Noname manuscript No.
(will be inserted by the editor)

Equation-free analysis of spike timing dependent plasticity

Carlo R. Laing · Ioannis G. Kevrekidis

Received: date / Accepted: date

Abstract Spike-timing dependent plasticity is the process by which the strengths of connections between neurons are modified as a result of the precise timing of the action potentials fired by the neurons. We consider a model consisting of one integrate-and-fire neuron receiving excitatory inputs from a large number — here, 1000 — of Poisson neurons whose synapses are plastic. When correlations are introduced between the firing times of these input neurons, the distribution of synaptic strengths shows interesting, and apparently low-dimensional, dynamical behaviour. This behaviour is analysed in two different parameter regimes using equation-free techniques, which bypass the explicit derivation of the relevant low-dimensional dynamical system. We demonstrate both coarse projective integration (which speeds up the time integration of a dynamical system) and the use of recently-developed data-mining techniques to identify the appropriate low-dimensional description of the complex dynamical systems in our model.

Keywords Spike timing dependent plasticity · equation-free · model reduction · neuronal network

1 Introduction

Neurons typically communicate with one another via trains of action potentials — each characterised by a transient increase and then decrease in the cell's membrane potential [17,27]. When an action potential from one neuron (the "presynaptic") arrives at the synaptic connection to the dendrite of another (the "postsynaptic") a current is induced, via a series of biochemical processes, that flows into the postsynaptic neuron. This current may make the postsynaptic neuron more likely to fire an action

C. R. Laing
Institute of Natural and Mathematical Sciences, Massey University, Private Bag 102-904 NSMC, Auckland, New Zealand.
Tel.: +64-9-414 0800 extn. 43512
Fax: +64-9-443 9790
E-mail: c.r.laing@massey.ac.nz

I. G. Kevrekidis
Department of Chemical and Biological Engineering and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, 08544, USA
Tel.: 609-258-2818
E-mail: yannis@princeton.edu

potential, in which case the connection is said to be "excitatory," or less likely ("inhibitory"). But the strength of the connection between two neurons, i.e. the magnitude of the current that flows into the postsynaptic neuron for each presynaptic action potential, is not fixed. Rather, such strengths are plastic, and their modification is regarded as the cellular basis for development and learning [6].

One common form of synaptic plasticity is spike-timing dependent plasticity (STDP) [5,36,43,11]. In this mechanism, the strength of the connection from a presynaptic neuron to a postsynaptic one is modified based on the precise firing times of the two neurons, relative to one another. Put simply, if the presynaptic neuron fires in a short window *before* the postsynaptic does, the connection is strengthened. If it fires *after* the postsynaptic, the connection is weakened. The effect is more pronounced the closer in time the two neurons fire. Clearly this mechanism can lead to competition between neurons, as a strong connection from one neuron to another will make the firing of the presynaptic more likely to result in the firing of the postsynaptic, resulting in strengthening the connection, and vice versa for weak connections. It can also lead to unbounded values of synaptic strength, which can be addressed in several ways [44,52,25]. This assumes that the connection is excitatory, which we will do here.

STDP has been studied experimentally since the mid-1990s [5,4,36] and has been the subject of many computer simulations and corresponding analysis [28,52,49,25,21,44,39,48,22,41,35,1]. From a dynamical systems point of view STDP has some interesting properties. Most simulations of networks with STDP are stochastic, in the sense that the firing times of some (or all) neurons are chosen stochastically. However, the distributions of synaptic strengths often seem to approach some approximately stationary state [49]. Another property, of primary interest here, is the slowness of the evolution of the distributions of synaptic strengths, relative to the dynamics of individual action potentials. These different timescales make simulations of networks with STDP very time consuming [41], but also suggest that recently-developed "equation-free" (EF) techniques [32,53,31,30,16,50] could be useful for accelerating such simulations and analysing the networks' dynamics. Roughly speaking, the EF approach provides a framework for accelerating the evolution towards — and analysing and predicting the existence of — long term stationary structures of a low-dimensional description of a dynamical system, even though the equations governing this description are not explicitly available [29]. A network with STDP is also of interest as it combines not only the dynamics *on* a network (i.e. the individual neurons' dynamics, which depend on the structure of the network) but also dynamics *of* a network, as the connection strengths evolve due to the dynamics of the neurons. This interplay may result in complex spatio-temporal behaviour [40]. Note that EF techniques have been used previously to study a network undergoing STDP giving rise to map formation in the laminar nucleus of the barn owl's auditory system [37].

Several authors have derived continuum-level descriptions of the synaptic strengths in networks of neurons undergoing STDP [52,44,25,10,39]. All have relied upon the smallness of the change in the strength of a synapse caused by a single pair of action potentials, which in turn makes the distribution of synaptic strengths evolve slowly. A Fokker-Planck equation for the evolution of the probability density of the synaptic strengths is often derived. The derivation of this equation involves several approximations, and often some of the terms in this equation cannot be calculated explicitly [10]. Alternatively, simplified model neurons such as the Poisson neuron must be used in order to make analytical progress [39,25,44, 9].

Our approach — being largely computer-assisted — does not require any of these assumptions, and is thus quite general. However, for the EF approach to be successful, other assumptions must be

satisfied [29]. Specifically, the system must be able to be described by a set of low-dimensional "coarse" variables. This implies the existence of a low-dimensional manifold onto which the system is rapidly attracted. Checking that this is the case, and determining what the coarse variables are, is a nontrivial issue. We also need a way of mapping between the coarse variable(s) and the full set of variables in the original network, in both directions ("lifting" and "restricting"). These issues will be addressed below.

The structure of the paper is as follows. We present the model network in Sec. 2; it is of the same form as that studied by several other groups [49,25]. We demonstrate a common numerical technique in Sec. 3, that of projective integration, i.e. the acceleration of time integration of a dynamical system. In Sec. 4 we choose a different parameter regime in which the network seems well-described by a particle moving in a double-well potential, under the influence of noise. We demonstrate data-mining techniques to support this conclusion and show how one can calculate properties of the network's behaviour using a single coarse variable (observable). We summarise with a conclusion in Sec. 5.

2 Model

We consider a model very similar to that of [49] and [25]. It consists of a single integrate-and-fire neuron with excitatory inputs from 1000 neurons and inhibitory inputs from 200. The strengths of the inhibitory synapses are fixed, but the strengths of the excitatory synapses evolve under the following rule: Define $\Delta t = t_{post} - t_{pre}$ to be the time difference between any pair of presynaptic and postsynaptic action potentials. Then the change in the strength of the synapse between the pre and postsynaptic neuron caused by this pair of firings is

$$\Delta g = \begin{cases} \lambda\,(1-g)^{\sigma}\, e^{-|\Delta t|/\tau}, & \Delta t > 0 \\ -\alpha\lambda g^{\sigma} e^{-|\Delta t|/\tau}, & \Delta t < 0 \\ 0, & \Delta t = 0 \end{cases} \tag{1}$$

where g is the strength of the synapse immediately before the interaction occurs ($0 \le g \le 1$), $\tau = 20$ msec, λ is an overall "learning rate" and σ and α are parameters. Note that the maximum increase in g caused by one pair of firings is λ, while the magnitude of the maximum decrease is $\alpha\lambda$. If the change Δg makes g negative we set $g = 0$, and if the change makes $g > 1$ we set $g = 1$. If $\sigma = 0$ we recover the dynamics of [49]. Rubin et al. [44] considered the cases $\sigma = 0$ and $\sigma = 1$, while [25] considered the effects of varying σ between 0 and 1.

To actually implement the model, the integrate-and-fire neuron obeys

$$\tau_m \frac{dV}{dt} = V_r - V + g_{max} g_{ex}(V_{ex} - V) + g_{in}(V_{in} - V) \tag{2}$$

where $g_{max} = 0.015$, $V_r = -70$mV, $V_{ex} = 0$mV, $V_{in} = -70$mV and $\tau_m = 20$ msec, together with the rule that once V reaches -54mV it is reset to -60mV, and the neuron is deemed to have fired. The inhibitory input consists of 200 independent Poisson processes, each having a rate of 10Hz. (This rate is fixed throughout the paper.) When an inhibitory input occurs, $g_{in} \mapsto g_{in} + \overline{g}_{in}$, where $\overline{g}_{in} = 0.05$, and otherwise

$$\tau_i \frac{dg_{in}}{dt} = -g_{in} \tag{3}$$

The excitatory input consists of $N = 1000$ independent Poisson processes, each having a rate specified in the text. When an excitatory input at synapse a occurs, where $a \in \{1, \dots N\}$, $g_{ex} \mapsto g_{ex} + g_a$, and otherwise

$$\tau_e \frac{dg_{ex}}{dt} = -g_{ex} \tag{4}$$

We have $\tau_e = \tau_i = 5$ msec. The quantity g_{max} is chosen to be dimensionless and the conductances g_{ex} and g_{in} are measured in units of leak conductance of the neuron and thus they and the associated quantities $\overline{g}_{in}$ and g_a are also dimensionless. The plasticity in (1) is implemented using the $N + 1$ dimensionless variables M and P_a for $a = 1, \dots N$. These satisfy

$$\tau \frac{dM}{dt} = -M \qquad \text{and} \qquad \tau \frac{dP_a}{dt} = -P_a \tag{5}$$

and every time the postsynaptic neuron fires, $M \mapsto M - \lambda\alpha$, and every time (excitatory) synapse a fires, $P_a \mapsto P_a + \lambda$. M is used to decrease synaptic strengths: if synapse a fires at time t, we have $g_a \mapsto \max[g_a + M(t)g_a^\sigma, 0]$. (Note that M is negative.) P_a is used to increase the strength of synapse a: if the postsynaptic neuron fires at time t, we have $g_a \mapsto \min[g_a + P_a(t)(1 - g_a)^\sigma, 1]$.

This system has been well-studied [25,49], and typical behaviour is shown in Fig 1. When σ is small the g_a typically split into two groups, one having large values and the other having small values. The firing rate of the excitatory inputs controls the relative number in each group: for a higher firing rate there are more synapses in the "weak" group [49]. Increasing σ moves the two "modes" of the distribution together until they eventually merge. Increasing λ for small σ "fills-in" the gap between the two populations seen in Fig. 1(c). One point of interest to us is the apparent slowness of the distribution evolution dynamics. As seen from Fig. 1 (a and b) it can take on the order of 1000 seconds for the system to reach an equilibrium distribution, even though the longest explicit timescale in the model is of order 10 msec. It is this separation of timescales between the microscopic dynamics (of the integrate-and-fire neuron, the synaptic dynamics, and the variables M and the P_a) and the macroscopic dynamics (of the probability density of the g_a) that we will exploit in Sec. 3.

