Sensitive finite state computations using a distributed network with a noisy network attractor

Peter Ashwin and Claire Postlethwaite

***Abstract*—We exhibit a class of smooth continuous-state neural-inspired networks composed of simple nonlinear elements that can be made to function as a finite state computational machine. We give an explicit construction of arbitrary finite-state virtual machines in the spatio-temporal dynamics of the network. The dynamics of the functional network can be completely characterised as a "noisy network attractor" in phase space operating in either an "excitable" or a "free-running" regime, respectively corresponding to excitable or heteroclinic connections between states. The regime depends on the sign of an "excitability parameter". Viewing the network as a nonlinear stochastic differential equation where deterministic (signal) and/or stochastic (noise) input are applied to any element, we explore the influence of signal to noise ratio on the error rate of the computations. The free-running regime is extremely sensitive to inputs: arbitrarily small amplitude perturbations can be used to perform computations with the system as long as the input dominates the noise. We find a counter-intuitive regime where increasing noise amplitude can lead to more, rather than less, accurate computation. We suggest that noisy network attractors will be useful for understanding neural networks that reliably and sensitively perform finite-state computations in a noisy environment.**

I. Introduction

ALTHOUGH there is a very good understanding of finite-state computational machines in both abstract and practical settings, and a good understanding of the dynamical properties of many aspects of the central nervous system of animals, there remains a debate on how an evolved biological system can apparently undertake finite-state computations in a routine and robust (albeit, error-prone) manner whilst remaining sensitive to inputs that may be of very low amplitude. In other words, it is still not really clear how a virtual (finite state) machine may be embedded within the imperfect and noisy behaviour of a distributed neuronal network [35]. Dynamical systems theory has contributed a lot to understanding how computational systems may operate for fixed inputs, but true computational systems act on time-varying inputs. The theory of non-autonomous dynamical system (e.g. using feedback control approaches for computation [24]) is much less developed than that for autonomous (input-free) systems. Nonetheless there are models of computational systems that are sensitive to arbitrary low-amplitude inputs. These include models with heteroclinic connections between: equilibria [29], [30], periodic orbits [2], [6], [27] or chaotic saddles/Milnor attractors [37]. Our model develops these ideas to construct explicit realizations of Turing Machines (TMs) using a system first described in [4].

Bournez and Campagnolo [9] review some attempts to use continuous time analogue (continuous state) dynamical systems to perform finite state computations of the type envisaged in [38]. Key problems are how to explain the functioning of apparent finite state computational processes in a robust way with continuous state variables, and to understand their complexity [33]. If computation can be achieved using only low-amplitude input signals and allow for noise and imperfections in the system, so much the better. Further work has attempted to implement more general types of computation (for example super-Turing [32]) but we do not consider this here.

Analogue models of computation using continuous time have been extensively considered since [26]. There is an extensive literature on Recurrent Neural Networks (RNNs) where simple nonlinear elements interact in ways that allows computation to emerge from the dynamics of the system [31]. Important examples include Echo State Networks (ESNs) [8], [21], [22], [20], [24], long short-term memory networks (LTSNs) [15], coupled spiking neurons [7], [17] and Universal Memcomputing Machines (UMMs) [10], [36]. Various researchers have highlighted critical behaviour, richness of dynamics and computational properties for coupled systems with even relatively simple dynamics. Various issues have been addressed, such as the realisation of TMs with explicit polynomial vector fields including bounded noise [14] and the modelling of the behaviour of noisy TMs [1] where at each stage the tape symbols can change with some small probability. Realistic models of neural computation should clearly include the possibility of imperfect computation [23]. Although the paradigm of TMs with a halting state is not ideal, it remains the benchmark for finite-state computational systems.

Non-deterministic analogues of TMs have been suggested to solve NP-complete problems in polynomial time [36]. Coupled phase oscillator models [7], [17] have been proposed for Turing-type machines [2] as have pulse-coupled phase oscillators [27] but in senses where the interfaces to the outside world may not be so simple.

In this paper, we explain how to embed a general finite-state machine (modelled as a TM) into the dynamics of a distributed network of coupled nonlinear dynamical systems. We use a method presented in [4] that couples two layers of systems, each system having one dynamic degree of freedom:

- There is a "classifying" network of mutually inhibiting systems (p-systems) with multiple attractors that are used to classify the state of the TM.

Center for Systems, Dynamics and Control/ Department of Mathematics, University of Exeter, Exeter EX4 4QF, UK and EPSRC Centre for Predictive Modelling in Healthcare, University of Exeter, Exeter, EX4 4QJ, UK. Email: P.Ashwin@exeter.ac.uk

Department of Mathematics, University of Auckland, Auckland, 1142, New Zealand. Email: c.postlethwaite@auckland.ac.nz

- There is a "connecting" network of systems (y-systems) that selectively excite certain of the classifying states and inhibit others. The connecting systems inhibit each other to ensure that a well-defined transition is achieved.

As discussed in [4], one can embed a dynamic network in the phase space of such systems using an explicit construction, and realise any given finite graph as a "network attractor" between states in phase space. These states are equilibria of the system and the network may be *excitable* or *heteroclinic*. In [5] we examine the robustness of such an autonomous system to noise whilst in this paper we turn our attention to what happens if in addition to noise, we permit deterministic inputs to the system that perturb particular nodes in the "connecting" network.

On changing a single global parameter ν that affects the excitability of the y-systems, the attractor for the dynamics can be varied between the following two regimes:

- In the "free-running" or heteroclinic regime ($\nu < 0$), there is a spontaneous wandering between states, each of which is of saddle type (the network is a *robust heteroclinic network* attractor in phase space). The states visited depend critically on the noise and other inputs to the system.
- In the "excitable" regime ($\nu > 0$), the system contains stable states, such that perturbations over some threshold can cause a transition to a new state on the network (the network is an *excitable network* in phase space). For this excitable network, perturbations must exceed a minimum threshold to wander between states, but this threshold can be arbitrarily small by choosing ν close to 0.

The "excitability parameter" ν can be thought of as providing a measure of criticality of the network dynamics: for $\nu < 0$ computation can be undertaken with arbitrarily small inputs while for $\nu > 0$ there is an input threshold under which computation will not progress. In the presence of both inputs (signals) and noise (randomness), in both free-running and excitable regimes there will be a competition between these effects, and one aim of this paper is to explore this competition.

This excitability parameter has an analogous role to neuromodulators within the brain, in that ν globally affects speed, sensitivity to inputs and error rate of the computation, but not the computation itself. The systems we discuss have many features of, and could possibly be used to realise, UMMs [36]: storage and processing are distributed to all locations rather than being divided between memory and central processing unit, though the detailed dynamics and structure we investigate is quite explicit.

The paper is organized as follows: in Section II we describe heteroclinic and excitable networks in phase space and introduce a particular family of systems from [4]. Some details in Appendix A outline how one can embed an arbitrary directed graph (with the restriction that it contains no one-cycles/self-loops) as such a network in a robust manner. Section III applies this construction to embed the behaviour of a generic TM in the dynamics of the system, where there is a "virtual paper tape" with an asynchronous symbol-dependent feedback to the dynamics. In Section IV we consider the three state "busy beaver" [19] as a case study of a particular TM realised using a network attractor in this way. We confirm that on the addition of a simple interface to the virtual paper tape the resulting differential equations can be used to faithfully reproduce the TM behaviour.

Section V uses an idealisation of a TM with recurrent dynamics (i.e. no halting state) to explore the performance of the system in terms of the error rates as a function of noise, signal and excitability. We classify transition errors into "read errors" and "wild errors". We find an intriguing and paradoxical result, that increasing noise can (in certain circumstances) reduce the error rate in the system. Finally, in Section VI we discuss how this work can be usefully extended to more realistic neural models, to neurally inspired computational architectures and to adaptive learning of distributed systems.

II. Deterministic and noisy networks in phase space

We briefly recall some of the concepts and notation used to describe network attractors in our previous work [4]. We consider the *non-autonomous stochastic differential equation* (NSDE)

$$dx = [f(x,\nu) + \zeta z(x,t)]\,dt + \eta\,dw \tag{1}$$

and the associated *autonomous ordinary differential equation* (AODE)

$$\frac{d}{dt}x = f(x,\nu), \tag{2}$$

where the terms are described below. In both (1) and (2), $x \in \mathbb{R}^d$ where $t \geq 0$, $f(x,\nu)$ is a smooth nonlinear function, and $\nu \in \mathbb{R}$ is a bifurcation parameter. The AODE is derived from the NSDE by removing both the noise term and the inputs ($\eta = \zeta = 0$). Specifically, components of the vector $w(t)$ are standard independent identically distributed (i.i.d.) Wiener processes, and $\eta = \mathrm{diag}(\eta_1, \ldots, \eta_d)$ gives the noise amplitudes. The vector $z(x,t)$ is an additional non-autonomous (control) input that affects the state according to a control amplitude $\zeta > 0$ that we view as a parameter for the system. The control $z(x,t)$ models inputs from the environment: in general the environment may be affected by outputs from the system and so a computational system will include two-way interaction with the environment. We consider an asynchronous feedback case where $z(x,t)$ is piecewise constant. We assume for simplicity that the noise in the various components of (1) is additive and uncorrelated; there is no difference between Itô and Stratonovich formalism in this case.

