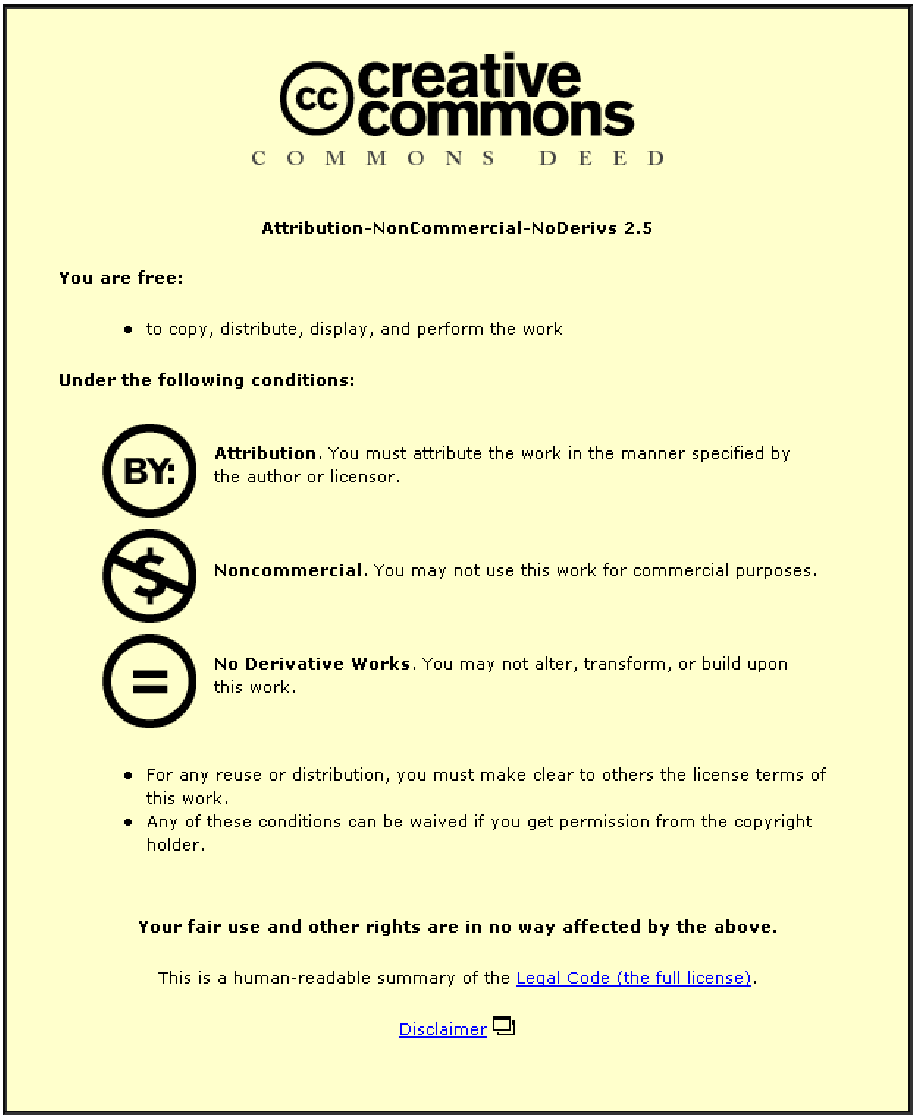


Nonlinear Dynamics of Pattern Recognition and Optimization

Christopher J. Marsden

A Doctoral Thesis

Submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University

October 2012

Abstract

We associate learning in living systems with the shaping of the velocity vector field of a dynamical system in response to external, generally random, stimuli. We consider various approaches to implement a system that is able to adapt the whole vector field, rather than just parts of it – a drawback of the most common current learning systems: artificial neural networks.

This leads us to propose the mathematical concept of self-shaping dynamical systems. To begin, there is an empty phase space with no attractors, and thus a zero velocity vector field. Upon receiving the random stimulus, the vector field deforms and eventually becomes smooth and deterministic, despite the random nature of the applied force, while the phase space develops various geometrical objects. We consider the simplest of these – gradient self-shaping systems, whose vector field is the gradient of some energy function, which under certain conditions develops into the multi-dimensional probability density distribution of the input.

We explain how self-shaping systems are relevant to artificial neural networks. Firstly, we show that they can potentially perform pattern recognition tasks typically implemented by Hopfield neural networks, but without any supervision and on-line, and without developing spurious minima in the phase space. Secondly, they can reconstruct the probability density distribution of input signals, like probabilistic neural networks, but without the need for new training patterns to have to enter the network as new hardware units. We therefore regard self-shaping systems as a generalisation of the neural network concept, achieved by abandoning the “rigid units - flexible couplings” paradigm and making the vector field fully flexible and amenable to external force. It is not clear how such systems could be implemented in hardware, and so this new concept presents an engineering challenge. It could also become an alternative paradigm for the modelling of both living and learning systems.

Mathematically it is interesting to find how a self shaping system could develop non-trivial objects in the phase space such as periodic orbits or chaotic attractors. We investigate how a delayed vector field could form such objects. We show that this method produces chaos in a class systems which have very simple dynamics in the non-delayed case. We also demonstrate the coexistence of bounded and unbounded solutions dependent on the initial conditions and the value of the delay. Finally, we speculate about how such a method could be used in global optimization.

Keywords: Nonlinear dynamics, Pattern Recognition, Optimization, Learning, Dynamical Systems, Delay.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr Natalia Janson for providing me with the opportunity to undertake this PhD and for her continuous support of my study and research.

Contents

List of Figures

1 Introduction

The human brain is superior to a computer at many tasks. Take, for example, the processing of visual information: a child is much faster and more accurate at recognizing objects than even the most advanced artificial intelligence (AI) system. The brain also has many features that are difficult to recreate in artificial systems [27]: it is robust and copes well with faults, for example, nerve cells die every day with minimal effect on its performance; it easily adjusts to new environments through learning; it is highly parallel, with many tasks and calculations being carried out simultaneously; and it can deal with noisy and inconsistent data.

Artificially intelligent devices currently are attributed a broad range of tasks including image and speech recognition, machine vision, language processing, and medical diagnostics [4]. However, these machines are really only able to perform two types of task: classification (identifying the category to which a pattern belongs) and optimization, which include decision-making. Learning has been understood merely as acquiring the ability to perform these tasks.

To achieve their desired task, traditional artificial intelligence devices perform a sequence of pre-defined commands, i.e. they are algorithmic. Even the latest generation of artificial intelligence devices, based on artificial neural networks, utilise algorithms at least at the stage of learning [27]. However, it appears that a biological brain does not work in this way – it does not seem to learn by an algorithm, nor does it naturally execute a series of commands, although it can learn to do this.

Specifically, learning is defined to be the process of gaining knowledge [56]. The basic mechanics of how the brain achieves this were demonstrated in a paper by Bliss and Tømo [6]. They showed that there is a physical change in the internal connections of the brain during the learning process. Closely linked to learning, is the process of recognition. Recognition is the process of identifying a particular feature within the given data. Within the field of artificial intelligence, these two tasks can be implemented by a variety of methods, for example, artificial neural networks, based loosely on biological neural networks.

Artificial neural networks, however, are not without their limitations. Almost all rely on some kind of supervision in their learning; many operate in a binary fashion; and those that do not, encounter technical issues like the formation of "spurious minima" – attractors not belonging to any class (Section 2.4).

This work investigates the modelling of unsupervised learning. The signal's input values arrive sequentially, with the system learning after each new value. It should enable the system to classify features in the signal without any pre-training. For example, if a video signal is applied, the system should identify classes in a particular property, such as a colour, without any a priori information about the existing classes (e.g., if we analyse colours, we do not know in advance that they may be broadly classified as red, blue, green, etc.). One can regard an artificial neural network as a highly nonlinear dissipative multi-dimensional dynamical system, and consequently use the methods of dynamical systems theory for its investigation. A series of papers adopting this approach belong to Michail Zak [89, 91, 90, 94]. Here we abandon the idea of artificial neural networks and propose a new kind of dynamical system, which perform the same tasks as neural networks, but without their limitations.

Section 2 provides the background information necessary to understand the work in this thesis: we give a history of the work in this field, provide some more technical details on artificial neural networks, review the relevant parts of Zak's work, and finally discuss the motivation behind the new methods introduced in this thesis.

In Section 3 we discuss the applicability and limitations of Zak's methods before proposing a modification to the dynamical systems he uses. We test the workability of these modified dynamical systems using various random data as the input.

In Section 4 we introduce a new kind of dynamical systems that seem to have all the desirable features of artificial neural networks, but without their technical limitations. We call these self-shaping dynamical systems. Using both numerically generated test examples and real musical data as inputs, we demonstrate as the proof of principle how the dynamical system of the new type, self-organises its velocity vector field while learning the typical patterns all by itself, i.e. without any human supervision. We then discuss the advantages of this new method as compared to the standard tools used for these tasks – artificial neural networks. These systems could offer a new paradigm of learning and adapting machines of a new generation.

In Section 5 we investigate the role that time delay plays in the simplest type of self-shaping systems – of gradient type. We investigate the impact of delaying the whole vector field in such systems. We show that this method produces chaos in a class of systems with very simple dynamics in the non-delayed case. We also demonstrate the coexistence of bounded and unbounded solutions dependent on the value of the delay. Finally, we discuss how gradient systems with delay might be relevant to the problem of global optimization.

2 Background

This general introductory section comprises four subsections. The first details the evolution of the knowledge and understanding of learning in the brain. The second explains the basic concepts in biological and artificial neural networks. The third explains the work of Zak using dynamical systems to model learning. The final part details the motivation behind the novel methods detailed in the thesis.

2.1 Understanding of the brain, some historical remarks

History is filled with speculation, theory and experimentation on the role of the brain. As far back as 300-400 years BCE, Ancient Greek philosopher and father of logic Aristotle, believed that the role of the brain was to cool the blood, with the heart being responsible for rational thought [52]. Meanwhile at a similar time, Hippocrates, father of medicine, believed the brain was responsible for thought and feeling, linking emotion and illness [69]. It was not until 150-200 CE that Roman physician Galen demonstrated experimentally that all muscles in the body are connected to the brain through a network of nerves [60]. There was then little conceptual progress for hundreds of years.

In the 17th century, French philosopher and mathematician, Descartes proposed that non-human animals were mindless automata, and that the majority of the behaviours of humans could be explained by machine [12]. This theory proved divisive among scientists and philosophers alike, and even today the concept of the brain as a machine remains controversial.

From the 1700s onward, the majority of our knowledge of the brain has come from the development of a new technique of investigation. The most important of these came courtesy of Cajal, with the theory of the individuality of the nerve cell in 1891 based on the experiments of Golgi [37]. Soon after, Waldayer gave this individual cell the name "neuron". This gave birth to the neuron doctrine – the fundamental theory of neuroscience, providing an overall structure to the previous smaller scale findings. It says that the neuron is the structural and functional unit of the nervous system. It is an individual cell comprising three parts: dendrites, the soma (cell body), and the axon – the end of which has several branches which make close contact to dendrites of other neurons [76].

The past century has brought numerous advances in our knowledge and understanding of the electrical and chemical behaviour in the brain. With regard to learning, the 20th century brought us the theory of synaptic plasticity – the ability of the connection between the neurons (synapse) to change in response to inputs [75]. This was started in 1949 by Hebb who famously hypothesised that "neurons that fire together, wire together" – explaining the process of associativity in learning. In slightly more technical terms: if two neurons are active simultaneously then the strength of their connection should be increased [25]. In 1973, Lømo and Bliss demonstrated the basic mechanics of the process in a paper on long term potentiation, showing clear evidence of activity-induced synaptic changes that lasted for at least several days [6].

In the 20th century, away from the biological studies, advancements in electronics brought hope to the possibility of recreating the key processes in the brain by a machine. The two most popular concepts were computers and artificial neural networks.

In his famous 1950 paper, Alan Turing introduced the idea of a machine that changes the rules of its operation in the process of learning [84]. He was referring to a digital computer - a machine with discrete states. Since then, computers have evolved to much more powerful processing systems, capable of performing computations far faster than a human could ever hope to achieve. The concepts, however, are fundamentally still the same – requiring algorithms, a set of predefined operations, to process an input. This restricts them to operating on only problems that we know how to solve.

The basis for artificial neural networks is the McCulloch-Pitts model of a neuron – simple neurons which were considered to be binary devices with fixed thresholds [54]. McCulloch and Pitts proposed attributing a weight to input signals entering the neuron through a synapse. In mathematical terms, such weights correspond to the strength of coupling between the neurons. The weight of an input is a number, which when multiplied with the input, gives the weighted input. These weighted inputs are then added together, and if they exceed a preset threshold value, the neuron fires. These weights give the network the ability to adapt to a particular situation by changing its weights and/or the threshold. The results of their model were simple logic functions such as OR and AND [27].

In 1958, Rosenblatt stirred considerable interest and activity in the field, when he designed and developed the perceptron [67]. The perceptron had three layers: an input, a middle layer known as the association layer, and an output. This system could learn to connect (associate) a given input to a random output unit. The perceptron caused great excitement at the time, as it was thought to be the path to producing programs that could think. However, in 1969, Minsky and Papert showed that the perceptron had severe limitations which meant that it could not learn certain types of functions (i.e. those which are not linearly separable – cannot be separated by a single line, e.g. XOR) [55]. Due to Minsky and Papert's proof, research into neural networks went into decline throughout the 1970s.

Despite this, some researchers continued study in this area. In 1974, Werbos developed and used the back-propagation learning method [87]. It was several years, however, before this approach was popularised by Parker and LeCun in 1985 and 1986 respectively, who independently discovered the back-propagation learning algorithm for multi-layer networks, which could solve problems that were not linearly separable [61, 44]. Back-propagation nets are one of the most well known and widely applied of neural networks today. In essence, the back-propagation net is a perceptron with multiple layers, a different threshold function in the artificial neuron, and a more robust learning rule.

In 1982, Hopfield introduced a major breakthrough to the field – an energy landscape [31]. The energy function contains the stable critical points corresponding to the most typical patterns that represent a certain class, and the input pattern will fall into the energy well corresponding to one of the classes. As time goes by, the energy can only decrease, and the neural network will gradually evolve towards the class centre.

In 1984, Hopfield made another breakthrough – the introduction of a continuous state and time neuron model [32]. Up until this point, artificial neural networks consisted of discrete state neurons, and the state of the network at any time was essentially a vector whose coordinates were 0s and 1s. This change brought artificial neural networks closer to biological ones.

Artificial neural networks continue to be studied now, and with the processing power of computers ever growing, they become faster and more efficient. The techniques and methods in artificial neural networks are explained in Section 2.2. An alternative route has also been pursued since the 1950s – modelling neural networks as dynamical systems. A dynamical system is essentially a system in which one can define the state at any given time moment, and the rule according to which the state evolves in time (the evolution operator). Often, dynamical systems are modelled by differential equations.

The first neural model in the form of a dynamical system appeared in 1952, with the Hodgkin-Huxley model of neuron firing and propagation [30]. This was a biologically plausible model, however, its mathematical analysis is difficult and time consuming. The FitzHugh-Nagumo system, 1961, provided a mathematically tractable simplification of the Hodgkin-Huxley model, enabling analysis of the dynamics of the system [15, 58]. This dynamical systems approach makes use of advances in complexity theory, and is particularly applicable to modelling of the brain, when one considers the different interactions and scales involved.

It is worth mentioning separately the work of Zak, who considered neural networks from the viewpoint of the dynamical systems approach [94]. He proposed a new type of dynamical system for modelling biological behaviour, motivated by the desire to remove the rigidity of artificial neural networks as opposed to their biological counterparts. This work made use of his previous work in dynamical systems theory on terminal attractors and repellors [89].

In 1990, in a paper with Toomarian, Zak considered the artificial neural network as an adaptive nonlinear dissipative dynamical system [94]. Representation in this way means that, incorporated into the neural network, there will be attractors in the phase space of the dynamical system corresponding to clusters of patterns. Surrounding these attractors will be their basins of attraction, which organise new patterns introduced to the neural network, sending them to their local attractor. Thus the network is able to perform pattern recognition using the attractors formed during learning.

In 2003, Zak proposed another approach utilising dynamical systems [91]. This time he coupled the input patterns to the vector field of the dynamical system. This uses the same concepts from dynamical systems as his previous method, however, the method for the construction of the vector field differs. Here, the approach uses the evolution of the probability density of the input signal to identify features, which is then coupled with its input in a dynamical system to classify the input as it arrives. Both these methods will be explained further in Section 2.3.

2.2 Artificial neural networks

An artificial neural network is a model designed to process information, inspired by the structure and functions of biological neural networks. They comprise large numbers of interconnected neurons, which combine to solve specific problems. Like their biological equivalent, artificial neural networks learn from examples, altering the synaptic connections between the neurons in the process.

Neural networks provide a different methodology to information processing and problem solving than their main alternative – computers. Computers function in an entirely algorithmic fashion – following a set of pre-defined instructions to solve a particular task. This however, limits their applicability to problems that it is known how to solve. Artificial neural networks can too be considered algorithmic, at least in the learning stage. Learning takes place through the adjustment of weights. The most common method for this is Hebbian learning – the weight between two neurons increases if the two neurons are activated simultaneously, and reduces if they are activated separately. The big advantage of neural networks is that after learning, in the operating stage, they are able to classify patterns which they have never seen before, based on what they have already learnt.

In some respect, artificial neural networks process the information in a similar way to the brain – connected elements work together, in parallel, to solve the problem. They are not programmed with a specific task in mind, but they are trained, using carefully selected examples, ensuring the desired functionality will be attained. Each approach is better suited to certain tasks, and in addition, a large number of tasks require a combination of the two methods. For example, computers can be used to supervise the artificial neural network to maximise the efficiency of its operation. Here we provide an introduction to the concepts involved.

The method and types of learning will be discussed in more detail at the end of this subsection. Initially, we will assume that the weights have been chosen in an optimal way.

A Biological Neuron

A neuron is an excitable cell. Its activity is demonstrated by local variations in its membrane potential – called “action potentials” or “spikes”. These occur due to the exchange of ions moving through the membrane from areas of highest concentration to areas of lowest concentration. This requires the opening and closing of gates in the ionic channels, which depends on the local membrane potential induced by local excitations.

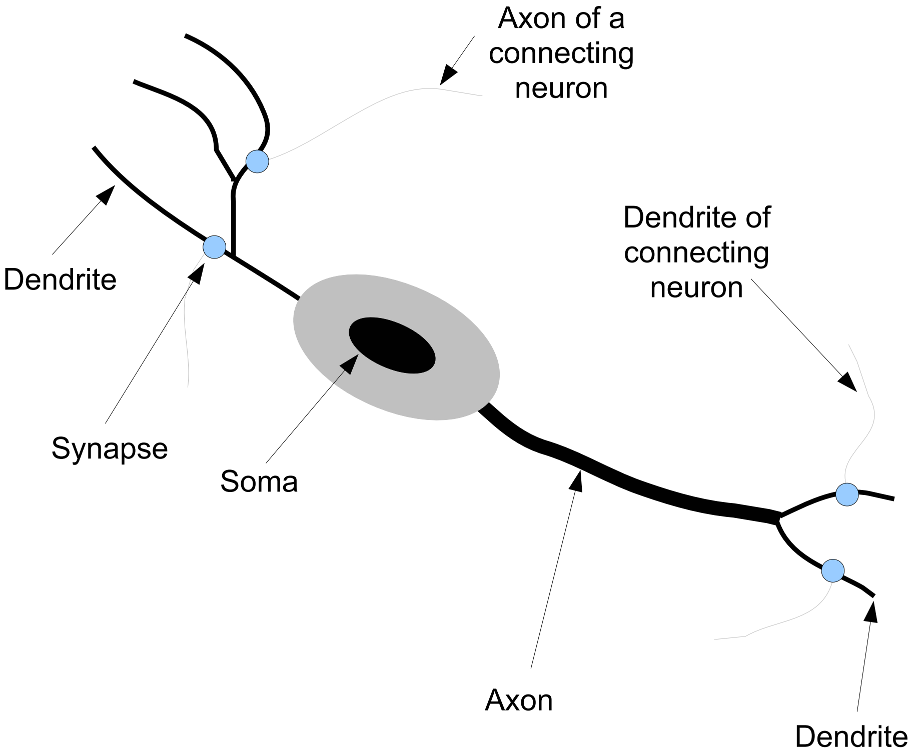


Figure 1: A biological neuron.

There are approximately 10^{11} neurons in a human brain connected in a tree-like network, with the axon of a typical neuron making a few thousand synapses with other neurons [27]. In a typical neuron, Fig. 1, the *soma* (cell body) collects electrical signals from a series of fine structures called *dendrites*, which are connected to other neurons. The neuron sends out spikes of electrical activity through a long, thin fibre called an *axon*, which splits into lots of smaller branches. At the end of each branch, a *synapse* assesses the electrical activity to a connected neuron. When a neuron receives excitatory input sufficiently greater than inhibitory input (i.e. it reaches a threshold), the neuron will *fire* – sending a spike of electrical activity down its axon. In the learning process, the synaptic connections are adjusted, altering the influence on connecting neurons [75].

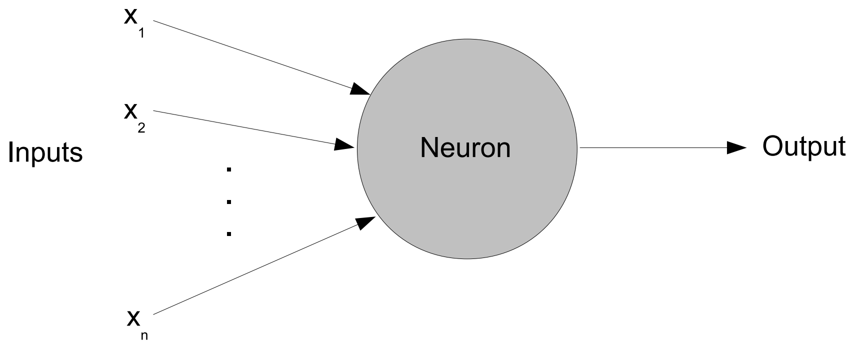


Figure 2: A simple artificial neuron.

An Artificial Neuron

An artificial neuron (Fig. 2) is a very simplified reconstruction of the biological neuron described previously. It contains many inputs (corresponding to the dendrites) and one output (corresponding to the axon). The artificial neuron has two modes of operation: the training mode, and the operating mode. In the training stage, the neuron learns when to fire for particular input patterns. In the operating stage, if the input corresponds to one of the taught patterns, the neuron outputs the associated response. If the input does not correspond with one of the taught patterns, a *firing rule* is used to determine the output.

The firing rule

The firing rule determines whether a neuron should fire for any input pattern. A simple rule is the Hamming distance technique [39]:

> Let the set of training patterns teaching the neuron to fire be called the *1-taught set*, and the set of training patterns teaching to neuron not to fire be called the *0-taught set*. For an input not in either training set, the neuron will fire if the input has more elements in common with nearest pattern in the 1-taught set, than the nearest pattern in the 0-taught set. If they are equivalent, the pattern is remains undefined.

A standard example goes as follows:

Consider a neuron with 3 inputs (X_1, X_2, X_3). The neuron is trained to output 1 if the input is $(1, 1, 1)$ or $(1, 0, 1)$, and to output 0 if the input is $(0, 0, 0)$ or $(0, 0, 1)$. So before applying the firing rule, we can express the inputs and outputs in a *truth table*.

X_1	0	0	0	0	1	1	1	1
X_2	0	0	1	1	0	0	1	1
X_3	0	1	0	1	0	1	0	1
OUT	0	0	0/1	0/1	0/1	1	0/1	1

Here, only half of the patterns are defined. So we now use the firing rule. Take the last undefined pattern in the truth table $(1, 1, 0)$. In the 0-taught set, it differs from $(0, 0, 0)$ by two elements, and $(0, 0, 1)$ by three elements. In the 1-taught set, it differs from $(1, 1, 1)$ by one element, and $(1, 0, 1)$ by two elements. Therefore, its nearest taught pattern is $(1, 1, 1)$, which belongs to the 1-taught set and so the neuron will fire and output will be 1.

By applying the firing rule to all the undefined patterns we obtain the following truth table.

X_1	0	0	0	0	1	1	1	1
X_2	0	0	1	1	0	0	1	1
X_3	0	1	0	1	0	1	0	1
OUT	0	0	0	0/1	0/1	1	1	1

There still remain two undefined patterns, where the input is equidistant from patterns in both the 0-taught set and the 1-taught set. However, the firing rule enables the neuron to respond sensibly to patterns not seen during training.

Pattern recognition

On a larger scale, pattern recognition can be implemented using a feed-forward network (Fig. 3). Here we consider three neurons, each with three inputs. To demonstrate how this works, we train the network to recognise the letters X and L (Fig. 4) . The associated patterns are all black, and all white respectively.

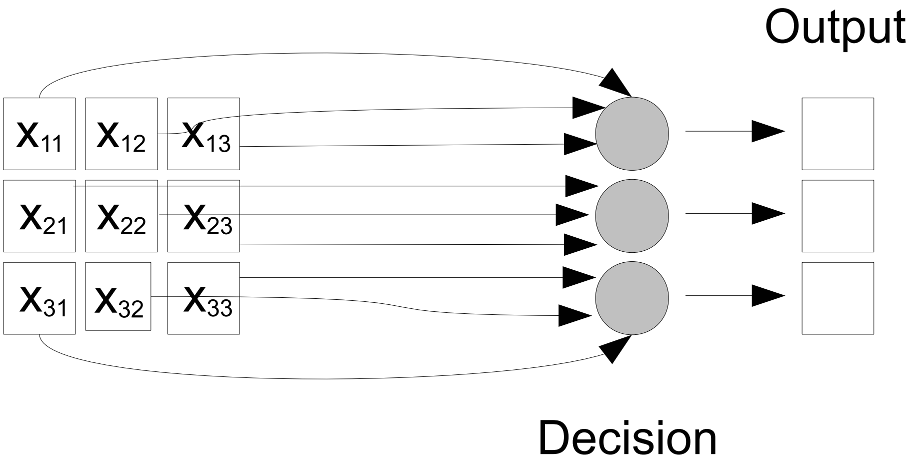


Figure 3: A feed-forward network for pattern recognition.

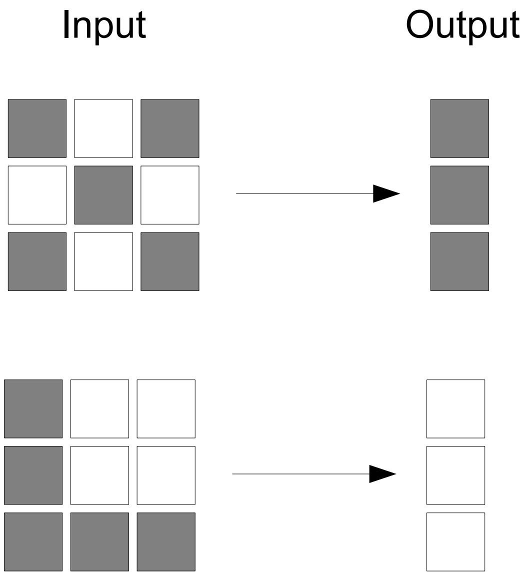


Figure 4: Training for pattern recognition in the X,L example.

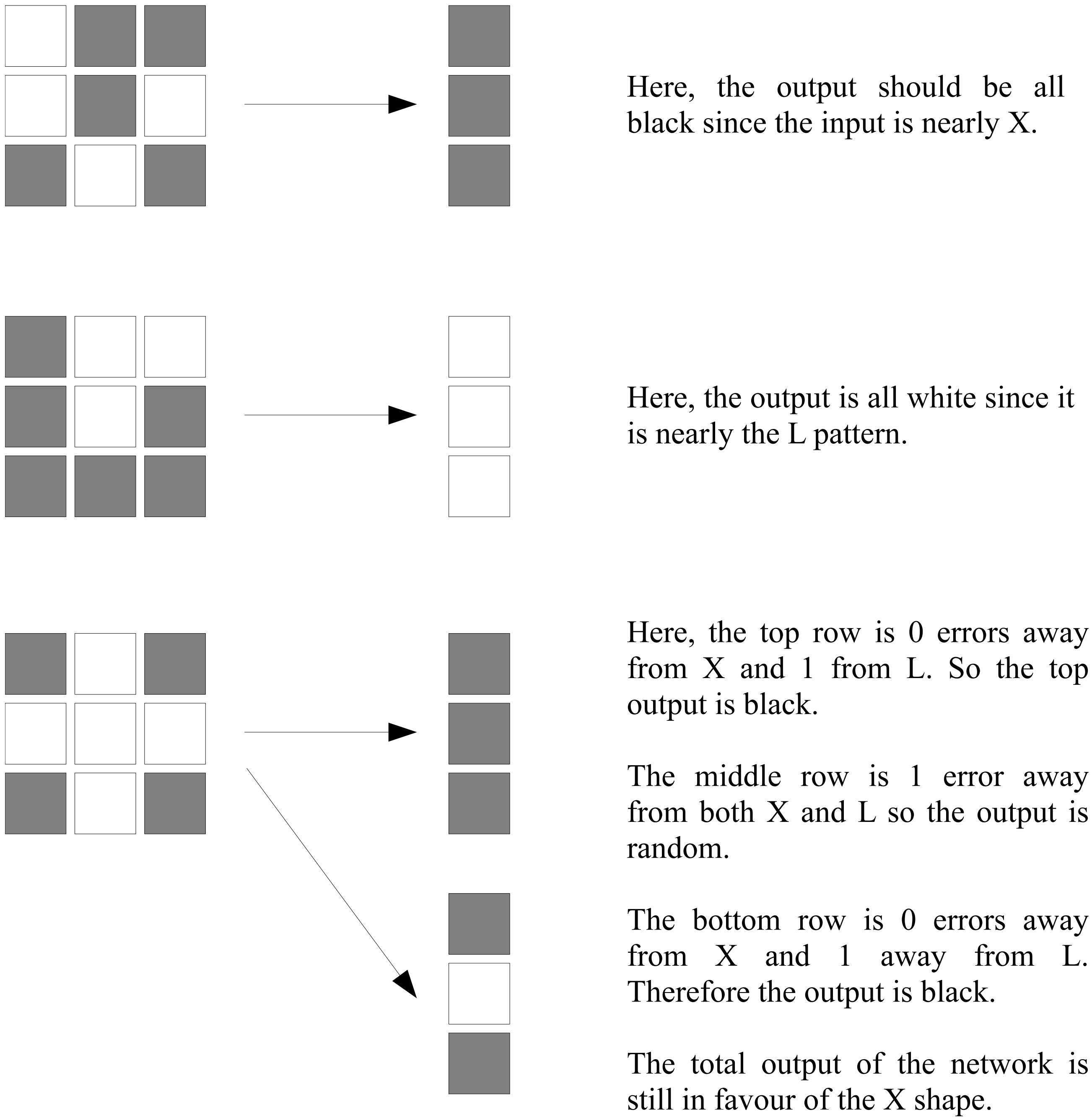
If we let black squares represent 1, and white squares represent 0 then the truth tables are

X_{11}	0	0	0	0	1	1	1	1
X_{12}	0	0	1	1	0	0	1	1
X_{13}	0	1	0	1	0	1	0	1
OUT	0	1	0/1	1	0	1	0	1

X_{21}	0	0	0	0	1	1	1	1
X_{22}	0	0	1	1	0	0	1	1
X_{23}	0	1	0	1	0	1	0	1
OUT	0/1	0/1	1	1	0	0	0/1	0/1

X_{31}	0	0	0	0	1	1	1	1
X_{32}	0	0	1	1	0	0	1	1
X_{33}	0	1	0	1	0	1	0	1
OUT	1	1	0	0	1	1	0	0

From these tables the following can be classified:



Here, the output should be all black since the input is nearly X.

Here, the output is all white since it is nearly the L pattern.

Here, the top row is 0 errors away from X and 1 from L. So the top output is black.

The middle row is 1 error away from both X and L so the output is random.

The bottom row is 0 errors away from X and 1 away from L. Therefore the output is black.

The total output of the network is still in favour of the X shape.

Figure 5: Solutions of the pattern recognition in the X,L example.

The inclusion of weights

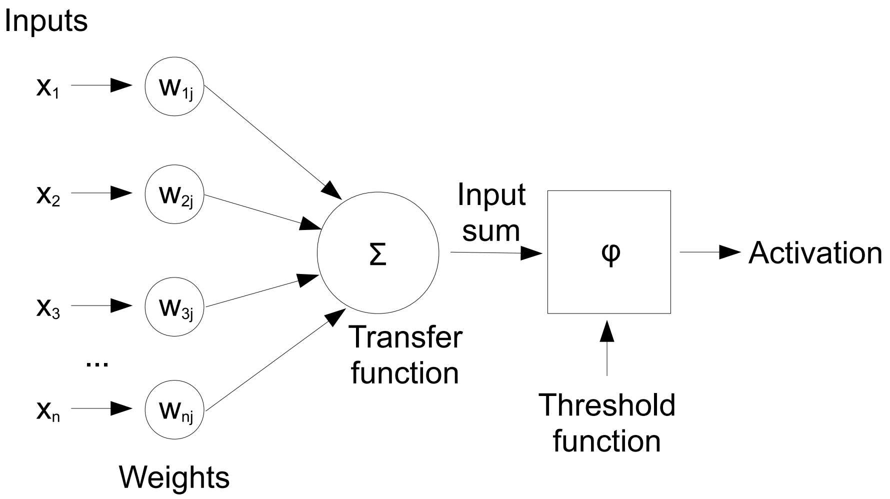


Figure 6: Let there be n inputs with signals $x_1, \ldots, x_n$, and weights $w_{1j}, \ldots, w_{nj}$. The inputs are multiplied by their respective weights, then summed together and compared to a threshold function. The output value then propagates to the input of the next layer of the network, through a synapse, or exits the system as part of an output vector.

More complicated learning and recognition tasks can be achieved by including weights attached to the inputs, e.g. using a perceptron (Fig. 6). The weights are adjustable and enable the system to learn by example. A typical training method for such a system is explained later in this subsection.

An alternative approach of associative learning was proposed in 1982 by Hopfield. He proposed the implementation of an artificial neural network with a non-algorithmic classification (pattern recognition) [31]. However, the learning stage is still algorithmic and consists in adjusting the strengths of couplings ("weights") in response to a training set of patterns. As a result, a potential energy landscape is formed in the phase space of the neural network (assuming neurons are coupled symmetrically), whose minima (attracting fixed points) represent the centres of classes, and the respective basins of attraction represent the classes. After the learning stage has finished, the weights are then fixed. New input patterns are then given by initial conditions, and classification occurs non-algorithmically as the network evolves towards the nearest attractor [27].

To understand how such an energy landscape is formed consider a network in which all nodes are connected symmetrically (i.e. bi-directional with equal weighting in either direction). Take two nodes i, j in the network connected by a positive weight w_{ij}. If node i is outputting 1 and j is outputting 0, when the system updates, the contribution from i to j will be positive and js activation could move above the threshold and could output 1. Due to symmetricity, the same could occur for $(0, 1)$. If both i and j are 1, then they would be reinforcing each others current output. So here the weight is acting to make i and j output 1. Thus $(1, 1)$ is the optimum state and we want this to have to lowest energy.

