9163357

53.	Pesaran B, Nelson MJ, Andersen RA. Dorsal premotor neurons encode the relative position of the hand, eye, and goal during reach planning. Neuron. 2006; 51(1):125–34. PMID: 16815337

54.	Batista AP, Santhanam G, Byron MY, Ryu SI, Afshar A, Shenoy KV. Reference frames for reach planning in macaque dorsal premotor cortex. Journal of neurophysiology. 2007; 98(2):966–83. PMID: 17581846

55.	Hoshi E, Tanji J. Integration of target and body-part information in the premotor cortex when planning action. Nature. 2000; 408(6811):466–70. PMID: 11100727

56.	Fogassi L, Gallese Vl, Di Pellegrino G, Fadiga L, Gentilucci M, Luppino G, et al. Space coding by premotor cortex. Experimental Brain Research. 1992; 89(3):686–90. PMID: 1644132

57.	Chang SWC, Dickinson AR, Snyder LH. Limb-specific representation for reaching in the posterior parietal cortex. The Journal of Neuroscience. 2008; 28(24):6128–40. doi: 10.1523/JNEUROSCI.1442-08.2008 PMID: 18550755

58.	Andersen RA, Snyder LH, Bradley DC, Xing J. Multimodal representation of space in the posterior parietal cortex and its use in planning movements. AnnuRevNeurosci. 1997; 20:303–30.

59.	Galletti C, Fattori P, Kutz DF, Battaglini PP. Arm Movement-related Neurons in the Visual Area V6A of the Macaque Superior Parietal Lobule. European Journal of Neuroscience. 1997; 9(2):410–3. PMID: 9058060

60. Chang SWC, Snyder LH. Idiosyncratic and systematic aspects of spatial representations in the macaque parietal cortex. Proceedings of the National Academy of Sciences. 2010; 107(17):7951–6. doi: 10.1073/pnas.0913209107

61. Lacquaniti F, Guigon E, Bianchi L, Ferraina S, Caminiti R. Representing spatial information for limb movement: role of area 5 in the monkey. Cerebral Cortex. 1995; 5(5):391–409. PMID: 8547787

62. Buneo CA, Andersen RA. The posterior parietal cortex: sensorimotor interface for the planning and online control of visually guided movements. Neuropsychologia. 2006; 44(13):2594–606. PMID: 16300804

63. Deneve S, Latham PE, Pouget A. Efficient computation and cue integration with noisy population codes. Nature neuroscience. 2001; 4(8):826–31. PMID: 11477429

64. McGuire LMM, Sabes PN. Sensory transformations and the use of multiple reference frames for reach planning. Nature neuroscience. 2009; 12(8):1056–61. doi: 10.1038/nn.2357 PMID: 19597495

65. Poggio T, Girosi F. Theory of networks for learning. Science. 1990; 247:978–82. PMID: 17776454

66. Lowe D, Broomhead D. Multivariable functional interpolation and adaptive networks. Complex systems. 1988; 2:321–55.

67. Moody J, Darken CJ. Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computation. 1989; 1(2):281–94. doi: 10.1162/neco.1989.1.2.281

68. Shalit U, Zinger N, Joshua M, Prut Y. Descending Systems Translate Transient Cortical Commands into a Sustained Muscle Activation Signal. Cerebral Cortex. 2011. doi: 10.1093/cercor/bhr267

69. Shpigelman L, Lalazar H, Vaadia E. Kernel-ARMA for Hand Tracking and Brain-Machine Interfacing During 3D Motor Control. 2009.

70. Pruszynski JA, Kurtzer I, Nashed JY, Omrani M, Brouwer B, Scott SH. Primary motor cortex underlies multi-joint integration for fast feedback control. Nature. 2011; 478(7369):387–90. http://www.nature.com/nature/journal/v478/n7369/abs/nature10436.html—supplementary-information. doi: 10.1038/nature10436 PMID: 21964335

71. Morrow MM, Miller LE. Prediction of muscle activity by populations of sequentially recorded primary motor cortex neurons. Journal of neurophysiology. 2003; 89(4):2279–88. PMID: 12612022

72. Pohlmeyer EA, Solla SA, Perreault EJ, Miller LE. Prediction of upper limb muscle activity from motor cortical discharge during reaching. Journal of Neural Engineering. 2007; 4(4):369. PMID: 18057504

73. Cherian A, Krucoff MO, Miller LE. Motor cortical prediction of EMG: evidence that a kinetic brain-machine interface may be robust across altered movement dynamics. Journal of Neurophysiology. 2011; 106(2):564–75. doi: 10.1152/jn.00553.2010. PMID: 21562185

74. Rathelot J-A, Strick PL. Muscle representation in the macaque motor cortex: An anatomical perspective. Proceedings of the National Academy of Sciences. 2006; 103(21):8257–62. doi: 10.1073/pnas.0602933103

75. Bizzi E, Kalil RE, Tagliasco V. Eye-Head Coordination in Monkeys: Evidence for Centrally Patterned Organization. Science. 1971; 173(3995):452–4. doi: 10.1126/science.173.3995.452 PMID: 17770450

76. Bock O. Localization of objects in the peripheral visual field. Behavioural brain research. 1993; 56 (1):77–84. PMID: 8397856

77. Suner S, Fellows MR, Vargas-Irwin C, Nakata GK, Donoghue JP. Reliability of signals from a chronically implanted, silicon-based electrode array in non-human primate primary motor cortex. Neural Systems and Rehabilitation Engineering, IEEE Transactions on. 2005; 13(4):524–41. doi: 10.1109/tnsre.2005.857687 PMID: 16425835

78. Cardoso dO, Gribova A, Donchin O, Bergman H, Vaadia E. Neural interactions between motor cortical hemispheres during bimanual and unimanual arm movements. European Journal of Neuroscience. 2001; 14(11):1881–96. PMID: 11860483

79. Pearson K. On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6. 1901; 2(11):559–72. doi: 10.1080/14786440109462720

80. Everitt B, Skrondal A. The Cambridge dictionary of statistics. 4th ed2010.