

Using Echo State Networks to Approximate Value Functions for Control

Allen G. Hart^a, Kevin R. Olding^a, Alexander M.G. Cox^a, Olga Isupova^b, Jonathan H.P. Dawes^a

^a*Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK*

^b*Department of Computer Science, University of Bath, Bath, BA2 7AY, UK*

Abstract

An Echo State Network (ESN) is a type of single-layer recurrent neural network with randomly-chosen internal weights and a trainable output layer. We prove under mild conditions that a sufficiently large Echo State Network can approximate the value function of a broad class of stochastic and deterministic control problems. Such control problems are generally non-Markovian.

We describe how the ESN can form the basis for novel and computationally efficient reinforcement learning algorithms in a non-Markovian framework. We demonstrate this theory with two examples. In the first, we use an ESN to solve a deterministic, partially observed, control problem which is a simple game we call ‘Bee World’. In the second example, we consider a stochastic control problem inspired by a market making problem in mathematical finance. In both cases we can compare the dynamics of the algorithms with analytic solutions to show that even after only a single reinforcement policy iteration the algorithms arrive at a good policy.

Keywords: Liquid State Machines, Reservoir Computing, Stochastic Optimal Control, Mathematical Finance, Reinforcement Learning

PACS: 0000, 1111

2000 MSC: 0000, 1111

1. Introduction

An Echo State Network (ESN) is a special type of single-layer recurrent neural network introduced at the turn of the millennium by [1] and [2] to study time series. Training is fast because the training step involves only the selection of weights in the output layer rather than updating the internal weights in the recurrent layer. Furthermore, the simple formulation of ESNs renders them amenable to mathematical analysis. Given a time series z_k (where k is the discrete time index) of d -dimensional data points, an ESN is set up as follows. We randomly generate a $n \times n$ reservoir matrix \mathbf{A} , a $n \times d$ input matrix \mathbf{C} and a $n \times 1$ bias vector ζ . Then we iteratively generate a sequence of n -dimensional *reservoir state* vectors x_k according to

$$x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k + \zeta)$$

where $\sigma(x)_i = \max(0, x_i)$ is the rectified linear unit (ReLU) activation function applied component-wise to the n -dimensional vector x . Observe that the k th reservoir state x_k depends on all past data-points \dots, z_{k-2}, z_{k-1} and therefore captures non-Markovian temporal correlations in the data. If the 2-norm of the reservoir matrix satisfies $\|\mathbf{A}\|_2 < 1$ then as n tends to infinity, the influence on the reservoir state x_{k+n} of the data points \dots, z_{k-2}, z_{k-1} in the distant past becomes arbitrarily small. This is called the *fading memory property* and is closely related to the *echo state property* (ESP) introduced in the context of ESNs by [1]. The ESP is the statement that the sequence of reservoir states $(x_k)_{k \in \mathbb{Z}}$ is, for a given input data sequence $(z_k)_{k \in \mathbb{Z}}$, uniquely determined. We can interpret the reservoir state vectors as the latent vectors which encode the infinite past observations in lower dimensional form.

When an ESN has the ESP, it can be applied to a class of supervised learning problems where we have a time series of d dimensional data points r_k , called *targets*, that depend on all previous input time series data $\dots, z_{k-3}, z_{k-2}, z_{k-1}$ and we seek to learn the relationship between the sequence of past states and the target for each k . We can train an ESN to solve this problem by finding the $m \times d$ matrix W that minimises

$$\sum_{k=0}^{\ell-1} \|W^\top x_k - r_k\|^2 + \lambda \|W\|^2,$$

where ℓ is the number of labelled data points, and $\lambda > 0$ is the Tikhonov regularisation (a.k.a. ridge regression) parameter. Throughout this paper, $\|\cdot\|$ denotes the matrix 2-norm, vector 2-norm or absolute value, depending on whether the input is a matrix, vector, or scalar, respectively.

This minimisation problem can be solved using regularised linear least squares regression, and hence we can both obtain W quickly, and guarantee that W is the global optimum. This compares extremely favourably with training a (deep) neural network with stochastic gradient descent and backpropagation which takes considerably longer, and may not converge to the global optimum [3].

Despite the training procedure being entirely linear, ESNs are universal approximators, and can therefore model arbitrarily complex relationships between the sequence of past data

points and the targets. This is made formal in a recent result by [4] that we review here and then build on. We emphasise that not only are ESNs theoretically very promising, they have performed remarkably well in practice on problems ranging from seizure detection, to robot control, handwriting recognition, and financial forecasting, where ESNs have won competitions [5], [6], [7], [8]. Impressively, ESNs outperformed RNNs and LSTMs at a chaotic time series prediction task by a factor of over 2400 [9]. ESNs have also proved themselves competitive in various tasks in reinforcement learning [10] and control [11].

Even in cases where practitioners prefer to use other recurrent neural networks (RNNs), such as Long Short Term Memory networks (LSTMs), the rigorous theory of ESNs should prove useful in architecture design. In [12], it is shown that different deep neural network architectures can be ranked by randomly initialising the internal weights and training only the outer weights by linear regression. Once the best performing architecture (with random internal weights) has been identified, the authors then train the internal weights of the highest ranking architecture. This is much faster than training the internal weights (a nonlinear problem) for every architecture. The ranking of architectures with random internal weights closely approximates the ranking of architectures with optimised internal weights. From our point of view, the authors are essentially approximating fully trained networks with (non-recurrent) ESNs.

In a sequence of papers, [13], [14], and [4] recently analysed ESNs in the context of nonlinear filters and functionals. Roughly speaking, a *filter* U is a map from a bi-infinite sequence $\dots, z_{-2}, z_{-1}, z_0, z_1, z_2, \dots$ of real vectors to another bi-infinite sequence of real vectors $\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots$, and a *functional* H maps a bi-infinite sequence $\dots, z_{-2}, z_{-1}, z_0, z_1, z_2, \dots$ of real vectors to a single real vector or number. We can view an ESN as a filter that maps an input sequence $\dots, z_{-2}, z_1, z_0, z_1, z_2, \dots$ to a reservoir sequence $\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots$, or a functional that maps $\dots, z_{-2}, z_1, z_0, z_1, z_2, \dots$ to the lone reservoir state x_0 . The theory of filters and functionals is therefore a natural theoretical setting for ESNs. Within this theory, this paper presents three novel results.

Our first result assumes that we have a time series of data z_k and a set of targets r_k that depend on all previous data points \dots, z_{k-2}, z_{k-1} via a functional \mathcal{R} which sends infinite sequences of data points to targets. We then have a supervised learning problem of finding the relationship between the data and targets. In the special case that $z_k = r_k$, this problem is *time series forecasting*. Our first novel result states that if we have sufficiently many data points z_k , drawn from a stationary, ergodic, and bounded process \mathbf{Z} , which need not be Markovian, and we obtain W using regularised linear least squares, then a sufficiently large ESN will approximate, as closely as required, the functional \mathcal{R} sending inputs \dots, z_{k-2}, z_{k-1} to the targets r_k .

This result has applications in the *statistical inference of dynamical systems*, which was recently reviewed by [15]. This area of research is especially focused on statistical inference (i.e. learning) of stationary ergodic processes. Furthermore, we can use this result in the context of reinforcement learning (RL) and optimal control. We envisage an agent operating under a given policy in the parlance of reinforcement learning or control in the parlance of control theory that generates a sequence of (reward, action, observation) triples $z_k = (r_k, a_k, \omega_k)$. Then the functional V that maps previous (reward, action, observation) triples \dots, z_{k-2}, z_{k-1}

to rewards z_k models the reward functional arbitrarily well. The set up does not assume the RL problem is Markovian, and allows for a continuous state space.

Our second novel result generalises the first, and encompasses the case where the functional V is the value functional of a stochastic control process, or Partially Observed Markov Decision Process (POMDP). By training an ESN to approximate the value functional, we establish a stepping stone toward developing an offline reinforcement learning algorithm supported by an ESN that can solve a large class of control problems. Moreover, since ESNs are recurrent, they can be used for non-Markovian problems, where a reinforcement learning agent must exploit its *memory* of past observations, actions and rewards. Our third result is presented in the context of building an online reinforcement algorithm that can, under certain conditions, determine the optimal value function for a given policy.

These results are part of a general push to take machine learning ideas typically applied to (partially observed) Markov processes and generalising them to hold on stationary ergodic processes. We can see for example [16] consider to clustering problems typically defined Markov processes applied to stationary ergodic processes.

We demonstrate some of these theoretical results numerically on two examples. The first is a deterministic game which we call ‘Bee World’. The goal of the game for the bee is to navigate a time varying distribution of nectar in order to maximise the total future discounted value of the nectar acquired over all future time. The optimal trajectory can be found explicitly via the calculus of variations but the constraint that the bee has a maximum speed of flight leads to unexpectedly complicated solution paths; it therefore provides a straightforward but not entirely trivial control problem. Since the bee does not have access to the entire state space, and only observes the nectar it collects at each moment in time, the problem is therefore a partially observed Markov Decision Process which requires memory of the past to solve. We demonstrate how a simple and easily-configurable reinforcement learning algorithm supported by an ESN can learn to play Bee World with respectable skill.

The second numerical example is inspired by a market making problem in mathematical finance. The mathematical formulation of this problem reduces to a seeking to control a one dimensional Brownian motion so that it stays near the origin. The cost of straying from the origin is quadratic in the distance from the origin, and the cost of applying a push toward the origin is quadratic in the strength of the push. The market maker must therefore balance the cost of applying the control against the cost of allowing the motion to drift too far from the origin. We briefly discuss the financial motivation for this problem, then solve it analytically in continuous and discrete time. The set up most commonly seen in the literature is continuous time, but only in discrete time is the problem suitable for an ESN. We then compare the optimal discrete time solution to a solution learned by a reinforcement learning agent supported by an ESN.

Finally, we note that our approach to the Market making problem is loosely related to the recent paper by [11] who introduce QuaSiModO: Quantization-Simulation-Modeling-Optimization. These authors analyse the interplay between the following four aspects:

1. Quantising the action space \mathcal{A} .
2. Simulating a system under a given control/policy.

3. Modelling the full system given a partial/full observation of the state space.
4. Optimising the control/policy.

The structure of the remainder of the paper closely follows the summary of results presented above. In section 2 we set up the mathematical formalism for ESNs that we wish then to approximate. Section 3 introduces our novel theoretical results, while sections 4 and 5 respectively present applications to the deterministic ('Bee World'), and then the stochastic ('market maker') optimal control problems. We conclude in section 6.

2. Background

In this section, we introduce the theory and notation of nonlinear filters (in relation to ESNs) developed by [13], [14], and [4]. First, we denote by $(\mathbb{R}^d)^{\mathbb{Z}}$ the set of maps with domain \mathbb{Z} and codomain \mathbb{R}^d . This is the set of bi-infinite \mathbb{R}^d -valued real sequences.

A *filter* is a map $U : (\mathbb{R}^d)^{\mathbb{Z}} \rightarrow (\mathbb{R}^n)^{\mathbb{Z}}$. A filter U is called *causal* if inputs from the past and present $\dots, z_{-2}, z_{-1}, z_0$ contribute to $U(z)$ but states in the future $z_1, z_2 \dots$ do not. More formally U is casual if $\forall z, y \in (\mathbb{R}^d)^{\mathbb{Z}}$ that satisfy $z_k = y_k \forall k \leq 0$ it follows that $U(z) = U(y)$. We define the time shift filter $T : (\mathbb{R}^d)^{\mathbb{Z}} \rightarrow (\mathbb{R}^d)^{\mathbb{Z}}$ by $T(z)_k = T(z)_{k+1}$ which we interpret as the map that steps forward one unit of time. A filter U is called *time invariant* if U commutes with the time shift operator T . If U is causal and time invariant filter then we call U a causal time invariant (CTI) filter.

A functional is a map $H : (\mathbb{R}^d)^{\mathbb{Z}} \rightarrow \mathbb{R}^n$. In [14] it is shown that there is a bijection between the space of CTI filters and the space of functionals. To see this, take a functional H and define the k th term of the associated filter U via $U(z)_k = HT^k(z)$. Conversely, given a filter U , the associated functional H is given by $H(z) = U(z)_0$

We can view an ESN as a CTI filter from the space of input sequences $\dots, z_{-1}, z_0, z_1, \dots$ to the space of reservoir sequences $\dots, x_{-1}, x_0, x_1, \dots$. To make this connection between ESNs and filters formal, we will first present a generalisation of an Echo State Network called a reservoir system.

Definition 2.1. (Reservoir system) Let $F : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^n$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}^s$. Then we call the following system of equations

$$\begin{aligned} x_{k+1} &= F(x_k, z_k) \\ r_k &= h(x_k) \end{aligned} \tag{1}$$

a reservoir system.

Remark 2.2. We can see that if

$$\begin{aligned} F(x, z) &= \sigma(\mathbf{A}x + \mathbf{C}z + \boldsymbol{\zeta}) \\ h(x) &= W^\top x \end{aligned}$$

then we retrieve an ESN with $n \times n$ reservoir matrix \mathbf{A} , $n \times d$ input matrix \mathbf{C} , bias vector $\boldsymbol{\zeta} \in \mathbb{R}^n$, linear output layer $W \in \mathbb{R}^n$, and activation function $\sigma = \text{ReLU}$, defined in the introduction.