The dynamics of this network are fairly simple, but more interesting behaviour can occur when correlations are introduced in the firing times of the excitatory inputs [39,25,48]. We will consider the case where the 1000 excitatory inputs are divided into two groups of 500 each, and correlations are introduced such that any two spike train inputs within one group have correlation coefficient c, but there are no correlations between any spike train from one group and any spike train from the other group. This is done as follows [39,25,22]. To solve the ODEs (2)-(5) time is discretised into steps of $\delta t = 0.05$ msec, and the forward Euler method is used. At the nth timestep the state of excitatory input a, X_a^n, is either 1, with probability $f_a \delta t$ (corresponding to neuron a firing) or 0 (if neuron a is not firing), where f_a is the firing rate of neuron a (constant over time, and the same for all neurons within the same group). Thus the state of all 500 excitatory neurons in one group at the nth timestep can be described as a vector with 500 elements $[X_1^n \; X_2^n \; \dots \; X_{500}^n]$. To introduce correlations, a "phantom" input neuron with index 0 is created, and this neuron's dynamics are statistically the same as any of the other neurons in the group, i.e. its firing times are generated from a Poisson process with rate $f = f_a$. Then at the nth timestep, and for each $a \in \{1, \dots 500\}$, with probability $\sqrt{c}$ we replace X_a^n by X_0^n (thus keeping the original X_a^n with probability $1 - \sqrt{c}$). Typical behaviour when $c = 0.01$ is shown in Fig. 2, where we have chosen all g_a in a population to initially be equal, but with a difference between

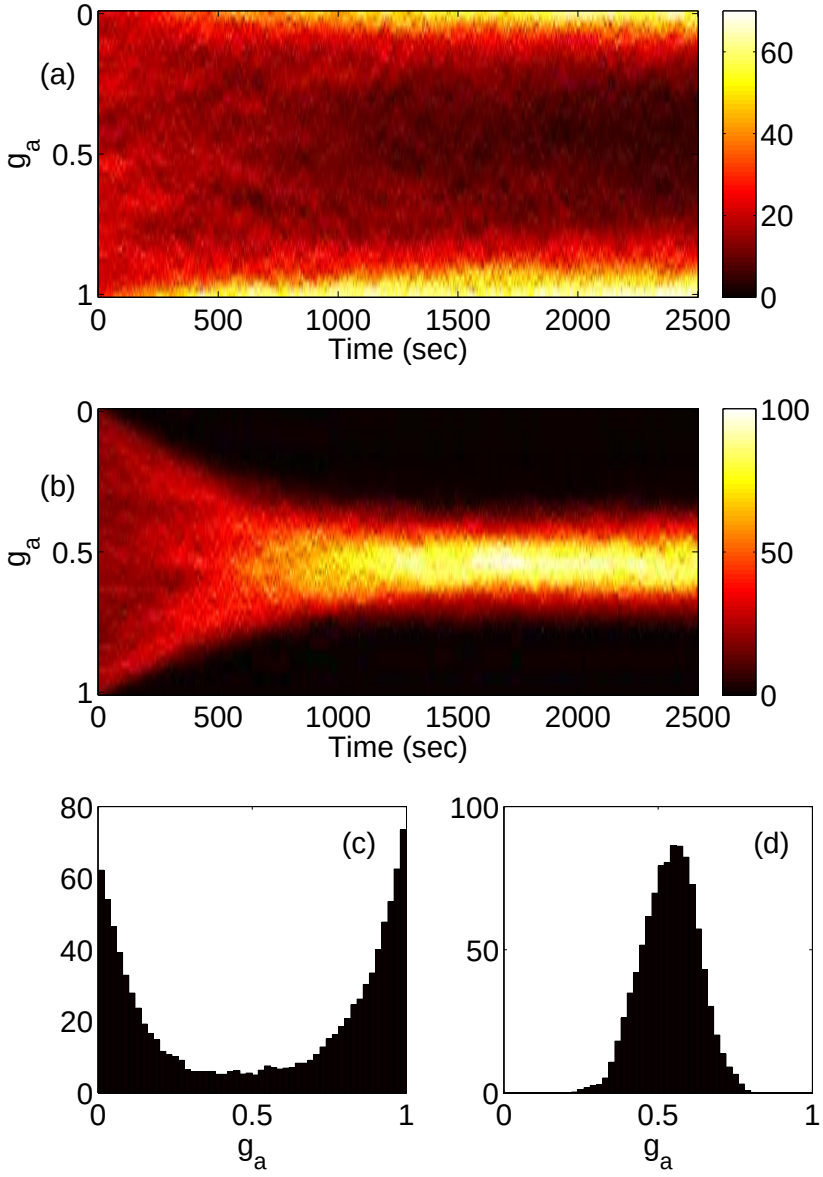

Fig. 1 (a) and (b): density plots of the g_a as a function of time. Initially the g_a are randomly and uniformly chosen from $[0, 1]$. (c): steady state distribution when $\sigma = 0.01$, corresponding to panel (a). (d): steady state distribution when $\sigma = 0.1$, corresponding to panel (b). Parameters are $\sigma = 0.01$ (a) and $\sigma = 0.1$ (b). $\lambda = 0.005, \alpha = 1.05$. Firing rate of excitatory inputs is 10 Hz.

populations, so that the evolution of each population is clear. The final distribution is bimodal, with the two modes corresponding to the two populations. (This happens even if all g_a are initially the same.) The evolution of the distributions is again slow, on the order of hundreds of seconds. In Sec. 3 we show how to speed up the simulation of these dynamics.

Note that there are two main strategies for simulating networks such as the one considered here: time-stepping and event driven [8]. Time-stepping is described above — ODEs are solved approximately by taking small time steps, and events are detected. In event-driven simulation the network is stepped from event to event, updating variables at each event. Such a strategy requires the equations to be

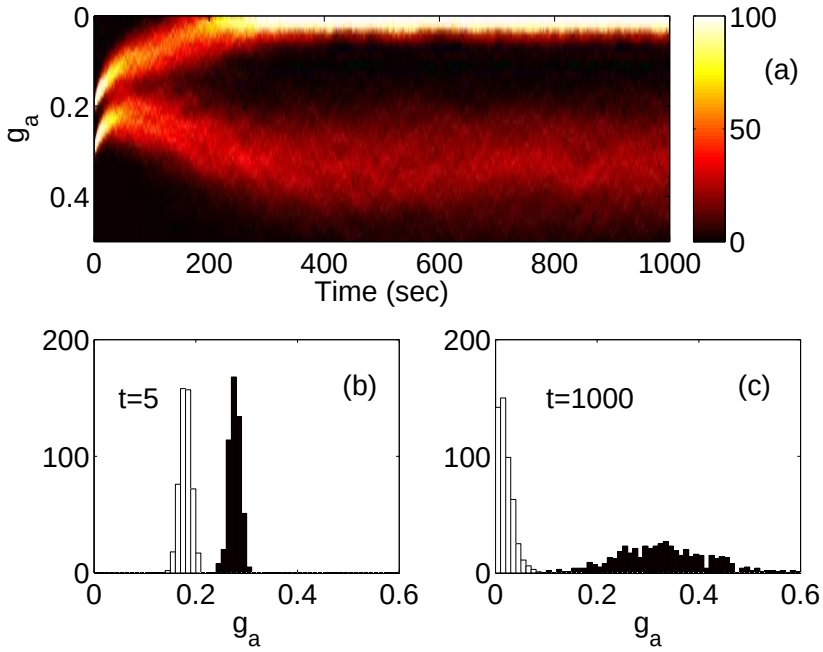

Fig. 2 (a): density plot of the g_a as a function of time. (b) and (c): histograms showing the distributions of the g_a for the two populations at $t = 5$ sec (b) and $t = 1000$ sec (c). Intially all g_a in one population were 0.3 and in the other, 0.2. Other parameters: $\lambda = 0.001, \alpha = 1.05, \sigma = 0.01, c = 0.01$. Firing rate of excitatory inputs is 40 Hz.

analytically solvable, and has been implemented for several models of STDP, including the one discussed here [7,12].

3 Projective integration

Coarse projective integration is a common EF technique used to speed up the time integration of a complex dynamical system [20,29]. We will demonstrate it here using projective integration of distributions [19]. The idea is that since the distributions of the g_a are changing slowly, we should be able to integrate the distributions forward in time, using much larger time steps than those used to integrate the full underlying system. The main issues are: how do we describe (parameterise) a distribution, and how do we find the evolution of these parameters, which we regard as variables in a low-dimensional description of the dynamical system? We will parametrise a distribution by expanding the inverse of the cumulative density function (CDF) as a finite sum of orthogonal polynomials; other possibilities exist [45]. Figure 3(a) shows the CDF for the two populations whose evolution is shown in Fig. 2, at $t = 1000$. The inverses of these functions are shown in panel (b). The horizontal axis must go from 0 to 1, and we use x to denote this variable. (We can obtain these plots by simply plotting the sorted values of the g_a within each population.) We now expand the functions shown in Fig. 3(b) in orthogonal polynomials on $[0, 1]$, with uniform weight. These are the shifted Legendre polynomials, which satisfy

$$\int_0^1 P_n(x)P_m(x)dx = \frac{\delta_{nm}}{2n+1} \tag{6}$$

The first few are $P_0(x) = 1, P_1(x) = 2x - 1, P_2(x) = 6x^2 - 6x + 1$.

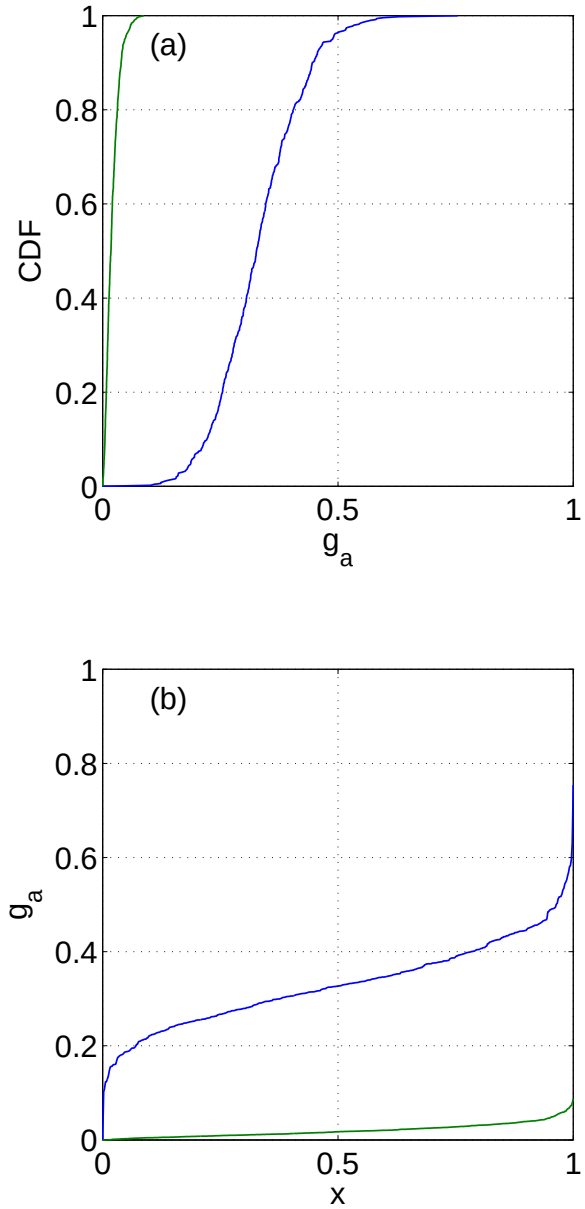

Fig. 3 (a): cumulative distribution functions for the two populations whose evolution is shown in Fig. 2, at $t = 1000$. (b): inverses of the CDFs shown in panel (a). Parameters as in Fig. 2.

