Computational aspects of feedback in neural circuits

Wolfgang Maass*, Prashant Joshi*,
and Eduardo D. Sontag†

Corresponding author: Wolfgang Maass (maass@igi.tugraz.at)

Abstract

It had previously been shown that generic cortical microcircuit models can perform complex real-time computations on continuous input streams, provided that these computations can be carried out with a rapidly fading memory. We investigate in this article the computational capability of such circuits in the more realistic case where not only readout neurons, but in addition a few neurons *within* the circuit have been trained for specific tasks. This is essentially equivalent to the case where the output of trained readout neurons is fed back into the circuit. We show that this new model overcomes the limitation of a rapidly fading memory. In fact, we prove that in the idealized case without noise it can carry out any conceivable digital or analog computation on time-varying inputs. But even with noise the resulting computational model can perform a large class of biologically relevant real-time computations that require a non-fading memory. We demonstrate these computational implications of feedback both theoretically and through computer simulations of detailed cortical microcircuit models. We show that the application of simple learning procedures (such as linear regression or perceptron learning) enables such circuits, in spite of their complex inherent dynamics, to represent time over behaviorally relevant long time spans, to integrate evidence from incoming spike trains over longer periods of time, and to process new information contained in such spike trains in diverse ways according to the current internal state of the circuit. In particular we show that such generic cortical microcircuits with feedback provide a new model for working memory that is consistent with a large set of biological constraints.

Although this article examines primarily the computational role of feedback in circuits of neurons, the mathematical principles on which its analysis is based apply to a large variety of dynamical systems. Hence they may also throw new light on the computational role of feedback in other complex biological dynamical systems, such as for example genetic regulatory networks.

*Institute for Theoretical Computer Science, Technische Universitaet Graz, Inffeldgasse 16b, A-8010 Graz, Austria, Tel. +43 316 873 5822, Fax. +43 316 873 5805

†Department of Mathematics, Rutgers, The State University of New Jersey, USA

1 Introduction

The neocortex performs a large variety of complex computations in real-time. It is conjectured that these computations are carried out by a network of cortical microcircuits, where each microcircuit is a rather stereotypical circuit of neurons within a cortical column. A characteristic property of these circuits and networks is an abundance of feedback connections. But the computational function of these feedback connections is largely unknown. Two lines of research have been engaged in order to solve this problem. In one approach, which one might call the constructive approach, one builds hypothetical circuits of neurons and shows that (under some conditions on the response behavior of its neurons and synapses) such circuits can perform specific computations. In another research strategy, which one might call the analytical approach, one starts with data-based models for actual cortical microcircuits, and analyses which computational operations such "given" circuits can perform under the assumption that a learning process assigns suitable values to some of their parameters (e.g. synaptic efficacies of readout neurons). An underlying assumption of the analytical approach is that complex recurrent circuits, such as cortical microcircuits, cannot be fully understood in terms of the usually considered properties of their components. Rather, system level approaches that address directly the dynamics of the resulting recurrent neural circuits are needed to complement the bottom-up analysis. This line of research started with the identification and investigation of socalled canonical microcircuits [1]. Subsequently it was shown that quite complex real-time computations on spike trains can be carried out by such "given" models for cortical microcircuits ([2–5], see [6] for a review). A fundamental limitation of this approach was that only those computations could be modeled which can be carried out with a fading memory, more precisely only those computations that only require to integrate information over a time span of 200 or 300 ms (its maximal length depends on the amount of noise in the circuit and the complexity of the input spike trains [7]). In particular, computational tasks that require a representation of elapsed time between salient sensory events or motor actions [8], or an internal representation of expected rewards [9],[10],[11], working memory [12], accumulation of sensory evidence for decision making [13], the updating and holding of analog variables such as for example the desired eye position [14], and differential processing of sensory input streams according to attentional or other internal states of the neural system [15] could not be modeled in this way. Previous work on concrete examples of artificial neural networks [16] and cortical microcircuit models [17] had already indicated that these shortcomings of the model might arise only if one assumes that learning affects exclusively the synapses of readout neurons that project the results of computations to other circuits or areas, without giving feedback into the circuit from which they extract information. This scenario is in fact rather unrealistic from a biological perspective, since pyramidal neurons in the cortex typically have in addition to their long projecting axon a a large number of axon collaterals that provide feedback to the local circuit [18]. Abundant feedback connections also exist on the network level between different brain areas

[19]. We show in this article that if one takes feedback connections from readout neurons (that are trained for specific tasks) into account, generic cortical microcircuit models can solve all of the previously listed computational tasks. In fact, one can demonstrate this also for circuits whose underlying noise levels and models for neurons and synapses are substantially more realistic than those which had previously been considered in models for working memory and related tasks.

We show in the first part of section 2 that the significance of feedback for the computational power of neural circuits and other dynamical systems can be explained on the basis of general principles. Theorem 1 implies that a large class of dynamical systems, in particular systems of differential equations which are commonly used to describe the dynamics of firing activity in neural circuits, gain universal computational capabilities for digital and analog computation as soon as one considers them in combination with feedback. A further mathematical result (Theorem 2) implies that the capability to process online input streams in the light of non-fading (or slowly fading) internal states is preserved in the presence of fairly large levels of internal noise. On the basis of this theoretical foundation one can explain why the computer models of generic cortical microcircuits, which are considered in the second part of section 2, are able to solve the previously mentioned benchmark tasks. These results suggest a new computational model for cortical microcircuits, which includes the capability to process online input streams in diverse ways according to different "instructions" that are implemented through high-dimensional attractors of the underlying dynamical system. The high-dimensionality of these attractors (which are discussed in section 2.2) results from the fact that only a small fraction of synapses need to be modified for their creation. In comparison with the commonly considered low dimensional attractors, such high-dimensional attractors have additional attractive properties such as compositionality (the intersection of several of them is in general non-empty), and compatibility with real-time computing on online input streams within the same circuit.

2 Results

2.1 Theoretical Analysis

The dynamics of firing activity in recurrent circuits of neurons is commonly modeled by systems of nonlinear differential equations of the form

$$x_i'(t) \;=\; -\lambda_i x_i(t) \,+\, \sigma\,(\sum_{j=1}^{n} a_{ij} x_j(t) + b_i \cdot v(t))\,, \quad i = 1, \ldots, n\,, \tag{1}$$

or

$$x_i'(t) \;=\; -\lambda_i x_i(t) \,+\, \sigma\,(\sum_{j=1}^{n} a_{ij} x_j(t)) \,+\, b_i \cdot \sigma(v(t))\,, \quad i = 1, \ldots, n \tag{2}$$

([20–23]). Here each $x_i, i = 1, \ldots, n$, is a real-valued variable which represents the current firing rate of the i^{th} neuron or population of neurons in a recurrent neural circuit, and $v(t)$ is an external input stream. The coefficients a_{ij}, b_i denote the strengths of synaptic connections, and the $\lambda_i > 0$ denote time constants. The function σ is some sigmoidal activation function (nondecreasing, with bounded range). In most models of neural circuits, the parameters a_{ij} in these differential equations are chosen so that the resulting dynamical system has a fading memory for preceding inputs. If one makes the synaptic connection strengths a_{ij} so large that recurrent activity does not dissipate, the neural circuit tends to exhibit persistent memory. But it is usually quite difficult to control the content of this persistent memory, since it tends to be swamped with minor details of external inputs (or initial conditions) from the distant past. Hence this chaotic regime of recurrent neural circuits (see [24] for a review) is apparently also not suitable for biologically realistic real-time computations on online input streams, that combine new information from the current input with selected (e.g., behaviorally relevant) aspects of external or internal inputs from the past.

Recurrent circuits of neurons (e.g. those described by equations (1) or (2)) are from a mathematical perspective special cases of dynamical systems. The subsequent mathematical results show that a large variety of dynamical systems, in particular also fading memory systems of type (1) or (2), can overcome in the presence of feedback the computational limitations of a fading memory without necessarily falling into the chaotic regime. In fact, feedback endows them with *universal* capabilities for *analog computing*, in a sense that can be made precise in the following way (see Fig. 1A-C for an illustration):

Theorem 1 *A large class $\mathcal{S}_n$ of systems of differential equations of the form*

$$x_i'(t) = f_i(x_1(t), \ldots, x_n(t)) + g_i(x_1(t), \ldots, x_n(t)) \cdot v(t), \quad i = 1, \ldots, n \tag{3}$$

are in the following sense universal for analog computing:

It can respond to an external input $u(t)$ with the dynamics of any n^{th} order differential equation of the form

$$z^{(n)}(t) = G(z(t), z'(t), z''(t), \ldots, z^{(n-1)}(t)) + u(t) \tag{4}$$

(for arbitrary smooth functions $G : \mathbb{R}^n \to \mathbb{R}$) if the input term $v(t)$ is replaced by a suitable memoryless feedback function $K(x_1(t), \ldots, x_n(t), u(t))$, and if a suitable memoryless readout function $h(x_1(t), \ldots, x_n(t))$ is applied to its internal state $\langle x_1(t), \ldots, x_n(t) \rangle$.

Also the dynamic responses of all systems consisting of several higher order differential equations of the form (4) can be simulated by fixed systems of the form (3) with a corresponding number of feedbacks.

This result says more precisely that for any n^{th} order differential equation (4) there exists a (memory-free) feedback function $K : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and a memory-free readout

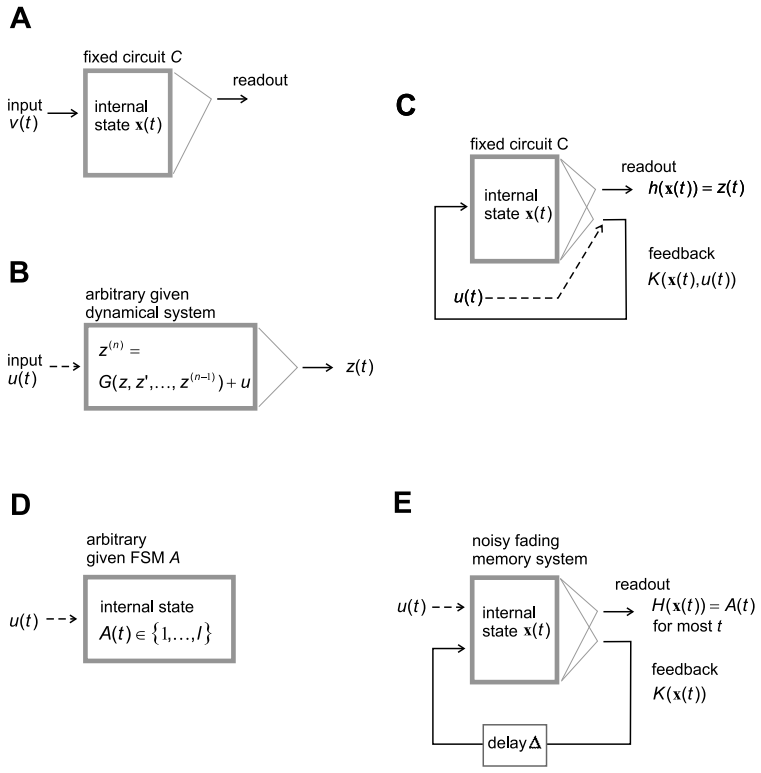

Figure 1: Computational architectures considered in Theorems 1 and 2. **(A)** A fixed circuit C whose dynamics is described by the system (3). **(B)** An arbitrary given n^{th} order dynamical system (4) with external input $u(t)$. **(C)** If the input $v(t)$ to circuit C is replaced by a suitable feedback $K(\mathbf{x}(t), u(t))$, then this fixed circuit C can simulate the dynamic response $z(t)$ of the arbitrarily given system shown in B, for any input stream $u(t)$. **(D)** Arbitrary given finite state machine (FSM) A with l states. **(E)** A noisy fading memory system with feedback can reliably reproduce the current state $A(t)$ of the given FSM A, except for time points t shortly after A has switched its state.

function $h : \mathbb{R}^n \to \mathbb{R}$ (which can both be chosen to be smooth, in particular continuous) so that, for every external input $u(t), t \geq 0$, and each solution $z(t)$ of the forced system (4) there is an input $u_0(t)$ with $u_0(t) \equiv 0$ for all $t \geq 1$, so that the solution $\mathbf{x}(t) = \langle x_1(t), \ldots, x_n(t) \rangle$ of the fixed system (3)

$$\mathbf{x}'(t) \;=\; f(\mathbf{x}(t)) + g(\mathbf{x}(t))K(\mathbf{x}(t), u(t) + u_0(t))\,, \quad \mathbf{x}(0) = 0 \tag{5}$$

(for $f : \mathbb{R}^n \to \mathbb{R}^n$ consisting of $\langle f_1, \ldots, f_n \rangle$ and $g : \mathbb{R}^n \to \mathbb{R}^n$ consisting of $\langle g_1, \ldots, g_n \rangle$) is such that

$$h(\mathbf{x}(t)) = z(t) \quad \text{ for all } \; t \geq 1\,.$$

Theorem 1 implies that even if some fixed dynamical system (3) from the class $\mathcal{S}_n$ has fading memory, a suitable feedback K and readout function h will enable it to carry out

specific computations with persistent memory. In fact, it can carry out *any* computation with persistent memory which could possibly be carried out by *any* dynamical system (4). To get a clear understanding of this universality property, one should note that the feedback function K and the readout function h depend only on the function G that characterizes the simulated system (4), but not on the external input $u(t)$ or the particular solution $z(t)$ of (4) that it simulates. Hence Theorem 1 implies in particular that any system (3) that belongs to the class $\mathcal{S}_n$ has in conjunction with feedback the computational power of a universal Turing machine (see [25] or [26] for relevant concepts from computation theory). This follows from the fact that every Turing machine (hence any conceivable digital computation, most of which require a persistent memory) can be simulated by systems of equations of the form (4) (this was shown in [27] for the case with continuous time, and in [28, 29] for recurrent neural networks with discrete time; see [30] for a review). But possibly more relevant for applications to biological systems is the fact that any fixed system (3) that belongs to the class $\mathcal{S}_n$ is able to emulate any conceivable *continuous* dynamic response to an input stream $u(t)$ if it receives a suitable feedback $K(x_1(t), \ldots, x_n(t), u(t))$, where K can always be chosen to be continuous. Hence one may argue that these systems (3) are also universal for *analog* computing on time-varying inputs.