We require that the reservoir system induces a *unique* filter from the input sequence to the reservoir sequence. This property is the Echo State Property that we briefly mentioned in the introduction.

Definition 2.3. (Echo State Property [1]) A reservoir system has the Echo State Property (ESP) if for any $(z_k)_{k \in \mathbb{Z}} \in (\mathbb{R}^d)^{\mathbb{Z}}$ there exists a unique $(x_k)_{k \in \mathbb{Z}} \in (\mathbb{R}^n)^{\mathbb{Z}}$ that satisfy the equations of the reservoir system (1).

To any reservoir system with the Echo State property we can associate a unique CTI reservoir filter $U : (\mathbb{R}^d)^{\mathbb{Z}} \rightarrow (\mathbb{R}^n)^{\mathbb{Z}}$ defined by $U(z) = x$. To this reservoir filter, we may assign a CTI reservoir functional $H : (\mathbb{R}^m)^{\mathbb{Z}} \rightarrow \mathbb{R}^d$ defined by $H(z) = x_0$. In a supervised learning context, we have a time series of data points $\dots, z_{-2}, z_{-1}, z_0$ and a time series of targets \dots, r_{-1}, r_0 that each depend on *all* previous data points. The output functional $h \circ H : (\mathbb{R}^d)^{\mathbb{Z}} \rightarrow \mathbb{R}$ is the map we use to approximate the relationship between the data and the targets, so $h \circ H(\dots, z_{-2}, z_{-1}, z_0, z_1, z_2, \dots) \approx r_k$. Note that $h \circ H$ is *causal*, so does not peer into the future and use data z_1, z_2, \dots that have not yet been revealed. When the reservoir system is an ESN, the map h is the linear map W^\top obtained by least squares ridge regression, so that $W^\top H(\dots, z_{-2}, z_{-1}, z_0, z_1, \dots) \approx r_k$. We assume there exists a *true* map from the data to the targets that we label $\mathcal{R} : \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}$ so that $\mathcal{R}(\dots, z_{-2}, z_{-1}, z_0, z_1, \dots) = r_k$. Our goal is to find W such that $W^\top H \approx \mathcal{R}$.

Definition 2.4. (ESN filter and functional) If an ESN has the ESP then we will write $H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$ to denote the reservoir functional associated to an ESN with parameters \mathbf{A}, \mathbf{C} and $\boldsymbol{\zeta}$. We will also write $H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$ to denote the output functional $W^\top H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$ (defined by left multiplication of $H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$ by the linear readout layer)

Next, we will present a procedure, introduced by [4], for randomly generating the ESN's internal weights \mathbf{A}, \mathbf{C} and biases $\boldsymbol{\zeta}$, which ensures the ESN has ESP and allows for the universal approximation of target functionals \mathcal{R} . The procedure differs from the procedure commonly seen in the literature, where $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ are populated with i.i.d Gaussians, or i.i.d uniform deviates, and then \mathbf{A} is rescaled so that its 2-norm (or spectral radius) is less than 1. Furthermore, the procedure introduced by [4] depends on some details of the input process, which must satisfy mild conditions stated below.

Definition 2.5. (Admissible input process) A $(\mathbb{R}^d)^{\mathbb{Z}}$ valued random variable \mathbf{Z} is called an admissible process if for any $T \in \mathbb{N}$ there exists $M_T > 0$ such that for all $k \in \mathbb{Z}$

$$\|\mathbf{Z}_{k-T}, \mathbf{Z}_{k-T+1}, \dots, \mathbf{Z}_k\| \leq M_T \tag{2}$$

Lebesgue-almost surely.

We will now present a procedure by which the matrices $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ are randomly generated.

Procedure: Initialising the random weights of an ESN.

Let $N \in \mathbb{N}$, $R > 0$ be the input parameters for the procedure. Suppose that \mathbf{Z} is an admissible input process. Consequently, for any $T_0 \in \mathbb{N}$ there exists M_{T_0} such that (for $k = 0$ in (2))

$$\|\mathbf{Z}_{-T_0}, \mathbf{Z}_{-T_0+1}, \dots, \mathbf{Z}_0\| \leq M_T$$

Lebesgue-almost surely. Then, for a given T_0 , we initialise the ESN reservoir matrix \mathbf{A} , input matrix \mathbf{C} , and biases $\boldsymbol{\zeta}$ according to the following procedure.

1. Draw N i.i.d. samples $\mathbf{A}_1, \dots, \mathbf{A}_N$ from the uniform distribution on $B_R \subset \mathbb{R}^{d(T_0+1)}$ where B_R is the ball of radius R and centre 0, and draw N i.i.d. samples $\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_N$ from the uniform distribution on $[-\max(M_{T_0}R, 1), \max(M_{T_0}R, 1)]$.
2. Let S and c be shift matrices defined

$$S = \begin{bmatrix} 0_{d,dT_0} & 0_{d,d} \\ I_{dT_0} & 0_{dT_0,d} \end{bmatrix} \quad c = \begin{bmatrix} I_d \\ 0_{dT_0,d} \end{bmatrix}$$

and set

$$\mathbf{a} = \begin{bmatrix} \mathbf{A}_1^\top \\ \mathbf{A}_2^\top \\ \vdots \\ \mathbf{A}_N^\top \end{bmatrix} \quad \bar{\mathbf{A}} = \begin{bmatrix} S & 0_{d(T_0+1),N} \\ \mathbf{a}S & 0_{N,N} \end{bmatrix}$$

$$\bar{\mathbf{C}} = \begin{bmatrix} c \\ \mathbf{a}c \end{bmatrix} \quad \bar{\boldsymbol{\zeta}} = \begin{bmatrix} 0_{d(T_0+1)} \\ \boldsymbol{\zeta}_1 \\ \vdots \\ \boldsymbol{\zeta}_N \end{bmatrix}$$

so that

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} & -\bar{\mathbf{A}} \\ -\bar{\mathbf{A}} & \bar{\mathbf{A}} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \bar{\mathbf{C}} \\ -\bar{\mathbf{C}} \end{bmatrix} \quad \boldsymbol{\zeta} = \begin{bmatrix} \bar{\boldsymbol{\zeta}} \\ -\bar{\boldsymbol{\zeta}} \end{bmatrix}.$$

We are now ready to present the key result by [4], (which generalises a result by [17]) and which holds in the following supervised learning context. Given time series data z_k (from an admissible process \mathbf{Z}) and a time series of targets r_k depending on all previous data \dots, z_{k-2}, z_{k-1} we wish to approximate the functional that sends \dots, z_{k-2}, z_{k-1} to r_k . We will denote this functional \mathcal{R} . The problem of approximating \mathcal{R} given the data and targets is a supervised learning problem. The result can be summarised as follows. Suppose we have an ESN with weights \mathbf{A}, \mathbf{C} and biases $\boldsymbol{\zeta}$ randomly generated by procedure 1. Then, the ESN admits a linear readout matrix W for which the ESN equipped with the matrix W (denoted $H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$) approximates the relationship \mathcal{R} between data points \dots, z_{k-2}, z_{k-1} and targets r_k

as closely as is required.

Theorem 2.6 ([4]). *Suppose that \mathbf{Z} is an admissible input process. Let $\mathcal{R} : (D_n)^\mathbb{Z} \rightarrow \mathbb{R}$ (where D_n is a compact subset of \mathbb{R}^n) be CTI and measurable with respect to some measure μ such that $\mathbb{E}_\mu[|\mathcal{R}(\mathbf{Z})|^2] < \infty$.*

Then for any $\epsilon > 0$ and $\delta \in (0, 1)$ there exists $N, T_0 \in \mathbb{N}$, $R > 0$ such that, with probability $(1 - \delta)$, the ESN with parameters $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ generated by the procedure in definition 1 (with inputs N, T_0, R) has the ESP and admits a readout layer $W \in \mathbb{R}^{2(d(T_0+1)+N)}$ such that

$$\left(\mathbb{E}_\mu \left[\left\| H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) - \mathcal{R}(\mathbf{Z}) \right\|^2 \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \right)^{1/2} := \left(\int_{(\mathbb{R}^d)^\mathbb{Z}} \left\| H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(z) - \mathcal{R}(z) \right\|^2 d\mu(z) \right)^{1/2} < \epsilon.$$

3. Novel results for ESNs

Theorem 2.6 is an existence result stating that *there exists* a linear readout layer W yielding an arbitrarily good approximation. Our first novel contribution is to strengthen the result under additional assumptions. The new result states that, given a sufficiently large ESN and sufficiently many training data z_k drawn from a stationary, ergodic and bounded process \mathbf{Z} , if we train an ESN using regularised least squares then the arbitrarily good readout layer W *will* be attained (with probability as close to 1 as desired). This result is analogous to the main result by [18] who prove a similar theorem for ESNs trained on deterministic inputs. Before we introduce the result we will present the definition of a stationary process, an ergodic process, and the ergodic theorem.

Definition 3.1. (Stationary Process [15]) A stochastic process $(\mathbf{Z}_k)_{k \in \mathbb{Z}} \equiv \mathbf{Z}$ is stationary if for any $\ell \in \mathbb{N}$ and finite subset $I \subset \mathbb{Z}$ the joint distribution $(\mathbf{Z}_i)_{i \in I}$ is equal to the joint distribution $(\mathbf{Z}_{i+\ell})_{i \in I}$.

Definition 3.2. (Stationary Ergodic Process [15]) A stationary stochastic process $(\mathbf{Z}_k)_{k \in \mathbb{Z}} \equiv \mathbf{Z}$ is called ergodic if for every $\ell \in \mathbb{N}$ and every pair of Borel sets A, B

$$\begin{aligned} \lim_{\ell \rightarrow \infty} \frac{1}{\ell} \sum_{k=0}^{\ell-1} \mathbb{P} \left((\mathbf{Z}_1, \dots, \mathbf{Z}_\ell) \in A, (\mathbf{Z}_k, \dots, \mathbf{Z}_{k+\ell}) \in B \right) \\ = \mathbb{P} \left((\mathbf{Z}_1, \dots, \mathbf{Z}_\ell) \in A \right) \mathbb{P} \left((\mathbf{Z}_1, \dots, \mathbf{Z}_\ell) \in B \right). \end{aligned}$$

Every stationary ergodic processes Z satisfies the celebrated Ergodic Theorem.

Theorem 3.3. (Ergodic Theorem) *If $(\mathbf{Z}_k)_{k \in \mathbb{Z}} \equiv \mathbf{Z}$ is a stationary ergodic process then for any $i \in \mathbb{Z}$*

$$\mathbb{E}_\mu[\mathbf{Z}_i] = \lim_{\ell \rightarrow \infty} \frac{1}{\ell} \sum_{k=0}^{\ell-1} \mathbf{Z}_{i+k}$$

almost surely.

Our result holds in the following supervised learning context. Given time series data z_k (from an admissible, stationary, ergodic, bounded process \mathbf{Z}) and a time series of targets r_k depending on all previous data \dots, z_{k-2}, z_{k-1} we wish to approximate the mapping from \dots, z_{k-2}, z_{k-1} to r_k . This mapping is denoted \mathcal{R} . Our result states that an ESN with weights \mathbf{A}, \mathbf{C} and biases $\boldsymbol{\zeta}$ randomly generated by the procedure in definition 1, which is fed the training data z_k , and then trained by regularised least squares, will yield a matrix W . This ESN equipped with the matrix W (denoted $H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}$) will approximate the relationship \mathcal{R} between data points \dots, z_{k-2}, z_{k-1} and targets r_k as closely as required.

Theorem 3.4. *Suppose that \mathbf{Z} is an admissible input process, that is also stationary and ergodic, with invariant measure μ . Let $\mathcal{R} : (D_n)^\mathbb{Z} \rightarrow \mathbb{R}$ (where D_n is a compact subset of \mathbb{R}^n) be CTI, μ -measurable, and satisfy $\mathbb{E}_\mu[|\mathcal{R}(\mathbf{Z})|^2] < \infty$. Let z be an arbitrary realisation of \mathbf{Z}*

Then for any $\epsilon > 0$ and $\delta \in (0, 1)$ there exist $N, T_0 \in \mathbb{N}$, $R > 0$, $\lambda^ > 0$ and $\ell \in \mathbb{N}$ such that the ESN with parameters $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ generated by the procedure in Definition 1 (with inputs N, T_0, R), and $W_\ell^* \in \mathbb{R}^{2(d(T_0+1)+N)}$ which minimises (over $W \in \mathbb{R}^{2(d(T_0+1)+N)}$) the least squares problem*

$$\frac{1}{\ell} \sum_{k=0}^{\ell-1} \left\| H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^{-k}(z) - \mathcal{R} T^{-k}(z) \right\|^2 + \lambda \|W\|^2,$$

(where $\lambda \in (0, \lambda^*)$) satisfies with probability $(1 - \delta)$ the inequality

$$\mathbb{E}_\mu \left[\left\| H_{W_\ell^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) - \mathcal{R}(\mathbf{Z}) \right\|^2 \middle| \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] < \epsilon.$$

Proof. Later in this paper, we state and prove a more general result (Theorem 3.6) which reduces to this result in the special case $\gamma = 0$. \square

In summary, we have stated that for any $\epsilon > 0$ and $\delta \in (0, 1)$ there exists an ESN of dimension $n = 2(d(T_0 + 1) + N)$ with output layer W trained by the Tikhonov-regularised least squares procedure against ℓ training points, whose output functional approximates the target arbitrarily closely with arbitrarily high probability. The theorem is (sadly) non constructive in the sense that the number of neurons n , number of training points ℓ and regularisation parameter λ^* are not computed for a given ϵ and δ . Ideally, we would establish uniform bounds on the number of number of neurons n and data points ℓ required for an approximation with tolerance ϵ to hold with probability δ . Though less ideal, one could establish an asymptotic order of convergence using the central limit theorem (CLT). The CLT (roughly) states that the error between the time average and the space average of a stationary ergodic process converges in law to a normal distribution with standard deviation of the order $1/\sqrt{\ell}$ as the number of data points ℓ grows to infinity. The CLT is stated below.

