nature neuroscience

A neural network that finds a naturalistic solution for the production of muscle activity

David Sussillo[1], Mark M Churchland[2], Matthew T Kaufman[1,4] & Krishna V Shenoy[1,3]

It remains an open question how neural responses in motor cortex relate to movement. We explored the hypothesis that motor cortex reflects dynamics appropriate for generating temporally patterned outgoing commands. To formalize this hypothesis, we trained recurrent neural networks to reproduce the muscle activity of reaching monkeys. Models had to infer dynamics that could transform simple inputs into temporally and spatially complex patterns of muscle activity. Analysis of trained models revealed that the natural dynamical solution was a low-dimensional oscillator that generated the necessary multiphasic commands. This solution closely resembled, at both the single-neuron and population levels, what was observed in neural recordings from the same monkeys. Notably, data and simulations agreed only when models were optimized to find simple solutions. An appealing interpretation is that the empirically observed dynamics of motor cortex may reflect a simple solution to the problem of generating temporally patterned descending commands.

Considerable controversy has centered on whether neural responses in motor cortex encode high-level parameters, such as reach direction, or low-level parameters, such as force or muscle activity[1–11]. An equally fundamental question remains largely unaddressed: how are those temporally complex responses generated. To execute a movement, such as a reach, there must presumably exist some pattern generator that receives the relevant parameters and produces the necessary output. Pattern generation might occur entirely upstream of motor cortex, with motor cortex representing and conveying the generated commands. In this case, motor cortex responses would be expected to resemble muscle responses. A second possibility is that pattern generation may occur downstream, such that motor cortex parameterizes a high-level command. This possibility is suggested by the decoding of high-level features from the population response[9,12,13]. A final possibility is that motor cortex is a key participant in generating outgoing commands[14–16]. This possibility, as with the first possibility, predicts a close relationship between neural and muscle activity[4,16–20]. However, it also predicts there will be additional response features that are signatures of pattern generation. Thus, many aspects of the neural response may be quite 'non-muscle like' even if muscle commands are the final output[21].

We recently reported[22,23] that the motor cortex population state exhibits quasi-oscillatory features that provide a potential basis set for outgoing muscle-like commands. A simple linear model of the underlying dynamics captured much of the response structure. These results are consistent with the third possibility described above. Yet the theoretical foundation for these observations remains unclear. Why do quasi-oscillatory dynamics dominate when many other solutions are presumably possible? We explored the hypothesis that the observed dynamics are a consequence of generating descending motor commands in as simple a fashion as possible. We optimized a family of recurrent neural networks (RNNs[24]) to generate the electromyographic (EMG) signals recorded from multiple muscles during the experiments described in ref. 23. We parameterized the family of RNNs by the complexity of allowable dynamics, from very simple to extremely complex.

Notably, RNNs were not trained to reproduce the empirical neural responses, only to reproduce our proxy for the descending motor commands, the recorded EMG. Beyond parameterizing the complexity of the RNN dynamics, we deliberately avoided imposing additional constraints. We did not constrain connectivity or attempt to impose structure based on known features of cortical connectivity. This allowed the RNNs to seek an optimum over a very broad range of dynamics, unconstrained by prior knowledge. Nevertheless, we found that the dynamics learned by the models resembled the dynamics seen in motor cortex. This was true both qualitatively and quantitatively and at both the single-neuron and population levels. However, the similarity between data and model was strong only if the RNN was heavily 'regularized' to encourage extremely simple solutions. This finding suggests that cortex displays the empirically observed dynamics because those dynamics provide a simple solution to the problem of generating temporally structured outputs.

RESULTS

Task

Two monkeys (J and N) performed a delayed reach task (**Fig. 1a**)[23]. To begin each trial, the monkey fixated and touched a central target. A maze configuration and target(s) then appeared, but the monkey

[1]Department of Electrical Engineering and Neurosciences Program, Stanford University, Stanford, California, USA. [2]Department of Neuroscience, Grossman Center for the Statistics of Mind, David Mahoney Center for Brain and Behavior Research, Kavli Institute for Brain Science, Columbia University Medical Center, New York, New York, USA. [3]Departments of Bioengineering and Neurobiology, Stanford Neurosciences Institute and Bio-X Program, Stanford University, Stanford, California, USA. [4]Present address: Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA. Correspondence should be addressed to D.S. (sussillo@stanford.edu).

Received 22 December 2014; accepted 15 May 2015; published online 15 June 2015; doi:10.1038/nn.4042

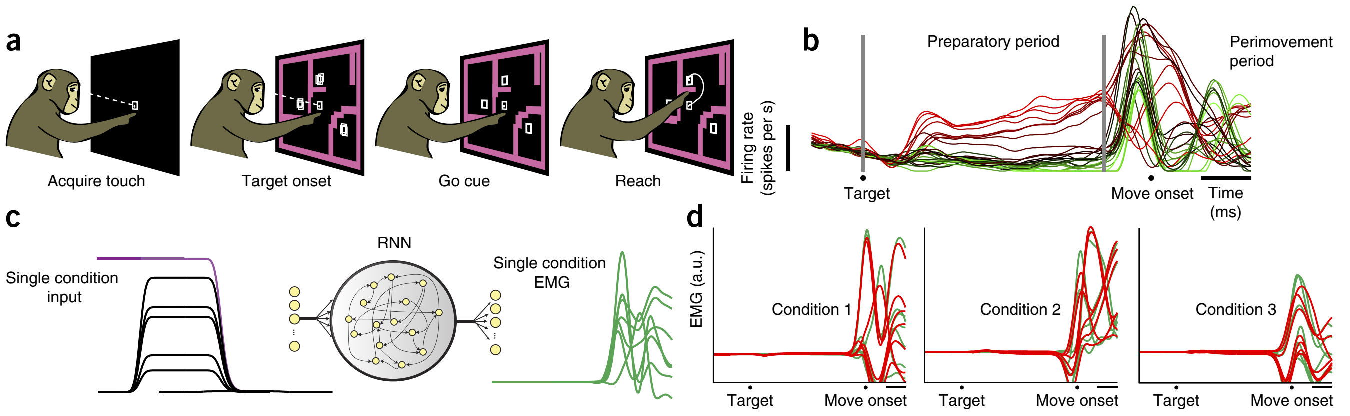


Figure 1 Monkey task and network task definition. (**a**) Monkeys performed a delayed reach maze task. After fixating and touching a central point, the target and maze turned on. Some conditions included distractor targets. During the preparatory period, the monkeys had to determine which target was reachable and prepare a reach that avoided any intervening barriers. A go cue prompted the monkey to execute the reach. We employed 27 conditions, each consisting of a particular configuration of target and barriers. The resulting reaches included a variety of straight and curved paths. (**b**) Example PSTH for a single neuron. Each trace plots the mean across-trial firing rate for one condition (27 total). Traces are colored green to red based on the level of preparatory activity. The first gray line shows the timing of target onset, that is the beginning of the preparatory period. The second gray line shows the end of the preparatory period. Vertical and horizontal scale bars indicate 20 spikes per s and 200 ms. (**c**) Networks were optimized to generate EMG. Network inputs consisted of a condition-independent hold cue (purple) and a six-dimensional condition-specific input (black), which specified the condition for which the network should generate EMG. This example shows the levels of those six inputs for condition 1. From these inputs the RNN generated the multi-dimensional EMG: green traces plot the recorded EMG from seven muscles for condition 1. To ensure the model fit signal and not noise, we filtered EMG signals and removed the (very minimal) noise during the baseline (Online Methods). (**d**) Three example conditions showing the multiple muscle target EMG (green, one trace per muscle) and the corresponding trained outputs of the regularized model for monkey J (red). Normalized error between the empirical EMG and the model output was 7%. Horizontal scale bars indicate 200 ms.

was required to withhold his reach until a 'go cue' appeared. Each maze and target configuration enforced a particular reach trajectory. We analyzed 27 such configurations, termed conditions. We defined the preparatory period as the interval from maze onset until 150 ms after the go cue, the moment at which neural activity begins to change rapidly just before reach onset. During the preparatory period, the monkey had complete information regarding the reach to be performed, but had not yet begun to move.

Responses of neural populations were recorded from primary motor cortex (M1) and the adjacent region of dorsal premotor cortex (PMd). Neurons typically displayed different levels of preparatory activity depending on the upcoming movement[25,26] (**Fig. 1b**). Approximately 150 ms before movement onset, the relatively stable plateau of preparatory activity transitioned to a complex pattern of movement-related activity. Muscle activity, recorded from the principal muscles of the upper arm, changed little during the preparatory period, but exhibited temporally complex patterns just before and during the movement.

We previously proposed that a purpose of preparatory neural activity is to initialize a dynamical system whose subsequent evolution during movement generates descending muscle-like commands[10,23] (**Fig. 1b**). If so, what is the nature of those dynamics? We examined solutions naturally found by recurrent neural networks. The resulting trained networks yielded a set of simple, but empirically constrained, hypotheses whose predictions could be compared against the experimentally observed patterns of neural activity. We stress that these are models of emergent dynamics, not of cortical architecture or implementation.

A simplified modeling framework for reach generation

Under natural circumstances, a stream of inputs guides reaching. These inputs include those that motivate and initiate the reach (for example, the sight of a desirable object and the decision to obtain it) and subsequent sensory feedback. We adopted a simplified set of just two inputs (**Fig. 1c**). We assumed that, during the preparatory period, cortex receives inputs specific to the reach being prepared. To avoid making assumptions about the reference frame of those inputs, we derived the static levels of the reach-specific inputs from the empirically recorded preparatory neural activity (Online Methods). We assumed that movement unfolds when a condition-independent 'hold' signal is released. The goal of the network was to utilize these temporally simple inputs (**Fig. 1c**) to produce, at the right moment, the temporally complex patterns of activity recorded across multiple muscles (**Fig. 1d**).

