Systems/Circuits

A Spiking Neural Integrator Model of the Adaptive Control of Action by the Medial Prefrontal Cortex

Trevor Bekolay,[1] **Mark Laubach,**[2] **and Chris Eliasmith**[1]

[1]Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and [2]The John B. Pierce Laboratory and Department of Neurobiology, Yale University School of Medicine, New Haven, Connecticut 06519

Subjects performing simple reaction-time tasks can improve reaction times by learning the expected timing of action-imperative stimuli and preparing movements in advance. Success or failure on the previous trial is often an important factor for determining whether a subject will attempt to time the stimulus or wait for it to occur before initiating action. The medial prefrontal cortex (mPFC) has been implicated in enabling the top-down control of action depending on the outcome of the previous trial. Analysis of spike activity from the rat mPFC suggests that neural integration is a key mechanism for adaptive control in precisely timed tasks. We show through simulation that a spiking neural network consisting of coupled neural integrators captures the neural dynamics of the experimentally recorded mPFC. Errors lead to deviations in the normal dynamics of the system, a process that could enable learning from past mistakes. We expand on this coupled integrator network to construct a spiking neural network that performs a reaction-time task by following either a cue-response or timing strategy, and show that it performs the task with similar reaction times as experimental subjects while maintaining the same spiking dynamics as the experimentally recorded mPFC.

Key words: adaptive control; delayed response RT task; mPFC; neural dynamics; neural engineering framework; neural integrator

Introduction

The medial prefrontal cortex (mPFC) has been implicated in regulating reinforcement learning parameters (Amiez et al., 2005; Khamassi et al., 2013), conflict monitoring (Botvinick et al., 2004; van Veen et al., 2004; Aarts et al., 2009; Sheth et al., 2012), performance monitoring (Carter et al., 1998; Fecteau and Munoz, 2003; Rushworth et al., 2004; Brown and Braver, 2005; Modirrousta and Fellows, 2008; Histed et al., 2009; Alexander and Brown, 2011; Horst and Laubach, 2012; Hyman et al., 2013), anticipatory control (Koyama et al., 2001; Aarts et al., 2008), arousal of the sympathetic nervous system (Critchley et al., 2003; Luu and Posner, 2003), and control of action timing (Muir et al., 1996; Naito et al., 2000; Mulert et al., 2003; Risterucci et al., 2003; Narayanan and Laubach, 2009). Conceptual models have been proposed that would allow the mPFC to be involved in several of these functions, depending on behavioral context (Luu and Posner, 2003; Botvinick et al., 2004). Mathematical models based on reinforcement learning explain experimental results at several levels (Alexander and Brown, 2011; Khamassi et al., 2013). However, these reinforcement learning models have not been shown to explain spiking activity of neural ensembles recorded during temporally constrained behavioral tasks. Here, we present a dynamical spiking neural model that relates directly to both neural and behavioral data in a simple mPFC-dependent task.

Specifically, we propose a two-dimensional dynamical system that can account for mPFC activity related to error monitoring and action timing in rodents performing a simple reaction-time task. Neural recordings in the motor cortex during the acquisition of this task reveal that an encoding of forthcoming errors develops with learning (Laubach et al., 2000). These activities depend on processing in the mPFC (Narayanan and Laubach, 2006), where neurons encode errors both prospectively and retrospectively (Narayanan and Laubach, 2006, 2008). Analyses of population activity during the task show that both the mPFC and motor cortex are dominated by relatively slow fluctuations in firing rates, starting from the initiation of the trial and terminating when feedback is given about the trial's outcome (Narayanan and Laubach, 2009).

Here, we show that a specific type of network structure, called a double-integrator network (Singh and Eliasmith, 2006), can play a central role in accounting for reaction-time task performance. With analyses identical to the experimental study, we show that the neural activity produced by the simulated spiking network closely matches the experimental data from the Narayanan and Laubach (2009) study. Furthermore, we propose a feedforward control system that can perform the simple reaction-time task and show that the double-integrator network can be used to modify the behavior of the control system based on the outcome of the previous trial. When the control system is implemented in a spiking neural network, the modified behavior

Received June 7, 2013; revised Nov. 29, 2013; accepted Dec. 23, 2013.

Author contributions: T.B., M.L., and C.E. designed research; T.B. and C.E. performed research; T.B., M.L., and C.E. contributed unpublished reagents/analytic tools; T.B. and M.L. analyzed data; T.B., M.L., and C.E. wrote the paper.

This work was supported by NSF Grant 1121147 to M.L., NSERC Discovery, CRC, CFI, and OIT to C.E., and NSERC CGS-D to T.B. We thank Benjamine Liu for technical contributions to the modeling effort and Nandakumar Narayanan who recorded the neurons that were analyzed by T.B. and M.L.

The authors declare no competing financial interests.

Correspondence should be addressed to Dr Chris Eliasmith, Centre for Theoretical Neuroscience, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1. E-mail: celiasmith@uwaterloo.ca.

DOI:10.1523/JNEUROSCI.2421-13.2014

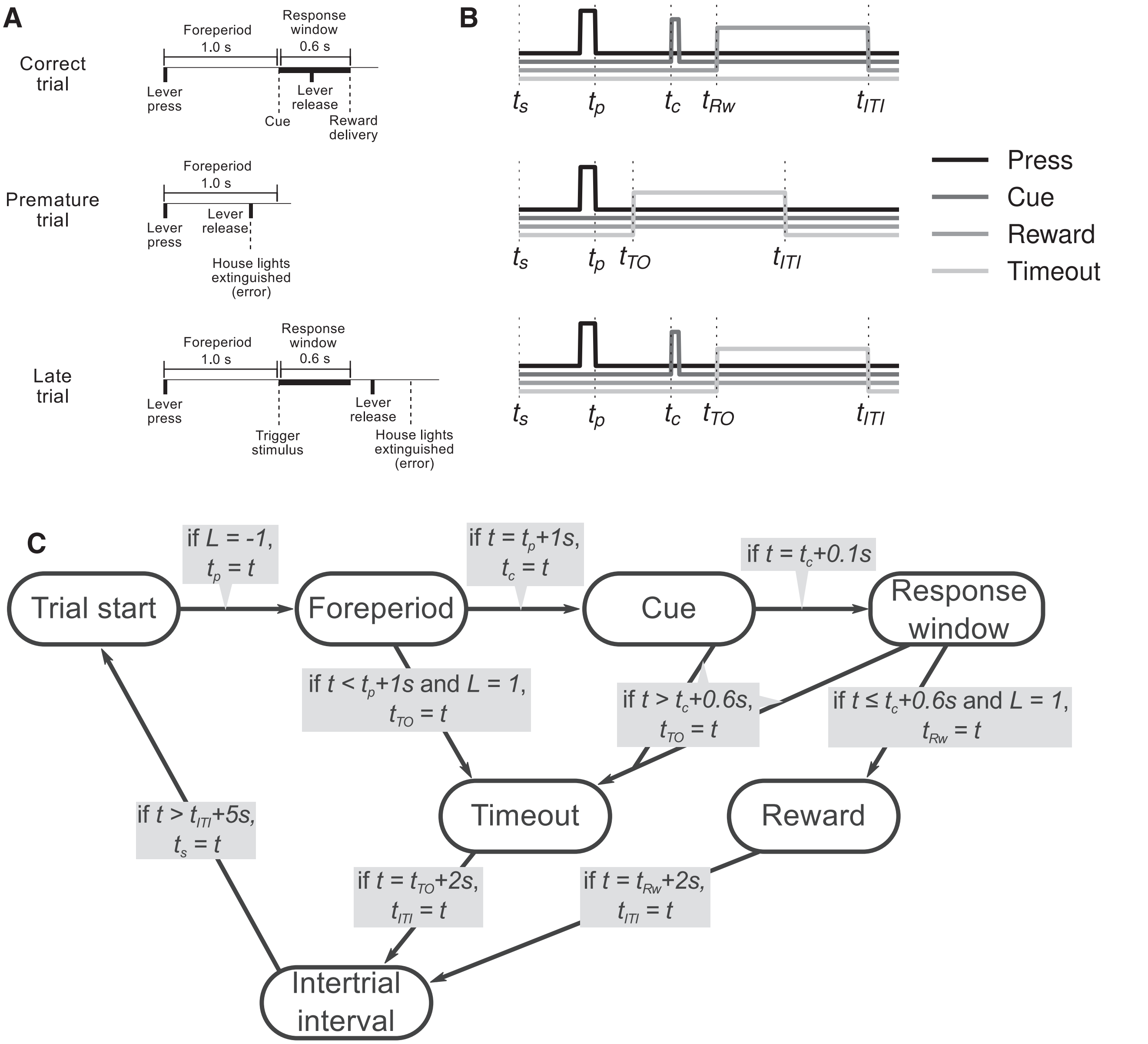
A
Correct trial
Foreperiod 1.0 s
Response window 0.6 s
Lever press
Lever release
Cue
Reward delivery
Premature trial
Foreperiod 1.0 s
Lever press
Lever release
House lights extinguished (error)
Late trial
Foreperiod 1.0 s
Response window 0.6 s
Lever press
Trigger stimulus
Lever release
House lights extinguished (error)
B
t_s t_p t_c t_{Rw} t_{ITI}
t_s t_p t_{TO} t_{ITI}
t_s t_p t_c t_{TO} t_{ITI}
Press
Cue
Reward
Timeout
C
Trial start
Foreperiod
Cue
Response window
Timeout
Reward
Intertrial interval
if $L = -1$, $t_p = t$
if $t = t_p+1s$, $t_c = t$
if $t = t_c+0.1s$
if $t < t_p+1s$ and $L = 1$, $t_{TO} = t$
if $t > t_c+0.6s$, $t_{TO} = t$
if $t \le t_c+0.6s$ and $L = 1$, $t_{Rw} = t$
if $t > t_{ITI}+5s$, $t_s = t$
if $t = t_{TO}+2s$, $t_{ITI} = t$
if $t = t_{Rw}+2s$, $t_{ITI} = t$

Figure 1. The simple reaction-time task used in the experiment. Trials are classified as correct, premature, or late depending on the time of lever release. The times at which important events occur in the trials are labeled. In order, t_s is the start of the trial, t_p is when the lever is fully pressed, t_c is the time of the auditory cue, t_{Rw} is when reward is delivered, t_{TO} is when house lights are extinguished, and t_{ITI} is the start of the intertrial interval. ***A***, Schematic of each trial type. ***B***, Results of simulating each trial type according to Equation 1. ***C***, Finite state machine tracking task state in the simulation.

closely matches the behavior of experimental subjects performing the same reaction-time task.