The class $\mathcal{S}_n$ of dynamical systems that become through feedback universal for analog computing subsumes systems of the form

$$x_i'(t) \;=\; -\lambda_i x_i(t) \;+\; \sigma\,(\sum_{j=1}^{n} a_{ij}\cdot x_j(t)) \;+\; b_i\cdot v(t)\,, \quad i=1,\ldots,n\,; \tag{6}$$

for example if the λ_i are pairwise different and $a_{ij}=0$ for all i,j, and all b_i are nonzero.[1] Systems of the form (1) or (2) are of a slightly different form, since there the activation function σ (that has a bounded range) is applied to the term $v(t)$). But such systems (1), (2) can still be universal for all *bounded* analog responses of arbitrary dynamical systems (4), because of the following observation:

For each constant $c>0$ there is a constant $C>0$ such that: for every external input $u(t), t\geq 0$, and each solution $z(t)$ of the forced system (4) such that

$$|u(t)|\leq c \text{ and } \left|z^{(i)}(t)\right|\leq c \quad \text{for all } i=0,\ldots,n-1\,, \quad \text{for all } t\geq 0$$

the input u_0 can be picked so that the feedback

$$v(t)=K(\mathbf{x}(t), u(t)+u_0(t)) \quad t\geq 0$$

to (1) or (2) satisfies:

$$|v(t)|\leq C \quad \text{for all } t\geq 0\,.$$

[1]Fewer restrictions are needed if more then one feedback to the system (6) can be used.

Thus, if we know *a priori* that we will only deal with solutions of the differential equation (4) that are bounded by c, and inputs are similarly bounded, we could also consider instead of (3) a system such as $\mathbf{x}'(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\sigma(v(t))$ with $f, g : \mathbb{R}^n \to \mathbb{R}^n$, where some bounded activation function $\sigma : \mathbb{R} \to \mathbb{R}$ (e.g. $q \cdot \tanh(v)$, for a suitable constant q) is applied to the term $v(t)$. The resulting feedback term $\sigma(K(\mathbf{x}(t), u(t) + u_0(t)))$ is then of a mathematical form which is adequate for modeling feedback in neural circuits.

In order to prove Theorem 1, one defines $\mathcal{S}_n$ as the class of dynamical systems (3) that are feedback equivalent to the special linear system (consisting of n differential equations)

$$\mathbf{x}'(t) = \mathbf{A}_n \mathbf{x}(t) + \mathbf{b}_n v(t) \tag{7}$$

with

$$\mathbf{A}_n := \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \qquad \mathbf{b}_n := \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

The notion of "feedback equivalence", which is in fact an equivalence relation, expresses that two systems of differential equations can be transformed into each other through application of a suitable feedback and a change of basis in the state and control value space (see section 5.2 in [31]). One can easily see that it preserves the universality property considered in Theorem 1. The linear system (7) has this universality property according to the well-known feedback linearization method (see [31]), which shows that arbitrary differential equations of the form (4) can be simulated by a linear system (7) with a suitable feedback. We refer to section 4.1.1 for further details of the proof.

Theorem 1 implies that a generic neural circuit may become through feedback a universal computational device, which can not only simulate any Turing machine, but also any conceivable model for analog computing with bounded dynamic responses. The "program" of such arbitrary simulated computing machine gets encapsulated in the static functions K that characterize the memoryless computational operations that are required from feedback units, and the static readout functions h. Since these functions are static, i.e. time-invariant, and continuous, they provide suitable targets for *learning*. More precisely, in order to train a generic neural circuit to simulate the dynamic response of an arbitrary dynamical system, it suffices to train - apart from readout neurons - a few neurons within the circuit (or within some external loop) to transform the vector $\mathbf{x}(t)$, that represents the current firing activity of its neurons, and the current external input $u(t)$ into a suitable feedback value $K(\mathbf{x}(t), u(t))$. This could for example be carried out by

training a suitable feedforward neural network within the larger circuit, which can approximate any continuous feedback function K [32]. Furthermore we will show in section 2.2 that these feedback functions K can in many biologically relevant cases be chosen to be linear, so that it would in principle suffice to train a single neuron to compute K.

It is known that the memory capacity of such circuit is reduced to some finite number of bits if these feedback functions K are not learnt perfectly, or if there are other sources of noise in the system. More generally, no analog circuit with noise can simulate arbitrary Turing machines [33]. But the subsequent Theorem 2 shows that fading memory systems with noise and imperfect feedback can still achieve the maximal possible computational power within this a-priori limitation: they can simulate any given finite state machine (FSM). Note that any Turing machine with tapes of finite length is a special case of a FSM. Furthermore any existing digital computer is a FSM, hence the computational capability of FSM's is actually quite large.

In order to avoid the cumbersome mathematical difficulties that arise when one analyses differential equations with noise, we formulate and prove Theorem 2 on a more abstract level, resorting to the notion of fading memory filters with noise (see section 4.1.2 for details). We assume here that the input-output behavior of those dynamical systems with noise, for which we want to determine the computational impact of (imprecise) state feedback, can be modeled by fading memory filters with additive noise on their output. The assumption that the amplitude of this noise is bounded is a necessary assumption according to [34]. We refer to [3], [4], [35] for further discussions of the relationship between models for neural circuits and fading memory filters. In particular it was shown in [35] that every time-invariant fading memory filter can be approximated by models for neural circuits, provided that these models reflect the empirically found diversity of time constants of neurons and synapses.

Theorem 2 *Feedback allows linear and nonlinear fading memory systems, even in the presence of additive noise with bounded amplitude, to employ for real-time processing of time-varying inputs the computational capability and non-fading states of any given FSM (see Fig. 1D-E).*

The precise formalization and the proof of this result is given in section 4.1.2. The external input $u(t)$ can in this case be injected directly into the fading memory system, so that the feedback $K(\mathbf{x}(t))$ depends only on the internal state $\mathbf{x}(t)$ (see Fig. 1E). One essential ingredient of the proof is a method for making sure that noise does not get amplified through feedback: the functions K that provide feedback values $K(\mathbf{x}(t))$ can be chosen in such a way that they cancel the impact of imprecision in the values $K(\mathbf{x}(s))$ for immediately preceding time steps $s < t$.

2.2 Applications to Generic Cortical Microcircuit Models

The preceding theoretical results imply that it is possible for dynamical systems to carry out computations with persistent memory without acquiring all the computational disadvantages of the chaotic regime, where the memory capacity of the system is dominated by noise. Feedback units can create selective "loopholes" into the fading memory dynamics of a dissipative system, that can only be activated by specific patterns in the input or circuit dynamics. In this way the potential content of persistent memory can be controlled by feedback units that have been trained to recognize such patterns. This feedback may arise from a few neurons within the circuit, or from neurons within a larger feedback loop. The task to approximate a suitable feedback function K is less difficult than it may appear on first sight, since it suffices in many cases to approximate a *linear* feedback function. The reason is that sufficiently large generic cortical microcircuit models have an inherent kernel property [7], in the sense of machine learning [36]. This means that a large reservoir of diverse nonlinear responses to current and recent input patterns is automatically produced within the recurrent circuit. In particular, nonlinear combinations of variables $a, b, c, \ldots$ (that may result from the circuit input or internal activity) are automatically computed at internal nodes of the circuit. Consequently numerous low degree polynomials in these variables $a, b, c, \ldots$ can be approximated by *linear* combinations of outputs of neurons from the recurrent circuit. An example of this effect is demonstrated in Fig. 2G, where it is shown that the product of firing rates $r_3(t)$ and $r_4(t)$ of two independently varying afferent spike train inputs can be approximated quite well by a linear readout neuron. This kernel property of biologically realistic cortical microcircuit models arises from the fact that these circuits have many additional nonlinearities besides those that appear in the equations (1), (2), (6).

We refer to those neurons where the weights of synaptic connections from neurons within the circuit are adapted for a specific computational task (rather than chosen randomly from distributions that are based on biological date, like for all other synapses in the circuit) in the following as *readout neurons*. Readout neurons were modeled in most of our simulations simply by weighted sums applied to low-pass filtered spike outputs from their presynaptic neurons (where the low-pass filter reflects properties of postsynaptic receptors and membrane of a biological neuron). The analog output of such linear readout neurons can be interpreted as a time-varying firing rate. However we show in Fig. 2 that these readout neurons can (with a moderate loss in performance) also be modeled by spiking neurons, like the other neurons in the simulated circuit. Hence not only those circuits that receive feedback from external readout neurons, but also generic recurrent circuits in which a few neurons have been trained for a specific task acquire computational capabilities for real-time processing that are not restricted to computations with fading memory.

Theorem 2 predicts that the training of a few of its neurons enables generic neural circuits to employ persistent internal states for state-dependent processing of online input

streams. Previous models for non-fading memory in neural circuits [12, 37–39] proposed that it is implemented through low-dimensional attractors in the circuit dynamics. These attractors tend to freeze or entrain the whole state of the circuit, and thereby shut it off from the online input stream (although independent local attractors could emerge in local subcircuits under some conditions [38]). In contrast, the generation of non-fading memory through a few trained neurons does not entail that the dynamics of the circuit is dominated by their persistent memory states. For example, an attractor created by a neuron with a constant target output $K(\mathbf{x}) = c$ only constrains the circuit state $\mathbf{x}$ to remain in the sub-manifold $\{\mathbf{x} : K(\mathbf{x}) = c\}$ of its high-dimensional state space. This sub-manifold is itself in general high-dimensional. In particular, if $K(\mathbf{x})$ is a linear function $\mathbf{w} \cdot \mathbf{x}$, which often suffices as we will show, the dimensionality of the sub-manifold $\{\mathbf{x} : K(\mathbf{x}) = c\}$ differs from the dimension of the full state space only by 1. Hence several such sub-manifolds have in general a high-dimensional intersection, and their intersection still leaves sufficiently many degrees of freedom for the circuit state $\mathbf{x}$ to also absorb continuously new information from online input streams.

We simulated generic cortical microcircuit models consisting of 600 integrate-and-fire (I&F) neurons (for Fig. 2, 3), and circuits consisting of 600 conductance-based Hodgkin-Huxley (HH) neurons (for Fig. 4), in either case with a rather high level of noise that reflects experimental data on the high conductance state in vivo [40]. We used biologically realistic models for dynamic synapses whose individual mixture of paired-pulse depression and facilitation (depending on the type of pre- and postsynaptic neuron) was based on experimental data from [41], [42]. These circuits were not constructed for any particular computational task. In particular, sparse synaptic connectivity between neurons was generated (with a biologically realistic bias towards short connections) by a probabilistic rule, and synaptic parameters were chosen randomly from distributions that depend on the type of pre- and postsynaptic neurons (in accordance with empirical data from [41], [42]). More precisely, the 600 neurons of each circuit were placed on the integer grid points of a $5 \times 5 \times 24$ grid. 20% of these neurons were randomly chosen to be inhibitory. The probability of a synaptic connection from neuron a to neuron b (as well as that of a synaptic connection from neuron b to neuron a) was defined as $C \cdot \exp(-D^2(a, b)/\lambda^2)$, where $D(a, b)$ is the Euclidean distance between neurons a and b, and λ is a parameter which controls both the average number of connections and the average distance between neurons that are synaptically connected (we set $\lambda = 3$). Depending on whether the pre- or postsynaptic neuron were excitatory (E) or inhibitory (I), the value of C was set according to [42] to 0.3 (EE), 0.2 (EI), 0.4 (IE), 0.1 (II), yielding an average of 10900 synapses for the chosen circuit size. External inputs and feedbacks from readouts were connected to populations of neurons in the circuit with randomly chosen connection strengths. Further details of the simulated microcircuit models can be found in section 4.2. Details of the subsequently discussed computer experiments are given in sections 4.3 - 4.5.