Theorem 3.5. *(Central Limit Theorem [15]) If $(\mathbf{Z}_k)_{k \in \mathbb{Z}}$ is a stationary ergodic process then there exists a covariance matrix Σ such that for any $i \in \mathbb{Z}$ and Borel set A*

$$\lim_{\ell \rightarrow \infty} \mathbb{P} \left(\frac{1}{\sqrt{\ell}} \sum_{k=0}^{\ell-1} (\mathbf{Z}_{i+k} - \mathbb{E}_\mu[\mathbf{Z}_i]) \in A \right) = \mathbb{P}(\mathcal{N}(0, \Sigma) \in A).$$

In other words, the random variables

$$\frac{1}{\sqrt{\ell}} \sum_{k=0}^{\ell-1} (\mathbf{Z}_{i+k} - \mathbb{E}_{\mu}[\mathbf{Z}_i])$$

converge in distribution to the multivariate normal $\mathcal{N}(0, \Sigma)$ as $\ell \rightarrow \infty$.

This suggests that the approximation of the target functional \mathcal{R} also converges with order $1/\sqrt{\ell}$ as the number of data points increases. Furthermore, related results by [4] use the CLT to establish uniform bounds on the number of neurons $n = 2(d(T+1) + N)$ required for a given approximation. This strongly suggests that the approximation in Theorem 3.4 converges with order $1/\sqrt{N}$.

We will now pivot towards our second novel result, which generalises the first. Suppose that we have a contraction mapping Φ on the space of functionals, and we seek a W^* such that the ESN functional $H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}$ approximates the unique fixed point of Φ . The existence of the unique fixed point is guaranteed by Banach's fixed point theorem. Finding the fixed point of a contraction mapping has applications in reinforcement learning because the optimal value function (and optimal quality function) of a Markov Decision Process (MDP) is a fixed point of a *Bellman operator*. The theory we are presenting here can be viewed as a generalisation of an MDP because the input processes we are considering may have long time correlations (violating the Markov property) which can only be recognised by filters with sufficiently long and robust memories; like Echo State Networks.

We can observe first of all if Φ is the constant map $\Phi(H) = \mathcal{R}$, then Φ is clearly a contraction mapping with fixed point \mathcal{R} . In this case, the problem is exactly the same as that solved by Theorem 3.4. We are especially interested in the case of Φ taking the form of the *Bellman Value operator*. To make this formal, we will consider a stationary ergodic process \mathbf{Z} with invariant measure μ . Then we define the map $T_{\mathbf{Z}}$ as a CTI filter on the bi-infinite sequences $(D_N)^{\mathbb{Z}}$, which returns the random variable:

$$T_{\mathbf{Z}}(z)_k = \begin{cases} T_{\mathbf{Z}}(z)_{k+1} & \text{if } k < 0 \\ \mathbf{Z}_{k+1} \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 & \text{if } k \geq 0. \end{cases}$$

Next, we introduce $\mathcal{R} : (D_N)^{\mathbb{Z}} \rightarrow \mathbb{R}$ as the CTI reward functional, giving a reward (or expectation over a distribution of rewards) to an agent that has observed a given sequence of (reward, action, observation) triples. We let $\gamma \in [0, 1)$ denote the discount factor, and define the operator

$$\Phi(H)(z) := \mathcal{R}(z) + \gamma \mathbb{E}_{\mu}[HT_{\mathbf{Z}}(z)]. \quad (3)$$

In this case, Φ is a contraction mapping with Lipschitz constant γ . With this, we will define the CTI value functional $V : (D_N)^{\mathbb{Z}} \rightarrow \mathbb{R}$ (with respect to the process \mathbf{Z}) as

$$V(z) := \mathbb{E}_{\mu} \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 \right].$$

The value functional V takes a sequence of (reward, action, observation) triples and returns the expected discounted sum of future rewards. Furthermore, the value function V is the unique fixed point of the Bellman operator Φ . Re-arranging the definition of $V(z)$ above, we have that:

$$\begin{aligned}
V(z) &= \mathbb{E}_\mu \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 \right] \\
&= \mathbb{E}_\mu \left[\sum_{k=1}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 \right] + \mathcal{R}(z) \\
&= \gamma \mathbb{E}_\mu \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^{k+1}(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 \right] + \mathcal{R}(z) \\
&= \gamma \mathbb{E}_\mu \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j < 0 \right] + \mathcal{R}(z)
\end{aligned}$$

where we have carried out straightforward relabellings of the indexing of terms in the sum by k . Then by the law of total expectation we may write this last expression as

$$\begin{aligned}
V(z) &= \gamma \mathbb{E}_\mu \left[\mathbb{E}_\mu \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = T_{\mathbf{Z}}(z)_j \ \forall j \leq 0 \right] \right] + \mathcal{R}(z) \\
&= \gamma \mathbb{E}_\mu [V T_{\mathbf{Z}}(z)] + \mathcal{R}(z) = \Phi(V)(z),
\end{aligned}$$

which shows that V is indeed a fixed point of Φ , and so is the unique such, since Φ is a contraction.

Our goal is now to seek a W^* such that the ESN functional $H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}$ closely approximates the unique fixed point V of Φ . One approach is to collect a dataset from a single training trajectory, and then perform least squares regression to find W^* . This is an example of *offline* learning (in the reinforcement learning parlance) because the training occurs after the data has been collected. This is in contrast to *online* learning where training takes place dynamically as new data becomes available. We will make this offline approach formal in the following theorem.

Theorem 3.6. *Suppose that \mathbf{Z} is an admissible input process, that is also stationary and ergodic with invariant measure μ . Let $\mathcal{R} : (D_N)^{\mathbb{Z}} \rightarrow \mathbb{R}$ be μ -measurable and satisfy $\mathbb{E}[|\mathcal{R}(\mathbf{Z})|^2] < \infty$ and define Φ using (3) on the μ -measurable functionals H that satisfy $\mathbb{E}_\mu[|H(\mathbf{Z})|^2] < \infty$. Let $\gamma \in [0, 1)$. Let z be an arbitrary realisation of \mathbf{Z}*

Then for any $\epsilon > 0$, $\delta \in (0, 1)$ there exists $N, T_0 \in \mathbb{N}, R, \lambda^ > 0$ and $\ell \in \mathbb{N}$ such that the ESN with parameters $\mathbf{A}, \mathbf{C}, \zeta$ generated by procedure 1 (with inputs N, T_0, R), and $W_\ell^* \in \mathbb{R}^{2(d(T_0+1)+N)}$ minimising (over $W \in \mathbb{R}^{2(d(T_0+1)+N)}$) the least squares problem*

$$\frac{1}{\ell} \sum_{k=0}^{\ell-1} \left\| W^\top (H^{\mathbf{A}, \mathbf{C}, \zeta} T^{-k}(z) - \gamma H^{\mathbf{A}, \mathbf{C}, \zeta} T^{1-k}(z)) - \mathcal{R}(z) \right\|^2 + \lambda \|W\|^2$$

where $\lambda \in (0, \lambda^*)$, then with probability $(1 - \delta)$

$$\mathbb{E}_\mu \left[\left\| H_{W_\ell^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - \Phi H_{W_\ell^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \right\|^2 \middle| \mathbf{A}, \mathbf{C}, \zeta \right] < \epsilon.$$

Proof. First let V be the unique fixed point of the contraction mapping Φ whose existence and uniqueness is guaranteed by Banach's fixed point theorem. Denote the Lipschitz constant of Φ with the symbol τ . Then we fix $\epsilon > 0$ and $\delta \in (0, 1)$, then by Theorem 2.6 there exists with probability $(1 - \delta)$ a linear readout $W \in \mathbb{R}^{2(d(T_0+1)+N)}$ such that

$$\mathbb{E}_\mu \left[\left\| H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - V(\mathbf{Z}) \right\|^2 \middle| \mathbf{A}, \mathbf{C}, \zeta \right] < \frac{\epsilon}{5(1 + \tau)}. \quad (4)$$

Then it follows that

$$\begin{aligned} & \mathbb{E}_\mu \left[\left\| H_W^{\mathbf{A}, \mathbf{C}, \zeta} - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta} \right\|^2 \middle| \mathbf{A}, \mathbf{C}, \zeta \right] \\ &= \mathbb{E}_\mu \left[\left\| H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) + V(\mathbf{Z}) - V(\mathbf{Z}) \right\|^2 \middle| \mathbf{A}, \mathbf{C}, \zeta \right] \\ &\leq \mathbb{E}_\mu[\|H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - V(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \mathbb{E}_\mu[\|V(\mathbf{Z}) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\ &= \mathbb{E}_\mu[\|H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - V(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \mathbb{E}_\mu[\|\Phi V(\mathbf{Z}) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\ &\leq \mathbb{E}_\mu[\|H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - V(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \tau \mathbb{E}_\mu[\|V(\mathbf{Z}) - H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\ &= (1 + \tau) \mathbb{E}_\mu[\|V(\mathbf{Z}) - H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\ &< (1 + \tau) \frac{\epsilon}{5(1 + \tau)} \text{ by (4)} \\ &< \frac{\epsilon}{5} \end{aligned}$$

which yields the estimate

$$\mathbb{E}_\mu[\|H_W^{\mathbf{A}, \mathbf{C}, \zeta} - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}\|^2 | \mathbf{A}, \mathbf{C}, \zeta] < \frac{\epsilon}{5}. \quad (5)$$

Now, we can choose λ^* such that for any $\lambda \in (0, \lambda^*)$

$$\lambda \|W\|^2 < \frac{\epsilon}{5}. \quad (6)$$

Next we define a sequence of vectors $(W_j^*)_{j \in \mathbb{N}}$ by

$$W_j^* = \arg \min_{U \in \mathbb{R}^{2(d(T_0+1)+N)}} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|H_U^{\mathbf{A}, \mathbf{C}, \zeta} T^{-k}(z) - \gamma H_U^{\mathbf{A}, \mathbf{C}, \zeta} T^{1-k}(z) - \mathcal{R} T^{-k}(z)\|^2 + \lambda \|U\|^2 \right).$$

We may view $\arg \min$ as continuous map on the space of strictly convex C^1 functions that returns their unique minimiser. The regularised linear least squares problem is a strictly

convex C^1 problem, so we may define $W_\infty^* \in \mathbb{R}^{2d(T_0+1)+N}$ by

$$\begin{aligned}
W_\infty^* &:= \arg \min_U \left(\mathbb{E}_\mu[\|H_U^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H_U^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z}) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \lambda \|U\|^2 \right) \\
&= \arg \min_U \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|H_U^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H_U^{\mathbf{A},\mathbf{C},\zeta}T^{1-k} - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|U\|^2 \right) \\
&= \lim_{j \rightarrow \infty} \arg \min_U \left(\frac{1}{j} \sum_{k=0}^{j-1} \|H_U^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H_U^{\mathbf{A},\mathbf{C},\zeta}T^{1-k} - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|U\|^2 \right) \\
&= \lim_{j \rightarrow \infty} W_j^*
\end{aligned}$$

where the second and third equalities hold by the Ergodic Theorem and continuity of $\arg \min$ respectively. Now, we may choose $\ell \in \mathbb{N}$ sufficiently large that

$$\begin{aligned}
&\mathbb{E}_\mu[\|W_\ell^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\
&\quad - \mathbb{E}_\mu[\|W_\infty^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] < \frac{\epsilon}{5}, \quad (7)
\end{aligned}$$

and

$$\begin{aligned}
&\left| \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W_j^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|W_j^*\|^2 \right) \right. \\
&\quad \left. - \frac{1}{\ell} \sum_{k=0}^{\ell-1} \|W_\ell^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|W_\ell^*\|^2 \right| < \frac{\epsilon}{5}, \quad (8)
\end{aligned}$$

and by the Ergodic Theorem

$$\begin{aligned}
&\left| \frac{1}{\ell} \sum_{k=0}^{\ell-1} \|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}(z)\|^2 \right. \\
&\quad \left. - \lim_{j \rightarrow \infty} \frac{1}{j} \sum_{k=0}^{j-1} \|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}(z)\|^2 \right| < \frac{\epsilon}{5}. \quad (9)
\end{aligned}$$