Ideally, we would have included a third input stage: the sensory feedback that arrives after the reach begins. We decided to not include this stage for two practical reasons. First, the structure of the feedback is difficult to estimate. Second, many of the features of the neural population response are apparent even before movement begins: the establishment of preparatory activity and its relationship to early movement-period activity unfold before feedback can have had an effect. Empirically, movement-period neural responses lead the motion of the hand by ~150 ms. Sensory feedback takes at least 25 ms to influence cortical responses and >50 ms to reflect the current goal[27]. Thus, during this ~200-ms interval, the neural dynamics are not yet affected by sensory feedback and should presumably be explained via internal dynamics. This is true even of optimal feedback control architectures, which employ a dynamically varying control policy and internal 'efference-copy' recurrence to generate time-varying output patterns before the arrival of feedback[28,29]. Given the practical choice to use a model without sensory feedback, we verified with additional simulations that the solutions found by the model were robust to the addition of reasonable forms of feedback (**Supplementary Fig. 1a,b**).

We used RNNs as a modeling tool for three reasons. First, an RNN can approximate any dynamical system[30]. Second, an RNN is an abstract model that is nevertheless inspired by biological neural circuits; the units are individually simple and must work together in a parallel and distributed fashion. Third, internal recurrent feedback, a defining aspect of RNNs, is essential for many forms of pattern

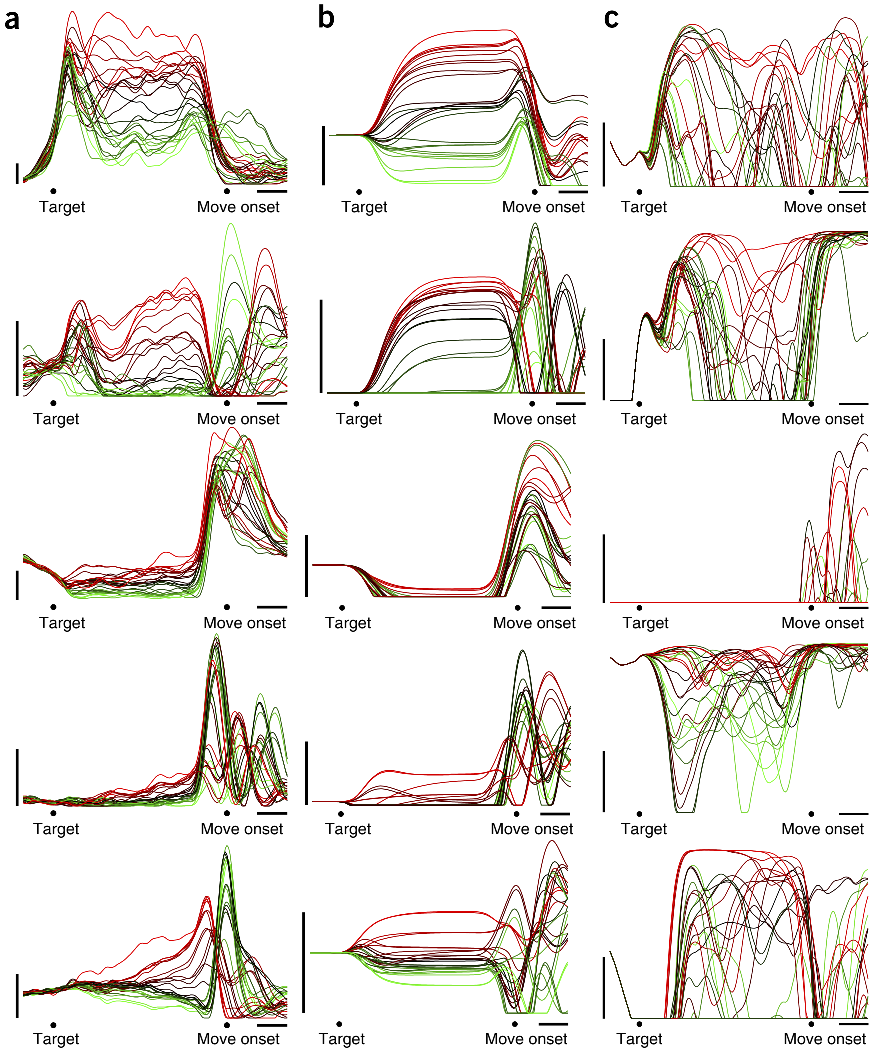


Figure 2 Example PSTHs from monkey J and the regularized and complicated models for monkey J. (**a**) Example PSTHs from five neurons for monkey J (data are presented as in **Fig. 1b**). Examples were chosen to illustrate the range of responses, including neurons with strong preparatory activity (first two rows), neurons with a broad rise in activation during the movement period (middle row) and neurons with oscillatory activity during the movement period (bottom two rows). Vertical and horizontal scale bars indicate 20 spikes per s and 200 ms. (**b**) Example PSTHs chosen from the regularized model for monkey J. Examples were chosen to both highlight the similarities between neural and model responses and to be representative of the patterns exhibited by the model units. (**c**) Example PSTHs from five units from the complicated model for monkey J. The PSTHs of the complicated models rarely bore a strong resemblance to those of the neural data.

generation. We produced two classes of trained networks: a regularized model and a complicated model. For the regularized model, we included regularization terms during optimization to encourage simple solutions. We included no regularization terms in the complicated model (**Supplementary Table 1**). For each monkey, we trained one network from each complexity class. All models successfully reproduced the recorded muscle activity for the 27 reaches (**Fig. 1d**). The normalized error was 7% for both models for monkey J and 3% for both models for monkey N.

Comparison of the model to data

Notably, networks were never trained to reproduce neural responses, only to generate the empirical EMG. This allowed us, after training, to compare network activity with recorded neural activity. To gain intuition, we first used the traditional single-neuron peristimulus time histogram (PSTH) format to qualitatively compare responses of single neurons and model units. We then used dimensionality reduction techniques to compare key features of the recorded and simulated population responses. Finally, we directly and quantitatively compared recorded and simulated population responses using canonical correlation analysis (CCA) (monkey J; **Figs. 2–8**; monkey N; **Supplementary Figs. 2–8**).

Single-unit PSTHs are shown for five neurons (**Fig. 2a**) and five units from the regularized model (**Fig. 2b**). Color-coding was based on the average preparatory period firing rate. We selected these PSTHs to illustrate a range of common patterns found in the neural and model populations. Such patterns included plateaus of preparatory activity and a variety of multi-phasic and monophasic movement-period responses. Representative PSTHs from the complicated model are shown in **Figure 2c**. The PSTHs of the complicated model are much more complex than, and bear little resemblance to, most neural responses. Thus, although both simple and complicated models generate a basis set of responses adequate to produce EMG, only the regularized model employs a basis set that qualitatively resembles the recorded neural responses.

Is the similarity at the single-neuron level also present at the population level? We first leveraged the recent observation[23] that projections of the neural data reveal population responses that follow roughly oscillatory dynamics (quasi-oscillatory dynamics). Because quasi-oscillatory dynamics have been robustly observed across many data sets, any hypothesis that does not predict such dynamics can be rejected. To compare data with model, we therefore applied a dimensionality reduction technique (jPCA[23], Online Methods) that isolates any, if present, quasi-oscillatory structure in the data.

Projections of the neural population responses revealed rotations of the neural state across multiple dimensions (**Fig. 3a**), consistent with quasi-oscillatory dynamics, as previously reported. Projections of the regularized model population response revealed similar rotations (**Fig. 3b**). These projections were obtained by fitting the population response with a purely oscillatory linear dynamical system. The goodness of fit (R^2) of those fits was similar for the neural and regularized model data: 0.60 and 0.61. Thus, ~60% of the temporal evolution of the population response could be explained by oscillatory dynamics. The frequencies found by jPCA were 2.1, 1.3 and 0.9 Hz (neural data), and 2.4, 1.6 and 0.9 Hz (regularized model). The total variance captured by the three jPC planes was 45% (neural) and 50% (model). Thus, roughly half of the structure of the data was captured by six dimensions (three planes), with oscillatory frequencies that were similar for neural and model data. We have previously

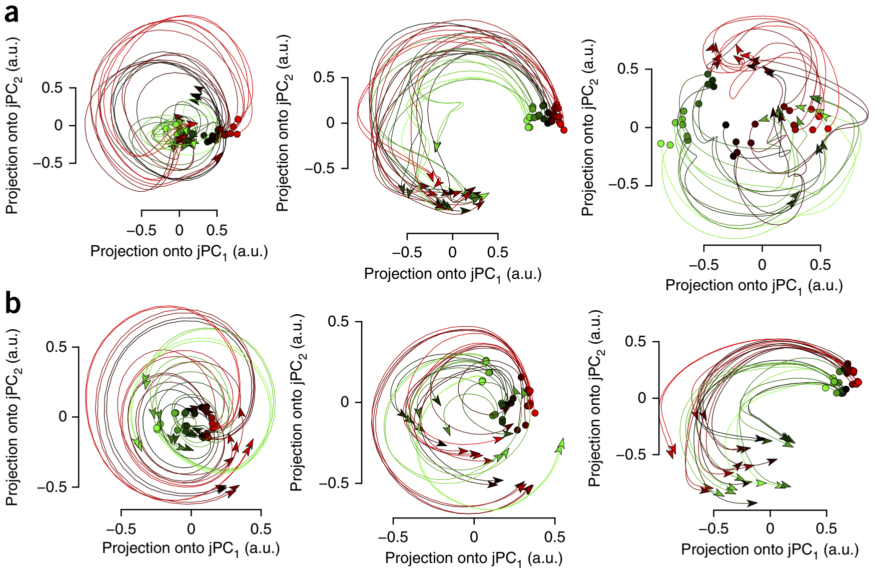


Figure 3 jPCA projections of the population responses. (**a**) jPCA projections for the neural data recorded from monkey J. Each trace shows the evolution of the neural state over 500 ms. Traces start −180 ms before movement onset, at the moment when the relatively stable preparatory state (circles) transitioned to the movement period trajectory. For visualization purposes, traces are colored on the basis of the preparatory-state projection onto jPC$_1$ (a.u., arbitrary units). The three projections correspond to the largest magnitude complex eigenvalue pairs of the matrix $\mathbf{M}_{\text{skew}}$, found when fitting the data with $\dot{\mathbf{x}} = \mathbf{M}_{\text{skew}}\mathbf{x}$ (Online Methods). These eigenvalues correspond to frequencies of 2.1, 1.3 and 0.9 Hz (left to right) with a quality of fit (R^2) for the optimal purely oscillatory linear system of 0.60. (**b**) jPCA projections for the regularized model of monkey J. Data are presented as in **a**. Frequencies are 2.4, 1.6 and 0.9 Hz. The linear system, $\dot{\mathbf{x}} = \mathbf{M}_{\text{skew}}\mathbf{x}$, had a quality of fit ($R^2$) of 0.61.

shown that standard models of motor cortex (for example, models assuming tuning for kinematics or muscle activity) do not display a strong rotational component[23]. It is therefore non-trivial that the network naturally produces strongly rotational dynamics with a set of frequencies similar to those observed in the data.

