DYNAMICAL COMPUTATION WITH ECHO STATE NETWORKS

By

MUSTAFA CAN OZTURK

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2007

© 2007 Mustafa Can Ozturk

To my parents, Necmi Ozturk and Saliha Ozturk, who made everything I have achieved possible, for their enduring love, support and generosity.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisory committee chair, Dr. Jose Principe, for his inspiring ideas, constant support, and guidance. His enthusiasm for life and imaginative way of understanding concepts inspired me to learn different aspects of science and engineering. Without him, this dissertation would not have been possible.

Secondly, I would like to thank Dr. John Harris, for serving on my supervisory committee and offering wise insights into academia and other aspects of life. I also wish to thank the other members of my supervisory committee, Dr. Arunava Benerjee and Dr. Thomas Demarse for their insightful comments and suggestions.

During my studies, I gained much from the interactive environment at the Computational NeuroEngineering Laboratory. All of my colleagues helped me by sharing their knowledge and inspirational ideas. I especially thank Dongming Xu for his help and long hours of productive discussions. Most of this dissertation was a result of collaboration with Aysegul Gunduz, Nicholas Dedual, Rodrigo Sachhi, Sohan Seth, Johan Nyqvist, Kyu-Hwa Jeong, Ismail Uysal, Harsha Sathyendra, and Mark Skowronski. I like to thank each one of them for their expertise and insightful comments.

I would like to specifically thank my wonderful friends, Jianwu Xu, Rui Yan, and Anant Hegde. Deep philosophical conversations, sleepless nights for projects are unforgettable. They were certainly more than friends to me. I will always remember them with great respect.

I would like to thank my big family for their constant support and trust in me. Everything I have achieved is possible only with their love, encouragement, and guidance.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DYNAMICAL COMPUTATION WITH ECHO STATE NETWORKS

By

Mustafa Can Ozturk

May 2007

Chair: Jose C. Principe
Major: Electrical and Computer Engineering

Echo state networks (ESN) were recently proposed as a new recurrent neural network (RNN) paradigm. ESN couples dynamics with computation in novel ways by conceptually separating RNN into two parts: a recurrent topology of nonlinear PEs that constitutes a "reservoir of rich dynamics" and an instantaneous linear readout. The interesting property of ESN is that only the memoryless readout is trained, whereas the recurrent topology has fixed connection weights. This reduces the complexity of RNN training to simple linear regression while preserving a recurrent topology, but obviously places important constraints in the overall architecture that have not yet been fully studied.

Present design of fixed parameters of ESN relies on the selection of the maximum eigenvalue of the linearized system around zero (spectral radius). However, this procedure does not quantify in a systematic manner the performance of the ESN in terms of approximation error. In this study, we proposed a functional space approximation framework to better understand the operation of ESNs, and proposed an information-theoretic metric (the average entropy of echo states) to assess the "richness" of the ESN dynamics. We also provided an interpretation of the ESN dynamics rooted in system theory as families of coupled linearized systems whose poles move according to the input signal dynamics. With this interpretation, we put forward a design

methodology for functional approximation where ESNs are designed with uniform pole distributions covering the frequency spectrum to abide to the "richness" metric, irrespective of the spectral radius. A single bias parameter at the ESN input, adapted with the modeling error, configures the ESN spectral radius to the input-output joint space. Function approximation examples compare the proposed design methodology versus the conventional design.

On further investigating the use of ESNs for dynamical pattern recognition, we postulated that ESNs are particularly well suited for dynamical pattern recognition and we proposed a linear associative memory (LAM) as a novel readout for ESNs. From the class of LAMs, we adopted the minimum average correlation energy (MACE) filter because of its high rejection characteristics that allow its use as a detector in the automatic pattern recognition literature. In the ESN application, the MACE interprets the states of the ESN as a two-dimensional image: one dimension is time and the other dimension is the processing element index (space). An optimal template image for each class, which associates ESN states with the class label, can be analytically computed using training data. During testing, ESN states were correlated with each template image, and the class label of the template with the highest correlation is assigned to the input pattern. The ESN–MACE combination leads to a nonlinear template matcher with robust noise performance, as needed in non-Gaussian, nonlinear digital communication channels. We used a real-world data experiment for chemical sensing with an electronic nose to demonstrate the power of this approach. The proposed readout can also be used with liquid state machines eliminating the need to convert spike trains into continuous signals by binning or low-pass filtering.

We applied ESN on interesting real-world problems such as brain machine interface design, water inflow prediction, detection of action potentials in neural recordings, matched

filtering in digital communications, channel equalization of a nonlinear channel and compared its performance to other standard techniques. We proposed ESNs for signal processing in the complex domain. The use of ESNs for complex domain is very convenient since system training is equivalent to simple linear regression, which is trivial in the complex domain. The derivatives of the nonlinear activation functions are never necessary since the recurrent part is fixed apriori.

We showed that Freeman model of the olfactory cortex can be considered in the same framework of ESN. This work provided two important contributions to Freeman networks. First, it emphasized the need for optimal readout, and showed how to adaptively derive them. Second, it showed that the Freeman model is able to process continuous signals with temporal structure.

We also investigated the dynamics of ESNs without the echo state condition. This investigation led to a novel computational mode for nonlinear systems with sigmoidal nonlinearity, which does not require global stability. In this mode, although the autonomous system is unstable, the input signal forces the system dynamics to become "transiently stable". Function approximation experiments showed that the transiently stable ESN without the echo state condition is still capable of useful computation.

CHAPTER 1
INTRODUCTION

Literature Review of Dynamic Computation

Static networks are those whose response depends only on the present stimulus. Multiplayer perceptrons (MLPs), radial basis function (RBFs), and self-organizing map (SOM) networks are examples of static neural network models that deal with static input patterns. These models have been well studied in function spaces, for regression and classification, using statistical principles (Haykin, 1998). On the other hand, time is an essential ingredient in many real-world problems, from cognitive tasks (such as vision and speech processing) to engineering problems (such as system identification, noise cancellation, and channel equalization). These real-world problems require the extraction of information embedded in the temporal structure of signals. To incorporate time into the operation of a network, *memory* must be built into it. To be precise, memory here stands for short-term memory, the ability to remember the recent past which provides the contextual information from the history of time series to unambiguously identify the current stimulus.

Short-term Memory Structures

General linear discrete-time memory structure of order N (Figure 1-1 A) is also known as the generalized tapped delay line memory: each block has the transfer function $G(z)$, or impulse response $g(n)$ (Haykin, 2001). This is a single-input multiple-output system and taps of the system are filtered versions of the input signal.

Basically, the memory reconstructs the state of the system that created the time series in a sufficiently large state space where time is implicit; creating a trajectory that has a one-to-one correspondence (with a unique inverse) to the time series.

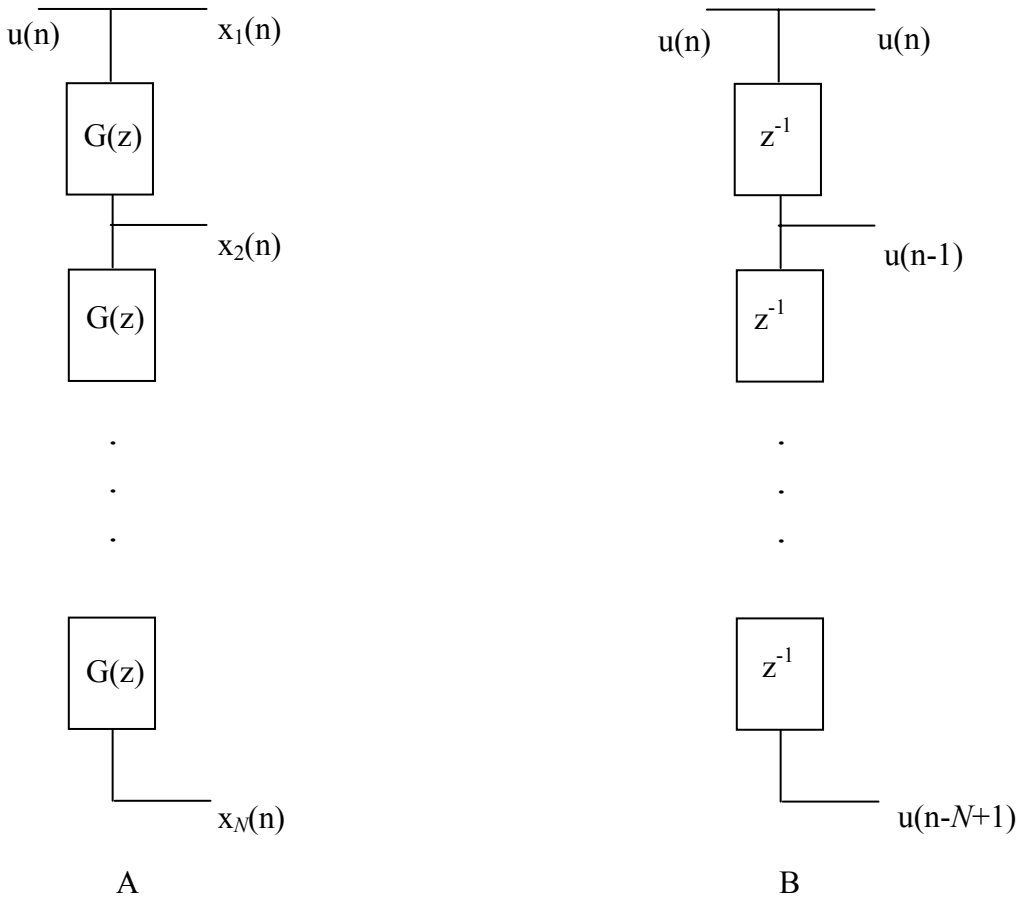


Figure 1-1. Delay line memory of order N. A) Generalized tapped delay line. B) Tapped delay line.

Quality of a memory structure is measured in two quantities: depth (*D*) and resolution (*R*) (Principe et al., 1993). Memory depth measures how long the memory can hold information about the past of a time series. Memory resolution is the amount of detail in the information memory holds. Formally, memory depth is defined as

$$D = \sum_{n=0}^{\infty} n g_N(n) \quad ,$$

where $g_N(n)$ is the overall impulse response of the memory obtained from N successive convolutions of $g(n)$. Resolution is defined as the number of taps in memory structure per unit time. For a linear memory structure of fixed taps, the product of memory depth and resolution is

a constant equal to the number of taps *N*. This creates a tradeoff between the memory depth and resolution and the choice of *G(z)* determines the values *R* and *D* in this tradeoff (Principe et al., 1993).

The time delay embedding (Takens, 1981), which can be implemented by a simple delay line, is the most commonly used form of short-term memory (Figure 1-1 B). The use of time delays for memory is also biologically motivated since signal delays are ubiquitous in the brain and essential in biological information processing (Braitenberg, 1990). The memory depth and resolution for the tapped delay line is *N* and 1 independent of the number of taps, respectively.

This restriction is relaxed in gamma filter which makes use of local feedback loops to create short-term memory (Principe et al., 1993; de Vries, 1991). Each section of gamma memory has the transfer function

$$G(z) = \frac{\mu \cdot z^{-1}}{1-(1-\mu)z^{-1}},$$

where μ is the adjustable feedback parameter. Note that gamma memory reduces to tapped delay line for μ=1. With the introduction of the feedback loop, stability becomes an issue. However, the overall system stability is easily guaranteed when each local feedback is stable (i.e. $0<\mu<2$). Memory depth for gamma memory is N/μ and the resolution is μ. This means that the memory depth can be improved by choosing the feedback parameter μ less than unity. However, this also reduces the resolution. Hence, in gamma memory, the memory depth can be improved without increasing the number of taps unlike the tapped delay line by sacrificing the resolution (Principe et al., 1993; de Vries, 1991).

Although tapped delay line and the gamma filter are the most common forms of short-term memory structures, other structures such as leaky integrator memory (context nodes or memory neurons) and Laguerre filter are also available in the literature. Time alignment filter, which is

obtained from gamma memory by allowing nonhomogenous μ values in different taps, extends the structure in Figure 1-1 A by allowing tap dependent modulations of memory depth and resolution (de Vries, 1991).

Network Architectures for Dynamical Computation

There are basically two ways of creating a dynamic network with short-term-memory. The first one is to stimulate a static network (e.g., linear mapper, MLP) via a memory structure at its input such as a tapped delay line or gamma memory (Haykin, 1998). This type of dynamical networks is commonly used in the literature. However, there are several practical drawbacks to this approach, which uses a spatial metaphor for time, especially when the dynamics of the time series are rich and time varying. The alternative approach for creating memory in an implicit manner is through the use of feedback. Feedback can be local at the single neuronal level or can be global covering the whole network. In neural networks literature, networks with global feedback loops are referred to as recurrent neural networks (RNN). The use of feedback provides with very powerful systems with rich dynamical behavior whereas it also brings in practical problems such as stability and training complexity (Haykin, 1998).

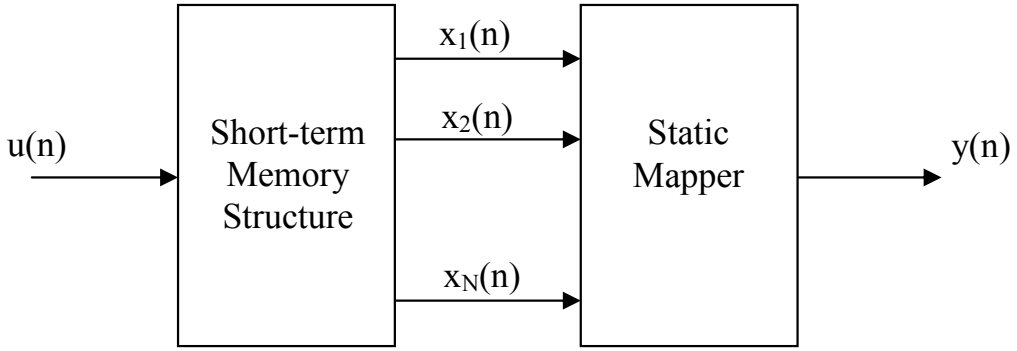


Figure 1-2. Dynamic network with a short-term memory (tapped delay line, gamma memory, etc.) structure followed by a static mapper (linear mapper, MLP, RBF etc.)

First we will consider the explicit representation of time where a static network (linear or nonlinear) is transformed into a dynamic network with a preprocessor which is a short-term memory structure. In these focused structures, the memory is restricted in the input layer of the network (Sandberg and Xu, 1997; Haykin, 1998). There is a clear separation of responsibilities in this approach of building a dynamical system, where memory represents time and static network accounts for mapping (Figure 1-2). Basically, memory reconstructs the state of the system that created the time series in a sufficiently large state space and static network maps the reconstructed space to the desired signal space. Moreover, the well-develop techniques of static network training such as least mean squares (LMS) or static back-propagation algorithm can be applied to the dynamic network.

The simplest structure for temporal processing in the form of Figure 1-2 is the finite impulse response filter (FIR) where the memory is a tapped delay line and the static network is a linear mapper. It has been widely used in adaptive signal processing applications due to its simplicity and the existence of effective learning algorithms (Haykin, 2001). Gamma filter is obtained when the memory structure in FIR is replaced by gamma memory (Principe et al., 1993; de Vries, 1991). Chains of first order integrators are interesting because they effectively decrease the number of delays necessary to create embeddings. The extra degree of freedom on the memory depth, gained from local feedback, has been fruitfully utilized in applications like system identification, echo cancellation, and speech classification (de Vries, 1991).