We consider the computational capabilities of (1) as arising from the interaction of the "pure nonlinear dynamics" of (2) with the noise $\eta\,dw$ and the control input $\zeta z(x,t)$.

Let $\Phi_t(x_0)$ denote the flow generated by (2), i.e. the solution $x(t)$ to the initial value problem starting at $x(0) = x_0$. We assume there is a set of $N \in \mathbb{N}$ equilibria ξ_j, i.e. points in $\mathbb{R}^d$ such that $f(\xi_j,\nu) = 0$; the existence and location of these may vary with ν but in the following we will assume that ν only changes the stability of the equilibria. How long the system remains near an equilibrium depends on the stability of the equilibrium, as well as any noise or nonautonomous inputs (when considering (1)). Generically, all equilibria ξ_j will have stability that is determined by the linearisation of (2) about ξ_j,

as long as the linearised system $J_j = df(\xi_j, \nu)$ is *hyperbolic* (i.e. no eigenvalue λ of J_j has $Re(\lambda) = 0$). We denote by n_s the number of stable eigenvalues, i.e. those with $Re(\lambda) < 0$ and by n_u the number of unstable eigenvalues, i.e. those with $Re(\lambda) > 0$.

If the equilibria are used to describe computational states then a computational process that changes these states needs to move the system dynamically between respective equilibria. On the one hand, a large instantaneous perturbation of the system could move the state from one stable equilibrium (i.e. one with $n_s = d$) to another. On the other hand, if the equilibrium has an instability (i.e. it has $n_u > 0$), then the nonlinear dynamics may take the system from one equilibrium to another if there is a connecting orbit between them.

More precisely, each hyperbolic equilibrium ξ_j with n_s stable eigenvalues and n_u unstable eigenvalues will have invariant sets

$$W^s(\xi_j) = \{x \in \mathbb{R}^d \ : \ |\Phi_t(x) - \xi_j| \to 0 \ \text{ as } t \to \infty\}$$
$$W^u(\xi_j) = \{x \in \mathbb{R}^d \ : \ |\Phi_t(x) - \xi_j| \to 0 \ \text{ as } t \to -\infty\}$$

where the *stable manifold* W^s is an embedded n_s-dimensional invariant manifold and the *unstable manifold* W^u is an embedded $n_u = (d - n_s)$-dimensional manifold, and the rate of limiting is exponential and determined by the eigenvalues of J_j: see for example [18, Theorem 2.1] for an exposition of this.

Although the invariant sets $W^{s,u}(\xi_j)$ are embedded manifolds that contain ξ_j, they may contain other invariant sets in their closures. Connecting heteroclinic orbits are trajectories that are contained in the unstable manifold of one equilibrium and the stable manifold of another. Heteroclinic networks composed of a network of such trajectories have been proposed by a number of authors as a way that neural systems can encode information and perform computations [2], [6], [27], [29], [30], [34].

For a heteroclinic network, each equilibrium that is part of a nontrivial cycle must have $0 < n_s$ and $0 < n_u$ and hence be of saddle type. More recently, it was noted in [4] that a bifurcation of such saddles within a network can lead to an excitable network where each equilibrium is stable ($n_u = 0$). In such a network there will be thresholds of excitability, where a small perturbation can lead to switching that is effected by reaching the stable manifold of the new equilibrium.

A. Heteroclinic and excitable networks

We now give definitions of heteroclinic and excitable networks [4] that are used in this paper. There is a *heteroclinic connection* from one equilibrium ξ_i to another ξ_j for (2) if

$$W^u(\xi_i) \cap W^s(\xi_j) \neq \emptyset$$

and (2) has a *heteroclinic network attractor* if there is an asymptotically stable compact connected set $\Sigma_H \subset \mathbb{R}^d$ such that for some set of saddle equilibria $\{\xi_i\}_{i=1}^N$ we have

$$\Sigma_H = \bigcup_{i=1}^{N} W^u(\xi_i) \tag{3}$$

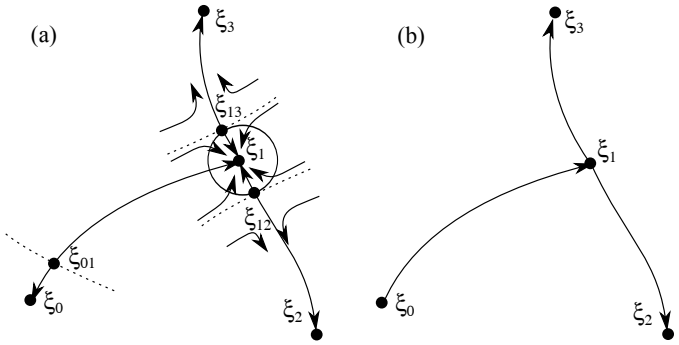

Fig. 1. Schematic diagram showing connections between equilibria ξ_i representing phase space states of the system, when switching from ξ_0 to ξ_1 and then to one of ξ_2 or ξ_3. The cases of (a) excitable ($\nu > 0$) and (b) heteroclinic ($\nu < 0$) connections for the AODE (6) are shown. The arrows indicate trajectories for the noise-free autonomous system. In case (a) observe that near ξ_1 there is a threshold (black circle) that must be exceeded by any impulsive perturbation to achieve switching to a new state. The basin boundary of ξ_1 is given by the stable manifolds of the threshold saddles ξ_{1i}. In case (b) there is an instability of ξ_1, meaning there is no minimum threshold of perturbation to achieve switching between states. This case naturally leads to faster but less controllable switching.

We say the system (2) has an *excitable connection for amplitude* $\delta > 0$ from one equilibrium ξ_i to another ξ_j if

$$B_\delta(\xi_i) \cap W^s(\xi_j) \neq \emptyset$$

(where $B_\delta(\xi)$ is the ball of radius δ centered at ξ), and this connection has *threshold* δ_{th} if

$$\delta_{th} = \inf\{\delta > 0 \ : \ B_\delta(\xi_i) \cap W^s(\xi_j) \neq \emptyset\}.$$

There is an *excitable network of amplitude* $\delta > 0$ if there is a set of equilibria $\{\xi_i\}$ such that the set

$$\Sigma_E = \bigcup_{i,j=1}^{n} \{\Phi_t(x) \ : \ x \in B_\delta(\xi_i) \text{ and } t > 0\} \cap W^s(\xi_j) \tag{4}$$

is attracting.

For an excitable network of amplitude δ we can follow an arbitrary path on the network by a mixture of trajectories and "jumps" of δ, while for a heteroclinic network the δ can be chosen arbitrarily small. In [4] we gave a particular construction of coupled nonlinear systems in the form of equations (2) where an arbitrary network can be constructed as a heteroclinic or as an excitable network in phase space. We included more details in Appendix A. The response of this network to noisy perturbations is considered in [5] and Figure 1 shows the situation in phase space for two values of an excitability parameter ν that bifurcates between these cases at $\nu = 0$.

B. Relating trajectory itineraries for the AODE and the NSDE

If we assume that typical trajectories of (1) spend most of their time close to a heteroclinic or excitable network Σ_N of (2), one can describe the motion in terms of the itinerary around the network, i.e. the sequence and timing of visits to

the equilibria ξ_k - this is detailed in [5] but we summarise it below. For fixed $0 < h < \frac{1}{2}\min_{i,j}|\xi_i - \xi_j|$ we define

$$K(x) := \begin{cases} k & \text{if there exists a } k \text{ such that } |x - \xi_k| \leq h \\ 0 & \text{otherwise.} \end{cases}$$

so if x is close to an equilibrium, then $K(x)$ gives the index of that equilibrium. Note that the choice of maximum of h means that $K(x)$ is uniquely defined. For a trajectory $x(t)$ of (1) we define

$$\tilde{K}(t) = \{K(x(\tilde{t})) \ : \ \tilde{t} = \sup\{\tilde{t} \leq t \ : \ K(x(\tilde{t})) \neq 0\}\} \quad (5)$$

which gives the "last visited equilibrium". If a trajectory starts near an equilibrium this will always be non-zero, though note it depends on the threshold h. For $K(x(t)) = k$ we say $x(t)$ is *close to the kth equilibrium*. As conjectured in [5], under suitable assumptions which include $|\eta|$ small and no input ($\zeta \equiv 0$), trajectories will remain close to one of the ξ_j for most of the time, and transitions will correspond to following heteroclinic or excitable connections within the network.