Rotational structure was also present for the complicated models, but was less strong overall (**Supplementary Fig. 9**). The R^2 of the best purely oscillatory linear system was 0.35, compared with 0.60 for the data and 0.61 for the regularized model. The rotational planes captured a reasonable proportion of data variance for the complicated model (41%), but the observed frequencies were roughly half what was found for the data: 1.3, 0.8 and 0.6 Hz. Thus, relative to the regularized model and the neural data, the dynamics of the complicated model were less well approximated by an oscillatory linear system. Notably, those oscillations that were present were considerably slower.

To directly compare neural and model populations and quantify their similarity, we applied canonical correlation analysis (CCA). Briefly, CCA attempts to find weightings for the individual units in both data sets such that the reweighted data sets are maximally correlated. In other words, CCA attempts to find the patterns common across two data sets. The reweighted data sets are called the canonical variables. The two sets of canonical variables are ordered by their degree of correlation, providing a series of correlation coefficients: the canonical correlations. If all canonical correlations are unity, then the two data sets are differently weighted versions of the same set of underlying patterns. If all canonical correlations are zero, then the two data sets have no underlying patterns in common. In practice, two data sets that share any broad similarity will typically have at least one or two canonical correlations that are high. The key question is across how many canonical variables do correlations remain high.

Figure 4a,**b** shows the canonical variables for the neural and regularized model. Each row captures a basic response pattern shared between neural and model populations. Each pattern is a response component, a firing rate versus time across conditions, present in the population. The canonical correlations give the correlation between the corresponding patterns. To illustrate the range of correlations, we plotted both the best and the most weakly correlated canonical variables. Neural and model patterns matched strongly among the top canonical variables, and matched modestly well even for the later canonical variables. Thus, the neural and regularized model data sets share many population-level patterns that unfold in a similar way across both time and condition. As expected, given the analysis in **Figure 3**, some of these patterns were oscillatory in nature, although some were not. Shown in **Figure 4c**,**d** are the canonical variables for the neural data and the complicated model. Correlations fall more quickly for the complicated model than for the regularized model. Thus, there are fewer matching patterns between the complicated model and the data than between the regularized model and the data.

We used the canonical coefficients to quantitatively compare the neural data with a variety of models: the regularized RNN (**Fig. 5a**), the complicated RNN, a traditional velocity model tuned for kinematic variables, such as velocity and position, and a more elaborate complex kinematic model. Given that some correlation is expected between almost any two data sets, we also analyzed an untrained complicated model (a random network that receives the correct inputs, but was not trained) as a baseline. For all models, there was at least one canonical variable with a very high correlation; all models shared basic temporal features with the data (for example, preparatory activity followed by movement activity). However, the canonical correlations remained higher for the regularized model than for any other model. To summarize, we computed the average canonical correlation across the first ten canonical variables (**Fig. 5b**). The average correlation was highest for the regularized model (0.74) and lower for the other models: 0.51, 0.49, 0.58 and 0.59. Thus, there were more shared patterns between the data and the regularized model than between the data and any of the other models.

How regularized dynamics produce EMG

Why does the regularized RNN most strongly resemble the data? What is the solution found by training? Do the essential features of that solution appear in the data? Because the parameters of the trained RNN are known, we can directly dissect its mechanism in the language of dynamics. The analyses described above suggested the following broad framework. First, preparatory inputs cause the network state to differ for each of the 27 reaches. The offset of the hold signal then 'turns on' strong dynamics with large oscillatory components. The resulting oscillatory neural trajectories successfully reproduce EMG when projected onto the output weights. Is this indeed what occurs? If so, how does the RNN achieve it?

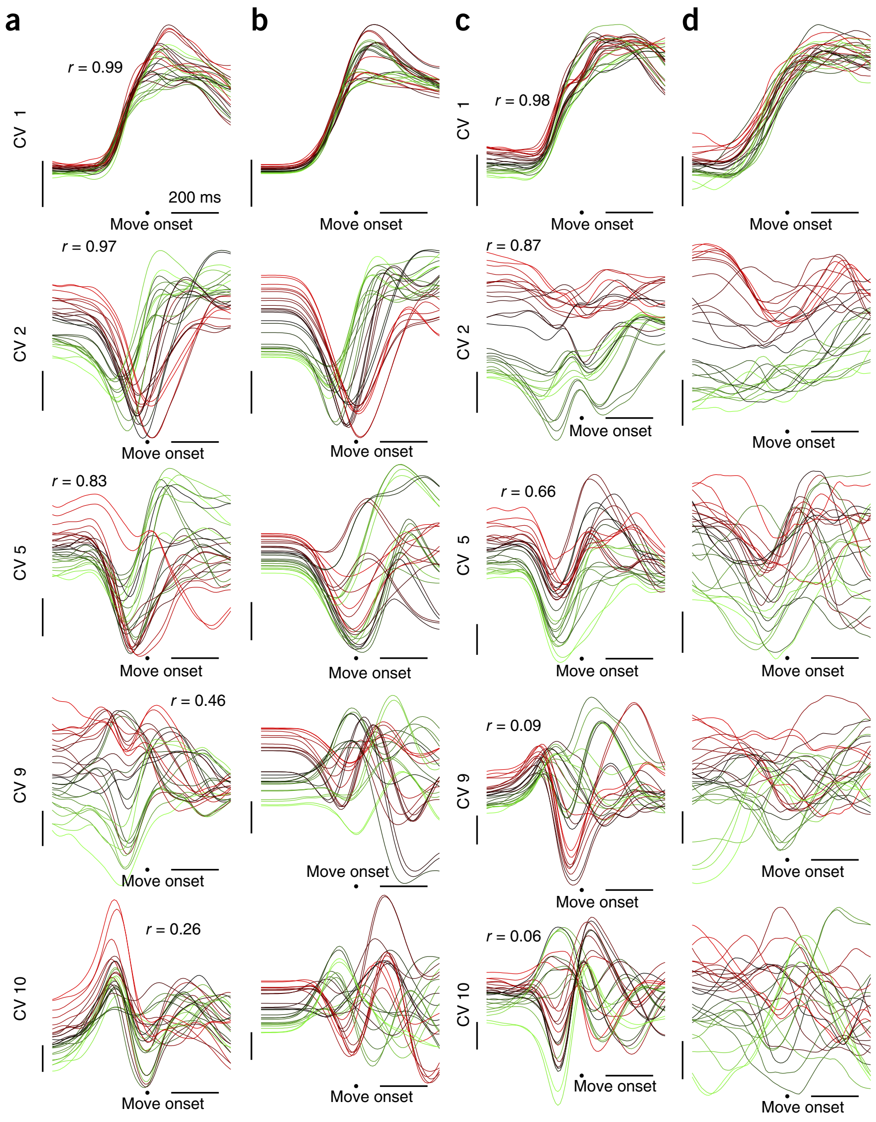


Figure 4 Canonical correlations analysis for monkey J. (**a**,**b**) CCA projections (canonical variables) of the neural population response (**a**) and the regularized model for monkey J (**b**). These projections involve the directions in state-space that maximally correlate the neural data with the model data, resulting in a series of maximally to minimally correlated variables. Each row shows one of the canonical variables (CVs) 1, 2, 5, 9 and 10, highlighting the most and least similar projections. The correlation *r* is also shown. Traces are colored on the basis of the value of the projection at the beginning of the trace. The vertical scale bars indicate 1 arbitrary unit and the horizontal scale bars represent 200 ms. (**c**,**d**) Canonical variables of the neural population response (**c**) and the complicated model for monkey J (**d**) (data are presented as in **a** and **b**).

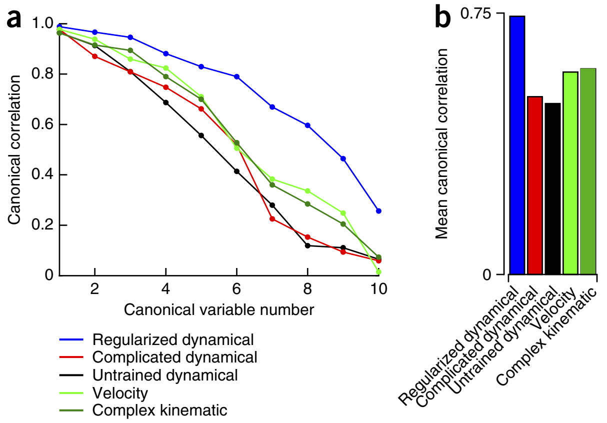
To understand how the RNN generates the EMG, we performed an additional step of reverse engineering[31]. This discovery phase employed standard procedures for analyzing nonlinear systems (for example, see ref. 32). How is preparatory activity transformed into a pattern of movement-period activity that produces the correct EMG output? For the regularized model, the underlying mechanism was surprisingly simple. The offset of the hold signal produced a single fixed point, and the dynamics around this fixed point governed the evolution of the neural state for all reach conditions.