The first successful demonstration of a neural network for temporal processing was NETtalk devised by Sejnowski and Rosenberg in 1987 (Sejnowski and Rosenberg, 1997). They used a system which is based on MLP to convert English speech to phonemes. A more popular network is the time delay neural network (TDNN) that replaces the linear mapper in the FIR with

a more powerful nonlinear network, MLP (Haykin, 1998). Lang and Hinton have successfully applied TDNN for the recognition of 4 isolated words: “bee”, “dee”, “ee”, and “vee” (Lang and Hinton, 1998). TDNN has become one of the most popular neural network architectures for temporal processing and has been successfully applied to time series prediction, system identification, control, and speech processing. The structure of TDNN can be generalized as shown in Figure 1-3. The first block is a bank of linear filters operating in parallel on the input stimulus whereas second block implements a static nonlinear network. This structure is a universal dynamic mapper (Sandberg and Xu, 1997, page 477) according to universal myopic mapping theorem:

Any shift-invariant myopic dynamic map can be uniformly approximated arbitrarily well by a structure consisting of two functional blocks: a bank of linear filters feeding a static network.

TDNN is a special case of the structure described in this theorem and is also a universal mapper. However, there are several drawbacks of focused structures which convert time series to spatial signals (Elman, 1990). First of all, it requires an interface to the external world which buffers the input for further processing. Secondly the selection of the length of buffer is not trivial and once selected it imposes a firm limit on the memory depth. Moreover, buffering requires all the patterns to have the same length which is especially problematic in areas like speech processing where different length speech signals have to be processed (Elman, 1990). Finally, two instances of same time-series that are very similar can be very dissimilar in the reconstructed space (spatially distant) (Elman, 1990).

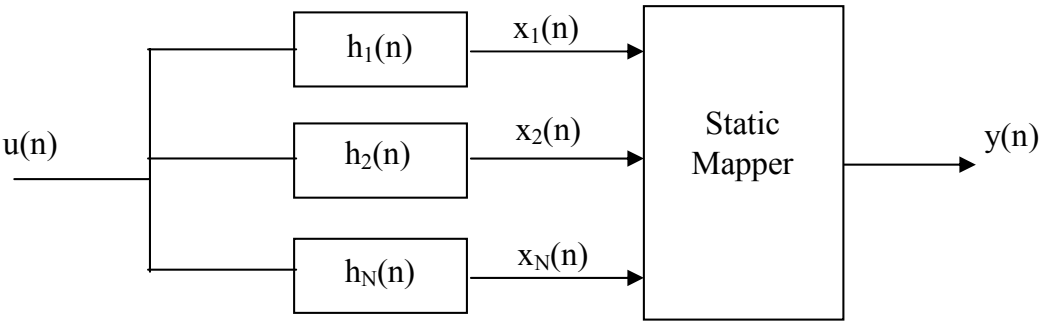


Figure 1-3. Generic structure for universal myopic mapping theorem, bank of filters followed by a static network

An alternative approach for creating memory is by means of recurrent connections within the network. In this approach, the system itself is dynamic and time, integrated into the system, is represented by the effect it has on processing. They are particularly powerful to deal with the time varying patterns and are practical for the problems where the dynamics of the considered process is complex. Various RNN architectures are available in the literature. One of the earlier architectures proposed by Jordan uses memory units that are fed from the system output to provide the contextual information (Jordan, 1986). Elman network is a modification of Jordan's, where the context units keep a copy of the hidden layer activations for the next update (Elman, 1990). Fully connected recurrent neural network contains several feedback loops between the processing elements (PE) in the hidden layer (Figure 1-4). RNNs can exhibit rich dynamical responses such as oscillations and even chaos. They have been widely used in many applications such as system identification and control of dynamical systems (Delgado et al., 1995; Feldkamp et al., 1998; Kechriotis et al., 1994; Principe, 2001). The computational power of fully connected RNNs (Siegelman and Sontag, 1991) has been mathematically proven by Siegelman and Sontag: All Turing machines may be simulated by fully connected recurrent networks built on neurons with sigmoid activation functions. However, the main problem with the RNNs is the difficulty to adapt the system weights. Various algorithms, such as back propagation through time (Werbos,

1990) and real time recurrent learning (Williams and Zipser, 1989) have been proposed to train recurrent systems. However, these algorithms suffer from a variety of problems: computational complexity resulting in slow training, complex performances surfaces, the possibility of instability, and the decay of gradients through the topology and time (Haykin, 1998). The problem of decaying gradients has been addressed with special PEs (Hochreiter and Schmidhuber, 1997). Alternative second order training methods based on extended Kalman filtering (Singhal and Wu, 1989; Puskorius and Feldkamp, 1996) and the multi-streaming training approach (Feldkamp et al., 1998) provide more reliable performance and have enabled practical applications.

Literature Review of Echo State Networks

Recently, a new recurrent network paradigm has been proposed by Jaeger under the name of echo state networks (ESN) (Jaeger, 2001; Jaeger, 2002, Jaeger and Hass, 2004). ESNs aim at addressing the problems with RNN training by separating the RNN architecture into two parts: a fixed recurrent topology of *dynamical reservoir* and an adaptive memoryless *readout* network. The input signal is fed to the dynamical reservoir which contains information about the history of input or/and output patterns when properly dimensioned (Figure 1-4). The outputs of the internal PEs (echo state) are fed to a memoryless but adaptive readout network (generally linear) that reads the reservoir and produces the network output. The interesting property of ESN is that only the memoryless readout is trained, whereas the recurrent topology (**W**) has fixed connection weights. This reduces the complexity of RNN training to simple linear regression while preserving a recurrent topology, but obviously places important constraints in the overall architecture that have not yet been fully studied.

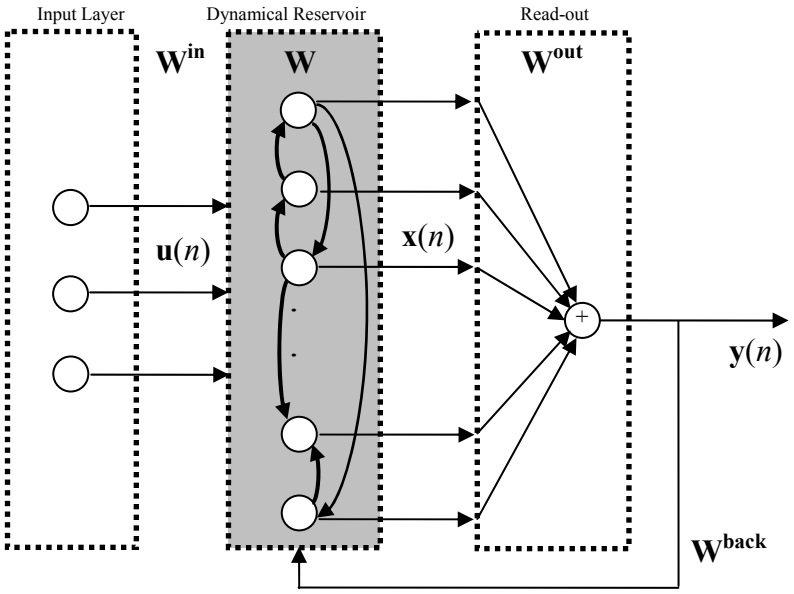


Figure 1-4. Diagram of an echo state network (ESN). ESN is composed of two parts: a fixed-weight (**W**) recurrent network and a linear readout. The recurrent network is a reservoir of highly interconnected dynamical components, states of which are called echo states. The memoryless linear readout is trained to produce the output.

Figure 1-4 depicts an ESN with M input units, N internal PEs and L output units. The value of the input unit at time n is $\mathbf{u}(n)=[u^1(n), u^2(n),\ldots,u^M(n)]^T$, of internal units are $\mathbf{x}(n)=[x^1(n), x^2(n),\ldots,x^N(n)]^T$, and of output units $\mathbf{y}(n)=[y^1(n), y^2(n),\ldots,y^L(n)]^T$. The connection weights are given in a NxM weight matrix $\mathbf{W}^{in}=(w_{ij}^{in})$ for connections between the input and the internal states, in an NxN matrix $\mathbf{W}=(w_{ij})$ for connections between the internal PEs, in an LxN matrix $\mathbf{W}^{out}=(w_{ij}^{out})$ for connections from internal units to the output units, and in an NxL matrix $\mathbf{W}^{back}=(w_{ij}^{back})$ for the connections that project back from the output to the internal units (Jaeger, 2001). The activation of the internal PEs (echo states) is updated according to

$$\mathbf{x}(n+1)=\mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1)+\mathbf{W}\mathbf{x}(n)+\mathbf{W}^{back}\mathbf{y}(n)) \tag{1-1}$$

where $\mathbf{f}=(f^1,f^2,...,f^N)$ are the internal unit's activation functions. Generally, all f^i's are hyperbolic tangent functions ($\frac{e^x-e^{-x}}{e^x+e^{-x}}$).

The readout is mostly a simple linear regressor network whose output is computed according to

$$\mathbf{y}(n+1) = \mathbf{W}^{out}\mathbf{x}(n+1) \tag{1-2}$$

When direct connections from the input units to the output units are used, the state vector is concatenated with the input vector to calculate the output.

A basic necessary property for ESN reservoir is the *echo state property* which states that for the ESN learning principle to work, the reservoir must asymptotically forget input history (input forgetting property). It has been shown in (Jaeger, 2001) that input forgetting property is equivalent to state forgetting, that is reservoir must forget its initial state after sufficiently long time. The echo state condition can be linked to the spectral radius (the largest among the absolute values of the eigenvalues of a matrix, denoted by $||.||$) of the reservoir's weight matrix, ($||\mathbf{W}||<1$). In fact, this condition states that the dynamics of the ESN is uniquely controlled by the input and the effect of initial states vanishes.

The design of fixed connections of ESN is based on the selection of the spectral radius. The reservoir is randomly and sparsely connected with a spectral radius suitable for the given problem. Sparse connectivity allows less coupling between the PEs encouraging the development of individual dynamics and increasing diversity in overall reservoir dynamics.

Thesis Goals

ESNs, although recently proposed, have proven very successful in applications such as system identification, chaotic time series prediction, and channel equalization. However, the state of the art is still immature. We see a few major shortcomings with the current ESN approach. First, the characterization of the reservoir properties is poorly understood. It is a mystery how a network with mostly random connections can be successful in solving difficult problems. A better understanding on the operation of dynamical reservoir is vital for improving

ESN theory. Second, imposing a constraint only on the spectral radius for ESN design is a weak condition to properly set the parameters of the reservoir, as experiments show (different randomizations with the same spectral radius perform differently for the same problem). Third, there is a fundamental problem with the way ESNs operate. The impact of fixed reservoir parameters for function approximation means that the information about the desired response is only conveyed to the output projection. This is not optimal, and strategies to select different reservoirs for different applications have not been devised.

We aim to address these problems by proposing a framework, a metric and a design principle for ESNs. We first deal with the analysis of ESNs and explains the framework and the metric proposed. The framework is a signal processing interpretation of basis and projections in functional spaces to describe and understand the ESN architecture. According to this interpretation, the reservoir states implement a set of bases functionals (representation space) constructed dynamically by the input, while the readout simply projects the desired response onto this representation space. The metric to describe the richness of the ESN dynamics is an information theoretic quantity, the *average state entropy* (ASE). Entropy measures the amount of information contained in a given random variable (Shannon, 1948). Here, the random variable is the instantaneous echo state from which the entropy for the overall state (vector) is estimated. Due to the time dependency of the states, the state entropy averaged over time (ASE) will be used as the quantifier to measure the richness of reservoir dynamics. We also interpret the ESN dynamics as a combination of *time varying* linear systems obtained from the linearization of the ESN nonlinear PE in a small local neighborhood of the current state.

We then propose a design methodology for the ESN reservoir in the light of the analysis tools developed for ESNs. According to this design principle, one should consider independently

the correlation among the basis and the spectral radius. In the absence of any information about the desired response, the ESN reservoir states should be designed with the highest ASE, independent of the spectral radius. The poles of the linearized ESN reservoir should have uniform pole distributions to generate echo states with the most diverse pole locations (which correspond to the uniformity of time constants). Effectively this will create the least correlated bases for a given spectral radius, which corresponds to the largest volume spanned by the basis set. When the designer has no other information about the desired response to set the basis, this principle distributes the system's degrees of freedom uniformly in space. It approximates for ESNs the well known property of orthogonal basis. The unresolved issue that ASE does not quantify is how to set the spectral radius, which depends again upon the desired mapping. a simple adaptive bias is added at the ESN input to control the spectral radius integrating the information from the input-output joint space in the ESN bases. For sigmoidal PEs, the bias adjusts the operating points of the reservoir PEs, which has the net effect of adjusting the volume of the state manifold as required to approximate the desired response with a small error. We show that ESNs designed with this strategy obtain systematically better results in a set of experiments when compared with the conventional ESN design.

We discuss the readouts for ESNs for function approximation and pattern recognition. The standard linear readout is used for function approximation tasks such as system identification, time series prediction, and channel equalization. We thoroughly investigate the use of ESN for temporal pattern recognition. The standard linear readout can be used for pattern recognition with ESNs by minimizing the mean-square error (MSE) with respect to a label. Classification of time series is different from the classification of static patterns since there is a structure between the samples of the time series over time. The dynamical classification problem can be reduced to

a static one by treating each sample of the time series individually. In this case, the time series is not treated as a whole and temporal information between samples is ignored. Moreover, one class label is associated with each time instant; therefore, post classification such as majority voting has to be applied. An alternative to this approach is embedding the time series to populate a short-term history of the pattern. Again, a static mapper can be used to classify the embedded pattern. Another major difficulty in dynamical pattern recognition is how to design the label, which should also be a time series. One straightforward method is to use +1 or -1 as the desired response throughout the application of the input pattern. However, the difficulty with this method is that the desired response is forced to be constant independent of the input signal dynamics. An alternative powerful technique is to create a one-step predictor for each class in order to capture the dynamics of the class generating the time-series (Zahalka and Principe, 1993). Then, during testing, the test pattern is presented to all predictors and the label of the predictor with the least prediction error is assigned as the class of the input pattern. We also propose an alternative readout for ESNs to be used in classification tasks that does not require a desired response. The goal is to design an ESN/LSM readout that will recognize a *class of inputs* that differ by some quantity (e.g. amplitude or shape considered as a distortion parameter). The proposed readout, called the minimum average correlation energy (MACE) filter, is adopted from optical pattern recognition literature, where it is used for recognition of a given object in 2-dimensional images in the presence of noise, geometric distortions and pose changes (Mahalanobis et al., 1987). Instead of using a desired response, the MACE filter creates a template from echo state responses corresponding to the training patterns of a given class. A MACE is trained for each class. During testing, the unknown time signal is fed to the ESN and

the states are correlated with each template image and the class label of the template with the highest correlation is assigned to the input pattern.

We investigate the implications of ESN idea on other networks. First, we investigate the echo state condition ($|\mathbf{W}|<1$) in a system theoretic framework to understand the effect of system stability in on the power of dynamical systems for computation. For this reason, we relax the echo state condition by allowing it to be slightly larger than 1. This introduces a new dynamical regime, called "transiently stable computation", where the system is autonomously unstable but it is stabilized by the input signal of sufficient power. In this regime, function approximation with a linear readout is still possible even though the system is not globally stable. Secondly, we investigate the biologically plausible model of the olfactory cortex, Freeman Model (FM), and propose to use a readout for FM to be able to use it as a universal computer. Without the proposed readout, the use of FM is limited to simple digit recognition, where the patterns are static and binary. We will demonstrate with experiments that FM coupled with a readout can process continuous valued signals. An interesting property of FM is the nonlinear function used in the model. FM nonlinearity does not have its largest slope at zero operating point unlike the sigmoidal nonlinearity used in ESNs.