Thus we write

$$g_a^j(x,t) \approx \sum_{i=0}^{q} a_i^j(t) P_i(x) \qquad j = 1,2 \tag{7}$$

where the superscript labels the population. From now on we set $q = 5$, as this has been found to be high enough to accurately capture the dynamics. Ideally, this value should be monitored as parameters are varied (by comparing results obtained using q polynomials with those obtained using $q-1$ or $q+1$ polynomials) and increased or decreased as necessary in order to accurately describe the full dynamics. The a_i^j can be easily extracted from a simulation of the full system at any time t by minimizing the

quantity

$$S = \sum_{k=1}^{500} \left[g_a^j(x_k, t) - \sum_{i=0}^{q} a_i^j(t) P_i(x_k) \right]^2 \tag{8}$$

where $x_k = (k - 0.5)/500$. Minimizing S is nothing more than linear least squares fitting of data like that in Fig. 3(b) to a linear combination of the polynomials $P_0(x)$ to $P_q(x)$ [32]. In equation-free terminology, this is the restriction operator, which maps from the g_a to the coefficients a_i^j. An example is shown in Fig. 4 for the simulation in Fig. 2, where only the results for $0 \leq t \leq 700$ are presented. As expected, these coefficients vary slowly, with a small amount of "noise" present. Interestingly, only the first two coefficients seem to be significant, with the higher-order coefficients fluctuating around zero. This suggests that the system is effectively low dimensional. The EF approach is to assume that the results in Fig. 4 can be obtained by integrating a dynamical system for only these variables, i.e. that the system "closes" at this level.

We now address the question of evolving these coefficients, given just their values. To do this we need to *lift, run* and then *restrict*, and then *project.* We proceed to explain these terms. Lifting involves initialising the full system in a way that is consistent with the values of the a_i^j. We use (7) to do this: given the a_i^j, we evaluate the right hand side (RHS) of (7) at $x = (k - 0.5)/500$ for $k = 1, 2 \ldots 500$. The other variables are initialised (rather arbitrarily) as $P_a = 0\ \forall a$, and others are randomly chosen from uniform distributions, the ranges of which are given in Table 1. The precise values to which these variables are initialised are not important, as due to the separation of timescales in the system, the variables will equilibrate over the times for which we observe the distribution of synaptic strengths. More specifically, the time-constants for the evolution of V, M, g_e, g_i and the P_a are between 5 and 20 msec (see Sec. 2) and these are the characteristic times over which these variables change.

Variable	V	M	g_e	g_i
Range	$[-60, -56]$	$[-0.001, 0]$	$[20, 25]$	$[0, 0.1]$

Table 1 Ranges of randomly chosen initial conditions for projective integration.

We then run the full network for 1 second, four times, with different realisations of the random processes (firing times and initial conditions) each time. For each of these four realisations we extract the time series of the a_i^j as above and then average across the realisations to give one time series per coefficient. Examples are shown in Fig. 5 for a_i^2. We then fit a straight line through each of these time series over the interval $[0.25, 1]$. The slope of this line is then our estimate of the rate of change of the corresponding coefficient; see Fig. 5. Using this technique we can effectively estimate the RHS of the dynamical system

$$\frac{d\mathbf{a}}{dt} = F(\mathbf{a}) \tag{9}$$

where $\mathbf{a} = (a_0^1\ a_1^1\ \ldots\ a_{q-1}^2\ a_q^2)^T$ and F is — by definition — the function describing the rate of change of $\mathbf{a}$, and thus integrate (9) forward in time.

A simulation time of 1 second for these bursts is long enough for the fast variables to have time to equilibrate, but short enough that the slow variables of interest change in only a linear fashion over this interval. Of course, averaging over more than 4 bursts will increase the accuracy of our estimate of

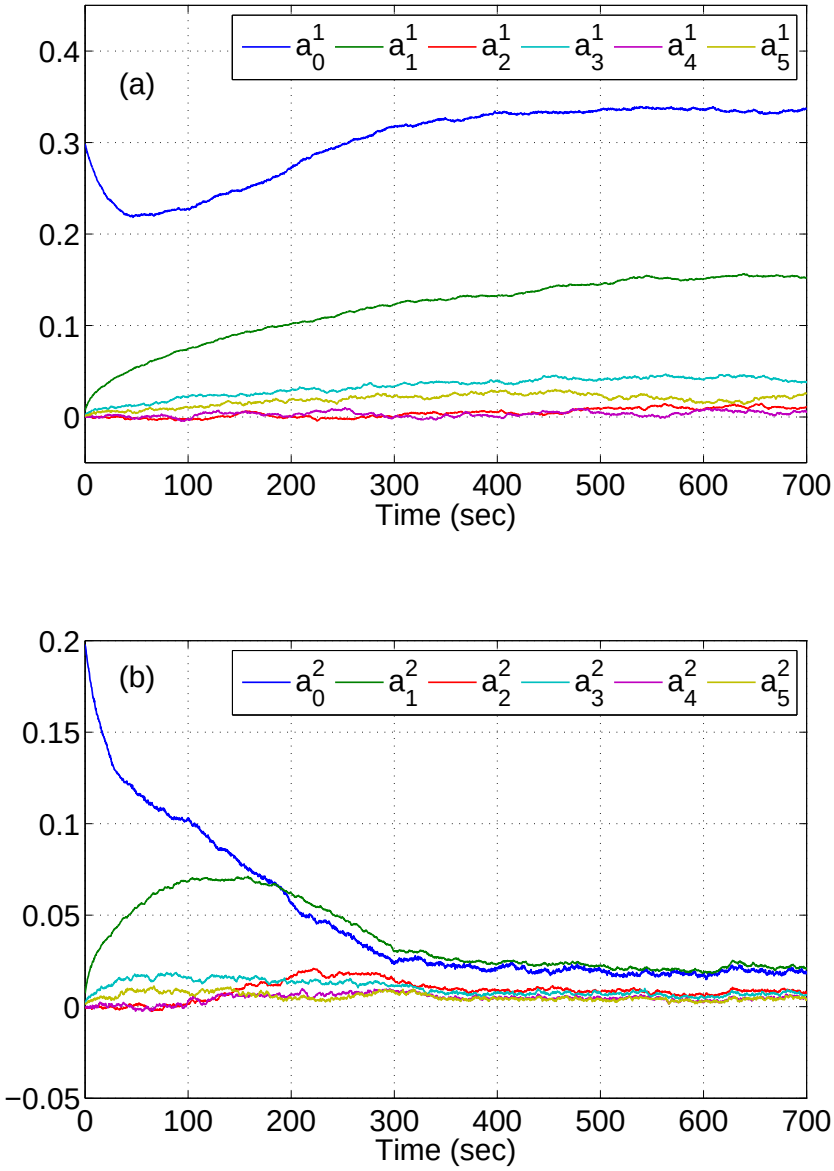

Fig. 4 Coefficients of the orthogonal polynomial expansion of the g_as whose evolution is shown in Fig. 2 for population 1 (panel (a)) and population 2 (panel (b)). Parameters as in Fig. 2.

$F(\mathbf{a})$, but at the expense of requiring more computation. Note in Fig. 5 that $\mathbf{a}$ changes slightly during the first 0.25 sec. of integration, and thus we are not estimating $F(\mathbf{a})$ at exactly the value of $\mathbf{a}$ at which we initialised the network, but at a close point nearby. This slight discrepancy can be removed by using implicit techniques [38].

The projective integration in Fig. 6 was done using Euler's method, with a fixed time-step of 4 seconds. Comparing with the results in Fig. 4, and keeping in mind that we are simulating a stochastic dynamical system, the agreement appears visually very good. Each time-step involves one evaluation of the RHS of (9) and each evaluation involves averaging over 4 bursts of length 1 second. Thus we need 4 seconds of simulation in order to take one step of 4 seconds, giving no speed-up (ignoring the cost of lifting, restricting, and taking the Euler step, which is reasonable). However, the bursts can trivially

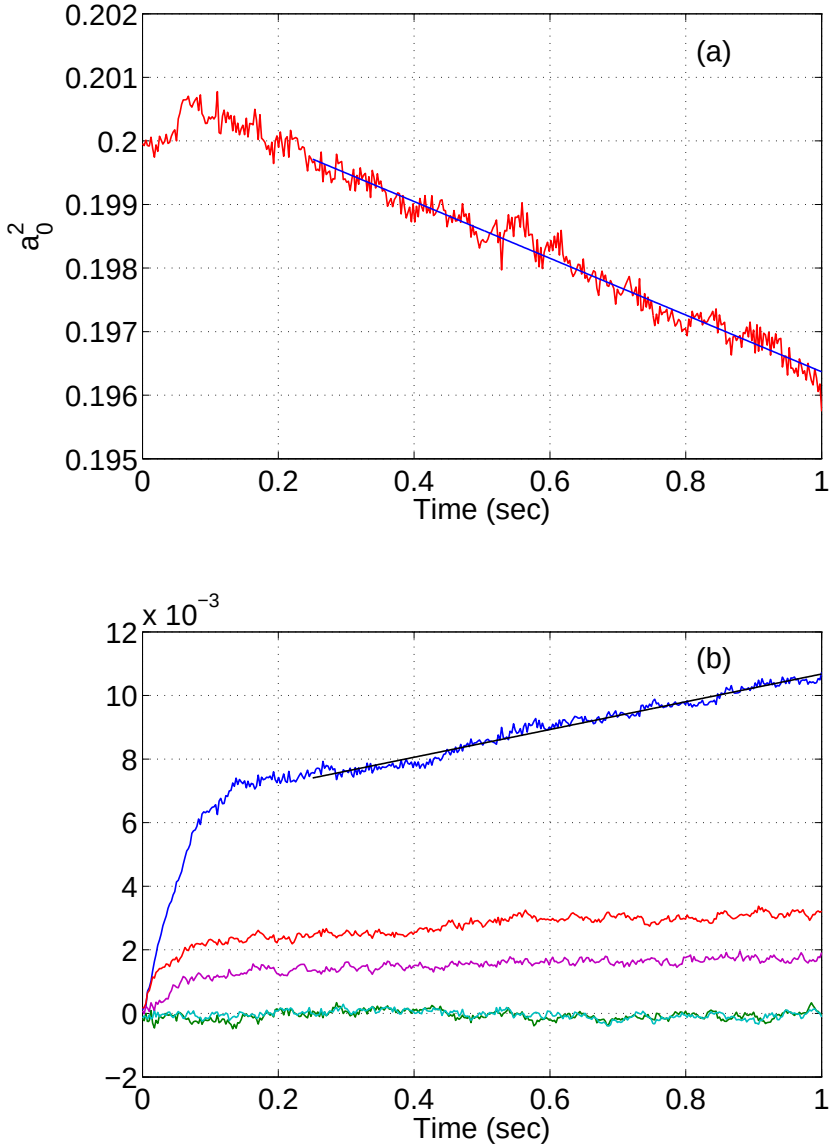

Fig. 5 Dynamics of coefficients in (7) during a short burst. Initially, $a_0^2 = 0.2$ and all other $a_i^2 = 0$. (a): a_0^2 and the best straight line fit over $0.25 < t < 1$. (b): a_i^2 for $i = 1, \ldots 6$. a_1^2 is shown in blue, and the best straight line fit to this data over $0.25 < t < 1$ is also shown. Parameters as in Fig. 2.

be done in parallel on a multicore processor, whereas there seems to be no obvious way to parallelise a direct simulation of the full system. Thus on a quad-core system, as was used for these calculations, the "wall clock" speedup is 4. There is no communication between processors during the bursts, as the only information that must be passed to them is the current value of **a**, and the only information they have to return is the values of **a** over the burst.