A three-dimensional visualization of the RNN activity that highlights these dynamics is shown in **Figure 6b**, which plots the evolution of the network population in state space (Online Methods) for all 27 conditions. There exists a single fixed point that organizes oscillatory neural trajectories for all reach conditions. During preparation, the neural state is far from this fixed point. Just before the onset of EMG generation, there is a left-to-right translation of the neural state, for all conditions, toward the fixed point. The neural state then rotates around the fixed point in a consistent direction (some conditions rotate out of the page and some conditions rotate into the page). This rotation is similar for every condition, but with a different phase and amplitude. This is the same rotation that can be seen 'head on' (**Fig. 3b**). The neural population trajectories exhibited a notably similar structure (**Fig. 6a**). This simple pattern was specific to the neural data and the regularized model. The complicated RNNs did not show single fixed points, but did display a very large number of approximate fixed points, indicative of a highly nonlinear and complex mechanism for producing EMG. Thus, the solution found by the regularized model is not inevitable. There are many other dynamical solutions; they simply don't resemble the neural data as closely.

To directly characterize dynamics (**Fig. 7**), we analyzed the linear dynamics around the single fixed point in the regularized RNN[31] (**Fig. 7b**). Linearization revealed multiple modes in the eigenvalue spectrum. The vast majority of linear modes decayed rapidly; a small handful of persistent modes dominated the local dynamics. At least three of these modes were strongly oscillatory in nature (that is, the eigenvalues have a sizeable imaginary component) and all had a time constant between ~100 and ~400 ms. This range of time constants was consistent with the neural data and with the time span over which EMG showed strong high-frequency features. The range of oscillatory frequencies (~0.5–2.5 Hz) of the persistent modes agreed with the frequencies seen in **Figure 3b**, where oscillations were between 0.9 and 2.4 Hz. In summary, dynamics around the fixed point are notably simple: they are dominated by a small number of oscillatory modes that decay on timescales consistent with the neural data.

Dynamics can be inferred either by analysis of connectivity (as above) or by fitting the data directly with a dynamical system (a step in jPCA). Both methods involve approximations, but one would nevertheless hope that they would roughly agree. If they do not, then the goal of inferring dynamics from data would be unobtainable without a full connectome. We therefore compared, for the model, the eigenvalues found by analyzing connectivity (**Fig. 7b**) with the eigenvalues found by applying jPCA. The eigenvalues reported by jPCA (which were constrained to be purely imaginary) revealed three frequencies that closely agreed with the top three frequencies found by analyzing connectivity. The key planes in the RNN state space (determined by the associated eigenvectors) were also very similar for the jPCA and connection-based approaches (**Fig. 7c**). The two planes

Figure 5 Comparison of simulated and neural population responses. (**a**) Summary of canonical correlations. CCA analysis provides a spectrum of correlation coefficients that can be used to directly compare one multi-dimensional data set to another. The canonical coefficients are shown for the various models, each compared with the neural data (blue indicates regularized dynamical model, also shown in **Figure 4**; red indicates complicated dynamical model, black indicates untrained complicated dynamical model with inputs, green indicates velocity model, dark green indicates complicated kinematic model). (**b**) The average of the canonical correlations (average of lines in **a**) between the models and the data. The average canonical correlation provides a single number for each model that quantifies how closely the model population response matches the recorded population response.



closely overlapped. Thus, the first jPCA plane corresponded closely with the plane containing the fastest oscillation found by analyzing the connectivity (the plane corresponding with the third eigenvalue). Thus, both approaches agree that oscillations in the ~0.5–2.5-Hz range form a large component of the dynamics.

For the recorded neural data, it is impossible to perform analyses that require knowing all connections. However, one can still estimate dynamics by fitting the responses themselves. Doing so via jPCA revealed a set of eigenvalues (**Fig. 7a**) that closely match those of the model: one slightly faster than 2 Hz, one at about 1.5 Hz and one slightly below 1 Hz. We also computed the top eigenvalues for an unconstrained linear fit to the neural data. The frequency content of the unconstrained linear model also closely matched that of the regularized model. This confirms the results of the jPCA analysis in **Figure 3**: both the data and the model showed prominent oscillatory structure with a similar set of frequencies.

Dynamical models that match the data are simple

The above analyses indicate that the regularized RNN finds a solution that resembles, in many ways, that seen in the recorded population of motor cortex neurons. This is potentially quite surprising: the RNN was not fit to neural data and was not constrained to obey any particular connectivity. Furthermore, it seems unlikely that RNN optimization imitates either biological learning or evolution. Why did the regularized model find the solution of a single fixed point that produces oscillatory dynamics? Is there an advantage to this solution that might explain the similarity between model and data? To address this question, we constructed models that initially had extremely complicated dynamics, but, as a result of strong regularization during training, end up finding dynamically simple solutions (as a technical side-note, this exercise employed slightly simplified model parameters to ensure robustness across multiple optimizations; Online Methods).

As optimization proceeded, we saved 'snapshots' of networks during optimization and compared their responses with neural responses using CCA (as in **Fig. 5**). The average canonical correlation, and thus the similarity to data, rose steadily with optimization (**Fig. 8a**). As expected, EMG fit error falls during training (**Fig. 8b**). However, this effect was rapid, and fit error actually increased very slightly over the second half of the training. During this period, the regularization term is driving the model to find simpler and simpler solutions. As it does so, the similarity between model and data increases steadily. This did not occur when regularization was turned off: the fully trained complicated model fits EMG very well, but resembles the data only slightly more than a completely untrained network. Thus, model responses become more similar to the neural data during optimization as a result of the constraint that the network must use simple dynamics to reproduce EMG.

Do simpler solutions convey benefits? We analyzed the robustness of the fully trained regularized model (**Fig. 8a**) and the fully trained complicated model. To simulate the effects of trial-by-trial noise, we analyzed how the models responded to random perturbations in the preparatory period inputs (Online Methods). The regularized network yielded a much smaller error in the EMG output (**Fig. 8c**). To simulate the effects of synaptic changes, such as dying neurons or unreliable synapses, we examined robustness to structural perturbations of the connectivity matrix (**J** in equation (1)). Again, the regularized network was much more robust than the complicated model (**Fig. 8d**).

Further model comparisons and extensions

Does the regularized model (having been built to generate EMG) perhaps resemble the neural data simply because the neural data resemble the EMG? Or are there response features in the model and neural populations that match, above and beyond, what is seen in the EMG? In short, there are, in multiple ways. First, both the regularized model population and the neural population showed rotational dynamics (**Fig. 3**), something not present in the muscle population[23]. Second, both the model and neural populations showed

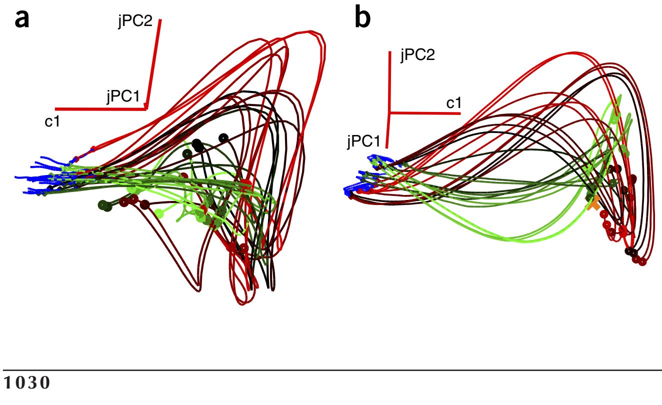


Figure 6 Monkey J and regularized model state-space visualizations. (**a**) Three-dimensional visualization of the neural data during the movement period for monkey J. The projection is comprised of the first jPC plane (**Fig. 3a**, left panel) and an additional dimension that captures variance from the cross-condition mean. Each trace is color-coded to show one of the 27 reach conditions. For all conditions, the trajectory during the preparatory period is colored blue. Time shown is 400 ms before to 220 ms after movement onset. Note that the jPC1 axis is projecting into the page. (**b**) Analogous three-dimensional visualization of the regularized model for monkey J (data are presented as in **a**). In addition, the single, condition-independent fixed point of the model, which organizes the dynamics of movement generation, is shown with an orange x. Time shown is 1,000 ms before to 220 ms after movement onset.

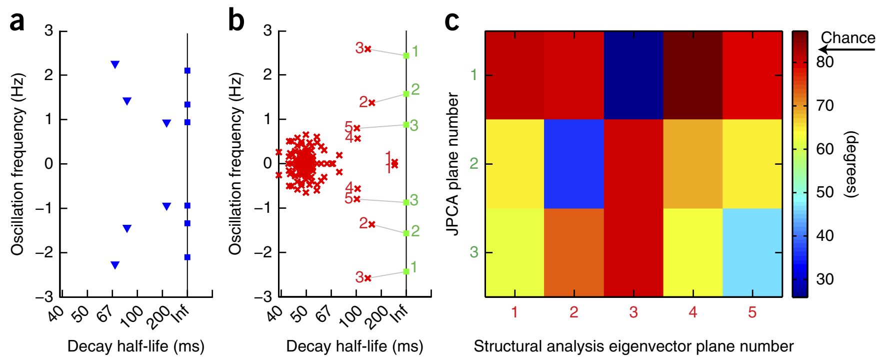


Figure 7 Frequency analysis of neural data and regularized model for monkey J. (**a**) Eigenvalue analysis of the neural data. Shown on the line of stability (Inf, neither decaying nor growing) are the purely imaginary eigenvalues associated with the jPCA analysis of the neural data in **Figure 3a** (blue squares). Also shown are the top eigenvalues of an unconstrained linear fit to the neural data (blue triangles). (**b**) The complex eigenvalue spectrum of the linearized system around the fixed point in the regularized model for monkey J (red x marks) based on a structural analysis of the weight matrix. Highlighted with red numbers are those modes of the linearized system that have a slow decay. Shown along the line of stability are the purely imaginary eigenvalues associated with the jPCA analysis of the regularized model data (green squares). Gray lines show the connection between the jPCA analysis and the structural analysis, as given by subspace angle analysis of eigenvectors in **c**. (**c**) Subspace angle analysis for the model, comparing the jPC planes (**b**, green squares) with the eigenvectors of the linearized system around the fixed point (**b**, red x marks). On the horizontal axis are listed the five slowest decaying oscillatory modes of the linearized system (corresponding to the red numbered modes in **b**). On the vertical axis are listed the three oscillatory planes found by jPCA (corresponding to the green numbered modes in **b**). Color indicates the minimum subspace angle (the minimum angle between the corresponding planes). For comparison, the minimum subspace angle between two randomly chosen planes in a $N = 300$ D space is 84 ±2 degrees (mean and s.d., black arrow labeled chance). Thus, a minimum subspace angle of 30–40 degrees indicates highly overlapping subspaces. In the present case, jPC plane 1 overlapped heavily with mode 3 (the highest frequency), jPC plane 2 overlapped heavily with oscillatory mode 2 (the second highest frequency) and jPC plane 3 overlapped more modestly with oscillatory mode 5 (the third highest frequency).

preparatory activity, which is essentially absent in the EMG. Third, both the model and the data were higher dimensional than the EMG itself (**Supplementary Fig. 11c,d**). For the model (and, by extension, possibly for the data), this higher dimensionality is a straightforward consequence of the fact that the internal dynamics that generate EMG must be higher dimensional than the final output. Finally, quantitative comparison via canonical correlation analysis revealed that the data resemble the regularized model more strongly than they do the EMG itself. Indeed, of all the possible comparisons—EMG activity, the regularized model, the complicated model, the various kinematic models—the one that most resembled the data is the regularized model (**Supplementary Fig. 11a,b**). Finally, models that incorporate muscle synergies or spinal cord modules also resembled the neural data closely (**Supplementary Fig. 1c,d**).