We can express the energy function as $e_{ij} = -w_{ij}x_ix_j$ [22]:

x_i	x_j	e_{ij}
0	0	0
0	1	0
1	0	0
1	1	$-w_{ij}$

Since w_{ij} is positive, the lowest energy occurs when both i and j are 1. The energy of the network is found by summation over all the pairings of the nodes [22],

$$E = -\frac{1}{2}\sum_i \sum_j w_{ij}x_ix_j. \tag{2.1}$$

To determine the change of energy after a particular node u is updated, then the total energy before the update will be (since $w_{mu} = w_{um}$)

$$E = -\frac{1}{2}\sum_{i \neq u} \sum_{j \neq u} w_{ij}x_ix_j - \sum_m w_{um}x_ux_m. \tag{2.2}$$

We can write this as

$$E = T - x_ua^u, \tag{2.3}$$

where from (2.2)

$$\begin{aligned} T &= -\frac{1}{2}\sum_{i \neq u} \sum_{j \neq u} w_{ij}x_ix_j, \\ a^u &= \sum_m w_{um}x_m. \end{aligned} \tag{2.4}$$

Then after the update, the change in energy, ΔE, through updating will then be given by

$$\Delta E = -\Delta x_ua^u, \tag{2.5}$$

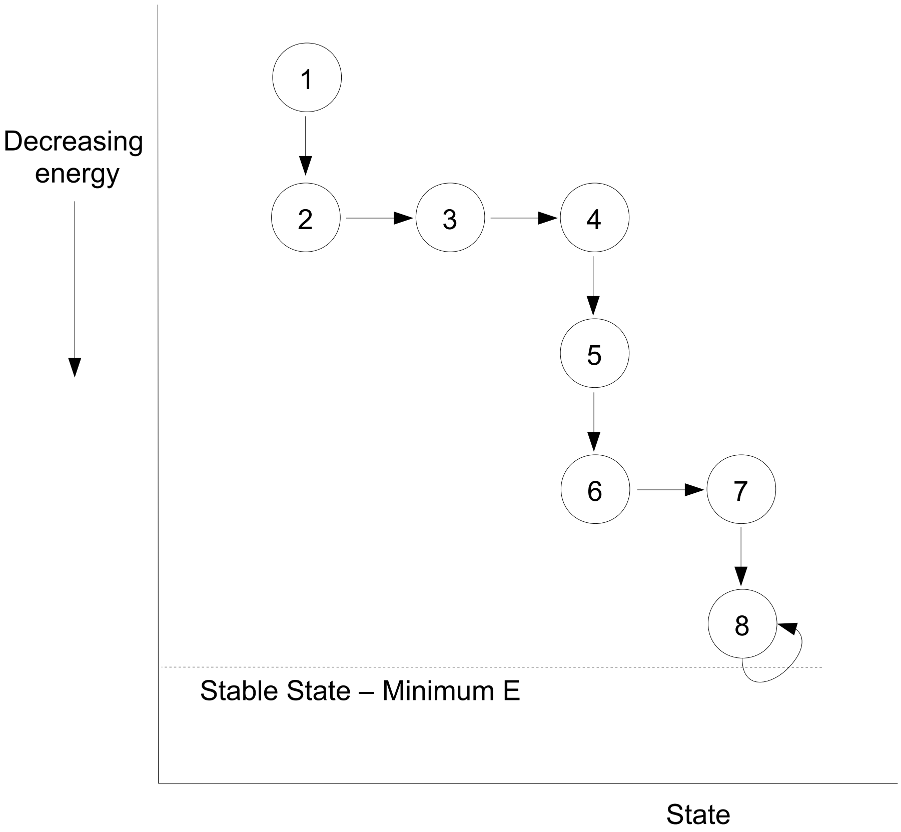


Figure 7: A schematic illustration of the path to recognition in an energy profile of a neural network. After each update the energy decreases or stays the same, and is bounded below by the stable state.

where $\Delta x_u = x_u^{(new)} - x_u$, is the change in output of neuron x_u.

If $a^u \geq 0$ then the output goes from 0 to 1 or stays at 1. Then $\Delta x_u \geq 0$, so that $\Delta x_u a^u \geq 0$, and therefore $\Delta E \leq 0$. If however, $a^u < 0$ then the output goes from 1 to 0 or stays at 0. Then $\Delta x_u \leq 0$, so that $\Delta x_u a^u \geq 0$, and therefore $\Delta E \leq 0$. In either case $\Delta E \leq 0$ and so the energy of the network decreases or stays the same. Furthermore, the energy is bounded below by the stable state $(1, 1)$, where once reached, the network will remain (Fig. 7).

Hopfield's second major breakthrough came in 1984, with the introduction of a continuous state and time model [32]. Here, he also introduced the possibility of external input and variable thresholds. A continuous Hopfield neural network can be written as follows [33, 34]:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_i + \sigma\bigg(\sum_{j=1}^{N} w_{ij}x_j - \Theta_i\bigg), \tag{2.6}$$

where x_i is the current state of ith neuron and w_{ij} is the connection strength (weight) between ith and jth neuron. Each neuron is essentially a threshold device, with threshold Θ_i or, in more general terms, a nonlinear device, with the nonlinearity described by the "sigmoid" function $\sigma(z)$, e.g. the logistic function $\sigma(z) = \frac{1}{1+e^{-z}}$.

This represents one of the possible models for artificial neural networks. It does not recreate the real firing and spiking transmission processes observed in biological neurons, but it does provides an approximate mathematical description of the most important feature of a neural network – the ability to classify.

If the function σ and the thresholds Θ_i are fixed, the system (2.6) can be perceived as a nonlinear dissipative dynamical system, whose vector field is determined by the weights w_{ij}. If the weights are symmetrical, i.e. $w_{ij} = w_{ji}$, one can introduce the concept of an energy function, E, with the right-hand sides of (2.6) being the coordinates of the gradient of E [34]. The function E typically has a number of local minima, each being a stable fixed point in the phase space with its own basin of attraction.

Each of the local minima of E, represent the most typical value of a certain class, or class centre. All patterns that belong to the same class are represented by the phase points in the basin of attraction of the respective stable fixed point. Since there are infinitely many points in the basin, there can be infinitely many patterns that belong to the same class, just like in reality. For example, infinitely many projections of a certain flower, registered by looking at it at different angles, are perceived as the same flower. The input pattern is represented by initial conditions in the phase space, which will fall into one of the basins of attraction available. The phase point will then follow the vector field and move towards the respective fixed point. When the fixed point is reached, the pattern is recognised.

This continuous approach means that the outputs can take any value between 0 and 1, rather than the traditional binary approach of allowing only the two end values. If the sigmoid function becomes steep, and approaches a step function, the energy minima will become close to the binary values of the discrete form. There were two major advantages to this approach. Firstly, the continuous nature of the time and the activation function brings artificial neural networks closer to their biological equivalents. Secondly, they were very easy to reproduce in an electronic system using amplifiers and resistors [22].

Learning

Learning in algorithm-based devices generally requires a teacher and can be fully supervised, semi-supervised, or reinforcement.

In a supervised learning environment, learning is done by comparing the output of the network with known answers [27]. Tasks that fall within this paradigm are classification and regression (also known as function approximation).

A variant of supervised learning is reinforcement learning, where the only feedback to the system is whether the output is correct or incorrect, not what the correct answer is [11]. Tasks that fall within this paradigm are control problems and other sequential decision making tasks. Another variant is semi-supervised learning, which makes use of both labelled and unlabelled data in the training process [8].

In an unsupervised learning environment, there is no learning goal defined [27]. The only available information is in the correlations of the input data. The network creates categories from these correlations and produces output signals corresponding to the input category. Tasks that fall within this paradigm are in general estimation problems whose applications include clustering, the estimation of statistical distributions, compression and filtering.

Furthermore, learning can take place on- or off-line. We say that a neural network learns off-line, if the learning phase and operating phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is typically performed on-line.

As mentioned previously, in neural networks, learning is achieved mostly through changes in the strengths of connections between neurons. To illustrate how this is achieved, consider a simple supervised learning technique for a network with output units whose states are a smooth continuous function of their directly connected inputs.

To begin, we assume that a network of N neurons is described by [28, 71]

$$x_j(t+\Delta t) = -\Theta_j + \sum_{i=1}^{N} w_{ji}\sigma(x_j(t)), \tag{2.7}$$

where x_j is the input to the jth neuron, Θ_j is the threshold of the jth unit, w_{ji} is the strength of connection between neurons i and j, and $\sigma(x)$ is the output (given by the sigmoid function).

We wish to train the network to produce a desired output for each input. In order to find w_{ji}, let us assume we have M classes, and that for each class c, we know the class centre $d_{j,c}$, $c = 1, \dots, M$, $j = 1, \dots N$. Furthermore, for a training set of patterns $x_j(t)$, we know to which class we belong. It is convenient to write $x_j(t) = x_{j,c}(t)$ to reflect our *a priori* *knowledge.* We adjust the weights in real time after each new training pattern arrives, i.e. at time t, we have $x_j(t) = x_{j,c}(t)$.

Suppose we have three classes, i.e. $M = 3$, and suppose that the training patterns arrive in a strict order: from class 1, then from class 2, then from class 3, then from class 1, and so on, i.e. $x_{j,1}(t), x_{j,2}(t+\Delta t), x_{j,3}(t+2\Delta t), x_{j,1}(t+3\Delta t), \ldots$.

Starting from any arbitrary configuration, the weights are adjusted to minimise the difference between the actual output and desired output. Smallest values of the learning rate, ϵ, give the optimal values for the weights, but increases the time taken for convergence to occur [28]. The weights are chosen as follows:

- $t, x_{j,1}(t)$

$$w_{j,1}(t+\Delta t) = w_{ji}(t) - \left[\epsilon \cdot \Big(\sigma\big(x_{j,1}(t)\big) - d_{j,1}\Big) \cdot \sigma\big(x_{i,1}(t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,1}(t)}\right] \tag{2.8}$$

- $t, x_{j,2}(t+\Delta t)$

$$\begin{aligned} w_{j,2}(t+2\Delta t) &= w_{ji}(t+\Delta t) - \Big[\epsilon \cdot \Big(\sigma\big(x_{j,2}(t+\Delta t)\big) - d_{j,2}\Big) \\ &\qquad \times \sigma\big(x_{i,2}(t+\Delta t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,2}(t+\Delta t)}\Big] \end{aligned} \tag{2.9}$$

- $t, x_{j,3}(t+2\Delta t)$

$$\begin{aligned} w_{j,3}(t+3\Delta t) &= w_{ji}(t+2\Delta t) - \Big[\epsilon \cdot \Big(\sigma\big(x_{j,3}(t+2\Delta t)\big) - d_{j,3}\Big) \\ &\qquad \times \sigma\big(x_{i,3}(t+2\Delta t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,3}(t+2\Delta t)}\Big] \end{aligned} \tag{2.10}$$

To obtain a single expression for the new weights after a training input from every class, take equation (2.10) for $w_{ji}(t+3\Delta t)$ and replace the $w_{ji}(t+2\Delta t)$ term with equation (2.9), and similarly for the subsequent w_{ji} until we have a single equation in terms of $w_{ji}(t)$,

$$\begin{aligned} w_{ji}(t+3\Delta t) = w_{ji}(t) \quad &- \quad \left[\epsilon \cdot \Big(\sigma\big(x_{j,1}(t)\big) - d_{j,1}\Big) \cdot \sigma\big(x_{j,1}(t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,1}(t)}\right] \\ &- \quad \Big[\epsilon \cdot \Big(\sigma\big(x_{j,2}(t+\Delta t)\big) - d_{j,2}\Big) \cdot \sigma\big(x_{j,2}(t+\Delta t)\big) \\ &\qquad \times \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,2}(t+\Delta t)}\Big] \\ &- \quad \Big[\epsilon \cdot \Big(\sigma\big(x_{j,3}(t+2\Delta t)\big) - d_{j,3}\Big) \cdot \sigma\big(x_{j,3}(t+2\Delta t)\big) \\ &\qquad \times \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,3}(t+2\Delta t)}\Big]. \end{aligned} \tag{2.11}$$

This can be written as

$$\begin{aligned} w_{ji}(t+3\Delta t) = w_{ji}(t) \quad - \quad & \epsilon \sum_{c=1}^{M=3} \Big[\Big(\sigma\big(x_{j,c}(t+(c-1)\Delta t)\big) - d_{j,c} \Big) \\ & \times \sigma\big(x_{i,c}(t+(c-1)\Delta t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,c}(t+(c-1)\Delta t)} \Big]. \end{aligned} \tag{2.13}$$

We have demonstrated that as time goes by, a new correction is added to w_{ji}. The closer the input is to the desired result, the smaller the change in w_{ji}. Additionally, it should be noted that, if we change the order of patterns, but keep the same number, K, of patterns for all classes, the final values of w_{ji} will be the same.

If at $t = 0$, $w_{ji}(0) = 0$, then

$$\begin{aligned} w_{ji}(MK\Delta t) \quad = \quad & -\epsilon \sum_{k=1}^{K} \sum_{c=1}^{M} \Big[\Big(\sigma\big(x_{j,c}((k-1)M+c)\Delta t)\big) - d_{j,c} \Big) \\ & \times \sigma\big(x_{i,c}((k-1)M+c)\Delta t)\big) \cdot \frac{d(\sigma(x))}{dx}\Big|_{x=x_{j,c}((k-1)M+c)\Delta t)} \Big]. \end{aligned} \tag{2.15}$$

This is only the simplest supervised learning rule, and many improvements have been proposed. A popular alternative solution, particularly for Hopfield networks, is the Hebbian learning rule, which rather than comparing with target outputs, is based on a multiplicative interaction between the pre- and post-synaptic activity. One major drawback of this, however, is the requirement for a large amount of high quality training data [73].

Comment

Although artificial neural networks are very successful at performing a predetermined learning and recognition task, this typically requires a large amount of diverse training for real world operation and a priori knowledge of the task at hand – a luxury not always available to living systems. A series of technical problems can occur as an artificial neural network learns, including the formation of spurious attractors. Also, the most natural way of learning for artificial neural network is supervised, while semi- or unsupervised learning require considerable complication of the algorithms. These drawbacks will be discussed further in Section 2.4.

2.3 Dynamical systems approach to intelligence

Zak and his collaborators offered some alternative methods to model learning in biological systems. These were based around dynamical systems theory, attempting to overcome some of the limitations of artificial neural networks. The central idea of the relevant papers of Zak [93, 91, 94], was to make the velocity vector field of a system that describes a living or learning system, depend on the input patterns it receives. In the context of learning, ideally the vector field should be formed without any supervision from a human. In this section, we discuss two different approaches by Zak: the phase velocity field approach, and Liouville feedback approach.

Phase velocity field approach

The phase velocity field approach came in 1990, and attempted to use an artificial neural network to control the structure of the phase space of the dynamical system. The approach aimed to use examples to train the network in an unsupervised learning environment, with attractors and basins of attraction forming at, and around, the location of the training examples. This is achieved by making the velocity vectors of the example nodes dependent of the proximity to their neighbours. The network is considered as an adaptive nonlinear dissipative dynamical system [94],

$$\dot{u}_i + ku_i = \sum_{j=1}^{N} w_{ij} g(u_j) + \eta_i, \tag{2.16}$$

where $i = 1, \ldots, N$, u_i represents the activity of the ith neuron, w_{ij} is the synaptic weight between neuron i and neuron j, g is a nonlinear function, η_i is a constant exterior input to the system for neuron i, and k is constant.

Equivalent patterns can vary (e.g. different shades of the same colour) due to, for example, noisy measurements, and so the system must be able to classify sufficiently similar patterns together. The goal of this work was to find the synaptic weights, w, and external inputs, η, to ensure that any trajectory originating inside the basin of attraction will tend towards its attractor. From this, any new pattern introduced to the network will be attracted to the local attractor, and thus the network performs associative pattern recognition.

The approach uses clusters in the training set as interpolation nodes of the vector field in the phase space. Then these velocity vectors impose constraints on the synaptic weights and external inputs. To choose these velocity vectors, he proposes the use of a “gravitational approach”, which simply makes the velocity vector dependent on the distances to other values. Suppose that the training point $u^{(m)}$ is attracted to all the other points $u^{(l)}$, $m \neq l$ by the gravitational force, with scaling coefficient α,

$$v_i^{(m)} = \alpha \sum_{l=1, l\neq m}^{M} \frac{u_i^{(l)} - u_i^{(k)}}{\left[\sum_{j=1}^{n} (u_j^{(l)} - u_j^{(m)})^2\right]^{3/2}}. \tag{2.17}$$

The actual velocities at same points are given by (2.16), which can be rearranged to give

$$\dot{u}_i^{(m)} = \sum_{j=1}^{N} w_{ij} g(u_j^{(m)} - u_{0i}) - k(u_i^{(m)} - u_{0i}). \tag{2.18}$$

One can then find the synaptic weights, w_{ij}, and the “centre of gravity” (class centre), u_{0i}, by minimising the error function,

$$E = \frac{1}{2} \sum_{m=1}^{M} \sum_{i=1}^{N} (v_i^{(m)} - \dot{u}_i^{(m)})^2. \tag{2.19}$$

The main advantage of this method is that the system learns in an unsupervised manner. However, the drawback is that this learning happens off-line, and there is a risk that the training set will not produce the desired classes, since they will be formed from a finite set of unmoderated data. Furthermore, there will be no possibility to correct itself later.

Liouville feedback approach

In 2003, Zak proposed the Liouville feedback approach, with the same fundamental aim of making the velocity vector field of a system depend on the input patterns it receives [91]. The approach this time, models the evolution of the probability distribution of the input as it is coupled to the vector field of a dynamical system describing a living system.

Consider a dynamical system,

$$\dot{y} = f(y, t), \tag{2.20}$$

with random initial conditions, $y(0) = Y$, and probability distribution given by

$$p_0 = p_0(Y), \tag{2.21}$$

subject to the constraints

$$p \geq 0, \qquad \int_{-\infty}^{\infty} p dY = 1. \tag{2.22}$$

The evolution of the probability density is given by the Liouville equation,

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial y}(pf) = 0, \tag{2.23}$$

which has a solution in the form $p = p(Y, t)$.

If (2.20) is run several times, then the probability density will evolve according to (2.23). The only source of randomness in (2.20) are the initial conditions.

Zak then proposes to shape the vector field in (2.20) by the instantaneous probability density distribution, p, so that the input creates the feedback to the vector field,

$$f(y,t) = \phi[p(y,t)], \tag{2.24}$$

so that,

$$\dot{y} = \phi[p(y,t)]. \tag{2.25}$$

He then chooses feedback in the form

$$f = -\sigma^2 \frac{\partial}{\partial y} \ln p, \tag{2.26}$$

so that,

$$\dot{y} = -\sigma^2 \frac{\partial}{\partial y} \ln p. \tag{2.27}$$

To be able to study this dynamical system, one must first have an expression for p, given by the Liouville equation. The choice of f in (2.26) means that the Liouville equation takes the form of the Fokker-Planck equation,

$$\frac{\partial p}{\partial t} = \sigma^2 \frac{\partial^2 p}{\partial Y^2}, \tag{2.28}$$

which, subject to initial conditions $p(Y, 0) = \delta(Y)$ (where δ denotes the Dirac delta function), has exact solution

$$p = \frac{1}{2\sigma\sqrt{\pi t}} \exp\left(-\frac{Y^2}{4\sigma^2 t}\right). \tag{2.29}$$

Substituting this back into (2.27) gives,

$$\dot{y} = \frac{y}{2t}, \tag{2.30}$$

which can be integrated to give,

$$y = A\sqrt{t}, \tag{2.31}$$

with constant of integration A.

Zak identified that, considering (2.31) with varying A, arrives at the whole ensemble characterizing the random process, and that the process is equivalent to Brownian motion (i.e. Gaussian white noise), with the same Fokker-Planck equation, (2.28). Furthermore, he demonstrated that interesting phenomena such as entanglement, and reversibility can occur in such systems [91, 92, 93].

This formulation is equivalent to an information based self-supervision, with the probability density providing feedback to the vector field. However, choosing the vector field in such a manner produces a Gaussian distribution which is not interesting on a dynamical level – the system modifies itself with time, but for the whole duration there is only one attractor, in the form of a stable fixed point. To enable the occurrence of more interesting behaviour, we want to generate a vector field with multiple attractors, rather than having to use networks of these Gaussian systems as proposed by Zak [93].

2.4 Motivation for this work

Having described some of the techniques to recreate learning in artificial systems, we now discuss the motivation for a new approach and the work contained in this thesis.

Mathematically, an artificial neural network can be regarded as a dynamical system with a unique structure. In fact, they seem to be the only class of dynamical system which *truly* modify themselves in response to external stimuli. Stochastic dynamical systems, for example, are perturbed by external stimuli, however when this stimulus is withdrawn, the system immediately reverts to its original structure. Artificial neural networks consist of individual units (neurons) with rigid velocity vector fields and flexible couplings between them. In a neural network we can *only* control values of couplings. Thus we are only controlling some of the parameters of the system – we do not have the ability to modify the whole structure of the vector field. On a local level, the structure may be changing as desired, however, in other areas changes may be undesirable or unexpected. Although a wide range of algorithms have been invented to find the optimal weights, none have found a way to solve this problem. This inability to control the whole vector field leads to the problem of "spurious minima".

Spurious minima are patterns that occur at local minima but do not correspond to any of the classes. They are typically formed by a linear combination of other stored patterns [27]. These minima mean that given suitable initial conditions, the network can evolve to one of these spurious states – corresponding to no known pattern. It should be said however, that their basins of attraction are smaller than those of the trained patterns making the chance of their result less. Nevertheless, their appearance in the phase space remains a problem.

We wish to introduce a kind of dynamical system that modifies itself in response to the external stimulus but without the limitations of artificial neural networks. Some progress was made in this direction by Michail Zak and his collaborators. They attempted to address the problem from two different viewpoints. In 1990, they attempted to find a method to determine the connection weights and training sequence of patterns such that it would lead to the desired configuration of attractors and their basins of attraction in the phase space [94]. Another interesting attempt came in 2003, coupling the input patterns to the velocity vector field [91].

The first of these methods, called the phase velocity field approach, appears to be an impractical approach to the problem. In practice, the sequence of training patterns is not generated in an optimal way, but is dictated by the situation, and thus in an unsupervised environment one cannot be sure that attractors will be formed in the desired locations. The second approach, using Liouville feedback, appears more promising, however does not seem to have been developed further. The main limitation in the current literature is the requirement that the process be a Gaussian random force, while in a real life system the random forces could be highly non-Gaussian. In this work we shall seek to extend the applicability of this concept to include non-Gaussian stimuli.

We aim to generalise the neural network concept, achieved by abandoning the “rigid units - flexible couplings” paradigm and making the vector field fully flexible and amenable to external force. This will produce the second class of dynamical systems that modify themselves to external force. However, we aim to be able to control the whole structure of the phase space, rather than just part of it. Furthermore, this will be performed in an unsupervised learning environment – overcoming the other major limitation of artificial neural networks. To achieve this, learning will be achieved by coupling the input to the vector field of the dynamical system – much like Zak's Liouville feedback approach [91]. However, we aim to remove the constraint on input processes and have a system that can function with any type of random force. Categories will be formed by similarities in these statistical properties. With the removal of the rigid units paradigm and the dependence of learning shifted entirely to the probability density distribution of the input, the problem of spurious attractors should be avoided.

Our aim also is to create a dynamical system that will function as an on-line learning system, that is learning and recognition take place simultaneously, with there being no need for pre-training. The combination of these changes will, hopefully, bring us closer to the workings of a biological learning system.

3 Dynamical Systems with Probability Density Feedback for Pattern Identification

In this section we aim to adopt a similar technique to the Liouville feedback approach of Zak, based on the coupling of the stimulus with the vector field of the system [91, 93]. In Section 2.3, we showed that Zak's method relied upon the input process being equivalent to Gaussian white noise. Here, we aim to create a method that works for a random input with arbitrary statistical properties. In this section, we generate such an input with the desirable probability density distribution by nonlinearly transforming a Gaussian process using two different techniques.

3.1 Non-Gaussian non-correlated random input

3.1.1 Creating a non-Gaussian non-correlated random signal

Here, we wish to generate a process with non-correlated values, i.e. a random process with a given probability density. To generate this process, the theory of nonlinear transformations will be used. We will start with a Gaussian white noise input and transform this to a process with the required distribution.

Let us consider a random process with uniform probability distribution so that the probability of generating a number between x and $x + dx$, denoted $p(x)dx$, is given by

$$p(x) = \begin{cases} const, & \text{if } a < x < b \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

and the probability distribution is normalised so that

$$\int_{-\infty}^{\infty} p(x)\, dx = 1. \tag{3.2}$$

Now, if we generate a uniform variate x (a particular outcome of a random variable) and take some function of it, $y = f(x)$, then the probability distribution of y denoted $p(y)dy$ is given by [63]

$$p(y) = p(x) \left| \frac{dx}{dy} \right| . \tag{3.3}$$

From (3.1) and setting $c = const$ gives

$$p(y) = c \left| \frac{dx}{dy} \right| , \tag{3.4}$$

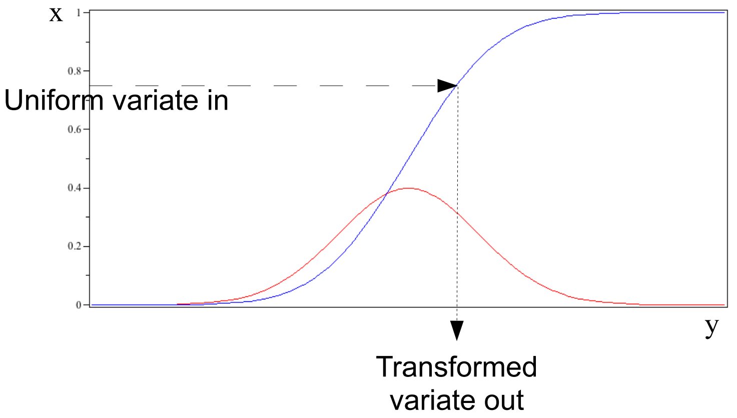


Figure 8: Transformation method for generating a random variate y (a particular outcome of a random variable) from a known probability distribution $p(y)$ (red). One must first select a uniform variate between 0 and 1. The corresponding value on the curve of the integral of the desired distribution (blue), $p(y)$, generates the new variate y. The integral of the desired distribution must be invertible.

then rearranging and integrating [3],

$$\begin{aligned} \frac{1}{c}\int p(y)dy &= x \\ \Rightarrow \frac{1}{c}F(y) &= x \\ \Rightarrow y &= F^{-1}(cx) \end{aligned} \tag{3.5}$$

where $F(z)$ is the indefinite integral of $f(z)$ and F^{-1} is the inverse function to F. This is demonstrated visually in Fig. 8. The ability to implement this transformation depends on the inversibility of the integral of F. For our case, $F(x)$ will be a monotonically increasing function, since it is the cumulative distribution function. A function $f : \mathbb{R} \to \mathbb{R}$ possesses an inverse as long as it is one-to-one, and thus this criteria will be met in our case.

Analytically we will have to perform the transformation twice: we start with Gaussian white noise which we transform to a uniform distribution, then from a uniform distribution to the desired distribution. When performing the first transformation, the problem is slightly different. In this case the required distribution is uniform rather than starting from a uniform variate. This in fact simplifies the problem. Performing the same calculation we find that $y = F(x)/c$ and no inversion for this part of the transformation is required. Computationally the problem is a little simpler. Random number generators programmed in software usually have a uniform distribution, and so we only have to apply one transformation.

This technique, however, can be used to create any distribution for which the probability density function is known and its cumulative distribution function is invertible. To begin, we shall consider a random process with a symmetric bimodal probability density distribution.

The next stage of this process is to calculate the evolution of the probability density distribution of this random process. In Zak's work this was achieved analytically using the Fokker-Planck equation – derived in Appendix A. The derivation requires a Gaussian random process and so we must attempt to modify this to allow for our non-Gaussian, transformed random process.

Specifically (A.18) will no longer be true, and therefore we will not be able to apply Isselis' theorem (A.19, A.20), meaning higher order terms in the Kramers-Moyal expansion (A.16) can no longer be ignored. Pawula's theorem states that if the expansion does not stop after the first or second term then it must contain an infinite number of terms [66]. Thus in this case, we know our expansion must contain an infinite number of terms. Given these technical issues, we now attempt to modify the equation for the evolution of the probability density for a non-Gaussian, non-correlated, transformed random process.

3.1.2 Evolution of the probability density of the non-correlated input

Recall, that in Zak's Liouville feedback approach (Section 2.3), the input to the system was coupled to the vector field of the dynamical system by its probability density distribution, given by the Fokker-Planck equation. The dynamical system was given by

$$\dot{y} = -\sigma^2 \frac{\partial}{\partial y} \ln p, \tag{3.6}$$

and the accompanying Fokker-Planck equation,

$$\frac{\partial p}{\partial t} = \sigma^2 \frac{\partial^2 p}{\partial Y^2}, \tag{3.7}$$

with random initial conditions $y(0) = Y$ whose time evolution of the probability density is represented by $p(Y, t)$.

Now, as already mentioned, since we no longer want the process to be of Gaussian nature, the evolution of its probability density distribution cannot be given by the Fokker-Planck equation. Here, we attempt to modify the equation to give the probability density evolution for an uncorrelated random process (i.e. a non-Gaussian white noise). We know that the evolution of the probability density distribution of a random process is given by the Kramers-Moyal expansion (A.16) and that this must contain an infinite number of terms. An equation with an infinite number of terms cannot be treated numerically and so we must consider whether we can reasonably approximate the infinite Kramers-Moyal expansion by an expansion truncated at a finite order, and indeed whether we can obtain this expansion. In our case we have

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial y} D^{(1)} p + \frac{\partial^2}{\partial y^2} D^{(2)} p + \ldots + (-1)^n \frac{-\partial^n}{\partial y^n} D^{(n)} p, \tag{3.8}$$

and we must now calculate the coefficients $D^{(n)}$. We know that, since our distribution is symmetric, odd numbered moments will be zero and thus we only need consider the calculation of even moments [82]. We consider a transformation of the Gaussian white noise, using the technique described in the previous section, with the following form

$$\xi(t) = F(G(\eta(t))), \tag{3.9}$$

where G transforms the Gaussian white noise, $\eta(t)$, to a uniform random process, and F transforms this uniform process to the process with required distribution. The coefficients $D^{(n)}$ of (3.8) are given by the moment functions [9]

$$D^{(n)} = \lim_{\Delta t \to 0} \frac{\langle \xi(t + \Delta t) - y \rangle^n}{n! \Delta t} \bigg|_{\xi_k(t) = x_k}. \tag{3.10}$$

Stratonovich showed that, for a polynomial transformation, the moment functions of the output process can be expressed as a linear combination of the moment functions of the Gaussian input process [82]. However, the formulae for the moment functions of the output process involve higher order moment functions of the input process. This is one of the characteristic features of nonlinear transformations as compared with linear transformations.

Now, the transformation function G is given by

$$\frac{1}{2} + \left[1 + \operatorname{erf}\left(\frac{z-\mu}{\sqrt{2\sigma^2}}\right)\right], \tag{3.11}$$

i.e., the cumulative distribution function of the Gaussian distribution.

To enable us to apply Stratonovich's transformation rule, we would like this transformation to be in the form of a polynomial, so we consider a Taylor expansion of erf(z),

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\left(z - \frac{z^3}{3} + \frac{z^5}{10} - \frac{z^7}{42} + \frac{z^9}{216} - \dots\right). \tag{3.12}$$

In particular, we wish to consider a symmetric bimodal probability density, and so we will consider a polynomial transformation function of order 5, of the form

$$F(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f. \tag{3.13}$$

To calculate the Kramers-Moyal coefficients for our transformed noise, we consider (A.31), where $\eta(t)$ is now the transformed noise function. We will again need to calculate the correlation functions that are used during the substitution in (A.32). To demonstrate that this method appears not to be feasible, let us consider a transformation

$$z(t) = g(\epsilon) = a\epsilon^2(t) - b\epsilon^4(t), \tag{3.14}$$

with input $\epsilon(t)$ be a Gaussian process with zero mean and correlation function

$$k(\tau) = \sigma^2 R(\tau). \tag{3.15}$$

The mean and correlation function are [82]

$$\begin{aligned}\langle z\rangle &= a\sigma^2 - 3b\sigma^4,\\ \langle zz_\tau\rangle &= a^2\sigma^4(1+2R^2) - 6ab\sigma^6(1+4R^2)\\ &\qquad + b^2\sigma^8(9+72R^2+24R^4).\end{aligned} \tag{3.16}$$

Since the correlation function for Gaussian white noise is equal to $2D\delta(t-t')$, we will have difficulties when encountering powers of the Dirac-delta function. Indeed for any polynomial transformation of Gaussian white noise, the correlation function will involve powers of the delta function, meaning that this method in its current form seems unnecessarily complicated. If we wanted to continue with this method, then the next stage would have been to introduce a coupling between the system and the probability density distribution of its input, using the formalism of Zak.

It appears that to utilise this analytical approach for the calculation of the probability density distribution, we must use a Gaussian random process. To enable us to achieve inputs to the system with different distributions, we will instead investigate using a nonlinear dynamical system with Gaussian white noise to produce a system with the desired probability density. We will then attempt to couple this system with its probability density evolution using Zak's formulation.

3.2 Non-Gaussian correlated input

We now model the required input random process as an output of another nonlinear system with a special design, subjected to Gaussian white noise, and then apply the Fokker-Planck equation approach to the whole system. Here, we introduce a probabilistic coupling between a nonlinear dynamical system and its internal structure. This ensures we avoid the need to modify the Fokker-Planck equation for non-Gaussian noise.

3.2.1 Creating a non-Gaussian correlated input signal

Again we wish to study a system with a bimodal probability density. For this we choose the stochastic van der Pol oscillator given by

$$\begin{cases} \dot{x} &= y, \\ \dot{y} &= \epsilon(1-x^2)y - x + D\eta(t), \end{cases} \tag{3.17}$$

where ϵ is the nonlinearity parameter, D is the strength of noise, and $\eta(t)$ denotes Gaussian white noise with mean μ and variance σ^2. We set $\epsilon = 0.1$ and $D = 0.1$.

The use of Gaussian white noise creates a computational issue. As stated previously, computer software typically provides a random number generator with a uniform probability distribution. If u_1 and u_2 are independent and uniformly distributed in the range 0 to 1, then consider y_1 and y_2 given by

$$y_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2)\sigma + \mu, \tag{3.18}$$

$$y_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2)\sigma + \mu. \tag{3.19}$$

This gives a modified Box-Muller transformation and is derived in Appendix B. It converts a 2-dimensional uniform distribution to a bivariate Gaussian distribution with mean μ and variance σ^2.

Using the modified Box-Muller transformation in our application of the system (3.17), we numerically solve the the system for a single realization using the Euler-Maruyama method for stochastic differential equations [40].

From Fig. 9 (Top), we see a single realization of $x(t)$ with the x value oscillating between approximately 2 and -2, with small variations around these values due to noise. Fig. 9 (Middle) shows the phase-portrait, y vs x, for the system. We see that the system demonstrates self sustained limit cycle oscillations. Fig. 9 (Bottom) shows the probability density

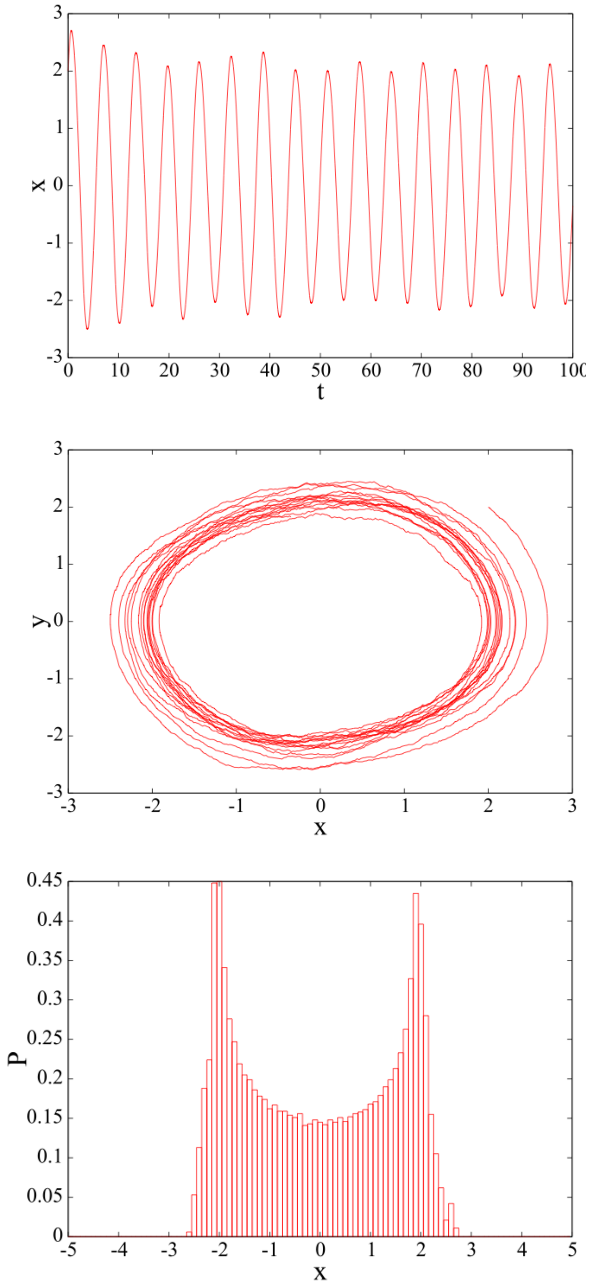


Figure 9: The stochastic van der Pol oscillator with noise (3.17) with $\epsilon = 0.1$, $D = 0.1$.

Top: A single realization of the x-variable, $x(t)$ vs t.

Middle: Phase portrait, $y(t)$ vs $x(t)$.

Bottom: Probability density of the x-variable, $P(x)$ vs x .

calculated numerically for the x variable from the realization. We see that the x variable demonstrates the desired bimodal probability density. Again we will investigate how the probability density evolves through time. This verifies that we have a system with bimodal probably density distribution for the input. We now wish to calculate the evolution of this probability density analytically using the Fokker-Planck equation.