The time-step of 4 seconds was chosen to demonstrate the advantages of projective integration. Using a larger time-step than this results in larger fluctuations of the variables in Fig. 6, and using a smaller time-step (while more accurate) results in less of a speedup. As with any numerical scheme, one can vary parameters associated with it, such as the number of bursts averaged over and the time-step used for Euler's method, and systematically investigate the effects of these changes [33]. Note that we are not

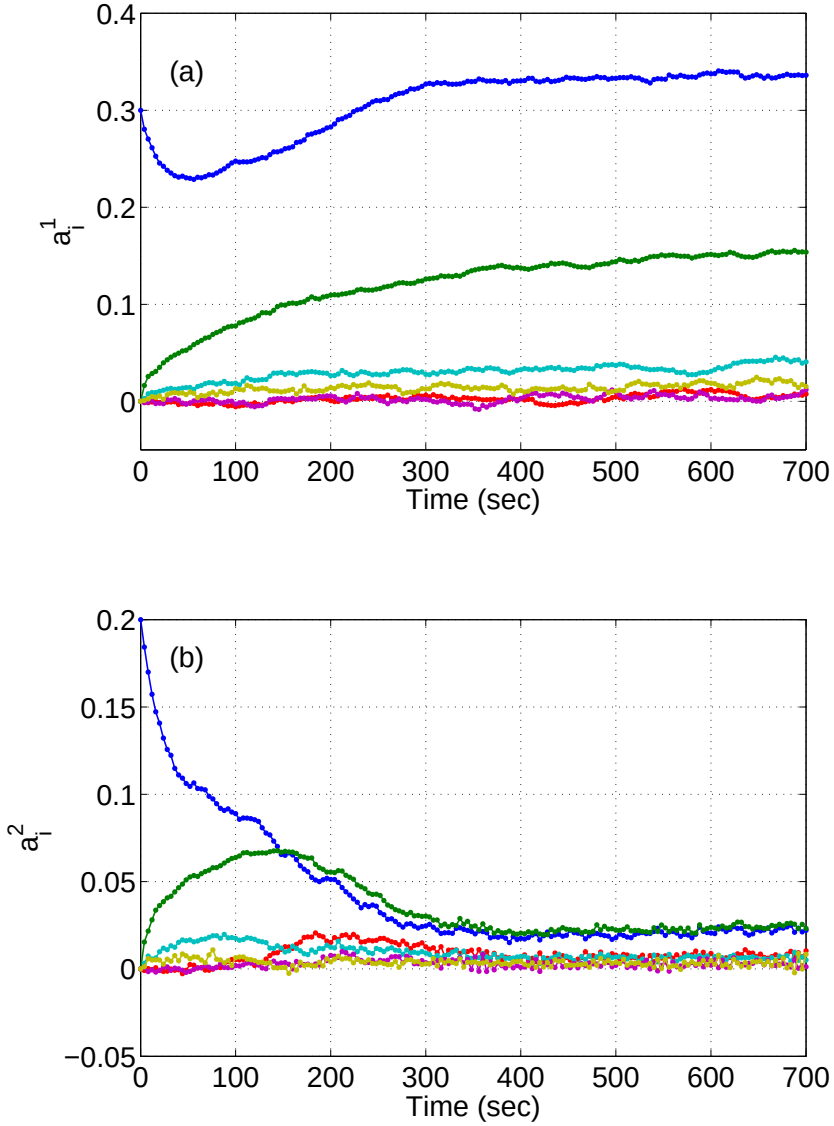

Fig. 6 Projective integration of (9). In each panel the blue curve shows a_0 and the green one, a_1. Compare with Fig. 4. Parameters as in Fig. 2.

restricted to Euler's method: any other valid scheme for time-stepping could be used. We could also use Newton's method to find zeros of (9) and their dependence on parameters, but that is computationally intensive, as many realisations must be averaged over in order to accurately evaluate derivatives using finite differences (see, for example, [3] for discussion on this). We thus leave these ideas for future, more quantitative, publications.

Our coarse projective integration has been successful (for the parameter values used) because the full network can in principle be described by a low-dimensional deterministic system of the form (9). Also, the precise values of most of the variables, when initialising the full network at the start of each burst, do not crucially determine the evolution of the network, for reasons mentioned above. These statements may no longer be true at different parameter values, and should be checked, ideally in an automated fashion.

While there exist some analytical results regarding the evolution of the distribution of synaptic weights [52,10,25,39,44], these all rely on some form of approximation. We emphasise that our techniques do not rely on any approximations regarding the model's dynamics, and can be applied, essentially unchanged, as "wrappers" around other models. Similarities between our results and those cited directly above can be regarded as confirmation that the approximations used in those works are indeed valid. The acceleration of the simulation of a network via projective integration is significant, but also of interest is the discovery that so few coefficients are needed in the expansion (7) to accurately represent the distribution of synaptic strengths. Note also that finding *unstable* fixed points of (9) (which we could do in principle) cannot be done by just integrating the full network forwards in time; this is another advantage of the EF approach.

We now consider a different parameter regime and show that here the network can be well-described by a single collective variable which appears to satisfy a stochastic differential equation.

4 Data-mining and a stochastic differential equation

Consider again a network with 1000 excitatory neurons, split into two populations of 500 each. All excitatory neurons fire at 30Hz and there are correlations within each of the two populations, as above. We again have 200 inhibitory inputs firing at 10Hz. Compared with the parameter values in Sec. 3 we increase λ and α, and consider a much larger value of the correlation: $c = 0.5$. Typical behaviour is shown in Fig. 7. We see apparently stochastic switching between periods where one population of inputs is dominant, and where the other population is dominant. Such behaviour is reminiscent of a particle in a double-well potential, subject to noise [30,16,15]. Note the symmetry of the system: there is nothing distinguishing one subpopulation of 500 excitatory inputs from the other.

Assuming that the network can be described in a low-dimensional way, the first task is to identify appropriate variable(s) with which to describe the state of the system, and then to find the dynamics of those variables. We address the first question using data-mining, specifically, using diffusion maps [13], a nonlinear manifold learning algorithm. The full data-set we have is the collection of "snapshots" of the state of the network (i.e. the g_a, the P_a and V, g_{in}, g_{ex}, M) at $30,000$ timepoints, each separated by 0.025 sec. The g_a as a function of time are shown in Fig. 7 (only the first 2/3 of the data set is shown), and we assume that the dynamics of these variables suffices to describe the network's behaviour. At each of these $30,000$ timepoints the g_a are regarded as a vector in $\mathbb{R}^{1000}$. To perform our data-mining we use a subset of these, selecting every 10th vector, i.e. we mine the set of g_a at 3000 timepoints, each 0.25 sec apart. The underlying assumption behind diffusion maps is that these data points lie on a low-dimensional (probably nonlinear) manifold in a 1000-dimensional space. We would like to find the dimensionality of this manifold, and to construct a coordinate system on it.

Let $\mathbf{x}_i \in \mathbb{R}^{1000}$ be the set of g_a at the ith such time-point, where $i \in \{1, 2, \ldots 3000\}$. We define a 3000×3000 matrix K by

$$K_{i,j} = \exp\left[-\left(\frac{||\mathbf{x}_i - \mathbf{x}_j||}{\varepsilon}\right)^2\right] \tag{10}$$

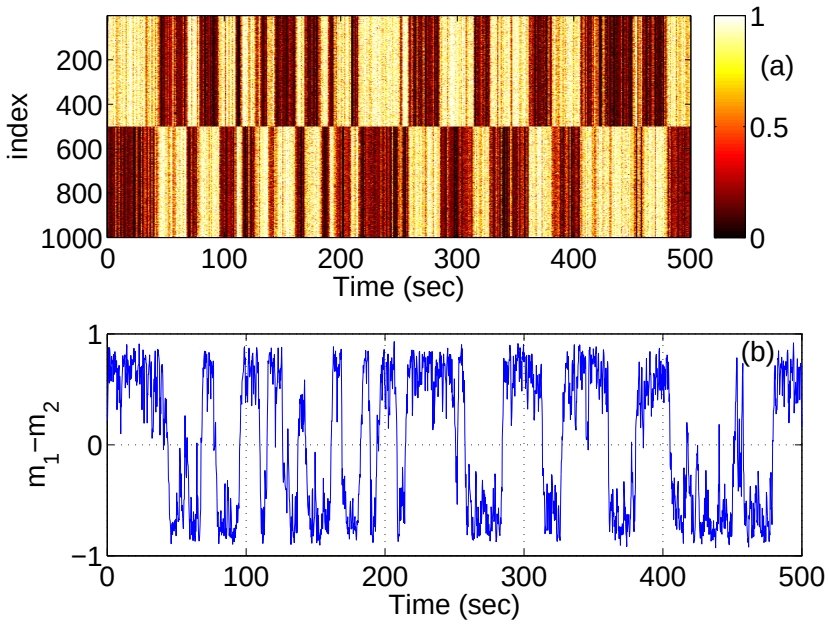

Fig. 7 The behaviour of a network with 1000 excitatory inputs at 30Hz and 200 inhibitory inputs at 10Hz. (a): the g_a. (b): the difference between the means of the two populations, $m_1 - m_2$. Other parameters: $g_{max} = 0.015, g_{in} = 0.05, \lambda = 0.01, \alpha = 1.5, \sigma = 0.01, c = 0.5$.

where $||\cdot||$ indicates the Euclidian norm and we choose $\varepsilon = 20$. We then define the diagonal matrix $\widehat{D}$ by

$$\widehat{D}_{i,i} = \sum_{j=1}^{3000} K_{i,j} \tag{11}$$

and the matrix $M = \widehat{D}^{-1}K$. The matrix M has eigenvalues $1 = \lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{2999} \geq 0$ with right eigenvectors $\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots \boldsymbol{\nu}_{2999}$. The eigenvector $\boldsymbol{\nu}_0$ is constant (all entries are 1). If there is a spectral gap after several eigenvalues, this suggests that the data set $\{\mathbf{x}_i\}$ is low dimensional, and the leading eigenvectors $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots$ provide a useful low-dimensional description of the data [31,30,15,47] (assuming that data in these dimensions is not a "harmonic" of directions already found [18]).

The spectral gap between the first and second non-trivial eigenvalues, λ_1 and λ_2, is shown in Fig. 8. This spectral gap strongly suggests that the data shown in Fig. 7 lies on a 1-dimensional manifold (a curve). The theory behind diffusion maps then says that the first non-trivial eigenvector, $\boldsymbol{\nu}_1$, provides a coordinate system along this curve. Specifically, the diffusion mapping is

$$\mathbf{x}_i \mapsto \boldsymbol{\nu}_1^{(i)} \in \mathbb{R} \tag{12}$$

for $i = 1, \ldots, 3000$, where $\boldsymbol{\nu}_1^{(i)}$ is the ith component of $\boldsymbol{\nu}_1$. This is a mapping from $\mathbb{R}^{1000}$ to $\mathbb{R}$. We refer to the unbolded ν_1 as the scalar "diffusion map coordinate" and $\boldsymbol{\nu}_1^{(i)}$ is a specific value of ν_1. Figure 9 shows the data from Fig. 7 projected onto the first two diffusion map coordinates, ν_1 and ν_2. We see that while there is some variation in the ν_2 direction, position in this data "cloud" can be parametrised by just the ν_1 coordinate.