For a given $x(0)$ and realisation of the noise and control, the trajectory $x(t)$ can be characterised as an itinerary of *epochs*

$$\{(k(n), \tau(n)) \ : \ n \in \mathbb{N}\}$$

where $\tilde{K}(t) = k(n)$ for the interval $t \in [\tau(n), \tau(n+1))$, and $k(n+1) \neq k(n)$. We define the *duration* of the nth epoch

$$\tilde{\tau}(n) = \tau(n+1) - \tau(n).$$

In [5] we discuss several statistical properties of the epoch durations (also known as *residence times*) and of the transition probabilities between equilibria for (1) in the absence of a control input. That paper shows there are heteroclinic and excitable networks of arbitrary complexity where the dynamics for low noise is well-modelled by a one-step Markov process.

C. Dynamic realizations of arbitrary network attractors

In order to undertake arbitrary computational tasks one can clearly use gradient systems with a high degree of multistability and then consider perturbations that take the state from the basin of attraction of one attractor to the basin of another. In simple terms, this is how electronic computers currently work: a large number N of bits of information are stored in bistable circuit elements that are as isolated from each other as possible. This gives rise to a multistable system with 2^N states. The associated circuits are about changing the states by appropriate control impulses. However this gives rise to circuit structures that are not very distributed - the central processing unit has a key role, and timing needs to be imposed externally rather than emerging from the circuit dynamics *per se*.

On the other hand, if we consider the computational task as sitting on a given directed graph of equilibria for a highly connected dynamic network (and inputs to the system determine the transitions between states) then one can embed a given finite direct graph in the dynamics of (2) and use the control to induce transitions between states.

Several recent studies [3], [4], [5], [11] address the question of how to construct a system whose dynamics gives an arbitrary network structure in phase space. The method we discuss here is from [4] which considers an arbitrary (one-cycle free) directed graph G with n_v nodes (vertices) and n_e edges. The construction embeds the graph into the dynamics of an explicitly given AODE on $(p, y) = (p_1, \ldots, p_{n_v}, y_1, \ldots, y_{n_e}) \in \mathbb{R}^{n_v+n_e}$ of the form

$$\begin{aligned} \frac{d}{dt}p_i &= f_i(p, y) \\ \frac{d}{dt}y_j &= f_{n_v+j}(p, y, \nu) \end{aligned} \quad (6)$$

for $i = 1, \ldots, n_v$ and $j = 1, \ldots, n_e$, where $f = (f_1, \ldots, f_{n_v+n_e})$ is described in equation (13) in Appendix A. The p variables classify which of the nodes of the graph is visited; in this system $p_i \approx 1$ (and all other $p_k \approx 0$) when the system is close to node i. The y variables are activated (i.e. become non-zero) during a transition between nodes: more details of this process is given in [4] and outlined in Appendix A. The parameter ν (which only affects the y-dynamics) is such that for $\nu < 0$ the embedding is as a heteroclinic network while for $\nu > 0$ it is as an excitable network where all edges have threshold approximately

$$\delta_{th} \approx \sqrt{\nu/2} \quad (7)$$

so $\delta_{th} \to 0$ as $\nu \to 0+$: see [4] for more details.

III. Network attractors and Turing machines

We start by recalling the classical TM, and then discuss our method of encoding these machines within the dynamics of noisy network attractors.

A. Turing machines

A finite state single-tape Turing Machine can be thought of as the seven-tuple [16]

$$(Q, \Gamma, b, \Sigma, \rho, q_1, F)$$

where a machine has:

- a finite set of *internal states* Q,
- a *starting state* $q_1 \in Q$,
- at least one, and possibly several, *halting states* $F \subset Q \setminus \{q_1\}$,
- a finite set Γ of *tape symbols*, one of which is a *blank* b and a set $\Sigma \subset \Gamma \setminus \{b\}$ of *input symbols*,
- a *transition function* $\rho : (Q \setminus F) \times \Gamma \to Q \times \Gamma \times \{L, R\}$. (N.B. we follow the notation of [16] but note that one can extend the definition of ρ to Q in a trivial way.)

Let $n_Q := |Q|$, the number of internal states, and $n_\Gamma := |\Gamma|$, the number of possible symbols. We number the states and symbols $q_i \in Q$, $i = 1, \ldots, n_Q$ and $s_j \in \Gamma$, $j = 1, \ldots, n_\Gamma$, so we can represent the action of the transition function ρ as

$$(\tilde{q}_{ij}, \tilde{s}_{ij}, \tilde{\sigma}_{ij}) := \rho(q_i, s_j). \quad (8)$$

where $\tilde{q}_{ij} \in Q$, $\tilde{s}_{ij} \in \Gamma$ and $\tilde{\sigma}_{ij} \in \{L, R\}$.

The *tape* consists of a doubly infinite string of symbols $\gamma_j \in \Gamma$, $j \in \mathbb{Z}$, where a finite number of the γ_j are in Σ and the infinite remainder are all blank b. The symbols on the tape are read and written by a moveable *head*. Starting at head position $j = 0$ and internal state $q(0) = q_1$ suppose that the

machine arrives after n steps at internal state $q(n) \in Q$ and head position $\beta(n)$. If $q(n) \in F$ then the computation has finished, while otherwise it performs the transition given by

$$(\tilde{q}, \tilde{\gamma}, \tilde{\sigma}) = \rho(q(n), \gamma_{\beta(n)})$$

where $q \in Q$, $\tilde{\gamma} \in \Gamma$, $\tilde{\sigma} \in \{L, R\}$. The machine then updates the internal state to $q(n+1) = \tilde{q}$, the head changes the symbol at site $\beta(n)$ to $\tilde{\gamma}$ and then the head moves along the tape according to $\beta(n+1) = \beta(n) + 1$ if $\tilde{\sigma} = R$ and $\beta(n+1) = \beta(n) - 1$ if $\tilde{\sigma} = L$. The machine repeats this either forever, or until it reaches a halting state and computation has finished.

We claim that the dynamics of (1) with f being defined by (6) and an appropriate interface to a tape with moving and printing capabilities is capable of efficiently encoding an arbitrary TM. We do this by considering the graph G used in the construction of equations (6) to be the graph of internal states of the TM with nodes Q and edges $q_i \to q_j$ if there is a symbol $s \in \Gamma$ such that $\rho(q_i, s) = (q_j, \tilde{s}, \tilde{\sigma})$ for any $\tilde{s}$, $\tilde{\sigma}$.

B. Asynchronous Turing machine realization using a network attractor

We construct a realisation of an arbitrary TM using a nonautonomous stochastic differential equation. Specifically, we consider equation (6) (with f described in (13) in Appendix A), with added noise and inputs, to give the system

$$\begin{aligned} dp_i &= [f_i(p, y, \nu) + \zeta z_i(t)]\, dt + \eta_p dw_{p,i} \\ dy_j &= [f_{n_v+j}(p, y, \nu) + \zeta z_{n_v+j}(t)]\, dt + \eta_y dw_{y,j} \end{aligned} \tag{9}$$

for $i = 1, \ldots, n_v$ and $j = 1, \ldots, n_e$. The components of vectors w_p and w_y are standard i.i.d. Weiner processes and the noise amplitudes η_p and η_y are allowed to vary independently: they correspond to noise amplitudes in the p and y systems respectively.

The z_i are the components of the input vector $z(t)$, which is determined by the symbol on the tape at the current head position. In the following, we define a function $\hat{\beta}(t)$ which gives the position of the head at time t, and functions $\hat{\gamma}_j(t)$ which give the symbols at position j on the tape at time t. Formally we have $z(t) = Z(\hat{\gamma}_{\hat{\beta}(t)}(t))$ for some function Z which acts on tape symbols. The functions $\hat{\beta}(t)$ and $\hat{\gamma}_j(t)$ are piecewise constant, and may have discontinuities at the start of each epoch (i.e. when $t = \tau(n)$ for some $n \in \mathbb{N}$).

This realisation has asynchronous feedback, i.e. the timing of the transitions is purely determined by the dynamics of the machine rather than externally imposed.

As mentioned in the introduction, our network realisation only works if the graph of states does not contain "self-loops". This is because self-loops would correspond to homoclinic orbits in the dynamical system, which are of codimension greater than zero (i.e. they can only exist for isolated parameter values). We consider only TMs that contain self-loops: this is not a real limitation as in Appendix B we explain how to remove any self-loops from a TM. In order to achieve this we add the action N that writes nothing on the tape, and the action 0 which does not move the head.