Now the proof proceeds directly

$$\begin{aligned}
&\mathbb{E}_\mu[\|H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \Phi H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\
&\quad = \mathbb{E}_\mu[\|H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z}) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\
&\quad = \mathbb{E}_\mu[\|W_\ell^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta].
\end{aligned}$$

Then we apply (7) which yields

$$\begin{aligned}
&\mathbb{E}_\mu[\|H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \Phi H_{W_\ell^*}^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] \\
&\quad < \mathbb{E}_\mu[\|W_\infty^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \frac{\epsilon}{5}.
\end{aligned}$$

Then we apply the Ergodic Theorem

$$\begin{aligned}
& \mathbb{E}_\mu[\|W_\infty^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \frac{\epsilon}{5} \\
&= \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W_\infty^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 \right) + \frac{\epsilon}{5} \\
&\leq \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W_\infty^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 \right) + \lambda \|W_\infty^*\|^2 + \frac{\epsilon}{5} \\
&= \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W_j^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|W_j^*\|^2 \right) + \frac{\epsilon}{5} \\
&\hspace{15em} \text{then apply (8)} \\
&< \frac{1}{\ell} \sum_{k=0}^{\ell-1} \|W_\ell^{*\top}(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|W_\ell^*\|^2 + \frac{2\epsilon}{5} \\
&\leq \frac{1}{\ell} \sum_{k=0}^{\ell-1} \|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 + \lambda \|W\|^2 + \frac{2\epsilon}{5} \\
&\hspace{15em} \text{then apply (9)} \\
&< \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 \right) + \lambda \|W\|^2 + \frac{3\epsilon}{5} \\
&\hspace{15em} \text{then apply (6)} \\
&< \lim_{j \rightarrow \infty} \left(\frac{1}{j} \sum_{k=0}^{j-1} \|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}T^{-k}(z) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T^{1-k}(z)) - \mathcal{R}T^{-k}(z)\|^2 \right) + \frac{4\epsilon}{5} \\
&\text{Then apply the Ergodic Theorem again} \\
&= \mathbb{E}_\mu[\|W^\top(H^{\mathbf{A},\mathbf{C},\zeta}(\mathbf{Z}) - \gamma H^{\mathbf{A},\mathbf{C},\zeta}T(\mathbf{Z})) - \mathcal{R}(\mathbf{Z})\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \frac{4\epsilon}{5} \\
&= \mathbb{E}_\mu[\|H_W^{\mathbf{A},\mathbf{C},\zeta} - \Phi H_W^{\mathbf{A},\mathbf{C},\zeta}\|^2 | \mathbf{A}, \mathbf{C}, \zeta] + \frac{4\epsilon}{5} \\
&\text{then apply (5)} \\
&< \epsilon
\end{aligned}$$

□

3.1. Connection to Partially Observed Markov Decision Processes

Theorem 3.6 applies to a reinforcement learning scenario where the observations are a stationary and ergodic process. This includes the case where observations emerge from a partially observed, stationary and ergodic Markov decision process. These are themselves a special case of a partially observed Markov decision process (POMDP) which are a common scenario studied in the reinforcement learning community. In particular, the results in this

paper apply to POMDPs in the special case that the underlying Markov process is stationary and ergodic. However, there exist stationary ergodic processes, which satisfy the conditions of Theorem 3.6, which are not the output of any partially observed decision Markov process.

The approach that we set out in this paper has a lot in common with POMDPs, but there are some subtle differences which we will clarify here. First of all, the value function in this paper is defined in terms of the complete sequence of (reward, action, observation) triples, rather than the current belief state. One advantage of our approach is that a belief state does not need to be computed explicitly, nor do any assumptions need to be made about the relationship between the hidden state of the environment and the observations. In the setting of this paper, the reservoir states x_k (which are explicitly computed by evaluating $H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}T^k(z)$) can be interpreted as latent states, very much like the latent states for POMDPs. We also stress that the value function V and reservoir functionals $H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ and $H_{W_\ell}^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ are causal and time invariant (CTI) so we are never using future information that is unavailable in the present, despite the input sequences being bi-infinite. Indeed, one of the strengths of our approach is that the learning procedure will be able to learn the impact of any unobserved or hidden states via the latent states x_k and linear regression.

3.2. Training ESNs with online learning

In some reinforcement learning applications, it is useful - or even essential - for the optimisation of W to occur dynamically as new data comes in; such algorithms are called *online* learning algorithms. In this section, we will present and discuss some preliminary novel results surrounding online learning algorithms that use ESNs. We will first introduce a lemma, stating that, under reasonable conditions, the ODE

$$\frac{d}{dt}W = -h(W) := -\mathbb{E}_\mu \left[H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}(\mathbf{Z})(H_W^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}(\mathbf{Z}) - \Phi H_W^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}(\mathbf{Z})) \right] \quad (10)$$

converges exponentially quickly to a globally asymptotic fixed point W^* , for which the associated ESN functional $H_{W^*}^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ is *close* to the unique fixed point of Φ . By *close* we mean that the orthogonal projection of $\Phi H_{W^*}^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ onto the finite dimensional vector space of functionals $\{H_W^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}} \mid W \in \mathbb{R}^d\}$ is $H_{W^*}^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$. Unlike the previous result (Theorem 3.6) we do not need to assume that the contraction mapping satisfies $\Phi(H) = R + \gamma\mathbb{E}[HT_{\mathbf{Z}}]$. We could choose for example $\Phi(H) = R + \gamma \sup_\pi \mathbb{E}[HT_{\mathbf{Z}(\pi)}]$ where $\mathbf{Z}(\pi)$ is a process under a control π . The fixed point of this operator is the *optimal* value function V^* .

Lemma 3.7. *Let \mathbf{Z} be an admissible input process. Let $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ be a $n \times n$, $n \times d$, and $n \times 1$ dimensional random reservoir matrix, input matrix and bias vector. Let $H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ and $H_W^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}$ denote the associated ESN functionals. Let Φ be a contraction mapping, with Lipschitz constant $0 \leq \tau < 1$, on the space of CTI filters $H : (D_N)^\mathbb{Z} \rightarrow \mathbb{R}$ that are μ -measurable and satisfy $\mathbb{E}[H(\mathbf{Z})^2] < \infty$. Suppose further that $0 \leq \tau < \kappa^{-1}$ where κ is the condition number of the autocorrelation matrix*

$$\Sigma = \mathbb{E}_\mu [H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}}(\mathbf{Z})H^{\mathbf{A},\mathbf{C},\boldsymbol{\zeta}\top}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}].$$

Then there exists a $\delta > 0$ such that the ODE

$$\frac{d}{dt}W = -h(W) := -\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) (H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})) \mid \mathbf{A}, \mathbf{C}, \zeta \right]$$

satisfies

$$\frac{d}{dt}\|W - W^*\| \leq -\delta\|W - W^*\| \quad (11)$$

where W^* is a globally asymptotic fixed point. W^* enjoys the further property that

$$H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta} = \mathcal{P}\Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}$$

where \mathcal{P} denotes the $L^2(\mu)$ orthogonal projection operator on the μ -measurable filters H satisfying $\mathbb{E}[H(\mathbf{Z})^2] < \infty$ and is defined

$$\mathcal{P}H(z) := H^{\mathbf{A}, \mathbf{C}, \zeta \top}(z) \Sigma^{-1} \mathbb{E}_\mu [H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z})H(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta].$$

Proof. To show that W^* is a globally asymptotic fixed point it suffices to show that there exists a $\delta > 0$ such that

$$(W - W^*) \cdot (h(W^*) - h(W)) \leq -\delta\|(W - W^*)\|^2$$

as this implies

$$\frac{d}{dt}\|W - W^*\| \leq -\delta\|W - W^*\|.$$

To construct this δ , we first note that

$$h(W) = \Sigma W - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right]$$

so, by a direct computation we have

$$\begin{aligned}
& (W - W^*) \cdot (h(W^*) - h(W)) \\
&= (W - W^*) \cdot \left(\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] \right) \\
&\quad - (W - W^*) \cdot \left(\Sigma W - \Sigma W^* \right) \\
&= (W - W^*) \cdot \left(\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] \right) \\
&\quad - (W - W^*)^\top \Sigma (W - W^*) \\
&\leq (W - W^*) \cdot \left(\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] \right) \\
&\quad - \sigma \|W - W^*\|^2 \quad \text{where } \sigma \text{ is the smallest eigenvalue of } \Sigma \\
&= (W - W^*) \cdot \left(\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] \right) \\
&\quad - \sigma \|W - W^*\|^2 \\
&\leq (W - W^*) \cdot \left(\mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) H_W^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) - H^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \zeta \right] \right) \tau \\
&\quad - \sigma \|W - W^*\|^2 \quad \text{because } \tau \text{ is the Lipschitz constant for } \Phi \\
&= \tau (W - W^*)^\top \Sigma (W - W^*) - \sigma \|W - W^*\|^2 \\
&\leq \tau \rho \|W - W^*\|^2 - \sigma \|W - W^*\|^2 \quad \text{where } \rho \text{ is the largest eigenvalue of } \Sigma \\
&= -(\sigma - \tau \rho) \|W - W^*\|^2,
\end{aligned}$$

so we can set $\delta := \sigma - \tau \rho$ and notice $\delta > 0$ because $0 \leq \tau < \kappa^{-1} = \sigma / \rho$. Next, to show that

$$H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta} = \mathcal{P} \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \zeta}$$

we observe that since W^* is an equilibrium point of the ODE

$$\dot{W} = -h(W)$$

it follows that $h(W^*) = 0$ and therefore

$$\begin{aligned}
0 &= \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) (H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) - \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z})) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
\implies 0 &= \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}^\top}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] W^* - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
\implies 0 &= \Sigma W^* - \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
\text{so, } \Sigma W^* &= \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
\text{so, } W^* &= \Sigma^{-1} \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
\text{so, } H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} &= H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}^\top} \Sigma^{-1} \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \Phi H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \\
&= \mathcal{P} \Phi (H_{W^*}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}).
\end{aligned}$$

□

One rather restrictive condition of this lemma is that the Lipschitz constant τ of the contraction Φ must be less than the reciprocal condition number κ^{-1} . κ is a measure of *how orthonormal* the columns of the autocorrelation matrix Σ are. In particular, if the columns are indeed orthonormal, then $\kappa = 1$ and this condition ceases to be restrictive at all. If the columns are close to being linearly dependant, then κ is large so the requirement that κ^{-1} is small becomes troublesome. If indeed there is a linear dependence, the matrix Σ is not even invertible and the theorem breaks down completely. If we interpret $H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z})$ as a vector of features, then κ grows with the correlation between features. Higher correlation between the features imposes a greater constraint on the Lipschitz constant τ . If we have no inter-feature correlation then $\kappa = 1$ and we have no restriction at all on τ .

To actually solve ODE (10) we may need to compute

$$h(W) := \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) (H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z})) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right] \quad (12)$$

which may, or may not, be practical. For example, if the process \mathbf{Z} is ergodic, we can approximate (12) by taking a sufficiently long time average of

$$H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z) (H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z) - \Phi H_W^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z)).$$

Alternatively, we may approach the problem of solving (10) by first considering the explicit Euler method (with time-steps $\alpha_k > 0$)

$$\begin{aligned}
W_{k+1} &= W_k - \alpha_k h(W_k) \\
&= W_k - \alpha_k \mathbb{E}_\mu \left[H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) (H_{W_k}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z}) - \Phi H_{W_k}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}}(\mathbf{Z})) \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\zeta} \right],
\end{aligned}$$

then we might (heuristically) expect the algorithm

$$W_{k+1} = W_k - \alpha_k H^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z) (H_{W_k}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z) - \Phi H_{W_k}^{\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}} T^k(z)) \quad (13)$$

to converge to W^* , where α_k are positive definite real numbers that satisfy

$$\sum_{k=1}^{\infty} \alpha_k = \infty \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

We believe this heuristic could be made rigorous under mild assumptions, because algorithm (13) closely resembles the major algorithm extensively studied in [19] and [20] for which similar results hold. Theorems 17 and 2.1.1. appearing in [19] and [20] respectively suggest that an algorithm much like (13) converges almost surely to W^* if its associated ODE (reminiscent of (10)) satisfies condition (11), and the input process \mathbf{Z} is strongly mixing. The conjecture that algorithm (13) converges to W^* is also reminiscent of Theorem 3.1 by [21], and related results by [22]. These results are closely related to Q-learning and stochastic gradient descent. We note that (sadly) finding the fixed point of the general contraction mapping Φ renders the estimation of W a nonlinear problem.

The theory yields an online reinforcement learning algorithm which we state below. We envision that the agent chooses a fixed policy π and continues executing the policy for $\ell - 1$ time steps. Under this policy, the agent makes observations z_k and receives rewards r_k . We define $z_k(a)$ as the input to the ESN at time k if the agent had instead executed action $a \in \mathcal{A}$ at time k .