Materials and Methods

Reaction-time task

The experimental data analyzed in this study are the same as that in Narayanan and Laubach (2009), where more detailed explanation of the experimental methods can be found. In that study, 12 male Long–Evans rats were trained to press down a lever for a fixed amount of time (the "foreperiod;" 1 s for this study) using standard methods (Laubach et al., 2000). After the foreperiod, an auditory cue is presented. The subject then has 0.6 s (the response window) in which to release the lever. If the lever is released in that time, the trial is a success, and the subject is rewarded with water. If the lever is released during the foreperiod, the trial is classified as a "premature error," and is penalized by a timeout with the house lights extinguished. If the lever is not released within the response window, the trial is classified as a "late error," and is also penalized by a timeout with the house lights extinguished. Figure 1*A* presents a schematic view of the task.

Rats included in the study reached performance criterion of >60% correct trials in 9.75 ± 1.5 sessions (1748 ± 343 trials). Microwire electrode arrays were then implanted into the prelimbic region of the rat cerebral cortex, which is generally considered to be part of the medial prefrontal cortex (Laubach, 2011) and the anterior cingulate cortex in particular (Medalla and Barbas, 2009). Data were recorded using a Plexon Many Neuron Acquisition Program for three sessions (567 ± 98 trials).

Simulation. To provide appropriate environmental input for the networks described in subsequent sections, the same reaction-time task is simulated using a finite state machine. The task can be in one of seven states (S): trial start (TS), foreperiod (FP), timeout (TO), cue (C), response window (R), reward (Rw), and intertrial interval (ITI). State transitions can be seen in the graphical depiction of the finite state machine in Figure 1*C*.

The auditory cue, house lights, and reward signals are provided by the environment based on the current task state.

$$u_C = \begin{cases} 1 & \text{if } S = \text{C} \\ 0 & \text{otherwise} \end{cases}$$

$$u_{TO} = \begin{cases} 1 & \text{if } S = \text{TO} \\ 0 & \text{otherwise} \end{cases}$$

$$u_{Rw} = \begin{cases} 1 & \text{if } S = \text{Rw} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The lever position does not depend on task state, but instead on press (u) and release (u_r) signals provided by a control system (see Control networks).

$$L = \int [-u(t) + u_r(t)]\, dt. \tag{2}$$

Double-integrator network

A key finding of the Narayanan and Laubach (2009) study was that the temporal profiles of the leading principal component from the mPFC resembled the cumulative sum of the second component, and vice versa. This finding suggested that population activity in the mPFC might reflect a process of integration. Using the Nengo simulator (Bekolay et al., 2014), a number of candidate networks were investigated and one particular network was found to resemble the dynamics of the mPFC recordings. This network is called a "double-integrator network," and was first proposed in the context of working memory by Singh and Eliasmith (2006). We hypothesize that the double-integrator network proposed by Singh and Eliasmith (2006) will exhibit the same neural dynamics as the experimental mPFC when modified to respond to events occurring during the simple RT task.

To test this hypothesis, we extend the double-integrator network by providing it with multiple inputs that represent events occurring during the simple RT task (including those in Eq. 1), while maintaining the same underlying dynamics. By driving the double-integrator network to certain parts of the state space, we make it possible to predict the state of the system when the cue occurs during the reaction-time task.

Dynamics. The underlying dynamics of the double-integrator network are captured in the following equations:

$$\begin{aligned} \dot{x}_1 &= u \\ \dot{x}_2 &= \beta x_1, \end{aligned} \tag{3}$$

where u represents an input signal and β is a scalar parameter representing the strength of the connection between the two integrators. This results in a two-dimensional system in which x_1 integrates its input, maintaining the accumulated value over time, and x_2 integrates the value of x_1. This is the system proposed by Singh and Eliasmith (2006), who predicted that this dynamical system is capable of time tracking; because x_2 integrates x_1, the amount of time elapsed since x_1 changes can be determined based on how much x_2 has changed.

In Singh and Eliasmith (2006), neural recordings were analyzed during a single trial of the vibrotactile discrimination task. This corresponds to a single trial of the simple RT task in which the input, u, is the lever press, and the initial state, x, is (0, 0). However, in the simple RT task we perform many consecutive trials. We therefore use other inputs from the environment to match the trajectories tracked by the experimental principal components (Fig. 2).

Because the principal components resemble the double-integrator network in the postcorrect case, we aim to reset the system to the origin point after correct trials. Figure 2*D* indicates that the principal components return to their original state after ~2 s, which is the time at which the subject receives reward. Therefore, we hypothesize that the trial's outcome (i.e., reward delivery or a timeout) resets the system to some particular state, as proposed in Narayanan and Laubach (2009). In the case of reward delivery, we implement this hypothesis by using the reward delivery signal, u_{Rw}, to drive the system toward (0, 0).

$$\begin{aligned} \dot{x}_1 &= u - Ru_{Rw}\, x_1 \\ \dot{x}_2 &= \beta x_1 - Ru_{Rw}\, x_2, \end{aligned} \tag{4}$$

where R is a scalar parameter indicating how strongly the system is driven to the origin point.

In the posterror case, the principal components in Figure 2*D* are not at the origin point. Because the lever press increases x_1, when $x_1 > 0$ we can conclude that the subject is currently performing the task. We therefore hypothesize that $x_1 < 0$ indicates that an error has recently occurred. We implement this hypothesis by using the house lights signal, u_{TO}, to decrease x_1.

$$\begin{aligned} \dot{x}_1 &= u - Ru_{Rw}\, x_1 - Eu_{TO} \\ \dot{x}_2 &= \beta x_1 - Ru_{Rw}\, x_2, \end{aligned} \tag{5}$$

where E is a scalar parameter controlling how quickly x_1 decreases. Note that x_2 is not directly driven by u_{TO}; however, because of the βx_1 term, x_2 will also decrease.

Finally, Figure 2*D* shows that the principal components are unstable during the intertrial interval. To model trial-by-trial cycles of excitability associated with task engagement, a slow oscillation (~0.2 Hz) was applied to the networks. The final dynamics of the double integrator are as follows:

$$\begin{aligned} \dot{x}_1 &= u - Ru_{Rw}\, x_1 - Eu_{TO} + \sin(0.2 \cdot 2\pi) \\ \dot{x}_2 &= \beta\, x_1 - Ru_{Rw}\, x_2, \end{aligned} \tag{6}$$

where β, R, and E are positive scalars. The effects of changing these free parameters are explored below in Results.

Control networks

To evaluate the double-integrator network's ability to influence action timing, we created a feedforward control network that performs the simple RT task by releasing the lever when the cue arrives; we refer to this feedforward system as the cue-responding network. We connected the double-integrator network to the cue-responding network such that the lever can also be released by the double-integrator network; we refer to this combined system as the adaptive control network.

Cue-responding network. The cue-responding network mimics how an experimental subject would react to the cue if the time of the cue cannot be predicted. The trial starts after the intertrial interval ends (t_s), at which point the lever is pressed.

$$u(t) = \begin{cases} 1 & \text{if } t > t_s \text{ and } L > -1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The lever is released once the cue occurs at t_c.

$$u_r(t) = \begin{cases} 1 & \text{if } t > t_c \text{ and } L < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Adaptive control network. There is a fixed lower bound on reaction times in the cue-responding network based on the time taken to detect the cue and for u_r to integrate to 1; with direct simulation of Equation 8 there is no delay between cue onset and cue detection, but in a spiking neural implementation there are signal transmission delays between each neural connection. Because the goal is to release the lever within the response window, the cue-responding strategy may be too slow if there are too many intervening connections between the neural population detecting the cue, and the neural population driving the muscles that eventually effect lever release.