Various applications of ESN for real-world problems are investigated. We first utilize ESNs for signal processing in the complex domain and compare complex ESN with other models for complex signal processing in a nonlinear channel equalization problem in a digital communication channel. Secondly, we propose ESNs to model brain machine interfaces and compare it with the linear Wiener filter. Brain machine interfaces aim at predicting the hand position of a primate using the brain signals. Thirdly, we apply ESNs to predict the water inflow at a hydro power plant using the previous values of water inflow levels. Then, we tackle two

temporal pattern recognition problems using ESNs with the MACE filter readout. In the spike detection problem, the goal is to detect action potentials in a noisy neural recording. The second problem deals with the design of a detector similar to a matched filter in a communication channel. We compare ESN-MACE filter to the matched filter in a baseband additive noise channel under different noise distributions.

CHAPTER 2
ANALYSIS OF ECHO STATE NETWORKS

ESN is a recurrent network paradigm described by a recurrently connected, *reservoir network* stimulated by an input signal, and an instantaneous, adaptive *readout*, which combines the reservoir states to form the desired signal. ESNs have been shown to be very powerful in many applications such as chaotic time series prediction, channel equalization, speech recognition. However, it remains a mystery how a network with mostly random connections can be so successful in solving difficult problems. This section aims at enlightening the basic mystery of ESNs by analyzing the operation of ESNs. The first subsection proposes a framework which is a signal processing interpretation of basis functions and projections in functional spaces to describe and understand the ESN architecture. In the second subsection, we interpret the ESN dynamics as a combination of linear systems. According to this interpretation, when the system operating point varies over time with the influence of the input signal, local ESN dynamics change. The third subsection introduces an information-theoretic metric, called average state entropy, to quantify the computational power of a dynamical network for function approximation. The ASE metric combined with the idea of bases and projections will lead to a design procedure for ESNs that will be discussed in the next chapter.

Echo States as Bases and Projections

Let us revisit the recursive update equation of an ESN. The activation of the internal PEs is updated according to

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n)), \tag{2-1}$$

where $\mathbf{f} = (f^1, f^2, \ldots, f^N)$ are the internal unit's activation functions. The output from the linear readout network is computed according to

$$\mathbf{y}(n+1) = \mathbf{W}^{out}\mathbf{x}(n+1), \tag{2-2}$$

ESNs resemble the RNN architecture proposed in (Puskorius and Feldkamp, 1994) and also used by (Sanchez, 2004) in brain machine interfaces. The critical difference is the dimensionality of the hidden recurrent PE layer and the adaptation of the recurrent weights. We submit that the ideas of approximation theory in functional spaces (bases and projections), so useful in optimal signal processing (Principe, 2001), should be utilized to understand the ESN architecture. Let $h(\boldsymbol{u}(t))$ be a real-valued function of a real-valued vector

$$\mathbf{u}(t) = [u_1(t), u_2(t), ..., u_M(t)].$$

In functional approximation, the goal is to estimate the behavior of $h(\mathbf{u}(t))$ as a combination of simpler functions $\varphi_i(t)$, called the bases functionals, such that its approximant $\hat{h}(\mathbf{u}(t))$, is given by

$$\hat{h}(\mathbf{u}(t)) = \sum_{i=1}^{N} a_i \varphi_i(t)$$

Here, a_i's are the projections of $h(\mathbf{u}(t))$ onto each basis functional. One of the central questions in practical functional approximation is how to choose the set of bases to approximate a given desired signal. In signal processing, the choice normally goes for complete set of orthogonal basis, independent of the input. When the basis set is complete and can be made as large as required, fixed bases work wonders (e.g. Fourier decompositions). In neural computing, the basic idea is to *derive the set of bases from the input signal through a multilayered architecture*. Consider a single hidden layer TDNN with N PEs and a linear output. The hidden layer PE outputs can be considered a set of non-orthogonal basis functionals dependent upon the input

$$\varphi_i(\mathbf{u}(t)) = g(\sum_j b_{ij} u_j(t))$$

Here, b_{ij}'s are the input layer weights, and g is the PE nonlinearity. The approximation produced by the TDNN is then

$$\hat{h}(\mathbf{u}(t)) = \sum_{i=1}^{N} a_i \varphi_i(\mathbf{u}(t)) \tag{2-3}$$

where a_i's are the weights of the output layer. Notice that the b_{ij} adapt the bases and the a_i adapt the projection in the projection space (readout). Here the goal is to restrict the number of bases (number of hidden layer PEs) because their number is coupled with the number of parameters to adapt which impacts generalization, training set size, etc. Usually, since all of the parameters of the network are adapted, the best basis in the joint (input and desired) space as well as the best projection can be achieved, and represents the optimal solution. The output of the TDNN is a linear combination of its internal representations, but to achieve a basis set (even if non-orthogonal), linear independence among the $\varphi_i(\mathbf{u}(t))$'s must be enforced. The effect of nonlinear transformation of the hidden PEs on the input signal has been investigated by many authors in R^N. Oh has shown that correlation among the weighted sums decrease after they pass through the sigmoid nonlinear function which can be approximated by piecewise linear functions (Oh and Lee, 1994). Shah and Poon showed that there exist weight values for MLP such that the outputs of its hidden layer PEs are linearly independent and hence, form a complete set of basis functions (Shah and Poon, 1999).

The ESN (and the RNN) architecture can also be studied in this framework. The reservoir states of Equation 2-1 correspond to the basis set which are recursively computed from the input, output and previous states through $\mathbf{W}^{\text{in}}$, $\mathbf{W}$ and $\mathbf{W}^{\text{back}}$. Notice however that none of these weight matrices are adapted, that is, the functional bases in the ESN are uniquely defined by the input and the initial selection of weights. In a sense ESNs are trading the adaptive connections in the RNN hidden layer by a brute force approach of creating fixed diversified dynamics in the hidden layer.

For an ESN with a linear read-out network, the output equation ($\mathbf{y}(n+1) = \mathbf{W}^{out}\mathbf{x}(n+1)$), has the same form of Equation 2-3, where the $\varphi_i(\mathbf{u}(t))$'s and a_i's are replaced by the echo states and the readout weights, respectively. These weights are adapted in the training data, which means that the ESN is able to find the optimal projection in the projection space, just like the RNN or the TDNN.

It is interesting that a similar perspective of basis and projections for information processing in biological networks has been proposed by (Pouget and Sejnowski, 1997). They explored the possibility that the response of single neurons in parietal cortex serve as basis functions for the transformations from the sensory input to the motor responses. They proposed that "the role of spatial representations is to code the sensory inputs and posture signals in a format that simplifies subsequent computation, particularly in the generation of motor commands" (Pouget and Sejnowski, 1997).

The central issue in ESN design is exactly the non adaptive nature of the basis set. Parameter sets in the reservoir that provide linearly independent states and possess a given spectral radius, may define drastically different projection spaces because the correlation among the bases is not constrained. A simple experiment was designed to demonstrate that the selection of the echo state parameters by constraining the spectral radius is not the most suitable for function approximation. Consider an echo state network of 100-units where the input signal is $\sin(2\pi n/(10\pi))$. Mimicking (Jaeger, 2001), the goal is to let the ESN generate the seventh power of the input signal. Different realizations of a randomly connected 100 unit ESN were constructed where the entries of $\mathbf{W}$ are set to 0.4, -0.4 and 0 with probabilities of 0.025, 0.025 and 0.95, respectively. This corresponds to a spectral radius of 0.88. Input weights are set to +1 or -1 with equal probabilities and $\boldsymbol{W}^{back}$ is set to zero. The input is applied for 300 time steps and

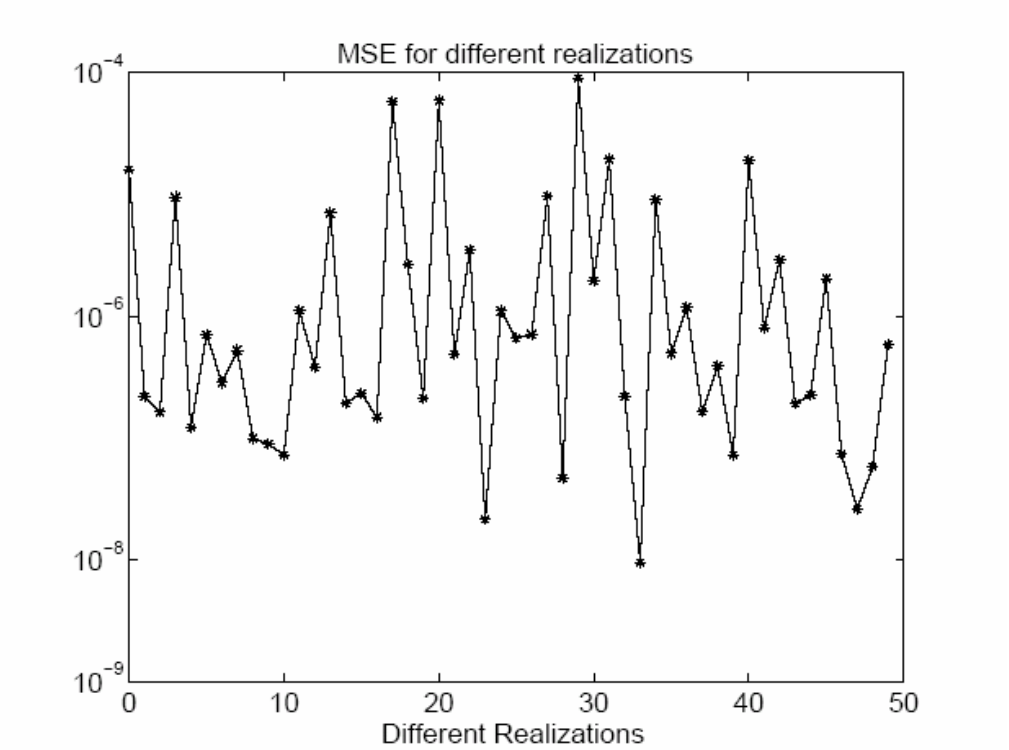


Figure 2-1. Performance of ESNs for different realizations of **W** with the same weight distribution. The weight values are set to 0.4, -0.4 and 0 with probabilities of 0.025, 0.025 and 0.95. All realizations have the same spectral radius of 0.88. In the 50 realizations, MSEs vary from $5.9\text{x}10^{-9}$ to $8.9\text{x}10^{-5}$. Results show that spectral radius of **W** is not the unique parameter that determines the performance of an ESN.

the echo states are calculated using Equation 2-1. The next step is to train the linear readout when a desired training signal, $\mathbf{d}(n)$ is available. One method to determine the optimal output weight matrix, $\mathbf{W}^{out}$ in the mean square error (MSE) sense (where MSE is defined by $(\frac{1}{2}\sum_{n}(\mathbf{d}(n)-\mathbf{y}(n))^{\perp}(\mathbf{d}(n)-\mathbf{y}(n)))$, is to use the Wiener solution given by (Haykin, 2001)

$$\mathbf{W}^{out} = E[\mathbf{x}\mathbf{x}^{\perp}]^{-1}E[\mathbf{x}\mathbf{d}] \cong \left(\frac{1}{T}\sum_{n=0}^{T}\mathbf{x}(n)\mathbf{x}(n)^{\perp}\right)^{-1}\left(\frac{1}{T}\sum_{n=0}^{T}\mathbf{x}(n)\mathbf{d}(n)\right) \tag{2-4}$$

Here $E[.]$, T and $^{\perp}$ denotes the expected value operator, the length of the training sequence and the conjugate transpose of a complex vector. Figure 2-1 depicts the MSE values for 50 different realizations of the ESNs. As observed, even though each ESN has the same sparseness and spectral radius, the MSE values obtained vary greatly among different realizations. The

minimum MSE value obtained among the 50 realizations is $5.9\text{x}10^{-9}$ whereas the maximum MSE is $8.9\text{x}10^{-5}$. This experiment demonstrates that a design strategy that is based solely on the spectral radius is not sufficient to specify the system architecture for function approximation. This shows that for each set of random weights that provide the same spectral radius, the correlation or degree of redundancy among the bases will change and different performances are encountered in practice.

ESN Dynamics as a Combination of Linear Systems

It is well known that the dynamics of a nonlinear system can be approximated by that of a linear system in a small neighborhood of an equilibrium point (Kuznetsov, 1998). Here, we will perform the analysis with hyperbolic tangent PEs and approximate the ESN dynamics by the dynamics of the linearized system in the neighborhood of the current system state. Hence, when the system operating point varies over time, the linear system approximating the ESN dynamics will change. We are particularly interested in the movement of the poles of the linearized ESN. Consider the update equation for the ESN without output feedback given by

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n))$$

Linearizing the system around the current state ***x(n)***, one obtains the Jacobian matrix, **J***(n+1)*, defined by

$$\begin{aligned}\mathbf{J}(n+1) &= \begin{bmatrix} \dot{f}(net_1(n))w_{11} & \dot{f}(net_1(n))w_{12} & \dots & \dot{f}(net_1(n))w_{1N} \\ \dot{f}(net_2(n))w_{21} & \dot{f}(net_2(n))w_{22} & \dots & \dot{f}(net_2(n))w_{2N} \\ \dots & \dots & \dots & \dots \\ \dot{f}(net_N(n))w_{N1} & \dot{f}(net_N(n))w_{N2} & \dots & \dot{f}(net_N(n))w_{NN} \end{bmatrix} \\ &= \begin{bmatrix} \dot{f}(net_1(n))w_{11} & 0 & \dots & 0 \\ 0 & \dot{f}(net_2(n)) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dot{f}(net_N(n)) \end{bmatrix} \cdot \mathbf{W} \\ &= \mathbf{F}(n) \cdot \mathbf{W} \end{aligned} \qquad (2\text{-}5)$$

Here, $net_i(n)$ is the i^{th} entry of the vector ($\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n)$) and w_{ij} denotes the $(i, j)^{\text{th}}$ entry of **W**. The poles of the linearized system at time n+1 are given by the eigenvalues of the Jacobian matrix $\boldsymbol{J}$(n+1). When the amplitude of each PE input changes, the local slope changes, and so the poles of the linearized system are time varying, although the parameters of ESN are fixed.

In order to visualize the movement of the poles, consider an ESN with 100 states. The entries of the internal weight matrix are chosen to be 0, 0.4 and -0.4 with probabilities 0.9, 0.05 and 0.05. **W** is scaled such that a spectral radius of 0.95 is obtained. Input weights are set to +1 or -1 with equal probabilities. A sinusoidal signal with a period of 100 is fed to the system and the echo states are computed according to Equation 2-1. Then, the Jacobian matrix and the eigenvalues are calculated using Equation 2-5. Figure 2-2 shows the pole tracks of the linearized ESN for different input values. A single ESN with fixed parameters implements a combination of many linear systems with varying pole locations, hence many different time constants that modulates the richness of the reservoir of dynamics as a function of input amplitude. Higher amplitude portions of the signal tend to saturate the nonlinear function and cause the poles to shrink toward the origin of the z-plane (decreases the spectral radius), which results in a system with large stability margin. When the input is close to zero, the poles of the linearized ESN are close to the maximal spectral radius chosen, decreasing the stability margin. When compared to their linear counterpart, an ESN with same number of states results in a detailed coverage of the z-plane dynamics (ε-coverage), which illustrates the power of nonlinear systems. Similar results can be obtained using signals of different shapes at the ESN input.