The following procedure was applied to train readout neurons, i.e. to adjust the weights of synaptic connections from neurons in the circuit to readout neurons for specific

computational tasks (while leaving all other parameters of the generic microcircuit model unchanged):

- First those readout neurons were trained that provide feedback, then the other readout neurons.
- During the training of readout neurons that provide feedback, their actual feedback was replaced by a *noisy* version of their target output ("teacher forcing").
- Each readout neuron was trained by linear regression to output at any time t a particular target value $f(t)$. Linear regression was applied to a set of data points of the form $\langle \mathbf{y}(t), f(t) \rangle$, for many time points t, where $\mathbf{y}(t)$ is the output of low pass filters applied to the spike trains of presynaptic neurons, and $f(t)$ is the target output.

Note that teacher forcing with noisy versions of target feedback values trains these readouts to correct errors resulting from imprecision in their preceding feedback (rather than amplifying errors).

In our first computer experiment, readout neurons were trained to turn a high-dimensional attractor on or off (Fig. 2D), in response to bursts in 2 of the 4 independent input spike trains. More precisely, 8 neurons were trained to represent in their firing activity at any time the information in which of the input streams 1 or 2 a burst had most recently occurred. If it had occurred most recently in stream 1, they were trained to fire at 40 Hz, and if a burst had occurred most recently in input stream 2, they were trained not to fire. Hence these neurons were required to represent the non-fading state of a simple FSM, demonstrating in an example the computational capabilities predicted by Theorem 2. Fig. 2G demonstrates that the circuit retains its kernel property inspite of the feedback injected into the circuit by these readouts. But beyond the emulation of a simple FSM, the resulting generic cortical microcircuit is able to combine information stored in the current state of the FSM with new information from the online circuit input. For example, Fig. 2E shows that other readouts from the same circuit can be trained to amplify their response to specific inputs if the high-dimensional attractor is in the "on"-state. Readouts can also be trained to change the function that they compute if the high-dimensional attractor is in the on-state (Fig. 2F). This provides an example for an online reconfigurable circuit. The readout neurons that provide feedback had been modeled in this computer simulation like the other neurons in the circuit: by I&F neurons with in-vivo like background noise. Hence they can be viewed equivalently as neurons *within* an otherwise generic circuit.

Another difficult problem in computational neuroscience is to explain how neural circuits can implement a parametric memory, i.e. how they can hold and update an *analog* value, that may represent for example an intended eye-position that a neural integrator computes from a sequence of eye-movement commands [43], an estimate of elapsed time [8], or accumulated sensory evidence [13]. Various designs have been proposed for

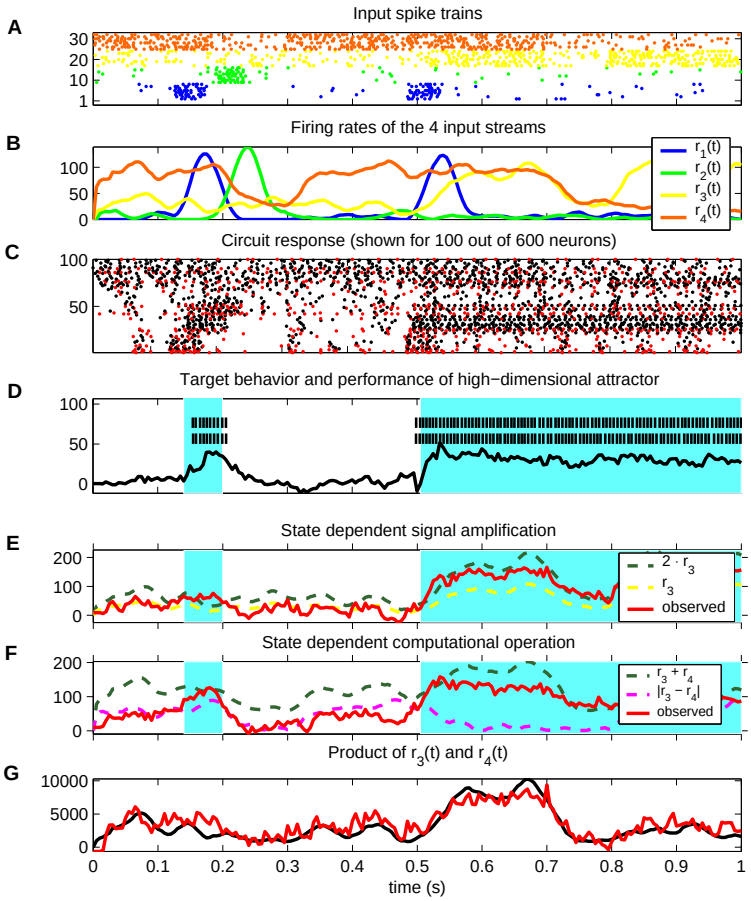

Figure 2: State-dependent real-time processing of 4 independent input streams in a generic cortical microcircuit model. **(A)** 4 input streams, consisting each of 8 spike trains generated by Poisson processes with randomly varying rates $r_i(t), i = 1, \ldots, 4$ (rates plotted in **(B)**; all rates are given in Hz). The 4 input streams and the feedback were injected into disjoint but densely interconnected subpopulations of neurons in the circuit. **(C)** Resulting firing activity of 100 out of the 600 I&F neurons in the circuit. Spikes from inhibitory neurons marked in red. **(D)** Target activation times of the high-dimensional attractor (blue shading), spike trains of 2 of the 8 I&F neurons that were trained to create the high-dimensional attractor by sending their output spike trains back into the circuit, and average firing rate of all 8 neurons (lower trace). **(E and F)** Performance of linear readouts that were trained to switch their real-time computation task in dependence of the current state of the high-dimensional attractor: output $2 \cdot r_3(t)$ instead of $r_3(t)$ if the high-dimensional attractor is on (E), output $r_3(t) + r_4(t)$ instead of $|r_3(t) - r_4(t)|$ if the high-dimensional attractor is on (F). **(G)** Performance of linear readout that was trained to output $r_3(t) \cdot r_4(t)$, showing that another linear readout from the same circuit can simultaneously carry out nonlinear computations that are invariant to the current state of the high-dimensional attractor.

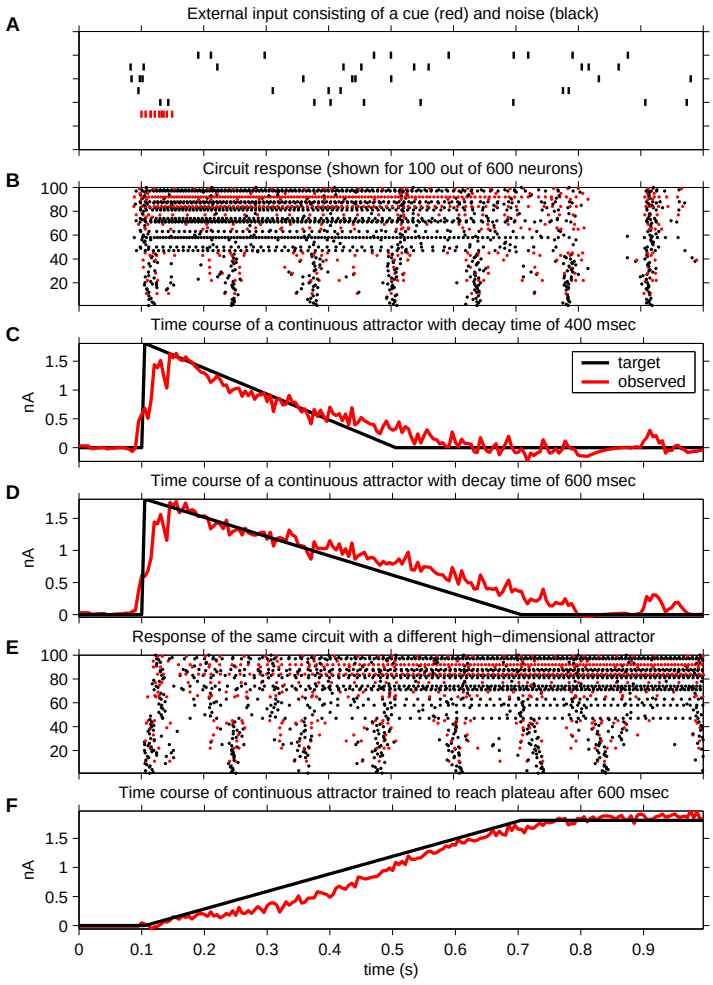

Figure 3: Representation of time for behaviorally relevant time spans in a generic cortical microcircuit model. **(A)** Afferent circuit input, consisting of a cue in one channel (red) and random spikes (freshly drawn for each trial) in the other channels. **(B)** Response of 100 neurons from the same circuit as in Fig. 2, which has here two co-existing high-dimensional attractors. The autonomously generated periodic bursts with a periodic frequency of about 8 Hz are not related to the task, and readouts were trained to become invariant to them. **(C and D)** Feedback from two linear readouts that were simultaneously trained to create and control two high-dimensional attractors. One of them was trained to decay in 400 ms (C), and the other in 600 ms (D) (scale in nA is the average current injected by feedback into a randomly chosen subset of neurons in the circuit). **(E)** Response of the same neurons as in (B), for the same circuit input, but with feedback from a different linear readout that was trained to create a high-dimensional attractor that increases its activity and reaches a plateau 600 ms after the occurrence of the cue in the input stream. **(F)** Feedback from the linear readout that creates this continuous high-dimensional attractor.

parametric memory in recurrent circuits, where continuous attractors (also referred to as line attractors) hold and update an analog value. But these approaches are inherently brittle [39], and have problems in dealing with high noise or online circuit inputs. On the other hand Fig. 3 shows that dedicated circuit constructions are not necessary, since feedback from readout neurons in *generic* cortical microcircuits models can also create high-dimensional attractors that hold and update an *analog* value for behaviorally relevant time spans. In fact, due to the high-dimensional character of the resulting high-dimensional attractors, two such analog values can be stored and updated independently (Fig. 3C,D), even within a fairly small circuit. In this example the readouts that provide feedback were simply trained to increase or reduce their feedback at each time point. Note that the resulting circuit activity is qualitatively consistent with recordings from neurons in cortex and striatum during reward expectation [9],[10],[11]. A similar ramp-like rise and fall of activity as shown in panels C, D, F has also been recorded in neurons of posterior parietal cortex of the macaque in experiments were the monkey had been trained to classify the duration of elapsed time [8]. The high-dimensionality of the continuous attractors in this model makes it feasible to constrain the circuit state to stay simultaneously in more than one continuous attractor, thereby making it in principle possible to encode complex movement plans that require specific temporal relationships between individual motor commands.

Our model for parametric memory in cortical circuits is consistent with high noise: Fig. 4G shows the typical trial-to-trial variability of a neuron in our simulated circuit of conductance based HH neurons with in-vivo like background noise. It qualitatively matches the "wide diversity of neural firing drift patterns in individual fish at all states of tuning" that was observed in the horizontal occulomotor neural integrator in goldfish [14], and the large trial-to-trial variability of neurons in prefrontal cortex of monkeys reported in [9]. In addition, this model is consistent with the surprising plasticity that has been observed even in quite specialized neural integrators [14], since continuous attractors can be created or modified in this model by changing just a few synaptic weights of neurons that are immediately involved. It does not require the presence of long-lasting postsynaptic potentials, NMDA-receptors, or other specialized details of biological neurons or synapses, although their inclusion in the model is likely to provide additional temporal stability [12]. Rather it points to complementary organizational mechanisms on the circuit level, that are likely to enhance the controllability and robustness of continuous attractors in neural circuits. The robustness of this learning-based model can be traced back to the fact that readout neurons can be trained to correct undesired circuit responses resulting from errors in their previous feedback. Furthermore such error correction is not restricted to linear computational operations, since the previously demonstrated kernel property of these generic circuits allows even linear neurons to implement complex nonlinear control strategies through their feedback. As an example we demonstrate in Fig. 4 that even under biologically realistic high noise conditions a linear readout can be trained to update a continuous attractor (Fig. 4D), to filter out input activity during certain time intervals