The diffusion map coordinate ν_1 is plotted as a function of time in Fig. 10, along with ν_2. (The plots of $\nu_3, \ldots \nu_6$ all appear similar to the plot of ν_2, not shown.) By comparing the top panel of Fig. 10 with

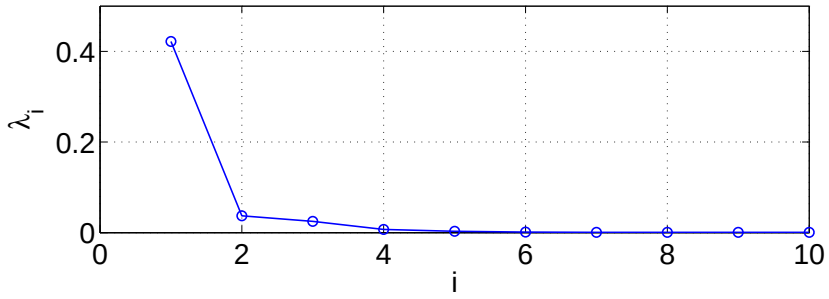

Fig. 8 First few eigenvalues of M, sorted by decreasing size.

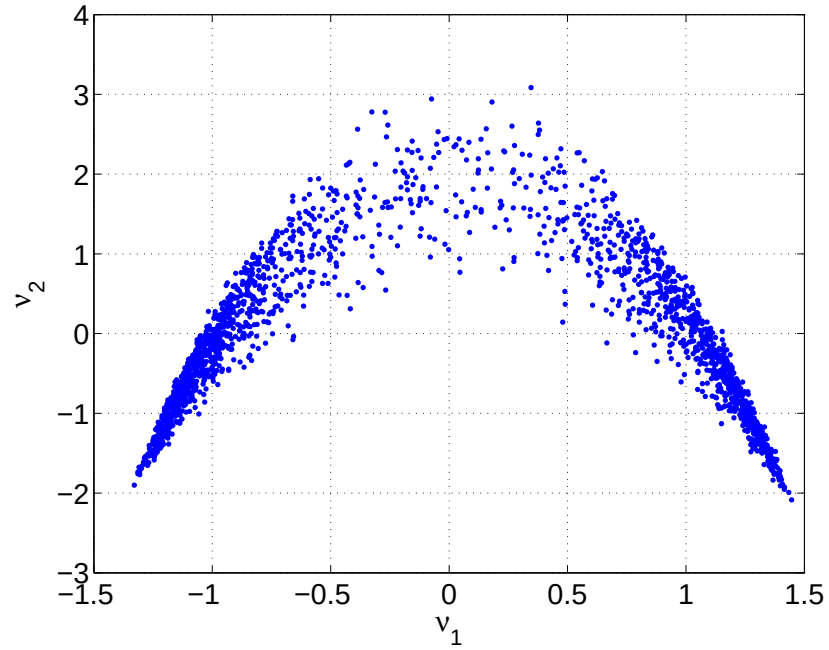

Fig. 9 Projection of the data shown in Fig. 7 onto the first two diffusion map coordinates, ν_1 and ν_2.

Fig. 7 we see that ν_1 seems to capture the dynamics of the network, with ν_1 taking on positive values when population 1 is active and negative values when population 2 is active. Indeed, if we define m_i to be the average of the g_a within population i, plotting $m_1 - m_2$ versus ν_1, as done in Fig. 11 shows that ν_1 captures the difference between the means of the two populations extremely well.

Having determined that ν_1 is an appropriate variable with which to describe the state of the system, we now need to find the dynamics of ν_1. Our assumption is that ν_1 satisfies a Langevin stochastic differential equation, and that the probability density function $P(\nu_1, t)$ satisfies the corresponding Fokker-Planck equation

$$\frac{\partial P(\nu_1,t)}{\partial t} = \left[-\frac{\partial}{\partial \nu_1}\mu(\nu_1) + \frac{\partial^2}{\partial \nu_1^2} D(\nu_1) \right] P(\nu_1, t) \tag{13}$$

One can estimate the functions μ and D from their definition [24,42,53,31,30]:

$$\mu(\nu_1) = \lim_{\Delta t \to 0} \left. \frac{\langle \nu(t+\Delta t) - \nu(t) \rangle}{\Delta t} \right|_{\nu(t)=\nu_1} \tag{14}$$

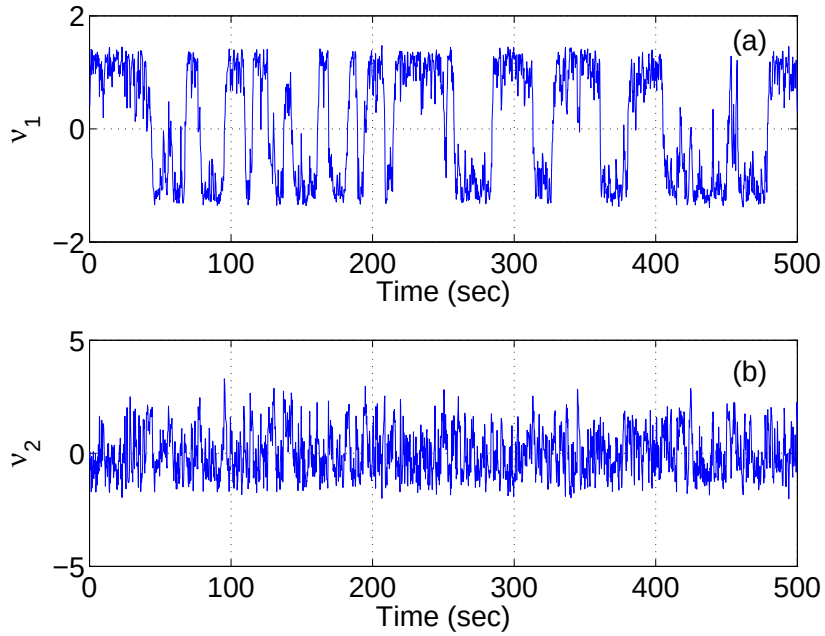

Fig. 10 The first two diffusion map coordinates, ν_1 (a) and ν_2 (b), as functions of time for the simulation shown in Fig. 7.

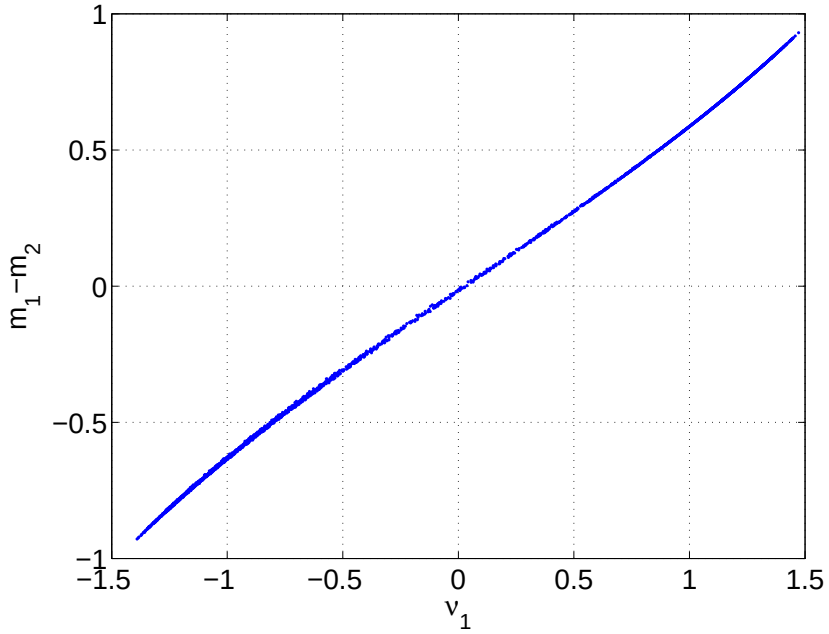

Fig. 11 The difference between the means of the two populations, $m_1 - m_2$, plotted against the first diffusion map coordinate, ν_1.

and

$$D(\nu_1) = \lim_{\Delta t \to 0} \left. \frac{\langle [\nu(t+\Delta t) - \nu(t)]^2 \rangle}{2\Delta t} \right|_{\nu(t)=\nu_1} \tag{15}$$

where angle brackets indicate averaging. In practice we use the approximations

$$\mu(\nu_1) \approx \left. \frac{\langle \nu(t+\Delta t) - \nu(t) \rangle}{\Delta t} \right|_{\nu(t)=\nu_1} \tag{16}$$

and

$$D(\nu_1) \approx \left. \frac{\langle [\nu(t+\Delta t) - \nu(t)]^2 \rangle}{2\Delta t} \right|_{\nu(t)=\nu_1} - \frac{\Delta t [\mu(\nu_1)]^2}{2} \tag{17}$$

where the last term is a finite-Δt correction [53,42]. In order to use these approximations we need (a) a large amount of data (so that averaging produces good results), and (b) for the data points to be closely spaced in time (so that small Δt is a good approximation to the limit $\Delta t \to 0$). Using only the data exploited in the data-mining is not a good idea, since there are only 3000 data points (and this cannot be increased by much, given the size of the matrices involved in the diffusion map algorithm). Instead, we will use all of our data, i.e. the $30,000$ data points, each separated by $\Delta t = 0.025$ sec. In order to do so, we need the values of ν_1 corresponding to all of these data points. (Recall, we only have ν_1 values for the 3000 points used in the data-mining.) We can find these values using the Nyström formula for eigenspace interpolation.

To do this, we note that $M = \widehat{D}^{-1/2} S \widehat{D}^{1/2}$, where $S = \widehat{D}^{-1/2} K \widehat{D}^{-1/2}$. M and S are thus similar, and have the same eigenvalues, $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{2999}$. Let $\{\mathbf{U}_j\}_{j=0,\ldots,2999}$ be the corresponding eigenvectors of S. These are related to the eigenvectors of M through

$$\widehat{D}^{1/2} \boldsymbol{\nu}_j = \mathbf{U}_j. \tag{18}$$

We have $S\mathbf{U}_j = \lambda_j \mathbf{U}_j$, or

$$\mathbf{U}_j^{(i)} = \frac{1}{\lambda_j} \sum_{k=1}^{3000} S_{i,k} \mathbf{U}_j^{(k)} \tag{19}$$

where $\mathbf{U}_j^{(i)}$ is the ith component of $\mathbf{U}_j$. Suppose we have a new vector $\mathbf{x}_{new} \in \mathbb{R}^{1000}$, consisting of the values of the g_a at some time, and we want to know the value of ν_1 associated with it. We create an 3000×1 vector $\mathbf{K}_{new}$ whose kth component is

$$\mathbf{K}_{new}^{(k)} = \exp\left[-\left(\frac{||\mathbf{x}_{new} - \mathbf{x}_k||}{\varepsilon} \right)^2 \right]. \tag{20}$$

We also have a generalised kernel vector $\mathbf{S}_{new}$ whose ith entry is

$$\mathbf{S}_{new}^{(i)} = \left(\sum_{k=1}^{3000} K_{i,k} \right)^{-1/2} \left(\sum_{k=1}^{3000} \mathbf{K}_{new}^{(k)} \right)^{-1/2} \mathbf{K}_{new}^{(i)}. \tag{21}$$