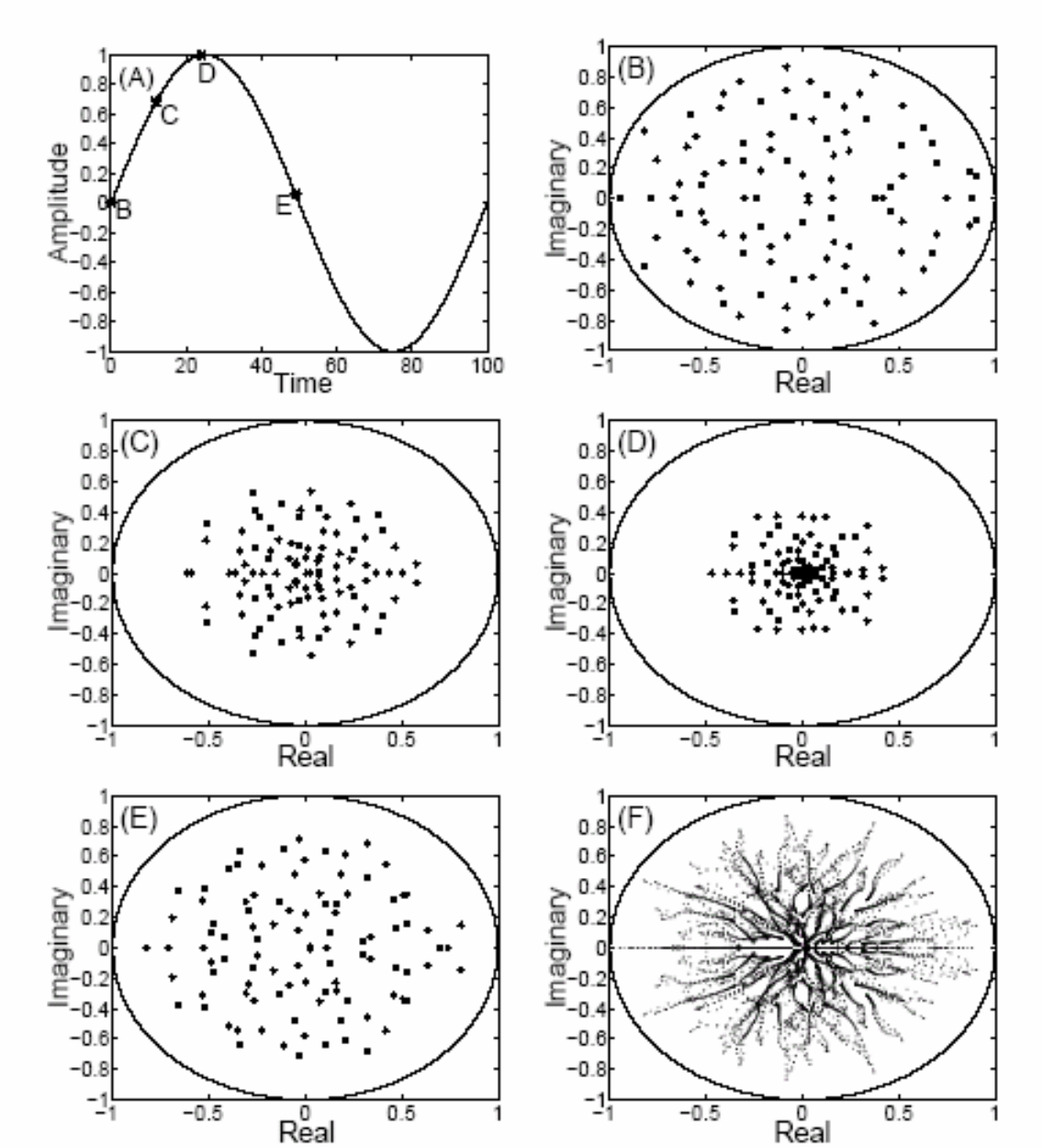


Figure 2-2. Pole tracks of the linearized ESN when the input goes through a cycle. A single ESN with fixed parameters implements a combination of many linear systems with varying pole locations. A) One cycle of sinusoidal signal with a period of 100. B)-E) show the positions of poles of the linearized systems when the input values are at B, C, D, and E in panel A. F) Cumulative pole locations show the movement of the poles as the input changes.

Theorem: The eigenvalues of the linearized system have the largest radius when the system state is zero.

Proof: Assume **W** has nondegenerate eigenvalues and corresponding linearly independent eigenvectors. Then, consider the eigendecomposition of **W**, where $\mathbf{W} = \mathbf{PDP}^{-1}$, **P** is the eigenvector matrix and **D** is the diagonal matrix of eigenvalues ($\mathbf{D}_{ii}$) of **W**. Since **F**(n) and **D** are diagonal, $\mathbf{J}(n+1) = \mathbf{F}(n)\mathbf{W} = \mathbf{F}(n)(\mathbf{PDP}^{-1)} = \mathbf{P}(\mathbf{F}(n)\mathbf{D})\mathbf{P}^{-1}$ is the eigendecomposition of **J**(n+1). Here, each entry of $\boldsymbol{F}(\mathrm{n})\boldsymbol{D}$, $\dot{f}(net_i(n))\mathbf{D}_{ii}$, is an eigenvalue of **J**.

$\left|\dot{f}\left(net_{\,i}(n)\right)\mathbf{D}_{ii}\right| \le \left|\mathbf{D}_{ii}\right|$ since $\left|\dot{f}(net_i(n))\right| \le \left|\dot{f}(0)\right|$.

A key corollary of the above analysis is that the spectral radius of an ESN can be adjusted using a constant bias signal at the ESN input without changing the recurrent connection matrix, **W**. The application of a nonzero constant bias will move the operating point to regions of the sigmoid function closer to saturation and always decrease the spectral radius due to the shape of the nonlinearity. This property will be exploited in the design of ESNs which will be discussed in the next chapter.

Average State Entropy as a Measure of the Richness of ESN Reservoir

The concept of "rich dynamical reservoir" (Jaeger, 2001) has not been quantified with a well-defined metric. Our framework of bases and projections leads to a new metric to quantify the richness of ESN reservoir. Here, we propose the *instantaneous state entropy* to quantify the distribution of instantaneous amplitudes across the ESN reservoir states. Entropy of the instantaneous ESN states is appropriate to quantify performance in function approximation because the ESN output is a mere weighted combination of the value of the ESN states. If the echo state's instantaneous amplitudes are concentrated on only a few values across the ESN state dynamic range, the ability to approximate an arbitrary desired response by weighting the states is limited (and wasteful due to redundancy between the different states) and performance will

suffer. On the other hand, if the ESN states provide a diversity of instantaneous amplitudes, then it is much easier to achieve the desired mapping. Hence, the instantaneous entropy of the states appears as a good measure to quantify "richness" of dynamics with instantaneous mappers. Due to the time structure of signals, the *average state entropy* (ASE), defined as the state entropy averaged over time will be the parameter used to quantify the diversity in the dynamical reservoir of the ESN. Moreover, entropy has been proposed as an appropriate measure of the volume of the signal manifold (Cox, 1946; Amari, 1990). Here, ASE measures the volume of the echo state manifold spanned by the trajectories.

Renyi's quadratic entropy is employed here because it is a global measure of information and efficient nonparametric estimator that avoids explicit pdf estimation has been developed (Principe et al., 2000;Erdogmus et al., 2003; Erdogmus and Principe, 2002). Renyi's entropy with parameter γ for a random variable X with a pdf $f_X(x)$ is given by (Principe et al., 2000)

$$H_\gamma(X) = \frac{1}{1-\gamma}\log E[f_X^{\gamma-1}(X)]$$

The Renyi's quadratic entropy is obtained for $\gamma=2$ (for $\gamma\rightarrow 1$). Shannon's entropy is obtained). Given N samples $\{x_1, x_2, \ldots, x_N\}$drawn from the unknown pdf to be estimated, Parzen windowing approximates the underlying pdf by

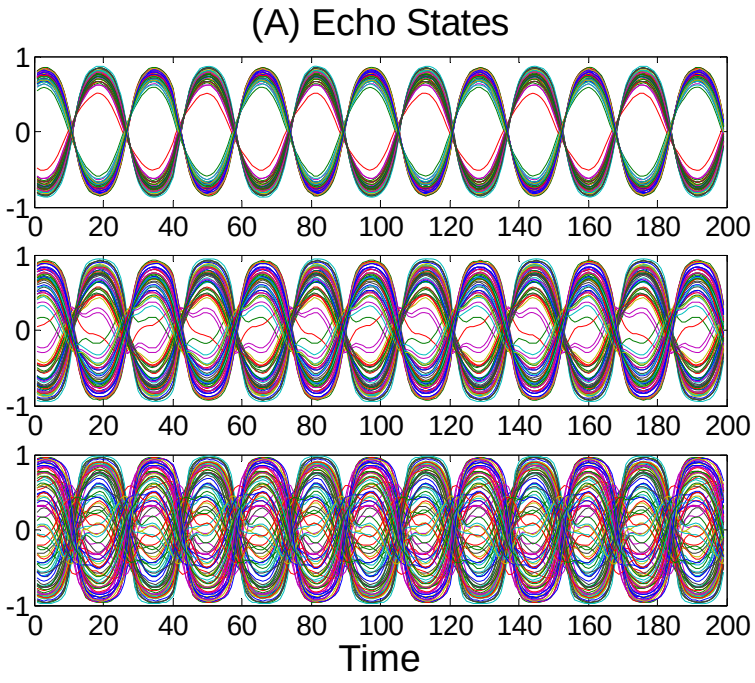
$$f_X(x) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x - x_i)$$

where K_σ is the kernel function with the kernel size σ. Then the Renyi's quadratic entropy can be estimated by (Principe et al.,2000; Erdogmus et al., 2003).

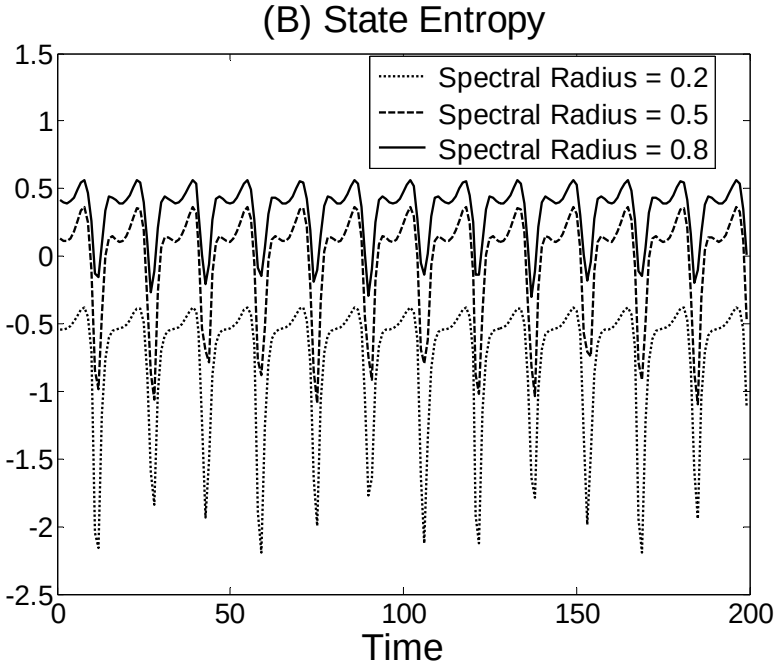
$$H_2(X) = -\log\left[\frac{1}{N^2}\sum_j\left(\sum_i K_\sigma(x_j - x_i)\right)\right] \tag{2-6}$$

The instantaneous state entropy is estimated using Equation 2-6 where the samples are the entries of the state vector $\boldsymbol{x}(n) = [x_1(n), x_2(n), ..., x_N(n)]^T$, of an ESN with *N* internal PEs. Results will be shown with a Gaussian kernel with kernel size chosen to be 0.3 of the standard deviation of the entries of the state vector. We will show that ASE is a more sensitive parameter to quantify the approximation properties of ESNs by experimentally demonstrating that ESNs with the same spectral radius display different ASEs.

Let us consider the same 100 unit ESN that we used in the previous section built with three different spectral radii 0.2, 0.5, 0.8 with an input signal of sin(2πn/20). Figure 2-3 A depicts the echo states over 200 time ticks. The instantaneous state entropy is also calculated at each time step using Equation 2-6 and plotted in figure 2-3 B. First, note that the instantaneous state entropy changes over time with the distribution of the echo states as we would expect, since state entropy is dependent upon the input signal that also changes in this case. Second, as the spectral radius increases in the simulation, the diversity in the echo states increases. For the spectral radius of 0.2, echo state's instantaneous amplitudes are concentrated only on a few values which is wasteful due to redundancy between different states. In practice, to quantify the overall representation ability over time, we will use ASE, which takes values -0.735, -0.007 and 0.335, for the spectral radii of 0.2, 0.5 and 0.8, respectively. Hence, ASE is affected by the spectral radius of the **W** matrix as we would expect. Moreover, even for the same spectral radius of 0.5, several ASEs are possible. Figure 2-4 shows ASEs from 50 different realizations of ESNs with same spectral radius of 0.5, which means that ASE is a finer descriptor of the dynamics of the reservoir. Although we have presented an experiment with sinusoidal signal, similar results are obtained for other inputs as long as the input dynamic range is properly selected.



A



B

Figure 2-3. Echo states and state entropy for different ESNs. A) Outputs of echo states (100 PEs) produced by systems with spectral radius of 0.2, 0.5 and 0.8, from up to bottom, respectively. B) Instantaneous state entropy is calculated using Equation 2-6 for echo states in panel A.

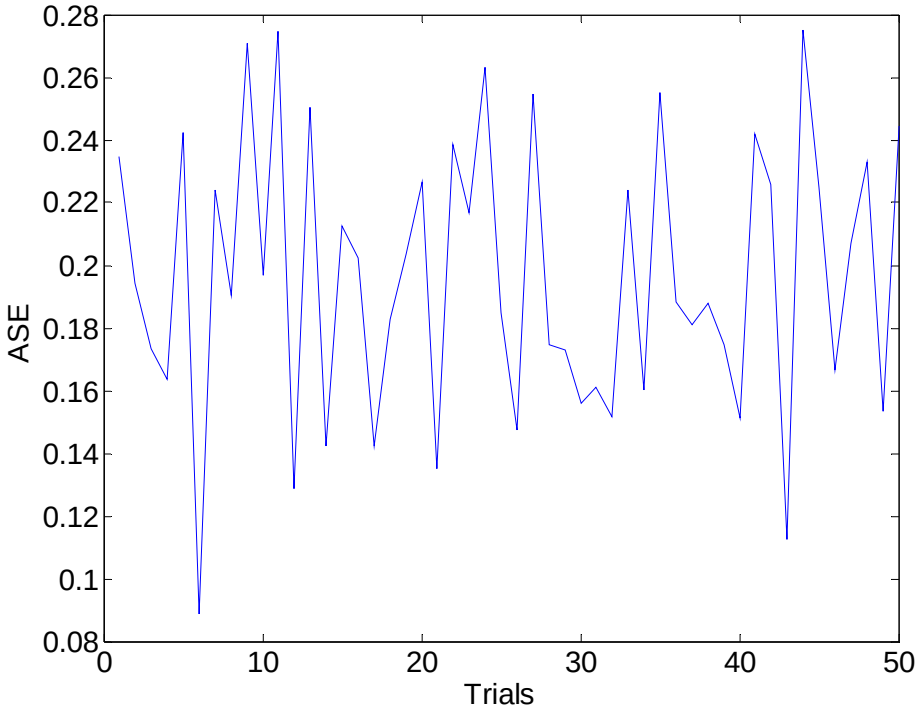


Figure 2-4. The ASE values obtained from 50 realizations of ESNs with the same spectral radius

Maximizing ASE means that the diversity of the states over time is the largest and should provide a basis set that is as uncorrelated as possible. This condition is unfortunately not a guarantee that the ESN so designed will perform the best, because the basis set is independent of the desired response and the application may require a small spectral radius. However, when the desired response is not accessible for the design of the ESN bases or when the same reservoir is to be used for a number of problems, the default strategy should be to maximize the ASE of the state vector.