The double-integrator network described previously can be combined with the cue-responding network to implement a faster strategy because the state of the dynamical system being tracked by the double integrator (Eq. 6) is approximately the same on each postcorrect trial. If the previous trial was a correct trial, then the double integrator's state approaches (1, 1) as the time of the cue approaches, for some values of β. We therefore add an additional term that effects lever release if the predicted cue time is approaching.

$$u_r = u_r + \frac{1}{1 + e^{-a(x_2 - b)}} \tag{9}$$

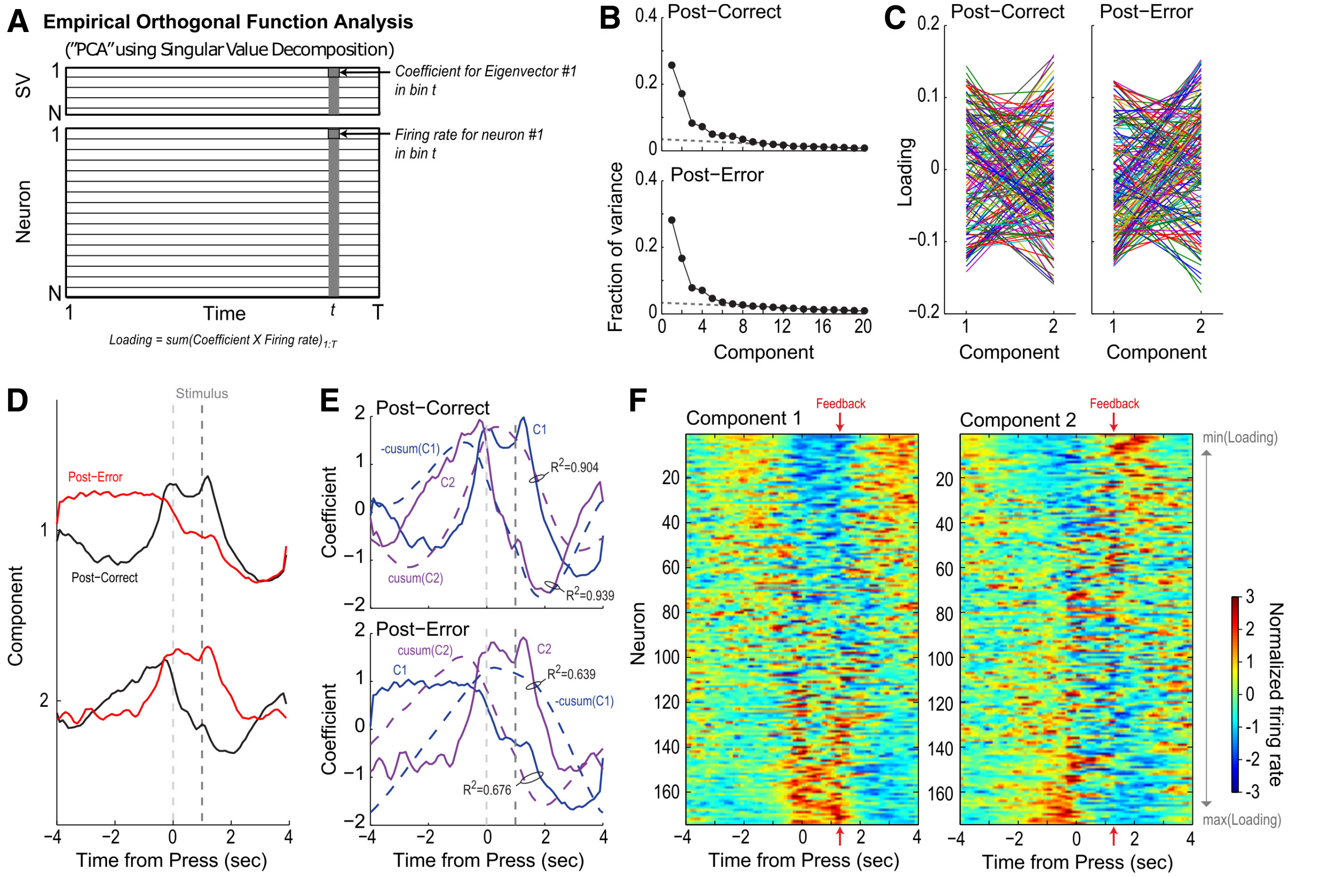

Figure 2. A summary of the principal component analysis done in Narayanan and Laubach (2009). ***A***, A graphical summary of principal component analysis. Normalized perievent neural data, in the form of the Z-scores of instantaneous firing rates, is organized in a matrix, with each row being the Z-scored firing rates of a single neuron on a single trial. Singular value decomposition is performed on the matrix, resulting in a matrix such that the number of columns (time bins) is the same. The rows are now ordered such that the first row contains largest eigenvector, which represents the value when the original data are projected onto the axes of highest variance. ***B***, The fraction of variance explained by each singular value when PCA is performed on the postcorrect neural data (top), and the posterror data (bottom). In both cases, the two eigenvectors with the highest singular values account for nearly 50% of variance. The next two eigenvectors also account for an amount of variance higher than would be expected when we linearly interpolate from the smaller singular values. ***C***, A summary of the loadings of the two top eigenvector on each neuron. If a neuron were encoding both the first and second eigenvectors to the same degree, we would expect a horizontal line. The varied lines crisscrossing each other suggest that each individual neuron has different sensitivities to the first and second principal components. ***D***, The top two principal components for the postcorrect trials (black) and the posterror trials (red). ***E***, Top, Plotting the cumulative sum (integral) of the first principal component closely matches the second principal component, and vice-versa (Pearson $R^2 = 0.904$ and $R^2 = 0.939$, respectively), in the postcorrect case. Bottom, In the posterror case, the first two principal components are no longer cumulative sums of each other (Pearson $R^2 = 0.639$ and $R^2 = 0.676$). This points to neural integration as a potential mechanism explaining mPFC activity. ***F***, Normalized spike-density functions for all of the neurons analyzed in the postcorrect case, organized by the loading (left) on the first principal component, and (right) on the second principal component.

The new term defines a sigmoid that increases from 0 to 1 when $x_2 \approx b$. The slope of the smooth increase from 0 to 1 is controlled by a; for very large a, the sigmoid is essentially a step function. We choose $a = 20$ and $b = 0.9$, which corresponds to a sigmoid that starts increasing when $x_2 \approx 0.6$ and reaches 1 when $x_2 \approx 1.2$.

If the previous trial was an error trial, the double integrator will follow a different trajectory in which $1/1 + e^{-a(x_2 - b)} \approx 0$ throughout the trial, and therefore the control system will only release the lever once the cue occurs. This allows the adaptive control network to adopt an aggressive prediction strategy if the last trial was successful, and to adopt a conservative cue-response strategy if the last trial was not successful.

Neural Engineering Framework

The previous sections describe a dynamical system hypothesis of the mPFC and surrounding areas of cortex that together perform the simple reaction-time task adaptively depending on the outcome of the previous trial. Although these can be numerically simulated to observe ideal system behavior, we construct a spiking neural network implementing the above equations to test the hypothesis in a biologically plausible setting that approximates the direct numerical simulation. Testing this neural implementation allows us to determine whether the approximation is able to perform the task in a manner comparable to the real brain by producing simulated neural data that can be compared directly with experimental neural data. We construct spiking neural networks using the principles of the Neural Engineering Framework (NEF; Eliasmith and Anderson, 2003) in the Nengo simulation environment (Bekolay et al., 2014). Using the NEF, each signal from the dynamical systems above is represented with a population of spiking neurons, and is transformed through the connections between those populations (see Fig. 3).

The NEF enables the simulation of dynamical systems by making the assumption that populations of spiking neurons represent real-valued vectors, and using a least-squares optimal method to compute connection weights that transform those representations through the connections between populations of neurons. A population can have recurrent connections to produce dynamics, such as the integrative populations used in this study.

The NEF's representation scheme is a kind of population coding (Georgopoulos et al., 1986; Salinas and Abbott, 1994), extended to n-dimensional vector spaces. Each neuron in a population is sensitive to a particular direction, called the neuron's encoder, denoted **e**. The activity of a neuron can be expressed as

$$a = G[J(\mathbf{x})], \quad J(\mathbf{x}) = \alpha \mathbf{e} \cdot \mathbf{x} + J_{bias} + J_{\eta}, \qquad (10)$$

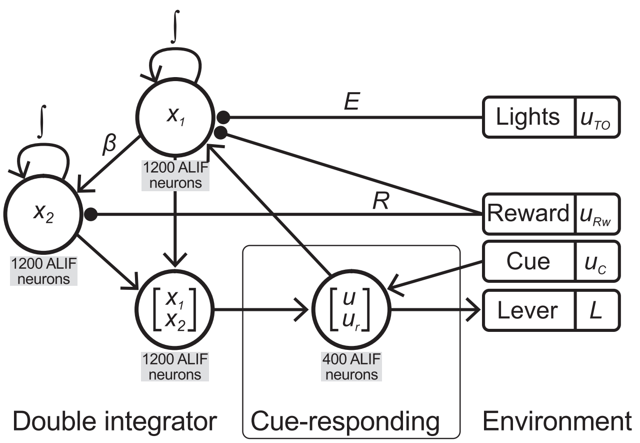

Figure 3. Block diagram of the spiking neural implementation of the adaptive control network. Circles represent populations of adaptive leaky integrate-and-fire neurons. Left, The adaptive control model contains the double-integrator network from Singh and Eliasmith (2006). It has been modified by accepting both press information (u) and error information ($E\, u_{TO}$) to drive x_1, and reward information ($R\, u_{Rw}$) to control the integration of both x_1 and x_2. Lines terminating with circles indicate negative connections; lines terminating with arrowheads indicate positive connections. The output of the double integrator is the only component that interacts with the cue-responding network. Middle, The cue-responding network alone is able to perform the simple reaction-time task by responding to the Cue (u_C) provided by the environment (see Materials and Methods, Cue-responding network). The influence of the double-integrator network allows the cue-responding network to respond adaptively based on the outcome of the previous trial. Right, Values that are not modeled with spiking neural populations, and are therefore considered signals that are provided by the environment. The lever is modified by the decoded output of the cue-responding network.

where $G[\cdot]$ is a spiking nonlinear neural activation function, α is a scaling factor (gain) associated with the neuron, $\mathbf{e}$ is the neuron's encoder, $\mathbf{x}$ is the vector to be encoded, J_{bias} is the background current of the cell when $\mathbf{x} = 0$, and J_η is noise current injected to match experimental neural variability. The encoded value, $\mathbf{x}$, can be estimated linearly:

$$\hat{\mathbf{x}}(t) = \sum_i \mathbf{d}_i a_i(t), \tag{11}$$

where a_i is the activity of neuron i and $\mathbf{d}_i$ is a decoding weight determined by solving a least-squares minimization of the difference between the decoded estimate and the actual encoded vector.