Algorithm 2: Online Learning

- 1: Choose initial output layer W_0 and reservoir state x_0
 - 2: Randomly generate $\mathbf{A}, \mathbf{C}, \boldsymbol{\zeta}$ according to procedure 1
 - 3: **for** each k from 0 to $\ell - 1$
 - 4: Compute $W_{k+1} = W_k - \alpha_k x_k (W_k^\top x_k - r_k - \max_a \{W_k^\top \sigma(\mathbf{A}x_k + \mathbf{C}z_k(a) + \boldsymbol{\zeta})\})$
 - 5: Compute $x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k + \boldsymbol{\zeta})$
-

4. Bee World

To demonstrate the theory presented in section 3, we created a game called *Bee World* and show that a simple reinforcement learning algorithm supported by an ESN can learn to play Bee World with respectable skill. The game is designed so that the theory presented previously is easy to visualise, rather than because the game is hard to master.

Bee World is set on the circle of unit circumference, which we denote by S^1 , and represent as an interval with edges identified. At every point y on the circle, there is a non-negative quantity of nectar which may be enjoyed by the bee without depletion. ‘Without depletion’ means that the bee takes a negligible amount of nectar from the point y , so the bee occupying

point y does not cause the amount of nectar at y to change. Furthermore, the nectar at every point y varies with time t according to the prescribed function

$$n(y, t) = 1 + \cos(\omega t) \sin(2\pi y) \quad (14)$$

(which we chose somewhat arbitrarily) that is unknown to the bee. Thus, the amount of nectar enjoyed by the bee at time t is a value that lies in the interval $[0, 2]$, which we will denote \mathcal{N} . Time advances in discrete integer steps $t = 0, 1, 2, \dots$, and at any time point t a bee at point y observes the quantity of nectar $r \in \mathcal{N}$ at point y and nothing else. Having made this observation, the bee may choose to move anywhere in the interval $(y - c, y + c)$ for some fixed $0 < c < 1$ and arrive at its chosen destination at time $t + 1$. The interval of possible moves $(-c, c)$ is called the action space and is denoted \mathcal{A} . The goal of the bee is to devise a policy whereby, given all its previous observations, the bee makes a decision as to where to move next, such that the discounted sum over all future nectar is as great as possible. The space of all previous (reward, action) pairs $(\mathcal{N} \times \mathcal{A})^{\mathbb{Z}^-}$ is contained by the space of bi-infinite sequences $(\mathbb{R}^2)^{\mathbb{Z}}$. The agent playing Bee World makes no observations beyond the rewards (nectar) and actions, but we could easily envision a more general game where the agent makes observations from a set Ω and therefore makes its decisions based on a left sequence of (reward, action, observation) triples.

The policy adopted by the bee may be realised as a deterministic policy $\pi : (\mathcal{N} \times \mathcal{A})^{\mathbb{Z}} \rightarrow \mathcal{A}$ (a CTI functional) for which the bee executes an action $a \in \mathcal{A}$ determined by the history of (reward, action) pairs. Alternatively, the bee may adopt a stochastic policy, for which every state history of (reward, action) pairs admits a distribution over actions \mathcal{A} from which the bee makes a random choice.

Though the evolution of Bee World is Markovian (and deterministic), the bee makes only a partial observation of the state of Bee World (i.e the amount of nectar the bee observes at time t) so the bee must take advantage of its *memory* to reconstruct the true state and find an optimal policy. This need for memory renders the problem suitable for an ESN, while ruling out the conventional theory of Markov Decision Processes. The problem of playing Bee World can therefore be formulated as a Partially Observed Markov Decision Process.

4.1. Approximating the value functional

Under a policy π , the nectar-action pairs experienced by the bee yield a realisation of the $(\mathcal{N}, \mathcal{A})^{\mathbb{Z}}$ -valued random variable \mathbf{Z} . It therefore makes sense to define the value functional $V : (\mathcal{N} \times \mathcal{A})^{\mathbb{Z}} \rightarrow \mathbb{R}$ associated to \mathbf{Z} by

$$V(z) = \mathbb{E}_{\mu} \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \ \forall j \leq 0 \right] \quad (15)$$

where $\mathcal{R} : (\mathcal{N} \times \mathcal{A})^{\mathbb{Z}} \rightarrow \mathbb{R}$ is the reward functional defined by $\mathcal{R}(z) = r_0$, where r_0 is the nectar collected at time 0, T is the shift operator, and $\gamma \in [0, 1)$ is the discount factor representing the relative importance of near and long term nectar consumption. We can see after a simple rearrangement of (15) that

$$V(z) = \mathcal{R}(z) + \gamma \mathbb{E}_{\mu}[V T_{\mathbf{Z}}(z)]$$

so V is the unique fixed point of the contraction mapping Φ defined by

$$\Phi(H)(z) := \mathcal{R}(z) + \gamma \mathbb{E}_\mu[HT_{\mathbf{Z}}(z)]$$

as discussed in Section 3. Thus, by Theorem 3.6, we can approximate the value function V using an ESN trained by regularised least squares as long as the nectar-action pairs $z \in (\mathcal{N} \times \mathcal{A})^{\mathbb{Z}}$ are drawn from a suitable ergodic process \mathbf{Z} . Therefore, we chose an initial policy π_0 such that \mathbf{Z} is ergodic. In particular, we chose a stochastic policy $\pi_0(z) \sim U(-c, c)$ for all histories of (reward, action) pairs $z \in (\mathcal{N} \times \mathcal{A})^{\mathbb{Z}}$ so that the bee takes a uniform sample from the action space $\mathcal{A} = (-c, c)$ at any point $y \in S^1$. For the purpose of playing a game, we set $c = 0.1$ and $\gamma = 0.5$. We allowed the bee to execute this policy for 2000 time steps and recorded the observed nectar at every time. The first 250 time steps are plotted in Figure 1.

Next, we set up an ESN of dimension $n = 300$, with reservoir matrix, input matrix, and bias $\mathbf{A}, \mathbf{C}, \zeta$ populated with i.i.d uniform random variables $U(-0.05, 0.05)$. \mathbf{A} was then multiplied by a scaling factor such that the 2-norm of \mathbf{A} satisfies $\|\mathbf{A}\|_2 = 1$. We choose an activation function $\sigma(x) := \max(0, x)$. We should pause here and note that ESN described here differs slightly from the ESN described in procedure 1. We instead generated $\mathbf{A}, \mathbf{C}, \zeta$ in a traditional way, which is empirically observed to be highly successful, as demonstrated in the literature, rather than the more cumbersome method described in procedure 1. These numerical results suggest that procedure 1 can be simplified.

We then computed a sequence of reservoir states $x_k \in \mathbb{R}^{300}$ for the ESN using the iteration

$$x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k + \zeta)$$

where $x_0 = 0$ and each $z_k \in (\mathcal{N} \times \mathcal{A})$ comprises 2 components: the first is the quantity of nectar observed by the bee at time k , and the second is the action $a \in (-c, c)$ executed at time k under policy π_0 . Now we return our attention to Theorem 3.6, and see that the W_ℓ^* minimising (over W)

$$\frac{1}{\ell} \sum_{k=0}^{\ell-1} \|W^\top (H^{\mathbf{A}, \mathbf{C}, \zeta} T^{-k}(z) - \gamma H^{\mathbf{A}, \mathbf{C}, \zeta} T^{1-k}(z)) - \mathcal{R}(z)\|^2 + \lambda \|W\|^2$$

converges to W minimising

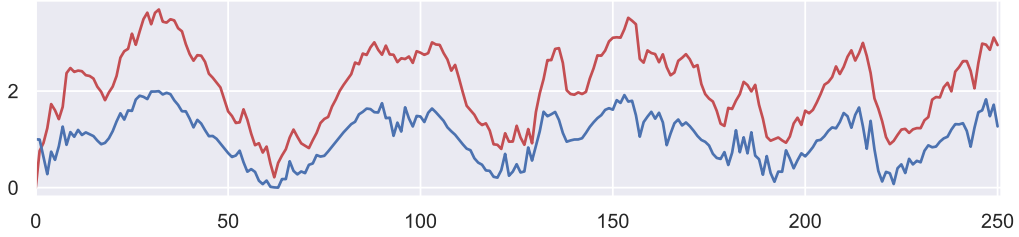
$$\|W^\top (x_k - \gamma x_{k+1}) - r_k\|^2 + \lambda \|W\|^2 \tag{16}$$

so we can immediately reformulate (16) as the least squares problem

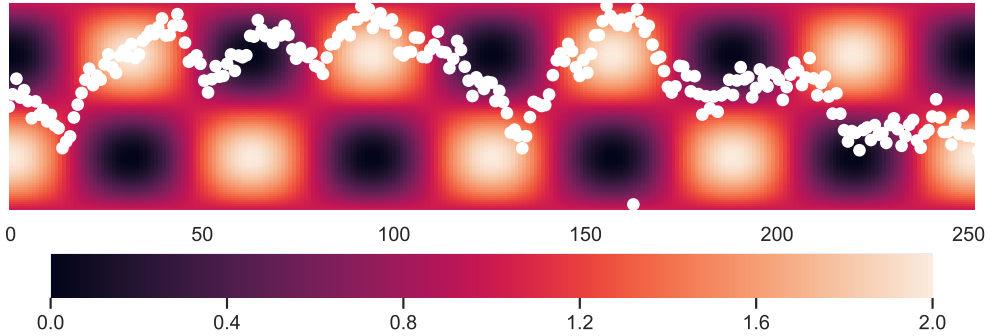
$$W = (\Xi^\top \Xi + \lambda I)^{-1} \Xi^\top U$$

where Ξ is the matrix with k th column

$$\Xi_k := x_k - \gamma x_{k+1}$$



(a) The nectar collected (blue) and the approximate value function under the initial policy π_0 (red) is plotted for the first 250 time steps (x -axis).



(b) The nectar function $n(y, t)$ (14) at every point represented as a heat map in the (t, y) plane, with the position of the bee at time t under the initial policy indicated by the overlaid white circles.

Figure 1: Dynamics of Bee World where the bee executes the initial policy $\pi_0(z) \sim U(-0.1, 0.1)$ for the first 250 time steps.

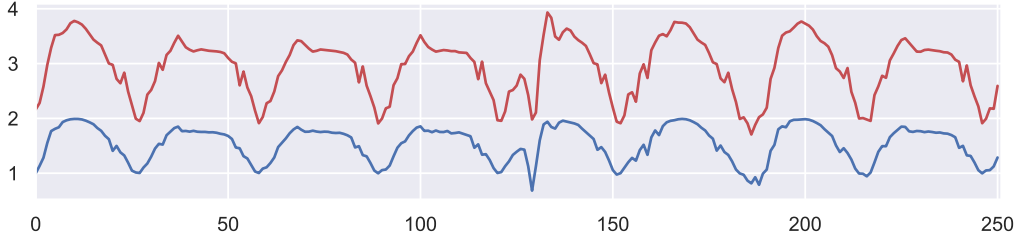
and U has k th entry r_k the k th quantity of nectar, and λ is the regularisation parameter which we set to 10^{-9} . We solved this linear system using the SVD. Now

$$V(z) \approx H_{W_\ell^*}^{\mathbf{A}, \mathbf{C}, \zeta}(z) \equiv (W_\ell^*)^\top H^{\mathbf{A}, \mathbf{C}, \zeta}(z) \equiv W^\top x$$

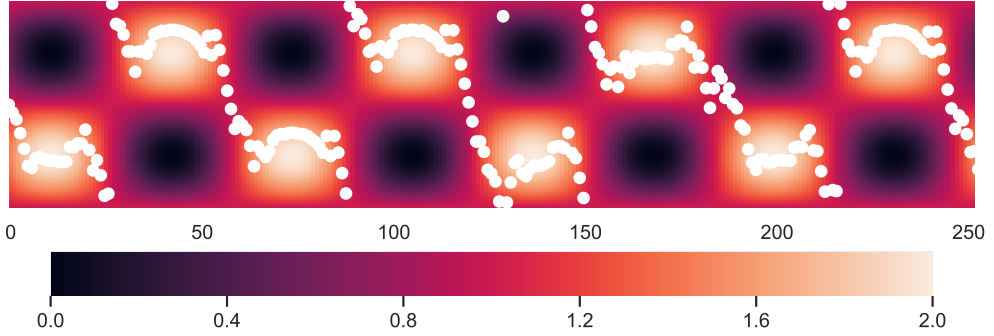
where x is the reservoir state associated to the left infinite input sequence z . Furthermore, the map $(W^\top \cdot)$ therefore approximates the unique fixed point of Φ (by Theorem 3.6) and this fixed point is exactly the value functional we are looking for. Thus, we can easily compute the approximate value of an arbitrary reservoir state x under the initial policy π by computing the inner product $W^\top x$. We illustrate this in Figure 1a by plotting, at each time $k = 1, \dots, 250$, the value of every observed state to accompany the observed nectar.

4.2. Updating the policy

Having computed an approximate value function under the initial policy $\pi_0(z) \sim U(-0.1, 0.1)$, we were faced with the problem of how to improve upon this policy. Exploring efficient and effective algorithms for iteratively improving a policy is a rich area of reinforcement learning research, but outside the scope of this section. Instead, we implemented a simple and greedy approach. For a given reservoir state x we consider 100 actions a_1, a_2, \dots, a_{100} uniformly sampled over $\mathcal{A} = (-0.1, 0.1)$, then for each action we consider the



(a) The nectar collected (blue) and the approximate value function (red) is plotted for the first 250 time steps (y-axis) under the improved policy π_1 .