Our implementation of the TM described in section III-A as the dynamical system (9) has a phase space $(p, y) \in \mathbb{R}^{n_v+n_e}$

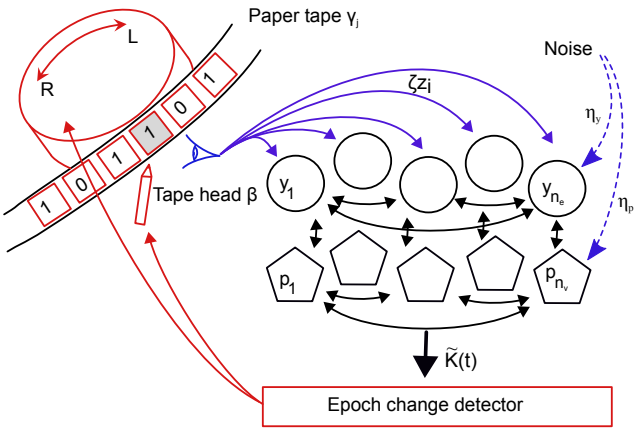

Fig. 2. Using the system (9) with an external paper tape to approximately realise an arbitrary TM. The inputs come from the tape head and noise sources with amplitudes ζ and $\eta_{y,p}$ respectively. The "connecting" layer of y cells has dynamics that can induce changes in the "classifying" layer of p cells. The output to the tape head is determined by a detected change in the state $\tilde{K}(t)$ of the p variables. The excitability parameter ν only affects the dynamics of the y variables.

where $n_v = n_Q$ (the number of states in the TM) and $n_e = n_Q n_\Gamma$ (i.e. one edge for each symbol choice at each state). We associate each equilibrium in the phase space ξ_i ($i = 1, \ldots, N_Q$) with a state $q_i \in Q$ of the TM. The transitions between states are described using a list of directed edges e_j for $j = 1, \ldots, n_e$. Each edge e_j starts at a equilibrium $\alpha(e_j)$, ends at equilibrium $\omega(e_j)$ and has associated with it a direction $\sigma(e_j) \in \{L, R\}$ (corresponding to the action on the tape). So, for $i = 1, \ldots, N_Q$ and $j = 1, \ldots, n_\Gamma$:

$$\begin{aligned} \alpha(e_{(i-1)n_\Gamma+j}) &= q_i \\ \omega(e_{(i-1)n_\Gamma+j}) &= \tilde{q}_{ij} \\ \sigma(e_{(i-1)n_\Gamma+j}) &= \tilde{\sigma}_{ij} \end{aligned}$$

where $\tilde{q}_{ij}$ and $\tilde{\sigma}_{ij}$ are given by the transition function ρ (8). The embedding of the graph G with nodes ξ_i and edges e_j as described, using equation (9) and the construction described in Appendix A, gives a network attractor that mirrors the graph of states of the TM. Figure 2 schematically shows this method of realization.

We use the process described in section II-B and the last visited node variable $\tilde{K}(t)$ (equation (5)) to classify the dynamics of $(y(t), p(t))$ into *epochs* $\{(k(n), \tau(n))\}$. The tape is implemented as a dynamical process external to the dynamical system (9), as follows. The position of the head $\beta(n)$ is described in the continuous-time setting as a step-function $\hat{\beta}(t)$, defined as

$$\hat{\beta}(t) = \beta(n), \quad \text{when } t \in [\tau(n), \tau(n+1)).$$

Note that this depends on the threshold $h > 0$ used for classification of the state.

Similarly, the symbols on the tape γ_j can each be considered in the continuous-time setting as step functions $\hat{\gamma}_j(t)$. These potentially change at the start of each epoch, i.e. whenever $t = \tau(n)$. The symbols $\hat{\gamma}_j(t)$ on the tape and the head's position $\hat{\beta}(t)$ are updated by the dynamics, and in turn affect

the dynamics. The dynamical system (9) is initialised at time $t = 0$ close to the starting state ξ_1 corresponding to q_1 and head's position $\hat{\beta}(0) = 0$.

C. The action of the dynamics on the tape

Each time the system starts a new epoch (i.e. at time $t = \tau(n)$, for some $n \in \mathbb{N}$), the trajectory in the phase space (p, y) becomes close to a new equilibrium, $\xi_{k(n)}$. That is, there is a transition on the TM from state $q_{k(n-1)}$ to state $q_{k(n)}$. For $\tilde{q}_{ij}$ again defined by ρ, there are three possibilities:

- If there is a j such that $s_j = \gamma_{\beta(n-1)}$, and

$$\tilde{q}_{k(n-1)j} = q_{k(n)}$$

  then we say there is a *read accurate transition*, or no error.
- If there is a j such that $s_j \neq \gamma_{\beta(n-1)}$ and

$$\tilde{q}_{k(n-1)j} = q_{k(n)}$$

  then we say there is a *read inaccurate transition* or a *read error*.
- If there is no j such that

$$\tilde{q}_{k(n-1)j} = q_{k(n)}$$

  then we say there is a *wild transition* or a *wild error*.

If there is a read accurate transition or a read error, and $\tilde{\sigma}_{k(n-1)j} \neq 0$, then we change the symbol on the tape at location $\beta(n-1)$ to $\tilde{s}_{k(n-1)j}$ and update the new location according to $\beta(n) = \beta(n-1) + 1$ if $\tilde{\sigma}_{k(n-1)j} = R$ and $\beta(n) = \beta(n-1) - 1$ if $\tilde{\sigma}_{k(n-1)j} = L$. If $\tilde{s}_{k(n-1)j} = N$ then we do not change the symbol on the tape, and if $\tilde{\sigma}_{k(n-1)j} = 0$ we do not move the tape. This corresponds to the "do nothing" possibility at a transition because we have put in additional transitions for each self-loop. If there is a wild error we similarly neither update the symbol on the tape nor move the tape. For a sequence of transitions of the system we define rates of read or wild errors as the proportion of transitions for which a read or wild error occurs.

D. The action of the tape on the dynamics

The current state of the tape, $\hat{\gamma}_{\hat{\beta}(t)}(t) \in \Gamma$ is used to determine the direction of the control perturbation $z(t) \in \mathbb{R}^{n_v+n_e}$. Suppose that $\hat{\gamma}_{\hat{\beta}(t)}(t) = s_j$ for some $j \in \{1, \ldots, n_\Gamma\}$, then we define a piecewise constant control $z(t) = Z(\hat{\gamma}_{\hat{\beta}(t)}(t))$ where

$$Z_l(s_j) = \begin{cases} 1 & \text{if } l = n_v + in_\Gamma + j \\ & \text{for some } i \in 0, \ldots, n_v - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

That is, the components of $z(t)$ corresponding to activated edges for node $\hat{\gamma}_{\hat{\beta}(t)}(t)$ are set equal to 1, and all others are zero.

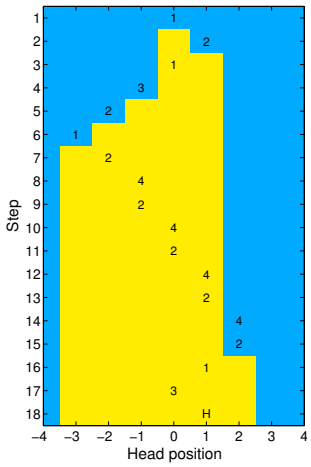

Fig. 3. The figure shows output from a simulation of the "busy beaver" routine. Each row shows the symbols of the tape at each step in the sequence. Blue colour indicates a 0 and yellow indicates a 1. The position of the numbers in the figure shows the position of the head at each step, and the value of the number indicates the current internal state of the TM.

IV. Network dynamic realizations of Turing machines

As a specific illustration, we consider the three-state "busy beaver" [19]. In this example, there are only two possible symbols on the tape, $s_1 = 0$, which is also the blank symbol b, and $s_2 = 1$. We show an example simulation in Figure 3: the symbols on the tape and the read/write head's position are illustrated at each step. More details of the parameters and numerical methods used in the computations are given in Appendix A.

The three-state two-symbol "busy beaver" TM of [19] is so-called because it takes the maximum number of steps of any three-state two-symbol TMs to arrive at a halting state. For larger numbers of states (and still only two symbols) the general problem of how long a busy beaver can take is an outstanding open problem in the theory of computation: some recent records are given in [25].

There are three states not including the halting state, so in terms of the formalism above we write

$$Q = \{q_1, q_2, q_3, H\},\ \Gamma = \{0, 1\},\ b = 0,\ \Sigma = \{1\},\ F = \{H\}$$

and the transition function can be tabulated as shown on the left of Table I. We show the graph of connections between states in Figure 4. Note the one-cycle in the associated graph corresponding to remaining in state q_2 if symbol 1 is read on the tape. We add an additional state q_4 and the possibility that there is no move N - this allows one to illustrate the machine transition function as a one-cycle free directed graph with transition function shown in the table on the right of Table I, and by the graph in Figure 4.

The output from an integration of the equations (9) is shown in Figure 3. A portion of the corresponding time series is shown in Figure 5.

Inputs		Outputs		
q_i	s_j	$\tilde{q}_{ij}$	$\tilde{s}_{ij}$	σ_{ij}
q_1	0	q_2	1	R
q_1	1	q_3	1	L
q_2	0	q_1	1	L
q_2	1	q_2	1	R
q_3	0	q_2	1	L
q_3	1	H	1	R

Inputs		Outputs		
q_i	s_j	$\tilde{q}_{ij}$	$\tilde{s}_{ij}$	σ_{ij}
q_1	0	q_2	1	R
q_1	1	q_3	1	L
q_2	0	q_1	1	L
q_2	1	q_4	1	R
q_3	0	q_2	1	L
q_3	1	H	1	R
q_4	0	q_2	N	0
q_4	1	q_2	N	0

TABLE I

LEFT: TRANSITION TABLE ρ FOR THE 3-STATE 2-SYMBOL "BUSY BEAVER". RIGHT: TRANSITION TABLE $\rho^\star$ FOR A ONE-CYCLE FREE VERSION OF THE SAME.