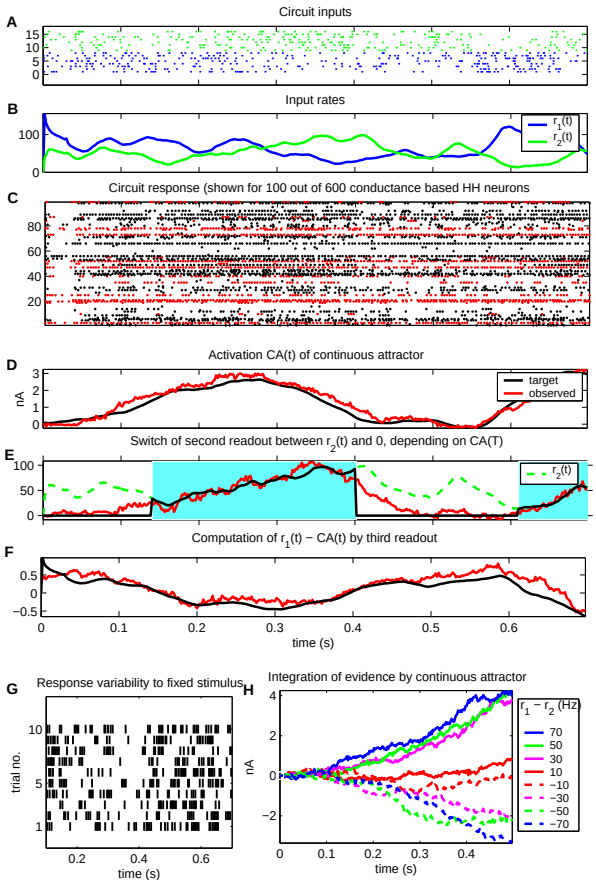

Figure 4: A model for analog real-time computation on external and internal variables in a generic cortical microcircuit (consisting of 600 conductance based HH-neurons). **(A and B)** Two input streams as in Fig. 2; their firing rates $r_1(t), r_2(t)$ are shown in (B). **(C)** Resulting firing activity of 100 neurons in the circuit. **(D)** Performance of a neural integrator, generated by feedback from a linear readout that was trained to output at any time t an approximation $CA(t)$ of the integral $\int_0^t (r_1(s) - r_2(s))ds$ over the difference of both input rates. Feedback values were injected as input currents into a randomly chosen subset of neurons in the circuit. Scale in nA shows average strength of feedback currents (also in panel H). **(E)** Performance of linear readout that was trained to output 0 as long as $CA(t)$ stayed below 1.35 nA, and to output then $r_2(t)$ until the value of $CA(t)$ dropped below 0.45 nA (i.e., in this test run during the shaded time periods). **(F)** Performance of linear readout trained to output $r_1(t) - CA(t)$, i.e. a combination of external and internal variables, at any time t (both r_1 and CA normalized into the range $[0, 1]$). **(G)** Response of a randomly chosen neuron in the circuit for 10 repetitions of the same experiment (with input spike trains generated by Poisson processes with the same time-course of firing rates), showing biologically realistic trial-to-trial variability. **(H)** Activity traces of a continuous attractor as in (D), but in 8 different trials for 8 different fixed values of r_1 and r_2 (shown on the right).

in dependence of the current state of the continuous attractor (Fig. 4E), or to combine the time-varying analog variable encoded by the current state $CA(t)$ of the continuous attractor with a time-varying variable $r_1(t)$ that is delivered by an online spike input. Hence intention-based information processing [15] and other tasks that involve a merging of external inputs and internal state information can be implemented in this way. Fig. 4C shows that a high-dimensional attractor need not entrain the firing activity of neurons in a drastic way, since it just restricts the high-dimensional circuit dynamics $\mathbf{x}(t)$ to a slightly lower dimensional manifold of circuit states $\mathbf{x}(t)$ that satisfy $\mathbf{w} \cdot \mathbf{x}(t) = f(t)$ for the current target output $f(t)$ of the corresponding linear readout. On the other hand Fig. 4E shows that the activity level $CA(t)$ of the high-dimensional attractor can nevertheless be detected by other linear readouts, and can simultaneously be combined in a nonlinear manner with a time-varying variable $r_2(t)$ from one afferent circuit input stream, while remaining invariant to the other afferent input stream.

Finally, the same generic circuit also provides a model for the integration of evidence for decision making that is compatible with in-vivo like high noise conditions. Fig. 4H depicts the time course of the same neural integrator as in panel D, but here for the case where the rates r_1, r_2 of the 2 input streams assume in 8 trials 8 different constant values after the first 100 ms (while assuming a common value of 65 Hz during the first 100 ms). The resulting time course of the continuous attractor is qualitatively similar to the meandering path towards a decision threshold that has been recorded from neurons in area LIP where firing rates represent temporally integrated evidence concerning the dominating direction of random dot movements (see Fig. 5A in [13]).

3 Discussion

We have presented a theoretically founded model for real-time computations on complex input streams with persistent internal states in generic cortical microcircuits. This model does not require a handcrafted circuit structure or biologically unrealistic assumptions such as symmetric weight distributions, static synapses that do not exhibit pair-pulsed depression or facilitation, or neuron models with low levels of noise that are not consistent with data on in-vivo conditions. Our model only requires the assumption that adaptive procedures (synaptic plasticity) in generic neural circuits can approximate linear regression. Furthermore, in contrast to classical learning paradigms for attractor neural networks, it is here not required that a large fraction of synaptic parameters in the circuit are changed when a new computational task is introduced, or a new item is stored in working memory. Rather, it suffices if those neurons that provide the circuit output and a few neurons that provide feedback are subject to synaptic plasticity. Such minimal circuit modifications have the advantage that thereby created attractors of the circuit dynamics are high-dimensional. We have shown that the circuit state can be simultaneously in several of such high-dimensional attractors, and still retain sufficiently many degrees of

freedom to absorb and process new information from online input streams. In particular, we have shown in Fig. 2 and 4 how bottom-up processing can be reconfigured in dependence of discrete internal states (implemented through high-dimensional attractors) by turning certain input channels on or off, and by changing the computational operations that are applied to input variables. Furthermore we have shown in and 4 that analog variables, which are extracted from an online input stream, can be combined in real-time computations with analog variables that are stored in high-dimensional continuous attractors. This provides in particular a model for the implementation of intention-based information processing [15] in cortical microcircuits.

It remains open how learning signals can induce neurons in a biological organism to compute specific linear feedback functions. But at least we have reduced this problem to the feasibility of perceptron-like learning (or more abstractly: to linear regression) for single neurons. Subsequent research will have to determine whether these learning requirements (which can partially be reduced to spike-timing dependent plasticity [44]) can be justified on the basis of results on unsupervised learning and reinforcement learning [45] in biological organisms.

Whereas it was previously already known that one can construct specific circuits that have universal computational capabilities for real-time computing on analog input streams, Theorems 1 and 2 of this article imply that a large variety of dynamical systems (in particular generic cortical microcircuits) can acquire through feedback such universal capabilities for computations that map time-varying inputs to time-varying outputs. It should be noted that these universal computational capabilities differ from the well known but much weaker universal approximation property of feedforward neural networks (see [32]), since not only the static output of an arbitrary continuous static function is approximated, but the dynamic response of arbitrary differential equations of higher order to time-varying inputs.

The theoretical results of this article also provide an explanation for the astounding computational capability and flexibility of echo state networks [16]. In addition they can be used to analyze computational aspects of feedback in other biological dynamical systems besides neural circuits. Several such systems, for example genetic regulatory networks, are known to implement complex maps from time-varying input streams (e.g. external signals) onto time-varying outputs (e.g. transcription rates). But little is known about the way in which these maps are implemented. Whereas feedback in biological dynamical systems is usually only analyzed and modeled from the perspective of control, we propose that an analysis of its computational aspects is likely to yield a better understanding of the computational capabilities of such systems.

4 Methods

4.1 Mathematical Methods

4.1.1 Details to the Proof of Theorem 1

We take $\mathcal{S}_n$ to be the class of n-dimensional *globally feedback linearizable* systems, that is, systems which under coordinate changes and feedback become linear controllable systems, cf. [31], Definition 5.3.1. [2] Instead of quoting that definition, let us provide an equivalent one, which is more useful in the present context (see [31], Lemma 5.3.5):

Definition: *A system (3), with smooth vector fields $f = \langle f_1, \ldots, f_n \rangle$ and $g = \langle g_1, \ldots, g_n \rangle$, belongs to $\mathcal{S}_n$ provided that there exists a diffeomorphism $T : \mathbb{R}^n \to \mathbb{R}^n$ and two smooth maps $\alpha, \beta : \mathbb{R}^n \to \mathbb{R}$, with $\beta(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in \mathbb{R}^n$, such that, for each $\mathbf{x} \in \mathbb{R}^n$:*

$$T_*(\mathbf{x})\, f(\mathbf{x}) \;=\; A_n T(\mathbf{x}) - \frac{\alpha(\mathbf{x})}{\beta(\mathbf{x})} b_n \tag{8}$$

and

$$\beta(\mathbf{x}) T_*(\mathbf{x})\, g(\mathbf{x}) \;=\; b_n\,, \tag{9}$$

where T_* denotes the Jacobian of T and

$$\mathbf{A}_n \;:=\; \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix} \qquad \mathbf{b}_n \;:=\; \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

An interpretation of this definition is as follows (see [31], Chapter 5, for more discussion): For each input $\mu(t)$ and each solution $z(t)$ of

$$z^{(n)} \;=\; \mu\,,$$

the vector function $\mathbf{x}(t) = T^{-1}(Z(t))$ satisfies (3) with the input $v(t) = \alpha(\mathbf{x}(t)) + \beta(\mathbf{x}(t))\mu(t)$, where

$$Z(t) \;=\; \big(z(t), z'(t), z''(t), \ldots, z^{(n-1)}(t)\big)\,.$$

Most treatments of feedback linearization focus on *local* feedback linearization, meaning that different T, α, β are allowed in different regions of $\mathbb{R}^n$. Local feedback linearization

[2] Feedback linearization is a useful general technique in control theory, and has often been used – for control purposes as opposed to simulation – in the context of neural computation models, see e.g. [46] and references there.

admits more elegant Lie algebraic characterizations than the global concept, as we briefly mention below. The Theorem that follows is obtained by combining the proofs of Proposition 5.3.9 and of Theorem 15 in [31], as applied to the global case. Recall that, in general, for any two vector fields f and g, one writes $\mathrm{ad}_f(g) := [f, g]$, where $[f, g]$ is the Lie bracket $g_* f - f_* g$. Iterations of the operator ad_f are defined in the obvious way: $\mathrm{ad}_f^0(g) = g$ and $\mathrm{ad}_f^{k+1}(g) = \mathrm{ad}_f(\mathrm{ad}_f^k(g))$. We use the notation $L_f\phi$, for any (smooth) vector field f and (smooth) function ϕ, to denote the Lie derivative of ϕ along f, that is, $\nabla\phi \cdot f$. This is again a smooth function, so one can also consider iterated applications of the operator L_f.

Theorem 3 *The system $\mathbf{x}' = f(\mathbf{x}) + g(\mathbf{x})v$ is globally feedback linearizable if and only if there exists a smooth function*

$$\varphi : \mathbb{R}^n \to \mathbb{R}$$

having everywhere nonzero gradient and satisfying the following properties:

1. *for each $\mathbf{x} \in \mathbb{R}^n$, the vectors $g(\mathbf{x}), ad_f g(\mathbf{x}), \ldots, ad_f^{n-1} g(\mathbf{x})$ are linearly independent;*
2. *for each $\mathbf{x} \in \mathbb{R}^n$ and each $j = 0, \ldots, n-2$, $\nabla\varphi(\mathbf{x}) \cdot ad_f^j g(\mathbf{x}) = 0$;*
3. *the map $\mathbf{x} \mapsto (\varphi(\mathbf{x}), L_f\varphi(\mathbf{x}), \ldots, L_f^{n-1}\varphi(\mathbf{x}))$ is a bijection $\mathbb{R}^n \to \mathbb{R}^n$.*

Observe that the conditions amount to the existence of a well-behaved solution φ of a set of first-order linear partial differential equations. Existence of a solution of this form is not trivial to verify, but the linear independence of the vectors $g(\mathbf{x}), \mathrm{ad}_f g(\mathbf{x}), \ldots, \mathrm{ad}_f^{n-1} g(\mathbf{x})$ together with the involutivity of the distribution generated by $g, \mathrm{ad}_f g, \ldots, \mathrm{ad}_f^{n-2} g$ (i.e., the Lie bracket of any two of these vectors should be, for each $\mathbf{x}$, a linear combination of these $n-1$ vectors) is a necessary condition, which is in a sense also sufficient, as follows by an application of Frobenius' Theorem. (To be precise, see Theorem 15 in [31], the conditions are only sufficient for local feedback linearization. However, it turns out often in examples that the conditions lead one to a globally defined solution, see e.g. example 5.3.10 in [31].)

Let us now show that the class $\mathcal{S}_n$ includes some fading memory systems of the form (6). Indeed, consider any system as follows:

$$\mathbf{x}' = -\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{x} + \mathbf{b} \cdot v \tag{10}$$

where the $\lambda_i \neq \lambda_j$ for each $i \neq j$ are all positive, $diag(\lambda_1, \ldots, \lambda_n)$ is the resulting diagonal matrix, and the column vector $\mathbf{b} = \mathrm{col}(b_1, \ldots, b_n)$ has nonzero entries: $b_i \neq 0$ for all i.