3.2.2 Evolution of the probability density of the correlated input

We again use the Fokker-Planck equation derived in Appendix A. This time we can use the full result since we are now considering Gaussian white noise. The Fokker-Planck equation in two spatial dimensions for probability $p(\mathbf{x}, t)$ is given by

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = -\sum_{i=1}^{2} \frac{\partial}{\partial x_i}[D_i^{(1)} p(\mathbf{x},t)] + \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{\partial^2}{\partial x_i \partial x_j}[D_{i,j}^{(2)} p(\mathbf{x},t)] \tag{3.20}$$

To calculate the coefficients, D, we use (A.35) and (A.40) in two spatial dimensions to give

$$\begin{aligned} D_1^{(1)} &= y, \\ D_2^{(1)} &= \epsilon(1-x^2)y - x, \\ D_{1,1}^{(2)} &= 0, \\ D_{1,2}^{(2)} &= 0, \\ D_{2,1}^{(2)} &= 0, \\ D_{2,2}^{(2)} &= D. \end{aligned} \tag{3.21}$$

Then the Fokker-Planck equation for this particular problem is given by

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}[yp] - \frac{\partial}{\partial y}[(\epsilon(1-x^2)y - x)p] + \frac{\partial^2}{\partial y^2}[Dp]. \tag{3.22}$$

We will solve this nonlinear partial differential equation numerically, using the finite element method. The approach is based either on making the partial differential equation (PDE) into a system of ordinary differential equations (ODEs), which can then be numerically integrated.

Unlike the finite difference method, which represents the domain as a set of grid points, the finite element method represents the domain as interconnected subregions called elements, giving a piecewise approximation to the governing equations. Since these elements can be put together in a variety of ways, they can be used to represent complex shapes [95].

To apply the method we derive equation (3.22) in its weak form. Let

$$\begin{aligned} g_1 &= y, \\ g_2 &= \epsilon(1-x^2)y - x. \end{aligned} \tag{3.23}$$

Equation (3.22) then becomes

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}[g_1 p] + \frac{\partial}{\partial y}[g_2 p] - \frac{\partial^2}{\partial y^2}[Dp] = 0. \tag{3.24}$$

Applying the product rule to the second and third terms gives

$$\frac{\partial p}{\partial t} + p\frac{\partial g_1}{\partial x} + g_1\frac{\partial p}{\partial x} + p\frac{\partial g_2}{\partial y} + g_2\frac{\partial p}{\partial y} - \frac{\partial^2}{\partial y^2}[Dp] = 0. \tag{3.25}$$

Applying a finite difference equation for the time step and then multiplying by a test function v gives

$$v\frac{p_n - p_{n-1}}{\Delta t} + vp\frac{\partial g_1}{\partial x} + vg_1\frac{\partial p}{\partial x} + vp\frac{\partial g_2}{\partial y} + vg_2\frac{\partial p}{\partial y} - v\frac{\partial^2}{\partial y^2}[Dp] = 0. \tag{3.26}$$

We now integrate over the whole domain, Ω,

$$\iint\limits_{\Omega} \left[v\frac{p_n - p_{n-1}}{\Delta t} + vp\frac{\partial g_1}{\partial x} + vg_1\frac{\partial p}{\partial x} + vp\frac{\partial g_2}{\partial y} + vg_2\frac{\partial p}{\partial y} - v\frac{\partial^2}{\partial y^2}[Dp]\right] dxdy = 0. \tag{3.27}$$

Defining $p = 0$ on $\partial\Omega$ and applying integration by parts to the final term on the left hand side gives

$$\iint\limits_{\Omega} \left[v\frac{p_n - p_{n-1}}{\Delta t} + vp\frac{\partial g_1}{\partial x} + vg_1\frac{\partial p}{\partial x} + vp\frac{\partial g_2}{\partial y} + vg_2\frac{\partial p}{\partial y} + \frac{D}{2}\frac{\partial v}{\partial y}\frac{\partial u}{\partial y}\right] dxdy = 0. \tag{3.28}$$

Substituting (3.23) into (3.28) gives (3.22) in its weak form

$$\begin{aligned}\iint\limits_{\Omega} \quad &\left[v\frac{p_n - p_{n-1}}{\Delta t} + vp\frac{\partial y}{\partial x} + vy\frac{\partial p}{\partial x} + vp\frac{\partial}{\partial y}[\epsilon(1-x^2)y - x]\right. \\ &\left. + [v\epsilon(1-x^2)y - x]\frac{\partial p}{\partial y} + \frac{D}{2}\frac{\partial v}{\partial y}\frac{\partial u}{\partial y}\right] dxdy = 0. \end{aligned}\tag{3.29}$$

We consider a triangular mesh with 100 grid points in each dimension, with $\Omega = [-10, 10]$,$[-10, 10]$, and for the sake of numerical simplicity set $p = 0$ on the boundary (i.e. absorbing boundary conditions). We use piecewise-linear interpolation functions and perform time integration using a simple Euler method with a Gaussian initial condition. We then obtain the evolution of the probability density function through time. In Fig. 10, we plot the stationary probability distribution, and the evolution of the probability distribution in the x dimension. This gives us the bimodal probability density evolution that we desired. It was obtained by calculating the marginal probability distribution using [53]

$$p_X(x) = \int_y p_{X,Y}(x, y)\, dy, \tag{3.30}$$

where $p_{X,Y}(x, y)$ is the joint probability distribution.

Having calculated the evolution of the probability density distribution, we are now in a position to introduce the coupling using the potential function.

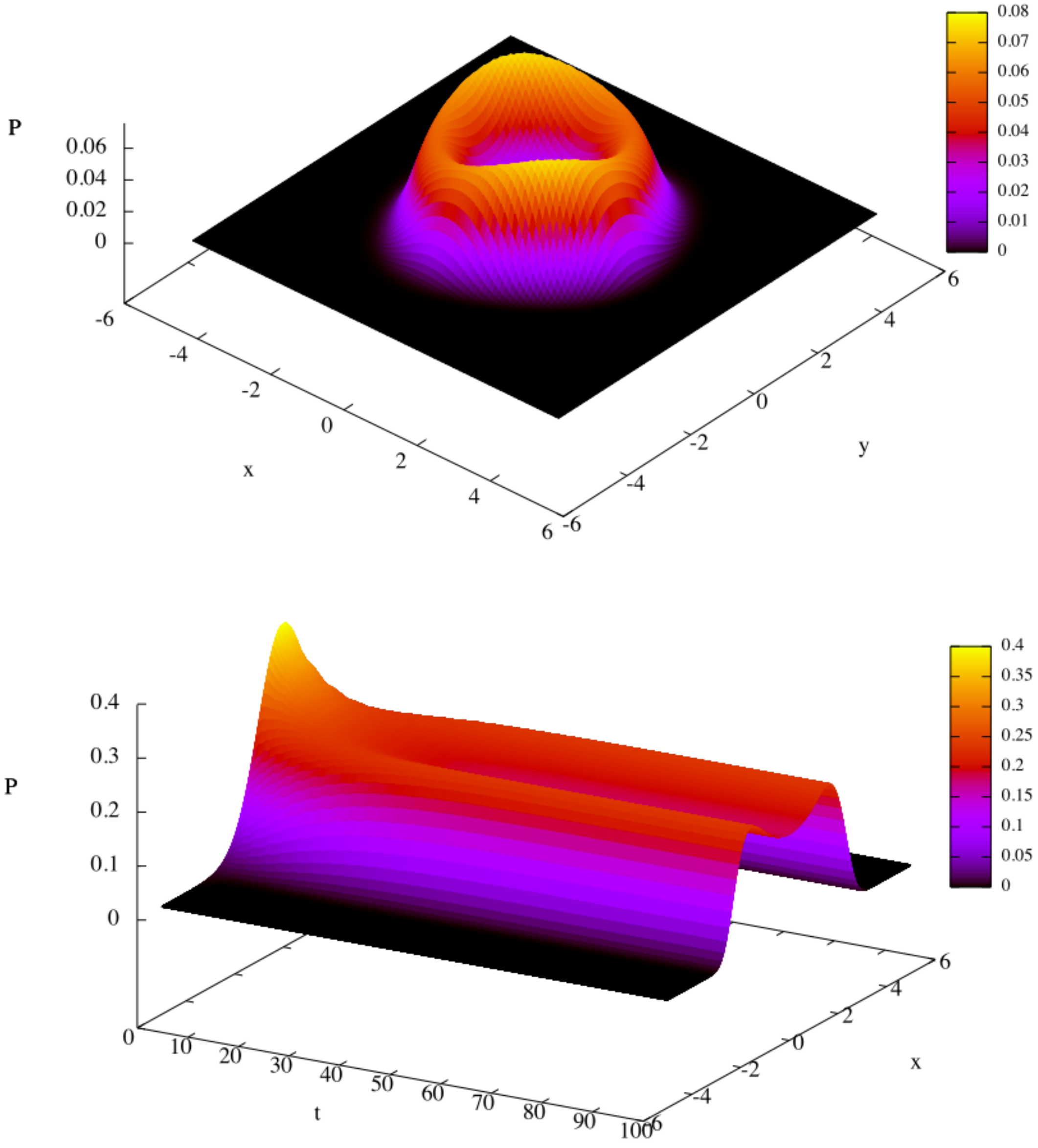


Figure 10: Top: plot of the stationary probability density function in 2 spatial dimensions, from the Fokker-Planck solution.
Bottom: plot of the evolution of marginal probability density function $p_1(x)$ through time.
These solutions are calculated from (3.29) and (3.30) with $\epsilon = 0.1$, $D = 0.1$.

3.2.3 Coupling via the potential function

We introduce the probabilistic coupling between the stochastic van der Pol oscillator and its internal structure in a similar style to Zak [91, 93]. The mathematical formalism is based upon coupling the dynamical system with the corresponding Fokker-Planck equation in a master-slave fashion: first the PDE is to be solved, then this solution is substituted into the coupled equation.

Taking the evolution of the probability distribution with opposite sign, forms a landscape with various maxima and minima. Following Zak's idea of introducing a coupling via a potential function, we propose a similar approach. From dynamical systems theory, if we take the gradient of this landscape with opposite sign we form a velocity vector field, dictating the motion of any trajectory within the system at any point in the phase space[1]. For a 1-D probability density distribution, the phase space will have fixed point attractors at the minima. These are formed due to the dissipative nature of the system.

Again following from Zak, we perturb the vector field by a force formed from the random input, v, to the evolution equation,

$$\dot{x} = \alpha \frac{\partial p}{\partial x} + qv. \tag{3.31}$$

The dynamics of the system are similar to the motion of a particle on the energy surface under the influence of gravity and friction, and subject to a random force. The particle slides downhill until it arrives at an attractor. When the random force is large enough for the particle to leave the attractor it escapes and moves to another. The basins of attraction correspond to the catchment areas around each minimum. The network creates categories from correlations in data, and produces outputs corresponding to the input category. We solve (3.31) using the Runge-Kutta numerical integration method with the results shown in Fig. 11.

From Fig. 11, we see the particle initially jumps around before the two peaks in the probability density form. When these peaks have formed, the particle jumps between the two attractors when the random force is great enough for the particle to leave its current well. Physically, this corresponds to the recognition of two categories in the input signal.

[1] Throughout this thesis, we study the system dynamics by considering the overdamped motion of a particle in the potential energy landscape, as is typical when studying dissipative gradient dynamical systems.

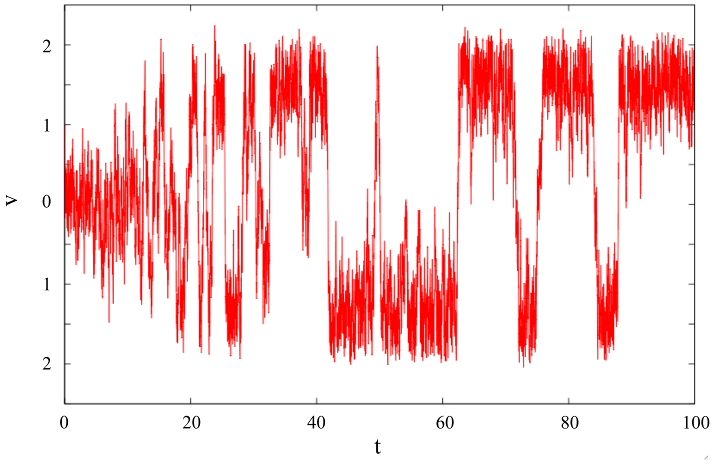


Figure 11: Numerical solution to the ODE of the coupled dynamical system given by (3.31) using the probability density shown in Fig. 10 with $\alpha = 1$, $q = 1$.

3.3 Discussion

We have considered two new approaches to modelling pattern recognition in an unsupervised learning environment. The main problem to occur was that the Fokker-Planck equation approach worked well if the random process at the input of the system was Gaussian white noise, but we want an input process with arbitrary characteristics. Two ways round this problem were investigated. The first method tried was to first apply a nonlinear transformation to Gaussian white noise, and then to use the Fokker-Planck equation approach. The second was to model the required input random process as an output of another nonlinear system subjected to Gaussian white noise, and then to apply the Fokker-Planck equation approach to the whole system.

We have shown that the first approach has problems when deriving the Fokker-Planck equation. We could not apply the Fokker-Planck equation due to the non-Gaussian nature of the input process. As discussed in Section 3.1.2, for a non-Gaussian noise the Fokker-Planck equation is not applicable and instead the Kramers-Moyal expansion must be used. We showed that for non-Gaussian noise the expansion must contain an infinite number of terms. As this is not numerically tractable, we were forced to consider truncating an infinite series the Kramers-Moyal expansion. Another problem was then encountered – the coefficients in the expansion became too complicated due to the correlation functions being powers of the delta function. Since no easy solution was found, we chose to discount this method in its current form.

The second approach allowed us to overcome these issues and the traditional Fokker-Planck equation could be used, since we had a nonlinear dynamical system with Gaussian white noise input. What we see from the second approach is that the system demonstrates some of the behaviour that is desired – it recognises the two distinct categories in the input. This approach does have one significant disadvantage in that the input signal is correlated. These correlations in the input signal make the problem statement less general and restrict the applicability of the method significantly, since we want the system to be able to deal with uncorrelated inputs too.

In the next section, we will consider an alternative method for the formation of the energy landscape. We aim to produce a dynamical system that adjusts itself to its input. This will, over time, form our energy function equivalent to the information based force from the probability densities in the previous methods. We will then again introduce the probabilistic coupling between this energy function and its input force. This technique will bypass the Fokker-Planck equation for calculating the evolution of the probability density and form the landscape in another way. This should overcome the issues found with the first two methods, using the more preferable features from each.

For all these methods the general idea remains the same, we couple the system to its probability density. In other words, we allow the vector field of the system to be influenced by the probability density distribution of its input. The strength in the concept lies in its ability to work in real time, continuously updating and improving through learning as time progresses. It requires no supervision, simply recognising patterns in the input using the likelihood of the input received in the signal. Once the mathematical difficulties have been overcome, we can consider more complex probability functions. We will then investigate using real life input signals and study how the system manages such an input.

4 Self-Shaping Systems

In this section, we propose a novel approach, again coupling the system with its statistical properties. However, this time we want to ensure that the stimulus is entirely random, without the correlations in its signal. It is clear that this problem arises due to the formation of the evolution of the probability density distribution, and so here we propose a new approach to construct this feature. In doing so, we create a new type of dynamical system subject to a continuous, generally random, external force. This force systematically *deforms* the velocity vector field according to a certain rule. In spite of the random nature of the force, it should give rise to a smooth vector field, which could eventually become fully deterministic and highly organised, and thus give rise to a new behaviour of the dynamical system. We will call such systems *self-shaping dynamical systems.*

Self-shaping systems will be different from the stochastic dynamical systems of the form $\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) + \epsilon(t)$, in which $\epsilon(t)$ is a random input and $f(x)$ is the stationary vector field. In these systems, the random input only *perturbs* the existing vector field, and once the input is removed the system reverts to its original structure. Whereas in self-shaping systems the vector field will be *created* by the random input.

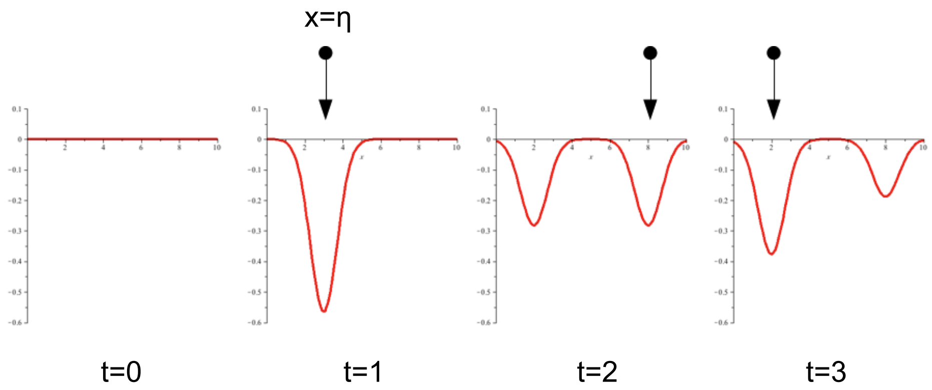


Figure 12: Illustration of the idea of the flexible energy landscape, for a one-dimensional landscape with domain in the x direction. We start by assuming that the landscape is initially flat, i.e. described as $U(x, 0) = 0$ (see $t = 0$). If a "stone" (the input) drops onto the landscape at $x = \eta$, a dent appears in the landscape, which is the deepest exactly at $x = \eta$, and gets shallower at larger distances from η (see $t = 1$). In other words, the landscape will learn about the occurrence of the stone and of its position. The process continues with further stones dropping at x values corresponding to the input (see $t = 2, 3$). Areas not subjected to input will begin returning to their original position.

4.1 Model

We will concentrate on the simplest form of these self-shaping systems, i.e. gradient (or potential) systems, where the vector field is the gradient of a certain energy function V,

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\frac{\partial V(\mathbf{x}, t)}{\partial \mathbf{x}}, \tag{4.1}$$

where x represents the location in N-dimensional space. We assume that the energy V is also a function of time t, taking into account the continuous shaping process. The state point in such a system behaves just like a massless particle that is placed into a potential energy landscape $V(\mathbf{x})$, which moves towards the relevant local minimum.

We now derive an equation describing the shaping of the energy landscape V in response to the random stimulus. On a conceptual level, the system is analogous to memory foam, as found in some bed mattresses. This foam takes the shape of a body pressed against it, but slowly returns to its original shape after the pressure is removed.

We use the auxiliary function $U(x,t)$ to describe the energy landscape, and assume that the landscape is elastic with elasticity factor $k < 1$, which models the capacity of the system to forget. We make the simple assumption that the deeper the dent at position x is, the faster the landscape tries to return to $U = 0$. However, the forgetting term could be modelled in other ways, depending on the requirements of the situation. The landscape is exposed to a continually varying external stimulus $\eta(t)$, where at any new time moment t a new "stone" (corresponding to the input) drops at position $\mathbf{x} = \eta(t)$. With this formulation, the landscape will undergo a continuous shaping process. The process is illustrated by Fig. 12.

Let us consider how the landscape changes over a small, but finite time interval Δt:

$$U(\mathbf{x}, t + \Delta t) = U(\mathbf{x}, t) - g(\mathbf{x} - \eta(t))\Delta t - kU(\mathbf{x}, t)\Delta t, \tag{4.2}$$

where $g(z)$ is some non-negative bell-shaped function, describing the shape of a single dent, e.g. a Gaussian function,

$$g(z) = \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left(- \frac{z^2}{\sigma_z^2}\right). \tag{4.3}$$

In (4.2) move $U(\mathbf{x}, t)$ to the left-hand side, divide both parts by Δt, and take the limit as $\Delta t \to 0$, to obtain

$$\frac{\partial U(\mathbf{x}, t)}{\partial t} = -g(\mathbf{x} - \eta(t)) - kU(\mathbf{x}, t). \tag{4.4}$$

Returning to the discrete form in (4.2) to study long term behaviour, let,

$$U(\mathbf{x}, 0) = 0. \tag{4.5}$$

Then,

$$
\begin{aligned}
U(\mathbf{x}, \Delta t) &= -g(\mathbf{x}-\eta_1)\Delta t, \\
\\
U(\mathbf{x}, 2\Delta t) &= -g(\mathbf{x}-\eta_1)(1-k)\Delta t - g(\mathbf{x}-\eta_2)\Delta t, \\
\\
U(\mathbf{x}, 3\Delta t) &= \big(-g(\mathbf{x}-\eta_1)(1-k)\Delta t - g(\mathbf{x}-\eta_2)\big)(1-k)\Delta t - g(\mathbf{x}-\eta_3)\Delta t \\
&= -g(\mathbf{x}-\eta_1)(1-k)^2\Delta t - g(\mathbf{x}-\eta_2)(1-k)\Delta t - g(\mathbf{x}-\eta_3)\Delta t, \\
\\
&\vdots \\
\\
U(\mathbf{x}, n\Delta t) &= -g(\mathbf{x}-\eta_1)(1-k)^{n-1}\Delta t - g(\mathbf{x}-\eta_2)(1-k)^{n-2}\Delta t \qquad (4.6) \\
&\quad - \ldots - g(\mathbf{x}-\eta_n)\Delta t. \qquad (4.7)
\end{aligned}
$$

For larger values of k, the shape of the energy landscape, U, is more influenced by the most recent inputs to the system, and it tends to forget what happened in the more distant past. We now perform the following change of variables to see if we can achieve some sort of stationary behaviour,

$$V = \frac{U}{t}, \quad \frac{\partial V}{\partial t} = \frac{1}{t}\left(\frac{\partial U}{\partial t} - V\right), \quad \frac{\partial U}{\partial t} = t\frac{\partial V}{\partial t} + V.$$

Rewrite this as follows,

$$t\frac{\partial V}{\partial t} + V = -g(\mathbf{x}-\eta) - kV(\mathbf{x}, t)t.$$

Then,

$$\frac{\partial V}{\partial t} = -\frac{1}{t}\big(V + g(\mathbf{x}-\eta)\big) - kV(\mathbf{x}, t). \qquad (4.8)$$

Depending on the statistical properties of the input we can demonstrate that, under suitable conditions for the input and kernel function, the landscape will tend to the negative of the probability density distribution. Here, we assume $k = 0$, i.e. the system does not forget what it learnt.

Let us now consider how the new landscape (4.8) changes over a small but finite time interval Δt with $k = 0$

$$V(\mathbf{x}, t + \Delta t) = V(\mathbf{x}, t) - \frac{\Delta t}{t + \Delta t}\big(g(\mathbf{x} - \eta(t)) + V(\mathbf{x}, t)\big). \qquad (4.9)$$

Let

$$V(\mathbf{x},0)=0. \tag{4.10}$$

Then,

$$\begin{aligned}
V(\mathbf{x},\Delta t) &= -g(\mathbf{x}-\eta_1),\\[2ex]
V(\mathbf{x},2\Delta t) &= -g(\mathbf{x}-\eta_1)-\frac{1}{2}\big(g(\mathbf{x}-\eta_2)-g(\mathbf{x}-\eta_1)\big)\\
&= -\frac{1}{2}g(\mathbf{x}-\eta_1)-\frac{1}{2}g(\mathbf{x}-\eta_2),\\[2ex]
V(\mathbf{x},3\Delta t) &= -\frac{1}{2}g(\mathbf{x}-\eta_1)-\frac{1}{2}g(\mathbf{x}-\eta_2)-\frac{1}{3}\left(g(\mathbf{x}-\eta_3)-\frac{1}{2}g(\mathbf{x}-\eta_1)-\frac{1}{2}g(\mathbf{x}-\eta_2)\right)\\
&= -\frac{1}{3}g(\mathbf{x}-\eta_1)-\frac{1}{3}g(\mathbf{x}-\eta_2)-\frac{1}{3}g(\mathbf{x}-\eta_3),\\
&\vdots\\[2ex]
V(\mathbf{x},n\Delta t) &= V(\mathbf{x},(n-1)\Delta t)\left(1-\frac{1}{n}\right)-\frac{1}{n}g(\mathbf{x}-\eta_n)\\
&= -\frac{1}{n}g(\mathbf{x}-\eta_1)-\frac{1}{n}g(\mathbf{x}-\eta_2)-\ldots-\frac{1}{n}g(\mathbf{x}-\eta_n). \qquad (4.11)
\end{aligned}$$

Now, the time averaged density of a random process is given by [43]

$$P(X)=\langle\delta(X-x(t))\rangle, \tag{4.12}$$

where

$$\langle f(x)\rangle=\frac{1}{N}\sum_{i=1}^{N}f(x_i). \tag{4.13}$$

So, that if $g(x)\to\delta(x)$ in equation (4.11) then we have the time averaged density of the random process $\eta(t)$.

Now consider the evolution of $V(\mathbf{x},t)$, where the N-dimensional input vector $\eta(t)$ is a realization of a *strict-sense stationary* and *ergodic* random process $W(t)$ with some arbitrary probability density distribution $p_N^W(\eta_1,\eta_2,\ldots,\eta_N)$.

Stationarity implies that p_N^W does not change in time, and due to *ergodicity*, any single realization $\eta(t)$ contains all information about p_N^W. In other words, any statistical characteristic can be obtained from $\eta(t)$ by averaging over time, rather than over the ensemble of realizations that would have been required for a non-ergodic process [82].

We will show after a period of time time, with suitable choice of kernel function, V takes the shape of p_N^W.

The term $g(\mathbf{x} - \mathbf{W})$ is a nonlinear smooth function of an ergodic process $\mathbf{W}$. As proved by Wolf [88], "zero-memory nonlinear operations on ergodic processes are ergodic" – therefore, $g(\mathbf{x} - \mathbf{W})$ is also an ergodic random process. Thus we can replace the time average (4.11) by the statistical average,

$$\overline{(V + g(\mathbf{x} - \mathbf{W}))} = \int_{-\infty}^{\infty} V p_N^W(\eta) d\eta + \int_{-\infty}^{\infty} g(\mathbf{x} - \eta) p_N^W(\eta) d\eta. \tag{4.14}$$

In the above, the integral with respect to η represents N integrals with respect to the components $\eta_1, \ldots, \eta_N$ of vector η.

Since V does not depend on η explicitly, the first term in the right-hand side of (4.14) is equal to V. The second term is the convolution of $p_N^W(\eta)$ with the function $g(\eta)$.

If $g(\mathbf{x} - \eta) = \delta(\mathbf{x} - \eta)$, where $\delta(z)$ is Dirac delta-function of several variables, this term is equal to $-p_N^W(\mathbf{x})$, due to the sifting property of delta-function [7],

$$\int_{-\infty}^{\infty} f(t)\delta(t - T)dt = f(T). \tag{4.15}$$

Since the behaviour of V is stationary, it follows that the expression (4.14) is equal to 0. We therefore demonstrated that as time t goes to infinity, $V(\mathbf{x}, t)$ tends to $-p_N^W(\mathbf{x})$, provided that $g(z)$ tends to the Dirac delta-function and $k = 0$.

The shaping mechanism which we employed for gradient systems is related to the kernel density estimation used in statistics [72]. Here, we incorporated this mechanism into the continuous dynamical shaping of the vector field, which is done for the first time to the best of our knowledge. Also, the standard assumptions about the kernel density estimators include the statistical *independence* of the successive values of the input. Namely, a sequence of input numbers/vectors is regarded as a collection of the values of some random (scalar or vector) *variable* with a certain probability density distribution. The only requirements used are those of stationarity and ergodicity of the random process.

Recognition in self-shaping systems

We have proposed a non-algorithmic learning process for self-shaping systems. Specifically, the system should adjust its internal structure in response to sensory input. We have introduced a mathematical prototype of this machine – a dynamical system, that shapes its vector field in response to the external stimulus – i.e. we describe the first component of the thinking process. The model does not require any pre-training, however, human influence can be included at any point if desired.

Recognition in self-shaping systems is achieved courtesy of the attractors in the phase space. Since we only consider gradient systems, the only stable states possible are fixed points. Considering (4.4) and using the inputs to the landscape as the initial conditions to the dynamical system,

$$\frac{d\mathbf{x}}{dt} = -\frac{\partial V(\mathbf{x}, t)}{\partial \mathbf{x}}, \tag{4.16}$$

the state will start at the input before evolving to the local minimum – equivalent to the most probable state in a particular class. We can then repeat this for the new input at every time moment (Fig. 13).

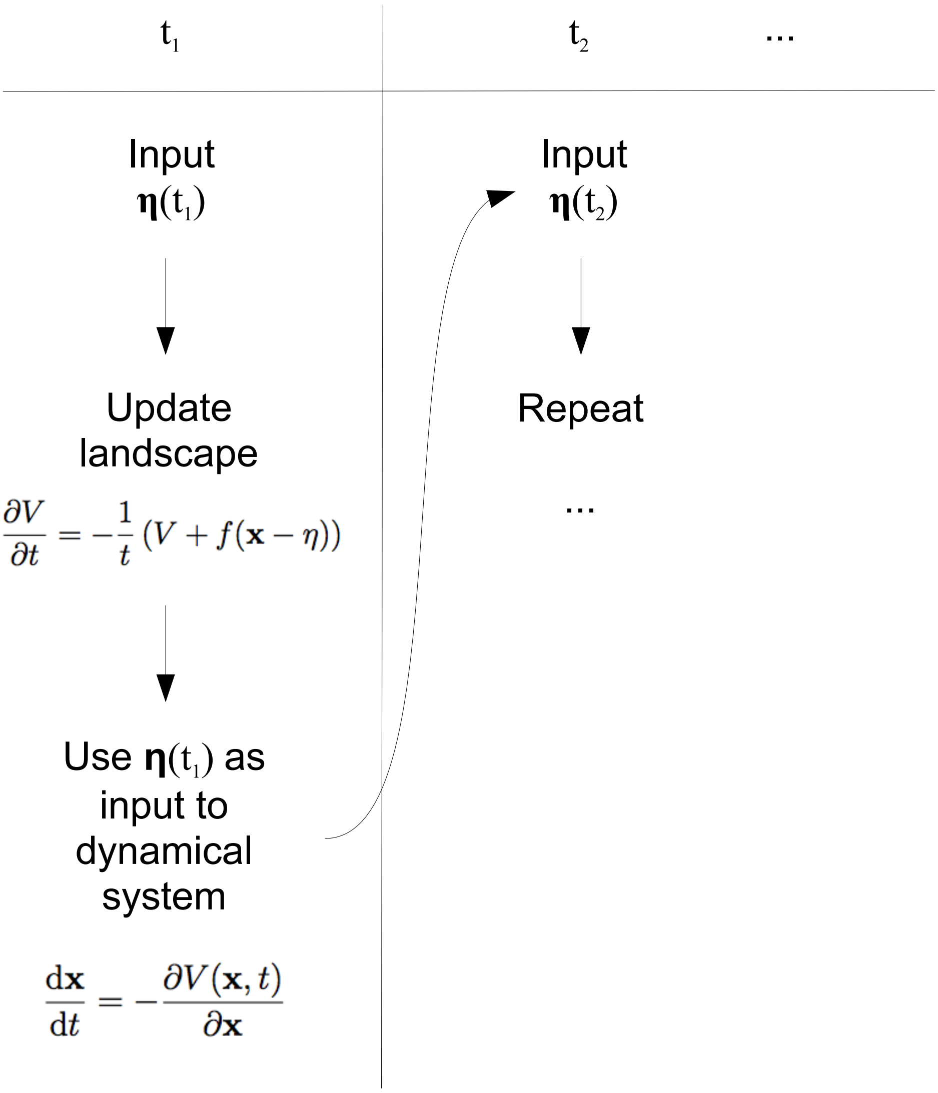

t_1
t_2
...
Input
$\boldsymbol{\eta}(t_1)$
Input
$\boldsymbol{\eta}(t_2)$
Update landscape
$\frac{\partial V}{\partial t} = -\frac{1}{t}\left(V + f(\mathbf{x} - \eta)\right)$
Repeat
...
Use $\boldsymbol{\eta}(t_1)$ as input to dynamical system
$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\frac{\partial V(\mathbf{x}, t)}{\partial \mathbf{x}}$

Figure 13: A flowchart describing the learning and recognition process in a self-shaping dynamical system.

More general dynamical systems

Dynamical systems can be more general than (4.16). Firstly, one can certainly make the landscape U depend on as many variables as we need (say N), in order to incorporate all state variables describing the device. Then instead of equation (4.16) we have N equations,

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = -\frac{\partial U(x_1, x_2, \ldots, x_N)}{\partial x_1}, \quad \ldots, \quad \frac{\mathrm{d}x_N}{\mathrm{d}t} = -\frac{\partial U(x_1, x_2, \ldots, x_N)}{\partial x_N}. \tag{4.17}$$

Secondly, in a general dynamical system, rates of change of its variables do not have to be derivatives of the same landscape function U, but can be some arbitrary functions

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = v_1(x_1, \ldots, x_N), \quad \ldots, \quad \frac{\mathrm{d}x_N}{\mathrm{d}t} = v_N(x_1, \ldots, x_N). \tag{4.18}$$

The more general idea is to let the random stimulus leave a localised “splash” in the vector field of (4.18) at every new time moment. Following the idea of the dents accumulated in the landscape, we can allow the vector field of (4.18) to accrue these splashes and thus reconfigure itself continuously. We predict that it will be possible to construct self-shaping systems that develop more complex attractors, such as limit cycles and chaotic attractors.

4.2 Examples of learning in a self-shaping system

Here, we will only consider gradient self-shaping systems. We shall test the validity of the concept, first using random data as the input, and second using musical data. In all cases we will let $k = 0$ in (4.8).

4.2.1 Random data as input

Here, we shall apply two types of scalar stimuli to the one-dimensional system (4.8). Both have a bimodal probability density distribution, with similar two-peak shape, but first we consider the case where the input has non-correlated values, and second the case where consecutive values are correlated. Non-correlated means that current and future values of a process have no dependence on its previous values, whereas a correlated process does have such a dependence. In Fig. 14, the evolution of $V(x,t)$ is illustrated, and the probability densities are shown by solid lines at the front of Fig. 14(a,c)).

The uncorrelated stimulus illustrated in Fig. 14(a,b), is obtained by taking Gaussian white noise, $\epsilon(t)$, and applying a nonlinear transformation (using the technique described in Section 3), f, that changed its probability density distribution,

$$\eta(t) = f(\epsilon(t)). \tag{4.19}$$

In this case we will not have the same difficulties as found in the method in Section 3.1.2. Here, there is no longer a requirement to find the Fokker-Planck equation. Instead, the vector field is being automatically shaped as it learns.

The correlated stimulus in Fig. 14(c,d) is obtained by applying Gaussian white noise to a differential equation describing a particle moving in a non-symmetric double-well potential with large viscosity [49],

$$\dot{\eta}(t) = g(x) + \epsilon(t) \tag{4.20}$$

where $g(x)$ is the nonlinear function providing the shape of the potential.

The actual signals applied are shown by filled circles in Fig. 14(b,d), with the depth of the landscape represented by the colour, and in $g(z)$ we used $\sigma_z = \sqrt{0.1}$. One can see that eventually both landscapes shape into the respective probability density distributions, but if the stimulus values are uncorrelated, the convergence is faster.

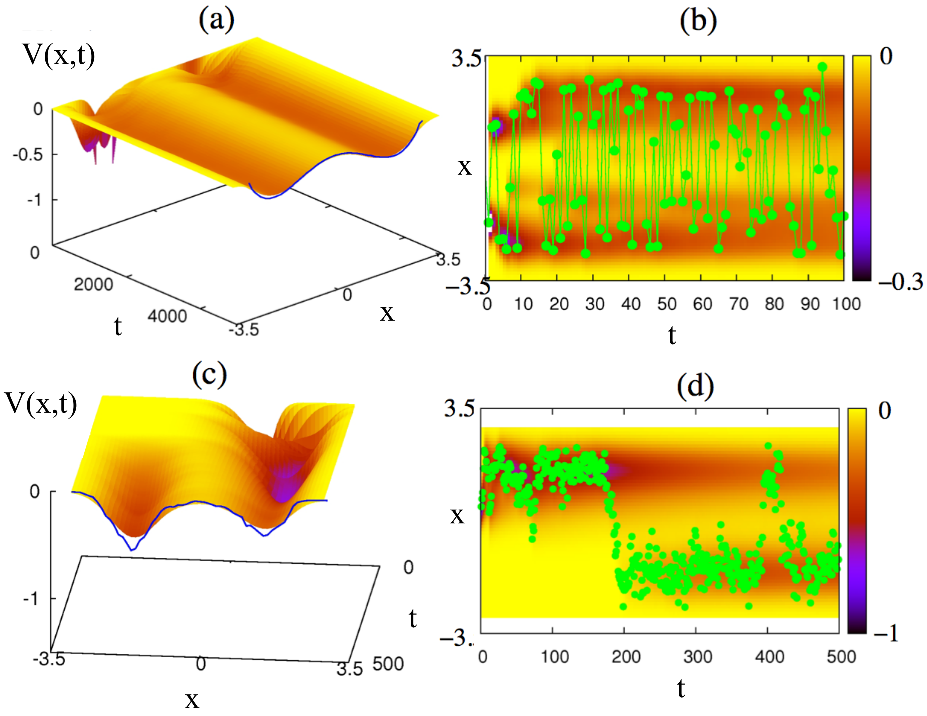


Figure 14: Evolution of the energy landscape $V(x,t)$ as the random stimulus is applied by numerically simulating Eq. (4.8) and using random numerical data as input. In (a,b) the consecutive values of the stimulus are uncorrelated (4.19) with a probability density distribution given by $P(f(\epsilon(t))\approx -0.01245x^4+0.1065x^2+0.0482$, and in (c,d) successive values are correlated (4.20) with $g(x)=3(x-x^3)/5$ and $\mathrm{Var}(\epsilon)=0.25$. (a,c) 3-D view of the evolution in time of the landscape; (b,d) projection of $V(x,t)$ onto (x,t) plane shown by colour, and the stimulus applied – by green circles. In (a,c) the probability density distribution (from realization) of stimulus is given by solid line at the front.