The least-squares minimization is solved with the following equations. $\mathbf{X}$ is a set of samples from the representational range of $\mathbf{x}$ (e.g., samples from $U(-1, 1)$ in a typical scalar case).

$$\mathbf{A} = \begin{bmatrix} a_0(\mathbf{X}) \\ a_1(\mathbf{X}) \\ \vdots \\ a_n(\mathbf{X}) \end{bmatrix} \tag{12}$$
$$\mathbf{d} = \mathbf{\Gamma}^{-1}\mathbf{Y}, \text{ where } \mathbf{\Gamma} = \mathbf{A}\mathbf{A}^{\mathrm{T}} \text{ and } \mathbf{Y} = \mathbf{A}\mathbf{X}^{\mathrm{T}}.$$

Neural activity is interpreted as a filtered spike train.

$$a_i(t) = \sum_s h(t - t_s) = \sum_s e^{-(t-t_s)/\tau_{PSC}}, \tag{13}$$

where $h(\cdot)$ is an exponential filter modeling postsynaptic current that is applied to each spike, and s is the set of all spikes occurring before the current time t.

Connection weights implementing a linear transform of the encoded vectors of a population can be computed as

$$\omega_{ij} = \alpha_j \mathbf{e}_j \mathbf{L} \mathbf{d}_i, \tag{14}$$

where i indexes the input population, j indexes the output population, and $\mathbf{L}$ is a linear operator.

Table 1. Parameters used in the neural implementation of the double integrator and control networks

Maximum firing rate	J_η	τ^{RC}	τ_{ref}	τ_{adapt}	G_{inc}
U(10 Hz, 50 Hz)	$U(-0.2\text{ nA}, 0.2\text{ nA})$	20 ms	1 ms	10 ms	U(0.001, 0.02)

Maximum firing rate and J_η are ensemble-level parameters as they apply to any neuron model $G[\cdot]$. The maximum firing rate is used to determine the gain (α) and bias currents (J_{bias}) associated with neuron (used in Eq. 10). All other parameters apply specifically to the ALIF neuron model.

Nonlinear transformations can be computed by solving for decoders that minimize the difference between the decoded estimate and a function of the encoded vector; i.e., $\varepsilon = \mathbf{A}f(\mathbf{X}^{\mathrm{T}})$ in Equation 12.

Dynamical systems of the form $\dot{x} = Ax + Bu$ can be implemented by connecting a population to itself with the linear operator $\mathbf{L} = A$ on the recurrent connection, and receiving input u from other populations and setting $\mathbf{L} = B$ on those connections.

The above principles are not dependent on a particular neuron model, $G[\cdot]$. In this study, we use the adaptive leaky integrate-and-fire (ALIF) model (Koch, 1999) to match Singh and Eliasmith (2006). This model is governed by the equations:

$$\begin{aligned} \frac{dV}{dt} &= -\frac{V(1 + RG_{adapt}) - J(\mathbf{x})R}{\tau^{RC}} \\ \frac{dG_{adapt}}{dt} &= -\frac{G_{adapt}}{\tau_{adapt}} \\ &\text{if } V > V_{th},\ G_{adapt} = G_{adapt} + G_{inc}, \end{aligned} \tag{15}$$

where R is the leak resistance, τ^{RC} is the RC time constant, and G_{adapt} tracks how much the conductance is modulated over time. G_{inc} and τ_{adapt} are parameters affecting G_{adapt}.

For each instance of each spiking neural network, the parameters of each ALIF neuron are randomly selected from a biologically plausible range (e.g., selecting the maximum firing rate from the distribution U(10 Hz, 50 Hz) produces average firing rates that match Narayanan and Laubach, 2009). The encoders, $\mathbf{e}_i$, are randomly selected from a uniform distribution of directions in the encoded vector space. Table 1 summarizes the parameters specific to the neural implementation. All other parameters are described in the Double-integrator network and Control network sections.

Data analysis

Both experimental and simulated experiments produced data at the neural and behavioral levels. The analyses below were done on both experimental and simulated data, unless otherwise noted.

Neural data. Experimental spike trains from Long–Evans rats were determined by online identification with an oscilloscope and audio monitor, and offline spike sorting using Plexon software. Spike sorting was based on principal component analysis (PCA; described below) and waveform shape. In all, 174 single units were identified in rodent mPFC and analyzed together (Narayanan and Laubach, 2009).

Simulated spike trains were generated by the Nengo simulator. The populations representing signals in the double integrator and control networks contained 1200 adaptive leaky integrate-and-fire neurons (Fig. 3); 174 of the neurons in an analyzed population were randomly selected to match the number of units from the experimental study.

Spike trains are analyzed with PCA, following procedures used by Narayanan and Laubach (2009) to characterize common firing patterns in the mPFC during the task. In the PCA, only neurons with an average firing rate >1 Hz are considered. Spike trains from 4 s before and 4 s after each press event are isolated. The perievent spike train is binned in 1 ms bins and convolved with a Gaussian filter (σ = 25 ms; PCs are consistent with many other σ values). These spike density functions are normalized to Z-scores, and then averaged over all trials to produce a matrix in which each row is the normalized average response of a neuron 4 s before and after a lever press. Singular value decomposition is performed on that matrix to isolate the principal components, which are normalized to Z-scores. The amount of variance accounted for by each principal component is computed as the square of the component's eigenvalue over the sum of all squared eigenvalues ($\mathbf{s}_i^2/\Upsilon\mathbf{s}^2$).

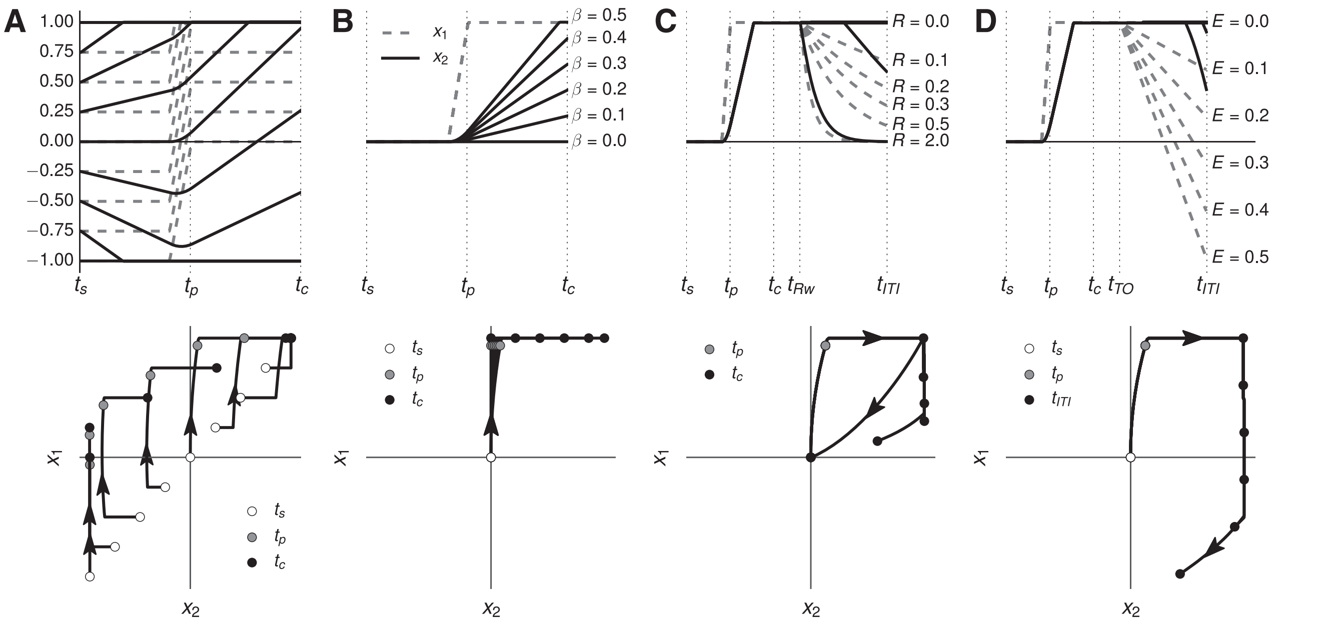

Figure 4. Time tracking in the double-integrator model. x_2 (solid black line) ramps up over time, enabling the network to predict the time of the cue. Time labels indicate important events during each trial. t_s is the start of the trial, t_p is when the lever is fully pressed, t_c is the time of the auditory cue, t_{Rw} is when reward is delivered, t_{TO} is when house lights are extinguished, and t_{ITI} is the start of the intertrial interval. ***A***, The state of the system at t_c depends on the initial state of the system at the start of the trial, t_p. The initial state depends on the outcome of the previous trial; after a correct trial the system should start at (0, 0), and after a premature or late trial, the system should start at (−1, −1). ***B***, The slope of the ramping signal x_2 depends on the parameter β, which represents the strength of the connection between x_1 and x_2. An ideal β pushes x_2 far from the initial state, but within the representational range of a population of spiking neurons, which is normalized to (−1, 1); therefore, the system should approach (1, 1) at the time of the cue, which occurs when $\beta \approx 0.44$. ***C***, The system is reset to (0, 0) after a correct trial by reward feedback. The parameter R controls the strength of the reward feedback; the system is properly reset if $R \geq 2$. ***D***, The system is driven near (−1, −1) after an error trial by error feedback. The parameter E controls the strength of the error feedback; the system is driven near (−1, −1) if $E \geq 1$.