(Such a system, which has the form (6) with $\sigma(A\mathbf{x}) \equiv 0$, consists of n first order linear differential equations in parallel, and is obviously fading-memory.) It is easy to see that, up to signs $(-1)^i$, we have

$$\mathrm{ad}_f^i g(\mathbf{x}) = \mathrm{col}(\lambda_1^i b_1, \ldots, \lambda_n^i b_n)$$

for $i > 0$, and the linear independence of $g(\mathbf{x}), \mathrm{ad}_f g(\mathbf{x}), \ldots, \mathrm{ad}_f^{n-1} g(\mathbf{x})$ follows from the fact that these constant vectors form a Vandermonde matrix. Then we can pick $\varphi(\mathbf{x})$ as a linear map $\mathbf{x} \to \mathbf{a}\mathbf{x}$, where $\mathbf{a}$ is any vector in $\mathbb{R}^n$ which is orthogonal to all of the vectors

$$\mathrm{col}(\lambda_1^i b_1, \ldots, \lambda_n^i b_n)\,, i = 0, 1, \ldots, n-2\,.$$

The map $\mathbf{x} \mapsto (\varphi(\mathbf{x}), L_f\varphi(\mathbf{x}), \ldots, L_f^{n-1}\varphi(\mathbf{x}))$ is represented then also by a Vandermonde matrix, so it is a bijection.

Finally, let us prove the simulation result.

Take any system (3) in $\mathcal{S}_n$ and any system (4) to be simulated. Using T, α, β as in the definition of the class $\mathcal{S}_n$, we define:

$$K(\mathbf{x}, w) \;:=\; \alpha(\mathbf{x}) \;+\; \beta(\mathbf{x})\,[G(T(\mathbf{x})) + w]$$

and we let $h(\mathbf{x})$ be the first coordinate of $T(\mathbf{x})$. Note that these functions K and h are smooth functions.[3]

Next, pick an external input $u(t), t \geq 0$, and a solution $z(t)$ of the forced system (4).

From the interpretation of feedback linearization given earlier, it follows that for any inputs $u(t)$ and $u_0(t)$ (in particular, one could take $u_0 \equiv 0$), and each solution $z(t)$ of

$$z^{(n)}(t) \;=\; G\left(z(t), z'(t), z''(t), \ldots, z^{(n-1)}(t)\right) + u(t) + u_0(t)$$

(that is, we use $\mu(t) = G\left(z(t), z'(t), z''(t), \ldots, z^{(n-1)}(t)\right) + u(t) + u_0(t)$ as the input to $z^{(n)} = \mu$), the vector function $\mathbf{x}(t) = T^{-1}(Z(t))$ satisfies (3) with input

$$v(t) = \alpha(\mathbf{x}(t)) + \beta(\mathbf{x}(t))\mu(t) = K\left(\mathbf{x}(t), u(t) + u_0(t)\right)\,.$$

Furthermore, $Z(t) = T(\mathbf{x}(t))$ means that $z(t) = h(\mathbf{x}(t))$, as required for the notion of simulation.

This almost proves the simulation result, except for the fact that there is no reason for the initial value $\mathbf{x}(0) = T^{-1}(Z(0))$ to be zero, since $z(t)$ is an arbitrary trajectory. This is where the input u_0 plays a role. Let $\xi := T(0)$. We will show that, given any solution $z(t)$ and any input $u(t)$, there is some input $u_0(t)$, with $u_0(t) \equiv 0$ for all $t \geq 1$, so that the solution of

$$y^{(n)}(t) \;=\; G\left(y(t), y'(t), y''(t), \ldots, y^{(n-1)}(t)\right) + u(t) + u_0(t) \tag{11}$$

[3]In the special case where (3) describes the dynamics of a circuit according to (6), α is a linear function, β is a constant, and T is an invertible linear map from $\mathbb{R}^n$ to $\mathbb{R}^n$.

with $y(0) = \xi$ has the property that $y(t) = z(t)$ for all $t \geq 1$. (Where $z(t)$ is the desired trajectory to be simulated, with $u_0 \equiv 0$.) Then letting $\mathbf{x}(t) = T^{-1}(Y(t))$ instead of $T^{-1}(Z(t))$ means that $\mathbf{x}(0) = 0$ and still $h(\mathbf{x}(t)) = y(t) = z(t)$ for all $t \geq 1$.

Consider now an arbitrary solution $z(t)$ of the equation (4) and let ζ be the vector with entries

$$\zeta_{i+1} := z^{(i)}(1),\ i = 0, \ldots, n-1.$$

We next pick a scalar differentiable function ϕ such that $\phi^{(i)}(0) = \xi_{i+1}$ and $\phi^{(i)}(1) = \zeta_{i+1}$ for $i = 0, \ldots, n-1$. (It is easy to see that such functions exist. For example, one may simply consider the linear system $\dot{p} = A_n p + b_n q$ with states p and input q. This is a completely controllable linear system, cf. [31], Chapter 3, so we just pick an input $q(t)$ which steers ξ into ζ, and finally let $\phi(t)$ be the first coordinate of $p(t)$.) Now we let

$$u_0(t) \ := \ \phi^{(n)}(t) - G(\phi(t), \ldots, \phi^{(n-1)}(t)) - u(t)$$

for $t < 1$, and $u_0(t) \equiv 0$ for $t \geq 1$, and claim that the solution of (11) with $y(0) = \xi$ has the property that $y(t) = z(t)$ for all $t \geq 1$. Since $u(t) + u_0(t) = u(t)$ for all $t \geq 1$, we only need to show that $y^{(i)}(1) = z^{(i)}(1)$ for every $i = 0, \ldots, n-1$. To see this, in turn, and using uniqueness of solutions of differential equations, it is is enough to show that $y(t) := \phi(t)$ satisfies

$$\phi^{(n)}(t) \ = \ G\left(\phi(t), \phi'(t), \phi''(t), \ldots, \phi^{(n-1)}(t)\right) + u(t)$$

on the interval $[0, 1]$ and has derivatives at $t = 0$ as specified by the vector ξ. But this is indeed true by construction.

Finally, we remark that if $|u(t)| \leq c$ and $\left|z^{(i)}\right|(t) \leq c$ for all $t \geq 0$ then $\mathbf{x}(t) = T^{-1}(Z(t))$ is bounded in norm by a constant that only depends on c (since T^{-1} is continuous, by definition of diffeomorphism), and the numbers $b_i := z^{(i)}(1)$ are also bounded by a constant that depends only on c, so $K(\mathbf{x}(t), u(t) + u_0(t))$ also is.

Corollary 4 *Analogous results can be shown for the simulation of systems consisting of any number k of higher order differential equations as in (4). In this case fixed systems of first order differential equations of a form as in (3), but with k memoryless feedback functions $K_1, \ldots, K_k$ that depend on the simulated higher order system, can be shown to be able to simulate the dynamic response of arbitrary higher order systems of differential equations.*

4.1.2 Details to the Proof of Theorem 2

A map (or filter) F from input- to output streams is defined to have fading memory if its current output at time t depends (up to some precision ε) only on values of the input

$\mathbf{u}$ during some finite time interval $[t-T,t]$. In formulas: F has fading memory if there exists for every $\varepsilon > 0$ some $\delta > 0$ and $T > 0$ so that $|(F\mathbf{u})(t) - (F\tilde{\mathbf{u}})(t)| < \varepsilon$ for any $t \in \mathbb{R}$ and any input functions $\mathbf{u}, \tilde{\mathbf{u}}$ with $\|\mathbf{u}(\tau) - \tilde{\mathbf{u}}(\tau)\| < \delta$ for all $\tau \in [t-T,t]$.[4] This is a characteristic property of all filters that can be approximated by an integral over the input stream $\mathbf{u}$, or more generally by Volterra- or Wiener series. Note that non-trivial Turing machines and FSMs can *not* be approximated by filters with fading memory, since they require a persistent memory.

The deterministic finite state machine (FSM), also referred to as deterministic finite automaton, is a standard model for a digital computer, or more generally for any realistic computational device that operates in discrete time with a discrete set of inputs and internal states [25]. One assumes that a FSM is at any time in one of some finite number l of states, and that it receives at any (discrete) time step one input symbol from some alphabet $\{s_1, \ldots, s_k\}$ that may consist of any finite number k of symbols. Its "program" may consist of any transition function $TR : \{s_1, \ldots, s_k\} \times \{1, \ldots, l\} \to \{1, \ldots, l\}$, where $TR(s_i, j') = j$ denotes the new internal state j which the FSM assumes at the next time step after processing input symbol s_i in state j'.

We consider here a slight variation of this model, which is more adequate for systems that operate in continuous time and receive analog inputs (for example trains of spikes in continuous time). We assume that the raw input is some arbitrary n-dimensional input stream $\mathbf{u}$ (i.e., $\mathbf{u}(t) \in \mathbb{R}^n$ for every $t \in \mathbb{R}$). Furthermore we assume that there exist pattern detectors $F_1, \ldots, F_k$ that report the occurrence of spatio-temporal patterns in the input stream $\mathbf{u}$ from k different classes $C_1, \ldots, C_k$. In the case where the input $\mathbf{u}$ consists of spike trains, these classes could consist for example of particular patterns of firing rates, of particular spike patterns, or particular correlation patterns among some of the input spike trains. It was shown in [4] that readouts from generic neural microcircuit models can easily be trained to approximate the role of such pattern detectors $F_1, \ldots, F_k$. We assume that the detection of a pattern from class C_i by pattern detector F_i affects the state of the FSM according to its transition function TR in a way which corresponds to the presentation of input symbol s_i in the discrete-time version: if j' was its preceding state, then it changes now within some finite switching time to state $j = TR(s_i, j')$.

In order to make an implementation of such FSM by a noisy system feasible, we assume that the pattern detectors $(F_1\mathbf{u})(t), \ldots, (F_k\mathbf{u})(t)$ always assume values ≤ 0, except during a switching episode. During a switching episode exactly one of the pattern detectors $(F_i\mathbf{u})(t)$ assumes values > 0.[5] In order to avoid that the subsequent construction is based

[4] We use in this section boldface letters to denote input streams, because they typically have a dimension larger than 1.

[5] We assume that this $(F_i\mathbf{u})(t)$ reaches values ≥ 1 during this switching episode. We also assume that the length of each switching episode (i.e., the time during which some $(F_i\mathbf{u})(t)$ assumes values > 0) is bounded from above by some constant δ, and that the temporal distance between the beginnings of any two different switching episodes is at least $\Delta + 3\delta$ (where Δ is the assumed temporal delay of the feedback in the circuit).

on unrealistic assumptions, we allow that each pattern detector F_i is replaced by some arbitrary filter $\hat{F}_i$ so that $(\hat{F}_i\mathbf{u})(t)$ is a continuous function of time (with values in some arbitrary bounded range $[-B, B]$) with $|(\hat{F}_i\mathbf{u})(t) - (F_i\mathbf{u})(t)| \leq \frac{1}{4}$ for any input stream $\mathbf{u}$ that is considered.

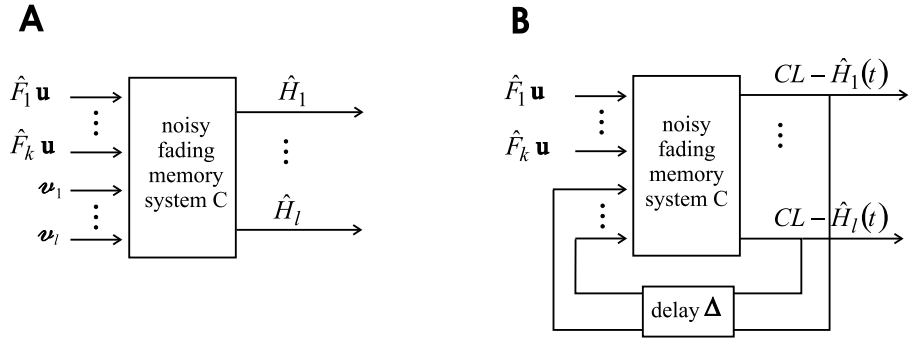

Figure 5: Emulation of a finite state machine (FSM) by a noisy fading memory system with feedback according to Theorem 5. **(A)** Underlying open loop system with noisy pattern detectors $\hat{F}_1, \ldots, \hat{F}_k$ and suitable fading memory readouts $\hat{H}_1, \ldots, \hat{H}_l$ (which may also be subject to noise). **(B)** Resulting noise-robust emulation of an arbitrary given FSM by adding feedback to the system in panel A. The same readouts as in A (denoted $CL - \hat{H}_j(t)$ in the closed loop) now encode the current state of the simulated FSM.

The informal statement of Theorem 2 is made precise by the subsequent Theorem 5 (see Fig. 5 for an illustration). It exhibits a simple construction method whereby fading memory filters with additive noise of bounded amplitude can be composed into a closed loop system C that emulates an arbitrary given FSM in a noise-robust manner. The resulting system C can be embedded into any other fading memory system, which receives the outputs $CL - \hat{H}_j(t)$ of C as additional inputs. In this way any given fading memory system can integrate the computational capability and non-fading states of the FSM that is emulated by C into its own real-time computation on time-varying input streams $\mathbf{u}$.