4.2.2 Application to musical data

We now demonstrate how a gradient self-shaping system automatically discovers and memorises musical notes and phrases (collections of notes). The nursery rhyme "Mary had a little lamb" (Fig. 15) was performed on a flute by an amateur musician, with the melody repeated six times [35]. The version played involves three musical notes (A, B and G), consists of 32 beats and was chosen for its simplicity to illustrate the principle.

The signal was recorded as a wave-file with sampling rate 8kHz. In agreement with what is usually done in speech recognition [16], the short-time Fourier Transform (STFT) was applied to the waveform with a sliding window of duration $\tau = 0.75$ sec, roughly the duration of each beat.

The STFT is used to determine the sinusoidal frequency and phase content of sections of a signal as it changes through time. The signal is multiplied by a window function which is non-zero for only a short period of time. The Fourier transform of the resulting signal is then taken as the window is slid along the time axis, resulting in a two-dimensional representation (X) of the signal [16]

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)\omega(t-\tau)e^{-i\omega t}dt, \tag{4.21}$$

where $\omega(t)$ is the window function – in our case this is a Hann window (a Gaussian-like bell shaped function, centred around zero), and $x(t)$ is the signal to be transformed. In essence, $X(\tau, \omega)$ is the Fourier transform of $x(t)\omega(t-\tau)$, a complex-valued function describing the phase and magnitude of the signal over time and frequency. The STFT produces a frequency-time domain representation of the signal – called a spectrogram. Taking a cross section of the spectrogram at a particular time moment will give the frequency spectrum.

In order to determine different notes within a piece of music, we must first know a little background. What we hear as a single sound when someone is speaking, is really a mechanical wave that is an oscillation of pressure typically through air (but could be a solid, liquid or gas), composed of frequencies within a range of hearing, sufficiently strong to stimulate the bodies hearing organs [56]. The waveform consists of a fundamental frequency (measured in Hertz), and a whole series of harmonics – multiples of the fundamental frequency. We do not normally hear the harmonics as separate tones because they have an increasingly lower amplitude than the fundamental frequency the higher they go. All sinusoidal and many non-sinusoidal waveforms are periodic, and the fundamental frequency is defined as the lowest frequency of such a waveform.

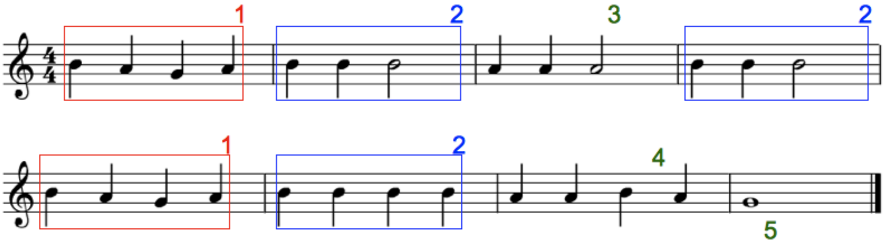


Figure 15: Musical manuscript for "Mary had a little lamb". Different coloured boxes around bars identify the different phrases.

Pitch is the major auditory attribute of musical tones, quantified as the fundamental frequency of the tone. When considering the Fourier transform of a waveform we observe peaks in the spectrum at the location of the fundamental frequencies and their harmonics. So to determine the musical note, the highest spectral peak was extracted for each window, which corresponded to the fundamental frequency, fHz, of the given note.

A sequence of frequencies $f(t)$ was used as the input to the system (4.8). It should be mentioned that each value of $f(t)$ was slightly different from the exact frequency of the respective note, because of the natural variability introduced by a human musician, and so the signal $f(t)$ was in fact random, as seen from Fig. 16(b).

Firstly, we illustrate how individual musical notes can be automatically identified. The one-dimensional system (4.8) received the input signal $\eta(t) = f(t)$, resampled to 8Hz to save computational time. We consider the function $f(t)$ as a realization of a 1st-order stationary and ergodic process $F(t)$, consisting of infinitely many repetitions of the same song, observed in a finite time. This process has a probability density distribution $p_1^F(f)$, which does not change in time. A Gaussian kernel $g(z)$ was used with $\sigma_z = \sqrt{5}$ Hz.

Fig. 16(a) shows that the energy converges to some probability density distribution (with negative sign) shown by the solid line. It automatically discovers the most probable frequencies as follows, figures in brackets showing the exact frequencies of the respective musical notes: 434Hz (440Hz) for A_4, 490Hz (493.88Hz) for B_4, and 388Hz (392Hz) for G_4. The classification process using attractors is shown in Fig. 17. From this we can see the evolution of the attractor and get some idea of the scope of the basins of attraction.

Secondly, we show how the system (4.8) can discover and memorize temporal patterns – in this case, musical phrases consisting of four beats. A 4-D landscape was used, and to each

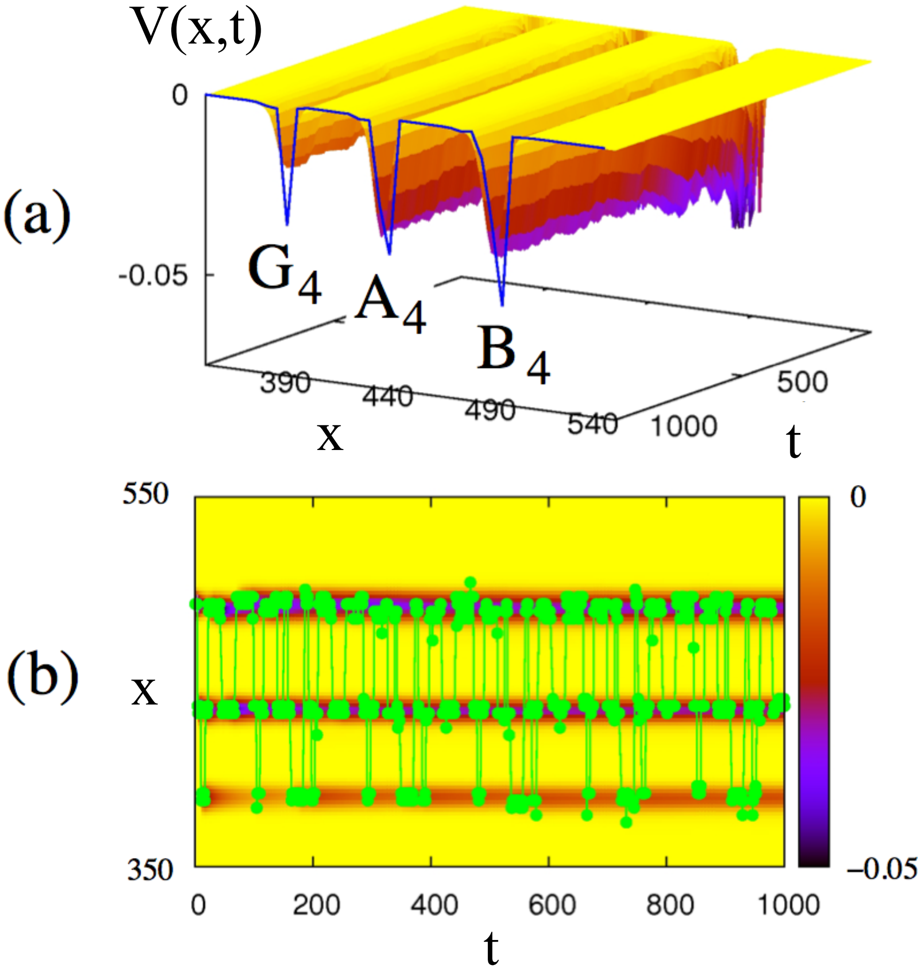


Figure 16: Musical note recognition. (a) Evolution of the energy landscape $V(x,t)$ in response to a musical signal. The local minima that are formed by the landscape are very close to the frequencies of the musical notes G_4, A_4 and B_4 that enter the song. (b) Green circles show the actual values of the input, and the colour of the background shows the depth of the energy function. System parameters given in text.

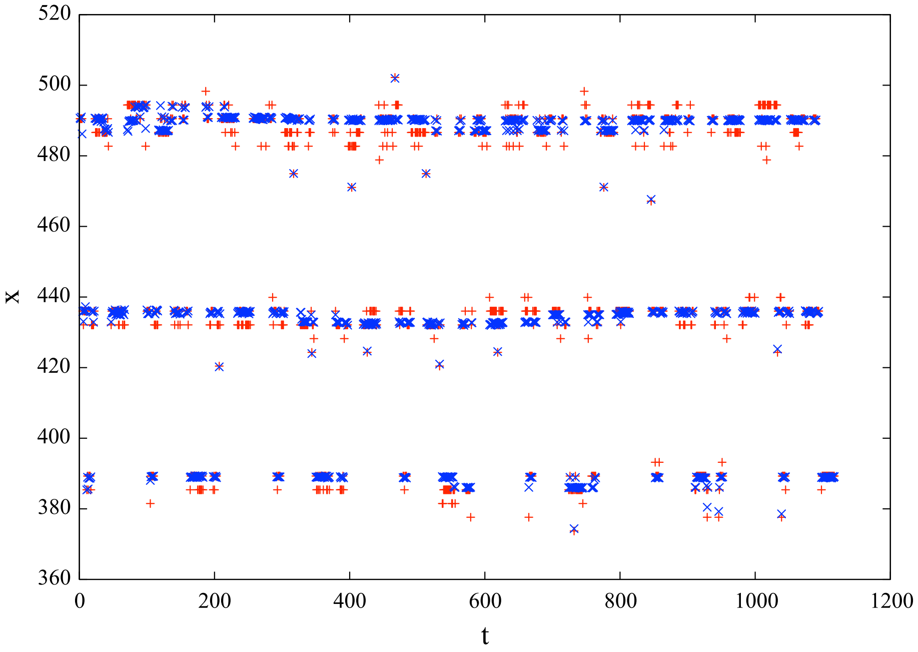


Figure 17: Classification of musical notes by evolution of the state to its local attractor. Initial condition (system input) is shown by a red +, and final state is shown by a blue X.

of its channels the same signal $f(t)$ was applied, but with a phase shift. Namely, at each time t the system (4.8) received a vector input $\psi(t) = (f(t), f(t+\tau), f(t+2\tau), f(t+3\tau))$, $\tau = 0.75$ sec. The procedure of creating a vector with the coordinates made of the delayed versions of the same signal is called delay embedding [83]. We regard $\psi(t)$ as a realization of a 4th-order stationary and ergodic vector random process $\Psi(t)$ (which we observe during finite time) with 4-dimensional probability density distribution $p_4^{\Psi}(f_1, f_2, f_3, f_4)$. We used a multivariate Gaussian kernel g with $\sigma_z = \sqrt{5}$ Hz in all of its four variables.

One cannot visualize evolution of a 4-D landscape in the same way as we did in Figs. 14-16, and we use an alternative representation. We take four half-axes and make their origins coincide (Fig. 18(a)). For each input $\psi = (f_1, f_2, f_3, f_4)$, we put 4 points with coordinates f_i on each of half-axes, and connect them by lines. Thus, the input pattern is represented by a polygon on a plane. The value of p_4^{Ψ} is represented in 2-D by the colour of the respective polygon - the polygon, whose colour is the darkest, is the most probable pattern (Fig. 18(b)). Unfortunately, when too many polygons overlap, it might be difficult to see them individually. But they can be found using the concept of a particle in the 4-D landscape,

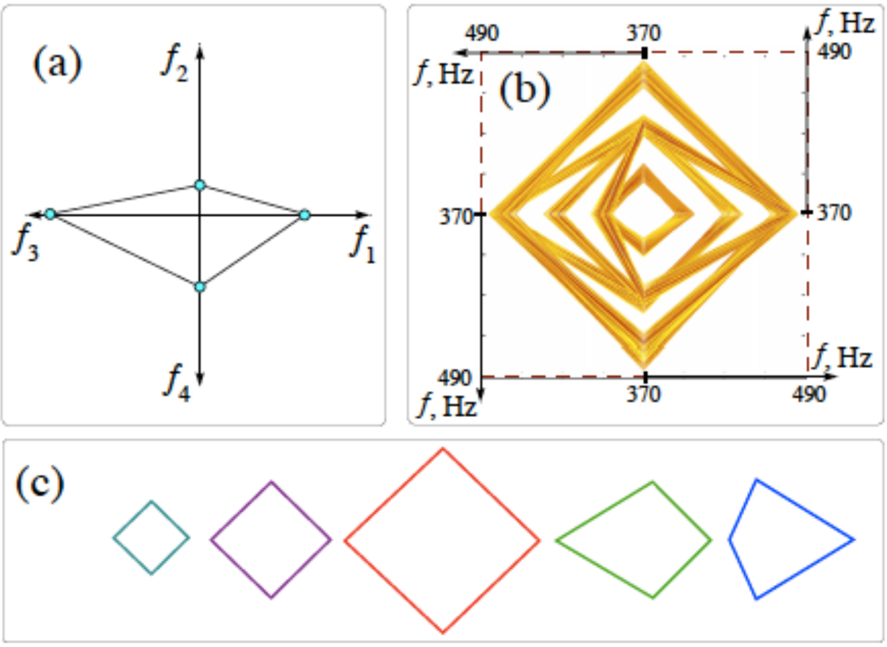


Figure 18: Automatic recognition of musical phrases. (a) Method of visualization for the 4-D landscape (b) A snapshot of the four-dimensional landscape V at the end of the recognition process. The darker colours represent the most probable musical patterns. (c) The most probable musical phrases automatically recognized by the system. System parameters given in text.

that will be attracted to one of the local fixed points representing one of the most probable patterns: five such patterns are given in smaller scale in Fig. 18(c).

Recognition of musical phrases is also illustrated by audio files [35] using the concept of a particle in the 4-D landscape. The frequency of the note will then be given by the location of the attractor. Using the locations of these attractors, one can recreate an approximation of the waveform, $y(t)$ using a simple additive synthesis

$$y(t) = \sum_t A_t sin(2\pi f n T), \tag{4.22}$$

with signal amplitude A_t, sample frequency T, frequency f and n the time step integer. The resulting file is however, subject to significant power loss, since only the peak corresponding to the fundamental frequency is used.

4.3 Comparison with artificial neural networks

We now explain how self-shaping dynamical systems are related to the two types of artificial neural networks: Hopfield neural networks, and probabilistic neural networks.

In Section 2.2, we have already explained Hopfield neural networks. Continuous state and time Hopfield neural networks use the same energy landscape principle for recognition, and if they were able to learn in an unsupervised and on-line manner, they would classify in a similar way to gradient self-shaping systems.

Gradient self-shaping systems also have one feature in common with another type of artificial neural networks, called *probabilistic neural networks* [80]. The purpose of these is to estimate the probability density distribution of the incoming patterns, which is then used to classify inputs, like in gradient self-shaping systems. These networks were developed in the attempt to overcome the spurious minima problem of the Hopfield neural networks.

As with all artificial neural networks, there is a collection of units with rigid architecture, and there are adjustable couplings between them. However, the architecture of the network is different to that of Hopfield neural networks. In particular, there is always a separate layer of neurons, such that each neuron codes a separate element of the training set. Thus, in order to take into account a new training pattern, one needs to physically add a new neuron to the system, thus making the whole system larger. This paradigm in fact accounts for the popular “grandmother neuron” hypothesis which, during the early ages of neuroscience, suggested that in the brain the memory about a certain object was coded by its own special neuron [19]. In practice this implies that only a finite number of training patterns can be used, which imposes a considerable restriction on the system’s performance. To remove the requirement of “one pattern – one neuron”, this technique was improved [5], but the general idea remained the same: the system needs to be expanded to learn better.

4.4 Discussion

Staying within the dynamical systems framework of Section 3, we have introduced a mathematical concept of a self-shaping dynamical system. We explained how such systems perform unsupervised learning and compare this mechanism with some learning techniques in artificial neural networks.

The self-shaping systems shape their velocity vector fields automatically under the influence of the external random stimulus. The resulting properties of the vector field, and consequently of the vector flow, are dictated by the probability density of the stimulus applied. We demonstrated how the simplest self-shaping systems (of gradient type) develop the fixed point attractors together with their basins of attraction. We showed that, for a stationary and ergodic input random process, the energy landscape of such gradient systems converges to a smoothed probability density distribution of the input signal. We illustrated the performance of gradient self-shaping systems using both random and musical data. Finally, we discussed how self-shaping systems are related to two types of artificial neural networks, and argued that they could serve the same purpose, but without their limitations.

What we present here is a mathematical proposal for the systems of a new class, however, the physical principles upon which such systems could be built are not obvious at the moment. Therefore, this concept represents an engineering challenge and requires the development of this new kind of device. Also, from a mathematical perspective, we predict that it will be possible to construct self-shaping systems that develop more complex attractors, such as limit cycles and chaotic attractors. Obviously, they would not be of a gradient type.

Advantages over Hopfield neural networks

The existing algorithms used for the adjustment of weights in Hopfield neural networks are quite good at developing the attractors (typically stable fixed points at the minima of the energy function) and of their basins of attraction in the right locations. However, these algorithms only enable one to control the vector field locally, and not how the whole of it changes, in response to the training input. This leads to the main problem of Hopfield neural networks – spurious minima. These are attractors corresponding to no valid class, which develop by themselves as the weights are adjusted. These minima affect pattern recognition, and this problem has still not been resolved after many years of effort.

Ideally, the energy landscape should possess local minima at the points, where the most probable class representatives appear, and have no other minima. A function that would perfectly satisfy this condition is a probability density distribution of all possible patterns, taken with a negative sign. Gradient self-shaping systems automatically produce the probability density distribution as the energy landscape, albeit, smoothed by a kernel function with a finite width. This ensures that, unlike in Hopfield neural networks, spurious minima do not occur in gradient self-shaping systems .

Advantages over probabilistic neural networks

The main problem with probabilistic neural networks is the requirement of the physical addition of new units for newly learnt patterns. Gradient self-shaping systems do not require the addition of these units to estimate the probability density distribution and can make use of as many patterns as needed, without any restrictions on their number.

Self-organization and self-shaping

A very important property of nonlinear systems (both natural and man-made) is their ability to self-organize. In terms of dynamical systems, self-organization is understood to be automatic shaping of the *solutions* that occur from a range of initial conditions. Here, there is a fixed structure of the vector field and/or of its perturbations. We now wish to extend the self-organization principle to the automatic shaping of the vector field itself. Perhaps most relevant to this idea, are living systems, that continuously modify themselves in response to their environment. Therefore, the suggested self-shaping approach might prove a helpful paradigm when modelling adaptation and development in living systems in general.

5 Gradient Systems with Delay

5.1 Motivation

In the previous section, we considered gradient self shaping systems assuming that the input to such systems is a stationary random signal. In the long term, the vector field converges to some stationary vector field with attractors in the form of fixed points. Mathematically it is interesting to find how a self shaping system could develop non-trivial objects in the phase space, such as periodic orbits or chaotic attractors. We know that in nonlinear dynamical systems, the introduction of a delay can result in the appearance of non-trivial attractors. In this section, we wish to explore the possibility of using a delayed vector field to create such non-trivial attractors in self shaping systems in the long term.

For simplicity, we consider self shaping systems in the long term assuming that their vector field has achieved some stationary shape. In this case we can model the energy landscape of such a system by a polynomial. Unlike most literature on delay systems, we do not simply introduce an extra delay term to the vector field, but instead delay the whole vector field of the system by the same amount. This problem should be relevant to the problem of optimization in which the energy landscape models a certain cost function depending on a number of adjustable parameters of the system. One needs to find the global minimum of the cost function and avoid being trapped in the local minima.

In terms of practical implementation, a delayed vector field in a gradient system will describe the well known latency problem. Therefore investigation of the gradient system with delayed vector field should shed light onto the influence of the latency in real electronic devices performing optimization.

5.2 Introduction

We have introduced the concept of gradient self-shaping systems. Gradient systems are also found in many optimization problems such as cost functions (see e.g. [1]). A gradient system in $\mathbb{R}^n$ is a system of differential equations of the form

$$\dot{x} = -\nabla V(x), \tag{5.1}$$

defined by the gradient of V where $V : \mathbb{R}^n \to \mathbb{R}$ is a C^∞ function (differentiable for all degrees of differentiation) and [29]

$$\nabla V = \left(\frac{\partial V}{\partial x_1}, \ldots, \frac{\partial V}{\partial x_n} \right). \tag{5.2}$$

From the definition of a gradient system it follows that the critical points of V are the equilibrium points of the system [29]. If a critical point is an isolated minimum of V then this point is an asymptotically stable equilibrium point, and there does not exist periodic solutions of x.

A gradient system exhibits regular, predictable behaviour. The potential function V of the system will consist of a smooth, continuous, differentiable function containing some maxima and minima. Its vector field (the gradient system) will also be smooth, continuous and differentiable with equilibrium points corresponding to the system's extrema. The flow of the vector field will always be toward the equilibria corresponding to the minima (stable equilibria) and away from the equilibria corresponding to the maxima (unstable equilibria). There will be no flow at the equilibrium points.

For example, consider a bistable system given by

$$V(x) = -\frac{ax^2}{2} + \frac{bx^4}{4}. \tag{5.3}$$

The vector field will be given by

$$f(x) = -\nabla V = -\frac{dV(x)}{dx} = ax - bx^3. \tag{5.4}$$

This system is depicted in Fig. 19. It has three equilibrium points: $x = 0$ and $x = \pm\sqrt{a/b}$, with solid black circles representing a stable equilibrium and an outlined black circle representing an unstable equilibrium. The arrows on the horizontal axis indicate the direction of the flow for each interval between equilibria: the arrows point to the right if $f(x) > 0$ and to the left if $f(x) < 0$. This shows that the trajectory of the system will

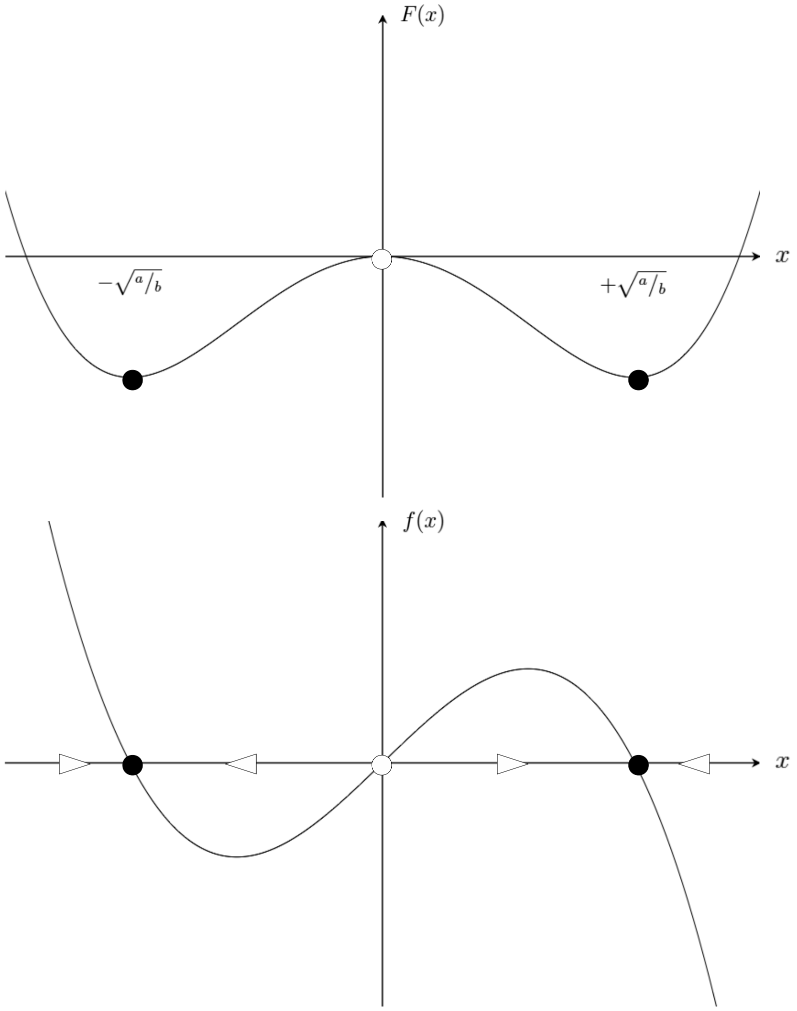


Figure 19: Top: Potential energy of the bistable system (5.3); Bottom: Vector field of the bistable system (5.4), arrows indicate the direction of the flow. Solid circles represent stable steady states, outline circles represent the unstable state.

approach the stable fixed point at $x = +\sqrt{a/b}$ with any initial condition $x_0 > 0$, and will approach the stable fixed point $x = -\sqrt{a/b}$ with any initial condition $x_0 < 0$.

Since the equilibria in gradient systems can only be fixed points, one must alter the structure of the system to allow a different behaviour. Simple nonlinear dynamical systems can exhibit complex behaviour called chaos with features typical to random oscillations. Chaos theory studies the behaviour of dynamical systems that are highly sensitive to initial conditions. Small changes to initial conditions lead to widely diverging outcomes, making prediction over a large period of time generally impossible. Instead, one must investigate the long term behaviour with the focus not on finding the exact solutions to the equations, but on qualitative features such as locating possible attractors and steady states of the system, or studying the dependence on initial conditions and parameter values. Chaos occurs even though these systems are deterministic (dynamics are fully determined by their initial conditions) with no stochastic (random) quantities involved.

Discrete systems can exhibit chaos in any dimension, e.g. *logistic map.* However, the *Poincaré-Bendixson theorem* shows that chaos cannot arise in a continuous dynamical system if it has less than three dimensions. Furthermore, linear systems are never chaotic, and so for a dynamical system to demonstrate chaotic behaviour it has to be nonlinear and have dimension greater than two if continuous, or be discrete.

We achieve this by introducing a time delay into the system. The time delay changes the system from an ordinary differential equation (ODE) to a delay differential equation (DDE). DDEs are an important class of dynamical systems that arise in models of situations in which the present state of the system has an explicit dependence on the state at a past time. They appear is many real world systems, for example, consider a technological control problem: here a controller monitors the state of a system and adjusts it based on its observations, however, a delay emerges between the observation and the control reaction since they can never occur simultaneously. Other applications include neural systems [47], epidemiology [10], and nonlinear optics [18], to name just a few.

There are different types of DDEs. We shall focus on the simplest: autonomous, evolutionary delay equations with a single constant time delay, τ. Other types include systems with multiple constant delays, state dependent delays, or distributed delays. Comprehensive and rigorous properties of such DDEs and their solutions can be found, e.g., in [23]. We will only mention the key features required for our work.

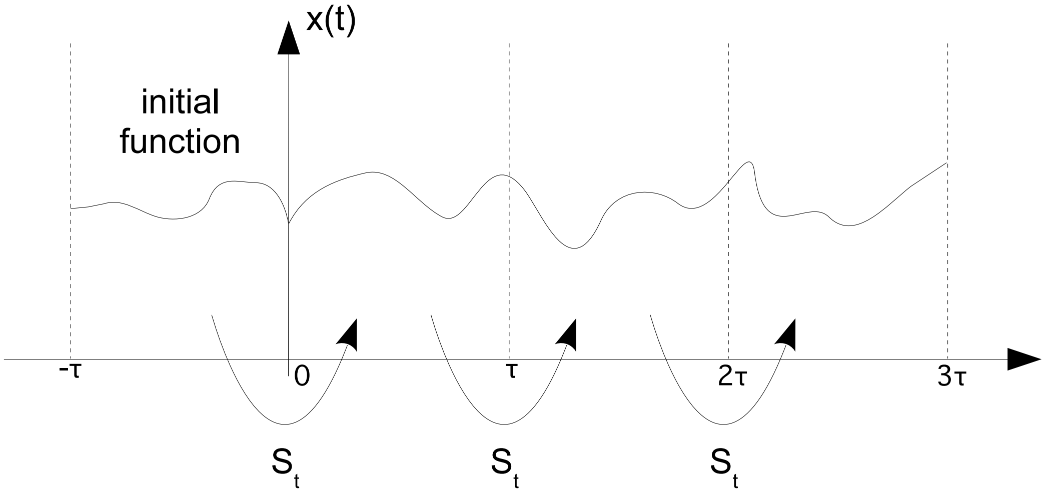


Figure 20: The action of the evolution operator S_t for a DDE. It takes a function defined over a time interval τ and maps it to another interval of the same length, e.g. the interval $[-\tau, 0]$ is mapped to $[0, \tau]$.

When describing ODEs of finite dimension, we need only specify an initial value for each state variable x_i. When describing DDEs, the specification of initial conditions is not so simple. In order to solve a DDE, at each time step we have to gather information from earlier values of $\mathbf{x}$ and so we must specify an initial function that gives the behaviour of the system prior to the initial time moment otherwise the right-hand side will not be defined for some t. Clearly this function must cover a period at least as long as the (largest) delay. So for our single constant delay system the initial function will be a function $\mathbf{x}(t)$ defined on the interval $[-\tau, 0]$. Expressed in another way, we need to define infinitely many values of x lying in the interval $t = -\tau$ to $t = 0$. So the state of a DDE is determined by the values of a function on some interval and thus DDEs become infinite dimensional problems. This feature of DDEs means that, as long as there is a nonlinearity, in such a system there is the possibility of periodic and chaotic behaviour.

To understand the dynamics of a DDE system one can think of the solution as a mapping (through an *evolution operator*) from functions on the interval $[t - \tau, t]$ into functions on the interval $[t, t + \tau]$. That is, the dynamical system can be thought of as a sequence of functions defined over an adjoining set of time intervals of length τ [70], illustrated by Fig. 20.

Most DDEs are not analytically solvable and so one must resort to numerical methods. When doing so, one has to be careful since at the point where the function determining the initial conditions meets the “actual time domain” solution there is likely to be a discontinuity in the first derivative unless a very careful initial function is chosen. These discontinuities propagate: if the first derivative is discontinuous at $t = 0$, then the second derivative will be discontinuous at $t = \tau$ since $\ddot{x}$ is related by the DDE to $\dot{x}(t - \tau)$, and so on [70]. For this reason one must be careful about the integration step size and method chosen. Furthermore, as with ODEs, we can find much information about the system from stability analysis of the equilibria.

In this work we will show the impact of a time delay on gradient systems, i.e. when the whole velocity vector field is delayed. First of all we will give a brief overview of delay differential equations, then we will demonstrate the effect of a fully delayed gradient system for a simple symmetric energy landscape, a simple non-symmetric energy landscape and a more complicated non-symmetric energy landscape.

5.3 Effect of delay on gradient systems

Here we shall consider the effect of a time delay on only gradient systems. In particular, we shall study systems where the whole vector field is delayed by the same constant τ,

$$\dot{\mathbf{x}} = f(\mathbf{x}(t-\tau)), \tag{5.5}$$

where f is a polynomial meeting the required properties of a gradient system. Consider, for example, the vector field produced by a delayed 1-dimensional quartic potential,

$$\dot{x} = x(t-\tau) - [x(t-\tau)]^3. \tag{5.6}$$

Equation (5.6) has been studied in [21, 85, 86, 81, 46, 62, 45], however, to the best of our knowledge, a generalization to other gradient systems has not been made. This approach of delaying the whole vector field differs from the typical approach when including a delay in such systems, done through a feedback term K, e.g.,

$$\dot{x} = x(t) - [x(t)]^3 + K(x(t), x(t-\tau)). \tag{5.7}$$

For delay equations like (5.5), monotonicity of the function f, generically implies the invertibility of solutions and one does not typically expect to find chaotic solutions such equations. In this regard, Mallet-Paret and Sell have shown that if the nonlinearity of f is monotone then the dynamics are very simple and non-chaotic [51, 50]. In contrast, the Mackey-Glass equation [48], where the nonlinearity is not monotonic, does have chaotic solutions for appropriate parameter values. We are investigating gradient systems, where nonlinearity of f is non-monotonic and shall see if this trend continues.

5.3.1 Local behaviour

To investigate the local behaviour near to the fixed points we shall perform linear stability analysis of the solutions. The equilibrium points $\mathbf{x}^*$ satisfy

$$f(\mathbf{x}^*) = 0. \tag{5.8}$$

When performing linear stability analysis of ODEs we assume that the system has been slightly perturbed away from the equilibrium. We proceed in a similar way with DDEs except that since we have an infinite dimensional function space, perturbations are now time dependent functions $\delta\mathbf{x}(t)$ over an interval of at least the delay. So,

$$\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x}, \tag{5.9}$$

and,

$$\dot{\mathbf{x}} = \dot{\delta\mathbf{x}} = f(\mathbf{x}^* + \delta\mathbf{x}_\tau), \tag{5.10}$$

where $\mathbf{x}_\tau$ denotes $\mathbf{x}(t-\tau)$.

Since $\delta\mathbf{x}_\tau$ is small, we can linearise the differential equation using a Taylor series,

$$\dot{\delta\mathbf{x}} \approx f(\mathbf{x}^*) + \mathbf{J}\delta\mathbf{x}_\tau, \tag{5.11}$$

where $\mathbf{J}$ is the Jacobian with respect to $\mathbf{x}_\tau$ evaluated at $\mathbf{x} = \mathbf{x}^*$. From (5.8) $f(\mathbf{x}^*) = 0$ so,

$$\dot{\delta\mathbf{x}} \approx \mathbf{J}\delta\mathbf{x}_\tau. \tag{5.12}$$

Suppose that the linear DDE (5.12) has exponential solutions, then

$$\delta\mathbf{x}(t) = Ae^{\lambda t}, \tag{5.13}$$

which on substitution back into (5.12) gives

$$\lambda A = \mathbf{J}Ae^{-\lambda\tau}. \tag{5.14}$$

Linear algebra theory tells us that this can only be true for non-trivial A if

$$|\mathbf{J}e^{-\lambda\tau} - \lambda\mathbf{I}| = 0, \tag{5.15}$$

where $\mathbf{I}$ is the identity matrix.

Equation (5.15) is called the characteristic equation of the equilibrium point. It appears similar to the traditional eigenvalue problem apart from the exponential terms. These terms

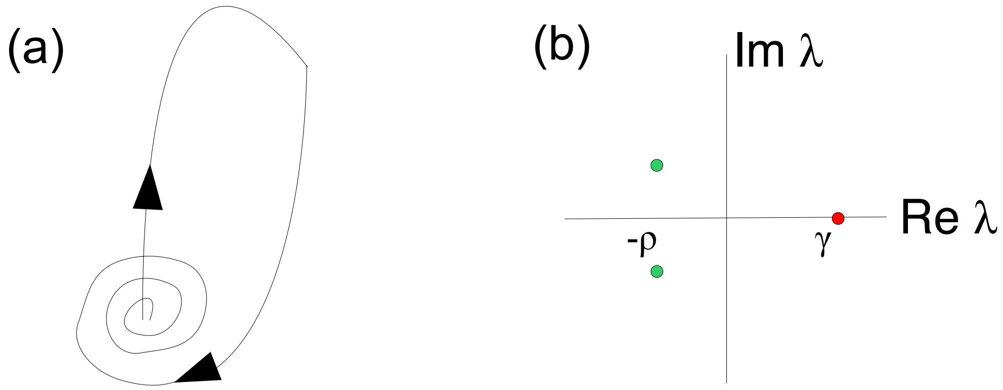


Figure 21: (a) A 3-D saddle-focus equilibrium at the point of homoclinicity. (b) Eigenvalues λ of a saddle-focus equilibrium $\lambda_{1,2} = -\rho \pm i\omega$, $\lambda_3 = \gamma > 0$.

mean that when the determinant is expanded, we have a quasi-polynomial which has an infinite number of complex roots [70]. As with linear stability analysis in ODEs, if any of the solutions of the characteristic equation have positive real parts then the equilibrium is unstable, if they have negative real parts then the equilibrium is stable, and if the leading characteristic values are zero then the stability cannot be deduced to linear order.

Let us first consider a 1-dimensional gradient system with delay, so the Jacobian becomes a constant J. Then the characteristic equation becomes

$$\lambda - Je^{-\lambda\tau} = 0. \tag{5.16}$$

Using (5.16), the equilibria of the maxima of the potential will always be unstable, since in this case $J \geq 0$, which means the solution of the characteristic equation will always have a positive real λ. In fact, these are *saddle-focus* equilibria (Fig. 21) (for any positive τ) since they have one positive real eigenvalue and infinite pairs of complex conjugate eigenvalues ($\frac{(n-1)}{2}$ for an n-dimensional system). From the stable manifold theorem, we know that the positive eigenvalue implies the existence of a 1-dimensional unstable manifold on which phase trajectories depart from the equilibrium. Likewise, the complex pairs of eigenvalues with negative real part imply that the origin has associated with it an infinite-dimensional stable manifold ($(n-1)$ for an n-dimensional system) in which solutions spiral asymptotically to the origin.