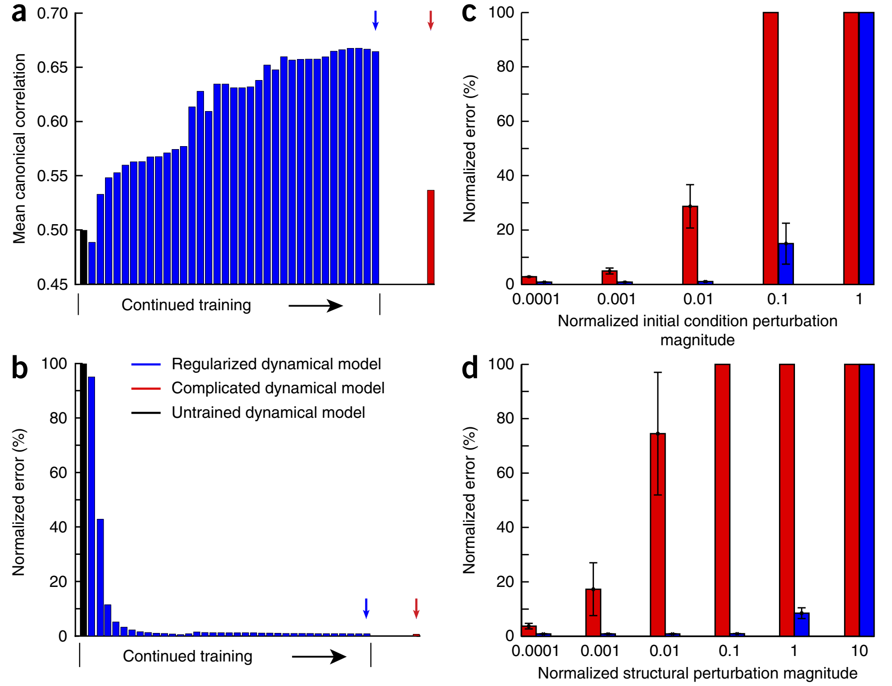


Figure 8 Regularization affects similarity to data and model robustness. (**a**) Average canonical correlation, as training progresses, between the regularized model and the neural data from monkey J. To provide a baseline, the black bar shows the mean canonical correlation between the untrained model with correct inputs and the neural data (0.50). As training with regularization progresses (blue), the model becomes more and more similar to the neural data, ending with a mean canonical correlation of 0.67 for this model (blue arrow). When trained to generate EMG without any regularization, the model has a mean canonical correlation with the data of 0.53 (red arrow). Black shows the canonical correlation of the untrained model with the data from monkey J. (**b**) The normalized error of the network output for the regularized model. Error decreased very quickly, even while the mean canonical correlation (**a**) continued to increase over a much longer period of training. The final error for the regularized model was comparable to the final training error for the complicated model (red). (**c**) Perturbation test of the initial conditions for the regularized and complicated models analyzed in **a** (blue and red arrows, respectively). The inputs were randomly perturbed according to a normalized percentage of the input strength (as given on horizontal axis). The network was then run and the mean normalized EMG error of the outputs (vertical axis) was averaged across 50 repetitions of this procedure. Error bars show s.d. The vertical axis is truncated at 100% error. (**d**) A structural perturbation test of the recurrent connectivity matrix in equation (1) for the regularized and complicated models analyzed in **a** (blue and red arrows, respectively). The connectivity matrix was randomly perturbed 50 times according to a normalized percentage of the mean absolute connection strength (as given on horizontal axis). The perturbed network was then run and the mean normalized EMG error of the outputs was averaged (vertical axis). Error bars show s.d. The vertical axis is truncated at 100% error.

DISCUSSION

Our central result is that an RNN trained to produce EMG exhibited dynamics that strongly resemble the empirically estimated dynamics of motor cortex, but only if model optimization promoted a highly regularized (that is, simple) solution. The resemblance between the regularized model and data was manifested at the level of single neuron PSTHs, at the level of oscillatory population trajectories and in direct quantitative comparison via CCA. Notably, this agreement was not achieved by fitting the RNN to the neural data. The simple preparatory period inputs to the RNN were derived from the neural data, but the RNN had no inputs that indicated the 'correct' patterns of movement period activity, nor was it trained to reproduce those patterns. Rather, the agreement between model and data emerged as a result of two factors: the need to generate the actual patterns of EMG and the requirement that the model use simple dynamics.

Analysis of regularized model dynamics revealed a sequence of four events. First, during the preparatory period, condition-specific inputs produce a set of states (one per condition) that act as initial states for the upcoming movement-period dynamical system. Second, the movement-period dynamical system is produced by the simultaneous removal of the hold cue and the condition-specific inputs. Third, movement-period dynamics are dominated by a single, condition-independent fixed point with approximately linear and strongly oscillatory dynamics. Fourth, those dynamics yield neural trajectories whose projections onto the output dimensions produce the patterns of EMG. The similarity of this sequence in model and data lend support to the view that motor cortex concerns itself with low-level features of movement generation[1,2,4,11,17,18,33,34].

Our modeling study provides a unified dynamical framework in which to understand a number of experimental findings. The model solution accords with the proposal that a key purpose of preparatory activity is to establish an attractive neural state that is appropriate, when triggered, to produce the desired movement[22,35]. Although we did not seek to model movement variability, the basic mechanics of the model are consistent with the finding that preparatory variability has behavioral consequences[36]. Finally, the network successfully generated unchanging EMG during the preparatory period. To achieve this, the model employed a muscle-null space to prevent preparatory period dynamics in the network from perturbing the output[37].

Two key features emerged when the network was optimized with regularization. First, the network became much more robust to perturbations of both inputs and connectivity, an anticipated and desirable consequence of regularization. Second, the network developed simple oscillatory dynamics that resembled the data. This resemblance increased steadily with training as the network found simpler and simpler solutions. These results indicate that relatively simple quasi-oscillatory dynamics are a natural and robust way of solving the problem of pattern generation.

This finding suggests an intriguing analogy between pattern generation in the motor cortices and encoding in the visual cortices. A previous study[38] optimized a feedforward neural network to encode natural images. Optimization yielded Gabor filters, resembling empirical receptive fields, but only when regularization encouraged sparseness. By analogy, there are many ways to generate EMG, and our network produces cortex-like responses only when regularized to encourage simple solutions.

It has long been debated whether spatial tuning in motor cortex (that is, cosine tuning for direction) reflects an abstract code for direction or a mechanistic role in the production of muscle forces. Recent models that embody the latter view[4,16,19,39,40] successfully predict properties of directional tuning, including the presence of broad tuning, the distribution of 'preferred directions', and shifts in tuning and response gain with starting position (see ref. 2). The model presented here is very much in this vein—the network was trained to produce patterns of EMG, but we concentrated much more on temporal response properties. For fast reaches, the empirical neural responses were very temporally complex and defied concise description in terms of a preferred direction. These same properties were seen in our model and reflect the mechanism used to produce EMG. That mechanism involved a set of rotations spanning a handful of planes in state space. The response of each individual neuron was an essentially random projection of this rotational subspace, resulting in the observed complexity and heterogeneity.

Our focus on temporal pattern generation is shared with a number of other models. In particular[11,41], it was proposed that response complexity might naturally be explained by a recurrent network. A recent study[15] employed a model with oscillatory (and rectilinear) components as a means for controlling a simple arm model. A major difference between the two approaches is that we began with EMG data and employed a systems identification strategy to discover the mechanism of a dynamical system that could generate the empirical EMG. The solution is in broad conceptual agreement with the previous study[15].

More generally, there has been considerable recent focus on the broad topic of pattern generating networks[24,42–44]. For example, inhibition-stabilized networks can generate temporally patterned outputs[45] via a basis set that includes quasi-oscillatory patterns, in qualitative agreement with ref. 23. However, the resulting 'non-normal' dynamics were not trained to produce any particular pattern—they simply contain a rich basis set of useful patterns. The generic nature of those patterns makes it unlikely that the model population quantitatively resembles the neural data from motor cortex. Yet it is quite possible that future modifications of that model, including optimization and regularization, might allow it to successfully fit EMG and match the neural data. More broadly, pattern-generating network models derive dynamics from recurrence, which can result from either internal connections (for example, the present model[41,45]) or external sensory feedback (for example, see ref. 16). Although these represent different model classes, one can anticipate unifying extensions. For example, a previously described model[16] employs only external feedback, but its replication of empirical preferred-direction distributions would likely hold were it extended to include efference copy or other internal feedback. Similarly, our model continued to find the same basic dynamical solution if provided with sensory feedback that was a filtered version of its output (**Supplementary Fig. 1a**,**b**,**e**,**f**). In this context, it should be stressed that, although our model reveals robust dynamical solution to the problem of producing multiphasic EMG, the scope of the recurrent circuitry, cortical, central and/or feedback, supporting those dynamics remains an open question. What is clear is that dynamics similar to those exhibited by the model can be seen in motor cortex. This is consistent with the interpretation that, however broad the relevant recurrent circuitry might be, motor cortex is sufficiently central that many key aspects of the dynamics can be observed there.