CHAPTER 3
DESIGN OF THE DYNAMICAL RESERVOIR

In ESNs, reservoir weights are selected randomly with the constraint of the echo state condition (Jaeger, 2001). However, such a random selection scheme is not optimal as it is demonstrated in chapter 2. In this chapter, we propose a design scheme for the design of reservoir weights that will lead to small approximation errors for a variety of tasks. For optimal approximation with a finite number of basis functionals, both the projection space (bases) and the projector (readout) require knowledge from the input and desired responses. However, in ESNs, the projection space is determined solely by the architecture of the ESN reservoir and the input signal (since **W** is fixed), without any knowledge of the space spanned by the desired target signal. The selection of basis functions with the knowledge of input signal only is an ill-posed problem. However, the selection of the reservoir weights must still be done using some rule, and here we hypothesize that a good design strategy is to let the ESN states cover with equal resolution the projection space to anticipate any possible mapping requirement (dictated by the unknown desired response).

Design of the Recurrent Connections

According to the interpretation of ESNs as coupled linear systems, the design of the internal connection matrix, **W**, will be based on the distribution of the poles of the linearized system around zero state. Our proposal is to design the ESN such that the linearized system has uniform pole distribution inside the unit circle of the z-plane. With this design scenario, the system dynamics will include uniform coverage of time constants arising from the uniform distribution of the poles, which also decorrelates as much as possible the bases functionals. This principle was chosen by analogy to the identification of linear systems using Kautz filters (Kautz, 1954) which shows that the best approximation of a given transfer function by a linear

system with finite order is achieved when poles are placed in the neighborhood of the spectral resonances. When no information is available about the desired response or when would like to use the same reservoir for a variety of tasks, we should uniformly spread the poles to anticipate good approximation to arbitrary mappings.

We again use a maximum entropy principle to distribute the poles inside the unit circle uniformly. The constraints of a circle as boundary conditions for discrete linear systems and complex conjugate locations are easy to include for the pole distribution (Thogula, 2003). The poles are first initialized at random locations; the quadratic Renyi's entropy is calculated using Equation 2-6 and poles are moved such that the entropy of the new distribution is increased over iterations (Erdogmus et al., 2003). This method is efficient to find a uniform coverage of the unit circle with an arbitrary number of poles. However, any other method can be used to find the location of poles with uniform coverage. Notice that this operation has to be done only once for a given number of poles, which is equal to the number of hidden unit PEs.

The system with the uniform pole locations can be interpreted using linear system theory. The poles that are close to the unit circle correspond to many sharp band pass filters specialized in different frequency regions whereas the inner poles realize filters of larger frequency support. Moreover, different orientations (angles) of the poles create filters of different center frequencies.

Now, our problem is to construct a weight matrix from the pole locations. This is equivalent to the problem of designing **W** when its eigenvalues are known. In principle we would like to create a sparse matrix, so we started with the sparsest matrix (with an inverse) which is the direct canonical structure given by (Kailath, 1980)

$$W = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{N-1} & -a_N \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{3-1}$$

The characteristic polynomial of **W** is

$$l(s) = \det(s\mathbf{I} - \mathbf{W}) = s^N + a_1 s^{N-1} + a_2 s^{N-2} + \dots + a_N = (s - p_1)(s - p_2)\dots(s - p_N) \tag{3-2}$$

where p_i's are the eigenvalues and a_i's are the coefficients of the characteristic polynomial of **W**. Here, we know the pole locations of the linear system obtained from the linearization of the ESN, so using Equation 3-2, we can obtain the characteristic polynomial and construct **W** matrix in the canonical form using Equation 3-1. We will name the ESN constructed based on the uniform pole principle ASE-ESN. All other possible solutions with the same eigenvalues can be obtained by $\mathbf{Q}^{-1}\mathbf{W}\mathbf{Q}$, where **Q** is any nonsingular matrix.

To corroborate our hypothesis, we would like to show that the linearized ESN designed with the internal connection weight matrix having the eigenvalues uniformly distributed inside the unit circle creates higher ASE values for a given spectral radius compared to other ESNs with random internal connection weight matrices. We will consider an ESN with 30 states, and use our procedure to create the uniformly distributed linearized ASE-ESN matrix for different spectral radius between [0.1, 0.95]. Similarly, we constructed ESNs with sparse random **W** matrices with different sparseness constraints. This corresponds to a weight distribution having the values 0, c and - c with probabilities p_1, (1- p_1)/2 and (1- p_1)/2, where p_1 defines the sparseness of **W** and c is a constant that takes a specific value depending on the spectral radius. We also created **W** matrices with values uniformly distributed between -1 and 1 (U-ESN), and scaled to obtain a given spectral radius (Jaeger and Hass, 2004). Then, for different $\mathbf{W}^{in}$ matrices, we run the ASE-ESNs with the sinusoidal input of period 20 and calculate ASE.

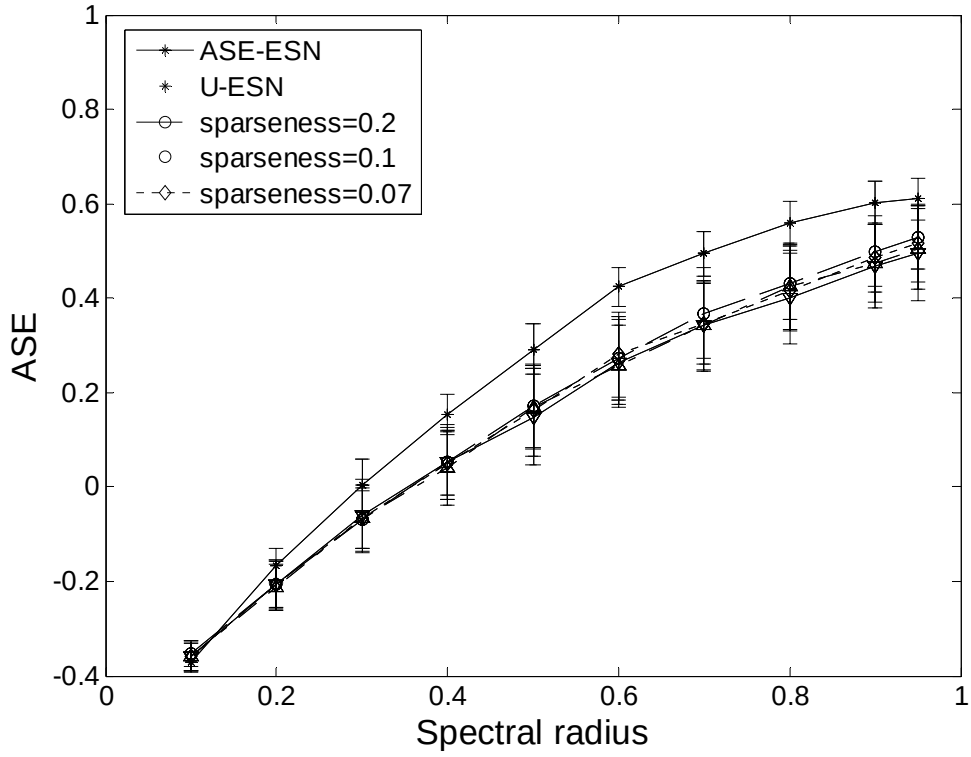


Figure 3-1. Comparison of ASE values obtained for ASE-ESN with U-ESN and randomly generated ESNs with different sparseness. As observed from the figure, the ASE-ESN with uniform pole distribution generates higher ASE on the average for all spectral radii compared to ESNs with random connections.

Figure 3-1 compares the ASE values obtained using different ESNs over 1000 realizations. As observed from the figure, the ASE-ESN with uniform pole distribution generates higher ASE on the average for all spectral radii compared to ESNs with sparse and uniform random connections. This approach is indeed conceptually similar to Jeffreys' maximum entropy prior (Jeffreys, 1946): it will provide a consistently good response for the largest class of problems. Concentrating the poles of the linearized system in certain regions of the space, only provide good performance if the desired response has energy in this part of the space, as is well known from the theory of Kautz filters (Kautz, 1954).

Design of the Adaptive Bias

In conventional ESNs, only the output weights are trained optimizing the projections of the desired response onto the basis functions (echo states). Since, the dynamical reservoir is fixed, the basis functions are only input dependent. However, since function approximation is a problem in the joint space of the input and desired signals, a penalty in performance will be incurred. From the linearization analysis that shows the crucial importance of the operating point of the PE nonlinearity in defining the echo state dynamics, we propose to use a single external adaptive bias to adjust the effective spectral radius of an ESN. Notice that, according to linearization analysis with tanh nonlinearity, bias can only reduce spectral radius. The information for adaptation of bias is the MSE in training, which modulates the spectral radius of the system with the information derived from the approximation error. With this simple mechanism, some information from the input output joint space is incorporated in the definition of the projection space of the ESN. The beauty of this method is that the spectral radius can be adjusted by a single parameter that is external to the system without changing reservoir weights.

The training of bias can be easily accomplished. Indeed, since the parameter space is only one dimensional, a simple line search method can be efficiently employed to optimize the bias. Among different line search algorithms, we will use a search that uses Fibonacci numbers in the selection of points to be evaluated (Wilde, 1964). The Fibonacci search method minimizes the maximum number of evaluations needed to reduce the interval of uncertainty to within the prescribed length. In our problem, a bias value is picked according to Fibonacci search. For each value of bias, training data is applied to the ESN and the echo states are calculated. Then, the corresponding optimal output weights and the objective function (MSE) are evaluated to pick the next bias value.

Alternatively, gradient based methods can be utilized to optimize the bias, due to simplicity and low computational cost. System update equation with an external bias signal, b, is given by

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{in}b + \mathbf{W}\mathbf{x}(n))$$

The update equation for $\mathbf{b}$ is given by

$$\frac{\partial O(n+1)}{\partial b} = -\mathbf{e}\cdot\mathbf{W}^{out} \times \frac{\partial \mathbf{x}(n+1)}{\partial b}$$
$$= -\mathbf{e}\cdot\mathbf{W}^{out} \times \left[\dot{\mathbf{f}}(net_{n+1})\cdot(\mathbf{W}\times\frac{\partial \mathbf{x}(n)}{\partial b} + \mathbf{W}^{in})\right]$$

Here, O is the MSE defined previously. This algorithm may suffer from similar problems observed in gradient based methods in recurrent networks training. However, we observed that the performance surface is rather simple. Moreover, since the search parameter is one dimensional, the gradient vector can only assume one of the two directions. Hence, imprecision in the gradient estimation should affect the speed of convergence, but normally not change the correct gradient direction.

Experiments

This section presents a variety of experiments in order to test the validity of the ESN design scheme proposed in the previous section.

Short-term Memory Structures

Modeling a dynamical system with input-output data requires access to a sufficiently long input history as the output of the dynamical system depends not only the current value of input signal but also the history of it. In signal processing, the common approach to overcome this difficulty is to embed the input signal using a tapped delay line (in FIR filter or TDNN). RNNs

provide a different type of embedding through the recurrent connections between the PEs. However, the length of the memory achieved is not as clear as in tapped delay line.

This experiment compares the short term memory (STM) capacity of ESNs with the same spectral radius using the framework presented in (Jaeger, 2002). Consider an ESN with a single input signal, *u(n)*, optimally trained with the desired signal *u(n-k)*,for a given delay *k*. Denoting the optimal output signal $y_k(n)$, the *k*-delay STM capacity of a network, MC_k, is defined as squared correlation coefficient between *u(n-k)* and $y_k(n)$ (Jaeger, 2002). The STM capacity, *MC*, of the network is defined as $\sum_{k=1}^{\infty} MC_k$. STM capacity measures how accurately the delayed version of the input signal is recovered with optimally trained output units. It has been shown in (Jaeger, 2002) that the memory capacity for recalling an independent identically distributed (i.i.d) input with an *N* unit RNN with linear output units is bounded by *N*.

Consider an ESN with 20 PEs and a single input unit. ESN is driven by an i.i.d random input signal, *u(n)*, that is uniformly distributed over [-0.5, 0.5]. The goal is to train the ESN to generate the delayed versions of the input, *u(n*-1), *u(n*-2), …, *u(n*-40),. We used four different ESNs, R-ESN, U-ESN, ASE-ESN and BASE-ESN. R-ESN is a randomly connected ESN used in (Jaeger, 2002) where the entries of **W** matrix are set to 0, 0.47, -0.47 with probabilities 0.8, 0.1, 0.1, respectively. This corresponds to a sparse connectivity of 20% and a spectral radius of 0.9. The entries of are uniformly distributed over [-1, 1] and scaled to obtain the spectral radius of 0.9. ASE-ESN also has a spectral radius of 0.9 and is designed with uniform poles. BASE-ESN has the same recurrent weight matrix as ASE-ESN and an adaptive bias at its input. In each ESN, the input weights are set to 0.1 or -0.1 with equal probability and direct connections from the input to the output are allowed whereas $\mathbf{W}^{back}$ is set to 0 (Jaeger, 2002). The echo states are calculated using Equation 2-1 for 200 samples of the input signal and the first 100 samples corresponding to

initial transient are eliminated. Then, the output weight matrix is calculated using Equation 2-4. For the BASE-ESN, the bias is trained for each task. All networks are run with a test input signal and the corresponding output and MC_k are calculated.

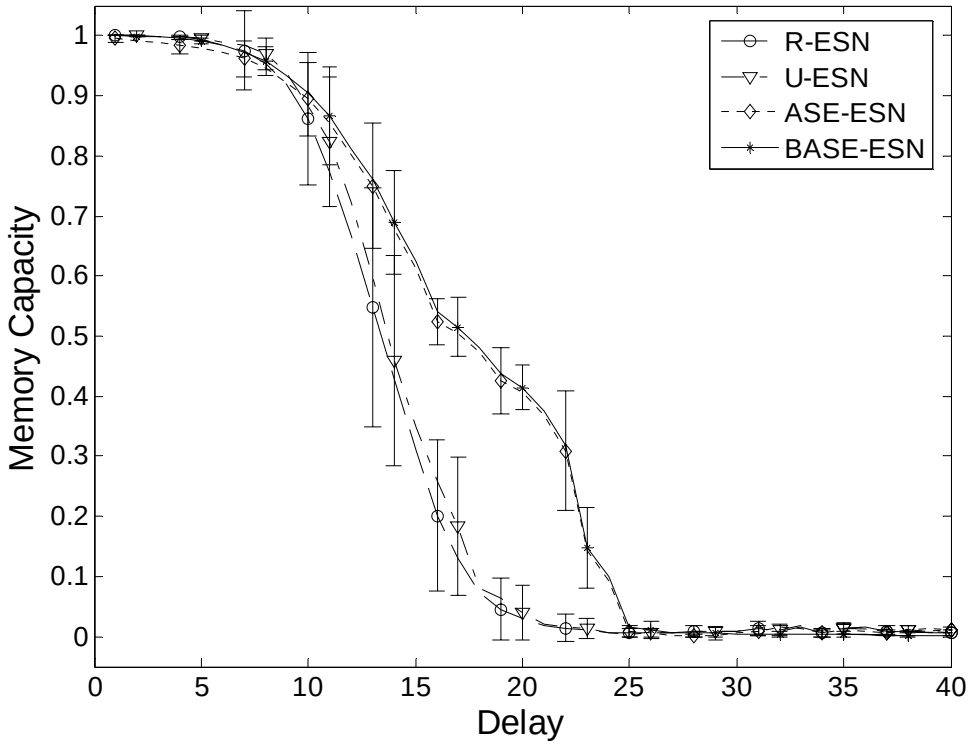


Figure 3-2. The *k*-delay STM capacity of each ESN for delays 1, . . . ,40 computed using the test signal. The STM capacities of R-ESN, U-ESN, ASE-ESN and BASE-ESN are 13.09, 13.55, 16.70 and 16.90, respectively.