The solution of (5.16) is given by [14]

$$\lambda = \frac{W_k(J\tau)}{\tau} \tag{5.17}$$

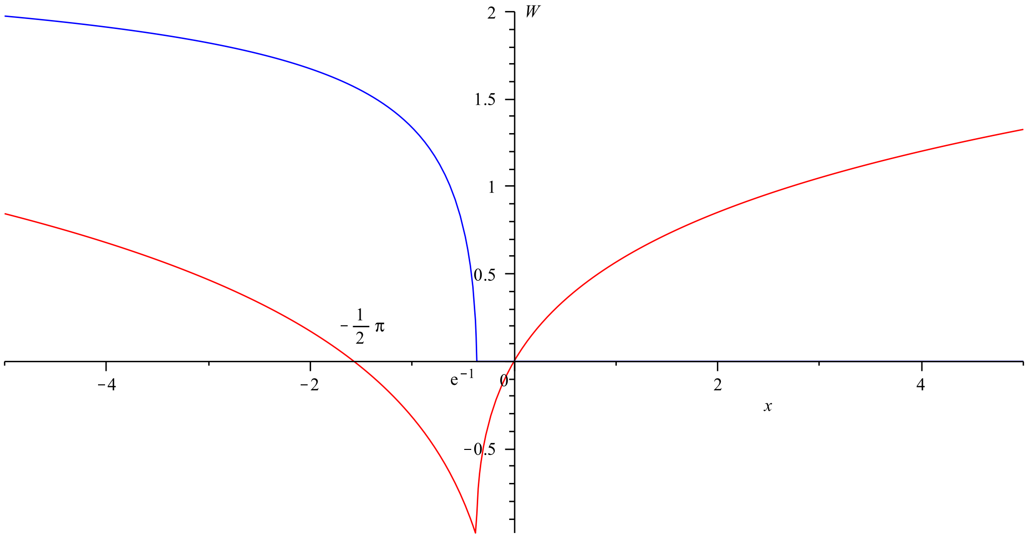


Figure 22: Real (red) and imaginary (blue) parts of the principal branch of the Lambert W function.

where W_k is the k-th branch of the Lambert W function. The Lambert W function (or product log function) is defined as the solutions $w \in \mathbb{C}$ of

$$we^w = z, \tag{5.18}$$

for $z \in \mathbb{C}$ and denoted W, i.e. $w = W(z)$.

The equation (5.18) has infinitely many solutions, that is the Lambert W function is an infinitely many valued function, in other words it has an infinite number of branches W_k, $k = 0, \pm 1, \pm 2, \ldots \pm \infty$, with W_0 called the principal branch. The range of the Lambert W function is symmetric with respect to the real axis, with the characteristic root $W_0(x) = s$ is always the rightmost among all the characteristic roots $s = W_k(x)$ [78].

At the minima of the potential, we have $J < 0$ and clearly all $\tau \geq 0$, and so $J\tau \leq 0$. From Fig. 22, the real part of W_0 is negative for $J\tau \in [-\frac{\pi}{2}, 0]$ and positive for $J\tau \in (-\infty, -\frac{\pi}{2})$. The imaginary part will be non-zero for $J\tau < -e^{-1}$. So for $\tau < \frac{-e^{-1}}{J}$, we will have purely negative real eigenvalues, i.e. the equilibrium will be a stable node. When $J\tau = -e^{-1}$, the equilibrium will change from a stable node to a stable focus, i.e. has all negative complex conjugate eigenvalues.

There are two criteria for the occurrence of an Andronov-Hopf bifurcation. Firstly, the real part of $W_0(J\tau)$ must cross zero as the fixed point loses stability. From Fig. 22, we can see this will occur when $J\tau = -\frac{\pi}{2}$. To verify this we need to show when (5.16) has purely imaginary roots, i.e. $\lambda = \pm\beta i$. Substituting this into (5.16) gives

$$\beta i - Je^{-\beta i\tau} = 0. \tag{5.19}$$

This can be written as

$$\beta i - J[(\cos(\beta\tau) - i\sin(\beta\tau)] = 0. \tag{5.20}$$

Separating real and imaginary parts gives

$$\begin{aligned} \cos(\beta\tau) &= 0, && (5.21) \\ \beta + Jsin(\beta\tau) &= 0. && (5.22) \end{aligned}$$

Equation (5.21) has infinitely many solutions in the form of conjugate pairs,

$$\beta = \frac{\left(\frac{\pi}{2} \pm n\pi\right)}{\tau}, \qquad n = 0, 1, 2, \ldots, \tag{5.23}$$

which we then substitute into (5.22) to give,

$$\tau = -\frac{\left(\frac{\pi}{2} + n\pi\right)}{J(-1)^n}, \qquad n = 0, 1, 2, \ldots, \tag{5.24}$$

which will have the same τ for both components of the conjugate pair. Since τ must be positive, $J < 0$ then n must be *even*. The Andronov-Hopf bifurcation will initially occur for the smallest such value of n, and so we choose $n = 0$ to get

$$\tau = -\frac{\pi}{2J} \tag{5.25}$$

as expected. Further complex conjugate pairs will cross the imaginary axis for subsequent even values of n in (5.24).

The second criterion to be satisfied for an Andronov-Hopf bifurcation to occur is [23]

$$\left.\frac{dRe(\lambda)}{d\tau}\right|_{\tau=\tau_k} > 0. \tag{5.26}$$

To verify this, differentiating (5.16) gives

$$\frac{d\lambda}{d\tau} = -Je^{-\lambda\tau}\left(\lambda + \tau\frac{d\lambda}{d\tau}\right). \tag{5.27}$$

Let $\lambda = \alpha + i\omega$ and separating real and imaginary parts gives

$$\begin{aligned} \frac{d\alpha}{d\tau} &= -Je^{-\alpha\tau}cos(\omega\tau)\left(\alpha + \tau\frac{d\alpha}{d\tau}\right) - Je^{-\alpha\tau}sin(\omega\tau)\left(\omega + \tau\frac{d\omega}{d\tau}\right), \\ \frac{d\omega}{d\tau} &= -Je^{-\alpha\tau}cos(\omega\tau)\left(\omega + \tau\frac{d\omega}{d\tau}\right) + Je^{-\alpha\tau}sin(\omega\tau)\left(\alpha + \tau\frac{d\alpha}{d\tau}\right). && (5.28) \end{aligned}$$

In equation (5.25), we have calculated that $\tau_k = -\frac{\pi}{2J}$. So we need to verify that

$$\left.\frac{d\alpha}{d\tau}\right|_{\tau=-\frac{\pi}{2J}} > 0. \tag{5.29}$$

At $\tau_k = -\frac{\pi}{2J}$,

$$\begin{aligned} \alpha &= 0, \\ \omega &= \frac{\pi}{2\tau_k} = -J. \end{aligned} \tag{5.30}$$

Then substituting these values into (5.28) gives

$$\begin{aligned} \frac{d\alpha}{d\tau} &= J^2 + \frac{\pi}{2}\frac{d\omega}{d\tau}, \\ \frac{d\omega}{d\tau} &= -\frac{\pi}{2}\frac{d\alpha}{d\tau}. \end{aligned} \tag{5.31}$$

Combining these equations gives

$$\frac{d\alpha}{d\tau} = \frac{4J^2}{(4+\pi^2)} > 0. \tag{5.32}$$

So we have met the criteria for the occurrence of an Andronov-Hopf bifurcation, in which the fixed point loses stability. In the numerical simulations performed here, the bifurcation is always supercritical and leads to the birth of a stable limit cycle.

After a finite-dimensional supercritical Andronov-Hopf bifurcation, the size of the stable limit cycle grows proportional to the square root of the distance of the parameter from the bifurcation value [42]. For an infinite dimensional system, the centre manifold theorem enables us to reduce the system to a finite dimensional one. Consequently, results in bifurcations of ODEs can be applied to DDEs [13]. We therefore assume that in gradient systems with delay, the periodic orbit will also grow in a similar style. The growth of the periodic orbit means that it can collide with nearby unstable saddle-focus equilibria in a global bifurcation.

Equation (5.24) can also be used to show the values of τ for which the complex conjugate pairs of the saddle-foci cross the imaginary axis. However, here there will be no change in stability. In this case $J \geq 0$ and so n must be *odd*. Again choosing the smallest value of n gives

$$\tau = \frac{3\pi}{2J}. \tag{5.33}$$

Further complex conjugate pairs will cross the imaginary axis for odd values of n in (5.24). It should be mentioned that, despite there being no change in stability, as these complex conjugate pairs of eigenvalues cross the imaginary axis, there will be changes in the manifolds of the fixed point. These changes are described in Appendix C, however they will not bring any new dynamical behaviour to the system.

This theory can be extended to gradient systems with an n-dimensional vector x. The approach will be similar, however, a $n \times n$ Jacobian matrix will be involved, resulting in an order-n quasi-polynomial for the characteristic equation which can be analysed to deduce the relevant bifurcation.

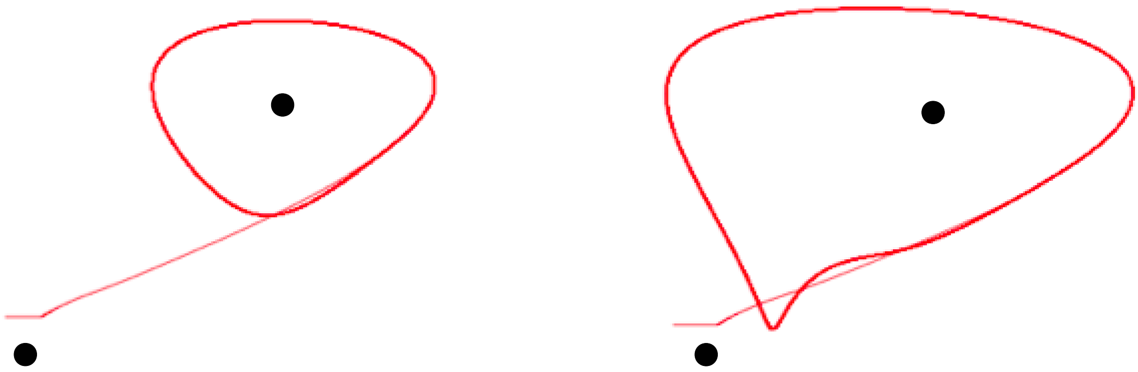


Figure 23: Distortion of the shape of the periodic orbit near homoclinicity. Black circles denote equilibria – the equilibrium outside the periodic orbit is the saddle-focus; the equilibrium inside the periodic orbit is the originally stable fixed point (now unstable focus).

5.3.2 Global behaviour

When the local qualitative properties of an object change, a local bifurcation occurs. However, local bifurcations are not the only type of bifurcation that can occur. Global bifurcations occur in the larger neighbourhood of an object. Linear stability analysis is no longer useful however, since we only detect the properties in the immediate vicinity of the object, not in its larger neighbourhood. Instead we investigate the behaviour using intuition and numerical simulation. We have seen that as we increase the delay, the periodic orbit grows. During this process it will grow closer and closer to the saddle-focus equilibrium. As the cycle gets close to the saddle-focus and therefore its manifolds, it stretches along the manifolds. This results in the distortion of its shape formed by the manifolds near the saddle-focus (Fig. 23). Furthermore, the motion on the part closer to the saddle-focus slows down, meaning the period of oscillations increases. As the cycle grows further it hits the saddle-focus, with the period of oscillations tending to infinity at the point of bifurcation (Fig. 24). Here the manifolds of the saddle-focus collect the cycle and close on it (Fig. 25). This forms what is called a homoclinic loop at a homoclinic bifurcation.

Contrary to what happens at a local bifurcation, nothing has happened to the fixed point here. It has not changed stability, nor has it been destroyed: however, the periodic orbit has been destroyed despite its local properties remaining unchanged. All that has happened is that another object within the phase space has fallen in the vicinity of the cycle.

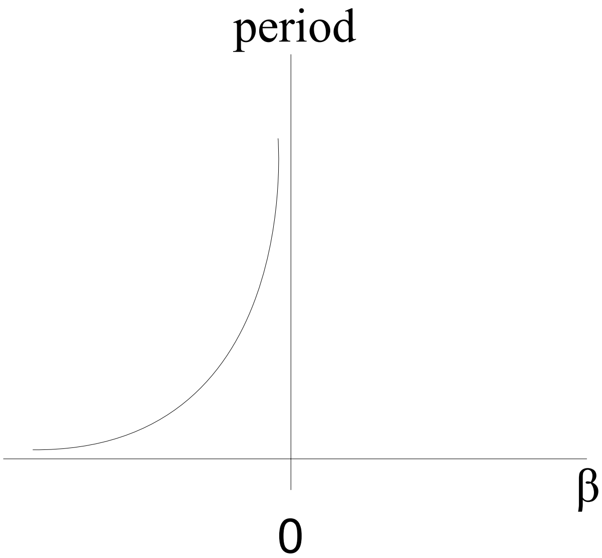


Figure 24: Growth of the period of the orbit as it approaches homoclinicity at $\beta = 0$ (for negative saddle quantity, $\sigma < 0$). The period tends to infinity as the parameter approaches the value of the homoclinic bifurcation.

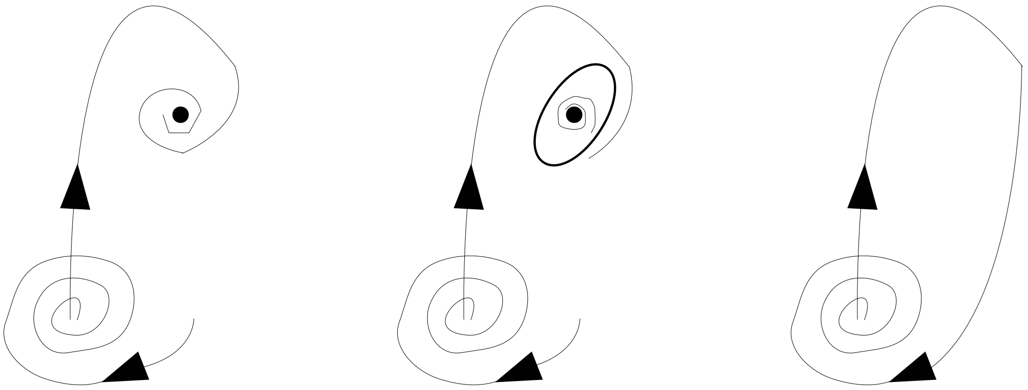


Figure 25: Stages in the homoclinic bifurcation of a saddle-focus.
Left: The unstable saddle-focus and the stable focus.
Middle: The unstable saddle-focus and the stable periodic orbit.
Right: Homoclinic orbit of the saddle-focus.

The stable manifold of the saddle-focus forms the boundary of the basin of attraction of the periodic orbit. By hitting the fixed point, the cycle has in fact touched the boundary of its own basin of attraction – known as a boundary crisis. This is called a global bifurcation, which causes changes in the shape of trajectories (even far away from where the homoclinic orbit appears in the phase space). As is the case for any global bifurcation, the normal form does not exist for the homoclinic bifurcation.

During the 1960s, Leonid Shilnikov performed extensive studies of the dynamics near these orbits (summarized in [77]). He defined a value called the saddle quantity, σ, which dictated the dynamics of the system. The saddle quantity is the sum of the real eigenvalue (λ_1) and the real part of the leading complex-conjugate eigenvalues ($\lambda_{2,3}$), i.e. $\sigma = \lambda_1 + Re(\lambda_{2,3})$. The leading complex-conjugate eigenvalue is that which is closest to the imaginary axis. Shilnikov showed [42] that (i) if $\sigma < 0$ then dynamics will be "simple", i.e. there will be a stable periodic orbit *in the neighbourhood* of the homoclinic orbit which will be destroyed by the bifurcation (or in the reverse direction, a periodic orbit will be created), and (ii) if $\sigma > 0$ then dynamics will be "complex". Case (ii) immediately implies that there will be infinitely many periodic orbits formed in each neighbourhood of the homoclinic orbit – an indication of chaos. In case (i), as the parameter increases further, past the point of homoclinicity, the periodic orbit has been destroyed and the unstable manifold of the saddle-focus collects any object around the stable manifold. Subsequent behaviour will be dictated by any other objects in the nearby phase space [42] (Fig. 26).

Gradient systems will have multiple attractors around the unstable equilibria. Even in case (i), more complex situations will occur when there is more than one different homoclinic bifurcation to the same saddle-focus. As explained, Shilnikov showed that the saddle quantity is critical in determining what happens after the homoclinic bifurcation. We can predict when this can happen by studying the characteristic equation (5.16). This has an infinite number of roots given by

$$\lambda = \frac{W_k(J\tau_h)}{\tau_h}, \tag{5.34}$$

where $W_k(x)$ is the kth branch Lambert W function evaluated at x, and τ_h is the value of the delay at which the homoclinic bifurcation occurs. Recall that $\sigma = \lambda_1 + Re(\lambda_{2,3})$. Since the saddle quantities are governed by the real parts of the leading eigenvalues we only need

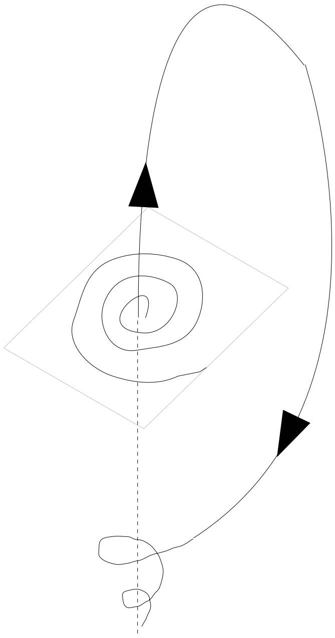


Figure 26: The stable and unstable manifolds of a 3-D saddle-focus post homoclinic bifurcation with negative saddle quantity ($\sigma < 0$).

to consider

$$\begin{aligned} \lambda_1 &= \frac{W_0(J\tau_h)}{\tau_h}, \\ Re\lambda_{2,3} &= Re\frac{W_1(J\tau_h)}{\tau_h}. \end{aligned} \tag{5.35}$$

So the critical value of σ occurs when $\lambda_1 = -Re\lambda_{2,3}$, i.e.

$$W_0(J\tau_h) = -ReW_1(J\tau_h), \tag{5.36}$$

which can be solved numerically (Fig. 27),

$$J\tau_h \approx 1.975. \tag{5.37}$$

So that if $J\tau_h > 1.975$ then $\sigma > 0$, and we will have the Shilnikov route to chaos, whereas if $J\tau_h < 1.975$ then $\sigma < 0$, and the dynamics will remain simple. In the short term, this means that the dynamics here will be dictated by neighbouring objects in the phase space (e.g. attracted towards another attractor). The long term dynamics are not intuitively obvious and will be studied numerically.

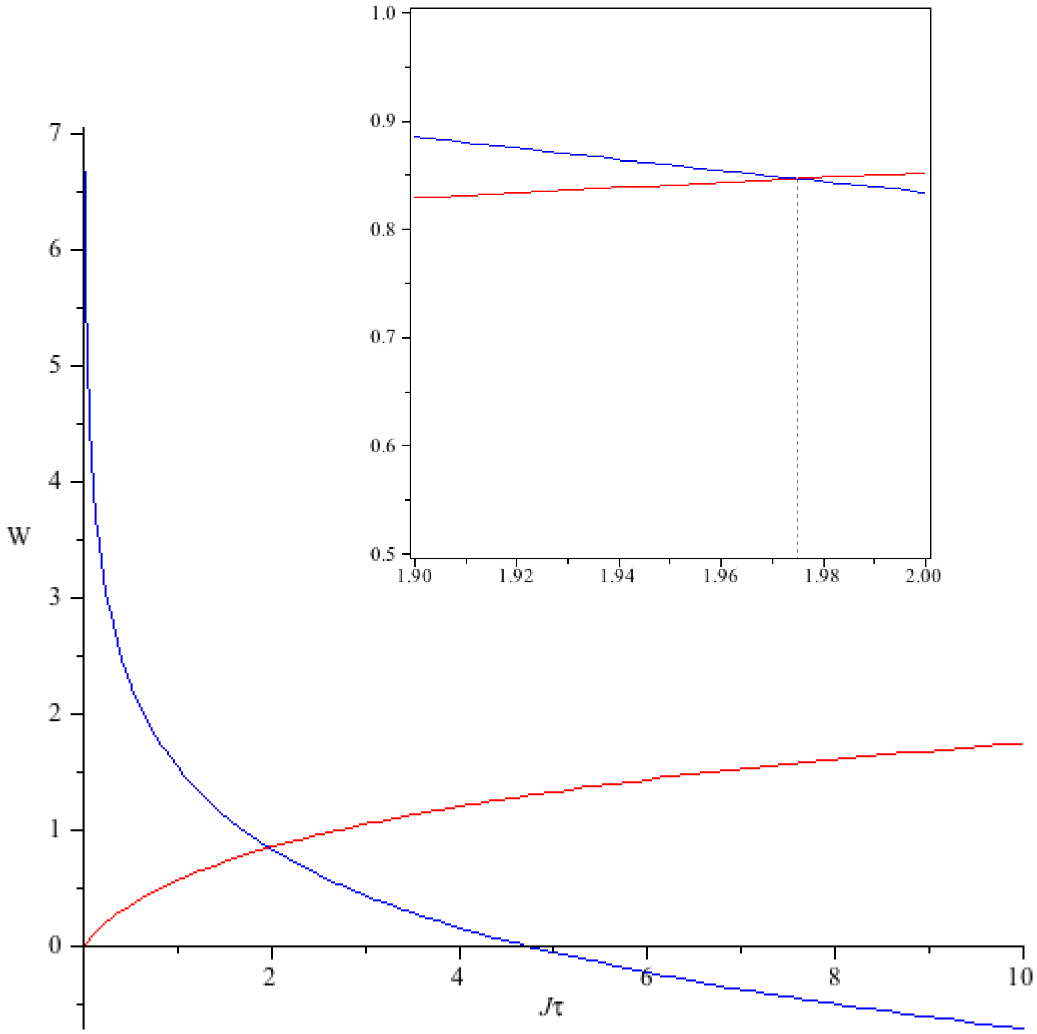


Figure 27: The critical value of the saddle quantity occurring when $\sigma = 0$, i.e. the intersection of the principal branch (red) of the Lambert W function with the negative of the real part of branch 1 (blue).

In higher order polynomial systems there is clearly room for several of these bifurcations to occur at different delays. To verify this behaviour, and to investigate numerically what happens when there is more than one homoclinic bifurcation to the same equilibrium, we consider some examples. In these examples, we perform numerical simulation using a 4th-order Runge-Kutta scheme with integration step 0.001. To enable the scheme to incorporate delay, interpolation must be used to calculate the history function [74].

5.4 Application to a quartic potential with delay

We first shall consider two cases of a double-well potential described by a fourth order polynomial: a symmetric potential, and a non-symmetric potential. One can depict the evolution of a dynamical system by considering a "particle", which moves along a path called its phase trajectory according to its vector field.

Symmetric potential

Recall equation (5.3), the delayed gradient system of such a potential is given by

$$\dot{x} = ax(t-\tau) - b[x(t-\tau)]^3. \tag{5.38}$$

The dynamics of such a system have been studied in [21, 85, 86, 81]. For simplicity we consider the case $a = b = 1$. Using (5.25) the two equilibria corresponding to the minima of the potential ($x = \pm 1$) will be stable fixed points for $\tau < \frac{\pi}{4}$. At $\tau = \frac{\pi}{4}$ a complex conjugate pair of eigenvalues will cross the imaginary axis, and thus the fixed points undergo an Andronov-Hopf bifurcation. The points are then unstable for $\tau > \frac{\pi}{4}$. As mentioned previously, the saddle-focus fixed point ($x = 0$) will always be unstable.

For consistency, we choose constant initial conditions within the right hand well (although by symmetry this is irrelevant), which are selected so as to avoid unbounded solutions (discussed later). For small delays, $\tau \leq 0.78$, the particle relaxes to the stable fixed point at $x = 1$, in agreement with the linear stability analysis. At $\tau \approx 0.79$ the fixed point loses stability as it undergoes an Andronov-Hopf bifurcation. This bifurcation means that a periodic orbit is formed within the basin of attraction of the now unstable equilibrium. For $0.79 < \tau < 1.32$ the particle oscillates in a periodic orbit, with the period growing as τ increases. The evolution of the system as the delay increases is shown in Fig. 28. To visualize phase space trajectories for DDEs, we project the trajectories in the infinite dimensional phase space into the 2-dimensional $(x, x(t-\tau))$-plane. From the second and third rows in Fig. 28 we see the distortion of the shape of the periodic orbit as its growth takes it near to the saddle-focus.

At $\tau \approx 1.32$ the periodic orbit meets the saddle-focus (at $x = 0$) and undergoes a saddle-focus homoclinic bifurcation. At exactly the same time, due to the symmetry in the problem, the orbit around the other equilibrium undergoes a homoclinic orbit with the saddle-focus. The saddle quantity here is negative and the outward breakdown of both homoclinic loops forms one large stable symmetric periodic orbit covering all the equilibria.

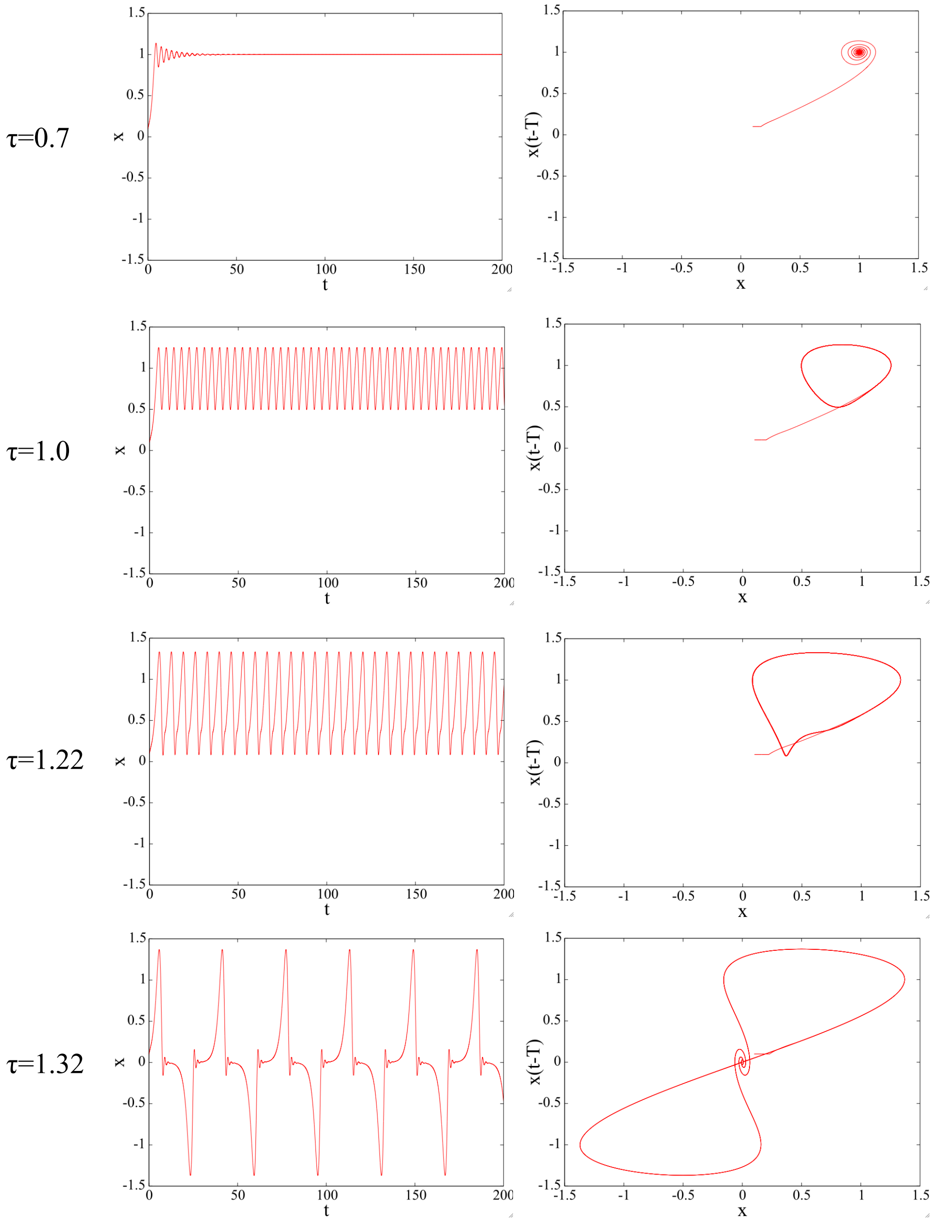


Figure 28: Plots of the realization of $x(t)$ vs t (first column) and the phase portrait $x(t-\tau)$ vs $x(t)$ (second column) as the parameter τ is varied for equation (5.38) with $a = b = 1$. The initial condition is set in the right hand well.

$\tau = 0.7$ - Stable focus.

$\tau = 1.0$ - Periodic orbit.

$\tau = 1.22$ - Periodic orbit deforms as approaches unstable saddle-focus equilibrium.

$\tau = 1.32$ - Homoclinic bifurcation leading to a large symmetric orbit covering both wells of the potential.

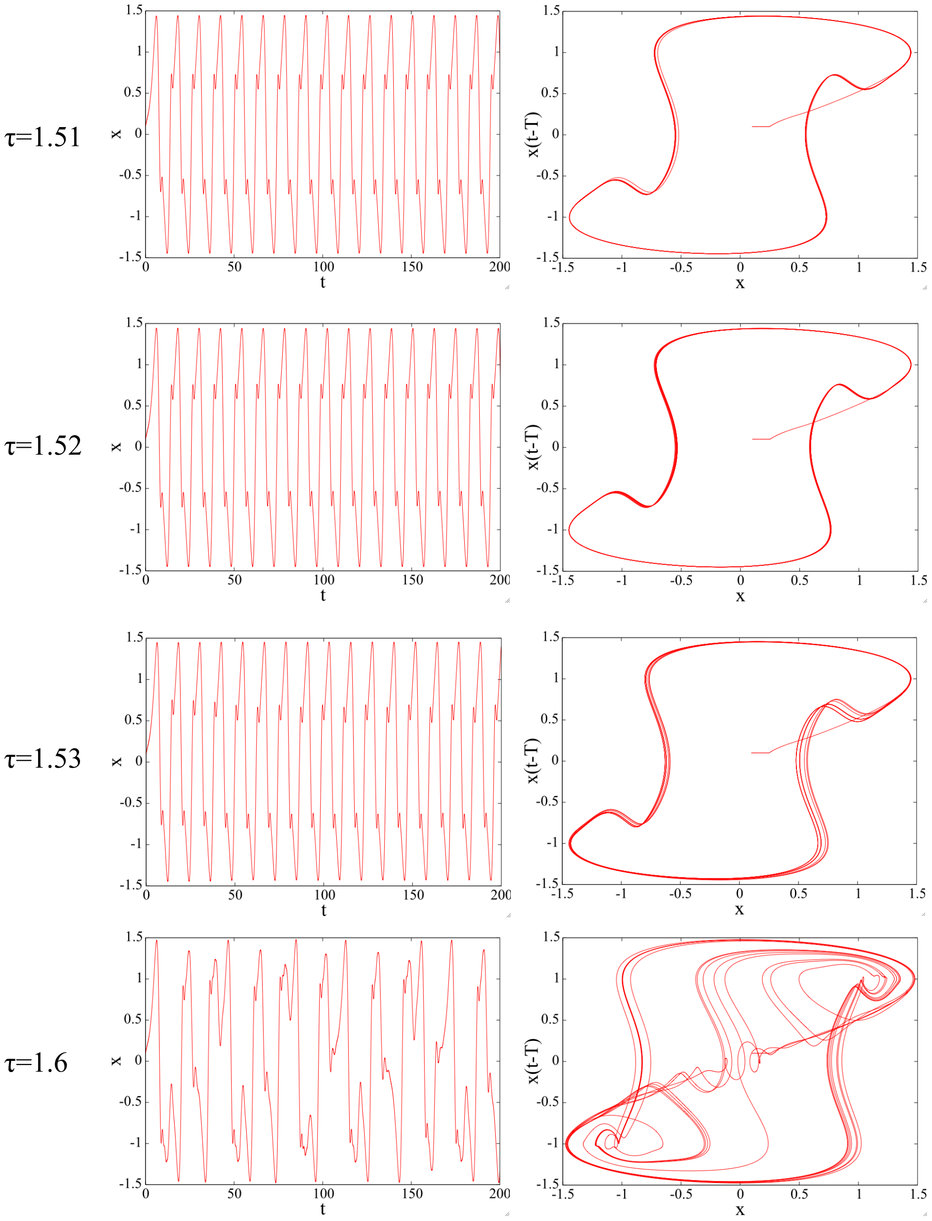


Figure 29: Plots of the realization of $x(t)$ vs t (first column) and the phase portrait $x(t-\tau)$ vs $x(t)$ (second column) as the parameter τ is varied for equation (5.38) with $a = b = 1$. The initial condition is set in the right hand well.

$\tau = 1.51$ - Symmetric periodic orbit.

$\tau = 1.52$ - Non-symmetric periodic orbit.

$\tau = 1.53$ - Period doubling cascade.

$\tau = 1.6$ - Chaos.

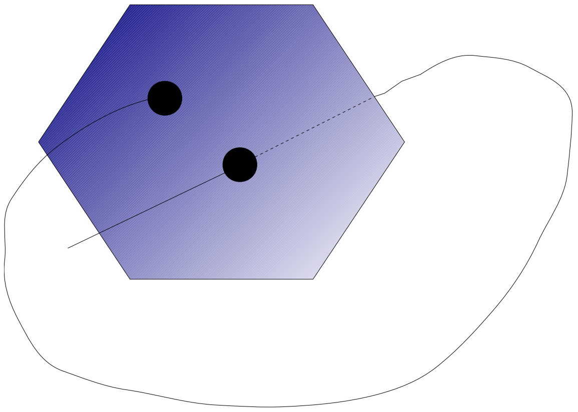


Figure 30: The Poincaré Section – the intersections of an orbit with a subspace of the phase space.

The trajectory is symmetric for $\tau < 1.52$ with the period of the orbit decreasing, at which point the symmetry then breaks, allowing the system to exhibit rich behaviour. A number of studies have found that symmetric periodic orbits will first bifurcate to non-symmetric orbits before period doubling can occur (see e.g. [41, 59, 79]).

Despite the global nature of the behaviour, since we have one large periodic orbit, the dynamics are subject to any bifurcations that can occur to periodic orbits. From the phase portraits in Fig. 29, it appears to have the hallmarks of a period doubling cascade, where a new periodic orbit emerges from an existing one, but with twice the period. However, we need to verify this. In addition to the phase portraits and realizations, period doubling can be identified by Poincaré sections, and power spectra. A Poincaré section, Fig. 30, is the intersection of an orbit with a particular transversal subspace of lower dimension than the phase space (e.g. a 2-D plane in a 3-D phase space) [20]. A period-1 orbit will intersect at the same point in the section for each oscillation. A period-2 orbit will intersect at two different points, and so on. This, however, could identify other dynamic scenarios and so we need to also check the power spectrum of the oscillations. The power spectrum is calculated by taking the square of the absolute value of the Fourier transform of the signal divided by its length, and is typically plotted with a logarithmic power scale. It shows the amount of power at each frequency component of the signal for a given system. In a period-1 system, a peak will occur at the fundamental frequency and then subsequent peaks at the harmonics, occurring at multiples of the fundamental frequency. As the system progresses towards chaos, more peaks will occur associated with the harmonics and sub-harmonics of the system. Other possible bifurcation scenarios for periodic orbits are torus breakdown [2], or intermittency (Pomeau-Manneville) [64]. The system undergoes a series of period doubling bifurcations (Figs. 31 - 33) before a chaotic solution is found for $\tau \approx 1.54$. For $1.54 < \tau < 1.73$ the system exhibits chaotic behaviour with sporadic periodic windows. The route to chaos is illustrated numerically in Fig. 29 and a bifurcation diagram is shown in Fig. 34.

The solution is unbounded for $\tau > 1.73$. Since we have a deterministic system, the location of the local attractor will be determined by the initial condition (in this case of infinite dimension). We see here that if a large enough constant initial condition over the interval $t \in [-\tau, 0]$ is chosen, then the particle oscillates with an ever increasing amplitude, i.e. the solution is unbounded (Fig. 36). The range of initial conditions for which unbounded solutions occur varies as τ varies. This indicates that the size of the delay changes the size and/or shape of the basins of attraction. From the numerical simulations we see that above a certain threshold value of τ, all solutions will be unbounded for typical initial conditions, Fig. 35. The basin of attraction being infinite dimensional makes visualization difficult, but we expect this apparent feature of a changing basin of attraction to be more obvious in non-symmetric cases.