The dynamical systems view of movement generation carries some general implications. First, model units contain a variety of responses that sometimes resemble the time course of position, velocity, speed and other variables. Yet none of these parameters is truly represented by the model. Furthermore, although the model certainly contains an implicit representation of the upcoming EMG, individual-neuron responses rarely match the patterns of EMG. The reason is not only that EMG-like signals are 'mixed' across neurons, but also that the network contains response components that are required for pattern generation, but do not resemble the final output. Just as a simple two-dimensional oscillator needs both a sine and a cosine as a dynamical necessity, pattern generation will typically require extra internal patterns necessary to support the dynamics. In the case of the model, and by extension possibly in the case of the data, it would be a mistake to explain each neuron's response as a representation of meaningful variables. Rather, the model should be understood through the set of population-level latent variables, their response to inputs, their internal dynamics and their influence on the output projection. This will be true not only of pattern generating networks, but of many networks with strong dynamics that subserve internal computations (for example, see ref. 32). In summary, it should be no surprise that individual-neuron responses are often quite mysterious[10,11,23,32,46]. Understanding neuronal responses in recurrent networks necessitates going beyond population analyses that read out variables and instead adopting population analyses that capture the internal dynamics underlying the central computations.

METHODS

Methods and any associated references are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank L. Abbott and S. Ganguli for insightful conversations. We thank S.I. Ryu for electrode array implantation surgical assistance, M. Mazariegos and J. Aguayo for surgical assistance and veterinary care, B. Otskotsky for IT support, and S. Eisensee, B. Davis and E. Castaneda for administrative assistance. This work was supported by The Searle Scholars Program (M.M.C.), The Sloan Foundation (M.M.C.), The McKnight Foundation (M.M.C.), The Grossman Charitable Trust (M.M.C.), US National Institutes of Health (NIH) Director's New Innovator Award DP2 NS083037 (M.M.C.), an NIH R01 MH93338-02 grant (M.M.C.), Burroughs Wellcome Fund Career Awards in the Biomedical Sciences (M.M.C., K.V.S.), a National Science Foundation graduate research fellowship (M.T.K.), an NIH T-RO1 Award R01NS076460 (K.V.S.), an NIH Director's Pioneer Award 8DP1HD075623 (K.V.S.) and a DARPA REPAIR Award N66001-10-C-2010 (K.V.S.).

AUTHOR CONTRIBUTIONS

D.C.S. and M.M.C. conceived the study and designed the simulations. D.C.S. performed the simulations and analyses. M.M.C. and M.T.K. designed and performed the experiments. D.C.S. and M.M.C. wrote the manuscript. K.V.S. supervised the study, analyses and manuscript writing.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Evarts, E.V. Relation of pyramidal tract activity to force exerted during voluntary movement. *J. Neurophysiol.* **31**, 14–27 (1968).
2. Mussa-Ivaldi, F.A. Do neurons in the motor cortex encode movement direction? An alternative hypothesis. *Neurosci. Lett.* **91**, 106–111 (1988).
3. Sanger, T.D. Theoretical considerations for the analysis of population coding in motor cortex. *Neural Comput.* **6**, 29–37 (1994).
4. Todorov, E. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat. Neurosci.* **3**, 391–398 (2000).
5. Hatsopoulos, N.G. Encoding in the motor cortex: was evarts right after all? Focus on 'motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks'. *J. Neurophysiol.* **94**, 2261–2262 (2005).
6. Scott, S.H. Inconvenient truths about neural processing in primary motor cortex. *J. Physiol. (Lond.)* **586**, 1217–1224 (2008).
7. Aflalo, T.N. & Graziano, M.S.A. Relationship between unconstrained arm movements and single-neuron firing in the macaque motor cortex. *J. Neurosci.* **27**, 2760–2780 (2007).
8. Kalaska, J.F. From intention to action: motor cortex and the control of reaching movements. *Adv. Exp. Med. Biol.* **629**, 139–178 (2009).
9. Georgopoulos, A.P., Schwartz, A.B. & Kettner, R.E. Neuronal population coding of movement direction. *Science* **233**, 1416–1419 (1986).
10. Shenoy, K.V., Sahani, M. & Churchland, M.M. Cortical control of arm movements: a dynamical systems perspective. *Annu. Rev. Neurosci.* **36**, 337–359 (2013).
11. Fetz, E. Are movement parameters recognizably coded in the activity of single neurons? *Behav. Brain Sci.* **15**, 679–690 (1992).
12. Pearce, T.M. & Moran, D.W. Strategy-dependent encoding of planned arm movements in the dorsal premotor cortex. *Science* **337**, 984–988 (2012).
13. Schwartz, A.B. Direct cortical representation of drawing. *Science* **265**, 540–542 (1994).
14. Fetz, E.E. Cortical mechanisms controlling limb movement. *Curr. Opin. Neurobiol.* **3**, 932–939 (1993).
15. Rokni, U. & Sompolinsky, H. How the brain generates movement. *Neural Comput.* **24**, 289–331 (2012).
16. Lillicrap, T.P. & Scott, S.H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77**, 168–179 (2013).
17. Morrow, M.M. & Miller, L.E. Prediction of muscle activity by populations of sequentially recorded primary motor cortex neurons. *J. Neurophysiol.* **89**, 2279–2288 (2003).
18. Schieber, M.H. & Rivlis, G. Partial reconstruction of muscle activity from a pruned network of diverse motor cortex neurons. *J. Neurophysiol.* **97**, 70–82 (2007).
19. Ajemian, R. *et al.* Assessing the function of motor cortex: single-neuron models of how neural response is modulated by limb biomechanics. *Neuron* **58**, 414–428 (2008).
20. Buys, E.J., Lemon, R.N., Mantel, G.W. & Muir, R.B. Selective facilitation of different hand muscles by single corticospinal neurones in the conscious monkey. *J. Physiol. (Lond.)* **381**, 529–549 (1986).
21. Churchland, M.M. & Cunningham, J.P. A dynamical basis set for generating reaches. *Cold Spring Harb. Symp. Quant. Biol.* Published online, doi:10.1101/sqb.2014.79.024703 (7 April 2015).
22. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Ryu, S.I. & Shenoy, K.V. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron* **68**, 387–400 (2010).
23. Churchland, M.M. *et al.* Neural population dynamics during reaching. *Nature* **487**, 51–56 (2012).
24. Sussillo, D. & Abbott, L.F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557 (2009).
25. Tanji, J. & Evarts, E.V. Anticipatory activity of motor cortex neurons in relation to direction of an intended movement. *J. Neurophysiol.* **39**, 1062–1068 (1976).
26. Weinrich, M., Wise, S.P. & Mauritz, K.H. A neurophysiological study of the premotor cortex in the rhesus monkey. *Brain* **107**, 385–414 (1984).
27. Pruszynski, J.A., Omrani, M. & Scott, S.H. Goal-dependent modulation of fast feedback responses in primary motor cortex. *J. Neurosci.* **34**, 4608–4617 (2014).
28. Todorov, E. & Jordan, M.I. Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* **5**, 1226–1235 (2002).
29. Scott, S.H. Optimal feedback control and the neural basis of volitional motor control. *Nat. Rev. Neurosci.* **5**, 532–546 (2004).
30. Doya, K. Bifurcations in the learning of recurrent neural networks. *Proc. IEEE Int. Symp. Circuits Syst.* **6**, 2777–2780 (1992).
31. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–649 (2013).
32. Mante, V., Sussillo, D., Shenoy, K.V. & Newsome, W.T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
33. Fetz, E.E. & Cheney, P.D. Postspike facilitation of forelimb muscle activity by primate corticomotoneuronal cells. *J. Neurophysiol.* **44**, 751–772 (1980).
34. Rathelot, J.-A. & Strick, P.L. Muscle representation in the macaque motor cortex: an anatomical perspective. *Proc. Natl. Acad. Sci. USA* **103**, 8257–8262 (2006).
35. Churchland, M.M., Yu, B.M., Ryu, S.I., Santhanam, G. & Shenoy, K.V. Neural variability in premotor cortex provides a signature of motor preparation. *J. Neurosci.* **26**, 3697–3712 (2006).
36. Churchland, M.M., Afshar, A. & Shenoy, K.V. A central source of movement variability. *Neuron* **52**, 1085–1096 (2006).
37. Kaufman, M.T., Churchland, M.M., Ryu, S.I. & Shenoy, K.V. Cortical activity in the null space: permitting preparation without movement. *Nat. Neurosci.* **17**, 440–448 (2014).
38. Olshausen, B.A. & Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
39. Tanaka, H. & Sejnowski, T.J. Computing reaching dynamics in motor cortex with Cartesian spatial coordinates. *J. Neurophysiol.* **109**, 1182–1201 (2013).
40. Todorov, E. Cosine tuning minimizes motor errors. *Neural Comput.* **14**, 1233–1260 (2002).
41. Maier, M.A., Shupe, L.E. & Fetz, E.E. Dynamic neural network models of the premotoneuronal circuitry controlling wrist movements in primates. *J. Comput. Neurosci.* **19**, 125–146 (2005).
42. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
43. Maass, W., Joshi, P. & Sontag, E.D. Computational aspects of feedback in neural circuits. *PLOS Comput. Biol.* **3**, e165 (2007).
44. Laje, R. & Buonomano, D.V. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* **16**, 925–933 (2013).
45. Hennequin, G., Vogels, T.P. & Gerstner, W. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron* **82**, 1394–1406 (2014).
46. Churchland, M.M. & Shenoy, K.V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J. Neurophysiol.* **97**, 4235–4257 (2007).