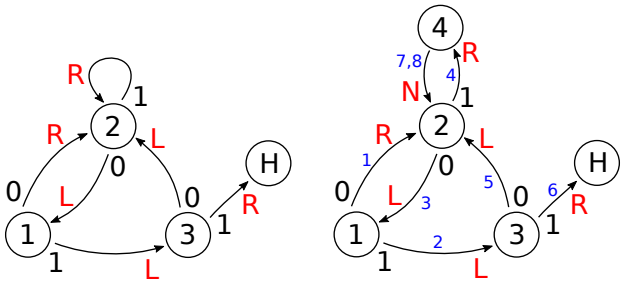

Fig. 4. The left figure shows the original 3-state "busy beaver" transition table as a directed graph, and the right shows the corresponding one-cycle-free directed graph on adding the "waiting state" 4 and the "do nothing" action N. The black numbers on the edges correspond to the tape symbols, and the red letters L and R to the direction the tape is moved during each transition. The small blue numbers on the right figure correspond to the labels of the edges; compare with the order of the lines in table I.

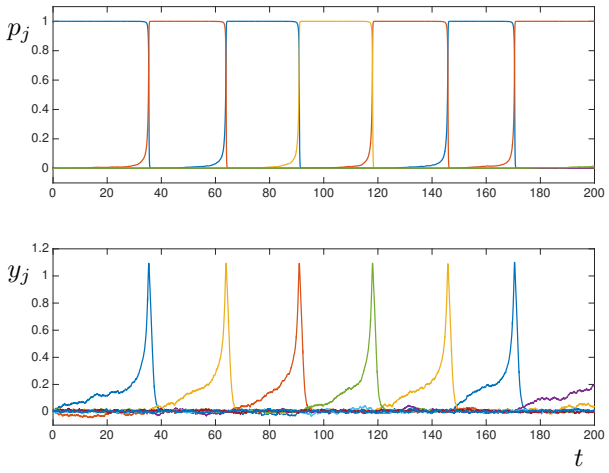

Fig. 5. The figures show a portion of the time series for a realisation of a reliable computation for the 3-state busy beaver using a noisy heteroclinic network. The top figure shows the p-variables, (p_1 blue, p_2 orange, p_3 yellow, p_4 green) and the bottom figure shows the y variables (y_1 blue, y_2 orange, y_3 yellow, y_4 purple, y_5 green). Compare the top figure with the sequence of states shown in Figure 3 (left). Parameters are $\nu = 0.05$ (so the network is excitable rather than heteroclinic), $\eta_y = 0.01$, $\eta_p = 0.0001$, $\zeta = 0.01$ and all others as the standard set given in Appendix A.

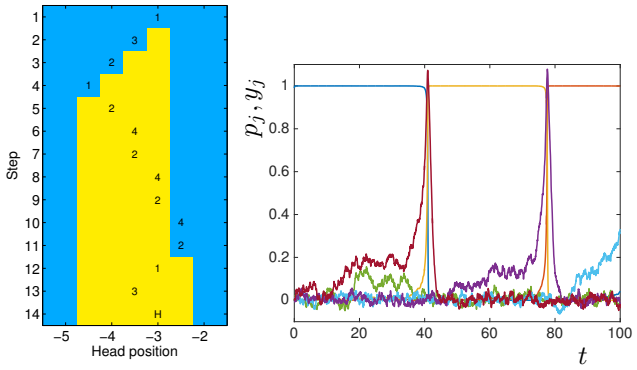

Fig. 6. The figures show a realisation of the busy beaver routine which makes an error in its very first step. Compare the left hand figure here with the correct output on the left of Figure 3. The time series on the right shows that between approximately $t = 20$ and $t = 30$, the p_1 variable (blue) is turned on and both the y_1 (green) and y_2 (red) connections begin to become activated. The input is in the y_1 dimension but the particular realisation of the noise in the y_2 dimension is in this instance sufficient that the transition in this dimension occurs first and the system moves to the p_2 state. Colours are: p_1 blue, p_2 yellow, p_3 orange, y_1 green, y_2 red. Parameters are $\nu = 0.05$ (so the network is excitable rather than heteroclinic), $\eta_y = 0.03$, $\eta_p = 0.0001$, $\zeta = 0.01$ and all others as the standard set given in Appendix A.

V. COMPUTATIONAL PERFORMANCE OF NOISY NETWORK ATTRACTORS

As noise amplitude η is increased relative to the system dynamics and the input amplitude ζ, errors will be made in the computation. These may be read errors or wild errors as defined in section III-C. In Figure 6 we show an example of a time-series and resulting tape output for a computation of the "busy beaver" which makes a read error in its first step.

The observed rate of errors in a computation is due to a balance between the noise and the input amplitudes (i.e. the size of the perturbations coming from the tape). In Figure 7 we give a schematic to show how we expect the error rate and speed of computation to vary as the noise and input amplitudes are varied both for the case of heteroclinic ($\nu < 0$) and excitable ($\nu > 0$) transitions. The influence of noise on mean residence times for heteroclinic and excitable network attractors is considered in [5]. In the excitable case $\nu > 0$ and for no input, the mean residence times are expected to scale according to a Kramers' law $\exp(K\nu^2/\eta_y^2)$ with K constant, so for values of η_y appreciably less than ν we expect to find very large residence times. Conversely, if there is no noise then it can be shown that if $\zeta < \zeta_c = \frac{\sqrt{2}}{3\sqrt{3}}\nu^{3/2}$ (see Appendix C for details) then all the equilibria corresponding to the states in the TM are stable and hence there will be no transitions at all. As either the noise or input amplitudes increase above these minima, we expect the residence times to decrease. In both the heteroclinic and excitable cases, if $\zeta \gg \eta_y$ or $\zeta \ll \eta_y$ then we expect the motion around the network to be dominated by either inputs or noise, and the error rates will tend towards 0 or 0.5 (purely random motion) respectively. In fact, we note from computations that if $\zeta > \eta_y$ (the input is greater than the noise) then the error rate is extremely small.

We simulated the computational properties of the network attractors in order to understand the speed of transitions and error rates as a function of input and noise strengths. We give

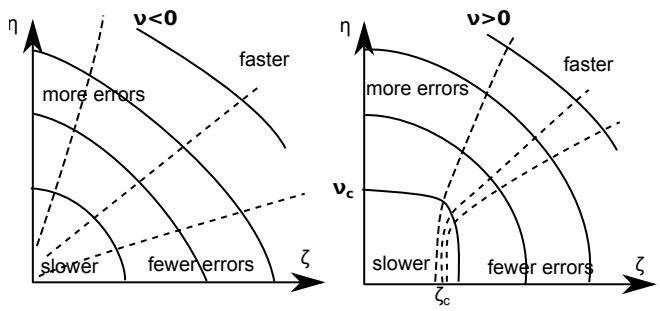

Fig. 7. Schematic showing how we expect speed of computation and error rate to vary, for the asynchronous feedback case as functions of the input strength ζ and a noise amplitude η, showing the heteroclinic case $\nu < 0$ and the excitable case $\nu > 0$. Note that accurate transitions are possible down to arbitrarily small signals in the heteroclinic case, at the expense of speed of transition. For the excitable case there is a threshold of signal that must be exceeded for an accurate transition to take place.

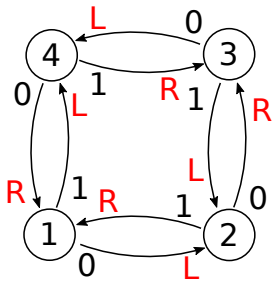

Fig. 8. The figure shows a "recurrent" machine that resembles a TM but (a) we assume that it writes the symbol '0' at every point in time and (b) there is no halting state, and so it will continue to make transitions *ad infinitum*.

results only for excitable networks but found that heteroclinic networks perform similarly, with the main difference being that excitable networks need a minimum input amplitude (relative to the excitability parameter ν) to perform computations successfully. To avoid being in the halting state, we consider a simple "recurrent" machine with no halting state and transitions shown in Figure 8. We start in state 1 and then progress at each point in an anticlockwise or clockwise fashion depending on whether the symbol read is '0' or '1'. At each state we write only '0' and we start with a tape containing only '0'. This means that at each stage we may do:

- an anticlockwise transition corresponding to a "no error"/"read accurate transition",
- a clockwise transition corresponding to a "read error" or
- a diagonal transition corresponding to a "wild error".