Figure 3-2 shows the *k*-delay STM capacity (averaged over 100 trials) of each ESN for delays 1, , for the test signal. The STM capacities of R-ESN, U-ESN, ASE-ESN and BASE-ESN are 13.09, 13.55, 16.70 and 16.90, respectively. First of all, ESNs with uniform pole distribution (ASE-ESN and BASE-ESN) have MCs that are much longer than the randomly generated ESN given in (Jaeger, 2002) in spite of all having the same spectral radius. In fact, the STM capacity of ASE-ESN is close to the theoretical maximum value of N=20. A closer look at the figure shows that R-ESN performs slightly better than ASE-ESN for delays less than 9.

Indeed, for small k, large ASE degrades the performance because the tasks do not need long memory depth. However, the drawback of high ASE for small k is recovered in BASE-ESN which reduces the ASE to the appropriate level required for the task. Overall, the addition of the bias to the ASE-ESN increases the STM capacity from 16.70 to 16.90. On the other hand, U-ESN has slightly better STM compared to R-ESN with only 3 different weight values although U-ESN has more distinct weight values than R-ESN. It is also significant to note that the MC will be very poor for an ESN with smaller spectral radius even with an adaptive bias since the problem requires large ASE and bias can only reduce ASE. This experiment demonstrates the need for maximizing ASE in ESN design.

Binary Parity Check

The effect of the adaptive bias was marginal in the previous experiment since the nature of problem required large ASE values and long short-term memory. However, there are tasks in which the optimal solutions require smaller ASE values and smaller spectral radius. Those are the tasks where the adaptive bias becomes a crucial design parameter in our design methodology.

Consider an ESN with 100 internal units and a single input unit. ESN is driven by a binary input signal, $u(n)$, that assumes the values 0 or 1. The goal is to train an ESN to generate the m-bit parity, where m is 3,…,8 Similar to the previous experiments, we used the R-ESN, ASE-ESN and BASE-ESN topologies. R-ESN is a randomly connected ESN where the entries of $\mathbf{W}$ matrix are set to 0, 0.06, -0.06 with probabilities 0.8, 0.1, 0.1, respectively. This corresponds to a sparse connectivity of 20% and a spectral radius of 0.3. ASE-ESN and BASE-ESN are designed with a spectral radius of 0.9. The input weights are set to 1 or -1 with equal probability and direct connections from the input to the output are allowed whereas $\mathbf{W}^{back}$ is set to 0. The echo states are calculated using Equation 2-1 for 1000 samples of the input signal and the first 100 samples corresponding to initial transient are eliminated. Then, the output weight matrix is calculated

using Equation 2-4. For ESN with adaptive bias, the bias is trained for each task. Binary decision is made by a threshold detector that compares the output of the ESN to 0.5. Figure 3-3 shows the number of wrong decisions (averaged over 100 different realizations) made by each ESN for m=3,…,8.

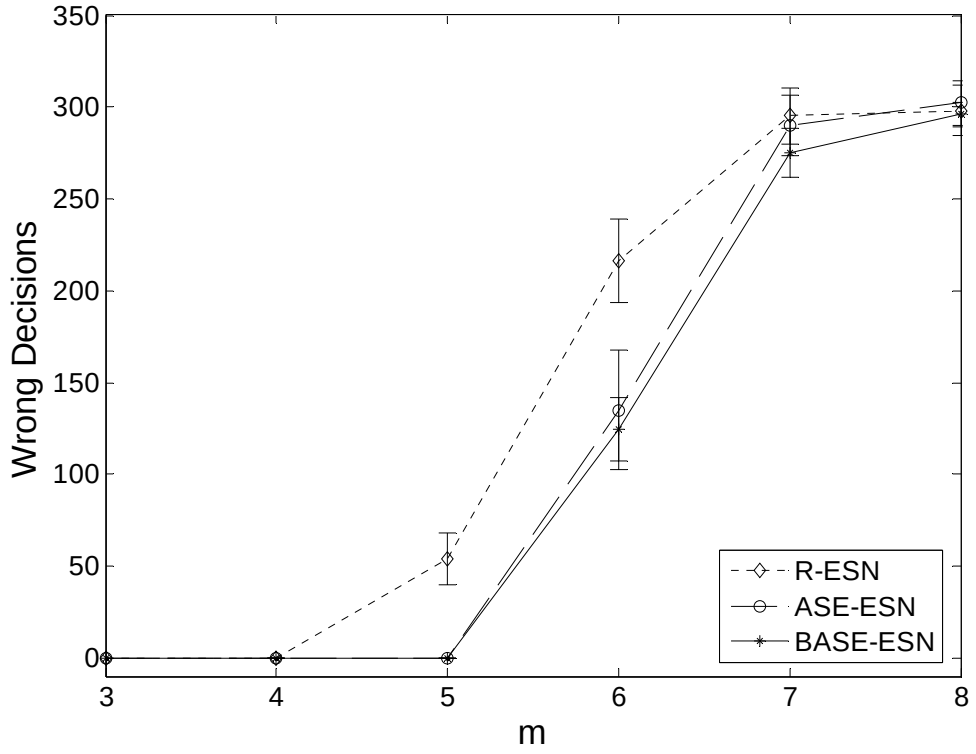


Figure 3-3. Number of wrong decisions made by each ESN for m = 3, . . . ,8 in the binary parity check problem. The total number of wrong decisions for m = 3, . . . ,8 of R-ESN, ASE-ESN and BASE-ESN are given by 722, 862 and 699.

The total number of wrong decisions for m=3,…,8 of R-ESN, ASE-ESN and BASE-ESN are 722, 862 and 699, respectively. ASE-ESN performs poorly since the nature of the problem requires short time-constant for fast response but ASE-ESN has large spectral radius. For 5-bit parity, the R-ESN has no wrong decisions whereas ASE-ESN has 47 wrong decisions. BASE-ESN performs a lot better than ASE-ESN and slightly better than the R-ESN since the adaptive bias reduces the spectral radius effectively. Note that for m = 7 and 8, the ASE-ESN performs

similar to the R-ESN, since the task requires access to longer input history, which compromises the need for fast response. Indeed, the bias in the BASE-ESN takes effect when there is errors (m>4) and when the task benefits from smaller spectral radius. The optimal bias values are approximately 3.2, 2.8, 2.6 and 2.7 for m = 3, 4, 5, and 6, respectively. For m = 7 or 8, there is a wide range of bias values that result in similar MSE values (between 0 and 3). In summary, this experiment clearly demonstrates the power of the bias signal to configure the ESN reservoir according to the mapping task.

System Identification

This section presents a function approximation task where the aim is to identify a nonlinear dynamical system. The unknown system is defined by the difference equation $y(n+1) = 0.3y(n) + 0.6y(n-1) + f(u(n))$ where $f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$. The input to the system is chosen to be sin(2πn/25).

We used an ESN with 30 internal units and a single input unit. Again, we used three different ESNs, R-ESN, ASE-ESN and BASE-ESN. The **W** matrices of all ESNs are generated as described in chapter 3.3.1 except that they are scaled to have a spectral radius of 0.95. In each ESN, the input weights are set to 1 or -1 with equal probability and direct connections from the input to the output are allowed whereas $\mathbf{W}^{back}$ is set to **0**. The optimal output signal is calculated for each ESN as described in section 3.3.1. The MSE values (averaged over 100 realizations) for R-ESN, ASE-ESN are $1.23\text{x}10^{-5}$ and $1.83\text{x}10^{-6}$, respectively. The addition of the adaptive bias to the ASE-ESN reduces the MSE value from $1.83\text{x}10^{-6}$ to $3.27\text{x}10^{-9}$. This experiment illustrates the superiority of the proposed design scheme for function approximation.

CHAPTER 4
DESIGN OF THE READOUT

In the previous chapter, we have discussed the design of the ESN reservoir and the bias. This chapter deals with the design of the readouts for ESNs. The problem is studied for function approximation and pattern recognition separately and different solutions are proposed for each case.

Design of the Readout for Function Approximation

The usual readout is the linear regression network. Throughout this dissertation, we will use the linear readout. The equation for the linear readout is given by

$$\mathbf{y}(n+1) = \mathbf{W}^{out}\mathbf{x}(n+1)$$

Alternative readouts are nonlinear readouts such as multilayer perceptron or local nonlinear modelling. This issue is outside the scope of this dissertation.

Design of the Readout for Dynamical Pattern Recognition

The applications of ESNs to problems such as system identification, control of dynamical systems, and time-series prediction has been extensively studied both in this work and in the literature (Jaeger, 2001; Jaeger and Hass, 2004; Ozturk et al., 2007; Prokhorov, 2005). However, the use of ESNs for dynamical pattern recognition has been limited. In the literature, RNNs and other neural architectures such as time-delay neural network have been utilized for dynamical pattern recognition tasks by minimizing the mean-square error (MSE) with respect to a label (Haykin, 1998; Bishop, 1995; Iso and Watanabe, 1991; Tebelskis, 1995). If the pattern to be classified is a time signal, the classification process is different from the classification of a static pattern. The dynamical classification problem can be reduced to a static one by treating each sample of the time series individually. In this case, the time series is not treated as a whole and temporal information between samples is ignored. Moreover, one class label is associated with

each time instant; therefore, post classification such as majority voting has to be applied. An alternative to this approach is embedding the time series to populate a short-term history of the pattern. Again, a static mapper can be used to classify the embedded pattern. In this approach, classification becomes extremely hard, if the dimensionality of the input space becomes too large. Alternatively, temporal processing of the time series can be done inside the classifier. In this approach, the classifier, which has a built-in memory, can accept single elements of time series. Although the classifier does not require external temporal processing, the design of the classifier may be complicated. Another major difficulty in dynamical pattern recognition is how to design the label, which should also be a time series. The obvious method is to enforce a class label (+1 or -1) as the desired response throughout the presentation of the input pattern. However, the difficulty with this method is that the desired response is forced to be constant independent of the input signal dynamics. An alternative powerful technique is to create a one-step predictor for each class in order to capture the dynamics of the class generating the time-series (Zahalka and Principe, 1993). Then, during testing, the test pattern is presented to all predictors and the label of the predictor with the least prediction error is assigned as the class of the input pattern. This method is a smart way of converting the pattern recognition problem into a regression problem where many techniques are available. This method can be easily applied to ESNs with an adaptive linear readout and used for classification of temporal patterns, but it is still dependent upon input normalization and it is sensitive to pattern duration changes.

In this chapter, we propose an alternative readout for ESNs for temporal pattern recognition and compare it with the standard linear readout of ESNs and other conventional techniques used in the literature. The goal is to design an ESN/LSM readout that will recognize a *class of inputs* that differ by some quantity (e.g. amplitude or shape considered as a distortion

parameter). The proposed readout is based on a biologically plausible linear associative memory (LAM) that implements a correlator specially trained for high specificity. The readout, called the minimum average correlation energy (MACE) filter, is adopted from optical pattern recognition literature, where it is used for recognition of a given object in 2-D images in the presence of noise, geometric distortions and pose changes (Mahalanobis et al., 1987). The MACE filter is a correlator whose weights are determined in close form by solving a minimization problem with a constraint, and it has been shown that the method is equivalent to a cascade of a pre-whitening filter for the image class followed by a LAM (Fisher, 1997). The recognition of a time pattern in a single time series is achieved by feeding the time series to an ESN/LSM, and utilizing a time window of states which can be interpreted as a 2-D image, one dimension being time and the other processing element number (space). Several MACEs are trained, one for each class. During testing, the unknown time signal is fed to the ESN and the states are correlated with each template image and the class label of the template with the highest correlation is assigned to the input pattern. An interesting implication of the MACE is that it can also be used for LSMs with proper regularization of the MACE solution. In LSMs, the proposed readout operates directly in the spike domain without converting the liquid state spike train outputs into continuous signals by binning or low-pass filtering.

In this chapter, we first present a brief background on linear associative memories followed by the detailed explanation of the MACE filter. Then, the use of MACE filter as the readout for ESN/LSMs is discussed. Finally, the experiments are presented to compare the proposed readout with the conventional methods.

Linear Associative Memories

A linear associative memory is a multiple input multiple output linear system that implements an association (memory) between input and desired outputs (Hinton and Anderson,

1981). The output is automatically created by the presentation of the proper input, which is referred as content addressability. LAMs are quite different from the concept of memory used in digital computers, which uses a memory location whose contents are accessible by its address. Unlike digital memory, LAMs store information in a more global way by having several PEs instead of local bit storage. Therefore, they are also robust to noise (Haykin, 1998; Hinton and Anderson, 1981). LAMs recall by association, hence the pattern that is closest to the input is recalled. LAMs are biologically plausible just like ESN/LSM and they stand as the most likely model for human memory (Principe et al., 2000).

Forced Hebbian rule can be used to train a LAM that associates a P dimensional input $\mathbf{p}$ to a Q dimensional desired output $\mathbf{q}$. The output of the LAM is given by $\mathbf{q}=\mathbf{W}^{\mathrm{LAM}}\mathbf{p}$, where QxP matrix $\mathbf{W}^{\mathrm{LAM}}$ can be computed using the outerproduct rule, $\mathbf{W}^{\mathrm{LAM}}=\mathbf{q}\mathbf{p}^{\perp}$ (Principe et al., 2000; Haykin, 1998). Multiple input-output vector pairs ($\mathbf{p}_k$,$\mathbf{q}_k$, for k=1,…K) can also be stored in the LAM by repeated presentation of each input. Using the superposition principle for the linear network, the final weight matrix is given by

$$\mathbf{W}^{LAM} = \sum_{k=1}^{K} \mathbf{q}_k \mathbf{p}_k^{\perp}$$

For an input vector $\mathbf{p}_l$, the output of the LAM is given by

$$\mathbf{y}_k = \mathbf{W}^{LAM}\mathbf{p}_l = \mathbf{q}_l \mathbf{p}_l^{\perp}\mathbf{p}_l + \sum_{k=1,k\neq l}^{K} \mathbf{q}_k \mathbf{p}_k^{\perp}\mathbf{p}_l \ .$$

The first term in the LAM output is the desired output and the second term, which is called *crosstalk*, is the interference of the other stored patterns to the true pattern. If the crosstalk term is small, the LAM can retrieve the pattern corresponding to the input. Notice that if the stored patterns are orthogonal to each other, the crosstalk term will be zero resulting in a perfect recall. Therefore, the number of patterns that can be stored perfectly in a LAM is limited by the size of

input space, which is one of the limitations of LAMs for information processing (Principe et al., 2000; Haykin, 1998).

The output patterns $\mathbf{q}_k$ for the pairs $(\mathbf{p}_k, \mathbf{q}_k)$, k=1,…,K, can be interpreted as the desired response for the LAM. The existence of a desired response allows the application of supervised learning to train LAMs using unconstrained optimization techniques such as least squares. For instance, the weights can be modified using the least-mean-squares algorithm by an amount given by

$$\Delta \mathbf{W}^{LAM} = \eta e_k \mathbf{p}_k^{\perp} = \eta \mathbf{q}_k \mathbf{p}_k^{\perp} - \eta \mathbf{y}_k \mathbf{p}_k^{\perp}.$$

This equation is a combination of the desired forced Hebbian ($\eta \mathbf{q}_k \mathbf{p}_k^{\perp}$) and an anti-Hebbian term ($-\eta \mathbf{y}_k \mathbf{p}_k^{\perp}$) that decorrelates the present output $\mathbf{y}_k$ from the input $\mathbf{p}_k$. The anti-Hebbian term reduces the crosstalk at each iteration improving the performance of LAMs in terms of crosstalk. A LAM trained with LMS is called an optimal linear associative memory (OLAM) (Principe et al., 2000).