As previously described by Narayanan and Laubach (2009) and summarized in Figure 2, the first two principal components of the experimental data are modulated around the lever press and appear to track whether the subject has pressed the lever and the relative amount of time that the lever has been pressed. Narayanan and Laubach (2009) also found that the neural activity in the mPFC changed significantly depending on the outcome of the previous trial; the first two principal components change significantly after errors, and appear to contain information that an error occurred. This information would be necessary to adapt behavior based on the last trial.

Interestingly, in the postcorrect case, the first principal component is highly correlated with the integral (i.e., cumulative sum) of the second principal component during the postcorrect trial (Fig. 2*E*). Because the first two principal components account for over half of the variance in the neural data (Fig. 2*B*), we hypothesize that the mPFC represents the task state (PC1) and the integral of the task state (PC2). Implementing integration in a population of neurons has been widely explored (Seung, 1996; Shadlen and Newsome, 2001; Wang, 2002; Eliasmith, 2005). In the posterror case, these correlations are significantly weaker, indicating that the integration taking place in the mPFC depends on prior outcomes. The firing patterns of individual neurons, related to the two leading PCs, are shown for postcorrect trials in Figure 2*F*. Nearly equal fractions of neurons became more or less active at times when the PCs were maximally positive or negative. In these plots, neural activity is synchronized to the start of the trial (time 0). The cue was presented at 1 s. The animals were required to respond to the cue with a reaction time <0.6 s, and feedback about the outcome was given ~0.1 s later. The range of feedback times are indicated near the top right corner of each plot. Importantly, the trials used for this analysis, and shown in Figure 2, were all correct trials. The only difference between trial types was the outcome of the previous trial (correct or incorrect). As in Narayanan and Laubach (2008), neural activity did not track the type of error that occurred (premature or late). Population activity was not analyzed for these subtypes of errors, as they were present in unequal fractions of trials over rats and were much less frequent than the previously correct trials.

Table 2. Parameters used to simulate the dynamical system shown in Equation 5

β	R	E
0.44	2	0.5

Appropriate values were found through simulation (Fig. 4).

Behavioral data. In both the experimental and theoretical studies, we record five relevant event times: lever press (t_p), lever release (t_r), reward delivery (t_{Rw}), cue onset (t_c), and house lights extinguishing (t_{TO}). A correct trial is identified by the sequence lever press, cue onset, lever release, reward delivery; a premature trial is identified by lever press, lever release, house lights extinguishing; and a late trial is identified by lever press, cue onset, house lights extinguishing. Trial outcomes are identified from the overall sequence of events by filtering for these sequences. Reaction times are calculated only for correct trials, and are defined as the difference between cue onset and lever release.

Results

Theoretical results

Here we describe the results of simulating the dynamical systems in the Double-integrator network and Control network sections directly. This provides a general characterization of the expected low-dimensional dynamics of the high-dimensional neural simulation.

The double integrator can predict cue time

The purpose of the double integrator is to predict the time at which the cue occurs. Figure 4 shows that this is possible. The parameter β controls how fast the second dimension of the double integrator state (x_2) increases (Eq. 5), resulting in different states at t_c (Fig. 4*B*); because this parameter is encoded in the strength of the connection between the two integrators, this is a natural target for learning after errors. Given a particular value for β, the state of the system at the time of the cue, t_c, can be predicted given the initial conditions of the double integrator at the start of the trial; values at t_c for $\beta = 0.44$ are shown in Figure 4*A*.

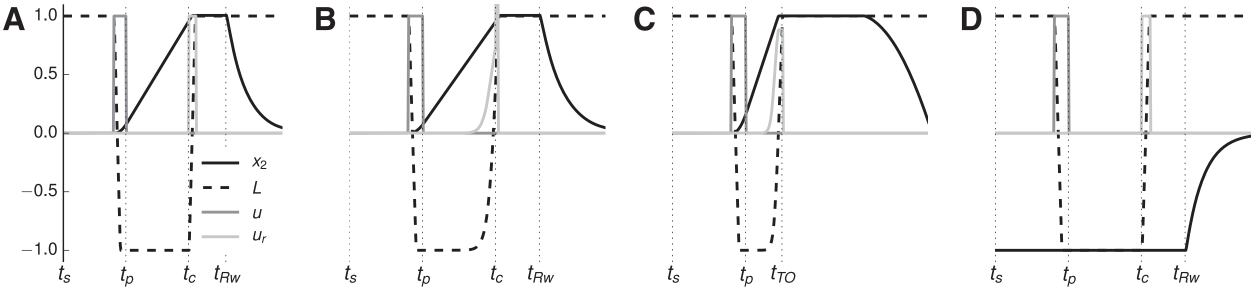

Figure 5. Results of control network simulations. Time labels indicate important events during each trial. t_s is the start of the trial, t_p is when the lever is fully pressed, t_c is the time of the auditory cue, t_{Rw} is when reward is delivered, and t_{TO} is when house lights are extinguished. ***A***, The cue-responding network. The release, u_r, is initiated only after cue onset, t_c. ***B***, The adaptive control network in the ideal case. The release is initiated such that the lever (dashed black line) is raised at the time of cue onset. ***C***, A premature release in the adaptive control network. The release is initiated before the time of cue onset (premature release is emulated by setting $\beta = 0.94$). ***D***, The adaptive control network follows the cue-responding strategy after error trials, due to the initial state of the double integrator.

The state of the system at the start of a trial (i.e., the initial conditions of the dynamical system) depend on the outcome of the previous trial. Equation 5 describes how the state changes based on the two possible outcomes, reward or light extinguishing. For sufficiently high values of R (Fig. 4*C*), the system starts the next trial near the origin, resulting in the system predicting the time of the cue, and therefore using the adaptive strategy. For sufficiently high values of E (Fig. 4*D*), the system starts the next trial near $(-1, -1)$, resulting in the system being unable to predict the time of the cue (Fig. 4*A*), and therefore using the cue-responding strategy. Note that correct and error trials can be distinguished, and therefore used to modify behavior and drive learning; however, premature trials and late trials cannot be distinguished on the next trial, which is consistent with the results of Narayanan and Laubach (2009).

Appropriate parameter values were determined using these simulations and used for all subsequent simulations, unless otherwise noted. The values used are listed in Table 2.

The adaptive control network can react faster

Equation 9 defines a signal that causes the network to release the lever when the predicted cue approaches. Figure 5 shows a representative correct trial from the cue-responding and adaptive control networks, as well as a trial in which the cue was mispredicted, and the subsequent corrected trials following the error. Importantly, Figure 5*B* shows the adaptive control network can release the lever at the exact time of the cue.

The double-integrator state can drive learning

The state of the system reflects the outcome of the previous trial, and therefore can affect behavior on the next trial. The system can also be used to drive learning to maximize long-term reward. Although the proposed system does not employ learning, previous work has proposed that the mPFC drives learning of action outcomes. Specifically, Alexander and Brown, 2011 have proposed the PRO model, in which the mPFC tracks the predicted values of action outcomes to compute prediction errors that drive learning. Although the signals tracked by the double-integrator model exist to modify action timing, the signals used to drive learning in PRO can be computed from the double-integrator model as well, as shown in Figure 6. The only difference that cannot be attributed to the task is that the state is tracked throughout the trial in the PRO model, whereas in the double integrator, reward turns off the integrator responsible for tracking task state. Despite this difference, the similarity in signals tracked by the two models suggests that the double integrator, which we use to adaptively control action timing, can also be used to drive learning.

Simulation results

The above theoretical results suggest that the underlying low-dimensional dynamics of the postulated network should be appropriate for capturing the qualitative features of mPFC activity. We now examine both the qualitative and quantitative fit of the high-dimensional spiking network to the recorded empirical data.

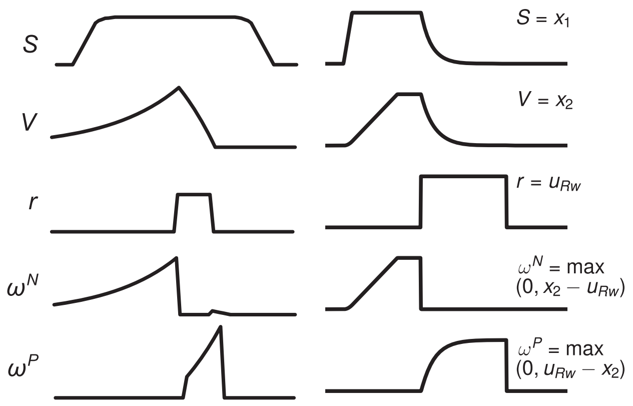

Figure 6. Left, Signals tracked by the PRO model (Alexander and Brown, 2011). S refers to task state; V refers to the value of an action-outcome; r refers to the actual reward; ω^N is a negative prediction error; and ω^P is a positive prediction error. Right, Signals tracked by or computed from the double-integrator network. In all but one case, the signals are similar, and differ mainly in task-related properties (e.g., the length and magnitude of the reward signal). In both models, all signals can be represented as vectors if appropriate for the task. The one divergent case is the state; in the PRO model, the state signal persists after the reward has been delivered, but in the double-integrator network is turned off by the reward.

The spiking double integrator has the same principal components as experimental data

Figure 7 compares the two leading principal components of the spiking implementation of the double-integrator network to those of the experimental data. The simulated principal components are very similar to the experimental principal components (Narayanan and Laubach, 2009): Pearson $R^2 = 0.82, 0.97, 0.95$ for the first component and $R^2 = 0.91, 0.81, 0.84$ for the second components in the postcorrect, postpremature, and postlate trials, respectively.

A single integrator does not have the same principal components as experimental data

We also performed principal component analysis on neural populations representing only one of the two dimensions tracked by the double-integrator network. Figure 8 shows the results of the analysis for the population tracking x_1. Whereas the first principal component closely matches the experimental data (Pearson $R^2 = 0.54, 0.86, 0.80$ for post-correct, postpremature, and postlate trials, respectively), the second principal component does not (Pearson $R^2 = 0.15, 0.26, 0.30$). This suggests that there exist neurons that are sensitive to both dimensions of the double integrator.