Figures 9 shows the results of many computations using this "recurrent" machine. Panel (a) shows the proportion of transitions that make read errors as the noise parameter η_y and the input amplitude ζ are varied, and panel (c) shows the mean residence times for the same computations. In this case, η_p is much smaller than η_y, and so read errors are more common that wild errors, because excitable connections between the nodes visited when a read error is made exist, while those between nodes visited during wild errors do not. Rate of wild errors are increased by increasing the noise amplitude η_p. Panel (b) shows the proportion of transitions that make read

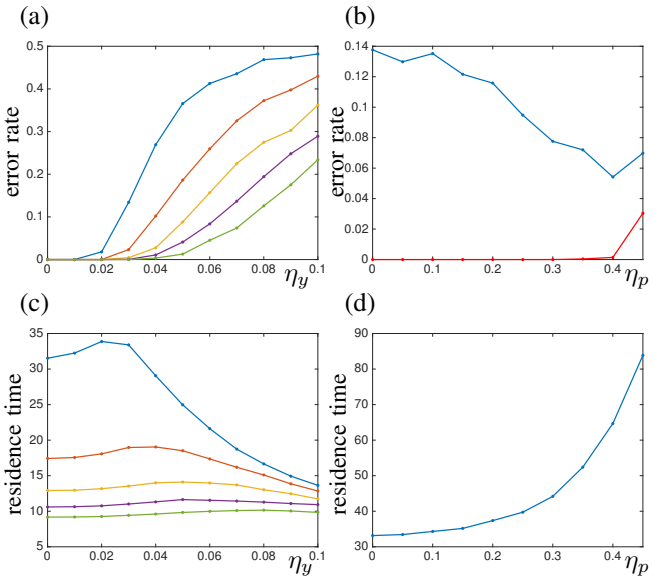

Fig. 9. The figures show error rates (a,b) and mean residence times (c,d) for the recurrent network as the noise parameters η_y and η_p are varied, for an excitable network with $\nu = 0.05$. In (a) and (c), η_y is varied, and each line is for a different value of ζ: blue, orange, yellow, purple, green correspond to $\zeta = (0.01, 0.02, 0.03, 0.04, 0.05)$. We set $\eta_p = 0.0001$, and only read errors are shown (wild errors are extremely rare). In (b) and (d), η_p is varied, and we set $\zeta = 0.01$ and $\eta_y = 0.03$. In (b), read errors are shown in blue and wild errors are shown in red. In each case, error rates and mean residence times are computed after the system is run long enough to observe $5,000$ transitions. All other parameters are as the standard set in Appendix A.

errors or wild errors, and panel (d) shows the mean residence time as the noise parameter η_p is varied. Examination of these figures shows two non-intuitive results. First, in panel (c), we see that the mean residence time is non-monotonic in η_y: we might expect that larger noise decreases the residence time, but this is not always the case. Second, in panel (b) we see that as η_p is increased, this initially has the effect of *decreasing* the rate of read errors and although the rate of wild errors increases the total rate of errors clearly also decreases. We explain both of these effects in the next two subsections.

A. *Non-monotonicity of residence time with increasing* η_y

Consider first the dynamics of the excitable system with no noise. Then, for $\zeta > \zeta_c$ (described in Appendix C), the equilibria ξ_i and ξ_{ij} as described in section II and Figure 1 (a) no longer exist, but have disappeared in a saddle-node bifurcation. There no longer exists an excitable network, but instead there is a periodic orbit. (Note that the periodic orbit exists in the recurrent network. In TMs which contain a halting state, the dynamics will not be periodic, although the dynamics past the previous location of the equilibria would be very similar.) In the literature on noisy perturbation of periodic orbits, it has previously been noted that counter-intuitive behaviour may emerge as increasing noise amplitude causes a progressively larger region of the nonlinear dynamics near the periodic orbit to be explored. This includes stochastic resonance effects in the absence of periodic forcing [13] and even reversal of direction of motion or bistability near the orbit [28].

In this study, a surprising observation is the non-monotonicity of the mean period (Figure 9(c)) on adding low amplitude noise. Although we do not perform a detailed analysis here, the reason is that small amplitude noise adds to the tail of distribution of first return times while larger amplitude noise causes accelerated crossing of the slow region in the periodic orbit, near the saddles. The crossover (and maximum mean period) therefore occurs when η_p and ζ are of comparable magnitude.

B. Decreasing rate of read errors with increasing η_p

To understand why increasing the noise parameter η_p can decreases the rate of read errors, it is helpful to consider the noise-free dynamics of the system (9) in a subspace with only one y variable and one p variable non-zero, such that there is an excitable connection from that p variable in that y direction. The dynamics is governed by:

$$\dot{p} = A_6 p(1-p^2) \tag{11}$$

$$\dot{y} = y((y^2-1)^2 + A_1 - A_2 p^2) + \zeta \tag{12}$$

where ζ is present only if there is input in that y direction. In Figure 10 (a) we show trajectories from the full system (9) projected onto a two-dimensional plane, where the y and p variables are chosen to correspond to the node which is currently being visited and the direction in which the trajectory escapes, respectively. In the upper panel, the noise in the p-direction is larger ($\eta_p = 0.4$) than in the lower panel ($\eta_p = 0.05$), and as expected, the variance of the trajectory in the p direction is much larger. However, notice that the mean of the p-variable is also shifted: in the lower panel the mean appears to be at approximately $p = 1$, but the mean in the upper panel is less than $p = 1$. We interpret this via the strong attraction of the unit sphere in the p-dynamics [4]: for increasing noise amplitude the noise spreads the distribution mostly along the surface of the unit sphere in p.

This reduction in mean p for larger noise has the consequence of changing the rate of escape near the node. To illustrate why this is, in panel (b) we show the nullclines of the two-dimensional system given by equations (11) and (12). The grey dashed line is the p-nullcline and the solid lines are the y-nullclines, for $\zeta = 0$ (black), $\zeta = 0.002$ (red) and $\zeta = 0.01$ (blue). Notice how the upper y-nullcline shifts to the left as ζ increases, meaning that escape from $y = 0$ becomes easier as ζ is increased, or conversely, for fixed ζ, escape becomes harder as p is reduced from $p = 1$. We demonstrate this by showing in panel (c) the mean time to escape for a one-dimensional escape problem (from $y = 0$ to $y = 1$) given by equation (12) with fixed p. Each line is for a different value of ζ, with again $\zeta = 0$ (black), $\zeta = 0.002$ (red) and $\zeta = 0.01$ (blue). The change in ratio of escape times for the system with $\zeta = 0$ and the system with non-zero ζ is what changes the probability of escaping in the correct direction. We show this by using the escape times in panel (c) to compute the probability of escaping in the (desired) direction with ζ non-zero rather than in a direction with $\zeta = 0$. Again the blue curve is for $\zeta = 0.01$ and the red curve is for $\zeta = 0.002$. We notice that as p is decreased from $p = 1$ the probability of leaving in the direction with non-zero

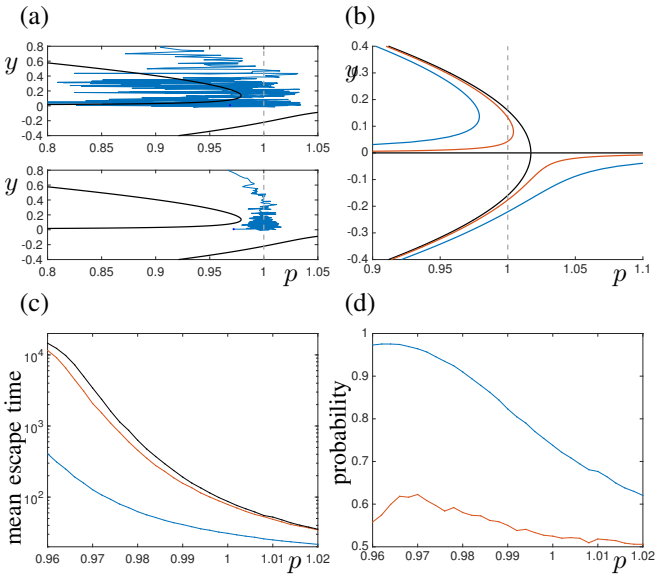

Fig. 10. The figures show aspects of the dynamics in a p-y subspace for the "recurrent" network. (a) shows two trajectories (in blue) of the full system, projected onto the two-dimensional subspace for the system given by equations (11) and (12). The grey dashed line is the p-nullcline and the black curves are the y-nullclines. In the top panel, $\eta_p = 0.4$, and in the lower panel, $\eta_p = 0.05$. In both, $\zeta = 0.01$, $\eta_y = 0.03$ and other parameters are as in the standard set in Appendix A. (b) shows the nullclines for the two-dimensional system (11) and (12) as ζ is varied. The dashed grey line is the p-nullclines and the solid lines are the y-nullclines, for $\zeta = 0$ (black), $\zeta = 0.002$ (red) and $\zeta = 0.01$ (blue). Panel (c) shows the mean residence times for an escape processes in the y-variable governed by (12) with constant p, for various ζ, colours as in (b). Finally, (d) shows the relative probability of escape in the y_1 direction for a two dimensional escape problem. Further details are given in the text.

ζ (i.e. the probability of making a correct transition) initially increases. As p decreases further, the probability reaches a maximum (this can be seen more clearly for the curve for $\zeta = 0.002$). This is because for smaller p the y-nullclines (see panel (b)) become closer together and so the dynamics of the non-zero ζ case asymptotes towards the $\zeta = 0$ case. We expect that as p decreases further, the probability will limit to 0.5.