(b) The nectar function $n(y, t)$ (14) at every point represented as a heat map in the (t, y) plane, with the position of the bee at time t under the improved policy is indicated by the overlaid white circles.

Figure 2: Dynamics of Bee World where the bee executes the improved policy π_1 for the first 250 time steps.

nectar-action pairs $z^{(1)}, \dots, z^{(100)} \in \mathcal{N} \times \mathcal{A}$ where the nectar for each pair is the current nectar; and is therefore the same in every pair. Then we compute the next reservoir states for each pair

$$x_{k+1}^{(i)} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k^{(i)} + \zeta)$$

and estimate the value of executing the i th action by computing $W^\top x_{k+1}^{(i)}$. Then we choose to execute the action a^* with the greatest estimated value - which determines our new policy π_1 - which yields a significant improvement over the initial policy π_0 , as illustrated in Figure 2. Under the initial policy π_0 the bee collected an average of approximately 1.05 nectar per unit time, in comparison to 1.52 nectar under the improved policy π_1 . This is much closer to the optimal value of approximately 1.60, which we obtain in the next section. The algorithm which first approximates the value function, and then updates the policy is described in Algorithm 3.

4.3. An Analytic Solution for Bee World

In this section, we will analyse Bee World so that we can compare the ESN solution to results that we can prove. To make our own lives easier, we consider a smooth version of Bee World, rather than the discrete time version solved by the ESN, so that we can formulate

Algorithm 3: One Step Offline Learning Algorithm (Bee World)

- 1: Choose initial reservoir state x_0
 - 2: Randomly generate $\mathbf{A}, \mathbf{C}, \zeta$
 - 3: **for** each k from 0 to $\ell - 1$
 - 4: Compute $x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}(r_k, a_k) + \zeta)$
 - 5: Find W that minimises $\sum_{k=0}^{\ell-1} \|W^\top(x_k - \gamma x_{k+1}) - r_k\|^2 + \lambda \|W\|^2$
 - 6: **for** each k from ℓ to $L - 1$
 - 7: Compute $a^* = \max_a \{W^\top \sigma(\mathbf{A}x_k + \mathbf{C}(r_k, a) + \zeta)\}$
 - 8: Compute $x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}(r_k, a^*) + \zeta)$
-

Bee World as a control problem that admits a solution via the Euler-Lagrange equation. We have the control system

$$\begin{aligned}\dot{\tau} &= 1 \\ \dot{y} &= u(y, \tau)\end{aligned}$$

where u is the controller dependant on y and τ . Then we have a cost function

$$\mathcal{C}(x, \tau, u) = f(u) - n(y, \tau)$$

where $f(x)$ is the penalty term for using the control u and $n(y, \tau)$ is the nectar function. In the above formulation of Bee World

$$f(u) = \begin{cases} 0 & \text{if } -c \leq u \leq c \\ \infty & \text{otherwise} \end{cases}$$

where $c = 0.1$. Then the objective is to find

$$u^* = \arg \min_u \left\{ \int_0^\infty \gamma^t \mathcal{C}(y, \tau, u) dt \right\}.$$

We can see that f is not a well defined function so we will introduce the family of functions

$$f_\epsilon(u) = -\epsilon \log(\cos(\pi u / (2c)))$$

where $\epsilon > 0$, and notice that f_ϵ approaches f pointwise as $\epsilon \rightarrow 0$. Next, we recall that the stationary points (including the minimum) of the integral functional

$$\mathcal{I}[y] = \int_0^\infty \mathcal{F}(t, y, \dot{y}) dt$$

all satisfy the Euler-Lagrange equation

$$\frac{d}{dt} \frac{\partial \mathcal{F}}{\partial \dot{y}} - \frac{\partial \mathcal{F}}{\partial y} = 0.$$

So, we let

$$\begin{aligned}\mathcal{F}(t, y, \dot{y}) &= \gamma^t \mathcal{C}(t, y, \dot{y}) \\ &= \gamma^t (-\epsilon \log(\cos(\pi \dot{y}/(2c))) - \cos(\omega t) \sin(2\pi y) - 1)\end{aligned}$$

then

$$\begin{aligned}0 &= \frac{d}{dt} \frac{\partial \mathcal{F}}{\partial \dot{y}} - \frac{\partial \mathcal{F}}{\partial y} \\ &= \frac{d}{dt} \left(\gamma^t \frac{d}{d\dot{y}} (-\epsilon \log(\cos(\pi \dot{y}/(2c)))) \right) + 2\pi \gamma^t \cos(\omega t) \cos(2\pi y) \\ &= \frac{\pi \epsilon}{2c} \frac{d}{dt} \left(\gamma^t \tan(\pi \dot{y}/(2c)) \right) + 2\pi \gamma^t \cos(\omega t) \cos(2\pi y) \\ &= \frac{\pi \epsilon}{(2c)} \left(\log(\gamma) \gamma^t \tan(\pi \dot{y}/(2c)) + \gamma^t \frac{\pi \ddot{y}}{2c} \sec^2(\pi \dot{y}/(2c)) \right) + 2\pi \gamma^t \cos(\omega t) \cos(2\pi y) \\ &= \frac{\pi \epsilon}{2c} \left(\log(\gamma) \tan(\pi \dot{y}/(2c)) + \frac{\pi \ddot{y}}{2c} \sec^2(\pi \dot{y}/(2c)) \right) + 2\pi \cos(\omega t) \cos(2\pi y),\end{aligned}$$

which we can reformulate as a dynamical system

$$\begin{aligned}\dot{v} &= -\frac{2c \cos^2(\pi v/(2c))}{\pi} \left(\frac{4c \cos(\omega \tau) \cos(2\pi y)}{\epsilon} + \log(\gamma) \tan(\pi v/(2c)) \right) \\ \dot{y} &= v \\ \dot{\tau} &= 1\end{aligned}\tag{17}$$

whose solutions are stationary points of the integral functional. For small ϵ , we approach the Bee World problem. We took $\epsilon = 10^{-5}$, $\gamma = 1/2$, initial position $y = 0$, and initial velocity $v = 0$ then simulated a trajectory of the ODE using `scipy.integrate.odeint`. We plotted this in Figure 3. The average nectar collected by under this policy was approximately 1.60.

5. Application to Stochastic Control

ESNs have shown remarkable promise in solving problems in mathematical finance - including by [23], [24], and [25] who used an ESN to predict the future values of stock prices. [26] used an ESN to learn the solution to a credit rating problem and [27] used an ESN to forecast exchange rates, comparing the results to forecasts made with an ARMA model. In this section we will introduce a stochastic optimal control problem arising in the market making problem. We will solve this problem analytically, and compare this to the solution obtained by a reinforcement learning agent supported by an ESN.

5.1. A Market Making Problem

We consider a stochastic control problem inspired by the motivations of a market maker acting in a general financial market. In practice the specific role of a market maker depends

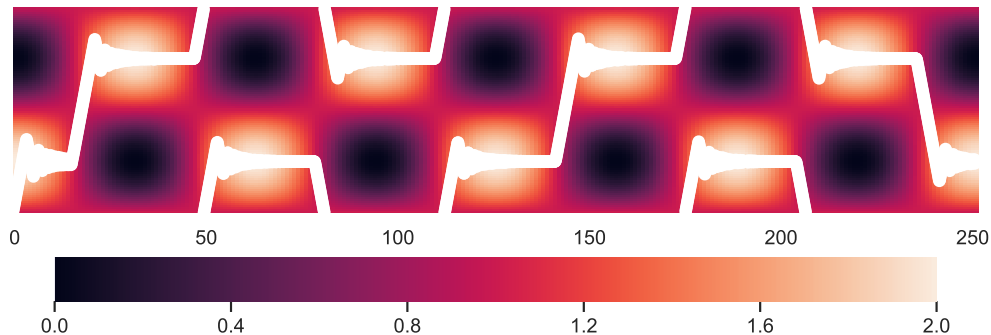


Figure 3: A numerical solution to the ODE (17) with $\epsilon = 10^{-5}$ (white line) superposed on the heat map of the nectar function $n(y, t)$ given in (14). Dark colours indicate regions of low nectar, light regions indicate high values of the nectar function. We observe that the solution trajectory spends much more time near local maxima of the nectar function but has complicated oscillatory fluctuations during transitions between local maxima. The oscillations are likely due to approaching a sort of singularity as $\epsilon \rightarrow 0$.

on the particular market, but we consider a market maker who provides liquidity to other market participants by quoting prices at which they are willing to sell (ask) and buy (bid) an asset. By setting the ask price higher than the bid price in general they can profit from the difference when they receive both a buy and sell order at these prices. However, the market maker faces risk, since if they buy a quantity of the asset the market price might move against them before they are able to find a seller.

The market making problem is a complex one, and has been studied extensively since the publication of the paper by [28]. The paper of [29] gives a good overview of much of this work. We consider a stylised version of this problem that focuses on inventory management without considering explicit optimal quoting strategies. We consider that a market maker acting relatively passively around the market price in ordinary conditions would expect to observe a random demand for buy and sell orders. If as a result of random fluctuations they find their inventory has drifted away from zero, they would set prices more competitively on either the ask or bid side to encourage trades to balance their position. Very broadly the conclusions of work on the market making problem are that there is a price to be paid to exert control over the inventory process and bring inventories closer to zero.

Motivated by this insight, we consider the market maker's inventory to be a stochastic process $(\mathbf{Y}_t)_{t \geq 0}$ with dynamics

$$d\mathbf{Y}_t = \boldsymbol{\pi}_t dt + \sigma d\mathbf{W}_t$$

where $(\mathbf{W}_t)_{t \geq 0}$ is a standard Brownian motion.

The parameter σ measures the volatility of the incoming order flow, and $(\boldsymbol{\pi}_t)_{t \geq 0}$ is the control process by which the market maker adds drift into their order flow by moving their bid and ask quotes. Naturally, there is a cost involved in applying the control, and a further cost to holding inventory away from zero. We introduce parameters α and β to quantify

these effects and model the market maker’s profit as a stochastic process solving

$$d\mathbf{Z}_t = (r - \alpha\pi_t^2 - \beta\mathbf{Y}_t^2)dt$$

where r is the rate of profit the market maker would achieve from the bid-ask spread if they did not have concerns about the asset price movements. We consider the case where the market maker seeks to maximise their long run discounted profit

$$v(y) = \max_{\pi} \mathbb{E}^y \left[\int_0^{\infty} e^{-\delta t} d\mathbf{Z}_t \right],$$

where \mathbb{E}^y is the expectation with the process started at $Y_0 = y$. We can show that the market maker’s value function and optimal control are

$$v(y) = -\alpha h y^2 + \frac{r - \alpha h \sigma^2}{\delta}, \quad \pi^*(y) = -h y, \quad (18)$$

where

$$h := \frac{-\alpha\delta + \sqrt{\alpha^2\delta^2 + 4\beta}}{2\alpha}$$

Further, the inventory process $\mathbf{Y}_{t \geq 0}$, when controlled by the optimal control $\pi^*(y) = -h y$ is given by the Ornstein-Uhlenbeck process

$$d\mathbf{Y}_t = -h\mathbf{Y}_t dt + \sigma d\mathbf{W}_t$$

whose stationary distribution is a Gaussian $\mathcal{N}\left(0, \frac{\sigma^2}{2h}\right)$.

We observe that this is an infinite horizon, Linear-Quadratic regulator (LQR) type problem, a class of problems which have a long history in the control literature, and more recently have been systematically studied in the reinforcement learning literature. Recent work on online learning for the LQR problem (e.g. [30, 31, 32]) has considered a range of variants of the LQR problem, including cases with uncertainty on the both the dynamics and the reward, and where the state variable may only be partially observed. However most of these approaches work in the setting of model-based learning approaches: that is, they attempt to learn a “model” of the world, and therefore exploit the fact that the LQR structure is known and can be learned from the data; in comparison, [30] still rely on the LQR structure, but do not directly try to learn the “model” of the world. The paper [33] analyses the difference between model-based and model-free approaches to the LQR problem, showing that one should expect an exponential separation between model-based and model-free approaches. In this context, our approach, which does not assume the LQR structure, can also be compared to model-free approaches, such as the classical work of [34], which takes a Q -learning approach.

5.2. Discretised problem

To turn this into a problem into one that can be used to train an Echo State Network we reformulate it in discrete time; we consider a process $\mathbf{Y}_0, \mathbf{Y}_1, \mathbf{Y}_2, \dots$ such that

$$\mathbf{Y}_{k+1} - \mathbf{Y}_k = \epsilon\pi_k + \sigma\sqrt{\epsilon}\mathcal{N}_k$$

where $(\mathcal{N}_k)_{k \in \mathbb{N}}$ are a sequence of i.i.d. random variables $\mathcal{N}_k \sim \mathcal{N}(0, 1)$ for each $k \in \mathbb{N}$, and $\epsilon > 0$ is the time increment. The control is now a sequence $\pi = (\pi_k)_{k \in \mathbb{N}}$. The profit function satisfies $\mathbf{Z}_0 = 0$ and

$$d\mathbf{Z}_k := \mathbf{Z}_{k+1} - \mathbf{Z}_k = \epsilon(r - \alpha\pi_k^2 - \beta\mathbf{Y}_k^2).$$

and the market maker seeks to maximise the value function

$$v(y) = \max_{\pi} \mathbb{E}^y \left[\sum_{k=0}^{\infty} e^{-\delta\epsilon k} d\mathbf{Z}_k \right],$$

over choices of the control π where \mathbb{E}^y is the expectation with the process started at $\mathbf{Y}_0 = y$.