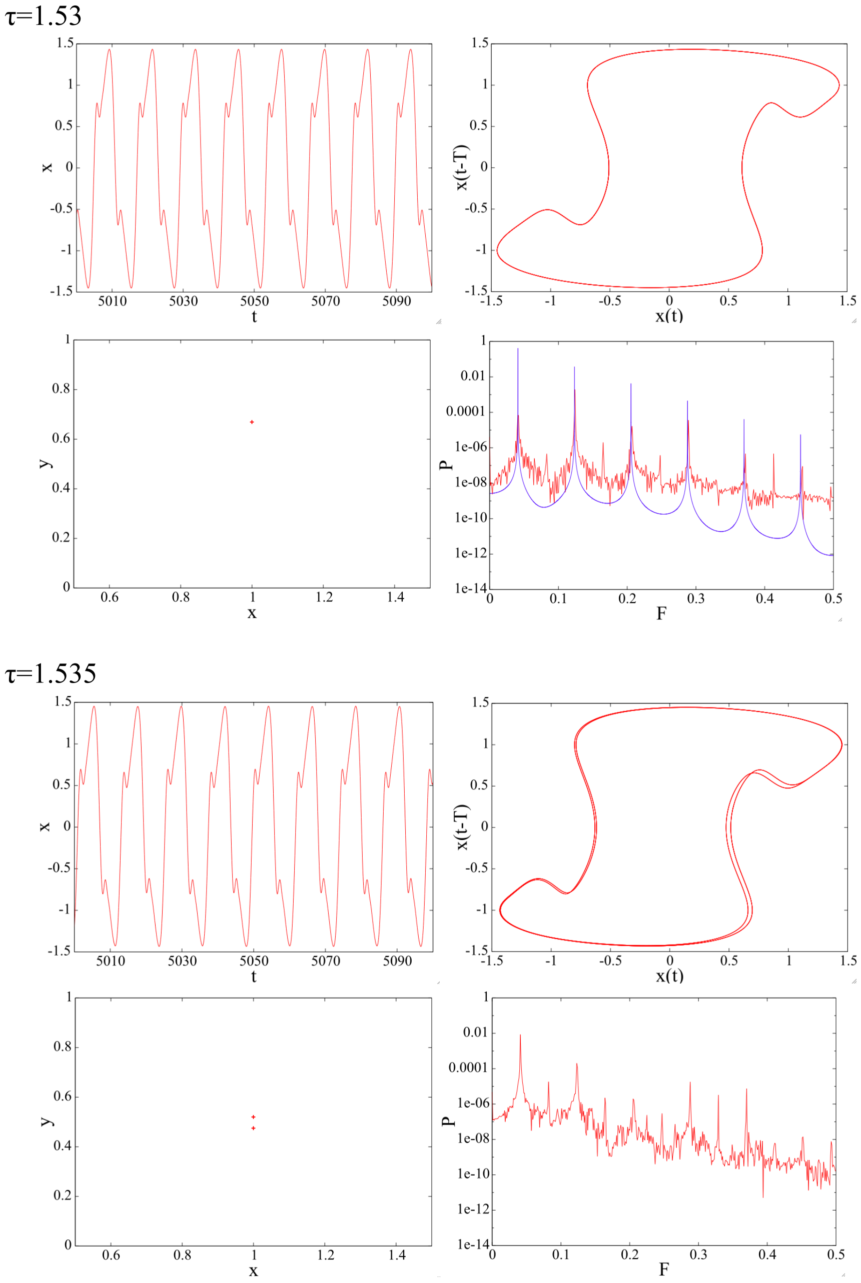


Figure 31: The period doubling cascade in the symmetric system: periods 1 (at $\tau = 1.53$) and 2 (at $\tau = 1.535$).

Top left: Realization $x(t)$ vs t. Top right: Phase portrait $x(t-\tau)$ vs $x(t)$.

Bottom left: Poincaré section. Bottom right: Power spectrum – red denotes current τ, blue denotes power spectrum for $\tau = 1.52$.

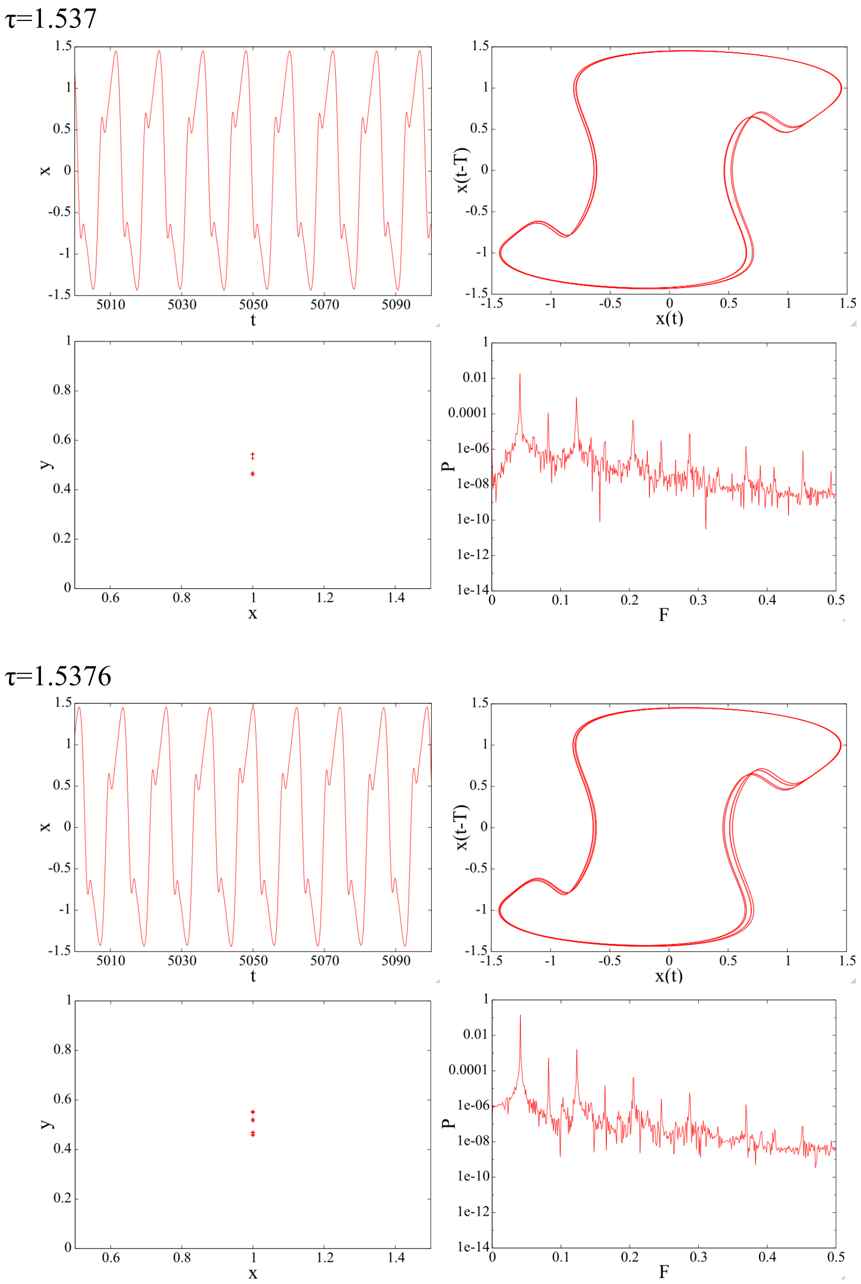


Figure 32: The period doubling cascade in the symmetric system: periods 4 (at $\tau = 1.537$) and 8 (at $\tau = 1.5376$).

Top left: Realization $x(t)$ vs t. Top right: Phase portrait $x(t-\tau)$ vs $x(t)$.

Bottom left: Poincaré section. Bottom right: Power spectrum.

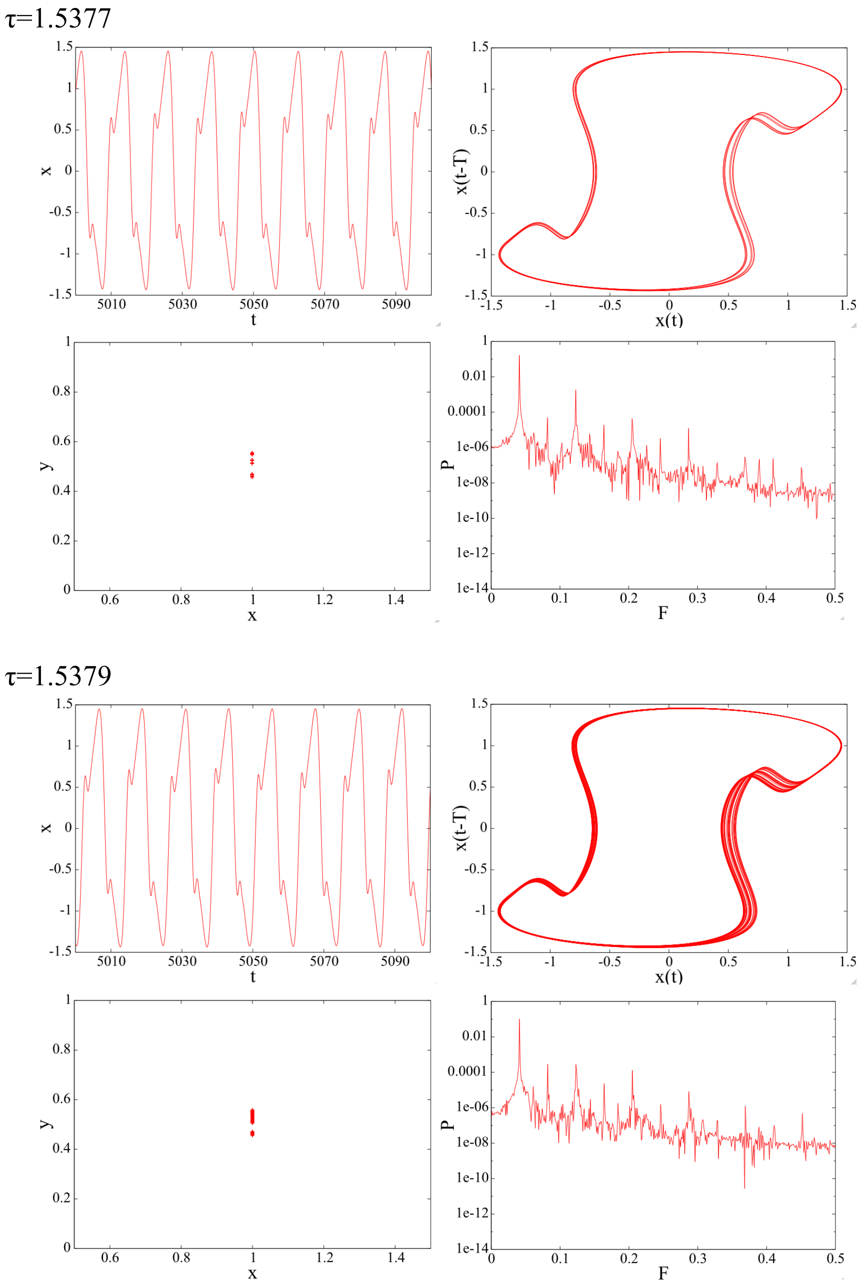


Figure 33: The period doubling cascade in the symmetric system: periods 16 (at $\tau = 1.5377$) and chaos (at $\tau = 1.5379$).

Top left: Realization $x(t)$ vs t. Top right: Phase portrait $x(t-\tau)$ vs $x(t)$.

Bottom left: Poincaré section. Bottom right: Power spectrum.

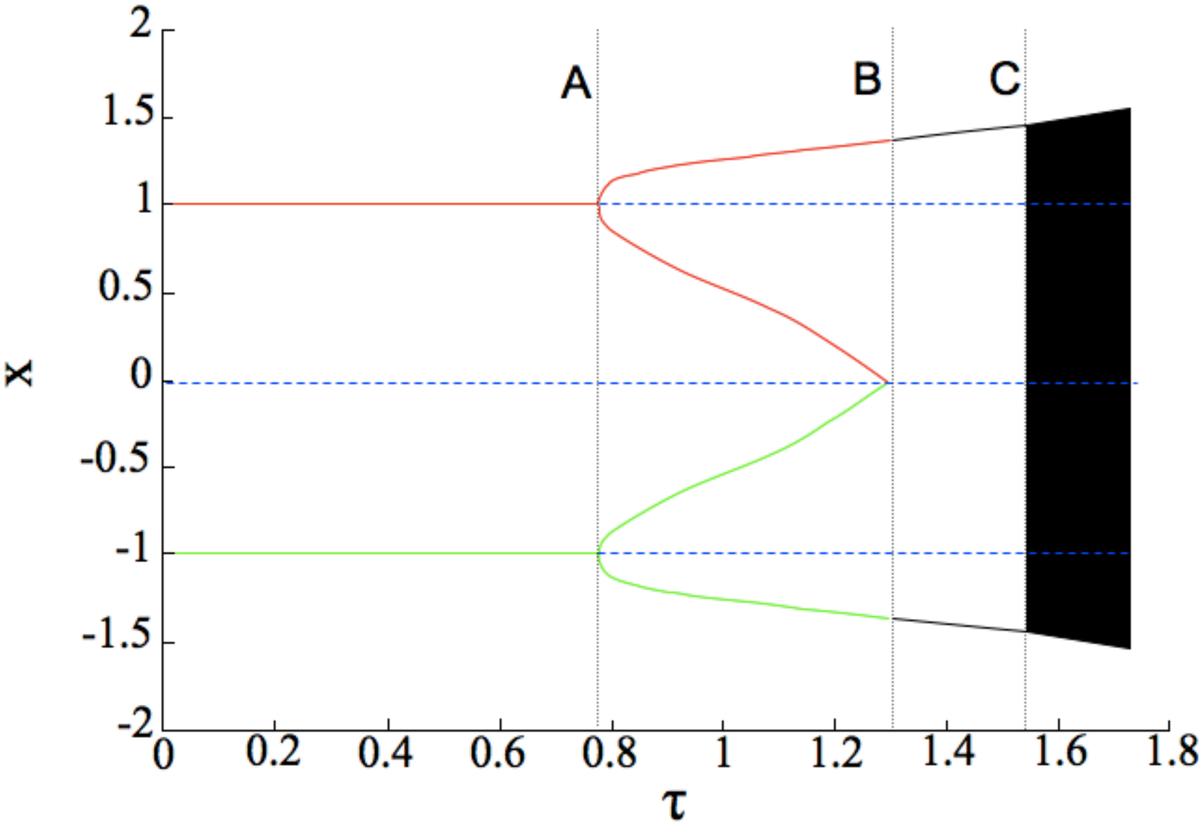


Figure 34: Bifurcation diagram for system (5.38) with $a = b = 1$ and constant initial conditions $x = 0.1$ (red) and $x = -0.1$ (green) as τ is varied. Blue points denote unstable points. All points in the diagram are the points of intersection between the special solution (fixed point, periodic orbit, chaotic attractor) with the surface defined as $\dot{x} = 0$, e.g. in region A one can see the fixed points, in region B one can see minimum and maximum of periodic orbits, in region C all local minima and maxima of $x(t)$ are shown.

A - Andronov-Hopf bifurcation.

B - Homoclinic bifurcation.

C - Chaos.

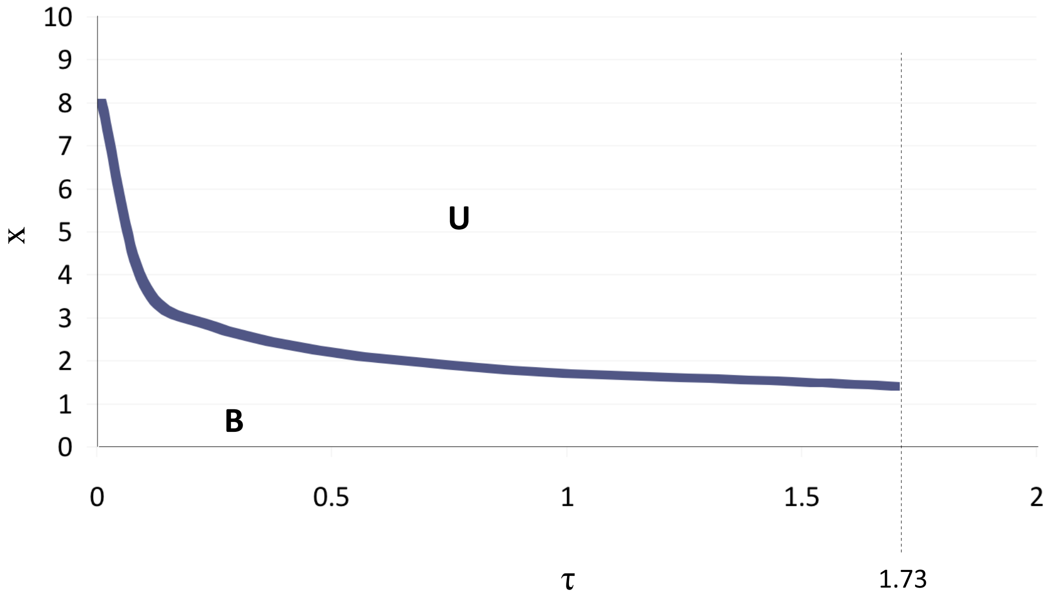


Figure 35: The boundary between the values of the initial conditions (where initial conditions are constant and fixed i.e. $x_0(t) = const, t \in [-\tau, 0]$) in the right hand well, starting on the right hand side of the equilibrium, that lead to a bounded solution (B) of the system, and those that lead to an unbounded solution (U). For $\tau \geq 1.73$ all solutions are unbounded.

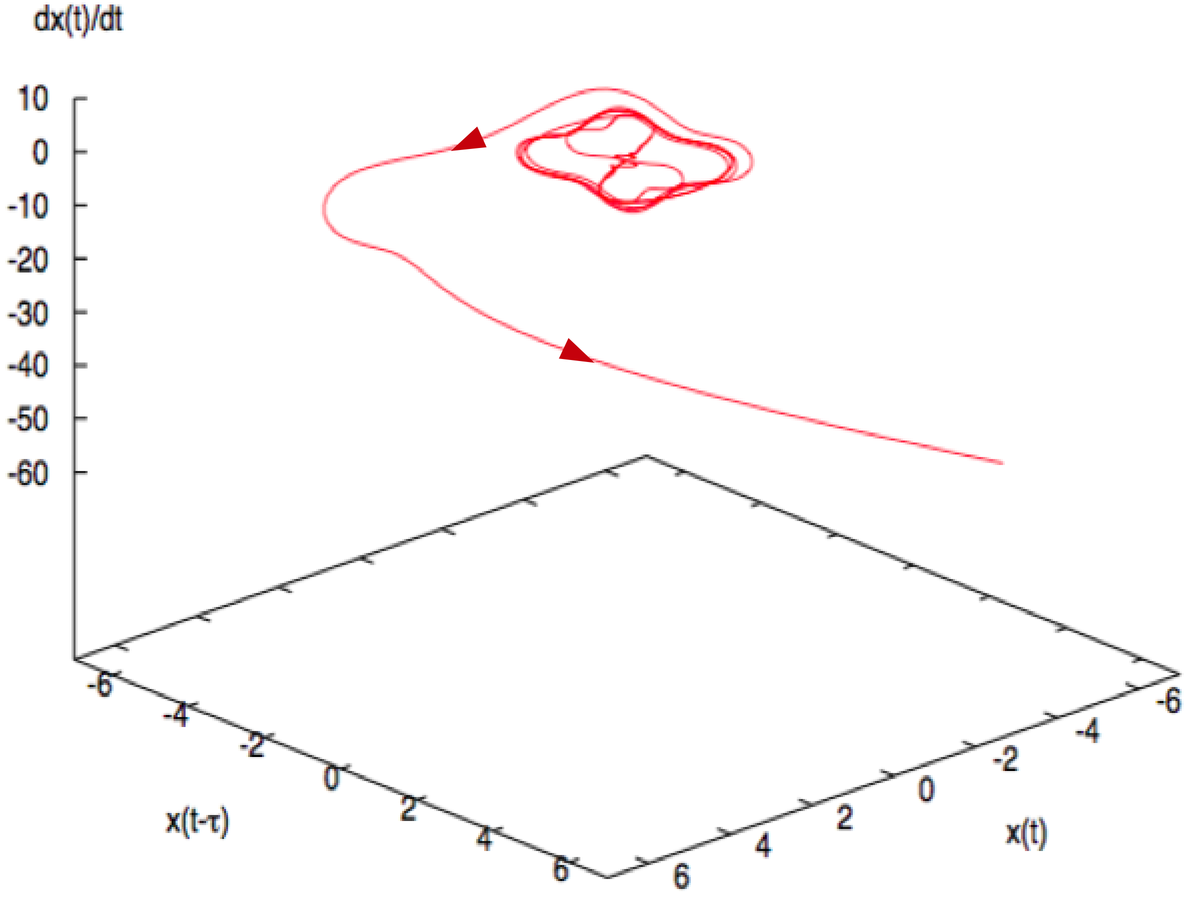


Figure 36: Example of path to unbounded solutions represented by a 3-D phase portrait: $\frac{dx}{dt}$ vs $x(t-\tau)$ vs $x(t)$. The solution circles around the equilibrium in what looks like a periodic orbit, however, these circles continually grow in size and eventually the solution grows to infinity.

τ	$\mathbf{x = -2.5}$	$\mathbf{x = 2}$
1.4	Andronov-Hopf bifurcation	
1.75		Andronov-Hopf bifurcation
2.38	Homoclinic bifurcation	
	All trajectories tend to stable periodic orbit around $x = 2$	
2.92		Homoclinic bifurcation
	Global periodic orbit	
2.93	Period doubling	
2.97	Chaos	
3.03	Unbounded solutions	

Table 1: Summary of bifurcations as τ is varied for equation (5.40) with $a = 0.5$, $b = 0.1$ and $c = 0.05$.

Non-symmetric potential

The symmetry involved in (5.3) clearly affects the dynamics, so we now consider, for generality, a non-symmetric potential given by

$$V(x) = -\frac{ax^2}{2} + \frac{bx^4}{4} + \frac{cx^3}{3}, \tag{5.39}$$

and its corresponding vector field

$$f(x) = -\nabla V = -\frac{dV(x)}{dx} = ax - bx^3 - cx^2. \tag{5.40}$$

Consider $a = 0.5, b = 0.1, c = 0.05$, with stable equilibria at $x = -2.5$ and $x = 2$ and an unstable equilibrium at $x = 0$.

Table 1 details the key values of τ for which the dynamics change, and selected realizations and phase portraits are shown in Fig. 37. The important difference between this and the previous example is the lack of symmetry here. This means that bifurcations of the stable equilibria occur at different values of τ to each other and the dynamics of the system become clearer. From Table 1, we see that at $\tau = 2.38$ the negative equilibrium undergoes a homoclinic bifurcation. Here the saddle-focus has a negative saddle quantity, and trajectories now tend to the periodic orbit around the equilibrium at $x = 2$. At $\tau = 2.92$, the positive equilibrium undergoes a homoclinic bifurcation with negative saddle quantity, and again we have one orbit covering all the equilibria. At $\tau = 2.97$ the orbit undergoes period doubling and the subsequent route to chaos seen in the first example. All solutions are unbounded for $\tau \geq 3.03$.

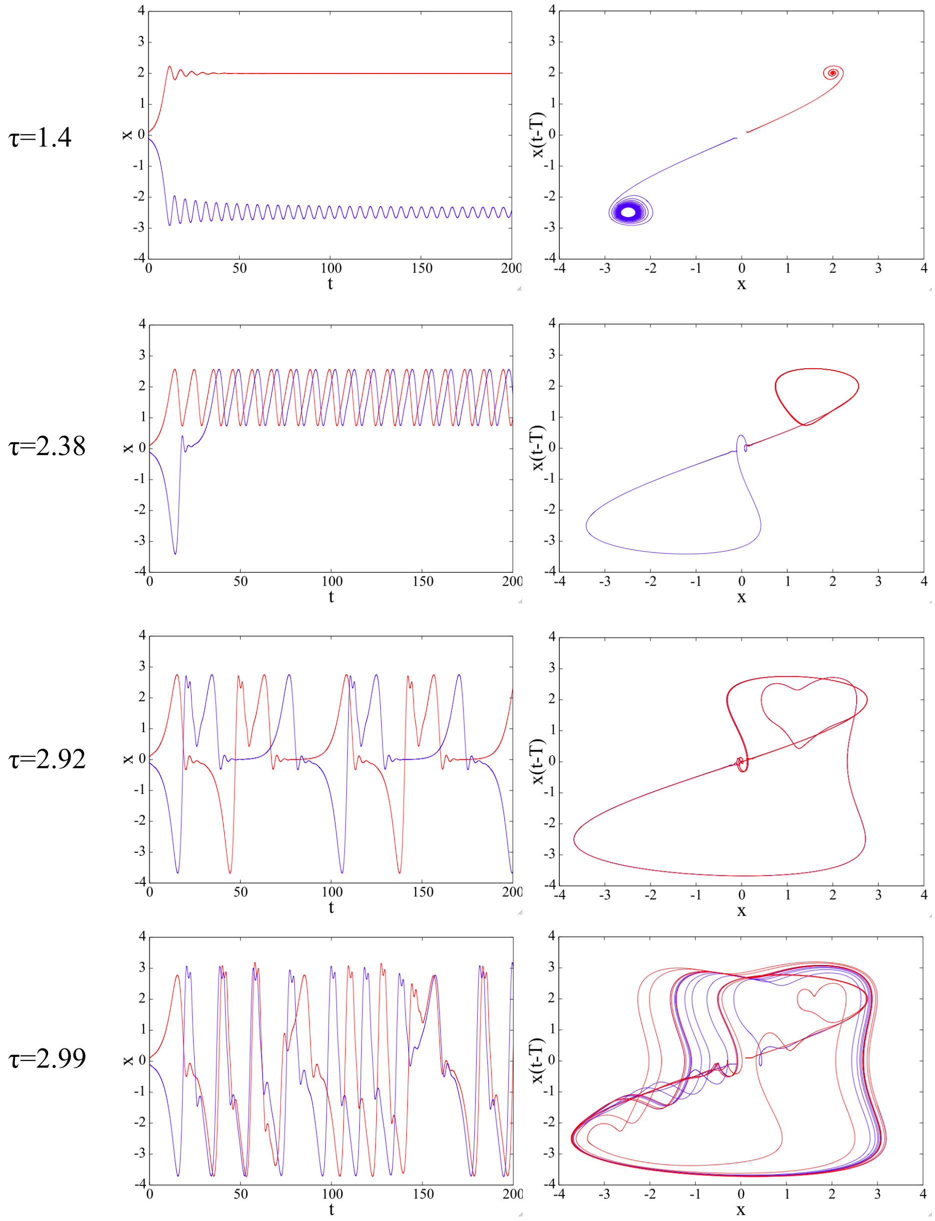


Figure 37: Plots of the realization of $x(t)$ vs t (first column) and the phase portrait $x(t-\tau)$ vs $x(t)$ (second column) as the parameter τ is varied for equation (5.40) with $a = 0.5$, $b = 0.1$ and $c = 0.05$. Blue line denotes initial condition $x = 0.1$; Red line denotes initial condition $x = -0.1$.
$\tau = 1.4$ - Periodic orbit in the well of the $x = -2.5$ equilibrium, stable focus in the well of the $x = 2$ equilibrium.
$\tau = 2.38$ - Homoclinic bifurcation of the $x = -2.5$ equilibrium and the saddle-focus. All trajectories tend to the periodic orbit around $x = 2$.
$\tau = 2.92$ - Homoclinic bifurcation of the $x = 2$ equilibrium with the saddle-focus, forming one large stable periodic orbit .
$\tau = 2.99$ - Chaos.

Hypothesised birth of a large stable orbit

We now put forward a hypothesised scenario for the formation of a large stable orbit around the multiple equilibria in a 1-D delayed gradient system – in the case with an initially stable fixed point either side of the unstable saddle-focus. As the delay increases, the stable equilibrium undergoes an Andronov-Hopf bifurcation, meaning the equilibrium loses stability and a stable periodic orbit is formed around it (Fig. 38, top and second row). This will happen to both stable equilibria at values of τ dependent on the initial system. As explained in Section 5.3.1, this periodic orbit will then grow. As the periodic orbit grows it approaches the manifolds of the saddle-focus, which distort the shape of the periodic orbit (Fig. 23). As the parameter further increases, the periodic orbit and saddle-focus will collide and form a homoclinic orbit (Fig. 38, second row). As mentioned in Section 5.3.2, if the saddle quantity here is positive we will have a chaotic set in the neighbourhood of the saddle-focus. Otherwise as the parameter increases beyond the bifurcation point, the trajectory flows to the other side of the stable manifold of the saddle-focus (Fig. 26) and is attracted to the remaining stable periodic orbit (Fig. 38, third row).

The remaining periodic orbit undergoes a similar process eventually colliding with the saddle-focus in another homoclinic bifurcation (Fig. 38, fourth row). Again the same saddle quantity theory is applicable, however, there are now no longer any local attractors in the phase space. In fact we have on each side of the saddle-focus a trajectory being directed to the opposite side. For a small range of τ values, this produces a large stable periodic orbit covering all equilibria (Fig. 38, bottom row). We are left with a periodic orbit that can undergo dynamic changes associated with local periodic orbits. From the examples considered so far, this typically appears to be a period doubling cascade to chaos.

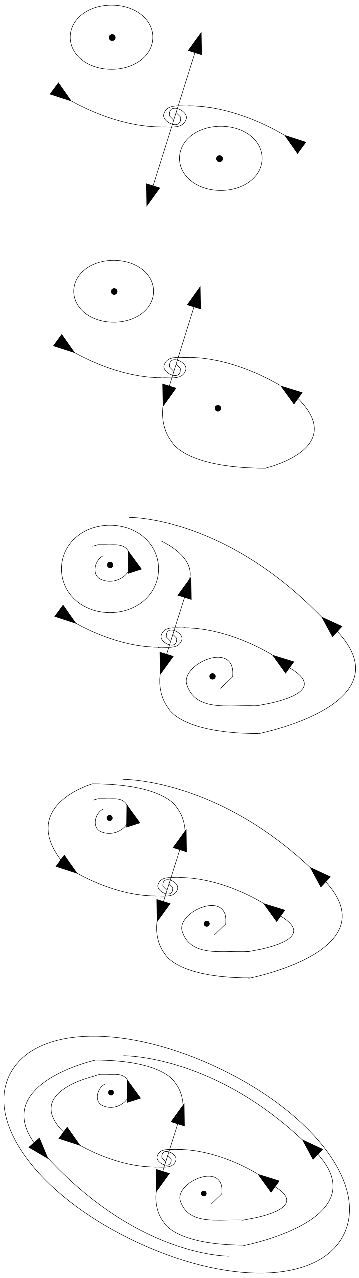


Figure 38: A schematic representation of the hypothesised formation of the large periodic orbit covering all equilibria in a projection of the infinite dimensional phase space.

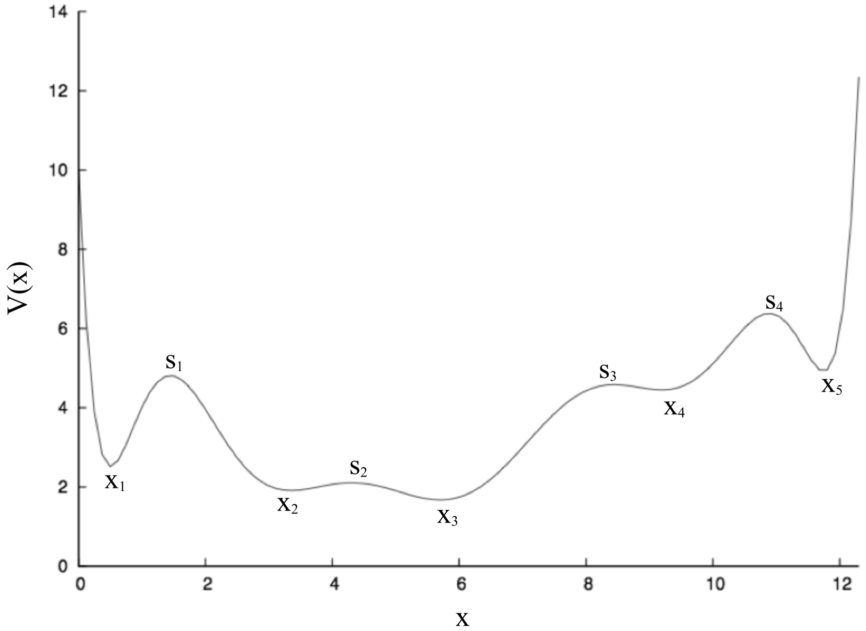


Figure 39: An example of a potential energy function given by an order-10 polynomial, specifically chosen due to its many minima and maxima. There are 5 minima $x_1 = 0.508$, $x_2 = 3.349$, $x_3 = 5.701$, $x_4 = 9.196$, $x_5 = 11.745$, and 4 maxima $s_1 = 1.449$, $s_2 = 4.307$, $s_3 = 8.443$, $s_4 = 10.886$.
Here $V(x) \approx 10 - 39.6x + 74.5x^2 - 63.5x^3 + 29.5x^4 - 8.2x^5 + 1.4x^6 - 0.2x^7 + 0.01x^8 - 0.0004x^9 + 0.000007x^{10}$.

5.5 Application to a multi-well potential with delay

We now consider a more complicated system - a potential energy function with 5 minima, defined by an order-10 polynomial, shown in Fig. 39. Again, we consider a particle subjected to a delay in the vector field $V(x)$, i.e. the gradient system of this potential $\dot{x} = V(x(t-\tau))$. Using (5.25) we find that the Andronov-Hopf bifurcations occur as given in Table 2.

The two equilibria nearest each end of the potential (x_1, x_5), undergo Andronov-Hopf bifurcations for very small values of τ due to the steep gradients of the potential around the fixed points, see Table 2. Their periodic orbits then grow quickly, again due to the steep gradients of the potential, and they soon meet the nearby saddle-focus equilibria in homoclinic bifurcations, at which point the periodic orbit is destroyed and their trajectories are taken by the neighbouring attractors, or become unbounded depending on the initial conditions.

	value of x at equilibrium	value of τ for Andronov-Hopf bifurcation
x_1	0.508	0.056
x_2	3.349	1.041
x_3	5.701	1.097
x_4	9.196	1.094
x_5	11.745	0.075

	value of x at equilibrium
s_1	1.449
s_2	4.307
s_3	8.443
s_4	10.886

Table 2: Top: The value of τ for which each equilibrium undergoes an Andronov-Hopf bifurcation.
Bottom: The locations of the saddle-focus equilibria.

The remaining three stable fixed point equilibria remain this way until $\tau \approx 1$. They then each undergo Andronov-Hopf bifurcations for similar values of τ, see Table 2. Their periodic orbits then grow at a rate depending on the gradients of the surrounding energy function.

At $\tau \approx 1.74$, the equilibrium x_2 undergoes a homoclinic bifurcation with saddle-focus s_2. Following the destruction of the periodic orbit, trajectories from x_2 are now sent to the periodic orbit around x_3. At $\tau \approx 1.78$, x_4 undergoes a homoclinic bifurcation with s_3. Similarly, trajectories from here are now sent to the remaining stable periodic orbit around x_3.

When $\tau \approx 1.91$, x_3 undergoes a homoclinic bifurcation with s_2 with negative saddle value, forming a periodic orbit covering both the x_2 and x_3 equilibrium wells. At $\tau \approx 2.22$, the orbit becomes locally chaotic around the x_2 and x_3 equilibria. This chaotic set then grows as τ increases. At $\tau \approx 2.54$ the chaotic set grows to include the orbits around x_4. This set grows further as τ increases before all solutions become unbounded at $\tau \approx 3$. Selected phase portraits are shown in Fig. 40, and the bifurcation diagram is shown in Fig. 41.

In Fig. 42, we illustrate the possibility of a "chain reaction" of large periodic orbits spreading across multiple saddle-foci. Since that there are now multiple saddle-foci in the system surrounded by minima in the potential, these large periodic orbits could then merge into even larger periodic orbits across several saddle-foci. Here, with a suitable choice of shape for the potential function, the periodic orbit could cover all the equilibria in such a multi-well potential.