An essential aspect of the proof of Theorem 5 is that suitable fading memory filters H_j can prevent in the closed loop the accumulation of errors through feedback, even if the ideal fading memory filters H_j are subsequently replaced by imperfect approximations $\hat{H}_j$. One just has to construct the ideal fading memory filters H_j in such a way that they take into account that their previous outputs, that have been fed back into the system C, may have been corrupted by additive noise. As long as this additive noise of bounded amplitude has not been amplified in the closed loop, the filters H_j can still recover which of the finitely many states of the emulated FSM A was represented by that noise-corrupted feedback.

From the perspective of neural circuit models it is of interest to note that the construction of the system C can be replaced by an adaptive procedure, whereby readouts from generic cortical microcircuit models are trained to approximate the target filters H_j.

General approximation results [3, 4, 35] imply that if the neural circuit is sufficiently large and contains sufficiently diverse components (for example dynamic synapses with slightly different parameter values), then the actual outputs $\hat{H}_j$ of these readouts can approximate the target filters H_j uniformly up to any given maximal error $\varepsilon > 0$. Theorem 5 guarantees that the resulting neural circuit model with these (imperfectly) trained readouts can in the closed loop emulate the given FSM A in a reliable manner, provided that the neural circuit model is sufficiently large and diverse so that its readout can achieve an approximation error ε not larger than 1/4.

Theorem 5 *One can construct for any given finite state machine (FSM) A some time invariant fading memory filters $H_1, \ldots, H_l$ with the property that any approximating filters $\hat{H}_1, \ldots, \hat{H}_l$ with $|H_j - \hat{H}_j| \leq 1/4$ provide in the closed loop with delay Δ (see Fig. 5) outputs $CL - \hat{H}_1, \ldots, CL - \hat{H}_l$ that simulate the FSM A in the following sense:*

If $[t_1, t_2]$ is some arbitrary time interval between switching episodes of the FSM A with noise-free pattern detectors $(F_1\mathbf{u})(t), \ldots, (F_k\mathbf{u})(t)$ during which A is in state j, then the outputs $CL - \hat{H}_i(t)$ of the approximating filters $\hat{H}_i$ in the closed loop with noisy pattern detectors $(\hat{F}_1\mathbf{u})(t), \ldots, (\hat{F}_k\mathbf{u})(t)$ satisfy $CL - \hat{H}_j(t) \geq \frac{3}{4}$ and $CL - \hat{H}_{j^*}(t) \leq \frac{1}{4}$ for all $j^* \neq j$ and all $t \in [t_1, t_2]$.

Proof: In order to prove that the given FSM A can be implemented in a noise robust fashion, we construct suitable time invariant fading memory filters $H_1, \ldots, H_l$. They receive as inputs the time-varying functions $(\hat{F}_1\mathbf{u})(t), \ldots, (\hat{F}_k\mathbf{u})(t)$. In addition they receive in the open loop inputs $v_1(t), \ldots, v_l(t)$, where each $v_j(t)$ will be replaced by a delayed version of the output of H_j (or $\hat{H}_j$) in the closed loop (see Fig. 5). The filters H_j will be defined in such a way that $H_j(t) \geq 1$ signals in the closed loop that the FSM A is at time t in state j. To make this implementation noise robust, we make sure that even if one replaces the filters H_j by noisy approximations $\hat{H}_j$ which satisfy in the open loop $|H_j(t) - \hat{H}_j(t)| \leq \frac{1}{4}$ (for all $t \in \mathbb{R}$ and any time varying inputs $(\hat{F}_1\mathbf{u})(t), \ldots, (\hat{F}_k\mathbf{u})(t)$ and $v_1(t), \ldots, v_l(t)$), then the closed loop version of such imperfect approximations $\hat{H}_j$ simulates the FSM A in such a way that $\hat{H}_j(t) \geq \frac{3}{4}$ implies that A is in state j at time t.

Let Δ be the time delay in the feedback for the closed-loop. We now define the target outputs $H_1(t), \ldots, H_l(t)$ (for the open loop version, where the H_j receive in addition to $(\hat{F}_1\mathbf{u})(t), \ldots, (\hat{F}_k\mathbf{u})(t)$ some arbitrary time-varying variables $v_1(t), \ldots, v_l(t)$ with values in $[-1, 2]$ as inputs). We define the target outputs of $H_1, \ldots, H_l$ as a stationary transformation of the time-varying inputs $v_j(t)$ and of the outputs of the following two other types of time invariant fading memory filters:

(i) $f_i(t) := \max\{(\hat{F}_i\mathbf{u})(\tau) : t - \Delta - \delta \leq \tau \leq t\}$ for $i = 1, \ldots, k$

(ii) $v_j(t - 2\delta)$ for $j = 1, \ldots, l$.

We will show below in Lemma 6 and Lemma 7 that both of these functions of time can be viewed as outputs of time invariant fading memory filters that receive as inputs the time-varying functions $(\hat{F}_i\mathbf{u})(t)$ (for some arbitrary input stream $\mathbf{u}$) and $v_j(t)$. On the basis of these two Lemmata it is clear that the H_j are time invariant fading memory filters if one can define $H_1(t), \ldots, H_l(t)$ as (static) continuous functions of the variables $v_j(t)$ and the outputs of the filters (i) and (ii).[6] In order to define such functions $H_j(t)$ we first define for each $j \in \{1, \ldots, l\}$ two disjoint closed and bounded sets $S_{j,0}, S_{j,1} \subseteq \mathbb{R}^{k+2l}$, and we set $H_j(\mathbf{x}) = 0$ for $\mathbf{x} \in S_{j,0}$ and $H_j(\mathbf{x}) = 1$ for $\mathbf{x} \in S_{j,1}$. Since the sets $S_{j,0}$ and $S_{j,1}$ will have positive distance (i.e., $\inf\{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x} \in S_{j,0} \text{ and } \mathbf{y} \in S_{j,1}\} > 0$), it follows from standard arguments of analysis that the definition of H_j can be continued outside of $S_{j,0}, S_{j,1}$ to yield a continuous function from $\mathbb{R}^{k+2l}$ into $\mathbb{R}$.

In order to define the sets $S_{j,0}, S_{j,1}$ we consider the following two types of conditions:

(A_j) There exist $i \in \{1, \ldots, k\}$ and $j' \in \{1, \ldots, l\}$ so that $TR(i, j') = j$,
$f_i(t) \geq \frac{3}{4}$ and $f_{i'}(t) \leq \frac{1}{4}$ for all $i' \neq i$,
$v_{j'}(t - 2\delta) \geq \frac{3}{4}$ and $v_{j^*}(t - 2\delta) \leq \frac{1}{4}$ for all $j^* \neq j'$.

(B_j) $f_i(t) \leq \frac{1}{4}$ for $i = 1, \ldots, k$, $v_j(t) \geq \frac{3}{4}$ and $v_{j^*}(t) \leq \frac{1}{4}$ for all $j^* \neq j$.

We say that a vector $(f_1(t), \ldots, f_k(t), v_1(t), \ldots, v_l(t), v_1(t - 2\delta), \ldots, v_l(t - 2\delta)) \in [-B, B]^k \times [-1, 2]^{2l}$ belongs to set $S_{j,1}$ if the conditions A_j or B_j apply, and to set $S_{j,0}$ if there exists some $j^* \neq j$ so that the conditions A_{j^*} or B_{j^*} apply.

It follows immediately from the definition of the sets $S_{j,0}$ and $S_{j,1}$ that they are closed and bounded. One can also verify immediately that for any $j, j' \in \{1, \ldots, l\}$ the conditions A_j and $B_{j'}$ can never be simultaneously satisfied (for any values of the variables $f_i(t), v_j(t), v_j(t - 2\delta)$). In addition the conditions A_j and $A_{j'}$ (B_j and $B_{j'}$) can never be simultaneously satisfied for any $j \neq j'$. This implies that the sets $S_{j,0}$ and $S_{j,1}$ are disjoint for each $j \in \{1, \ldots, l\}$.

We define for each $j \in \{1, \ldots, l\}$ a continuous function $H_j : \mathbb{R}^{k+2l} \to [0, 1]$ by setting

$$H_j(\mathbf{x}) := \begin{cases} 1 & , \quad \text{if } dist(\mathbf{x}, S_{j,0}) \geq dist(S_{j,0}, S_{j,1}) \\ dist(\mathbf{x}, S_{j,0})/dist(S_{j,0}, S_{j,1}) & , \quad \text{otherwise} \ , \end{cases}$$

where $dist(\mathbf{x}, S) := \inf\{\|\mathbf{x} - \mathbf{y}\| : y \in S\}$ for any set $S \subseteq \mathbb{R}^{k+2l}$. It is then obvious that H_j is a continuous function from $\mathbb{R}^{k+2l}$ into $[0, 1]$ with $H_j(\mathbf{x}) = 0$ for all $\mathbf{x} \in S_{j,0}$ and $H_j(\mathbf{x}) = 1$ for all $\mathbf{x} \in S_{j,1}$. These functions H_j will prevent the amplification of noise in the closed loop, since they assume outputs 1 or 0 in all relevant situations, even if their inputs deviate by up to $\frac{1}{4}$ from their "ideal" values.

We consider some arbitrary imprecise and/or noisy versions $\hat{H}_j$ of these filters H_j (with inputs $(\hat{F}_1\mathbf{u})(t), \ldots, (\hat{F}_k\mathbf{u})(t)$ and additional inputs $v_1(t), \ldots, v_l(t)$) whose output differs

[6] In the following we sometimes refer to $H_1, \ldots, H_l$ as static functions of input vectors $(f_1(t), \ldots, f_k(t), v_1(t), \ldots, v_l(t), v_1(t-2\delta), \ldots, v_l(t-2\delta))$ from $\mathbb{R}^{k+2l}$, and sometimes as filters with time-varying inputs $\hat{F}_i\mathbf{u}$ and v_j (if we view the filters (i) and (ii) as being part of the computation of H_j).

at any time t by at most $\frac{1}{4}$ from that of H_j (of course in the closed loop these deviations could be accumulated and amplified to values $> \frac{1}{4}$). We want to show that for any such $\hat{H}_1, \ldots, \hat{H}_l$ the closed loop version of the circuit implements the given FSM A. As initial condition we assume that the given FSM A is in state 1 for $t \leq 0$, and consequently also that $\hat{H}_1(t) \geq \frac{3}{4}$ and $\hat{H}_j(t) \leq \frac{1}{4}$ for $j = 2, \ldots, l$, as well as $f_i(t) \leq \frac{1}{4}$ for all $t \leq 0$ and $i = 1, \ldots, k$.

We will now prove the claim of Theorem 5 for arbitrary time intervals $[t_1, t_2]$ outside of switching episodes. We assume without loss of generality that t_2 marks the beginning of the next switching episode $[t_2, t_3]$ for some $t_3 > t_2$ with $|t_3 - t_2| \leq \delta$. Furthermore we assume that either $t_1 = 0$ (Case 1), or t_1 is the endpoint of the preceding switching episode $[t_0, t_1]$ with $|t_1 - t_0| \leq \delta$ (Case 2). The formal proof is carried out by induction on the number of preceding switching episodes (and Case 2 represents the induction step). In both cases one just needs to analyze the outputs of the previously defined filters $\hat{H}_j(t)$ in the case where some of their inputs are delayed feedbacks of their previous outputs.

Case 1: $t_1 = 0$

We prove by a nested induction on $m \in \mathbb{N}$ that $CL - \hat{H}_1(t) \geq \frac{3}{4}$ and $CL - \hat{H}_j(t) \leq \frac{1}{4}$ for all $j > 1$ holds for all $t \in [m \cdot \Delta, (m+1) \cdot \Delta) \cap [t_1, t_2]$. Since by assumption no switching episode occurs during $[t_1, t_2]$, one has $f_i(t) \leq \frac{1}{4}$ for $i = 1, \ldots, k$ and for all $t \in [t_1, t_2]$. Furthermore by our assumption on the initial condition of the FSM A (for $m = 0$), or by the induction hypothesis of the nested induction (for $m > 0$) we can assume that the variables $v_j(t)$ of the open loop have now been assigned in the closed loop the values $CL - \hat{H}_j(t - \Delta)$, therefore they are $\geq \frac{3}{4}$ for $j = 1$ and $\leq \frac{1}{4}$ for all $j > 1$. Hence condition B_1 in the definition of the sets $S_{j,0}, S_{j,1}$ applies, and the current circuit input is therefore in $S_{1,1}$. Thus $H_1 = 1$ and $H_j = 0$ for $j > 1$, which implies $\hat{H}_1 \geq \frac{3}{4}$ and $\hat{H}_j \leq \frac{1}{4}$ for $j > 1$ in the open loop, hence $CL - \hat{H}_1(t) \geq \frac{3}{4}$ and $CL - \hat{H}_j(t) \leq \frac{1}{4}$ for $j > 1$ in the closed loop (since $v_j(t) = CL - \hat{H}_j(t - \Delta)$ in the closed loop).