The entries in $\mathbf{S}_{new}$ quantify the pairwise similarities between $\mathbf{x}_{new}$ and the vectors in $\{\mathbf{x}_k\}$, consistent with the definition of S. The eigenvector component $\mathbf{U}_j^{new}$ corresponding to $\mathbf{x}_{new}$ is then

$$\mathbf{U}_j^{new} = \frac{1}{\lambda_j} \sum_{i=1}^{N} \mathbf{S}_{new}^{(i)} \mathbf{U}_j^{(i)} \tag{22}$$

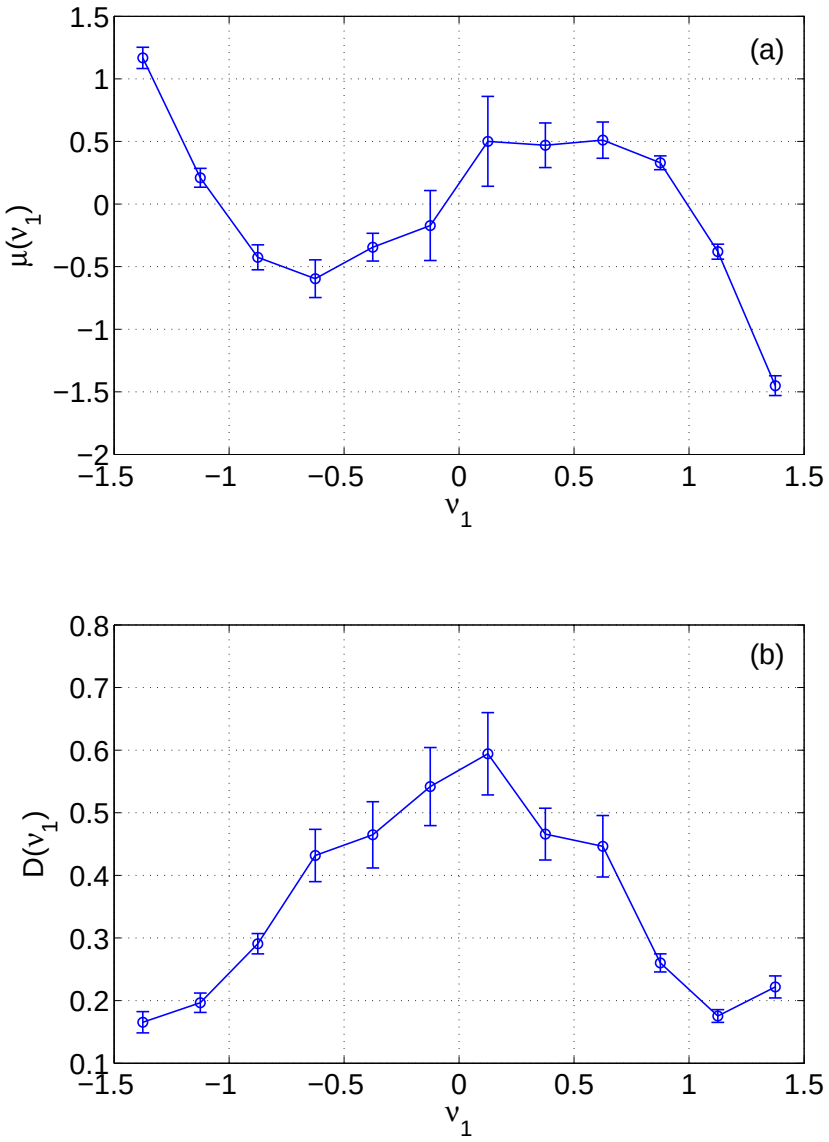

Fig. 12 Estimates of the functions $\mu(\nu_1)$ (a) and $D(\nu_1)$ (b) from 30,000 equally-spaced values of ν_1, using the approximations (16)-(17).

for $j = 0, 1$ (since we only want the value of ν_1) and the value of ν_1 corresponding to $\mathbf{x}_{new}$ is

$$\nu_1^{new} = \frac{\mathbf{U}_1^{new}}{\mathbf{U}_0^{new}}. \tag{23}$$

By performing this calculation for the 30,000 data points we obtain 30,000 values of ν_1, each 0.025 sec. apart, and can use these to estimate the quantities in (16)-(17). The results are shown in Fig. 12.

Several observations can be made. Firstly, the drift term, $\mu(\nu_1)$, has three zeros. If the diffusive term, $D(\nu_1)$, could somehow be set to zero, the dynamics of ν_1 would simply be $d\nu_1/dt = \mu(\nu_1)$, and this system has two stable fixed points at $\nu_1 \approx \pm 1$ and one unstable fixed point at $\nu_1 \approx 0$. Thus the reduced system for ν_1 can be usefully regarded as being bistable, but subject to noise. The other observation regards the error bars shown in Fig. 12. They tend to be large where there is little data and vice versa. We will address this issue below.

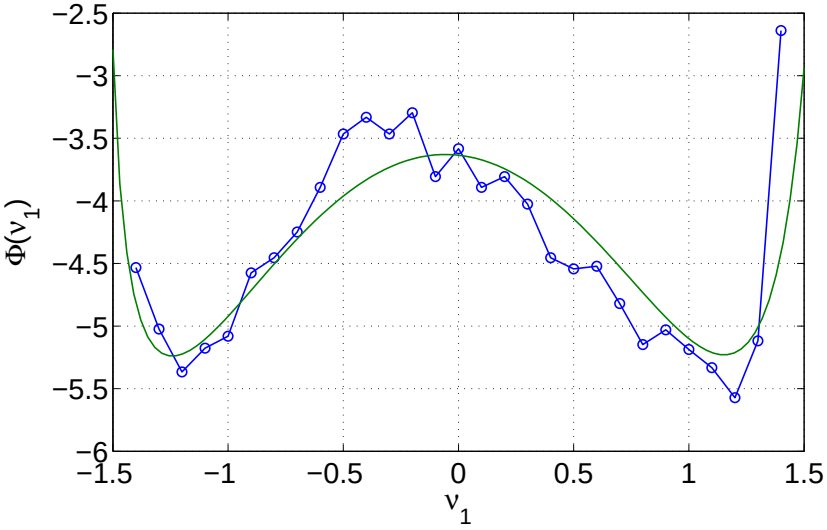

Fig. 13 Effective potential, $\Phi(\nu_1)$, as a function of ν_1. The solid curve comes from numerically evaluating (24), while the circles joined by lines show the result of binning the 30,000 values of ν_1 and inverting $P(\nu_1) \propto \exp[-\Phi(\nu_1)]$.

Having obtained the data in Fig. 12 we can now easily perform several calculations. The first is to estimate the effective potential $\Phi(\nu_1)$, related to $\mu(\nu_1)$ and $D(\nu_1)$ by

$$\Phi(\nu_1) = \text{const.} - \int_{\nu_m}^{\nu_1} \frac{\mu(s)}{D(s)} ds + \log D(\nu_1) \tag{24}$$

where ν_m is the minimum value of ν_1 [16,30]. The steady state of the Fokker-Planck equation (13) is given by $P(\nu_1) \propto \exp[-\Phi(\nu_1)]$. We fit a third order polynomial through the mean values shown in Fig. 12 (top) and a third order polynomial through the mean values in Fig. 12 (bottom), and then numerically evaluated the integral in (24). The result is shown in Fig. 13, along with the result determined by inverting $P(\nu_1) \propto \exp[-\Phi(\nu_1)]$. The underlying system is symmetric, but we have not forced this symmetry on μ, D nor Φ. The double-well potential is clearly seen.

Another calculation of interest is that of the distribution of switching times, i.e. the amount of time that the system will remain in one potential well before switching to the other. (Or the amount of time that the integrate-and-fire neuron is driven by one subpopulation, before switching to being driven by the other.) For a well at $\nu_1 = \nu_{min}$, the average switching time is given by

$$\tau \approx \frac{2\pi \exp(\Delta\Phi)}{\overline{D}\sqrt{-\Phi''(\nu_{min})\Phi''(\nu_{max})}} \tag{25}$$

where the local maximum of Φ occurs at $\nu_1 = \nu_{max}$, $\Delta\Phi \equiv \Phi(\nu_{max}) - \Phi(\nu_{min})$, and $\overline{D}$ is the average value of $D(\nu_1)$ over the range shown in Fig. 12. Calculating these for the potential found using (24) we found that the average time between switchings is about 10 seconds. The original (750 second) simulation had an average time between switchings of about 15 seconds, in excellent agreement. The distribution of escape times is shown in Fig. 14(a). We can also numerically integrate the associated Langevin equation

$$\frac{d\nu_1}{dt} = \mu(\nu_1) + \sqrt{2D(\nu_1)}\eta(t) \tag{26}$$

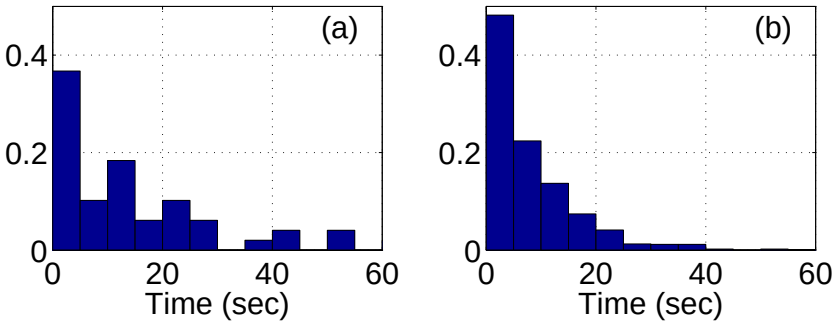

Fig. 14 (a) Distribution of escape times for the full simulation, duration 750 seconds. (b) Distribution of escape times for simulation of (26), duration 7500 seconds.

where $\eta(t)$ is uncorrelated Gaussian white noise with mean zero and unit variance. Doing so for a long time (7500 seconds) gives an average time between switches of about 8 seconds. The distribution of escape times for such a simulation is shown in Fig. 14(b).

As mentioned, the error bars in Fig. 12 vary across ν_1, as the density of ν_1 during the long simulation is not uniform. One way to address this, and obtain better estimates of $\mu(\nu_1)$ and $D(\nu_1)$, is to repeatedly initialise the full system at a specific value of ν_1, run a short burst from this initial condition, process the results using the Nyström formula to find how ν_1 varies during the burst, and estimate $\mu(\nu_1)$ and $D(\nu_1)$ by averaging over the many bursts. This can be done for a variety of different values of ν_1, and data like that shown in Fig. 12 can be found.

The problem of initialising the full system at a particular value of ν_1 ("lifting") is nontrivial, and a variety of methods have been used [31,30,16,50]. Here we take a simple approach. We wish to estimate $\mu(\nu_1)$ and $D(\nu_1)$ at a set of evenly-spaced values of ν_1, as in Fig. 12. If ν_1^* is one of these values, we find all snapshots in our original long simulation for which ν_1 is between $\nu_1^* - \delta\nu/2$ and $\nu_1^* + \delta\nu/2$, where $\delta\nu$ is the spacing between ν_1 values in Fig. 12. We then repeatedly randomly choose one from this collection of snapshots, initialise the full system at this state (recall that the snapshots are of all variables) and run for a short time (0.025 sec.). For each of these bursts we use the Nyström formula (with our 3000 original points) to calculate ν_1 at the end of the burst, and use the difference between this and the value of ν_1 for the initial condition to estimate $\mu(\nu_1)$ and $D(\nu_1)$. Averaging over these bursts gives an estimate of $\mu(\nu_1^*)$ and $D(\nu_1^*)$. The fact that snapshots may be used multiple times as initial conditions, or that they have already been used in our calculation of the $30,000$ values of ν_1, is not relevant, as the dynamics of ν_1 will be different in each of these new bursts, due to the stochastic nature of the firing times of all input neurons.