Spiking networks can predict the cue

Figure 9 shows that the decoded values of spiking neural networks implementing the cue-responding and adaptive control networks closely resemble

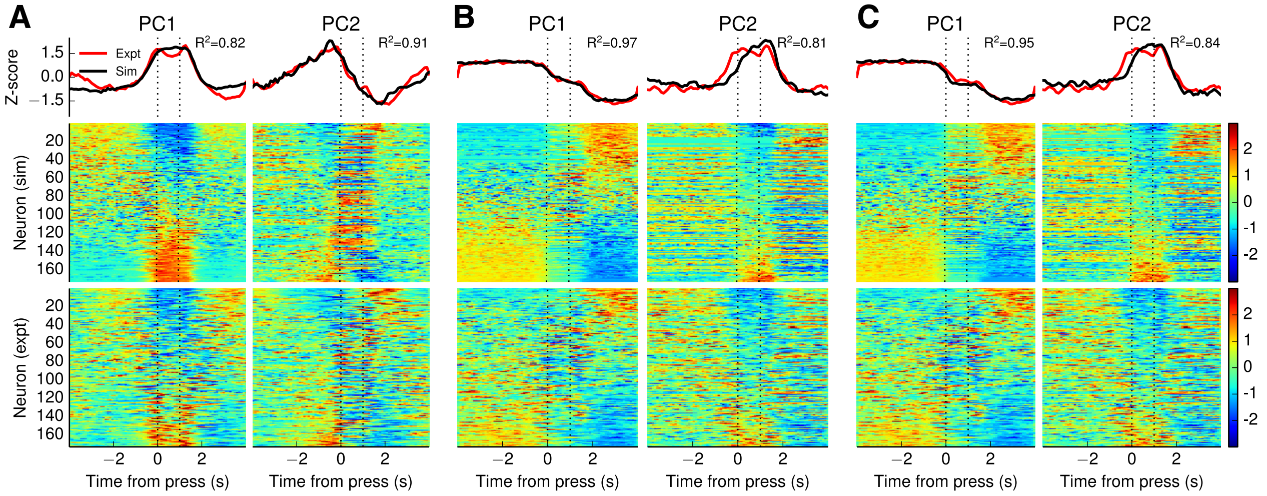

Figure 7. Results of principal component analysis on the data generated by the double-integrator model for correct presses in (***A***) postcorrect trials, (***B***) postpremature trials, and (***C***) postlate trials. The top row compares the top two principal components of the model to those of the experimental mPFC data. In all cases, the PCs are significantly similar (Pearson $R^2 > 0.8$). The middle row shows normalized peri-event spike densities for 174 randomly sampled neurons from the simulated double-integrator model. The bottom row shows normalized peri-event spike densities for all 174 neurons recorded during the experimental study. The patterns of sensitivity compared in the middle and bottom rows are similar, although the recorded neuron responses appear more variable over time.

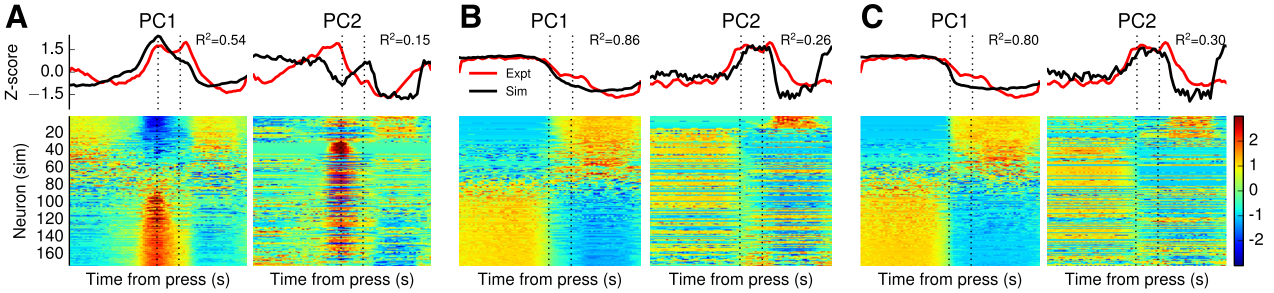

Figure 8. Results of PCA on the data generated by the integrator tracking x_1 in the double-integrator model for correct presses in (***A***) postcorrect trials, (***B***) postpremature trials, and (***C***) postlate trials. The top row of panels compares the top two principal components of the model to those of the experimental mPFC data. In all cases, the first PC is significantly similar (Pearson $R^2 > 0.5$), but the second PC is not (Pearson $R^2 \leq 0.3$). The bottom row shows normalized peri-event spike densities for 174 randomly sampled neurons from the simulated integrator population.

the signals directly simulated with the dynamical system in correct, premature, and late trials. Despite the introduction of the oscillation and the additional injected noise current, the cue can be predicted, and the outcome of the previous trial has a significant effect on the system state during a trial. The spiking networks are capable of the adaptive control described in the theoretical results, as can be seen in Figure 9*A*.

Adaptive control network matches observed performance

Figure 10 shows the performance of the experimental subjects, the spiking cue-responding network, and the full spiking adaptive control network. The lengths of the simulations were varied to match the number of trials performed by subjects. The mean of the median reaction times is 272 ± 48 ms for the experimental subjects, 365 ± 24 ms for the cue-responding models, and 240 ± 62 ms for the adaptive control model. The mean performance (percentage correct trials) is 70.8% for experimental subjects, 98.6% for cue-responding models, and 92.5% for adaptive control models. Therefore, only the adaptive control model has statistically indistinguishable reaction times compared with the experimental subjects. The cue-responding network is significantly slower, but makes fewer errors than the adaptive control network. Both simulated models make fewer errors than experimental subjects.

Discussion

Our results show that a spiking implementation of the double-integrator network (Singh and Eliasmith, 2006) can reproduce the dynamics of population activity in the rat mPFC (Narayanan and Laubach, 2009) and can be used to modify a feedforward control system to adaptively control the timing of actions. The ability to predict the time of the cue produces faster reaction times at the cost of additional premature errors.

An interesting feature of the model is that there is significant randomness involved in model generation (see Materials and Methods, Neural Engineering Framework). This results in significant variability in the performance of simulated subjects, despite all performing the task well (Fig. 10). Also notable is that the model does not employ learning: the weights determined before the simulation do not change. The ability to keep track of the state of the system over time through recurrent connections is what enables the model to adapt. By defining the dynamics of other tasks and implementing them in recurrently connected neural networks, we can start to form general theories of how biological systems can learn to perform complex tasks.

Relation to other models

In this study, we make an adaptive control hypothesis of mPFC function in reaction-time tasks. This hypothesis is compatible with previous hypotheses suggesting that the mPFC plays a role in the regulation of reinforcement learning parameters. One such example is the PRO model by Alexander and Brown (2011) (although these arguments also apply to related models, such as Khamassi et al., 2013). The PRO model proposes that the mPFC is involved in a form of temporal difference learning. As shown in

Figure 6, the positive and negative reward prediction error computed by PRO can also be computed by the double integrator, assuming that the value function learned by the system steadily increases.

In our model, we interpret the steadily increasing signal as representing time elapsed rather than value over time; this means that there is only one parameter that must be learned (β, the speed at which x_2 increases), rather than learning the value at every timestep in the simulation. In addition to being an easier target for learning, we further suggest that time tracking is a more natural interpretation for such a ramping signal. Theoretically, it is more likely that the value signal associated with those prediction errors is constant over time rather than steadily increasing (Niv and Schoenbaum, 2008). We suggest that the ramping signal shown in Alexander and Brown (2011) might reflect the temporal dynamics of the task instead of serving as a value signal.

The model presented in this study keeps track of the results of the previous trial, and therefore can only learn on a trial-by-trial basis. Some quantities like volatility, which is thought to be encoded in the mPFC (Behrens et al., 2007), would require additional integrative populations to aggregate information across many trials. Although we have not presented such a model in this study, we believe that a similar dynamical approach can be used to construct models that encode volatility and other quantities in spiking neural networks, which can be directly compared with neural recordings.