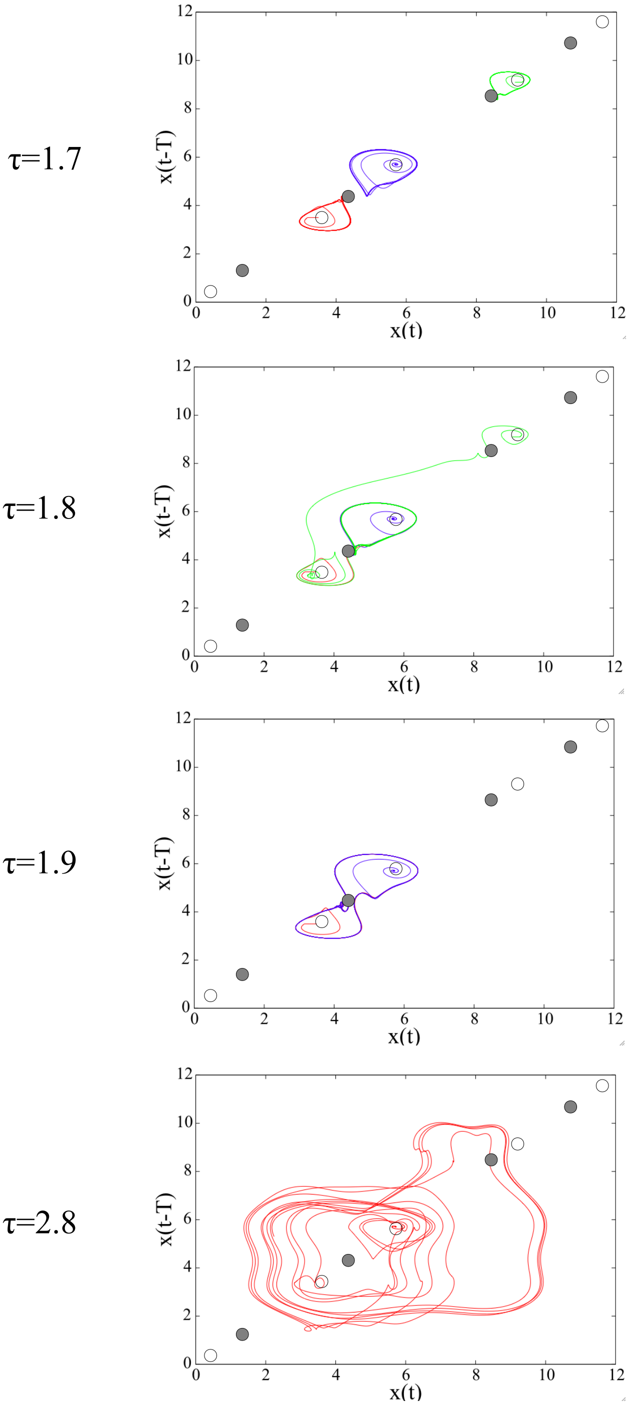


Figure 40: Selected phase portraits of the multi-well system with delay for varying τ. Stable equilibria are shown by clear circles, unstable equilibria are shown by filled circles.

$\tau = 1.7$ - Separate periodic orbits around the x_2, x_3, x_4 equilibria.

$\tau = 1.8$ - Trajectories tend towards the periodic orbit around x_3.

$\tau = 1.9$ - Periodic orbit covering x_2, s_2, x_3.

$\tau = 2.8$ - Chaotic set covering multiple equilibria.

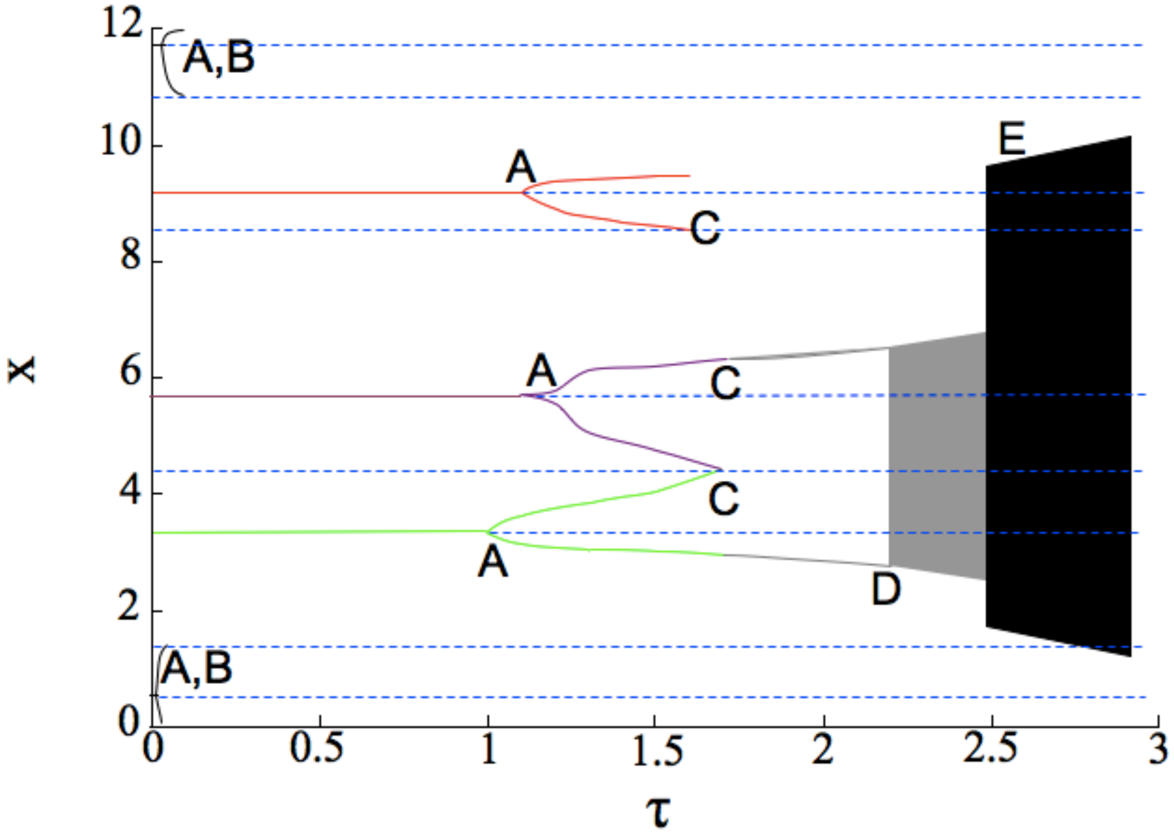


Figure 41: Bifurcation diagram for all the equilibria in the order-10 polynomial gradient system with delay, with constant initial conditions. The notation remains the same as in Fig. 34. Blue points denote unstable states, otherwise different colours denote different initial conditions.

A - Andronov-Hopf bifurcations.

B - Unbounded solutions.

C - Homoclinic bifurcations (For the red line, trajectories now tend a local attractive state (see text for details).

D - Local chaos.

E - Global chaos.

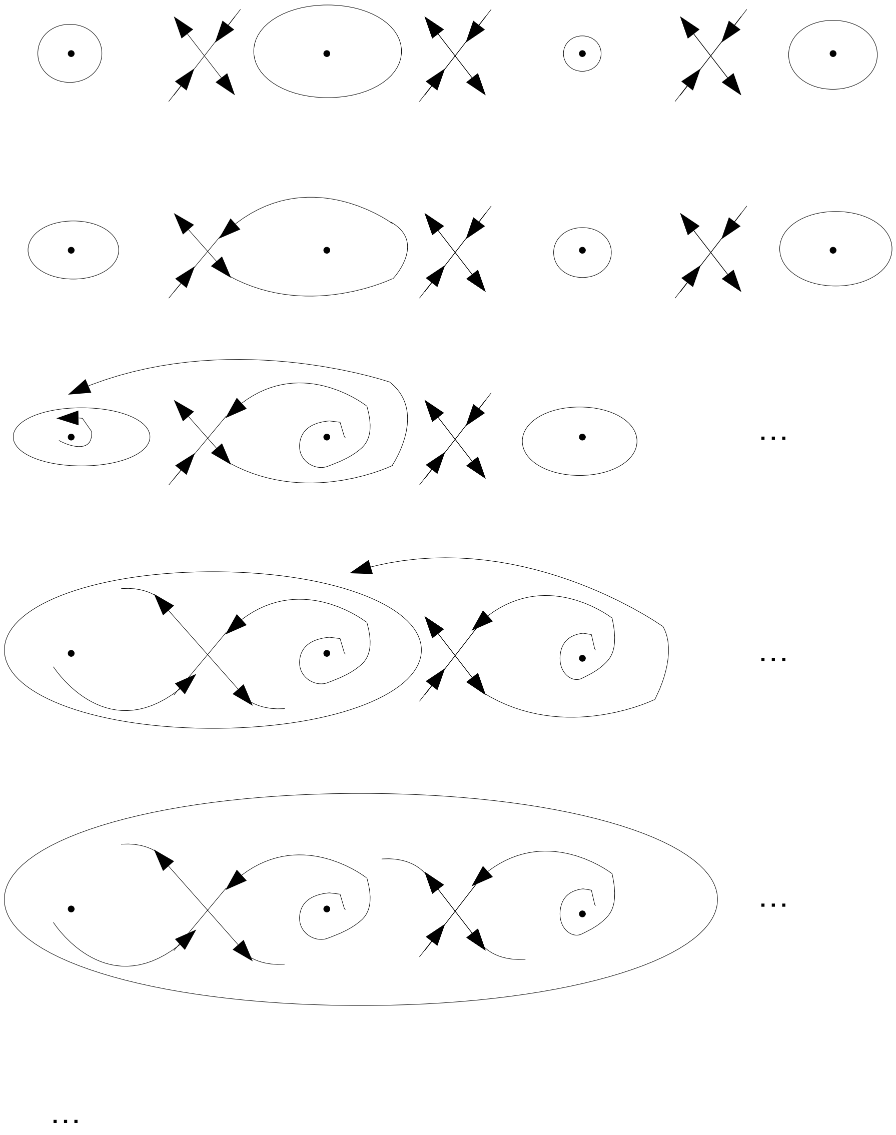


Figure 42: A schematic representation of the hypothesised formation of orbits covering multiple saddle-foci in a projection of the infinite dimensional phase space.

5.6 Discussion

To investigate the effect of time delay on self shaping systems, we assume that, in the long term, their vector field has achieved some stationary shape. Then we can consider the problem as a fixed gradient system with delay. Here, we have investigated the effect of delaying the whole vector field of the system by the same amount. Using linear stability analysis, we have derived an equation to find the value of the delay, τ, for which the Andronov-Hopf bifurcation will occur for each of the stable equilibria of any such system. We have shown that this bifurcation will be supercritical and will produce a stable periodic orbit. We have also shown that all unstable equilibria in the non-delayed system will become saddle-foci when a delay is introduced. As the delay is increased, we have shown numerically that there will be a seemingly standard behaviour. The periodic orbit around the original stable equilibria will grow, before it meets a nearby saddle-focus. At this point the pair of equilibria will undergo a homoclinic bifurcation resulting in a homoclinic orbit. This will happen around each of the stable equilibria. At this point the subsequent behaviour then depends on the sign of the saddle quantity. If negative, trajectories are sent to another attractor in the phase space. As this process continues further large periodic orbits covering multiple equilibria form, and chaotic sets appear through period doubling. This chain reaction of homoclinic bifurcations can eventually lead to a global chaotic attractor. Alternatively, if the saddle quantity is positive there will be Shilnikov chaos. In any case, chaos appears to be inevitable. As τ increases further the chaotic set grows before all solutions diverge to infinity (i.e. become unbounded).

Furthermore, we have found that unbounded solutions can occur dependent on the value of the delay. These appear to occur due to the changing shape of the boundary of the basin of attraction as the delay changes. The coexistence of bounded and unbounded solutions is relevant should an external force be applied to the system. It implies that if subject to a small perturbation, the system could diverge to infinity from previous convergent states.

Despite its apparent simplicity, delaying all state variables in a gradient system appears to be an effective method of producing chaos, although the route to which this happens varies depending on the parameter. The environment should always have a similar structure to 1-D potential so we would expect similar behaviour for higher dimensional potentials.

The obvious extension of this work is to verify the behaviour for higher dimensional potential energy profiles, and to form a more rigorous justification and explanation of the dynamics.

Furthermore, the non-stationary self-shaping system case could be investigated. For systems with no stationary behaviour, generalisations would be difficult. The design of self-shaping systems is such that it shapes itself to the inputs it receives, and so without knowledge of the statistical properties of these inputs it would be impossible to predetermine the behaviour if delay were introduced. At each moment in time we will still have a conventional gradient system and so the periodic and homoclinic behaviour should still occur. However, they will be there in conjunction with the stochastic nature of the underlying system and so intuitively it would appear that unbounded solutions may be harder to avoid due to the changing nature of the vector field and the possibility of becoming isolated from the attractors. A suitable choice of initial conditions and parameters, however, should still ensure the formation of trajectories covering all the equilibria and the appearance of chaotic like behaviour – although given the stochastic nature of the underlying system the chaos could not be classified as deterministic.

Global optimization by delay annealing

We have seen that we can create large periodic orbits and then chaotic sets covering all equilibria in the gradient system, which leads us to ask – could this be applied to any existing, practical problems? Perhaps the most obvious application, is to the problem of global optimization within a potential energy function.

The problem of optimization is stated as follows. A so-called cost function is introduced that usually depends on several parameters. This function typically has a lot of local minima and maxima. One needs to find a set of parameters, at which the cost function takes its global maximum or minimum. Usually, they aim to find the global minimum, and if the maximum is required, the cost function is simply taken with the negative sign.

The problem might not sound too difficult, but there are several aspects of it that make the solution non-trivial. Usually, one either does not know the exact analytical expression describing the cost function on the whole domain, or this expression is too large or complicated for it to be efficient to find the location of all maxima and minima and then and compare their “depths” and “heights” [26]. One assumes that the cost function can be only estimated at the chosen sets of parameters and in their close vicinities. Therefore, in this case it is impossible to find the global minimum by analysing the cost function on its domain in the usual way.

To solve optimization problems, several methods were introduced in computer science literature [17]. One popular method is to assume that the cost function is treated as an energy landscape, E, and the current set of parameters as coordinates of a massless particle moving in this landscape. Then the equation for this particle is

$$\frac{d\mathbf{x}}{dt} = -\frac{\partial E}{\partial \mathbf{x}} \tag{5.41}$$

To trace the behaviour of the particle in time, one only needs to know its current location, the value of the cost function and its local gradient. It is obvious, that the behaviour of the particle depends solely on the choice of its initial position. Namely, if the initial conditions fall within the potential well of some non-global minimum, the particle converges to this minimum and stays there. Thus, the global minimum is not achieved.

To overcome this problem, one can introduce a stochastic term, $\epsilon(t)$, into the equation

$$\frac{d\mathbf{x}}{dt} = -\frac{\partial E}{\partial \mathbf{x}} + \epsilon(t). \tag{5.42}$$

This stochastic term will allow the particle to escape from local minima, however, finding the global minimum is still a non-trivial task. If the stochastic term is too small, the particle can still get trapped at a local minimum, too large and the particle will never converge to any of the minima. One solution to this, is to make the inclusion of the stochastic term dependent on a certain function.

The system decides whether to accept the new solution at each step with certain probability dependent on a new parameter, T. This parameter decreases in time. The probability is such that the choice is almost random when T is large, but increasingly selects the better (lower energy in minimisation) solution as T decreases, in other words, movements in the to higher energy states become less likely. This allows the particle to initially move freely across the domain, before gradually converging towards the lowest energy state. Being able to sometimes move in the "wrong" direction should prevent the particle from becoming trapped at a local minimum (Fig. 43).

The method above is often referred to as stochastic or simulated annealing [38]. The term "annealing" is borrowed from metallurgy. There, annealing is the process of heating a metal causing changes in its properties (such as hardness), before cooling the metal to produce a refined structure resulting in improved cold working conditions. In this case the parameter T would be the temperature.

Now, compare the behaviour of a gradient system with delayed vector field to the behaviour of a particle in an energy landscape with noise. With large delays the particle can travel across the whole domain, and then if the delay is reduced, the particle converges to a stable

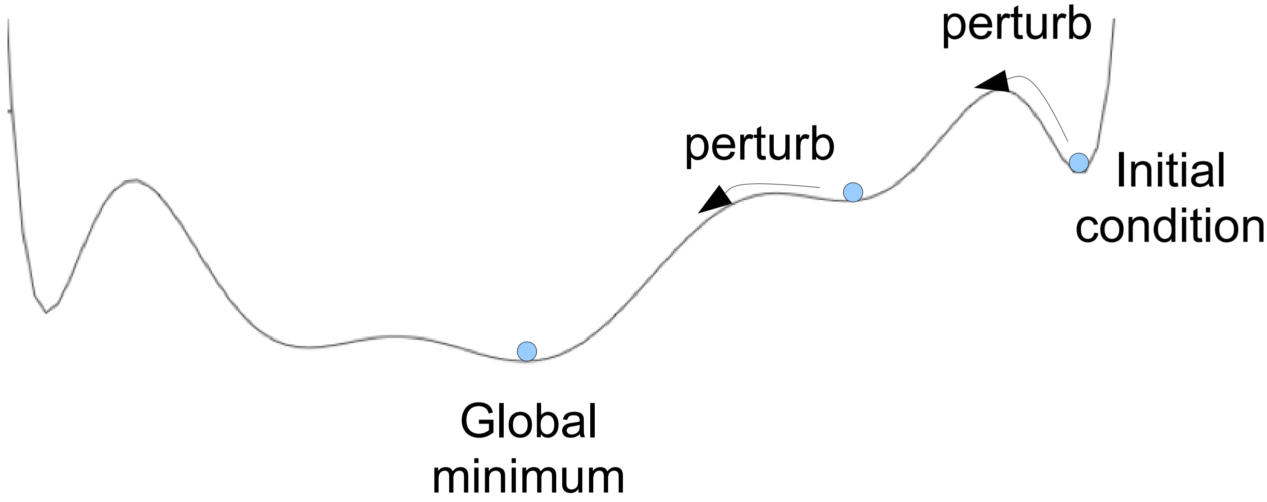


Figure 43: Graphical representation of simulated annealing. Perturbations enable the trajectory to escape local minima. Full description of the method in text.

periodic orbit, and then to the corresponding stable fixed point. It seems possible that this could provide a similar "annealing" based idea, with the parameter T being the delay. Although here the principle is not fully conceived, it could form the beginnings of a viable solution to the global optimization problem we shall refer to as *delay annealing*.

The most obvious issue is that of avoiding unbounded solutions. Clearly, the global trajectory will be unattainable if solutions are unbounded near one or more attractors. The unboundedness arises from the steepness of the potential energy surrounding the attractor. Steeper slopes will increase the rate with which the periodic orbit grows. To prevent this we could, for example, apply a function transformation to the vector field to reduce the gradients involved, e.g. $tanh(df/dx)$ (Fig. 44). It should also be said, that many of the unbounded solutions can also be avoided through choices of initial conditions, and so an easier (but less ideal) solution would be to simply change initial conditions in unbounded solutions are found.

We also need to ensure that the global optimum is locatable in the "cooling" or parameter reduction stage. Assuming that we start with a large enough delay and have avoided unbounded solutions, we will have a trajectory covering all the equilibria. We then want to be able to "turn down" the delay to end up at the global minimum. We have seen from the numerical examples that the point at which the equilibria lose stability is dependent on the

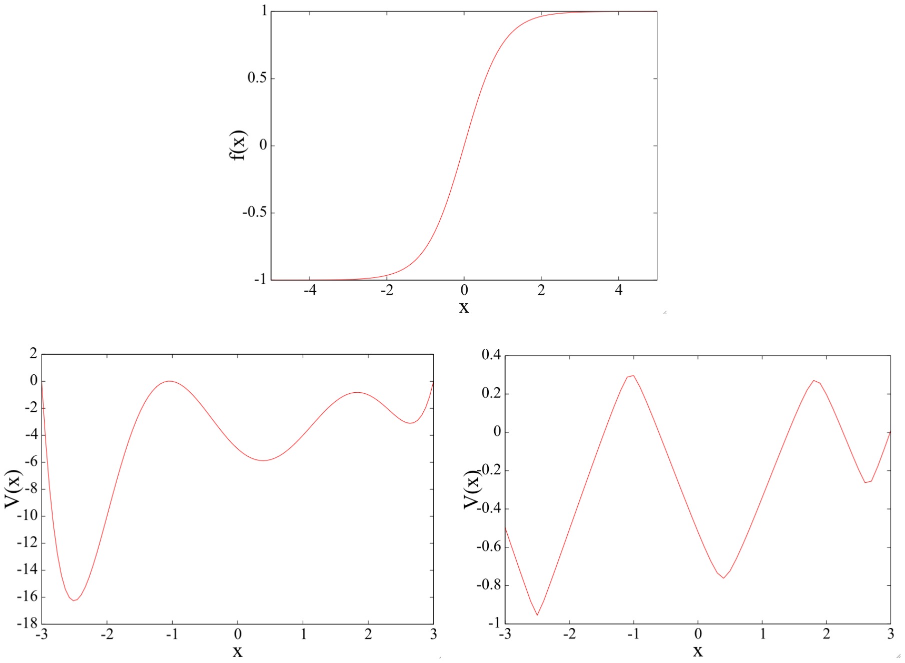


Figure 44: Transformation of the gradient system using $tanh$, reducing the gradient of the potential to avoid the possibility of unbounded solutions.

Top: The function $f(x) = tanh(x)$.

Bottom left: A potential energy function $V(x) = g(x)$. Bottom right: The transformed potential energy function $V(x) = \int_x tanh(g'(x))$.

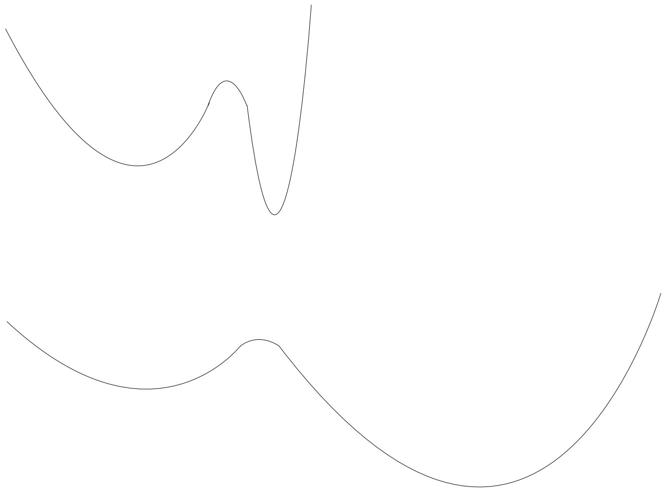


Figure 45: Sketch of the application of a stretching function to a potential energy function.
Top: Sketch of a potential energy function.
Bottom: The stretched potential energy function.

surrounding gradients. Furthermore, we have seen that as we increase delay, the trajectory will tend to stable attractors, and will keep doing this until there are no stable attractors remaining. This implies that if we are working in the opposite direction and are turning the delay down, the first local stable attractor (periodic orbit) to appear will be where the trajectory tends and stays as the delay is further reduced. So how can we go about ensuring that this will coincide with the global minimum?

One hypothetical solution could be to use a “stretching” function transformation, that could stretch the domain of the function depending on the depth. This would ensure that the deepest minima would have the shallowest gradients and largest basins of attraction (Fig. 45).

We would then start with a large delay, before slowly turning the delay down. The trajectory would locate itself in the first stable periodic orbit to appear, and would then stay in this basin as the delay further decreases, converging on the attractor (Fig. 46). The idea of a stretching function is an initial hypothetical solution to the problem, and not an ideal one as it requires evaluation of the potential function over the whole domain. More appropriate solutions to this problem could be the subject of future work. Nevertheless, the use of delay on the vector field of a potential energy function, might offer a solution to the problem of global optimization.

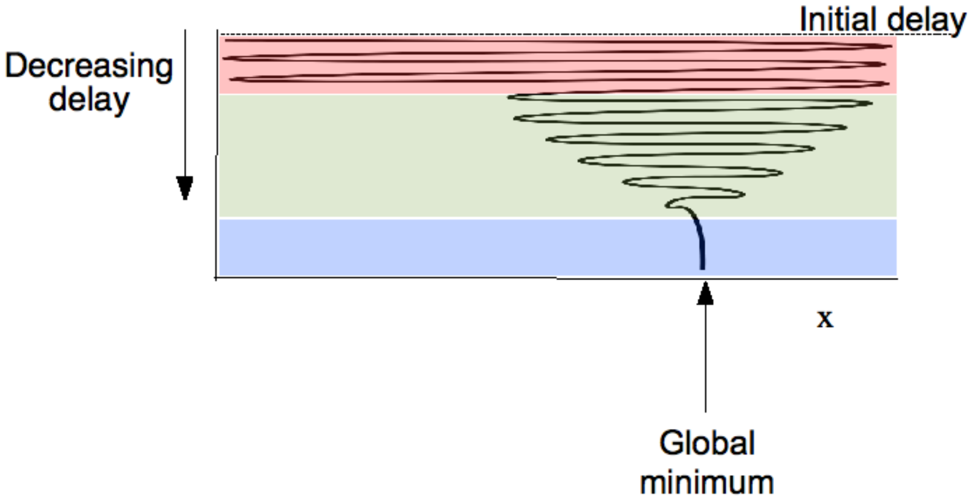


Figure 46: Sketch of an example of a delay annealing trajectory. The system starts with large delay where the trajectory spans the whole set (red region). As the delay is turned down, the trajectory is attracted to the first stable periodic orbit (green region). As delay is further reduced the trajectory converges on the fixed point corresponding to the global minimum (blue region).

6 Overall Conclusions

One of the main challenges of the 20th century was to create a mathematical model of the human mind. Two main paradigms were proposed: of a computer [84] and of an artificial neural network [76]. Both lead to impressive applications, but essentially failed to reproduce the work of a real mind. Machines have surpassed us at very specific tasks but the brain sets the bar for artificial intelligence extremely high with its ability to continually learn new skills.

Supercomputers have provided arithmetic fire-power that humans could never hope to achieve. However, this is only performing numeric calculations. Artificial neural networks offer something different – the ability to generalise. Nevertheless, the full capabilities of the brain are yet to be captured. The brain is working subconsciously, making intuitive leaps, using imagination. Those other qualities that are inherent to intelligent beings make us unique. They are the reason that artificial intelligence is such an interesting and difficult challenge. What is truly remarkable about our brains is that everything we learn to do becomes automatic with practice. Our brain does this with experience and it is a complex problem trying to replicate this. The ultimate goal for the field of artificial intelligence is to create devices that could rival human beings in creating new pieces of art, literature, music, etc. The ability to imagine and innovate is still elusive to artificial devices. We propose what could be the next step in the progression towards such devices.

We began by defining learning and recognition and described the basics of computers and artificial neural networks. We saw that artificial neural networks enabled the ability to generalise that was not possible with computers. However, they also had limitations – namely the requirement of good quality training data, and the formation of spurious attractors. An alternative approach is to use dynamical systems theory to try and overcome these problems. Zak proposed a couple of novel approaches to model learning (and living systems in general) using dynamical systems. These both were based around the idea of controlling the velocity vector field. These approaches also had some technical difficulties, the main one being the reliance upon the system being subject to Gaussian input processes. We attempted to extend this concept to non-specific input distributions, however, this relied on correlations in the input data, again reducing the applicability of the method.

To remove such difficulties, we then proposed what might become the third main learning paradigm – self-shaping dynamical systems – a simple, easy to interpret mathematical model reproducing the adaptability and evolutionary nature of the brain. The system receives inputs sequentially and learns from them in a similar way to the brain – the structure of the system is physically changed. Inputs with similar properties are grouped together in a class. The system also performs recognition at the same time as learning – attributing each input to its relevant class. This whole process is performed automatically, without supervision. However, the incorporation of an outside influence would be possible at any stage.

From a mathematical perspective we have introduced a new kind of dynamical system. These systems are shaped by their input, with the ability to modify the whole structure of the phase space. This differentiates them from stochastic dynamical systems, for example, which are only perturbed by external stimuli and return to their original structure once the stimulus is withdrawn. It would seem that this ability makes them, along with artificial neural networks, the only type of system which can truly modify themselves to external stimuli. However, in artificial neural networks, there is only the possibility to modify the values of couplings, i.e. part of the system, and thus control of the phase space is only on a local level. Here the control is global, the input changes the evolution operator of the system – producing a flexible, self-organising vector field. This will produce a unique system depending only on its experience, much like the brain. Only the simplest of these systems, of gradient type are studied here and their performance is illustrated with an example in the form of a musical pattern. Namely, it is shown how the system automatically discovers separate musical notes and musical phrases. From a biological perspective, this ability to change in response to input is not unique to learning. It occurs in much more general adaptation of living systems, e.g. the growth of muscle after exercise. Self-shaping systems could provide the framework to model this general adaptation of living systems.

We then went on to investigate what interesting dynamics could occur in such a system. In particular, the role that time delay plays in such gradient systems. We simplified the problem to that of a traditional gradient system with delay by arguing that over time the tendency to a stationary solution will result in a similar dynamical behaviour. We investigated the impact of delaying the whole vector field in such systems. We show that this method produces chaos in a class of systems with very simple dynamics in the non-delayed case. We demonstrated that it is possible to analytically predict when periodic orbits will appear and that homoclinic bifurcations occurred when periodic orbits meet with the unstable equilibria before forming large periodic orbits spanning multiple equilibria, then undergoing a period doubling transition to chaos. We also showed that an alternative route to chaos should be possible – Shilnikov chaos, and found the conditions under which that would occur.

This work has multiple avenues down which to continue study that fall into two broad groups: the advancement of self-shaping systems, and the continuation of the study of gradient systems (and then self-shaping systems) with delay.

To advance the concept and theory of self-shaping systems:

- The self-shaping system derived and studied here, is of the simplest kind – of gradient type. This can be extended to more general gradient systems, which would enable the formation of more interesting dynamical objects in the phase space, e.g. periodic orbits, chaotic sets, etc.

- The physical principles upon which these systems could be built are not obvious at the moment and what we have presented here is a mathematical proposal for systems of a new class. We have argued that if implemented in hardware, such systems would have considerable advantages over artificial neural networks. One major advantage of artificial neural networks is the simplicity with which they can be engineered [22]. To promote self-shaping systems as a viable alternative to artificial neural networks requires a similar manufacturability, and so the proposal represents an engineering challenge and calls for the development of devices of a new kind.

To continue the study of the delayed gradient systems:

- In gradient self-shaping system subject to a time delay, only the local behaviour has been proven, and a more rigorous justification of the global behaviour is necessary to fully explain the dynamics.

- The potential for this to serve as a global optimization technique can be further studied. Firstly, to overcome some of the current limitations, and secondly to prove the validity of the method.

- The generalisation to all gradient self-shaping remains to be achieved. Here, only systems where the long term behaviour is stationary (i.e. tends to a gradient system) is studied. The effect of delay on a non-stationary velocity vector field remains to be studied.

To summarize, we have proposed the concept of a dynamical system with a fully flexible vector field that spontaneously self-organises in response to stimulation, while automatically utilising its previous experience. Having compared their capability at learning and recognition of these *self-shaping systems* with the traditional approach of artificial neural networks, we have seen that they have the potential to be more powerful than artificial neural networks, overcoming the limitations of spurious attractors. Generalizations by introducing the ability to forget would lead to a completely new class of dynamical systems. Such systems could offer a mathematical prototype of learning and adapting machines of a new generation. Then, having incorporated a time delay on such systems (under certain assumptions) and studied their behaviour, we have speculated that this might offer a solution to the problem of global optimization.

The work of Section 4 is partly covered by articles in the ArXiv repository [35, 36].

A Appendix - Derivation of the Fokker-Planck equation

In Section 3, the Fokker-Planck equation is used to calculate the time evolution of the probability density of a random process. Here, we provide a derivation of the equation, combining and expanding on the work in [66, 9].

Consider a system with N stochastic variables $\boldsymbol{\xi} = \xi_1, \xi_2, \ldots, \xi_N$. The transition probability $Q(\mathbf{x}, t+\Delta t|z)$ for N variables is given by the Chapman-Kolmogorov equation,

$$p(\mathbf{x}, t+\Delta t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} Q(\mathbf{x}, t+\Delta t|z, t)p(z,t)\, dz_1 dz_2 \ldots dz_N, \tag{A.1}$$

where $p(\mathbf{z}, t)$ is the probability density of $\mathbf{z}$ at time t and $p(\mathbf{x}, t+\Delta t)$ is the probability density of $\mathbf{x}$ at time $t+\Delta t$.

Now [24],

$$Q(x, t+\Delta t|z, t) = \int_{-\infty}^{\infty} \delta(y-x) Q(y, t+\Delta t|z, t)\, dy, \tag{A.2}$$

which can be extended to N dimensions, so that

$$Q(\mathbf{x}, t+\Delta t|\{z\}, t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \delta(\mathbf{y}-\mathbf{x}) Q(\mathbf{y}, t+\Delta t|\mathbf{z}, t)\, dy_1 dy_2 \ldots dy_N, \tag{A.3}$$

where,

$$\delta(\mathbf{x}) = \delta(x_1)\delta(x_2)\ldots\delta(x_N) \tag{A.4}$$

denotes the Dirac-delta function in N variables.

A Taylor series expansion at $y = z$ of the Dirac-delta function is given by

$$\begin{aligned} \delta(y-x) &= \delta(z-x+y-z) \\ &= \sum_{n=0}^{\infty} \frac{(y-z)^n}{n!}\left(\frac{\partial}{\partial z}\right)^n \delta(z-x) \\ &= \sum_{n=0}^{\infty} \frac{(y-z)^n}{n!}\left(\frac{-\partial}{\partial x}\right)^n \delta(z-x). \end{aligned} \tag{A.5}$$

This can be generalised to a function of N variables by

$$f(x_1, \ldots, x_N) = \sum_{n_1=0}^{\infty}\cdots\sum_{n_N=0}^{\infty} \frac{(x_1-a_1)^{n_1}\cdots(x_N-a_N)^{n_N}}{n_1!\cdots n_N!}\left(\frac{\partial^{n_1+\cdots+n_N} f}{\partial x_1^{n_1}\cdots\partial x_N^{n_N}}\right)(a_1, \ldots, a_N). \tag{A.6}$$

Combining (A.5) and (A.6), we have

$$\begin{aligned}\delta(\mathbf{y}-\mathbf{x}) &= \delta(\mathbf{z}-\mathbf{x}+\mathbf{y}-\mathbf{z}) \\ &= \sum_{n_1=0}^{\infty}\cdots\sum_{n_N=0}^{\infty}\frac{1}{n_1!\ldots n_N!}\frac{(-\partial)^{n_1+\ldots+n_N}}{\partial x_1^{n_1}\ldots\partial x_N^{n_N}}(y_1-z_1)^{n_1}\ldots(y_N-z_N)^{n_N}\delta(\mathbf{z}-\mathbf{x}) \\ &= \frac{(-\partial)}{\partial x_1}(y_1-z_1)\delta(\mathbf{z}-\mathbf{x})+\frac{(-\partial)}{\partial x_2}(y_2-z_2)\delta(\mathbf{z}-\mathbf{x})+\ldots \\ &\quad+\frac{1}{2}\left[\frac{(-\partial)^2}{\partial x_1{}^2}(y_1-z_1)^2\delta(\mathbf{z}-\mathbf{x})+\frac{(-\partial)^2}{\partial x_1\partial x_2}(y_2-z_2)^2\delta(\mathbf{z}-\mathbf{x})+\ldots\right] \\ &\quad+\ldots\,. \end{aligned} \tag{A.7}$$

Now let us incorporate summation notation (i.e. we perform the summation over j_i indices without writing down the summation signs) to simplify the appearance of the expressions that will follow.

So,

$$\delta(\mathbf{y}-\mathbf{x})=\sum_{\nu=0}^{\infty}\frac{1}{\nu!}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\partial x_{j_2}\ldots\partial x_{j_\nu}}(y_{j_1}-z_{j_1})(y_{j_2}-z_{j_2})\ldots(y_{j_\nu}-z_{j_\nu})\delta(\mathbf{z}-\mathbf{x}). \tag{A.8}$$

Now the νth moment is defined by [66]

$$\begin{aligned}M^{(\nu)}_{j_1,j_2,\ldots,j_\nu}(\mathbf{z},t,\Delta t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}&(y_{j_1}-z_{j_1})(y_{j_2}-z_{j_2})\ldots(y_{j_\nu}-z_{j_\nu}) \\ &\times Q(\mathbf{y},t+\Delta t|\mathbf{z},t)\,dy_1dy_2\ldots dy_N,\end{aligned} \tag{A.9}$$

and then using (A.9) during the substitution of (A.8) in (A.3) gives

$$Q(\mathbf{x},t+\Delta t|\mathbf{z},t)=\left[1+\sum_{\nu=1}^{\infty}\frac{1}{\nu!}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\partial x_{j_2}\ldots\partial x_{j_\nu}}M^{(\nu)}_{j_1,j_2,\ldots,j_\nu}(\mathbf{z},t,\Delta t)\right]\delta(\mathbf{z}-\mathbf{x}). \tag{A.10}$$

We then define

$$\frac{M^{(\nu)}_{j_1,j_2,\ldots,j_\nu}(\mathbf{x},t,\Delta t)}{\nu!}=D^{(\nu)}_{j_1,j_2,\ldots,j_\nu}(\mathbf{x},t)\Delta t+\mathcal{O}(\Delta t^2). \tag{A.11}$$

Using (A.11) whilst substituting (A.10) into (A.1) gives

$$
\begin{aligned}
p(\mathbf{x}, t+\Delta t) \; = \; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\left[1+\sum_{\nu=1}^{\infty}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\ldots\partial x_{j_\nu}}\frac{M^{(\nu)}_{j_1,\ldots,j_\nu}}{\nu!}(\mathbf{z},t,\Delta t)\right] \\
& \times\delta(\mathbf{z}-\mathbf{x})p(\mathbf{z},t)\,dz_1dz_2\ldots dz_N \qquad \text{(A.12)}
\end{aligned}
$$

$$
\begin{aligned}
= \; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p(\mathbf{z},t)\delta(\mathbf{z}-\mathbf{x})\,dz_1dz_2\ldots dz_N \\
& +\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\sum_{\nu=1}^{\infty}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\ldots\partial x_{j_\nu}}D^{(\nu)}_{j_1,\ldots,j_\nu}(\mathbf{z},t)\Delta t \\
& \times\,\delta(\mathbf{z}-\mathbf{x})p(\mathbf{z},t)dz_1dz_2\ldots dz_N \qquad \text{(A.13)}
\end{aligned}
$$

and using the following property of the Dirac-delta function,

$$\int_{-\infty}^{\infty} f(t)\delta(t-a)\,dt = f(a), \qquad \text{(A.14)}$$

which gives

$$p(\mathbf{x},t+\Delta t)-p(\mathbf{x},t)=\sum_{\nu=1}^{\infty}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\ldots\partial x_{j_\nu}}D^{(\nu)}_{j_1,\ldots,j_\nu}(\mathbf{x},t)\Delta t\cdot p(\mathbf{x},t) \qquad \text{(A.15)}$$

dividing the resulting equation by Δt and taking the limit as $\Delta t \to 0$ gives the forward Kramers-Moyal expansion for N variables [9],

$$\frac{\partial p(\mathbf{x},t)}{\partial t}=\sum_{\nu=1}^{\infty}\frac{(-\partial)^{\nu}}{\partial x_{j_1}\partial x_{j_2}\ldots\partial x_{j_\nu}}D^{(\nu)}_{j_1,\ldots,j_\nu}(\mathbf{x},t)p(\mathbf{x},t). \qquad \text{(A.16)}$$

The solution of (A.16) with initial condition [9],

$$p(\mathbf{z},t')=Q(\mathbf{x},t'|\mathbf{z},t')=\delta(\mathbf{x}-\mathbf{z}) \qquad \text{(A.17)}$$

is the transition probability Q.