Case 2: t_1 **is the endpoint of a preceding switching episode** $[t_0, t_1]$**.**

Assume that $(\hat{F}_i \mathbf{u})(t)$ is the (approximating) pattern detector that assumes a value $\geq \frac{3}{4}$ during the preceding switching episode $[t_0, t_1]$, while $(\hat{F}_{i'} \mathbf{u})(t) \leq \frac{1}{4}$ for all $i' \neq i$ during $[t_0, t_1]$. Let $t' \in [t_0, t_1]$ be the first time point where $(\hat{F}_i \mathbf{u})(t)$ reaches a value $\geq \frac{3}{4}$. Then $f_i(t) \geq \frac{3}{4}$ and $f_{i^*}(t) \leq \frac{1}{4}$ for all $i^* \neq i$ and for all $t \in [t', t' + \Delta + \delta]$ (by the definition of the filters $f_i(t)$). Furthermore one has by the induction hypothesis that for the state j' in which the FSM A was before the switching episode $[t_0, t_1]$ that $CL - \hat{H}_{j'}(t - \Delta - 2\delta) \geq \frac{3}{4}$ and $CL - \hat{H}_{j^*}(t - \Delta - 2\delta) \leq \frac{1}{4}$ for all $j^* \neq j'$ and all $t \in [t', t' + \Delta + 2\delta]$. We exploit here that $t_0 \leq t' \leq t_1 \leq t_0 + \delta$, hence $t_0 - \Delta - 2\delta \leq t - \Delta - 2\delta \leq t_0$ for all $t \in [t', t' + \Delta + \delta]$. Furthermore we have assumed that the minimal distance between the beginnings of switching episodes is $\Delta + 3\delta$. Therefore the considered range $[t_0 - \Delta - 2\delta, t_0]$ for $t - \Delta - 2\delta$ is contained in the preceding time interval before the switching episode $[t_0, t_1]$ to which the induction

hypothesis applies.

The previously listed conclusions imply that for $t \in [t', t' + \Delta + \delta]$ the current input to the open loop lies in the set $S_{j,1}$ for $j = TR(i, j')$, hence $H_j = 1$ and $\hat{H}_j \geq \frac{3}{4}$, while $H_{j^*} = 0$ and $\hat{H}_{j^*} \leq \frac{1}{4}$ for all other j^*. But if one chooses as inputs $v_1(t), \ldots, v_l(t)$ to the open loop just those values which the circuit receives in the closed loop, one gets that $CL - \hat{H}_j(t) \geq \frac{3}{4}$ and $CL - \hat{H}_{j^*}(t) \leq \frac{1}{4}$ for all $j^* \neq j$ and all $t \in [t', t' + \Delta + \delta]$, in particular for all $t \in [t_1, t_1 + \Delta]$.

One can then prove by a nested induction on $m \in \mathbb{N}$ like in Case 1 that the outputs $CL - \hat{H}_{j^*}(t)$ for $j^* = 1, \ldots, l$ have the desired values for $t \in [t_1 + m\Delta, t_1 + (m+1) \cdot \Delta] \cap [t_1, t_2]$. The preceding argument provides the verification of the claim for the initial step $m = 0$ of this nested induction.

In order to complete the proof of Theorem 5 it only remains to verify the following two simple facts about time invariant fading memory filters.

Lemma 6 *Assume that $\hat{F}_i$ is some arbitrary time invariant fading memory filter, and Δ, δ are arbitrary positive constants. Then the map which assigns to an input stream $\mathbf{u}$ the function $f_i(t) := \max\{(\hat{F}_i\mathbf{u})(\tau) : t - \Delta - \delta \leq \tau \leq t\}$ is also a time invariant fading memory filter.*

Proof of Lemma 6: Assume some $\varepsilon > 0$ is given. Fix δ' and $T > 0$ so that $|(\hat{F}_i\mathbf{u})(\tau) - (\hat{F}_i\mathbf{v})(\tau)| < \varepsilon$ for all $\tau \in [t - \Delta - \delta, t]$ and all $\mathbf{u}, \mathbf{v}$ with $\|\mathbf{u}(s) - \mathbf{v}(s)\| < \delta'$ for all $s \in [t - \Delta - \delta - T, t]$. Then $|\max\{(\hat{F}_i\mathbf{u})(\tau) : t - \Delta - \delta \leq \tau \leq t\} - \max\{(\hat{F}_i\mathbf{v})(\tau) : t - \Delta - \delta \leq \tau \leq t\}| < \varepsilon$. ∎

Lemma 7 *The filter which maps for some arbitrary fixed $\delta > 0$ the function $u(t)$ onto the function $u(t - 2\delta)$ is time invariant and has fading memory.*

Proof of Lemma 7: Follows immediately from the definitions (choose $T \geq 2\delta$ in the condition for fading memory). ∎

This completes the proof of Theorem 5, which shows that any given FSM can be reliably implemented by fading memory filters with feedback even in the presence of noise. ∎

Remark: In the application of this theory to cortical microcircuit models we train readouts from such circuits to *simultaneously* assume the role of the pattern detectors $\hat{F}_1, \ldots, \hat{F}_k$, which become active if some pattern occurs in the input stream that may trigger a state change of the simulated FSM A, *and* the role of the fading memory filters

$\hat{H}_1, \ldots, \hat{H}_l$, that create high-dimensional attractors of the circuit dynamics that represent the current state of the FSM A.

4.2 Details of Computer Simulations of Cortical Microcircuit Models

I&F neurons were chosen with a time constant of 30 ms, which subsumes the time constants of synaptic receptors as well as the time constant of the neuron membrane. Other parameters: absolute refractory period 3 ms (excitatory neurons), 2 ms (inhibitory neurons), threshold 15 mV (for a resting membrane potential assumed to be 0), reset voltage drawn uniformly from the interval [13.8, 14.5 mV] for each neuron, input resistance 1 MΩ, constant non-specific background current I_b uniformly drawn from the interval [13.5 nA, 14.5 nA] for each neuron, an additional time-varying noise input current I_{noise} was drawn every 5 ms from a Gaussian distribution with mean 0 and SD chosen for each neuron randomly from the uniform distribution over the interval [4.0 nA, 5.0 nA]. For each simulation, the initial condition of each I&F neuron, i.e., its membrane voltage at time $t = 0$, was drawn randomly (uniform distribution) from the interval [13.5 mV, 14.9 mV].

HH-neurons: We chose conductance based single compartment HH neuron models with passive and active properties modeled according to [47, 48]. A cortical neuron receives synaptic inputs not only from immediately adjacent neurons (which were modeled explicitly in our computer model), but also smaller background input currents from a large number of more distal neurons, causing a depolarization of the membrane potential and a lower input resistance commonly referred to as 'high conductance state' (for a review see [40]). This was reflected in our computer model of HH-neurons by background input currents that were injected into each neuron.

In accordance with experimental data on neocortical and hippocampal pyramidal neurons ([49–52]) the active currents in the HH neuron model comprise a voltage dependent Na^+ current ([53]) and a delayed rectifier K^+ current ([53]). For excitatory neurons a non-inactivating K^+ current ([54]) responsible for spike frequency adaption was included in the model. The peak conductance densities for the Na^+ current and delayed rectifier K^+ current were chosen to be $500pS/\mu m^2$ and $100pS/\mu m^2$ respectively, and the peak conductance density for the non-inactivating K^+ current was chosen to be $5pS/\mu m^2$. The membrane area of the neuron was set to be 34636 μm^2 as in [47]. For each simulation the initial conditions of each neuron, i.e. the membrane voltage at time $t = 0$, were drawn randomly (uniform distribution) from the interval [-70, -60] mV.

The conductances of **background input currents** to each neuron were modeled according to [47] as a one-variable stochastic process similar to an Ornstein-Uhlenbeck process with mean $g_e = 0.012\mu$S and $g_i = 0.057\mu$S, variance $\sigma_e = 0.003\mu$S and $\sigma_i = 0.0066\mu$S,

and time constants $\tau_e = 2.7$ ms and $\tau_i = 10.5$ ms, where the indices e/i refer to excitatory and inhibitory background input conductances, respectively. According to [47] this model captures the spectral and amplitude characteristics of the input conductances of a detailed biophysical model of a neocortical pyramidal cell that was matched to intracellular recordings in cat parietal cortex *in vivo*. Furthermore the ratio of the average contributions of excitatory and inhibitory background conductances was chosen to be 5 in accordance with experimental studies during sensory responses [55–57]. The maximum conductances of the synapses were chosen from a Gaussian distribution with a SD of 70% of its mean (with negative values replaced by values chosen from an uniform distribution between 0 and two times the mean).

We modeled the (short term) **dynamics of synapses** according to the model proposed in [41], with the synaptic parameters U (use), D (time constant for depression), F (time constant for facilitation) randomly chosen from Gaussian distributions that model empirically found data for such connections (see supplementary information). This model predicts the amplitude A_k of the EPSC for the k^{th} spike in a spike train with interspike intervals $\Delta_1, \Delta_2, \ldots, \Delta_{k-1}$ through the equations

$$\begin{aligned} A_k &= w \cdot u_k \cdot R_k \\ u_k &= U + u_{k-1}(1-U)exp(-\Delta_{k-1}/F) \\ R_k &= 1 + (R_{k-1} - u_{k-1}R_{k-1} - 1)exp(-\Delta_{k-1}/D) \end{aligned}$$

with hidden dynamic variables $u \in [0,1]$ and $R \in [0,1]$ whose initial values for the first spike are $u_1 = U$ and $R_1 = 1$ (see [58] for a justification of this version of the equations, which corrects a small error in [41]).

Synaptic parameters: Depending on whether a and b were excitatory (E) or inhibitory (I), the mean values of the three parameters U, D, F (with D,F expressed in seconds, s) were chosen according to [42] to be .5, 1.1, .05 (EE), .05, .125, 1.2 (EI), .25, .7, .02 (IE), .32, .144, .06 (II). The SD of each of these parameters was chosen to be 50% of its mean. The mean of the scaling parameter w (in nA) was chosen to be 70 (EE), 150 (EI), -47 (IE), -47 (II). In the case of input synapses the parameter w had a value of 70 nA if projecting onto a excitatory neuron and -47 nA if projecting onto an inhibitory neuron. The SD of the parameter w was chosen to be 70% of its mean and was drawn from a gamma distribution. The postsynaptic current was modeled by an exponential decay $\exp(-t/\tau_s)$ with $\tau_s = 3$ ms ($\tau_s = 6$ ms) for excitatory (inhibitory) synapses. The transmission delays between neurons were chosen uniformly to be 1.5 ms (EE), and 0.8 ms for the other connections.

As input $\mathbf{x}(t)$ to linear **readout neurons** we used low-pass filtered versions (linear filter with an exponential decay of 30 ms that qualitatively reflects time constants of synaptic receptors and the membrane of a generic readout neuron) of spike trains from

neurons in the circuit. Readout functions were modeled by weighted sums $\mathbf{w} \cdot \mathbf{x}$, whose weights $\mathbf{w}$ were trained to minimize the mean squared error with regard to a desired target output function. During training the feedback from readouts was replaced by noisy variations of their target outputs. After training the weights $\mathbf{w}$ were fixed, and the performance of the otherwise generic circuit was evaluated for new input streams that had not been used for training.

The synaptic weights $\mathbf{w}$ of readout neurons were computed by linear regression to minimize the normalized root mean square error (NRMSE) with regard to a specific target signal $f(\mathbf{x}(t))$ (which is described for each case in the text or figure legends) for a series of randomly generated circuit input streams $\mathbf{u}$ of length up to 1 second. Up to 200 such test inputs $\mathbf{u}$ were used for training, amounting to at most 200 seconds of simulated biological time for training the readouts.

The same performance measure NRMSE had previously been used in [16]. In the case where the circuit was simulated for T ms of biological time, the outputs $\hat{f}(\mathbf{x}(t))$ of readouts were sampled every ΔT ms, and compared with the target output $f(\mathbf{x}(t))$ (we chose $T = 1000$ ms and $\Delta T = 5$ ms in Fig. 2 and 3, $T = 700$ ms and $\Delta T = 2$ ms in Fig. 4). The NRMSE was defined by

$$\left(\sum_{i=1}^{T/\Delta T} \frac{(\hat{f}(\mathbf{x}(i \cdot \Delta T)) - f(\mathbf{x}(i \cdot \Delta T)))^2}{(T/\Delta T) \cdot \sigma^2} \right)^{1/2},$$

where σ^2 was the variance of the target signal $f(\mathbf{x}(t))$.

All simulations were carried out with the software package CSIM [59], which is freely available from http://www.lsm.tugraz.at. It uses a C^{++}-kernel with Matlab interfaces for input generation and data analysis. As simulation time step we chose 0.5 ms.

4.3 Technical Details to Figure 2

4 randomly generated test input streams, each consisting of 8 spike trains (see Fig. 2A), were injected into 4 disjoint (but interconnected) subsets of $5 \times 5 \times 5 = 125$ neurons in the circuit consisting of 600 neurons. Feedbacks from readouts were injected into the remaining 100 neurons of the circuit. The set of 100 neurons for which the firing activity is shown in Fig. 2C contained 20 neurons from each of the resulting 5 subsets of the circuit.