One question follows naturally: what is the difference between a LAM and a linear regressor? The linear regressor passes a single hyperplane by *all* the desired samples, so we normally want more patterns than the size of the input dimension to *generalize* but it looses information about the covariance on the data space (Principe et al., 2000). On the other hand, in the LAM, the output is as close as possible to *each* of the training samples. This different behavior is controlled by the number of data points used for training, which must be must be less than the input dimensionality to minimize crosstalk. The MACE filter that will be discussing next is a cascade of a preprocessor (a whitening filter over the class of training data) followed by a LAM (Fisher, 1997).

The MACE Filter

The technique of matched spatial filters (MSF) or correlators for optical pattern recognition has been well studied and used for the recognition of 2-D objects (Vanderlugt, 1964; Kumar, 1986). The goal is to design an optimal correlator filter that represents an *object class* under noise and/or distortions using a set of training images. The reason it is possible to match a *single* template to a *class* of objects is because of the huge number of weights (degrees of freedom) that the correlator has in 2-D (N^2 weights for an NxN input image). Moreover, MACE filter can incorporate the *covariance* information of the class in its weights. The MACE filter is the most widely used MSF due to superior discrimination properties (Mahalanobis, 1987; Casasent and Ravichandran, 1992).

The formulation of the MACE will be presented in the frequency domain for convenience and ease of implementation (Mahalanobis, 1987). Consider a set of training images for a single object class. The i^{th} training image (i=1,..,K) is described as a 1-D discrete sequence (obtained by lexicographic ordering the columns of the image) denoted by $\underline{\mathbf{x}}_i = [x_i(1), x_i(2), ..., x_i(d)]^T$, where d and K are the number of pixels in the image and the number of training images for the class, respectively. The discrete Fourier transform of $\underline{\mathbf{x}}_i$ is denoted by $\underline{\mathbf{X}}_i$. Define the overall dxK training matrix by $\underline{\mathbf{X}} = [\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, ..., \underline{\mathbf{X}}_K]$. The dx1 column vector $\underline{\mathbf{h}}$ denotes the coefficients of a filter in space domain and $\underline{\mathbf{H}}$ its Fourier transform. The correlation function of the i^{th} image with the filter $\underline{\mathbf{h}}$ is given by $\underline{\mathbf{g}}_i = \underline{\mathbf{h}} \otimes \underline{\mathbf{x}}_i$. The filter output g defines a 3-dimensional space where x and y axes are the indices (lags) of the 2-dimensional correlation function and z-axis is the value of the correlation for the corresponding indices. We will call it the correlation space. The energy of the i^{th} correlation plane in the frequency domain is $E_i = \underline{\mathbf{H}}^{\perp} \underline{\mathbf{D}}_i \underline{\mathbf{H}}$ where $\underline{\mathbf{D}}_i$ is a dxd diagonal

matrix whose elements are the magnitude squares of the associated elements of $\underline{\mathbf{X}}_i$. The average correlation plane energy over the training images is defined as

$$E_{av} = (1/K)\sum_{i=1}^{K} E_i = (1/K)\underline{\mathbf{H}}^{\perp}\left(\sum_{i=1}^{K}\underline{\mathbf{D}}_i\right)\underline{\mathbf{H}} = (1/K)\underline{\mathbf{H}}^{\perp}\underline{\mathbf{D}}\underline{\mathbf{H}}$$ (Mahalanobis, 1987).

The MACE filter solution designs a correlation filter that minimizes the output energy while providing predetermined values ($\mathbf{c} = [c_1, c_2, ..., c_K]^T$) at the origin of the correlation plane. Each c_i is the i^{th} peak correlation value obtained by correlating the i^{th} image with the filter $\underline{\mathbf{H}}$ at zero lag. More specifically, the problem is to solve the constraint optimization problem (Mahalanobis, 1987):

$$\min_{\underline{H}} \ (1/K)\underline{\mathbf{H}}^{\perp}\underline{\mathbf{D}}\underline{\mathbf{H}} \ \text{ subject to } \ \underline{\mathbf{X}}^{\perp}\underline{\mathbf{H}} = \mathbf{c}$$

The solution to this problem can be found analytically using the method of Lagrange multipliers (Mahalanobis, 1987) and is given by

$$\underline{\mathbf{H}} = \underline{\mathbf{D}}^{-1}\underline{\mathbf{X}}(\underline{\mathbf{X}}^{\perp}\underline{\mathbf{D}}^{-1}\underline{\mathbf{X}})^{-1}\mathbf{c} \qquad (4\text{-}1)$$

The selection of the exact value of **c** is somewhat arbitrary since it simply scales the peak correlation amplitude. Usually, the entries of **c** are chosen to be 1 for in-class training images and a smaller value (around 0.1) for out-of-class images if at all used (Mahalanobis, 1987). The MACE filter solution achieves superior discrimination properties when compared to other existing correlation filters such as synthetic discriminant function filters (Mahalanobis, 1987; Casasent and Ravichandran, 1992).

The ESN/LSM - MACE for Dynamical Pattern Recognition

Pattern recognition is normally formulated as the design of a nonlinear static system trained discriminantly with the information of a set of labels, with data that are vectors in R^n. An alternative to this approach is obtained by creating models for each class which are only trained

with the class data (nondiscriminant training). When the number of classes to be classified is unknown at the training time (open set classification), the latter provides robust results. The correlator or matched filter is a very well known example of a system that recognizes a single class data (Helstrom, 1995; Proakis, 2001), and can be easily extended to multiple dimensions.

Dynamical pattern recognition can also be formulated in a similar manner by using dynamical systems as classifiers and training them discriminantly with an appropriate desired signal. Instead, in this study, we propose to use the MACE filter as a readout for ESN/LSMs for dynamical pattern recognition tasks, nondiscriminantly trained for each class. In ESN/LSM, the proposed associative memory considers a time window of states which can be interpreted as a 2-D image, one dimension being time and the other the space of the states (see Figure 4-1). Let us explain the operation of ESN-MACE in mathematical terms.

Assume that we have P different classes of temporal patterns. The goal is to compute one MACE filter, $\underline{\mathbf{h}}^p$ for each class. Assume also that for each class, the training set consists of K input patterns each with a particular temporal shape of length T. The procedure is to compute each $\underline{\mathbf{h}}^p$ for p=1,…,P as follows.

1. The i^{th} training input pattern for the p$^{\text{th}}$ class $\mathbf{u}_i = [\mathbf{u}_i(1), \mathbf{u}_i(2), ..., \mathbf{u}_i(T)]$ of dimension MxT is used to calculate the echo states using Equation 1-1.

2. The resulting NxT matrix of echo states forms the equivalent of a 2-D image in section 2.2 (Figure 4-1). The echo states are then lexicographically ordered by the columns to get the 1-D column vector $\underline{\mathbf{x}}_i = [\mathbf{x}_i(1)^T, \mathbf{x}_i(2)^T, ...\mathbf{x}_i(T)^T]^T$ with NxT elements. Here each $\mathbf{x}_i(n)$ is an Nx1 vector with the value of the echo state at time n for the i^{th} training sequence.

3. The discrete Fourier transform of $\underline{\mathbf{x}}_i$ is denoted by $\underline{\mathbf{X}}_i$. Define the overall *d*x*K* training matrix in the frequency domain by $\underline{\mathbf{X}} = [\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, ..., \underline{\mathbf{X}}_K]$.

4. The optimal coefficients of the LAM for the class are computed using Equation 4-1 in the frequency domain and the corresponding MACE filter weights $\underline{\mathbf{h}}^p$ are obtained by inverse discrete Fourier transform. The output of the MACE for the i^{th} training input pattern for the p^{th} class can be obtained by $\underline{\mathbf{x}}_i^T \underline{\mathbf{h}}^p$.

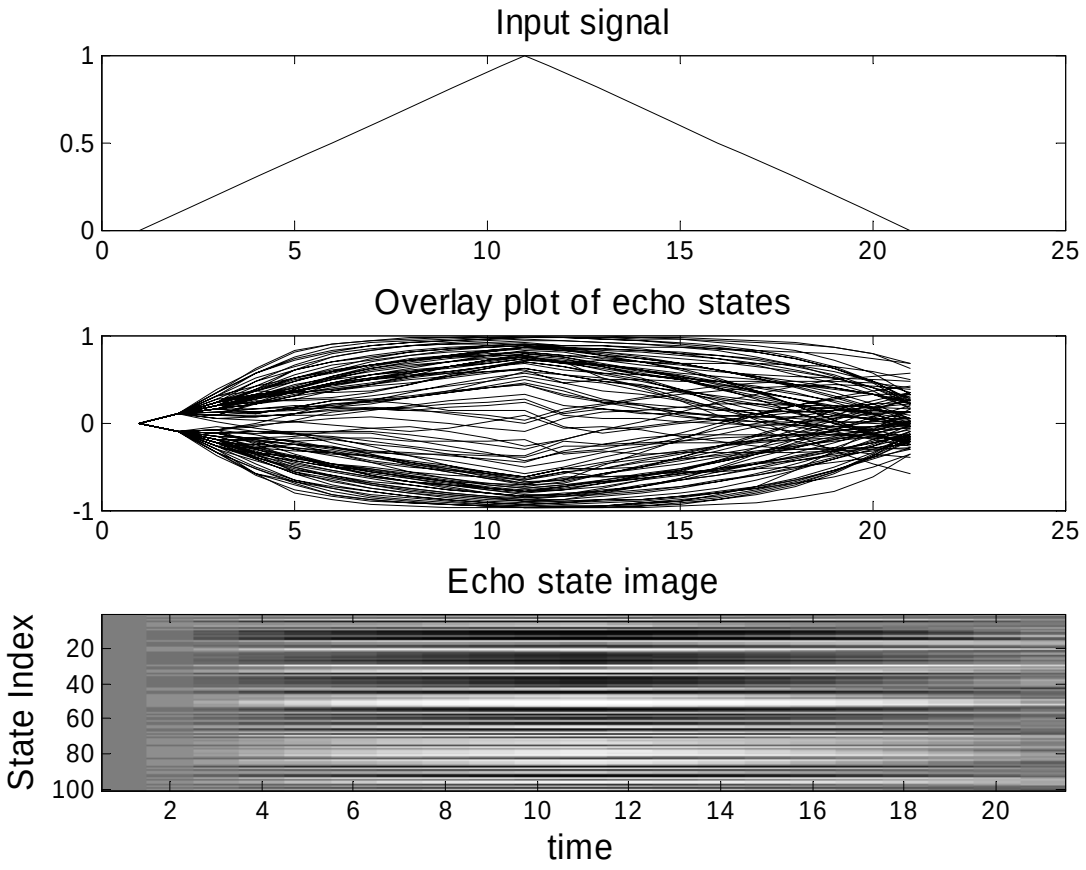


Figure 4-1. Interpretation of echo states as a 2-d image. A) The triangular input signal. B) The echo states of a 100 unit ESN. C) The echo state image where a point in state index and time is interpreted similar to a pixel in an image.

The same procedure is repeated for the training sequences of other classes to obtain an optimal filter for each class.

During testing, the input signal can be either a continuous stream (asynchronous operation) or a series of frames (synchronous operation). Depending on the mode of operation; the choice of the initial conditions for ESN in step 1 varies. In synchronous operation, the timing of the signal and the frames of interest are already known both in training and testing. Therefore, ESN is initialized with the same zero initial condition both in training and testing. The zero initial condition is chosen such that the system dynamics is not biased and the system states are controlled by the input. During testing, the MACE output is calculated for each frame by correlating the filter coefficients with the echo states generated by the frame at zero lag and with a zero initial condition.

When the signal timing is unknown (asynchronous), the MACE filter generates an output at each time instant n by correlating the filter coefficients with the echo states between time instants n-T+1 and n. and the maximum in time above a certain pre specified threshold will be picked to represent the occurrence of the pattern. In case of a continuous stream, the initial conditions for each frame are dictated by the history of the input signal during testing; therefore they are not necessarily zero. In order to mimic the test conditions while training the MACE, the initial conditions of the ESN are not set to zero but determined by the history of the input. In other words, the whole input signal is used to create echo states and then the resulting NxT dimensional echo state images are extracted and used to train the MACE filter using the above algorithm.

The Design of ESNs for Dynamical Pattern Recognition

One of the drawbacks of LAMs is the storage capacity that is limited by the input space dimension (Haykin, 1998). However, the ESN/LSM with a LAM readout becomes very powerful since echo/liquid states provide the LAM with a *user defined high dimensional input space* increasing the number of patterns that can be stored in the LAM with minimal cross talk. In

ESNs, the reservoir states can be interpreted as a set of bases functionals (representation space) constructed dynamically and nonlinearly from the input (Ozturk et al., 2007). Due to the nonlinearity, it is easy to get states that are linearly independent of each other (Ito, 1996; Ozturk et al., 2007). Therefore, the matrices in the MACE filter solution are typically full-rank, allowing the computation of inverses. Although linearly independent, the states are not orthogonal to each other since they are connected through predefined weight values. An information-theoretic-metric, called average state entropy (ASE), has been recently proposed to describe the richness of ESN dynamics (Ozturk et al., 2007). ASE quantifies the distribution of state values across the dynamic range. When ASE is maximized, the states of the system are evenly distributed across the dynamic range instead of being populated only on a few values. The ASE measure is related to the system design by uniformly distributing the poles of the linearized system around the origin (eigenvalues of **W**) (Ozturk et al., 2007). We propose to use the same design methodology to construct the representation layer of ESNs with the LAM readout. This approximates for ESNs the well-known property of orthogonal basis and eases the LAM training. In the experiments performed for this study, we have observed that the numerical rank (the condition number) of the inverted matrices is well-conditioned for ESNs.

Likewise for LSMs, the value of each liquid state at each time instant is interpreted by the MACE as 1 or 0 depending on the existence or absence of a spike. This creates a very sparse state matrix which likely leads to singular matrices in the MACE filter computation. Therefore, the matrices have to be regularized in order to compute appropriate MACE filter weights with the outlined formulation. The regularization can be done by adding zero mean Gaussian noise with a small variance into the states before computing the MACE solution.

Experiments

This section presents a variety of experiments to compare the MACE filter readout proposed for ESN/LSM with the conventional techniques for dynamical pattern recognition.

Classification of temporal signals with unknown peak values

In this experiment, the aim is to classify a temporal signal into one of two possible classes. The two classes are represented by a triangular and a step function of 20 samples. The interesting property of the signals in this experiment is that the peak values are unknown and can assume any value between 0.5 and 1. In other words, each class is not a signal but a *family of signals*. The signals are also corrupted with additive, zero mean, white Gaussian noise.

We compare the performance of three different ESNs, denoted by R-ESN1-MACE, R-ESN2-MACE, ASE-ESN-MACE, with MACE filter readout. All ESNs have 30 PEs and are constructed using different techniques. R-ESN1-MACE is a randomly connected ESN where the entries of **W** matrix are set to 0, 0.47, -0.47 with probabilities 0.8, 0.1, 0.1, respectively. This corresponds to a sparse connectivity of 20% and a spectral radius of 0.9. R-ESN2-MACE is a randomly connected ESN where the entries of the **W** matrix are chosen from a Gaussian distribution with zero mean. The weights are scaled such that the spectral radius is set to 0.9. The internal weight matrix, **W**, of the ASE-ESN-MACE is designed such that the eigenvalues of **W** (poles of the linearized system around origin) are uniformly distributed inside the unit circle. This principle is introduced in (Ozturk et al., 2007) and it distributes the states of the system evenly across the dynamic range instead of populating them only on a few locations. The entries of $\mathbf{W}^{in}$ are set to 1 or -1 with equal probability for all three ESNs.