For each value of ν_1 we have run 5000 such bursts. The means of the 5000 values of $\mu(\nu_1)$ and $D(\nu_1)$ are shown in Fig. 15. In comparison with Fig. 12 we see that the error bars are much more uniform over ν_1, and smaller in general. We can repeat the calculations above to find the potential and switching times from these new estimates, and they give very similar results (not shown).

We can also vary parameters and see how the networks' dynamics change. The results of varying σ are shown in Figs. 16 and 17. Here we ran simulations at 20 equally-spaced values of σ, each of duration 750 seconds, and for each simulation, performed the analysis described above, i.e. used 3000 equally spaced snapshots to generate values of ν_1, then use the Nyström formula on 30,000 data points. We processed these 30,000 values to get estimates of $\mu(\nu_1)$ on a grid of ν_1 values. At each value of σ we fit a cubic

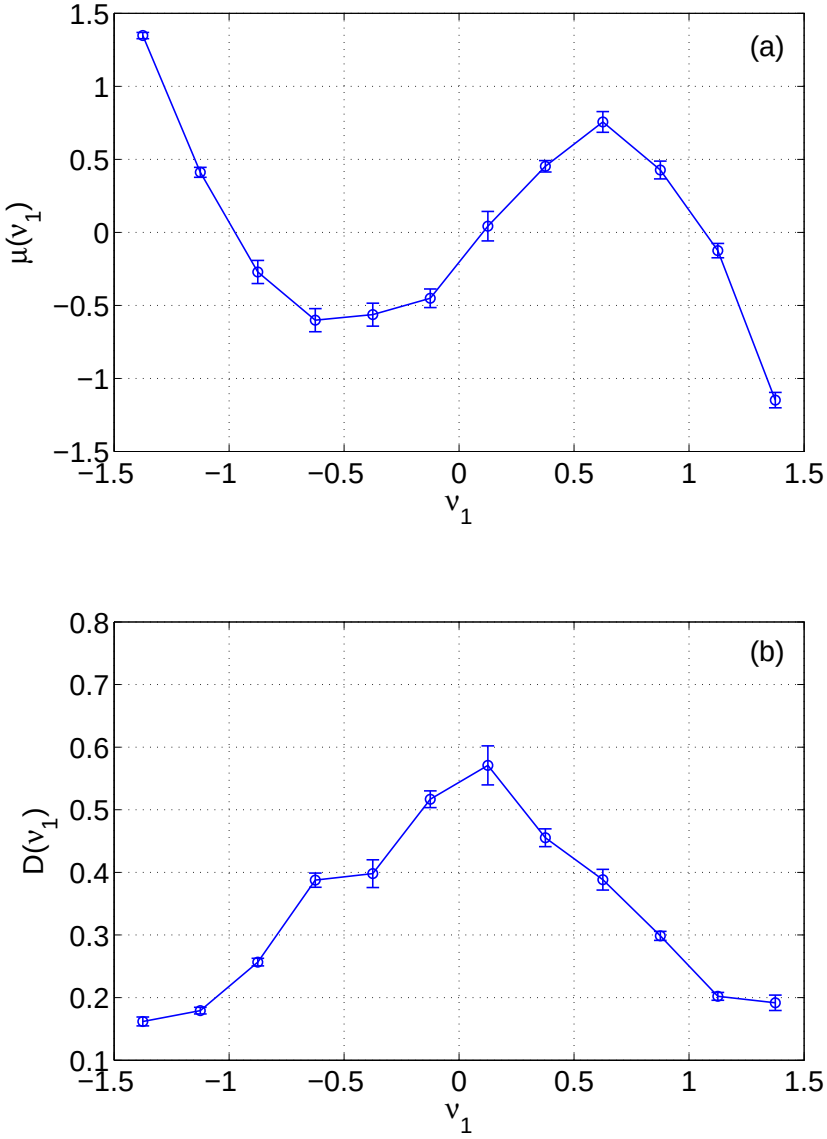

Fig. 15 Estimates of the functions $\mu(\nu_1)$ (a) and $D(\nu_1)$ (b) resulting from initialising the network at specific values of ν_1 (see text).

$\mu(\nu_1) = b_3\nu_1^3 + b_2\nu_1^2 + b_1\nu_1 + b_0$ to these data points. The coefficients $b_3, \ldots b_0$ are shown in Fig. 16, along with a straight line fit through them, defining the linear functions $b_3(\sigma), \ldots b_0(\sigma)$. From the symmetry of the problem we expect b_2 and b_0 to be approximately zero, and for b_3 and b_1 to have some dependence on σ, and that is what is seen. Figure 17 shows the real roots of $b_3(\sigma)\nu_1^3 + b_2(\sigma)\nu_1^2 + b_1(\sigma)\nu_1 + b_0(\sigma)$ at the 20 different values of σ used, and we see that the system undergoes a perturbed pitchfork bifurcation as σ is increased. Note that at each value of σ we used only the data from the corresponding simulation for the data-mining.

We have demonstrated that, for the parameter values considered, the network can be well-described by a stochastic differential equation for a single variable. As in Sec. 3, the correctness of this assumption may change as parameters are varied, and should be checked. The stochastic differential equation derived is much quicker to simulate than the full network and could thus form a "module" for simulating the

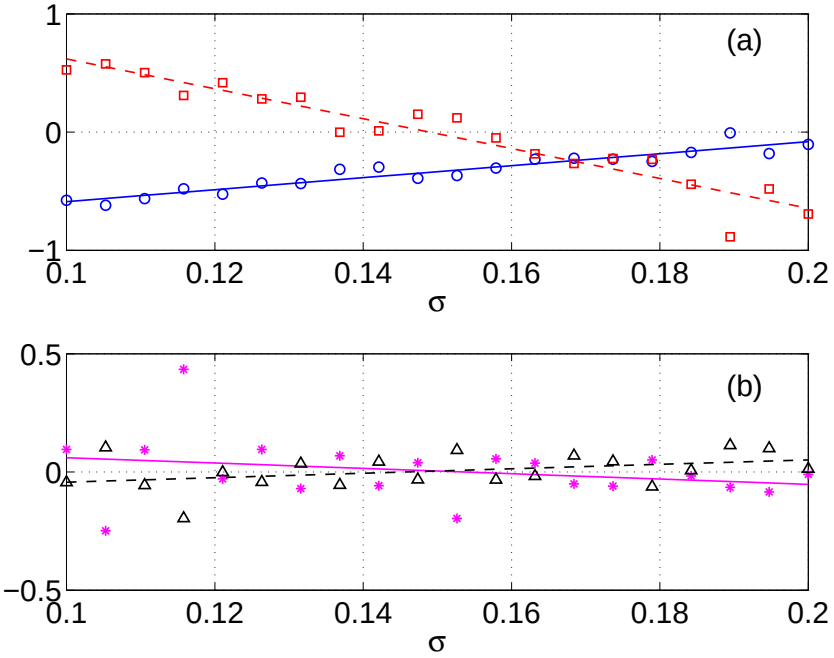

Fig. 16 Coefficients of the cubic function $\mu(\nu_1) = b_3\nu_1^3 + b_2\nu_1^2 + b_1\nu_1 + b_0$, for the values of σ shown. (a): b_3 (blue circles and solid line) and b_1 (red squares and dashed line). (b): b_2 (magenta stars and solid line) and b_0 (black diamonds and dashed line).

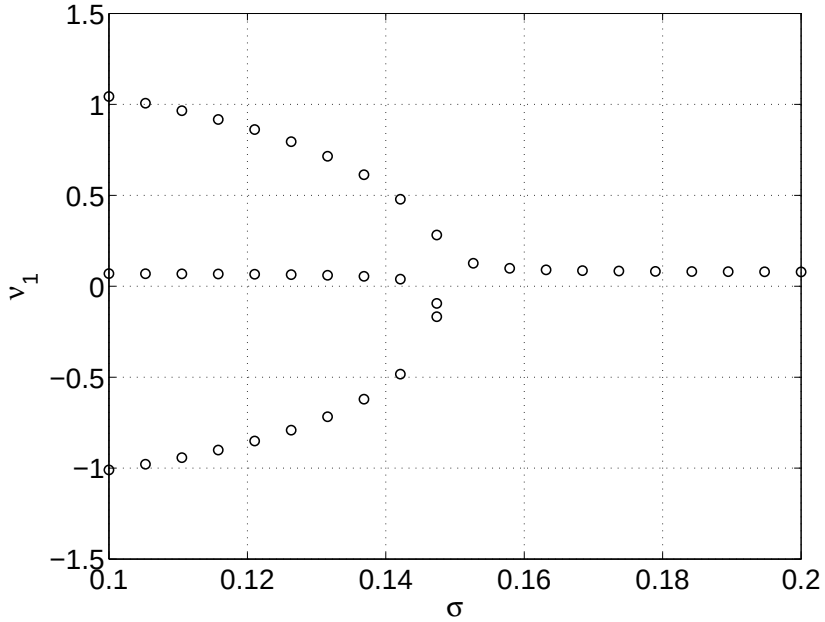

Fig. 17 Real roots of $b_3(\sigma)\nu_1^3 + b_2(\sigma)\nu_1^2 + b_1(\sigma)\nu_1 + b_0(\sigma)$ as a function of σ, where the dependence of the coefficients is linear, as shown in Fig. 16. This shows a perturbed pitchfork bifurcation; enforcing $b_2 = b_0 = 0$ would result in a perfect pitchfork bifurcation.

dynamics of this small network within a larger simulation, to be "called" when necessary. By following roots of $\mu(\nu_1)$ as σ is varied (Fig. 17) we have uncovered a pitchfork bifurcation which lies behind the cessation of switching that occurs as σ increases.

5 Summary

We have studied a simple well-known network of neurons whose connectivities undergo STDP, in two parameter regimes. In the first we observed that the distribution of connection strengths evolved on a much slower time-scale than any of the explicit time-scales in the model. This separation of time-scales suggested that projective integration could be used to speed up simulations of the network. We showed this to be accurate, expanding the inverse of the cumulative distribution function of the connection strengths in (a small number of) orthogonal polynomials. The coefficients of these polynomials provide a low-dimensional description of the system, and their evolution can be calculated using conventional algorithms for numerical integration. In principle we could also follow (stable and unstable) fixed points of the equations describing the evolution of these coefficients as parameters are varied, detecting bifurcations. In the second regime we uncovered, using data-mining, a good observable and demonstrated its relation to a physically meaningful variable (the difference between the means of the two populations). We showed that the network's dynamics could be well-described by a single stochastic differential equation for this single variable and demonstrated the estimation of the functions in this equation. This stochastic differential equation is much faster to simulate than the full network whose behaviour it replicates. We also followed fixed points of the "deterministic" part of this stochastic differential equation, showing that a pitchfork bifurcation occurs as a parameter is varied.