For a process with δ-correlated Gaussian noise all coefficients $D^{(\nu)}$ with $\nu \geq 3$ vanish. This is apparent from the properties of Gaussian white noise,

$$
\begin{aligned}
\overline{\xi(t)} &= 0, \\
\overline{\xi(t')\xi(t'')} &= 2\tilde{D}\delta(t'-t''). \qquad \text{(A.18)}
\end{aligned}
$$

and Isserlis' theorem [9],

$$\overline{\xi(t_1)\xi(t_2)\ldots\xi(t_{2\nu})} = \sum\prod_{i_r>i_s}\overline{\eta_{i_r}\eta_{i_s}}, \tag{A.19}$$

$$\overline{\eta(t_1)\eta(t_2)\ldots\eta(t_{2\nu+1})} = 0. \tag{A.20}$$

For $\nu = 2$ for example,

$$\begin{aligned}\overline{\eta(t_1)\eta(t_2)\ldots\eta(t_{2\nu})} &= 4\tilde{D}^2\{\delta(t_1-t_2)\delta(t_3-t_4) \\ &\quad +\delta(t_1-t_3)-\delta(t_2-t_4)+\delta(t_1-t_4)\delta(t_2-t_3)\},\end{aligned} \tag{A.21}$$

which gives rise to a $D^{(4)}$ of order $(\Delta t)^2$.

From (A.20) we see $D^{(2\nu+1)}$ are all zero, and from (A.18) and (A.19) we have $D^{(2\nu)} \sim (\Delta t)^{\nu}$, and then

$$\lim_{\Delta t\to 0}\frac{1}{\Delta t}D^{(2\nu)} = 0, \qquad \nu > 1. \tag{A.22}$$

However, if the noise is not Gaussian, higher order terms must be included in the expansion and we no longer have the Fokker-Planck equation.

So we now have that the probability density then satisfies

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = -\sum_{i=1}^{N}\frac{\partial}{\partial x_i}\left[D_i^1(\mathbf{x})p(\mathbf{x},t)\right] + \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2}{\partial x_i x_j}\left[D_{ij}^2(\mathbf{x})p(\mathbf{x},t)\right], \tag{A.23}$$

with, from (A.9) and (A.11),

$$D_i^{(1)}(\mathbf{x},t) = \lim_{\Delta t\to 0}\frac{1}{\Delta t}\langle\xi_i(t+\Delta t)-\xi_i(t)\rangle\bigg|_{\xi_k(t)=x_k}, \tag{A.24}$$

$$\begin{aligned}D_{ij}^{(2)}(\mathbf{x},t) &= D_{ji}(\mathbf{x},t) \\ &= \frac{1}{2}\lim_{\Delta t\to 0}\frac{1}{\Delta t}\langle[\xi_i(t+\Delta t)-\xi_i(t)]\,[\xi_j(t+\Delta t)-\xi_j(t)]\rangle\bigg|_{\xi_k(t)=x_k}.\end{aligned} \tag{A.25}$$

To make these equations more tractable, consider the general Langevin equations [9],

$$\dot{\xi}_i = \lambda_i(\boldsymbol{\xi},t) + \mu_{ij}(\boldsymbol{\xi},t)\eta_j(t), \tag{A.26}$$

with Gaussian white noise $\eta_j(t)$, then the corresponding integral equations are

$$\xi_i(t+\Delta t) - x_i = \int_t^{t+\Delta t} [\lambda_i(\boldsymbol{\xi}(t'), t') + \mu_{ij}(\boldsymbol{\xi}(t'), t')\eta_j(t')]\, dt'. \tag{A.27}$$

If we substitute the Taylor expansions of the functions λ and μ,

$$\lambda_i(\boldsymbol{\xi}(t'), t') = \lambda_i(\mathbf{x}, t') + \left[\frac{\partial}{\partial x_k}\lambda_i(\mathbf{x}, t')\right](\xi_k(t') - x_k) + \dots, \tag{A.28}$$

$$\mu_{ij}(\boldsymbol{\xi}(t'), t') = \mu_{ij}(\mathbf{x}, t') + \left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x}, t')\right](\xi_k(t') - x_k) + \dots, \tag{A.29}$$

into (A.27) gives

$$\begin{aligned}\xi_i(t+\Delta t) - x_i &= \int_t^{t+\Delta t} \lambda_i(\mathbf{x}, t')\, dt' + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\lambda_i(\mathbf{x}, t')\right](\xi_k(t') - x_k)\, dt' + \dots \\ &\quad + \int_t^{t+\Delta t} \mu_{ij}(\mathbf{x}, t')\eta_j(t')\, dt' + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x}, t')\right]\eta_j(t')(\xi_k(t') - x_k)\, dt' \\ &\quad + \dots .\end{aligned} \tag{A.30}$$

We can then iterate $(\xi_k(t') - x_k)$ components to give

$$\begin{aligned}\xi(t+\Delta t) - x_i &= \int_t^{t+\Delta t} \lambda_i(\mathbf{x}, t')\, dt' \\ &\quad + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\lambda_i(\mathbf{x}, t')\right] \int_t^{t'} \lambda_k(\mathbf{x}, t'')\, dt'\, dt'' \\ &\quad + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\lambda_i(\mathbf{x}, t')\right] \int_t^{t'} \mu_{kj}(\mathbf{x}, t'')\eta_j(t'')\, dt'\, dt'' \\ &\quad + \dots \\ &\quad + \int_t^{t+\Delta t} \mu_{ij}(\mathbf{x}, t')\eta_j(t')\, dt' \\ &\quad + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x}, t')\right] \int_t^{t'} \lambda_k(\mathbf{x}, t'')\eta_j(t')\, dt''\, dt' \\ &\quad + \int_t^{t+\Delta t} \left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x}, t')\right] \int_t^{t'} \mu_{kj}(\mathbf{x}, t'')\eta_j(t'')\eta_j(t')\, dt''\, dt' \\ &\quad + \dots .\end{aligned} \tag{A.31}$$

By taking the average of (A.31) and using the properties (A.18) and (A.14) we have

$$\begin{aligned}\overline{\xi_i(t+\Delta t)-x_i} &= \int_t^{t+\Delta t}\lambda_i(\mathbf{x},t')\,dt' \\ &\quad+\int_t^{t+\Delta t}\left[\frac{\partial}{\partial x_k}\lambda_i(\mathbf{x},t')\right]\int_t^{t'}\lambda_k(\mathbf{x},t'')\,dt''\,dt' \\ &\quad+\ldots \\ &\quad+\int_t^{t+\Delta t}\left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x},t')\right]\int_t^{t'}\mu_{kj}(\mathbf{x},t'')2\tilde{D}\delta(t'-t'')\,dt''\,dt' \\ &\quad+\ldots\,. \end{aligned} \tag{A.32}$$

Now using the following properties of the Dirac-delta function,

$$\begin{aligned}\int_0^a f(t)\delta(t-a)\,dt &= \frac{1}{2}f(a) \\ \int_t^{t'}\delta(t'-t'')\,dt'' &= \int_0^{t'-t}\delta(x)dx \quad = \quad \frac{1}{2}, \qquad t<t''<t'\end{aligned} \tag{A.33}$$

we have

$$\int_t^{t'}\mu_{kj}(\mathbf{x},t'')2D\delta(t'-t'')\,dt''=\tilde{D}\mu_{kj}(\mathbf{x},t'), \tag{A.34}$$

so that (we ignore integrals that will give a contribution of the form $(\Delta t)^\nu$ which will tend to zero in the limit $\Delta t\to 0$)

$$\begin{aligned}D_i^{(1)}(\mathbf{x},t) &= \lambda_i(\mathbf{x},t)+\lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}\left[\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x},t')\right]D\mu_{kj}(\mathbf{x},t') \\ &= \lambda_i(\mathbf{x},t)+\tilde{D}\mu_{kj}(\mathbf{x},t)\frac{\partial}{\partial x_k}\mu_{ij}(\mathbf{x},t).\end{aligned} \tag{A.35}$$

Similarly for $D_{ij}^{(2)}$ we have

$$\begin{aligned}[\xi_i(t+\Delta t)-x_i]^2 &= \left(\int_t^{t+\Delta t}\lambda_i(\mathbf{x},t')\,dt'\right)^2 \\ &\quad+2\int_t^{t+\Delta t}\lambda_i(\mathbf{x},t')\,dt'\int_t^{t+\Delta t}\mu_{ij}(\mathbf{x},t')\eta_j(t')\,dt' \\ &\quad+\left(\int_t^{t+\Delta t}\mu_{ik}(\mathbf{x},t')\eta_j(t')\,dt'\right)^2+\ldots\,.\end{aligned} \tag{A.36}$$

Using Fubini's theorem [68],

$$\left(\int_a^b f(x)\,dx\right)^2=\int_a^b\int_a^b f(x)f(y)\,dxdy, \tag{A.37}$$

then,

$$\begin{aligned}
[\xi_i(t+\Delta t) - x_i]^2 \;\; = \;\; & \int_t^{t+\Delta t} \int_t^{t+\Delta t} \lambda_i(\mathbf{x}, t')\lambda_i(\mathbf{x}, t'')\, dt'\, dt'' \\
& + 2\int_t^{t+\Delta t} \lambda_i(\mathbf{x}, t')\, dt' \int_t^{t+\Delta t} \mu_{ij}(\mathbf{x}, t')\eta_j(t')\, dt' \\
& + \int_t^{t+\Delta t} \int_t^{t+\Delta t} \mu_{ik}(\mathbf{x}, t')\mu_{jk}(\mathbf{x}, t'')\eta_j(t')\eta_j(t'')\, dt'\, dt'' \\
& + \ldots . \qquad \text{(A.38)}
\end{aligned}$$

Again neglecting terms of the order $(\Delta t)^\nu$ gives

$$\overline{[\xi_i(t+\Delta t) - x_i]^2} = \int_t^{t+\Delta t} \int_t^{t+\Delta t} \mu_{ik}(\mathbf{x}, t')\mu_{jk}(\mathbf{x}, t'')2\tilde{D}\delta(t' - t'')\, dt'\, dt''. \qquad \text{(A.39)}$$

Then,

$$D_{ij}^{(2)}(\mathbf{x}, t) = \tilde{D}\mu_{ik}(\mathbf{x}, t)\mu_{jk}(\mathbf{x}, t). \qquad \text{(A.40)}$$

Using the coefficients (A.40) and (A.40), in (A.23) gives us the Fokker-Planck equation.

B Appendix - A Modified Box-Muller Transformation

Random number generators on computers are typically uniformly distributed random processes. To apply the transformations of random processes in Section 3, it is necessary to transform a uniformly distributed process, to a Gaussian random process. This is achieved using a Box-Muller transformation. This method produces a Gaussian random process with mean 0 and variance 1. Here, we explain how one can achieve a Gaussian random process with mean μ and variance σ^2, using a modified Box-Muller transformation.

If u_1 and u_2 are independent and uniformly distributed in the range 0 to 1, then consider y_1 and y_2 given by [57]

$$y_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2)\sigma + \mu, \tag{B.1}$$

$$y_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2)\sigma + \mu. \tag{B.2}$$

This gives a modified Box-Muller transformation. It converts a 2-dimensional uniform distribution to a bivariate Gaussian distribution with mean μ and variance σ^2. Let us verify this.

First let us rearrange (B.1), then

$$u_1 = \exp\left[-\frac{1}{2}\left(\frac{y_1-\mu}{\cos(2\pi u_2)\sigma}\right)^2\right], \tag{B.3}$$

and rearranging (B.2),

$$u_1 = \exp\left[-\frac{1}{2}\left(\frac{y_2-\mu}{\sin(2\pi u_2)\sigma}\right)^2\right]. \tag{B.4}$$

Equating the right hand sides of these equations gives

$$\begin{aligned}
\frac{y_2-\mu}{\sin(2\pi u_2)\sigma} &= \frac{y_1-\mu}{\cos(2\pi u_2)\sigma} \\
\Rightarrow \sin(2\pi u_2)\sigma(y_1-\mu) &= \cos(2\pi u_2)\sigma(y_2-\mu) \\
\Rightarrow \tan(2\pi u_2) &= \frac{y_2-\mu}{y_1-\mu} \\
\Rightarrow u_2 &= \frac{1}{2\pi}\arctan\left(\frac{y_2-\mu}{y_1-\mu}\right).
\end{aligned} \tag{B.5}$$

Substituting this into (B.1) gives

$$u_1 = \exp\left[-\frac{1}{2}\left(\frac{y_1-\mu}{\sigma\cos\left(\arctan\left(\frac{y_2-\mu}{y_1-\mu}\right)\right)}\right)^2\right], \tag{B.6}$$

then using the trigonometric identity

$$\cos(\arctan(x)) = \frac{1}{\sqrt{1+x^2}}, \tag{B.7}$$

gives

$$\begin{aligned} u_1 &= \exp\left[-\frac{1}{2}\left((y_1-\mu)\Big/\left(\frac{\sigma}{\sqrt{1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}}}\right)\right)^2\right] \\ &= \exp\left[-\frac{1}{2}\left(\frac{(y_1-\mu)\sqrt{1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}}}{\sigma}\right)^2\right] \\ &= \exp\left[-\frac{1}{2}\left(\frac{(y_1-\mu)^2\left(1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}\right)}{\sigma^2}\right)\right] \\ &= \exp\left[-\frac{1}{2}\left(\frac{(y_1-\mu)^2+(y_2-\mu)^2}{\sigma^2}\right)\right]. \end{aligned} \tag{B.8}$$

Now the fundamental transformation law of probabilities states that [63]

$$|p(y)dy| = |p(x)dx|, \tag{B.9}$$

where $p(x)dx$ is the probability that x lies between x and $x+dx$, and $p(y)dy$ is the probability that y lies between y and $y+dy$. The extension of this into n dimensions is [65]

$$p(Y)dY = p(X)J(X,Y)dY, \tag{B.10}$$

where $p(X)$ is the joint probability distribution of X with $X=(x_1,x_2,\ldots,x_n)$, $p(Y)$ is the joint probability density of Y with $Y=(y_1,y_2,\ldots,y_n)$, $dY = dy_1dy_2\ldots dy_n$ and $J(X,Y)$ is the Jacobian determinant. Hence,

$$\begin{aligned}
\frac{\partial(u_1,u_2)}{\partial(y_1,y_2)} &= \begin{vmatrix} \frac{\partial u_1}{\partial y_1} & \frac{\partial u_1}{\partial y_2} \\ \frac{\partial u_2}{\partial y_1} & \frac{\partial u_2}{\partial y_2} \end{vmatrix} \\
&= -\frac{(2y_1-2\mu)\exp\left[-\frac{1}{2}\frac{(y_1-\mu)^2+(y_2-\mu)^2}{\sigma^2}\right]}{4\sigma^2\pi(y_1-\mu)\left(1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}\right)} \\
&\quad - \frac{(2y_2-2\mu)(y_2-\mu)\exp\left[\frac{1}{2}\frac{(y_1-\mu)^2+(y_2-\mu)^2}{\sigma^2}\right]}{4\sigma^2\pi(y_1-\mu)^2\left(1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}\right)} \\
&= -\frac{\exp\left[-\frac{1}{2}\frac{(y_1-\mu)^2+(y_2-\mu)^2}{\sigma^2}\right]}{4\sigma^2\pi}\left[\frac{2y_1-2\mu}{(y_1-\mu)\left(1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}\right)}+\frac{(2y_2-2\mu)(y_2-\mu)}{(y_1-\mu)^2\left(1+\frac{(y_2-\mu)^2}{(y_1-\mu)^2}\right)}\right] \\
&= -\frac{1}{2\sigma^2\pi}\left[\exp\left(-\frac{(y_1-\mu)^2}{2\sigma^2}\right)\right]\left[\exp\left(-\frac{(y_2-\mu)^2}{2\sigma^2}\right)\right]. \qquad \text{(B.11)}
\end{aligned}$$

Since this is the product of a function of y_2 alone and a function of y_1 alone, we see that each y is independently distributed according to the Gaussian distribution.

C Appendix - Andronov-Hopf Bifurcation of a Saddle-Focus Fixed Point

Here, we illustrate the changes to the stable and unstable manifolds of the unstable saddle-focus fixed point when complex-conjugate pairs of eigenvalues cross the imaginary axis (in an Andronov-Hopf bifurcation) before it occurs at the neighbouring stable fixed points, mentioned in section 5.3.1. For this to occur, recall that the stable fixed point undergoes an Andronov-Hopf bifurcation at

$$\tau = -\frac{\pi}{2J_{min}}, \tag{C.1}$$

where τ is the time delay, and J_{min} is the value of the Jacobian evaluated at the stable fixed point (for stable fixed points $J < 0$). Since we are considering 1-D gradient systems with delay the Jacobian will be a constant (Section 5.3.1).

For the unstable saddle-focus fixed point to undergo an Andronov-Hopf bifurcation

$$\tau = \frac{3\pi}{2J_{max}}, \tag{C.2}$$

where J_{max} is the value of the Jacobian evaluated at the unstable fixed point (for stable fixed points $J > 0$).

To have any effect on the system dynamics, the saddle-focus must undergo its Andronov-Hopf bifurcation before the periodic orbit around the originally stable equilibrium collides with it, i.e. undergoes a homoclinic bifurcation (Section 5.3.2). From (C.1) and (C.2), we can see that for this to occur J_{max} must be significantly larger than $-J_{min}$.

We sketch (in 3-D) the key geometrical objects in the phase space to illustrate the impact of the Andronov-Hopf bifurcation of the saddle-focus occurring before the homoclinic loop with the nearby cycle. If this occurs, a saddle cycle is formed by the unstable manifold (Before: Fig. 47; After: Fig. 48).

In 3-D there is no possibility of the homoclinic bifurcation because the unstable manifold of the saddle-cycle serves as a barrier preventing the nearby stable cycle from collision with the fixed point (Fig. 48). However, in an infinite dimensional phase space (as we will have in the delay system) there is no longer a barrier and the periodic orbit can easily collide with the fixed point. Here, the only difference from the "simple" case in the main text (Section 5.3.2), is the saddle-focus fixed point is *more* unstable, which means it is more likely to give rise to the birth of the non-attracting chaotic set as seen in the "non-simple" case (although this is

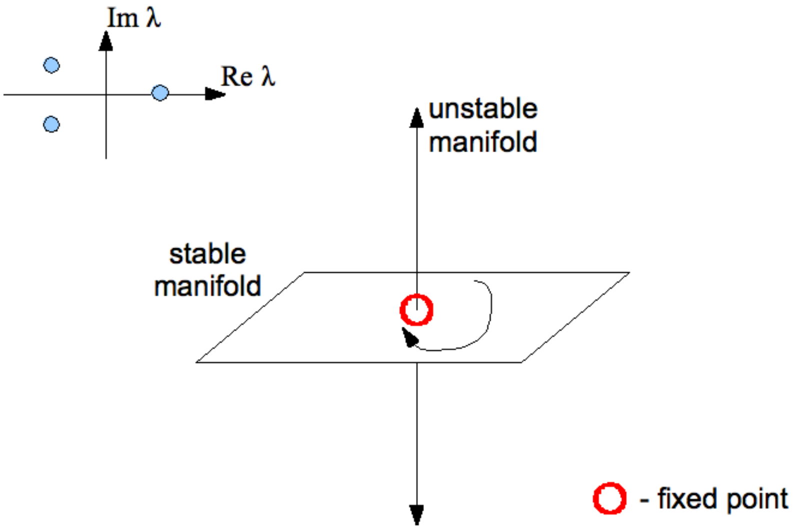


Figure 47: The stable and unstable manifolds (in 3-D) of the saddle-focus fixed point (red), with a positive real eigenvalue and a pair of negative complex conjugate eigenvalues (top left). This is a sketch of these objects before any bifurcation takes place.

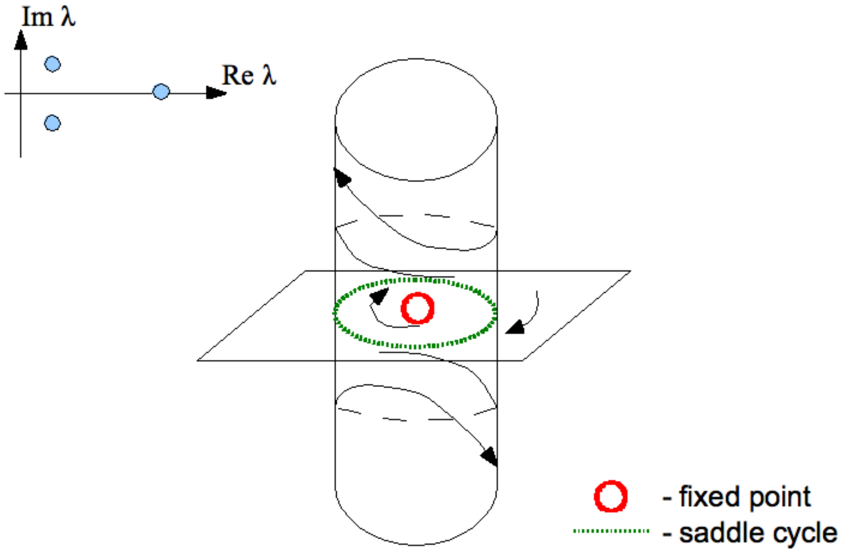


Figure 48: The stable and unstable manifolds (in 3-D) of the saddle-focus fixed point (red) after it undergoes an Andronov-Hopf bifurcation. The unstable manifold is now 3-dimensional in the form of a saddle-cycle. Nearby trajectories are attracted towards the fixed point, but the cylindrical unstable manifold acts as a barrier.

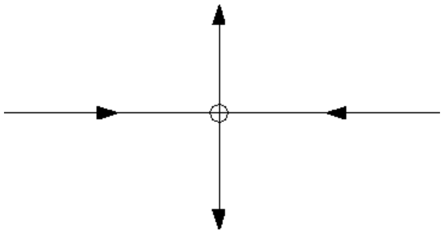


Figure 49: Poincaré section of the saddle-focus fixed point in Fig. 47. Since the fixed point is 1-dimensional, it is not visible in the section.

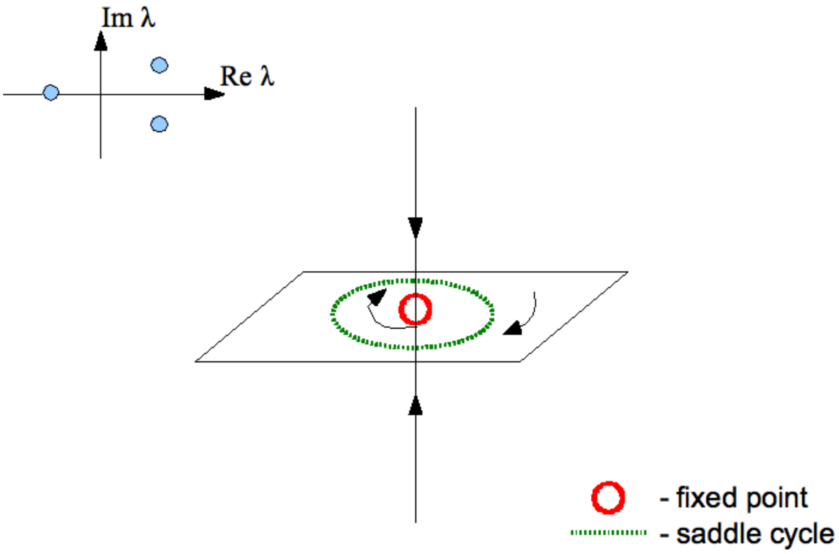


Figure 50: The stable and unstable manifolds (in 3-D) of the saddle-focus fixed point (red), with a negative real eigenvalue and a pair of positive complex conjugate eigenvalues (top left). This is a sketch of these objects before any bifurcation takes place.

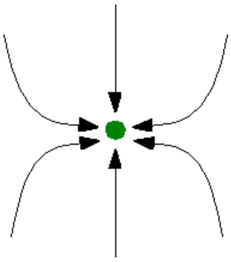


Figure 51: Poincaré section of the saddle-focus fixed point in Fig. 50. Since we have an attracting saddle-cycle, it is represented by a point (green) in the section.

not guaranteed).

In summary, both the simple and non-simple scenarios remain valid if the local maximum in the non-delayed gradient system is much steeper than the neighbouring minima, however, if this is the case it is more likely that a non-attracting chaotic set will be born immediately after the homoclinic loop is formed.

References

[1] P.A. Absil and K. Kurdyka. On the stable equilibrium points of gradient systems. *Systems & control letters*, 55(7):573–577, 2006.

[2] V.S. Afraimovich and L.P. Shilnikov. Invariant two-dimensional tori, their breakdown and stochasticity. *Amer. Math. Soc. Transl*, 149(2):201–212, 1991.

[3] V.S. Anishchenko and T.E. Vadivasova. *"Lectures on statistical radiophysics" (Lektsii po statisticheskoi radiofizike).* Saratov State University Publishing, 1992.

[4] M.A. Arbib. *The handbook of brain theory and neural networks.* The MIT Press, 2003.

[5] M.R. Berthold and J. Diamond. Constructive training of probabilistic neural networks. *Neurocomputing*, 19(1-3):167–183, 1998.

[6] T.V.P. Bliss and T. Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, 1973.

[7] R. Bracewell. *The Fourier Transform and Its Applications (2nd ed.).* McGraw-Hill, 1986.

[8] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, MA:, 2006.

[9] W. Coffey, Y.P. Kalmykov, and J.T. Waldron. *The Langevin equation: with applications to stochastic problems in physics, chemistry, and electrical engineering.* World Scientific Pub Co Inc, 2004.

[10] R.V. Culshaw and S. Ruan. A delay-differential equation model of hiv infection of cd4+ t-cells* 1. *Mathematical biosciences*, 165(1):27–39, 2000.

[11] P. Dayan. Reinforcement learning. In R.A. Wilson and F. Keil, editors, *Encyclopedia of Cognitive Science*, pages 715–717. England: MacMillan Press., 2001.

[12] R. Descartes. *Treatise on man.* 1648.

[13] O. Diekmann. *Delay equations: functional-, complex-, and nonlinear analysis*, volume 110. Springer, 1995.

[14] A.E. Dubinov and I.D. Dubinova. How can one solve exactly some problems in plasma theory. *Journal of plasma physics*, 71(5):715, 2005.

[15] R. Fitzhugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466, 1961.

[16] J.L. Flanagan. *Speech analysis synthesis and perception (2nd ed.).* Springer-Verlag, Berlin - New York, 1972.

[17] C.A. Floudas and C.E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[18] H.M. Gibbs. Optical bistability: controlling light with light. 1985.

[19] C.G. Gross. Genealogy of the grandmother cell. *The Neuroscientist*, 8(5):512–518, 2002.

[20] J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer-Verlag, 1997.

[21] S. Guillouzic, I. LHeureux, and A. Longtin. Rate processes in a delayed, stochastically driven, and overdamped system. *Physical Review E*, 61(5):4906, 2000.

[22] K. Gurney. Lecture notes. Lecture 5: Associative Memories: The Hopfield Net, Brunel University.

[23] J.K. Hale and S.M.V. Lunel. *Introduction to functional differential equations*, volume 99. Springer, 1993.

[24] P. Hänggi. Stochastic Processes I: Asymptotic behaviour and symmetries. *Helv. Phys. Acta*, 51:183–201, 1978.

[25] D.O. Hebb. The organisation of behaviour, 1949.

[26] E.M.T. Hendrix and G. Boglárka. *Introduction to Nonlinear and Global Optimization.* Springer Cambridge, UK, 2010.

[27] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley Publishing Company, 1991.

[28] G.E. Hinton. Connectionist learning procedures. *Artificial intelligence*, 40(1):185–234, 1989.

[29] M.W. Hirsch, S. Smale, and R.L. Devaney. *Differential equations, dynamical systems, and an introduction to chaos.* Academic Press Boston, 2004.

[30] A.L. Hodgkin and A.F. Huxley. Propagation of electrical signals along giant nerve fibres. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 140(899):177–183, 1952.

[31] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554, 1982.

[32] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088, 1984.

[33] J.J. Hopfield and D.W. Tank. neural computation of decisions in optimization problems. *Biological cybernetics*, 52(3):141–152, 1985.

[34] J.J. Hopfield and D.W. Tank. Computing with neural circuits: A model. *Science*, 233(4764):625–633, 1986.

[35] N.B. Janson and C.J. Marsden. Memory foam approach to unsupervised learning. *Arxiv 1107.0674*, 2011.

[36] N.B. Janson and C.J. Marsden. Self-shaping dynamical systems and learning. *Arxiv 1111.4443*, 2011.

[37] E.G. Jones. Golgi, cajal and the neuron doctrine. *Journal of the History of the Neurosciences*, 8(2):170–178, 1999.

[38] S. Kirkpatrick, C.D. Gelatt Jr, and M.P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[39] L.A. Klein. *Sensor and data fusion: a tool for information assessment and decision making*, volume 138. Society of Photo Optical, 2004.

[40] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer-Verlag, 1992.

[41] E. Knobloch and N.O. Weiss. Bifurcations in a model of double-diffusive convection. *Physics Letters A*, 85(3):127–130, 1981.

[42] I.U.A. Kuznetsov. *Elements of applied bifurcation theory*, volume 112. Springer Verlag, 1998.

[43] C. Laing and G.J. Lord. *Stochastic methods in neuroscience*. Oxford University Press, 2009.

[44] Y. Le Cun. Learning process in an asymmetric threshold network. *Disordered systems and biological organization*, pages 233–240, 1986.

[45] C. Li, X. Liao, and J. Yu. Hopf bifurcation in a prototype delayed system. *Chaos, Solitons & Fractals*, 19(4):779–787, 2004.

[46] K.E. Lonngren and E.W. Bai. On the ucar prototype model. *Int. Journal of Engineering Science*, 40:1855–1857, 2002.

[47] H. Lu. Chaotic attractors in delayed neural networks. *Physics Letters A*, 298(2-3):109–116, 2002.

[48] M.C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[49] A. N. Malakhov. Time scales of overdamped nonlinear brownian motion in arbitrary potential profiles. *Chaos*, 7:488–504, 1997.

[50] J. Mallet-Paret and G.R. Sell. The poincaré-bendixson theorem for monotone cyclic feedback systems with delay. *Journal of Differential Equations*, 125(2):441–489, 1996.

[51] J. Mallet-Paret and G.R. Sell. Systems of differential delay equations: Floquet multipliers and discrete lyapunov functions. *Journal of differential equations*, 125(2):385–440, 1996.

[52] S.F. Mason. *A History of the Sciences*. Collier Books New York, 1962.

[53] A.M. Mathai and P.N. Rathie. *Probability and statistics*. Macmillan, 1977.

[54] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, 1943.

[55] M. Minsky and S. Papert. Perceptrons. 1969.

[56] W. Morris. *The American heritage dictionary of the English language*, volume 2007. Houghton Mifflin Company, 2000.

[57] M.A. Murison. A method for directly generating a Gaussian distribution with nonunit variance and nonzero mean from uniform random deviates, 2000.

[58] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[59] S. Novak and R.G. Frehlich. Transition to chaos in the duffing oscillator. *Physical Review A*, 26(6):3660, 1982.

[60] S. Ochs. *A history of nerve functions: from animal spirits to molecular mechanisms.* Cambridge Univ Pr, 2004.

[61] D.B. Parker. Learning-logic (TR-87). *Center for Computational Research in Economics and Management Science, MIT*, 1985.

[62] M. Peng. Bifurcation and chaotic behavior in the euler method for a uçar prototype delay model. *Chaos, Solitons & Fractals*, 22(2):483–493, 2004.

[63] A.D. Polyanin and A.I. Chernoutsan. *A concise handbook of mathematics, physics, and engineering sciences.* CRC Press, 2010.

[64] Y. Pomeau and P. Manneville. Intermittent transition to turbulence in dissipative dynamical systems. *Communications in Mathematical Physics*, 74(2):189–197, 1980.

[65] W.H. Press. *Numerical recipes: the art of scientific computing.* Cambridge Univ Pr, 2007.

[66] H. Risken. *The Fokker-Planck equation: Methods of solution and applications.* Springer Verlag, 1996.

[67] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

[68] J.S. Rosenthal. *A first look at rigorous probability theory.* World Scientific Pub Co Inc, 2006.

[69] D.J. Rothman, S. Marcus, and S.A. Kiceluk. *Medicine and Western civilization.* Rutgers Univ Pr, 1995.

[70] M.R. Roussel. Lecture notes for chemistry 5850: Nonlinear dynamics. Lecture 11: Delay-Differential Equation, University of Lethbridge, 2005.

[71] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[72] D.W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization.* Wiley, New York, 1992.

[73] N.M. Seel. *Encyclopedia of the Sciences of Learning.* Springer Verlag, 2011.

[74] L.F. Shampine, I. Gladwell, and S. Thompson. *Solving ODEs with MATLAB.* Cambridge Univ Press, 2003.

[75] C.A. Shaw, J.C. McEachern, and J. McEachern. *Toward a theory of neuroplasticity.* Psychology Pr, 2001.

[76] G.M. Shepherd. *Foundations of the neuron doctrine*, volume 352. Oxford Univ Press, 1991.

[77] L.P. Shilnikov, A.L. Shilnikov, D.V. Turaev, and L.O. Chua. *Methods of qualitative theory in nonlinear dynamics, Part 2*, volume 5. World Scientific Pub Co Inc, 2001.

[78] H. Shinozaki. Lambert-w function approach to stability and stabilization problems for linear time-delay systems. 2008.

[79] C. Sparrow. *The Lorenz equations: bifurcations, chaos, and strange attractors.* Springer-Verlag, 1982.

[80] D.F. Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

[81] J.C. Sprott. A simple chaotic delay differential equation. *Physics Letters A*, 366(4-5):397–402, 2007.

[82] R.L. Stratonovich. *Topics in the theory of random noise.* Gordon and Breach, 1967.

[83] F. Takens. *Dynamical Systems and Turbulence, Detecting strange attractors in turbulence, eds Rand D and Young L-S.* Lecture Notes in Mathematics 898, 1981.

[84] A.M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[85] A. Uçar. A prototype model for chaos studies. *International journal of engineering science*, 40(3):251–258, 2002.

[86] A. Uçar. On the chaotic behaviour of a prototype delayed dynamical system. *Chaos, Solitons & Fractals*, 16(2):187–194, 2003.

[87] P.J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Unpublished doctoral dissertation, Harvard University. Cited on*, page 33, 1974.

[88] A. A. Wolf. An ergodic theorem and its generalization. *Journal of the Franklin Institute*, 283(4):286–299, 1967.

[89] M. Zak. Terminal attractors for addressable memory in neural networks. *Physics Letters A*, 133(1):18–22, 1988.

[90] M. Zak. Creative dynamics approach to neural intelligence. *Biological cybernetics*, 64(1):15–23, 1990.

[91] M. Zak. Self-supervised dynamical systems. *Chaos, Solitons & Fractals*, 19(3):645–666, 2004.

[92] M. Zak. From reversible thermodynamics to life. *Chaos, Solitons & Fractals*, 26(4):1019–1033, 2005.

[93] M. Zak. Physics of life from first principles. *EJTP*, 4:11–96, 2007.

[94] M. Zak and N. Toomarian. Unsupervised learning in neurodynamics using the phase velocity field approach. In *Advances in neural information processing systems 2*, pages 583–589. Morgan Kaufmann Publishers Inc., 1990.

[95] O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The finite element method for fluid dynamics, 5th edition.* Butterworth-Heinemann, 2000.