It can be shown that in the limit as $\epsilon \rightarrow 0$, the optimal control and value function for this problem converge precisely to the optimal control and value function in the continuous case.

We state here the results in the case $\epsilon = 1$, the value we will use for the application of the Echo State Network below. Writing $\gamma = e^{-\delta}$, we find in this case that the value function and optimal control are given by

$$v(y) = -\alpha p y^2 + \frac{r - \gamma \alpha p \sigma^2}{1 - \gamma}, \quad \pi^* = -p y$$

where

$$p := \frac{(\alpha(\gamma - 1) + \gamma\beta) + \sqrt{(\alpha(\gamma - 1) + \gamma\beta)^2 + 4\alpha\beta\gamma}}{2\gamma\alpha}.$$

The process \mathbf{Y} controlled by π^* is Markovian, and has transition operator

$$\begin{aligned} (\mathcal{T}s)(y) &= \int_{-\infty}^{\infty} \mathbb{P}(\mathbf{Y}_{k+1} = y \mid \mathbf{Y}_k = x) s(x) dx \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(y-(1-p)x)^2}{2\sigma^2}} s(x) dx. \end{aligned}$$

It is straightforward to verify that the Gaussian probability density function

$$s^*(y) = \frac{\sqrt{p(2-p)}}{\sqrt{2\pi}\sigma} e^{-\frac{y^2 p(2-p)}{2\sigma^2}}, \quad (19)$$

is a fixed point of \mathcal{T} and hence that the controlled process has stationary distribution $\mathcal{N}\left(0, \frac{\sigma^2}{p(2-p)}\right)$.

5.3. Solving the Market Making Problem with an ESN

In this section, we seek to solve the the market making problem with a reinforcement learning algorithm supported by an ESN. In this set up, we assume the market maker has no knowledge of the cost function, and no knowledge of the effect of executing an action. The agent must execute a variety of actions in a variety of states to learn about the environment and the effect of its actions. Then, the market maker makes reasonable changes to its policy to arrive at a policy that reduces the long term costs of operation. The policy obtained by the reinforcement learning approach is compared to the optimal policy derived with full knowledge of the system.

5.3.1. Approximating the value functional

For the purpose of running the simulation, we let the cost of operating the control $\alpha = 1$, the cost of straying from the origin $\beta = 1$, the timestep $\epsilon = 1$, and the volatility parameter $\sigma = 1$. We take the baseline profit parameter $r = 0$. The inventory held, and action taken, by the market maker at time k will be denoted y_k and a_k respectively. A sequence of (inventory, action) pairs will be denoted $z \in (\mathbb{R}^2)^{\mathbb{Z}}$ with $z_k = (y_k, a_k)$. The value functional for the market maker problem is defined

$$V(z) = \mathbb{E}_{\mu} \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R} T^k(\mathbf{Z}) \mid \mathbf{Z}_j = z_j \forall j \leq 0 \right]$$

where $\mathcal{R} : (\mathbb{R}^2)^{\mathbb{Z}} \rightarrow \mathbb{R}$ is the reward functional

$$\mathcal{R}(z) = -(\alpha a_{-1}^2 + \beta y_0^2),$$

T is the shift operator, and $\gamma \in [0, 1)$ is the discount factor representing the relative importance of near and long term costs. We can see after a simple rearrangement that

$$V(z) = \mathcal{R}(z) + \gamma \mathbb{E}_{\mu}[V T_{\mathbf{Z}}(z)]$$

so V is the unique fixed point of the contraction mapping Φ defined by

$$\Phi(H)(z) = \mathcal{R}(z) + \gamma \mathbb{E}_{\mu}[H T_{\mathbf{Z}}(z)]$$

as discussed in Section 3. Thus, by Theorem 3.6, we can approximate the value function V using an ESN trained by regularised least squares *if* the (inventory, action) pairs (y_k, a_k) are the realisation of a stationary ergodic process. Consequently, we sought an initial policy π_0 such that the process \mathbf{Z} comprising the inventory-action pairs under policy π_0 is stationary and ergodic. In particular, we chose

$$\pi_0(y) \sim \mathcal{N}(0, \sigma_i^2) - \eta y \tag{20}$$

with $\eta = 0.05$ a constant representing the rate of exponential drift toward 0 and $\sigma_i^2 = 1$. We ran this policy for 10000 time steps, and recorded the pairs z_k along with the rewards r_k . Next, we set up an ESN of dimension $n = 300$, with reservoir matrix, input matrix, and bias $\mathbf{A}, \mathbf{C}, \zeta$ populated with i.i.d uniform random variables $U(-0.05, 0.05)$. \mathbf{A} was then multiplied by a scaling factor such that the 2-norm of \mathbf{A} satisfies $\|\mathbf{A}\|_2 = 1$. As in the previous example we chose σ to be the ReLU activation function. We then computed reservoir states

$$x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k + \zeta)$$

starting with an initial reservoir state $x_0 = 0$. An arbitrary reservoir state x then encodes the left infinite sequence of (inventory, action) pairs z . We seek an expression for the value of the reservoir state x by solving the least squares problem

$$W = (\Xi^{\top} \Xi + \lambda I)^{-1} \Xi^{\top} U$$

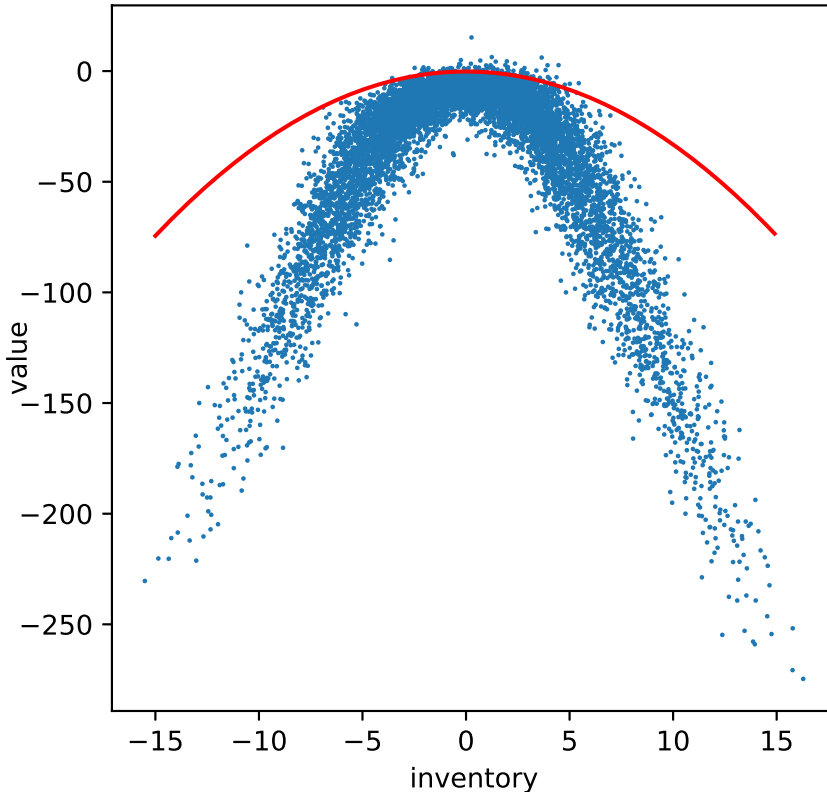


Figure 4: Under the initial policy, the value $V(\mathbf{Y})$ (y -axis) learned by the ESN at the inventory \mathbf{Y} (x -axis) at each of the 10000 timesteps is shown. The parabolic shape is consistent with the analytically derived optimal value function (19) shown in red. We note that the value function under the initial policy π_0 is not expected to match the value function under the optimal policy π^* .

(using the singular value decomposition) where Ξ is the matrix with k th column is

$$\Xi_k := x_k - \gamma x_{k+1}$$

and U is the vector of observations where the k th entry is the reward r_k , and λ is the regularisation parameter which we set to $1\text{e-}6$. We also chose $\gamma = e^{-1}$. In practice, the discount factor is usually much larger. With this, we obtain an expression for value of the reservoir state x given by $W^\top x$. The results of this policy are shown in Figures 4 and 5. The procedure which estimates the value function and improves upon the policy is described in Algorithm 4.

5.3.2. Updating the policy

We sought to create a new and improved policy based on the observations of under the initial policy using a naïve approach. At each time step, we consider 100 trial actions

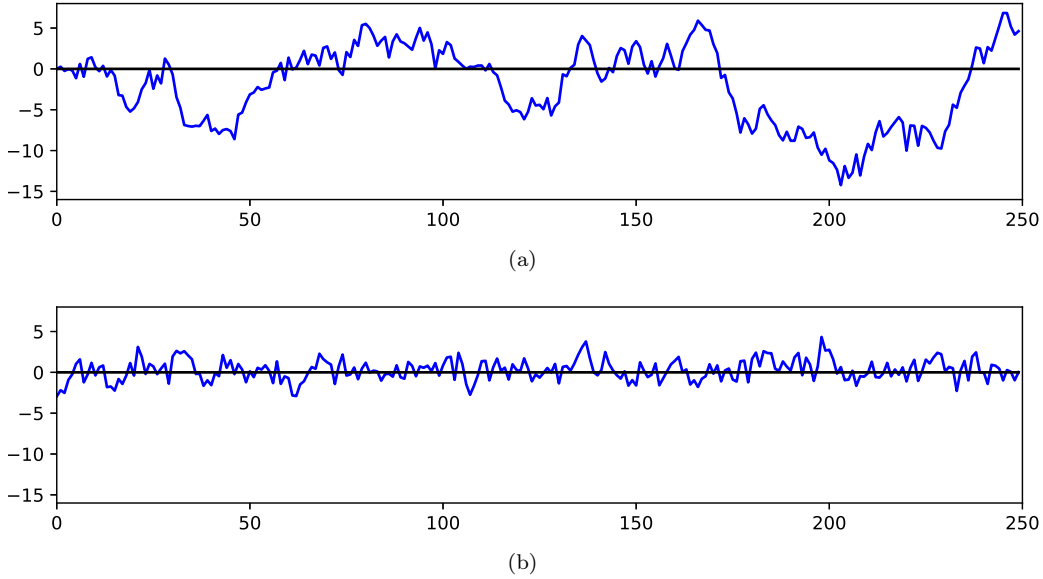


Figure 5: Dynamics of the market maker over time executing (a) the initial policy π_0 and (b) the improved policy π_1 . For each plot, the inventory (y -axis) is shown evolving with time (x -axis).

$a^{(1)}, a^{(2)}, \dots, a^{(100)}$ drawn from the standard normal distribution $\mathcal{N}(0, 1)$ and compute

$$x_{k+1}^{(i)} = \sigma(\mathbf{A}x_k + \mathbf{C}z_k^{(i)} + \zeta)$$

where $z_k^{(i)}$ is the (inventory, action) pair $(y_k, a^{(i)})$, and $a^{(i)}$ is trial action. For each i , we compute $W^\top x_{k+1}^{(i)}$ to obtain the predicted value of executing action $a^{(i)}$. We then choose to execute the action a^* with the greatest predicted value, and update the reservoir state using this (inventory, action) pair (y_k, a^*) . This defines our new policy. We ran this new policy for 10,000 time steps and illustrated the results in Figures 6a, and 6b.