The model we studied was simple in the sense that it was only the strengths of connections *from* the excitatory inputs that were modified by STDP. The dynamics of the inputs (their firing times) were unchanged by this evolution, as there were no connections from the integrate-and-fire neuron back to them. This will not generally be the case, and more complex dynamics are expected from fully connected networks, as has been observed by several authors [12,40]. In such networks, data-mining may be essential to determine a low-dimensional description of the network dynamics (if this exists), and current data-mining techniques may well need to be extended to deal with data measured from dynamic networks. We note that STDP is but one form of synaptic plasticity (and the version presented here is not the only formulation of STDP [1,26,2]) — another example is homeostatic plasticity, which acts to promote stable network behaviour [51,23]. If this form of plasticity is slow relative to the dynamics of individual neurons, the techniques presented here could be used to study such slow dynamics.

We studied a neuronal network in which the dynamics of synaptic strengths played a crucial role in determining the network's behaviour. However, the equation-free approach is quite general and flexible, and we believe that it will be useful in the study of other complex networks of neurons whose macroscopic behaviour appears to be quite simple [34,14,46].

Acknowledgements

The work of CRL was supported by the Marsden Fund Council from Government funding, administered by the Royal Society of New Zealand. The work of IGK was supported by the US National Science Foundation.

References

1. Appleby, P.A., Elliott, T.: Stable competitive dynamics emerge from multispike interactions in a stochastic model of spike-timing-dependent plasticity. Neural computation **18**(10), 2414–2464 (2006)
2. Appleby, P.A., Elliott, T.: Multispike interactions in a stochastic model of spike-timing-dependent plasticity. Neural computation **19**(5), 1362–1399 (2007)
3. Avitabile, D., Hoyle, R., Samaey, G.: Noise reduction in coarse bifurcation analysis of stochastic agent-based models: an example of consumer lock-in. SIAM Journal on Applied Dynamical Systems **13**(4), 1583–1619 (2014)
4. Bell, C.C., Han, V.Z., Sugawara, Y., Grant, K.: Synaptic plasticity in a cerebellum-like structure depends on temporal order. Nature **387**(6630), 278–281 (1997)
5. Bi, G.q., Poo, M.m.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. The Journal of neuroscience **18**(24), 10,464–10,472 (1998)
6. Bliss, T.V., Collingridge, G.L.: A synaptic model of memory: long-term potentiation in the hippocampus. Nature **361**(6407), 31–39 (1993)
7. Brette, R.: Exact simulation of integrate-and-fire models with synaptic conductances. Neural Computation **18**(8), 2004–2027 (2006)
8. Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J.M., Diesmann, M., Morrison, A., Goodman, P.H., Harris Jr, F.C., et al.: Simulation of networks of spiking neurons: a review of tools and strategies. Journal of computational neuroscience **23**(3), 349–398 (2007)
9. Burkitt, A.N., Gilson, M., van Hemmen, J.L.: Spike-timing-dependent plasticity for neurons with recurrent connections. Biological cybernetics **96**(5), 533–546 (2007)
10. Burkitt, A.N., Meffin, H., Grayden, D.B.: Spike-timing-dependent plasticity: the relationship to rate-based learning for models with weight dynamics determined by a stable fixed point. Neural Computation **16**(5), 885–940 (2004)
11. Caporale, N., Dan, Y.: Spike timing-dependent plasticity: A hebbian learning rule. Annual Review of Neuroscience **31**(1), 25–46 (2008). PMID: 18275283
12. Chen, C.C., Jasnow, D.: Event-driven simulations of a plastic, spiking neural network. Physical Review E **84**(3), 031,908 (2011)
13. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences of the United States of America **102**(21), 7426–7431 (2005)
14. DeVille, R.L., Peskin, C.S.: Synchrony and asynchrony in a fully stochastic neural network. Bulletin of mathematical biology **70**(6), 1608–1633 (2008)
15. Erban, R., Frewen, T.A., Wang, X., Elston, T.C., Coifman, R., Nadler, B., Kevrekidis, I.G.: Variable-free exploration of stochastic models: A gene regulatory network example. The Journal of Chemical Physics **126**(15), 155103 (2007)
16. Erban, R., Kevrekidis, I., Adalsteinsson, D., Elston, T.: Gene regulatory networks: A coarse-grained, equation-free approach to multiscale computation. The Journal of chemical physics **124**, 084,106 (2006)
17. Ermentrout, G.B., Terman, D.H.: Mathematical foundations of neuroscience, vol. 64. Springer (2010)
18. Ferguson, A.L., Panagiotopoulos, A.Z., Debenedetti, P.G., Kevrekidis, I.G.: Systematic determination of order parameters for chain dynamics using diffusion maps. Proceedings of the National Academy of Sciences **107**(31), 13,597–13,602 (2010)
19. Gear, C.: Projective integration methods for distributions. NEC Technical Report **TR 2001-130** (2001)
20. Gear, C.W., Kevrekidis, I.G., Theodoropoulos, C.: 'Coarse' integration/bifurcation analysis via microscopic simulators: micro-galerkin methods. Computers & chemical engineering **26**(7), 941–963 (2002)
21. Gerstner, W., Kempter, R., van Hemmen, J.L., Wagner, H., et al.: A neuronal learning rule for sub-millisecond temporal coding. Nature **383**(6595), 76–78 (1996)

22. Gilson, M., Burkitt, A.N., Grayden, D.B., Thomas, D.A., van Hemmen, J.L.: Emergence of network structure due to spike-timing-dependent plasticity in recurrent neuronal networks. i. input selectivity–strengthening correlated input pathways. Biological cybernetics **101**(2), 81–102 (2009)
23. Golowasch, J., Casey, M., Abbott, L., Marder, E.: Network stability from activity-dependent regulation of neuronal conductances. Neural Computation **11**(5), 1079–1096 (1999)
24. Gradišek, J., Siegert, S., Friedrich, R., Grabec, I.: Analysis of time series from stochastic processes. Physical Review E **62**(3), 3146–3155 (2000)
25. Gütig, R., Aharonov, R., Rotter, S., Sompolinsky, H.: Learning input correlations through nonlinear temporally asymmetric hebbian plasticity. The Journal of neuroscience **23**(9), 3697–3714 (2003)
26. Izhikevich, E.M., Desai, N.S.: Relating STDP to BCM. Neural computation **15**(7), 1511–1523 (2003)
27. Keener, J., Sneyd, J.: Mathematical Physiology, vol. 8. Springer (1998)
28. Kempter, R., Gerstner, W., van Hemmen, J.L.: Hebbian learning and spiking neurons. Phys. Rev. E **59**, 4498–4514 (1999). DOI 10.1103/PhysRevE.59.4498. URL http://link.aps.org/doi/10.1103/PhysRevE.59.4498
29. Kevrekidis, I.G., Gear, C.W., Hyman, J.M., Kevrekidid, P.G., Runborg, O., Theodoropoulos, C., et al.: Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. Communications in Mathematical Sciences **1**(4), 715–762 (2003)
30. Laing, C., Frewen, T., Kevrekidis, I.: Coarse-grained dynamics of an activity bump in a neural field model. Nonlinearity **20**, 2127 (2007)
31. Laing, C., Frewen, T., Kevrekidis, I.: Reduced models for binocular rivalry. Journal of computational neuroscience **28**(3), 459–476 (2010)
32. Laing, C.R., Kevrekidis, I.G.: Periodically-forced finite networks of heterogeneous globally-coupled oscillators: A low-dimensional approach. Physica D: Nonlinear Phenomena **237**(2), 207–215 (2008)
33. Lee, S.L., Gear, C.W.: Second-order accurate projective integrators for multiscale problems. Journal of Computational and Applied Mathematics **201**(1), 258 – 274 (2007). DOI http://dx.doi.org/10.1016/j.cam.2006.02.018
34. Lim, S., Rinzel, J.: Noise-induced transitions in slow wave neuronal dynamics. Journal of Computational Neuroscience **28**(1), 1–17 (2010). DOI 10.1007/s10827-009-0178-y. URL http://dx.doi.org/10.1007/s10827-009-0178-y
35. Lubenov, E.V., Siapas, A.G.: Decoupling through synchrony in neuronal circuits with propagation delays. Neuron **58**(1), 118–131 (2008)
36. Markram, H., Lübke, J., Frotscher, M., Sakmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science **275**(5297), 213–215 (1997)
37. Marschler, C., Faust-Ellsässer, C., Starke, J., van Hemmen, J.L.: Bifurcation of learning and structure formation in neuronal maps. EPL (Europhysics Letters) **108**(4), 48,005 (2014)
38. Marschler, C., Sieber, J., Berkemer, R., Kawamoto, A., Starke, J.: Implicit methods for equation-free analysis: convergence results and analysis of emergent waves in microscopic traffic models. SIAM Journal on Applied Dynamical Systems **13**(3), 1202–1238 (2014)
39. Meffin, H., Besson, J., Burkitt, A., Grayden, D.: Learning the structure of correlated synaptic subgroups using stable and competitive spike-timing-dependent plasticity. Physical Review E **73**(4), 041,911 (2006)
40. Mikkelsen, K., Imparato, A., Torcini, A.: Emergence of slow collective oscillations in neural networks with spike-timing dependent plasticity. Physical Review Letters **110**(20), 208,101 (2013)
41. Morrison, A., Aertsen, A., Diesmann, M.: Spike-timing-dependent plasticity in balanced random networks. Neural computation **19**(6), 1437–1467 (2007)
42. Ragwitz, M., Kantz, H.: Indispensable finite time corrections for fokker-planck equations from time series data. Phys. Rev. Lett. **87**, 254,501 (2001)
43. Roberts, P.D., Bell, C.C.: Spike timing dependent synaptic plasticity in biological systems. Biological Cybernetics **87**(5-6), 392–403 (2002)
44. Rubin, J., Lee, D., Sompolinsky, H.: Equilibrium properties of temporally asymmetric hebbian plasticity. Physical Review Letters **86**(2), 364–367 (2001)
45. Setayeshgar, S., Gear, C., Othmer, H., Kevrekidis, I.: Application of coarse integration to bacterial chemotaxis. Multiscale Modeling & Simulation **4**(1), 307–327 (2005)
46. Smith, J.C., Abdala, A., Koizumi, H., Rybak, I.A., Paton, J.F.: Spatial and functional architecture of the mammalian brain stem respiratory network: a hierarchy of three oscillatory mechanisms. Journal of Neurophysiology **98**(6), 3370–3387 (2007)

47. Sonday, B.E., Haataja, M., Kevrekidis, I.G.: Coarse-graining the dynamics of a driven interface in the presence of mobile impurities: Effective description via diffusion maps. Phys. Rev. E **80**, 031,102 (2009)
48. Song, S., Abbott, L.: Cortical development and remapping through spike timing-dependent plasticity. Neuron **32**(2), 339–350 (2001)
49. Song, S., Miller, K., Abbott, L.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neuroscience **3**, 919–926 (2000)
50. Sriraman, S., Kevrekidis, I., Hummer, G.: Coarse nonlinear dynamics and metastability of filling-emptying transitions: water in carbon nanotubes. Physical review letters **95**(13), 130,603 (2005)
51. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience **5**(2), 97–107 (2004)
52. Van Rossum, M., Bi, G., Turrigiano, G.: Stable hebbian learning from spike timing-dependent plasticity. The Journal of Neuroscience **20**(23), 8812–8821 (2000)
53. Zou, Y., Fonoberov, V., Fonoberova, M., Mezic, I., Kevrekidis, I.: Model reduction for agent-based social simulation: Coarse-graining a civil violence model. Physical Review E **85**(6), 066,106 (2012)