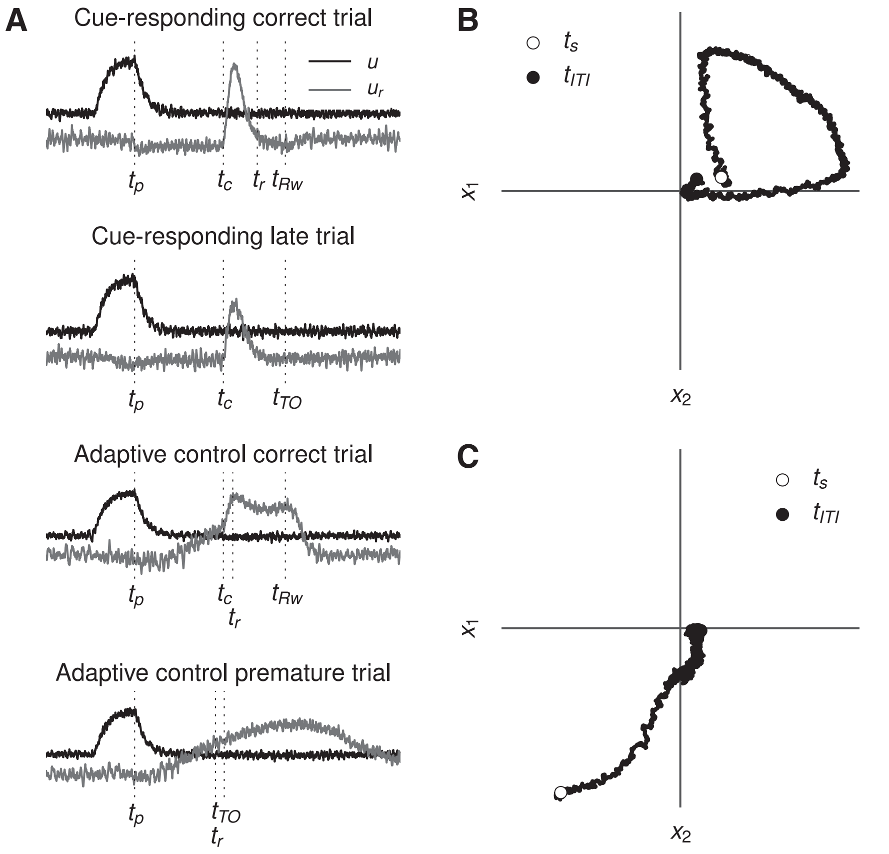

Figure 9. ***A***, Decoded values of representative trials of the control networks simulated with spiking neurons. From top to bottom, the trials shown are a correct trial in the cue-responding network, a late trial in the cue-responding network, a correct trial in the adaptive control network, and a premature trial in the adaptive control network. Time labels indicate important events during each trial. t_p is when the lever is fully pressed, t_c is the time of the auditory cue, t_r is the time of lever release, t_{Rw} is when reward is delivered, and t_{TO} is when house lights are extinguished. ***B***, The decoded value of the double-integrator network that is embedded in the adaptive control network on a correct trial following a correct trial. The system starts (t_s) and finishes (t_{ITI}) the trial near the origin point. ***C***, The decoded value of the double-integrator network on a correct trial following an error trial. The system starts near the bottom left portion of the state space, but because the trial was correct, finishes near the origin point.

Adaptation and learning

A primary difference between the adaptive control network we have presented and previous hypotheses of mPFC function is that we do not employ learning to change connection weights during a simulation. Instead, we achieve adaptive behavior through recurrently connected integrative populations that maintain information across trials. We believe that this type of adaptability should be explored in more depth in mPFC models.

We are able to construct a spiking neural network that performs the simple RT task because we have made a hypothesis of the dynamics governing the task. One issue that we have not attempted to address in this work is how those dynamics could be learned from the sensory information available during the task. We believe that this learning procedure is central to understanding the mPFC, and the brain in general. However, to characterize this learning procedure, we must first have examples of the endpoints of the learning procedure. We believe that the adaptive control network, which uses the double integrator to predict cue times, is an example of one possible endpoint. By understanding the dynamics of other tasks and how those dynamics might be implemented in spiking neural networks, we can begin to form theories of how spiking neural networks performing complex tasks might be learned. In previous work, we demonstrated that a spike-based learning rule can generate a neural integrator like those used in this study (MacNeil and Eliasmith, 2011), but learning the entire network described here is the subject of future work.

Limitations

Our model is specific to the simple reaction-time task being performed by the experimental subjects, and therefore can only be directly applied to experimental data of subjects performing this task. However, we believe that the methods used to determine the dynamical system governing the task and to implement the dynamical system in a spiking neural network are general, and can be used to explain many other tasks. Determining the dynamics of other tasks in a dynamical systems framework can help identify general principles of neural function.

Even in the context of the simple reaction-time task, our model cannot be directly compared with all of the experimental data available. We have chosen to implement the model in a spiking neural network, as we believe that it provides the most direct comparison to the data recorded from the biological system. However, because the mappings from spiking activity to EEG and fMRI recordings are not well understood, we cannot directly explain the wealth of data from humans performing similar reaction-time tasks. Previous studies of mPFC function in

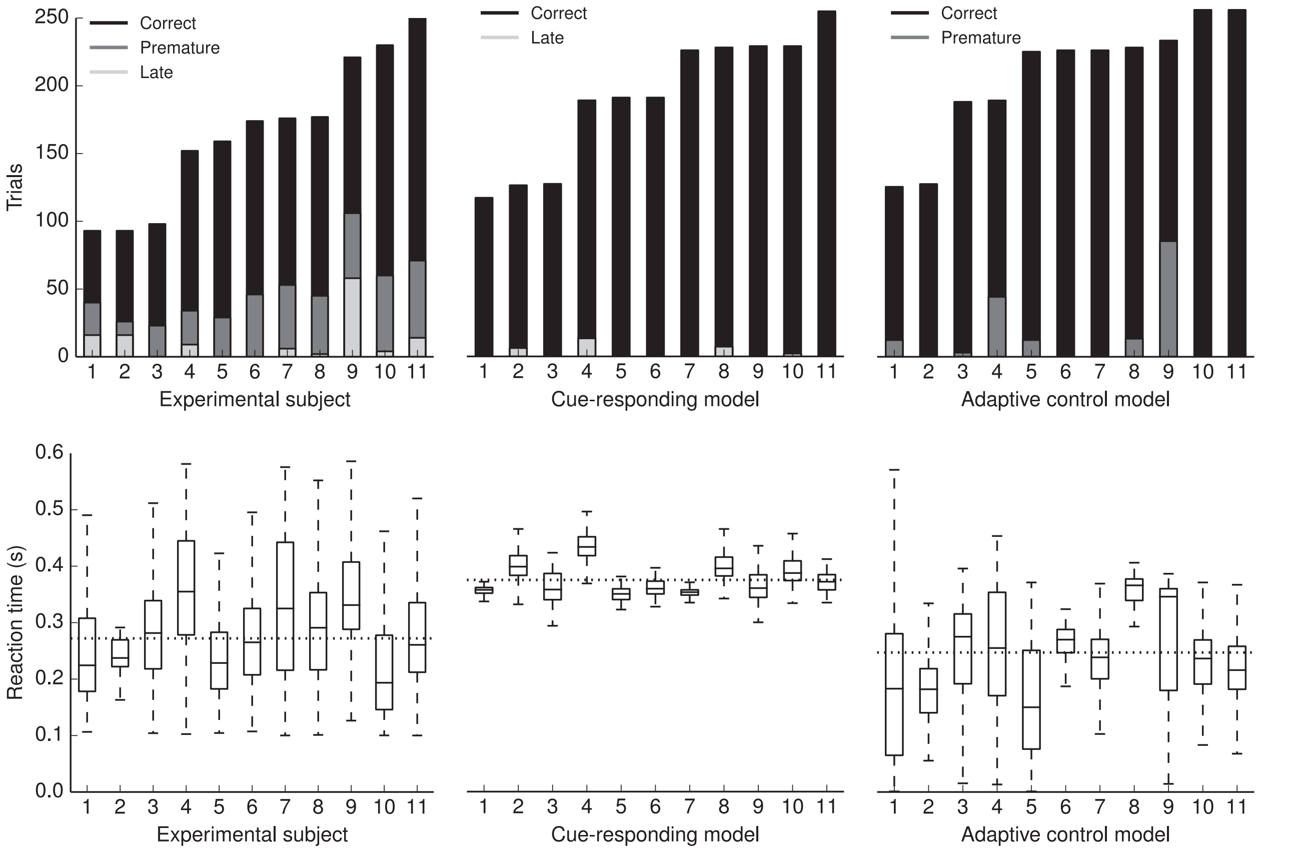

Figure 10. Behavioral performance of the experimental subjects and the cue-responding and adaptive control models. The top row summarizes the number of correct, premature, and late trials for each experimental or simulated subject. The number of total trials for the simulations was approximately matched to the experimental subjects. The bottom row summarizes the reaction times on correct trials for each subject. The dotted black line represents the mean of all subjects' median reaction times (272 ± 48 ms for the experimental subjects, 365 ± 24 ms for the cue-responding models, and 240 ± 62 ms for the adaptive control models).

humans have found evidence for low-frequency rhythmic activity being generated when mistakes are made and on trials after such errors (Cavanagh et al., 2009). A recent study reported the same types of error-encoding cortical rhythms in rat mPFC (Narayanan et al., 2013) and showed that mPFC activity constrains rhythmic activity in motor cortex. It has been suggested that the expression of these rhythms reflects alterations in cross-laminar processing (Carracedo et al., 2013). These effects could be examined by using specific models for different cell types and laminar patterns for connecting neurons together.

Predictions

Unlike previous mPFC models, we predict that mPFC inactivation negatively impacts behavioral performance of tasks that require precisely timed actions, in addition to disrupting the learning of such tasks.

At the neural level, the spiking implementation of the network predicts that neurons in the mPFC will ramp up in activity over the course of a goal-directed task. However, we predict that those ramping signals are an attempt to encode the temporal dynamics of the task, not an attempt to encode action-outcome values. This prediction could be tested by recording neural ensembles in the mPFC and motor cortex during learning. One important study would be to record the mPFC of naive rats, as was done in the motor cortex in Laubach et al. (2000). In that earlier study, a major limitation was that the rats only experienced brief foreperiods (400–600 ms) before the trigger stimulus, and ramping activity was not apparent in the PCA-style analyses (M. Laubach, unpublished observations). A better test of this idea would be to extend the required timing of the action to 1 s and record in the two cortical areas simultaneously. Neural integration could develop with the increased length of the foreperiod, and the rate of integrator "ramping" might track the animals' prediction of the cue time. Alternatively, recordings could be made in well trained rats that experience two foreperiods presented in blocks. We predict that the rate of integration would change following block changes, e.g., as the foreperiod switches from relatively long to short. To our knowledge, no studies like these have been done.