Algorithm 4: One Step Offline Learning Algorithm (Market Making)

- 1: Choose initial reservoir state x_0
 - 2: Randomly generate $\mathbf{A}, \mathbf{C}, \zeta$
 - 3: **for** each k from 0 to $\ell - 1$
 - 4: Compute $x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}(y_k, a_k) + \zeta)$
 - 5: Find W that minimises $\sum_{k=0}^{\ell-1} \|W^\top(x_k - \gamma x_{k+1}) - r_k\|^2 + \lambda \|W\|^2$
 - 6: **for** each k from ℓ to $L - 1$
 - 7: Compute $a^* = \max_a \{W^\top \sigma(\mathbf{A}x_k + \mathbf{C}(y_k, a) + \zeta)\}$
 - 8: Compute $x_{k+1} = \sigma(\mathbf{A}x_k + \mathbf{C}(y_k, a^*) + \zeta)$
-

5.4. Comparison between the analytic and learned solutions

The one step reinforcement learning algorithm did not perfectly replicate the analytically derived optimal control, but has moved in a promising direction. We can see in Figure 6a

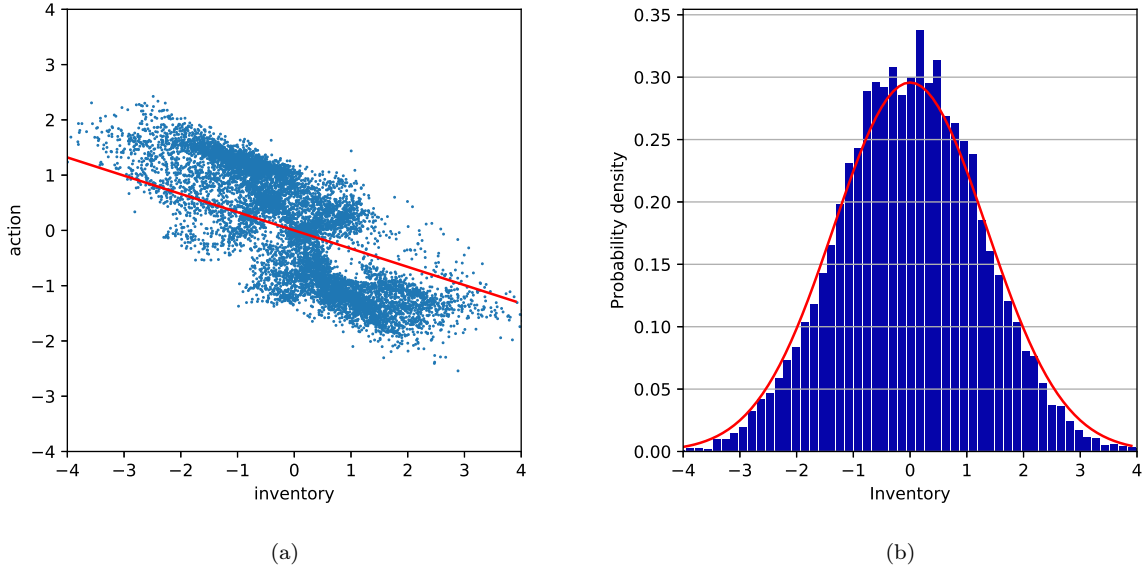


Figure 6: (a) Illustrates the (inventory, action) pairs (y_k, a_k) under the improved policy π_1 are represented as points on the scatter plot. The inventory is on the x -axis, and action is on the y -axis. The red line represents the analytically derived optimal control (equation (18)). (b) Illustrates the invariant measure of the inventory process under the improved policy π_1 is approximated with a histogram. The histogram is compared to the analytically derived invariant measure of the optimal control process $\mathcal{N}(0, 1.82)$ (equation (19)).

that the inventory process under the improved policy produces (inventory, action) pairs that have some scatter relative to the optimal policy indicated by the red straight line. This suggests that the market maker trained by reinforcement learning is behaving well in some average sense, despite performing many sub-optimal actions. It also appears that the reinforcement learning algorithm uses the control more aggressively than is optimal. This sub-optimal control results in greater costs than the optimal control. In particular the average cost incurred under the improved policy π_1 is 2.65, while the average cost under the analytically derived optimal policy is $\sigma/\sqrt{p(2-p)} = 1.35$.

Despite these sub-optimal moves, it seems that the inventory process learned by the market maker has an invariant measure that closely matches the optimal invariant measure. It is reassuring to see that an invariant measure appears, at least numerically, to exist, because the controlled process is assumed to be stationary and ergodic (and therefore admits an invariant measure) in Theorem 3.6.

It is also worth noting that the inventory process, controlled either by the ESN or the optimal control, has support on \mathbb{R} , which is not a compact space. Therefore, the conditions of Theorem 3.6 don't technically hold. However, the numerical results here suggest that the ESN has learned the value functional adequately well, suggesting that Theorem 3.6 may hold under relaxed conditions. Of course, realisations of the stochastic processes always explore only bounded subsets of \mathbb{R} .

6. Conclusions and future work

In this paper we have presented three novel mathematical results concerning Echo State Networks trained on data drawn from a stationary ergodic process. The first applies to offline supervised learning. The theorem states that, given a target function, enough training data and a large enough ESN, the least squares training procedure will yield an arbitrarily good approximation to the target function. The second result applies to an agent performing a stochastic policy π . After the agent has collected enough training data, and given a sufficiently large ESN, the least squares training procedure will yield an arbitrarily good approximation to the value function associated to the policy π . The third result is relevant to online reinforcement learning. Though the result is quite preliminary, the lemma is introduced with the intention of developing online algorithms (inspired by Q-learning) to learn the optimal policy for non-Markovian problems.

We demonstrated the second result (which generalises the first) on a deterministic control problem (Bee World) and a stochastic control problem (the market making problem). We chose these ‘toy model’ problems to understand the performance of the algorithm completely in cases that are solvable analytically, although these optimal solutions themselves are not entirely trivial. The reinforcement learning algorithm we use to improve the policy in both Bee World and the market making problem is extremely simple. It is essentially one iteration of an ϵ -greedy policy [35], with ϵ set to 0. Despite the simplicity of the algorithm, the single iteration considerably improved the policy, resulting in a reasonable approximation to the optimal policy.

It therefore seems a natural direction of future work to develop more sophisticated learning algorithms. Notably the linear upper confidence bound (linUCB) algorithm [35] has a linear structure that fits cleanly into the linear training framework of the ESN. As this work develops, it will become essential to have a rigorous framework describing the relationship between filters, functionals, random processes and reinforcement learning. The theory presented in this paper tentatively connects these objects using ideas from Markov Decision Processes, but the theory is far from complete.

Acknowledgements

Allen Hart and Kevin Olding are funded through the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (SAMBa), grant number EP/L015684/1.

We thank Jeremy Worsfold for insights about reinforcement learning and the linUCB algorithm, and for refactoring the Bee World code. We also thank Adam White for useful discussions about reinforcement learning settings.

References

- [1] H. Jaeger, The “echo state” approach to analysing and training recurrent neural networks, GMD-Report 148, German National Research Institute for Computer Science (01 2001).
- [2] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation* 14 (11) (2002) 2531–2560. doi: 10.1162/089976602760407955.

- [3] M. Schlegel, A. Jacobsen, M. Zaheer, A. Patterson, A. White, M. White, General value function networks, arXiv:1807.06763 (2018).
- [4] L. Gonon, L. Grigoryeva, J.-P. Ortega, Approximation bounds for random neural networks and reservoir systems, arXiv:2002.05933 (2020).
- [5] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Computer Science Review* 3 (3) (2009) 127 – 149. doi:<https://doi.org/10.1016/j.cosrev.2009.03.005>.
- [6] M. Lukoševičius, H. Jaeger, B. Schrauwen, Reservoir computing trends, *Künstliche Intelligenz*. 26 (4) (2012) 365–371.
- [7] A. Rodan, P. Tino, Minimum complexity echo state network, *IEEE transactions on neural networks* 22 (1) (2011) 131–144.
- [8] F. Triefenbach, A. Jalalvand, B. Schrauwen, J.-P. Martens, Phoneme recognition with large hierarchical reservoirs, in: J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, A. Culotta (Eds.), *Advances in Neural Information Processing Systems 23*, Curran Associates, Inc., 2010, pp. 2307–2315.
- [9] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science* 304 (5667) (2004) 78–80. arXiv:<https://science.sciencemag.org/content/304/5667/78.full.pdf>, doi:10.1126/science.1091277.
- [10] I. Szita, V. Gyenes, A. Lőrincz, Reinforcement learning with echo state networks, in: S. D. Kollias, A. Stafylopatis, W. Duch, E. Oja (Eds.), *Artificial Neural Networks – ICANN 2006*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 830–839.
- [11] S. Peitz, K. Bieker, On the universal transformation of data-driven models to control systems, arXiv:2102.04722 (2021).
- [12] A. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, A. Ng, On random weights and unsupervised feature learning, In *Proceedings of the 28th International Conference on Machine Learning* (2011).
- [13] L. Grigoryeva, J.-P. Ortega, Echo state networks are universal, *Neural Networks* 108 (2018) 495 – 508. doi:<https://doi.org/10.1016/j.neunet.2018.08.025>.
- [14] L. Grigoryeva, J.-P. Ortega, Differentiable reservoir computing, *Journal of Machine Learning Research* 20 (179) (2019) 1–62.
URL <http://jmlr.org/papers/v20/19-150.html>
- [15] K. McGoff, S. Mukherjee, N. Pillai, Statistical inference for dynamical systems: A review, *Statist. Surv.* 9 (2015) 209–252. doi:10.1214/15-SS111.
URL <https://doi.org/10.1214/15-SS111>
- [16] A. Khaleghi, D. Ryabko, Clustering piecewise stationary processes, in: *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2753–2758. doi:10.1109/ISIT44484.2020.9174045.
- [17] A. Hart, J. Hook, J. Dawes, Embedding and approximation theorems for echo state networks, *Neural Networks* 128 (2020) 234 – 247. doi:<https://doi.org/10.1016/j.neunet.2020.05.013>.
URL <http://www.sciencedirect.com/science/article/pii/S0893608020301830>
- [18] A. G. Hart, J. L. Hook, J. H. Dawes, Echo state networks trained by tikhonov least squares are $l^2(\mu)$ approximators of ergodic dynamical systems, *Physica D: Nonlinear Phenomena* (2021) 132882doi: <https://doi.org/10.1016/j.physd.2021.132882>.
URL <https://www.sciencedirect.com/science/article/pii/S0167278921000403>
- [19] A. Benveniste, M. Métivier, P. Prioure, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, 1990.
- [20] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, Hindustan Book Agency, 2009.
- [21] F. S. Melo, M. I. Ribeiro, Q-learning with linear function approximation, in: N. H. Bshouty, C. Gentile (Eds.), *Learning Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 308–322.
- [22] Z. Chen, S. Zhang, T. T. Doan, S. T. Maguluri, J.-P. Clarke, Performance of q-learning with linear function approximation: Stability and finite-time analysis, arXiv:1905.11425 (2019).
- [23] X. Lin, Z. Yang, Y. Song, Short-term stock price prediction based on echo state networks, *Expert*

- Systems with Applications 36 (3, Part 2) (2009) 7313 – 7317. doi:<https://doi.org/10.1016/j.eswa.2008.09.049>.
URL <http://www.sciencedirect.com/science/article/pii/S0957417408006519>
- [24] H. Zhang, J. Liang, Z. Chai, Stock prediction based on phase space reconstruction and echo state networks, *Journal of Algorithms & Computational Technology* 7 (1) (2013) 87–100. arXiv:<https://doi.org/10.1260/1748-3018.7.1.87>, doi:10.1260/1748-3018.7.1.87.
URL <https://doi.org/10.1260/1748-3018.7.1.87>
- [25] J. Dan, W. Guo, W. Shi, B. Fang, T. Zhang, Deterministic echo state networks based stock price forecasting, *Abstract and Applied Analysis* 2014 (2014) 137148. doi:10.1155/2014/137148.
URL <https://doi.org/10.1155/2014/137148>
- [26] J. Bozsik, Z. Ilonczai, Echo state network-based credit rating system, in: 2012 4th IEEE International Symposium on Logistics and Industrial Informatics, 2012, pp. 185–190.
- [27] L. Maciel, F. Gomide, D. Santos, R. Ballini, Exchange rate forecasting using echo state networks for trading strategies, in: 2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr), 2014, pp. 40–47.
- [28] M. Avellaneda, S. Stoikov, High-frequency trading in a limit order book, *Quantitative Finance* 8 (3) (2008) 217–224. arXiv:<https://doi.org/10.1080/14697680701381228>, doi:10.1080/14697680701381228.
URL <https://doi.org/10.1080/14697680701381228>
- [29] O. Guéant, Optimal market making, *Applied Mathematical Finance* 24 (2) (2017) 112–154. arXiv:<https://doi.org/10.1080/1350486X.2017.1342552>, doi:10.1080/1350486X.2017.1342552.
URL <https://doi.org/10.1080/1350486X.2017.1342552>
- [30] M. Fazel, R. Ge, S. Kakade, M. Mesbahi, Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator, in: *International Conference on Machine Learning*, PMLR, 2018, pp. 1467–1476, iSSN: 2640-3498.
- [31] Z. Mhammedi, D. J. Foster, M. Simchowitz, D. Misra, W. Sun, A. Krishnamurthy, A. Rakhlin, J. Langford, Learning the linear quadratic regulator from nonlinear observations, arXiv:2010.03799 (2020).
- [32] S. Dean, H. Mania, N. Matni, B. Recht, S. Tu, On the Sample Complexity of the Linear Quadratic Regulator, *Foundations of Computational Mathematics* 20 (4) (2020) 633–679. doi:10.1007/s10208-019-09426-y.
URL <https://doi.org/10.1007/s10208-019-09426-y>
- [33] S. Tu, B. Recht, The Gap Between Model-Based and Model-Free Methods on the Linear Quadratic Regulator: An Asymptotic Viewpoint, in: *Conference on Learning Theory*, PMLR, 2019, pp. 3036–3083, iSSN: 2640-3498.
URL <http://proceedings.mlr.press/v99/tu19a.html>
- [34] S. Bradtke, B. Ydstie, A. Barto, Adaptive linear quadratic control using policy iteration, in: *Proceedings of 1994 American Control Conference - ACC '94*, Vol. 3, 1994, pp. 3475–3479 vol.3. doi:10.1109/ACC.1994.735224.
- [35] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, MIT-press, 2015.