Overall, as p is decreased, the probability of making a correct transition increases. Thus, as η_p increases, the trajectory spends more time in a region where p is smaller, and hence the overall probability of making an error decreases.

VI. Discussion

The purpose of this paper is to give an explicit way to embed a classical finite state TM in the nonlinear dynamics of a nonautonomous stochastic differential equation interacting with an external "tape". This is based on [4] and uses an explicit construction of network attractors with arbitrary graph structure. It also allows controllable switching between heteroclinic and excitable regimes of dynamics, which we interpret as allowing us to use the excitability parameter ν to switch between zero threshold for $\nu < 0$ and a small but finite threshold for $\nu > 0$.

We verify that reliable computation is possible in this setup (i.e. error rates can be made very small) depending on the type of network attractor considered and the according balance of

perturbation size, noise amplitude, speed of the computation. Clearly also the speed of writing and reading tape symbols will provide limits to speed, but we have not fully quantified the trade-offs in this paper.

The model we present is an explicit nonlinear dynamical model that may be helpful to understand more general cases of continuous state dynamical system (such as an ESNs [21], [22] or UMMs [36]) used to model a neural computational process in phase space. It particularly gives insight to cases where the transitions between states may occur via a competition between spontaneous instabilities (heteroclinic connections between states) or where there are thresholds that need to be crossed to effect a transition (excitable connections between states). Note that the excitability refers to excitability of the emergent states in phase space rather than to coupled networks of excitable units (where emergent dynamics may, for example, be periodic). Although we consider a single excitability parameter, common to all connections, this can clearly be generalised to consider excitability parameters that differ on each connection. Our work gives more insight into how computations of a UMM [10] can be undertaken following the trajectories of an underlying nonlinear system.

The sensitivity of the system is of particular interest: the computation takes place even for inputs that may be arbitrarily small for the heteroclinic case, while in the excitable case there is a finite threshold for inputs - although this may be made arbitrarily small by choosing an excitability parameter close to zero. The free-running heteroclinic case gives residence times in each state determined by the time it takes perturbations to grow to a point where nonlinear terms take over - in this case the feedback is asynchronous. The excitable case will have residence times that will be determined by the timing of the first perturbation that exceed threshold.

An intriguing insight is that changes in the computational properties of the system (in particular, speed and error rates) can be induced by global changes in excitability of the states. As previously noted, in the brain it is possible that changes in concentration of neuromodulators (e.g. dopamine, GABA, serotonin) cause effects through changes not just in heteroclinic connections [34] - they may also potentially switch connections between heteroclinic and excitable behaviour in such computational networks. For example, a reduction in excitability of the states in our system is analogous to a neuromodulator that depresses excitability - and will typically lead to both slower computations and changed error rates, according to Figure 7. This may be helpful in understanding the effects of neuromodulators on cognitive function and timing, as well as providing models for neural malfunction in disease.

We do not discuss possible learning/training strategies for this system, although both qualitative learning (such as learning the dependence of tape symbol on the transition between states, or learning sequences) and quantitative learning (such as the rate of transition or excitability of individual state) are possible using the framework as considered in [21]. Our study suggests that the excitability parameter will be an interesting parameter for adaptation: it gives a trade-off between computational speed and reliability.

Finally, we mention that one of the current major challenges of machine-based computation is to reduce the power usage in massively parallel computational systems [12]. Since large perturbations typically result in more energy expenditure than small ones, a computational strategy using a system close to a excitable/heteroclinic bifurcation may suggest novel insights and designs of computational systems that give an optimal balance between noise tolerance, energy usage and speed of calculation.

ACKNOWLEDGMENTS

PA gratefully acknowledges the financial support of the EPSRC via grant EP/N014391/1. CMP acknowledges travel funding from the University of Auckland and support from the London Mathematical Laboratory. We thank Antony Galton, Lorenzo Livi and the referees for valuable comments concerning this work, including pointing out relevant literature.

REFERENCES

[1] Asarin E Collins P: Noisy Turing machines. *Automata, Languages and Programming*, Lecture Notes in Computer Science **3580**:1031-1042, 2005.

[2] Ashwin P, Borresen J: Discrete Computation Using a Perturbed Heteroclinic Network. *Phys Letts A* **347**:208–214, 2005.

[3] Ashwin P, Postlethwaite C: On designing heteroclinic networks from graphs. *Physica D* **265**:26–39, 2013.

[4] Ashwin P, Postlethwaite C: Designing heteroclinic and excitable networks in phase space using two populations of coupled cells. *J Nonlinear Science*, DOI:10.1007/s00332-015-9277-2 (online) 2015.

[5] Ashwin P, Postlethwaite C: Quantifying noisy attractors: from heteroclinic to excitable networks. *SIAM J. Appl. Dynamical Systems* **15**:1989–2016, 2016.

[6] Ashwin P, Orosz G, Wordsworth J, Townley S: Dynamics on networks of clustered states for globally coupled phase oscillators. *SIAM J. Appl. Dyn. Sys.* **6**(4):728–758, 2007.

[7] Ashwin P, Coombes S, Nicks R: Mathematical frameworks for oscillatory network dynamics in neuroscience. *Journal of Mathematical Neuroscience* **6**:1–92, 2016.

[8] Bianchi F M, Livi L, Alippi C: Investigating Echo-State Networks dynamics by means of recurrence analysis. *IEEE TNNLS*, DOI:10.1109/TNNLS.2016.2630802, 2016.

[9] Bournez O, Campagnolo ML: A survey on continuous time computations, in Cooper, S.B.; Löwe, Benedikt; Sorbi, Andrea (Eds.), *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 383-424 Springer, 2008.

[10] Di Ventra M, Traversa F L, Ovchinnikov I V: Topological field theory and computing with instantons, *Annalen der Physik 2017*, 1700123, 2017.

[11] Field M: Heteroclinic networks in homogeneous and heterogeneous identical cell systems. *J. Nonlinear Sci.*, **25**:779-813, 2015.

[12] Furber SB, Galluppi F, Temple S, Plana LA: The SpiNNaker Project. *Proceedings of the IEEE* **102**:652–665, 2014.

[13] Gang H, Ditzinger T, Ning C Z, Haken H: Stochastic resonance without external periodic force, *Phys. Rev. Letts* **71**:807–810 1993.

[14] Graça DS, Campagnolo DL, Buescu J: Computability with polynomial differential equations, *Advances in Applied Mathematics* **40**:330–349, 2008.

[15] Greff K, Srivastava R K, Koutník J, Steunebrink B R, Schmidhuber J: LSTM: A Search Space Odyssey, *IEEE TNNLS* to appear 2017.

[16] Hopcroft J, Ullman J: *Introduction to Automata Theory, Languages, and Computation* Addison-Wesley, Reading Mass 1979.

[17] Izhikevich E M: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*, MIT press, 2007.

[18] Kuznetsov, Yu A: *Elements of applied bifurcation theory*, Applied mathematical sciences **112**, Springer, 1998.

[19] Lin S, Rado T: Computer studies of Turing machine problems. *J. ACM* **12**:196–212, 1965.

[20] Livi L, Bianchi FM, Alippi C: Determination of the edge of criticality in echo state networks through Fisher information maximization, *IEEE TNNLS*, DOI:10.1109/TNNLS.2016.2644268, 2017.

[21] Jaeger H: Adaptive nonlinear system identification with Echo state networks. *Proceedings of the 15th International Conference on Neural Information Processing Systems*:609-616, MIT Press Cambridge, 2002.

[22] Jaeger H, Maass W, Principe J: Special issue on echo state networks and liquid state machines: Editorial. *Neural Networks*, **20**:287–289, 2007.

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

[24] Maass W, Joshi P, Sontag ED: Computational Aspects of Feedback in Neural Circuits *PLoS Computational Biology* **3**:e165, 2017.

[25] Marxen H, Buntrock J: Attacking the Busy Beaver 5. *Bulletin of the EATCS* **40**:247–251, 1990

[26] McCulloch W, Pitts, W: A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* **5**:115–133, 1943.

[27] Neves, FS, Timme, M: Computation by Switching in Complex Networks of States. *Phys. Rev. Letts* **109**:018701, 2012.

[28] Newby J M, Schwemmer M A: Effects of moderate noise on a limit cycle oscillator: counterrotation and bistability. *Phys. Rev. Letts*, **112**:114101, 2014.

[29] Rabinovich M I, Varnoa P, Selverston A I, Abarbanel H D I: Dynamical principles in neuroscience, *Rev. Mod. Phys.* **78**(4):1213, 2006.

[30] Rabinovich M I, Varona P, Tristan I, Afraimovich V S: Chunking Dynamics: Heteroclinics in Mind. *Frontiers in Computational Neuroscience* **8**:00022, 2014.

[31] Siegelmann H A, Sontag E D: On the computational power of neural nets. In: *J. Comput. Syst. Sci.* **50.1**:132150, 1995

[32] Siegelmann H A: Turing on Super-Turing and adaptivity, *Progress in Biophysics and Molecular Biology*, **113**:117–126, 2013.