ONLINE METHODS

Recordings of physiological data. Recordings were made from the cortex of two monkeys performing a delayed reach task (**Fig. 1a**). Animal protocols were approved by the Stanford University Institutional Animal Care and Use Committee. Our basic methods have been described previously[22,23,47,48]. Briefly, monkeys performed both straight reaches and reaches that curved around one or more intervening barriers. This task was beneficial because of the large variety of different reaches, and thus EMG patterns, that were evoked. There were 27 different reach types (conditions), each of which was repeated many times (~20–50 trials). Each trial began when a central spot was visually fixated, touched and held briefly. The onset of a target (and any accompanying barriers) marked the beginning of the preparatory period. If the hand or eye moved during this period the trial was aborted. The preparatory period ranged from 0–1,000 ms. Only trials with preparatory periods >400 ms were analyzed. Physiological recordings (neural and EMG) were averaged across trials and filtered[23] to create a smooth rate as a function of time. Averages were made locked to target onset, the go cue and movement onset. To create a single trace as a function of time, these three traces were truncated and aligned, and the resulting gaps between them were interpolated[22].

Recordings were made from M1 (both surface and sulcal) and from the adjacent (caudal) aspect of dorsal premotor cortex (PMd) using both standard single-electrode and array recording techniques (Blackrock Microsystems). For each monkey we created a single large data set that included neurons recorded using both techniques (161 and 307 units for monkeys J and N). Sulcal M1, surface M1 and caudal PMd are contiguous. While there are important differences in their average response properties (for example, preparatory period activity is more common in PMd), these differences are far from absolute: M1-like neurons are frequently found in caudal PMd and vice versa. Our principal analyses thus considered all neurons without attempting to divide based on either anatomy or response properties. **Supplementary Figure 10** provides an additional analysis where anatomy is considered.

EMG data were recorded, as described previously[35], from the major muscles of the upper arm. When feasible we included repeated recordings from different aspects of key muscles, and the target of the model was based on the highest quality recordings (7 and 8 for monkeys J and N, respectively). For both monkeys we employed recordings from the anterior, medial and posterior deltoid, pectoralis major, trapezius and biceps brachii. For monkey J we included a second recording from the biceps brachii. For monkey N we included two additional recordings from the trapezius, and one additional recording from the anterior deltoid. EMG records were rectified, smoothed and averaged before further analysis. Sampling error (due to a finite number of trials) resulted in small idiosyncratic differences between conditions during the baseline and preparatory periods. To avoid having the model attempt to fit these small differences, they were simply removed before fitting.

Representational models. In addition to the RNNs, we simulated two models, the velocity model and complex kinematic model, for which neural activity was 'tuned' for standard movement parameters (**Fig. 5** and **Supplementary Figs. 5** and **11**). These models took the general form

$$r_n(t) = f_n(param_1(t), param_2(t), param_3(t), \ldots)$$

where $r_n(t)$ is the firing rate of neuron n at time t, f_n is a tuning function, and $param_1(t), param_2(t) \ldots$ are represented parameters such as hand velocity or target position. These models are described in ref. 23. Briefly, in the velocity-tuned model, movement-period activity was tuned for horizontal reach velocity, vertical reach velocity and reach speed. Each unit thus had a 'preferred direction' in velocity space. Preferred directions were assigned randomly. Preparatory activity was based upon three additional underlying factors: horizontal reach endpoint, vertical reach endpoint and peak reach speed. The complex kinematic model was similar, but units were tuned to a greater variety of kinematic factors: position, velocity, acceleration and jerk.

RNN definition. We implemented the dynamical system, $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u})$, using a standard continuous-time RNN equation of the form

$$\tau \dot{x}_i(t) = -x_i + \sum_{k=1}^{N} J_{ik} r_k(t) + \sum_{k=1}^{I} B_{ik} u_k(t) + b_i^x \qquad (1)$$

where the x_i variables are the activations of the network units and r_k are the corresponding firing rates. The network has N units and I inputs. The firing rates are related to the activation variables via a saturating nonlinearity (see **Supplementary Table 1** and other details below). The variables in the network interact through the synaptic weight matrix, $\mathbf{J}$. The inputs to the system are given by u_k and come into the system through input weights, $\mathbf{B}$. The units each have an offset bias, b_i^x. A single time constant, τ, sets the time-scale of the network.

In order to compose EMG from network activity, we define a linear readout

$$z_i(t) = \sum_{k=1}^{N} W_{ik} r_k(t) + b_i^z$$

The readout, z_i, is a weighted sum of the firing rates with weights, $\mathbf{W}_{i:}$, plus a bias term, b_i^z. There are M readouts, one per recorded muscle.

For all models, the value of τ was 50 ms, and N was 300 (see **Supplementary Table 1** for parameters that varied by model). The condition-specific inputs were six-dimensional (see below). In addition we added a condition-independent hold cue input. Thus, I was 7. The elements of $\mathbf{J}$ were initialized with zero mean, Gaussian entries with variance g^2/N. The elements of $\mathbf{B}$ were initialized with zero mean, Gaussian entries with variance h^2/I. The output weights and all biases were initialized to 0. The network was simulated using Euler integration with time steps of $\tau / 10 = 5$ ms. There were two sets of models per monkey, one for **Figures 1**–**7**, and a second for the analysis of training in **Figure 8**.

Training the network. Networks were optimized to generate multidimensional EMG. The error function was the squared error between the network output and the EMG

$$E = \frac{1}{CMT} \sum_{c=1}^{C} \sum_{m=1}^{M} \int_0^T (z_m(c,t) - EMG_m(c,t))^2 \, dt$$

where $EMG(c, t)$ is the M-dimensional EMG across all $C = 27$ reach conditions and across all time, T, including the baseline, preparatory and movement periods. The set of parameters modified to minimize E was $\{\mathbf{B}, \mathbf{J}, \mathbf{W}, \mathbf{b}^x, \mathbf{b}^z\}$. We report normalized error, which is E normalized by the EMG variance averaged over all conditions and muscles.

For the regularized models we modified the cost function to encourage the network to generate EMG as simply as possible. To this end, we included three separate regularization terms in the overall objective function: a standard L2 regularization on the weights, R_{L2}; a regularization on the firing rates, R_{FR}; and a novel regularization that encouraged simple dynamics, R_J. The error function minimized during training was

$$E^R = E + \alpha R_{L2} + \beta R_{FR} + \gamma R_J$$

with the α, β and γ hyperparameters setting the relative strengths of the regularization (**Supplementary Table 1**). The four terms that comprise the regularized error, taken together, dictate that the optimization procedure should create networks that produce an output very close to the empirical EMG (E), and do so as simply as possible in terms of dynamics (R_{L2}, R_{FR} and R_J).

The first regularization term is a standard L2 penalty on the input weights and the output weights, defined as

$$R_{L2} = \sum_{i,j=1}^{N,I} B_{ij}^2 + \sum_{i,j=1}^{M,N} W_{ij}^2$$

We included a second regularization term, defined as

$$R_{FR} = \frac{1}{CNT} \sum_{c=1}^{C} \sum_{i=1}^{N} \int_0^T r_i(c,t)^2 \, dt$$

This regularization helped to keep the simulated units from permanently saturating, something that rarely happens with biological neurons.

Finally, we included a novel form of regularization inspired by, but conceptually different from[49], and defined by

$$R_J = \frac{1}{CT} \sum_{c=1}^{C} \int_0^T \left\| \frac{\partial(\mathbf{J}\mathbf{r}(c,t))}{\partial \mathbf{x}(c,t)} \right\|_F^2 dt$$

where $||\cdot||_F$ is the Frobenius norm. Conceptually, R_J penalizes the network for making unnecessarily complicated state-space trajectories. It accomplishes this by forcing the first-order Taylor series expansion of network equation (1) to be a low-dimensional system, with all unnecessary modes decaying very quickly on a time scale of τ (that is R_J preserves only the decay term in equation (1)). As the linearized dynamical system around the single fixed point explains the functioning of the nonlinear RNN to good approximation[31], this is an intuitive approach to simplifying state-space dynamics. In implementing the derivative of R_J with respect to the network weights, we used a simplified derivative that computed the direct dependence of R_J on the parameter **J**, namely

$$\partial R_J / \partial J_{kl} = \frac{2}{CT} \sum_{c=1}^{C} \int_0^T J_{kl} (\mathrm{r}_l'(c,t))^2 \, dt$$

The second portion of the derivative, which gives the indirect dependence of R_J on previous values of $\mathbf{r}(c,t)$, was not used for two reasons. First, the majority of terms in the expression for the Hessian of the second portion are not guaranteed to be positive definite. Positive definiteness in the Hessian is required for the Hessian-Free optimization technique[50] used in this study. Second, the indirect term goes to zero as R_J goes to zero. Use of L2 weight regularization on J achieves similar results.

We optimized all network weights and biases to minimize E^R using the Hessian-Free (HF) training method for RNNs. HF is an exact second order method that uses back-propagation-through-time to compute the gradient of the error with respect to the network parameters. After training, all networks performed the task well; within 7% normalized error for monkey J and 3% normalized error for monkey N for the EMG during the movement period. While it was possible to reduce the EMG error quite a bit further, the ultimate goal of the study was to compare the model internals to the neural data collected for the monkeys. As such, we found that optimizing further did not help make the models more similar to the data, presumably due to irrelevance or noisiness of small features of the recorded EMG.

Model hyperparameters. There were a number of hyperparameters that were set manually when training the models, (for example, input gain, recurrent gain, amount of regularization, etc.; **Supplementary Table 1**). The hyperparameter g sets the scale of the recurrent weight matrix. If one picks $g < 1$, then the network will suffer from the vanishing gradient problem and be very difficult to train. If $g \gg 1$, then the network will be chaotic and may under some circumstances be difficult to train. In our studies, we examined the $g > 1$ and $g \gg 1$ chaotic initialization ranges.

To produce a model with rich dynamics (the complicated model) we used $g >> 1$ to produce dynamics that were initially rich, and did not regularize further during optimization (α, β, and $\gamma = 0$). To produce a model with simple dynamics (the 'regularized model') we used $g > 1$ to produce less rich initial dynamics and then simplified dynamics further via regularization during optimization. This successfully led to two classes models that both reproduced the EMG but with very different degrees of dynamic complexity. As a technical aside, versions of the regularized model could also be produced by initialization with $g >> 1$ and allowing regularization to produce the simplification. However, when using the rectified tanh function in combination with heavy regularization, in practice this nearly always led to stalled optimizations, presumably due to either local minima or pathological curvature. For this reason we avoided this regime ($g >> 1$ and regularization) for all simulations that use the rectified tanh (**Figs. 1**–**7**).