We have shown that neurons in mPFC are tuned to multiple aspects of the simple RT task (Figs. 7, 8). Because we find neurons that are sensitive to both task state (x_1) and the length of time in that state (x_2), we predict that single neurons would be sensitive to additional aspects of the task in more complex tasks. This is due to the mPFC as a whole tracking the dynamics of that task to modify behavior. Initial support for this idea can be found in a recent study of the mPFC that used a delayed spatial alternation task and found similar dynamics in principal component space across the mPFC ensembles (Horst and Laubach, 2012). This prediction could be further tested by using a task that has different temporal dynamics in different trial blocks or task contexts, and determining whether there are neurons with loadings on many primary components in a principal component analysis. One such task was used by Delatour and Gisquet-Verrier (2001).

Notes

Supplemental data, i.e., the source code for simulations and data analyses, for this article are available at https://github.com/tbekolay/jneurosci2013/releases/tag/2013-11-29. This material has not been peer reviewed.

References

Aarts E, Roelofs A, van Turennout M (2008) Anticipatory activity in anterior cingulate cortex can be independent of conflict and error likelihood. J Neurosci 28:4671–4678. CrossRef Medline

Aarts E, Roelofs A, van Turennout M (2009) Attentional control of task and response in lateral and medial frontal cortex: brain activity and reaction time distributions. Neuropsychologia 47:2089–2099. CrossRef Medline

Alexander WH, Brown JW (2011) Medial prefrontal cortex as an action-outcome predictor. Nat Neurosci 14:1338–1344. CrossRef Medline

Amiez C, Joseph JP, Procyk E (2005) Anterior cingulate error-related activity is modulated by predicted reward. Eur J Neurosci 21:3447–3452. CrossRef Medline

Behrens TE, Woolrich MW, Walton ME, Rushworth MF (2007) Learning the value of information in an uncertain world. Nat Neurosci 10:1214–1221. CrossRef Medline

Bekolay T, Bergstra J, Hunsberger E, DeWolf T, Stewart TC, Rasmussen D, Choo X, Voelker A, Eliasmith C (2014) Nengo: a Python tool for building large-scale functional brain models. Front Neuroinform 7:48. CrossRef

Botvinick MM, Cohen JD, Carter CS (2004) Conflict monitoring and anterior cingulate cortex: an update. Trends Cogn Sci 8:539–546. CrossRef Medline

Brown JW, Braver TS (2005) Learned predictions of error likelihood in the anterior cingulate cortex. Science 307:1118–1121. CrossRef Medline

Carracedo LM, Kjeldsen H, Cunnington L, Jenkins A, Schofield I, Cunningham MO, Davies CH, Traub RD, Whittington MA (2013) A neocortical delta rhythm facilitates reciprocal interlaminar interactions via nested theta rhythms. J Neurosci 33:10750–10761. CrossRef Medline

Carter CS, Braver TS, Barch DM, Botvinick MM, Noll D, Cohen JD (1998) Anterior cingulate cortex, error detection, and the online monitoring of performance. Science 280:747–749. CrossRef Medline

Cavanagh JF, Cohen MX, Allen JJ (2009) Prelude to and resolution of an error: EEG phase synchrony reveals cognitive control dynamics during action monitoring. J Neurosci 29:98–105. CrossRef Medline

Critchley HD, Mathias CJ, Josephs O, O'Doherty J, Zanini S, Dewar BK, Cipolotti L, Shallice T, Dolan RJ (2003) Human cingulate cortex and autonomic control: converging neuroimaging and clinical evidence. Brain 126:2139–2152. CrossRef Medline

Delatour B, Gisquet-Verrier P (2001) Involvement of the dorsal anterior cingulate cortex in temporal behavioral sequencing: subregional analysis of the medial prefrontal cortex in rat. Behav Brain Res 126:105–114. CrossRef Medline

Eliasmith C (2005) A unified approach to building and controlling spiking attractor networks. Neural Comput 17:1276–1314. CrossRef Medline

Eliasmith C, Anderson CH (2003) Neural engineering: computation, representation, and dynamics in neurobiological systems. Cambridge, MA: MIT.

Fecteau JH, Munoz DP (2003) Exploring the consequences of the previous trial. Nat Rev Neurosci 4:435–443. CrossRef Medline

Georgopoulos AP, Schwartz AB, Kettner RE (1986) Neuronal population coding of movement direction. Science 233:1416–1419. CrossRef Medline

Histed MH, Pasupathy A, Miller EK (2009) Learning substrates in the primate prefrontal cortex and striatum: sustained activity related to successful actions. Neuron 63:244–253. CrossRef Medline

Horst NK, Laubach M (2012) Working with memory: evidence for a role for the medial prefrontal cortex in performance monitoring during spatial delayed alternation. J Neurophysiol 108:3276–3288. CrossRef Medline

Hyman JM, Whitman J, Emberly E, Woodward TS, Seamans JK (2013) Action and outcome activity state patterns in the anterior cingulate cortex. Cereb Cortex 23:1257–1268. CrossRef Medline

Khamassi M, Enel P, Dominey PF, Procyk E (2013) Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters. Prog Brain Res 202:441–464. CrossRef Medline

Koch C (1998) Biophysics of computation: information processing in single neurons. Oxford: Oxford UP.

Koyama T, Kato K, Tanaka YZ, Mikami A (2001) Anterior cingulate activity during pain-avoidance and reward tasks in monkeys. Neurosci Res 39:421–430. CrossRef Medline

Laubach M (2011) A comparative perspective on executive and motivational control by the medial prefrontal cortex. In: Neural basis of motivational and cognitive control (Mars RB, Sallet J, Rushworth MFS, Yeung N, eds), pp. 95. Cambridge, MA: MIT.

Laubach M, Wessberg J, Nicolelis MA (2000) Cortical ensemble activity increasingly predicts behaviour outcomes during learning of a motor task. Nature 405:567–571. CrossRef Medline

Luu P, Posner MI (2003) Anterior cingulate cortex regulation of sympathetic activity. Brain 126:2119–2120. CrossRef Medline

MacNeil D, Eliasmith C (2011) Fine-tuning and the stability of recurrent neural networks. PLoS ONE 6:e22885. CrossRef Medline

Medalla M, Barbas H (2009) Synapses with inhibitory neurons differentiate anterior cingulate from dorsolateral prefrontal pathways associated with cognitive control. Neuron 61:609–620. CrossRef Medline

Modirrousta M, Fellows LK (2008) Dorsal medial prefrontal cortex plays a necessary role in rapid error prediction in humans. J Neurosci 28:14000–14005. CrossRef Medline

Muir JL, Everitt BJ, Robbins TW (1996) The cerebral cortex of the rat and visual attentional function: dissociable effects of mediofrontal, cingulate, anterior dorsolateral, and parietal cortex lesions on a five-choice serial reaction-time task. Cereb Cortex 6:470–481. CrossRef Medline

Mulert C, Gallinat J, Dorn H, Herrmann WM, Winterer G (2003) The relationship between reaction time, error rate and anterior cingulate cortex activity. Int J Psychophysiol 47:175–183. CrossRef Medline

Naito E, Kinomura S, Geyer S, Kawashima R, Roland PE, Zilles K (2000) Fast reaction to different sensory modalities activates common fields in the motor areas, but the anterior cingulate cortex is involved in the speed of reaction. J Neurophysiol 83:1701–1709. Medline

Narayanan NS, Laubach M (2006) Top-down control of motor cortex ensembles by dorsomedial prefrontal cortex. Neuron 52:921–931. CrossRef Medline

Narayanan NS, Laubach M (2008) Neuronal correlates of post-error slowing in the rat dorsomedial prefrontal cortex. J Neurophysiol 100:520–525. CrossRef Medline

Narayanan NS, Laubach M (2009) Delay activity in rodent frontal cortex during a simple reaction-time task. J Neurophysiol 101:2859–2871. CrossRef Medline

Narayanan NS, Cavanagh JF, Frank MJ, Laubach M (2013) Common medial frontal mechanisms of adaptive control in humans and rodents. Nat Neurosci 16:1888–1895. CrossRef Medline

Niv Y, Schoenbaum G (2008) Dialogues on prediction errors. Trends Cogn Sci 12:265–272. CrossRef Medline

Risterucci C, Terramorsi D, Nieoullon A, Amalric M (2003) Excitotoxic lesions of the prelimbicinfralimbic areas of the rodent prefrontal cortex disrupt motor preparatory processes. Eur J Neurosci 17:1498–1508. CrossRef Medline

Rushworth MF, Walton ME, Kennerley SW, Bannerman DM (2004) Action sets and decisions in the medial frontal cortex. Trends Cogn Sci 8:410–417. CrossRef Medline

Salinas E, Abbott LF (1994) Vector reconstruction from firing rates. J Comput Neurosci 1:89–107. CrossRef Medline

Seung HS (1996) How the brain keeps the eyes still. Proc Natl Acad Sci U S A 93:13339–13344. CrossRef Medline

Shadlen MN, Newsome WT (2001) Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. J Neurophysiol 86:1916–1936. Medline

Sheth SA, Mian MK, Patel SR, Asaad WF, Williams ZM, Dougherty DD, Bush G, Eskandar EN (2012) Human dorsal anterior cingulate cortex neurons mediate ongoing behavioural adaptation. Nature 488:218–221. CrossRef Medline

Singh R, Eliasmith C (2006) Higher-dimensional neurons explain the tuning and dynamics of working memory cells. J Neurosci 26:3667–3678. CrossRef Medline

van Veen V, Holroyd CB, Cohen JD, Stenger VA, Carter CS (2004) Errors without conflict: implications for performance monitoring theories of anterior cingulate cortex. Brain Cogn 56:267–276. CrossRef Medline

Wang XJ (2002) Probabilistic decision making by slow reverberation in cortical circuits. Neuron 36:955–968. CrossRef Medline