[33] Sima J, Orponen P: General-Purpose Computation with Neural Networks: A Survey of Complexity Theoretic Results *Neural Computation* **15**:27272778, 2003.

[34] Shaw K M, Lyttle D N, Gill J P, et al: The significance of dynamical architecture for adaptive responses to mechanical loads during rhythmic behavior. *Journal of Computational Neuroscience* **38**:25–51, 2015.

[35] Sloman A, Chrisley R: Virtual Machines and Consciousness, *Journal of Consciousness Studies*, **10**:133-172, 2003.

[36] Traversa F L, Di Ventra M: Polynomial-time solution of prime factorization and NP-complete problems with digital memcomputing machines, *Chaos* **27**:023107, 2017.

[37] Tsuda I: Chaotic itinerancy and its role in cognitive neurodynamics, *Current Opinion in Neurobiology* **31**:67–71, 2015.

[38] Turing A M: On computable numbers, with an application to the Entscheidungsproblem. *Proc. LMS*, **42**:230–265, 1936.

APPENDIX

A. Construction of dynamics realising network attractors

As outlined in [4] we consider a system of coupled autonomous ordinary differential equations (AODEs) that realises a arbitrary directed graph $G = (\mathcal{V}, \mathcal{E})$ as a heteroclinic or an excitable network. The graph G is defined by the set of vertices $\mathcal{V}$ and set of edges $\mathcal{E}$. There are n_v vertices v_i which are each associated with an equilibrium ξ_i in the AODEs, and n_e edges e_j. Each edge e_j has a "starting" and "ending" equilibrium, given respectively by $v_{\alpha(j)}$ and $v_{\omega(j)}$. The AODEs are of the form given in equation (6) with f defined by:

$$\begin{aligned} f_i(p,y) &= [p_i(A_6(1-p^2) + A_4(p_i^2p^2 - p^4)) \\ &\quad + A_5(-Z_i^{(o)}(p,y) + Z_i^{(i)}(p,y)] \\ f_{n_v+j}(p,y,\nu) &= \left[B\left(y_j, A_1 - A_2p_{\alpha(j)}^2 + A_3(y^2 - y_j^2)\right)\right] \end{aligned} \tag{13}$$

for $i = 1, \cdots, n_v$ and $j = 1, \cdots, n_e$, where $p^2 = \sum_{j=1}^{n_v} p_j^2$, $p^4 = \sum_{j=1}^{n_v} p_j^4$, $y^2 = \sum_{j=1}^{n_e} y_j^2$ and $A_1, \ldots, A_6$ are constants except for A_2 which depends on ν. The function B is defined by

$$B(y_k, \lambda) = -y_k\left((y_k^2 - 1)^2 + \lambda\right) \tag{14}$$

while the inputs to the p_i cells from the y cells are:

$$\begin{aligned} Z_i^{(o)}(p,y) &= \sum_{\{k\ :\ \alpha(k)=i\}} y_k^2 p_{\omega(k)} p_i \\ Z_i^{(i)}(p,y) &= \sum_{\{k'\ :\ \omega(k')=i\}} y_{k'}^2 p_{\alpha(k')}^2 \end{aligned} \tag{15}$$

(NB there is a sign error in the definition of Z^o in [4] that we have corrected here).

For $\eta \equiv 0$ the system is an ODE and ξ_j denote the unit basis vectors $(p, y) \in \mathbb{R}^{n_v+n_e}$: the first n_v correspond to unit vectors where one of the p_j is non-zero. As shown in [4], the subspaces

$$P_\ell = \{(p,y)\ :\ y_k = 0 \text{ if } k \neq \ell \text{ and } p_j = 0 \text{ if } j \neq \alpha(\ell) \text{ or } \omega(\ell)\}$$

for $\ell = 1, \ldots, n_e$ are invariant for the flow generated by system (9) and for suitable choice of parameters contain connections that realise the graph G as a heteroclinic/excitable network embedded in phase space.

We choose default parameters

$$A_1 = 0.5,\ A_2 = 1.5 - \nu,\ A_3 = 2,\ A_4 = 10,\ A_5 = 4,\ A_6 = 2. \tag{16}$$

For $\nu < 0$ close to zero this realises a heteroclinic network, while for $\nu > 0$ close to zero it realises an excitable network with a small threshold. In all cases we choose a threshold to classify the state that is $h = 0.29$. The case $\nu = 0$ corresponds to bifurcation between the two types of network: see [4, Fig. 4] for more details and justification that the networks are heteroclinic/excitable for these parameter values. For numerical solution of the NSDE we use a Heun integration method with fixed timestep $dt = 0.05$.

B. Removal of self-loops from the Turing machine

In this appendix we describe how to remove self-loops from the TM. Suppose there exists some i and j such that $\rho(q_i, s_j) = (\tilde{q}_{ij}, \tilde{s}_{ij}, \tilde{\sigma}_{ij})$ and $q_i = \tilde{q}_{ij}$. Then this is a *self-loop* in the graph of internal states of the TM. Due to the restrictions of not allowing self-loops (homoclinic loops) in the heteroclinic network defined by equation (9), we define a variant of the TM that avoids self-loops in the graph of states.

Suppose that the graph of states has n_S self loops, and let $S = \{(i_1, j_1), \ldots, (i_{n_s}, j_{n_s})\}$ be the set of pairs of indices of states and symbols corresponding to self loops. We then append n_S additional *self-loop states* q_{n_Q+i} $(i = 1, \ldots, n_S)$ to the TM, and define a new transition function $\rho^\star$ for our variant TM as follows.

- For $i = 1, \ldots, n_Q$, and $(i, j) \notin S$, then
$$\rho^\star(q_i, s_j) = \rho(q_i, s_j) = (\tilde{q}_{ij}, \tilde{s}_{ij}, \tilde{\sigma}_{ij}).$$
- For $i = 1, \ldots, n_Q$ and $(i, j) = (i_l, j_l) \in S$ then
$$\rho^\star(q_i, s_j) = (q_{n_Q+l}, \tilde{s}_{ij}, \tilde{\sigma}_{ij}),$$
where $\tilde{s}_{ij}$ and $\tilde{\sigma}_{ij}$ are defined by ρ, given above.
- For $i = n_Q + 1, \ldots, n_Q + n_S$, write $l = i - n_Q$ and then
$$\rho^\star(q_{n_Q+l}, s_j) = (q_{i_l}, N, 0),$$

for all $j \in 1, \ldots, n_\Gamma$, and where i_l is the first component of the pairs in the set S defined above.

We have added the action N which writes nothing on the tape, and the action 0 which does not move the head, corresponding together to a "do nothing" action when returning from the self-loop. Note that this construction creates n_Γ connections back from each extra state to the state which originally had the self-loop, one for each of the possible tape symbols.

The new TM has $N_Q = n_Q + n_S$ internal states.

C. Computation of ζ_c

In this appendix we show how to compute an approximation for ζ_c, the minimum value of input required for transitions in the excitable network case, in the case where there is no noise (as described in section V).

We consider a subspace of the system (9) with $p_1 = 1$, y_1, y_2 non-zero, and all other coordinates zero. We assume there are connections from p_1 in the y_1 and y_2 directions, and input in the y_1 direction only (i.e. $z_{n_v+1} = 1$ and all other z_i are zero), resulting in the two-dimensional system

$$\begin{aligned}\dot{y}_1 &= -\nu y_1 + 2y_1^3 - A_3 y_1 y_2^2 - y_1^5 + \zeta \\ \dot{y}_2 &= -\nu y_2 + 2y_2^3 - A_3 y_2 y_1^2 - y_2^5\end{aligned}$$

For $0 < \nu \ll 1$ and $\zeta = 0$ this system has a stable equilibrium on the coordinate axis at $y_1 = \sqrt{\nu/2} + O(\nu)$, $y_2 = 0$, and an unstable equilibrium at $y_1 = y_2 = 0$. As ζ is increased, these two equilibria disappear in a saddle-node bifurcation at $\zeta = \zeta_c \equiv \frac{\sqrt{2}}{3\sqrt{3}}\nu^{3/2} + O(\nu^2)$. For $\zeta > \zeta_c$ trajectories which start near the origin will move in the y_1 direction until y_1 becomes O(1), and for trajectories close to, but not in this subspace will move away, affected by the other dynamics in the system.

Peter Ashwin is a Professor of Mathematics at the University of Exeter. His research interests are nonlinear dynamical systems and applications thereof, especially for systems that are coupled and/or symmetric: in these settings it is often possible to find attractors that have network structure. He is also interested in ergodic properties of dynamical systems, nonautonomous dynamics and computational modelling for a variety of applications.

Claire Postlethwaite is an Associate Professor of Mathematics at the University of Auckland. Her research interests include applications of dynamical systems and mathematical ecology. Within the area of applied dynamical systems, one of her main interests is understanding the dynamics near heteroclinic cycles and networks, and using them as models for physical systems.