For the simulations in **Figure 8**, the goal was specifically to examine network behavior when dynamics are initially very rich (g>>1) and are then regularized slowly over the course of training. For this set of simulations we did not use the rectified tanh for the technical reason discussed above, and used the simpler tanh function instead. It would have been slightly preferable to continue to use the rectified tanh as it disallows negative firing rates and thus produces more realistic single-neuron responses. However, the use of the tanh allowed for robustly repeatable results for the analysis in **Figure 8** as the optimization was reliable for this simpler function, even under challenging circumstances. In practice the quantitative match between the neural and model responses was very nearly as good with the tanh as with the rectified tanh.

In summary, for each monkey we used two sets of hyperparameters (regularized and complicated model), each tailored to the needs of the simulations being performed. The first set was used for nearly all analyses, and the second was used for the analysis in **Figure 8** (**Supplementary Table 1**).

Inputs to the RNNs. The input to the RNNs contained a condition-independent hold cue, preceded by condition-specific inputs that indicated reach condition. Condition-specific inputs were derived from the preparatory period neural activity as follows: we took the time-and-trial averaged preparatory activity, a matrix of size $N \times C$, where N is the number of recorded neurons, and performed PCA to reduce it to a matrix of size $K \times C$. This yielded K numbers that allowed preparatory activity to encode a particular reach condition. We chose $K = 6$ to ensure that while inputs were not overly complex, they still captured much of the variance in the empirical data (81% and 72% for monkeys J and N) and were thus rich enough to distinguish between conditions. We created a simple temporal profile that turned this K-dimensional input on and off (**Fig. 1c**). The condition-independent hold cue was on at the beginning of the each simulation, and it turned off with the same offset dynamics of the condition-specific input (**Fig. 1c**). In summary, the networks received a seven-dimensional input, with $K = 6$ reach-dependent inputs and a single, condition-independent hold cue.

The regularized models of **Figures 1**–**7** (**Supplementary Table 1**) employed multiple delays between the onset of the preparatory input and the onset of the hold cue. We added this feature to avoid concerns about implicit time locking of model activity to the beginning of the simulation, and to ensure that the model was in fact producing EMG in response to the offset of the hold cue. Timing was thus as follows: the condition-specific inputs were followed, after a 100–800-ms delay, by the offset of the hold cue. The model EMG output began to change ~100 ms later. All simulation data shown had a delay of 650 ms.

jPCA. jPCA is described at length in ref. 23. Briefly, jPCA considers the neural state across times and conditions, $\mathbf{x}(t, c)$, and its temporal derivative, $\dot{\mathbf{x}}(t,c)$, and fits a linear model $\dot{\mathbf{x}}(t,c) = \mathbf{M}\mathbf{x}(t,c)$, where $\mathbf{M}$ is constrained to be skew-symmetric in order to test the hypothesis that the population state evolves according to oscillatory dynamics. jPCA provides summary features relevant to that hypothesis, including the quality of the fit, the eigenvalues and the associated frequencies. jPCA also allows visualization of any two-dimensional projections of the data that contain rotational structure. To ensure that jPCA focused on patterns that were robustly present, data were preprocessed using PCA to reduce dimensionality from the number of neurons or units in the data set to the 12 dimensions that captured the most variance. We analyzed a time period where neural activity was in strong flux: 280 ms before movement onset to 220 ms after movement onset. Unlike most analyses in ref. 23, the cross-condition mean was not subtracted from the neural responses. Some projections thus capture structure that is very similar across conditions. For present purposes, by not subtracting the mean we gain the advantage that it becomes straightforward to compare structure found via jPCA with structure found by linearizing around fixed points (see below). To do so, we compare the eigenvalues of the linearized dynamics with the eigenvalues of **M** (**Fig. 7b**). To ensure that the eigenvalues of **M** were not oscillatory simply due to the skew-symmetric constraint, we removed that constraint for one analysis (**Fig. 7a** and **Supplementary Fig. 7a**).

CCA. CCA was used to directly compare model and neural population responses. As a preprocessing step, both the monkey and model data were first reduced to ten dimensions using PCA. This ensured that CCA did not find dimensions of high correlation but low data variance. The period of comparison was broader than that for jPCA: from −400 ms to 400 ms relative to movement onset. This allowed CCA to compare activity before, during, and after the period where neural activity was in strong flux.

Fixed-point finding. To understand the mechanism embodied in the trained models, we used standard nonlinear dynamical systems methods of linearization around a fixed point. The application of this technique to high-dimensional RNNs was described in detail in ref. 31. The result is a set of points in state-space, $\{\mathbf{x}^{1^*}, \mathbf{x}^{2^*}, \mathbf{x}^{3^*}, \ldots\}$, where the dynamics described by equation (1) are at equilibrium, for example, $\dot{\mathbf{x}}^* = \mathbf{F}(\mathbf{x}^*, \mathbf{u}^{const}) = 0$, for some constant input, $\mathbf{u}^{const}$. These points are particularly insightful as the linearized dynamics around them approximately describe the nonlinear dynamics for some volume around the fixed point. Thus, for some region around the fixed point we can exchange the nonlinear dynamical system (1) for the linear dynamical system, $\dot{\boldsymbol{\delta}}\mathbf{x} = \mathbf{M}\boldsymbol{\delta}\mathbf{x}$, with $\boldsymbol{\delta}\mathbf{x} \equiv \mathbf{x} - \mathbf{x}^*$, and $\mathbf{M} \equiv \mathbf{F}'(\mathbf{x}^*)$, the first-order Taylor series expansion of $\mathbf{F}(\mathbf{x}, \mathbf{u}^{const})$ around $\mathbf{x}^*$.

For the regularized models, we performed the fixed-point analysis during the movement period. During the movement period, when all inputs were turned off, that is $\mathbf{u}^{const} = 0$, the fixed-point analysis yielded a single fixed point for almost all randomly initialized models (the remaining models had a tight cluster of 2–3 fixed points). In **Figure 7b**,**c** (**Supplementary Fig. 7b**,**c**) we examined the eigenvalues and eigenvectors of **M**, the linear dynamical system around the single fixed point, to determine the nature of the dynamical system that generated the EMG signals.

Three-dimensional visualization. We visualized the population response by projecting it into a three-dimensional subspace (**Fig. 6** and **Supplementary Fig. 6**). We chose the subspace spanned by the first jPC plane (labeled j1 and j2 in the figures) as well as an additional dimension that captured the variance of the cross-condition mean, labeled c1. We did so because such a dimension was prominent for both the neural and model data. The cross-condition mean was defined as the trajectory through time when all 27 conditions were averaged together. To find dimensions that reveal this trajectory, we computed the top two PCs of the $N \times T$ cross-condition mean data matrix. For both the data and the regularized data sets this always revealed a dimension that captured a largely monotonic change in the cross-condition mean. For example this was PC2 for monkey J and PC1 for monkey N. Since the variance in the top two PCs is roughly comparable, the choice of whether to use PC 1 or PC 2 was based on which captured the monotonic trajectory. The visualization subspace was defined by orthogonalizing these three vectors (j1, j2 and c1). The axes in each plot indicate the original three vectors before orthogonalization. The vector describing the cross-condition mean was largely orthogonal to the plane described by j1 and j2.

Subspace overlap analysis. The first subspace angle (also known as the first principal angle) was used to compare how closely two subspaces overlapped (**Fig. 7c** and **Supplementary Fig. 7c**). The first subspace angle gives the largest of all the angles necessary to rotate one high-dimensional subspace into the other. A subspace angle of zero indicates that the two subspaces span the same space. A subspace angle of 90 degrees indicates that there is at least one dimension in one subspace that is orthogonal to all dimensions in the other. However, there may still be considerable overlap among the other dimensions. Thus, the subspace angle is conservative when considering many dimensions: a subspace angle of 30 degrees in an $N = 300$D space indicates that two subspaces are extremely similar. In **Figure 7c**, for the regularized model, we compared the subspace spanned by each of the jPCA planes to that spanned by each of the five oscillatory planes, found by eigenvector analysis applied to the matrix **M**, of the linearized system, $\dot{\boldsymbol{\delta}}\mathbf{x} = \mathbf{M}\boldsymbol{\delta}\mathbf{x}$.

Input perturbation robustness analysis. We added a random, Gaussian distributed constant to each of the six condition-specific inputs that specified which of the 27 reaches the network should generate. The random constant was scaled as a normalized percentage of input scale. This process was repeated 50 times for each perturbation level and the errors were averaged to yield **Figure 8c** (**Supplementary Fig. 8c**). The period of comparison was between 400 ms before movement onset to 400 ms after movement onset.

Structural robustness analysis. Structural noise was added to the **J** matrix in the form of additive Gaussian perturbations. Specifically, a trained recurrent matrix, **J**, was transformed by $J_{ij} \leftarrow J_{ij} + \beta_{ij}$, where β_{ij} was sampled from a zero-mean, Gaussian distribution. The variance of this distribution was scaled to the normalized mean absolute weight of the regularized model. This process was repeated 50 times for each perturbation level and the errors were averaged to yield **Figure 8d** (**Supplementary Fig. 8d**). The period of comparison was between 400 ms before movement onset to 400 ms after movement onset.

A **Supplementary Methods Checklist** is available.

47. Kaufman, M.T. *et al.* Roles of monkey premotor neuron classes in movement preparation and execution. *J. Neurophysiol.* **104**, 799–810 (2010).

48. Kaufman, M.T., Churchland, M.M. & Shenoy, K.V. The roles of monkey M1 neuron classes in movement preparation and execution. *J. Neurophysiol.* **110**, 817–825 (2013).

49. Rifai, S. *et al.* Learning invariant features through local space contraction. Preprint at http://arxiv.org/abs/1104.4153 (2011).

50. Martens, J. & Sutskever, I. Learning recurrent neural networks with hessian-free optimization. *Proc. 28th Int. Conf. Mach. Learn.* (2011).