Generation of input streams for training and testing: The time-varying firing rate $r_i(t)$ of the 8 Poisson spike trains that represented input stream i was chosen as follows. The baseline firing rate for streams 1 and 2 (see the lower half of Fig. 2A) was chosen to be 5 Hz, with randomly distributed bursts of 120 Hz for 50 ms. The rates for the Poisson processes that generated the spike trains for input streams 3 and 4 were periodically drawn randomly from the two options 30 Hz and 90 Hz. The actual firing rates (i.e. spike counts within a 30 ms window) resulting from this procedure are plotted in Fig. 2B.

In order to demonstrate that readouts that send feedback into the circuit can just as well represent neurons *within* the circuit, we had chosen the readout neurons that send feedback to be I&F neurons with noise, like the other neurons in the circuit. Each of them received synaptic inputs from a slightly different randomly chosen subset of neurons within the circuit. Furthermore the signs of weights of these synaptic connections were restricted to be positive (negative) for excitatory (inhibitory) presynaptic neurons.

The 8 readout neurons that provided feedback were trained to represent in their firing activity at any time the information in which of input streams 1 or 2 a burst had most recently occurred. If it occurred most recently in input stream 1, they were trained to fire at 40 Hz, and they were trained not to fire whenever a burst had occurred most recently in input stream 2. The training time was 200 s (of simulated biological time). After training, their output was correct 86% of the time (average over 50 s of test inputs; counting the high-dimensional attractor as being in the on-state if the average firing rate of the 8 readout neurons was above 34 Hz). It was possible to train these readout neurons to acquire such persistent firing behavior, although they only received input from a circuit with fading memory, because they were actually trained to acquire the following behavior: fire whenever the rate in input stream 1 becomes higher than 30 Hz, or if one can detect in the current state $\mathbf{x}(t)$ of the circuit traces of recent high feedback values, provided the rate of input stream 2 stayed below 30 Hz. Obviously this definition of the learning target for readout neurons only requires a fading memory of the circuit.

The trained readouts achieved in 50 tests for new inputs over 1 s (that had been generated by the same distribution as the training inputs, see the preceding description) the following average performance:

Task of panel E:	NRMSE	=	0.0411
Task of panel F:	NRMSE	=	0.0586
Task of panel G:	NRMSE	=	0.0387 .

4.4 Technical Details to Figure 3

The same circuit as for Fig. 2 was used. First 2 linear readouts with feedback were simultaneously trained to become highly active after the occurrence of the cue in the spike input, and then to linearly reduce their activity, but each within a different time span (400 versus 600 ms). Their feedback into the circuit consisted of 2 time-varying analog values (representing time-varying firing rates of 2 population of neurons), which were both injected (with randomly chosen amplitudes) into the same subset of 350 neurons in the circuit. Their weights $\mathbf{w}$ were trained by linear regression for a total training time of 120 s (of simulated biological time), consisting of 120 runs of length 1 s with randomly generated input-cues (a burst at 200 Hz for 50 ms) and noise inputs (5 spike trains at 10 Hz).

4.5 Technical Details to Figure 4

Time-varying firing rates for the two input streams (consisting each of 8 Poisson spike trains) were drawn randomly from values between 10 and 90 Hz. The 16 spike trains from the 2 input streams, as well as feedback from trained readouts were injected into randomly chosen subsets of neurons. In contrast to the experiment for Fig. 2, these circuit inputs were not injected into spatially concentrated clusters of neurons, but to a sparsely distributed subset of neurons scattered throughout the 3-dimensional circuit. As a consequence, the firing activity $CA(t)$ of the high-dimensional attractor (see Fig. 4D) cannot be readily detected from the spike raster in Fig. 4C. Both the linear readout that sends feedback, and subsequently the other two linear readouts (whose output for a test input to the circuit is shown in Fig. 4E,F), were trained by linear regression during 140 s of simulated biological time.

Average performance of linear readouts on 50 new test inputs of length 700 ms (that had been generated from the same distribution as the training inputs):

$$\begin{array}{lllll} \text{Task of panel D:} & \text{NRMSE} & = & 0.0591 \\ \text{Task of panel E:} & \text{NRMSE} & = & 0.0413 \\ \text{Task of panel F:} & \text{NRMSE} & = & 0.0434\ . \end{array}$$

5 Acknowledgments

Comments from Wulfram Gerstner, Stefan Haeusler, Herbert Jaeger, Konrad Koerding, Henry Markram, Gordon Pipa, Misha Tsodyks, and Tony Zador are gratefully acknowledged. Written under partial support by the Austrian Science Fund FWF, project # P15386 and # S9102-N04, project # IST2002-506778 (PASCAL) and project # FP6-015879 (FACETS) of the European Union.

References

[1] R. J. Douglas, C. Koch, M. Mahowald, K. Martin, and H. Suarez. Recurrent excitation in neocortical circuits. *Science*, 269(5226):981–985, 1995.

[2] D. V. Buonomano and M. M. Merzenich. Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, 267:1028–1030, Feb. 1995.

[3] W. Maass and E. D. Sontag. Neural systems as nonlinear filters. *Neural Computation*, 12(8):1743–1772, 2000.

[4] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[5] S. Häusler and W. Maass. A statistical analysis of information processing properties of lamina-specific cortical microcircuit models. *submitted for publication*, 2005.

[6] A. Destexhe and E. Marder. Plasticity in single neuron and circuit computations. *Nature*, 431:789–795, 2004.

[7] W. Maass, T. Natschläger, and H. Markram. Fading memory and kernel properties of generic cortical microcircuit models. *J. of Physiology (Paris)*, 2005. in press.

[8] M. I. Leon and M. N. Shadlen. Representation of time by neurons in the posterior parietal cortex of the macaque. *Neuron*, 38(2):317–322, 2003.

[9] K. Hikosaka and M. Watanabe. Delay activity of orbital and lateral prefrontal neurons of the monkey varying with different rewards. *Cerebral Cortex*, 10(3):263–267, 2000.

[10] L. Tremblay and W. Schultz. Modifications of reward expectation-related neuronal activity during learning in primate orbitofrontal cortex. *J Neurophysiol*, 83(4):1877–1885, 2000.

[11] W. Schultz, L. Tremblay, and J. R. Hollerman. Changes in behavior-related neuronal activity in the striatum during learning. *Trends Neurosci*, 26(6):321–328, 2003.

[12] X. J. Wang. Synaptic reverberation underlying mnemonic persistent activity. *Trends Neurosci.*, 24(8):455–463, 2001.

[13] M. E. Mazurek, J. D. Roitman, J. Ditterich, and M. N. Shadlen. A role for neural integrators in perceptual decision making. *Cerebral Cortex*, 13(11):1257–1269, 2003.

[14] G. Major, R. Baker, E. Aksay, B. Mensh, H. S. Seung, and D. W. Tank. Plasticity and tuning by visual feedback of the stability of a neural integrator. *Proc Natl Acad Sci*, 101(20):7739–7744, 2004.

[15] M.N. Shadlen and J.I. Gold. The neurophysiology of decision-making as a window on cognition. In M. S. Gazzaniga, editor, *The Cognitive Neurosciences*, pages 1229–1241. MIT Press, 3^{rd} edition, 2005.

[16] H. Jäger and H. Haas. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004.

[17] P. Joshi and W. Maass. Movement generation with circuits of spiking neurons. *Neural Computation*, 17(8):1715–1738, 2005.

[18] E. L. White. *Cortical Circuits*. Birkhaeuser, Boston, 1989.

[19] O. Sporns and R. Kötter. Motifs in brain networks. *PLOS Biology*, 2:1910–1918, 2004.

[20] J. D. Cowan. Statistical mechanics of neural nets. In E. R. Caianiello, editor, *Neural Networks*, pages 181–188. Springer, Berlin, 1968.

[21] M. A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. Systems, Man, and Cybernetics*, 13:815–826, 1983.

[22] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. USA*, 81:3088–3092, 1984.

[23] P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT-Press, 2001.

[24] R. A. Legenstein and W. Maass. Edge of chaos and prediction of computational power for neural microcircuit models. *submitted for publication*, 2005.

[25] J. E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley (Reading, MA), 1998.

[26] W. Maass and H. Markram. Theory of the computational function of microcircuit dynamics. In S. Grillner and A. M. Graybiel, editors, *The Interface between Neurons and Global Brain Function*, Dahlem Workshop Report 93. MIT Press, 2005. in press.

[27] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138:67–100, 1995.

[28] H. Siegelmann and E. D. Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131(2):331–360, 1994.

[29] H. Siegelmann and E. D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50:132–150, 1995.

[30] P. Orponen. A survey of continuous-time computation theory. In D.-Z. Du and K.-I. Ko, editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, 1997.

[31] E. D. Sontag. *Mathematical Control Theory*. Springer-Verlag, 1998.

[32] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing, New York, 1994.

[33] W. Maass and P. Orponen. On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10:1071–1095, 1998.

[34] W. Maass and E. Sontag. Analog neural nets with Gaussian or other common noise distribution cannot recognize arbitrary regular languages. *Neural Computation*, 11:771–782, 1999.

[35] W. Maass and H. Markram. On the computational power of recurrent circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593–616, 2004.

[36] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[37] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 79:2554–2558, 1982.

[38] D. J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex*, 7(3):237–252, 1997.

[39] C. D. Brody, R. Romo, and A. Kepecs. Basic mechanisms for graded persistent activity: discrete attractors, continuous attractors, and dynamic representations. *Curr Opin Neurobiol*, 13(2):204–211, 2003.

[40] A. Destexhe, M. Rudolph, and D. Pare. The high-conductance state of neocortical neurons in vivo. *Nat. Rev. Neurosci.*, 4(9):739–751, 2003.

[41] H. Markram, Y. Wang, and M. Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *PNAS*, 95:5323–5328, 1998.

[42] A. Gupta, Y. Wang, and H. Markram. Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287:273–278, 2000.

[43] G. Major, R. Baker, E. Aksay, H. S. Seung, and D. W. Tank. Plasticity and tuning of the time course of analog persistent firing in a neural integrator. *Proc Natl Acad Sci*, 101(20):7745–7750, 2004.

[44] R. A. Legenstein, C. Näger, and W. Maass. What can a neuron learn with spike-timing-dependent plasticity? *Neural Computation*, 17(11):2337–2382, 2005.

[45] J. Wickens and R. Kötter. Cellular models of reinforcement. In J. C. Houk, J. L. Davis, and D. G. Beiser, editors, *Models of Information Processing in the Basal Ganglia*. MIT Press, Cambridge, 1998.

[46] A. Yesidirek and F.L. Lewis. Feedback linearization using neural networks. *Automatica*, 31:1659–1664, 1995.

[47] A. Destexhe, M. Rudolph, J. M. Fellous, and T. J. Sejnowski. Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience*, 107(1):13–24, 2001.

[48] A. Destexhe and D. Pare. Impact of network activity on the integrative properties of neocortical pyramidal neurons in vivo. *J. Neurophysiol.*, 81(4):1531–1547, 1999.

[49] D. A. Hoffman, J. C. Magee, C. M. Colbert, and D. Johnston. K+ channel regulation of signal propagation in dendrites of hippocampal pyramidal neurons. *Nature*, 387(6636):869–875, 1997.

[50] J. C. Magee and D. Johnston. Characterization of single voltage-gated Na+ and Ca2+ channels in apical dendrites of rat CA1 pyramidal neurons. *J. Physiol.*, 487 (Pt 1):67–90, 1995.

[51] J. Magee, D. Hoffman, C. Colbert, and D. Johnston. Electrical and calcium signaling in dendrites of hippocampal pyramidal neurons. *Annu. Rev. Physiol.*, 60:327–346, 1998.

[52] G. J. Stuart and B. Sakmann. Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature*, 367(6458):69–72, 1994.

[53] R. D. Traub and R. Miles. *Neuronal Networks of the Hippocampus*. Cambridge Univ. Press, Cambridge, UK, 1991.

[54] Z. T. Mainen, J. Joerges, J. R. Huguenard, and T. J. Sejnowski. A model of spike initiation in neocortical pyramidal neurons. *Neuron*, 15(6):1427–1439, 1995.

[55] J. Anderson, I. Lampl, I. Reichova, M. Carandini, and D. Ferster. Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex. *Nature Neuroscience*, 3(6):617–621, 2000.

[56] L. J. Borg-Graham, C. Monier, and Y. Fregnac. Visual input evokes transient and strong shunting inhibition in visual cortical neurons. *Nature*, 393(6683):369–373, 1998.

[57] J. A. Hirsch, J. M. Alonso, R. C. Reid, and L. M. Martinez. Synaptic integration in striate cortical simple cells. *J. Neurosci.*, 18(22):9517–9528, 1998.

[58] W. Maass and H. Markram. Synapses as dynamic memory buffers. *Neural Networks*, 15:155–161, 2002.

[59] T. Natschläger, H. Markram, and W. Maass. Computer models and analysis tools for neural microcircuits. In R. Kötter, editor, *Neuroscience Databases. A Practical Guide*, chapter 9, pages 123–138. Kluwer Academic Publishers (Boston), 2003.