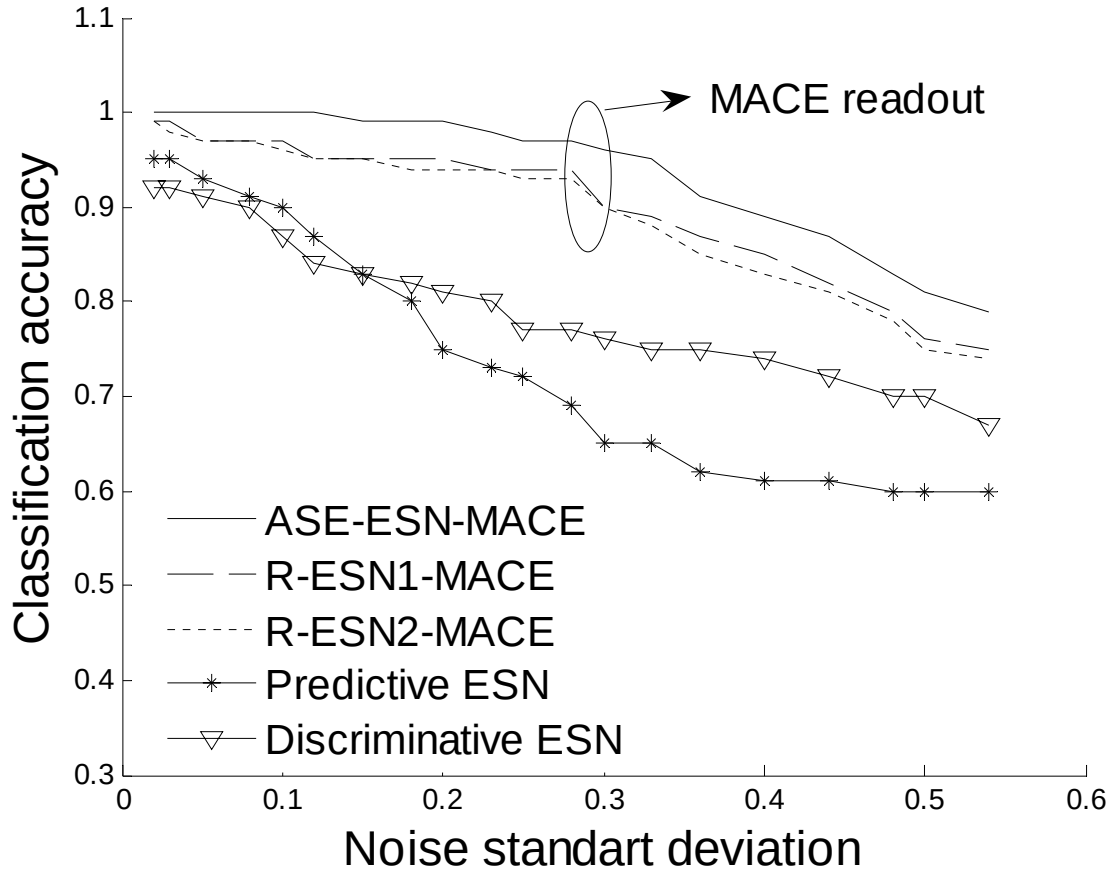


Figure 4-2. Performance comparisons of ESNs for the classification of signals with unknown peak values. ESNs with the MACE filter readout performs better for all noise levels compared to discriminative and predictive ESNs. The classification accuracy of ASE-ESN-MACE is almost perfect up to a noise standard deviation of 0.3. ASE-ESN-MACE performs better than randomly connected ESNs with MACE readout since it distributes the echo states uniformly and increases separation between states corresponding to different classes.

To train the MACE filter readout for the ESNs, 100 different realizations for each class of signals are generated and the echo states are calculated using Equation 2-2 with zero initial conditions. This creates an image of dimension 30x20, where 30 is the number of PEs and 20 is the signal length. One MACE filter is trained for each class using only data from the corresponding class. Output correlation peak amplitudes are assigned to be 1.0 for the training data. The two MACE filters are synthesized in the frequency domain using Equation 4-1 and the

corresponding image plane filters are obtained by inverse discrete Fourier transform. For a given test signal, each filter output is calculated by correlating the echo states of dimension 30x20 with the filter coefficients at zero lag. Figure 4-2 shows the correct classification rate of the 3 methods as the standard deviation of the noise signal varies. The results are averaged over 100 realizations of each ESN.

We also compare the performance of ESNs with MACE filter readout to the ASE-ESN utilizing the conventional linear readout. When the conventional readout is used for dynamical pattern recognition, the problem is the design of the desired signal. One solution is to train a *discriminative readout* with constant target output signal of amplitude 1 for one of the two classes and a constant signal of amplitude -1 for the other class. The duration of the target signals is the same as the duration of the input signal. During testing, echo states, the output signal and the corresponding error signals one for each label are computed for a given input signal. Then, the two MSE values, integrated over the duration of the input pattern, are compared to assign the class of the input pattern to the lowest MSE. In this solution, the readout has to generate a constant signal independent of the input signal dynamics. An alternative also tested is to train two independent *predictive readouts,* one for each class to predict the next sample of the input signal. During testing, the echo states, the output signal and the error signal are computed for each readout. Then, the label of the readout with the lower MSE value, integrated over the duration of the input pattern, is assigned as the class of the input signal. The predictive readout seems a more natural choice since the desired response is not independent of the input dynamics. The same training data used for the MACE filter is employed to calculate the readout weights for the discriminative and predictive ESNs.

Figure 4-2 shows the correct classification rate as the standard deviation of the noise signal varies. The results are averaged over 100 trials. As observed from the figure, ESNs with the MACE filter readout perform better for all noise levels compared to the ESNs with conventional or predictive readouts. The classification accuracy of ASE-ESN-MACE is almost perfect up to a noise standard deviation of 0.3. The correct classification rate of the discriminative ESN for the no noise case, is only 0.92, while for the predictive ESN is 0.95, which is slightly better than the discriminative ESN. Since all three systems have the same reservoir, the better performance of the MACE readout is attributed to the fact that the problem requires modeling a class of functions all at once in a single template and the MACE filter is able to build such a template containing covariance information about the full training set. The conventional and predictive readouts train for the best compromise of the input class peak amplitude of the training set. Being a nonlinear system, ESN dynamics are coupled to the input signal. Therefore, the response of the system to input signals with varying amplitudes can be quite different making the task of projecting the states to a constant value very challenging. The wave shape is also challenging, since for the pulse (a sequence of zero values followed by a sequence of ones) the linear readout has to transform the highly irregular states generated by this pattern to a constant value. Similarly, for the predictive ESN the variation in the input amplitudes affects the ESN dynamics, and the performance drops fast as the noise standard deviation is increased. Morever, the MACE filter readout is expected to be more robust w.r.t selection of $\mathbf{W}$ and $\mathbf{W}^{in}$ in the ESN design compared to the conventional readout design methodology where the states have to be carefully selected in order to estimate the desired signal with a linear readout. In contrast, the LAM readout does not try to estimate a desired response but rather generates a template from the states.

Among the ESNs with MACE readout, the ASE-ESN performs the best compared to randomly generated ESNs. This was an expected result, since ASE-ESN distributes the echo states uniformly and increases separation between states corresponding to different classes. Therefore, ASE-ESN should be the preferred design for ESNs with MACE filter readout.

Robust signal detection in a digital communication channel

We applied ESN-MACE to detection of a known waveform transmitted through a digital communication channel. Please refer to Chapter 6 for details of this study.

Classification of odors with an electronic nose

In this experiment, the goal is to classify odors using signals collected with a Cyranose 320 electronic nose. The Cyranose 320 has an array of 32 sensors which change their resistance in response to odors and produce a unique response pattern for different odors (Figure 4-3). For our experiment, the spices, basil, cinnamon, ginger, and rosemary form the 4 classes of odors to be classified. Each of these spices is mixed with air and exposed to the electronic nose in 4 different concentrations; 25%, 50%, 75% and 100%. For each concentration of each spice, data corresponding to 10 different trials is collected. There are a total of 160 trials, 40 for each class of spice. We submit two approaches to solve the problem. The first approach is based on obtaining static features that are the steady-state values of all 32 sensors as given by the Cyranose. We obtain a 32x160 matrix containing 32 features for all 160 trials of the 4 classes. From the feature matrix, we extract the part corresponding to 40 trials (10 trials for each class with representatives from 4 different concentrations) as the training set to train a linear classifier with 4 outputs, each representing one of the 4 classes. The linear classifier is used due to the availability of an analytical solution. Each output of the linear classifier is trained to output 1 when the input feature is from the representative class and 0 otherwise. The 4x32 weight matrix of the linear network is computed using least squares. During testing, the 32x1 feature vector is

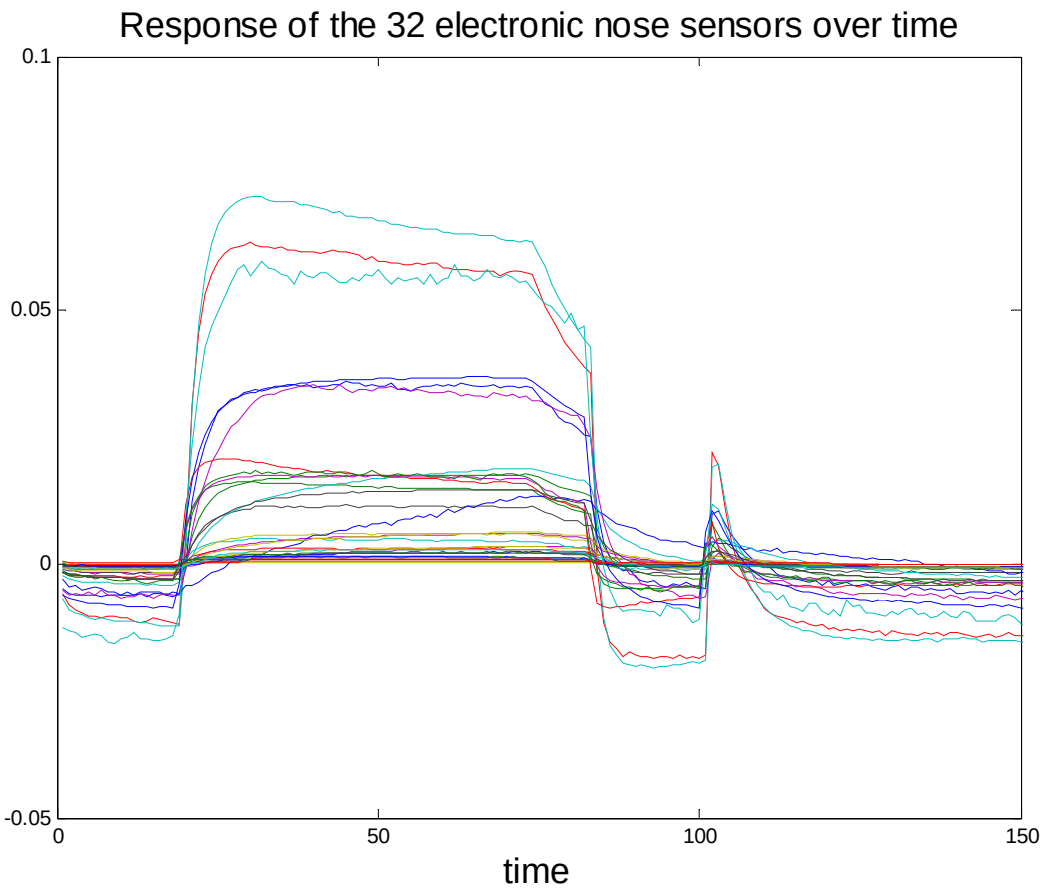


Figure 4-3. Time-series representing the response pattern of the 32 electronic nose sensors exposed to rosemary.

used to compute the 4 outputs and the label of the output with the maximum value is assigned as the class of the input vector. The classification accuracy of the linear classifier evaluated on the 160 feature vectors is found to be 78.7% and the confusion matrix is found to be

$$\begin{bmatrix} 40 & 0 & 0 & 0 \\ 4 & 22 & 14 & 0 \\ 0 & 0 & 40 & 0 \\ 0 & 0 & 16 & 24 \end{bmatrix}$$

The entry aij in row i and column j of the confusion matrix is the number of odors that should have been classified in class i but were classified as j. The correct decisions are given on

the main diagonal of the confusion matrix. The 4 classes are presented in the order basil, cinnamon, ginger, and rosemary.

The second approach is based on the proposed ESN-MACE. ESN exploits the dynamical response of the sensors which also contain information about the odor class. We create a template MACE filter for each class from the states of the ESN which represents the variations among different spice concentrations. Again, 40 trials (10 trials for each class with representatives from 4 different concentrations) are used as the training set. Unlike the static case, each training trial consists of a 32 dimensional time series representing the response of the electronic nose. We use an ESN with 100 internal units and 32 inputs. The internal weight matrix, **W**, of the ESN is designed with uniform eigenvalue distribution and a spectral radius of 0.9. The entries of $\mathbf{W}^{in}$ are set to 5 or -5 with equal probability. In the previous two experiments, the selection of the length of the patterns was dictated by the known signal lengths. However, in this experiment, we do not have pre-defined signals making the selection of the pattern length, *T*, a design parameter (Figure 4-3). Here, we choose *T* to be 80 so that the template pattern contains the rising and falling edges of the sensor response. This selection uses the transient sensor dynamics of the particular odor in the design of the MACE readout. For each input pattern, the echo states are calculated using Equation 2-2 with zero initial conditions. This creates an image of dimension 100x80, where 100 is the number of PEs and 80 is the signal length. 10 in-class patterns are used to train each one of the MACE filters corresponding to 4 classes. Output correlation peaks are assigned to be 1.0. During testing, echo states are calculated for each pattern and 4 MACE filter outputs are calculated. The label of the MACE filter with the highest value is assigned as the class of the input patter. The classification accuracy of the ESN-MACE evaluated on the 160 input patterns is found to be 63.1% and the confusion matrix is found to be

$$\begin{bmatrix} 21 & 18 & 1 & 0 \\ 5 & 35 & 0 & 0 \\ 7 & 16 & 17 & 0 \\ 0 & 1 & 11 & 28 \end{bmatrix}$$

The results are, indeed, worse than the linear classifier trained on static features. When looked closely, the MACE filter gives correlation values around 1.0 for the in-class data. However, the response of the MACE filter corresponding to 2nd class (cinnamon) for out-of-class data is bigger than 1 for most of the patterns which results in the false classifications. The MACE filter trained nondiscrimantly does not constrain the response of the MACE for out-of-class data. One solution to this problem is to use out-of-class data in the MACE filter training as is done in (Mahalanobis et al., 1987). Different from the previous experiments, both in-class and out-of-class training data are used to train each one of the 4 MACE filters corresponding to 4 classes. For each MACE filter, 10 in-class and 30 out-of-class (10 for each class) echo state images are used for *discriminative* training. Output correlation peak amplitudes are assigned to be 1.0 for the in-class and 0.1 for the out-of-class training data.

Using the training data, four MACE filters are synthesized in the frequency domain using Equation 4-1 and the corresponding image plane filters are obtained by inverse discrete Fourier transform. For a given test signal, the output of the each MACE filter is calculated by correlating the echo states of dimension 100x80 with the filter coefficients. The label of the filter with the largest correlation is assigned as the class of the input pattern. Out of the 160 trials, the correct classification rate is found to be 93.1 %. Note that this is much better than the performance of the classifier based on static features. Similarly, the confusion matrix is found to be

$$\begin{bmatrix} 38 & 2 & 0 & 0 \\ 0 & 37 & 3 & 0 \\ 0 & 5 & 35 & 0 \\ 0 & 0 & 1 & 39 \end{bmatrix}.$$

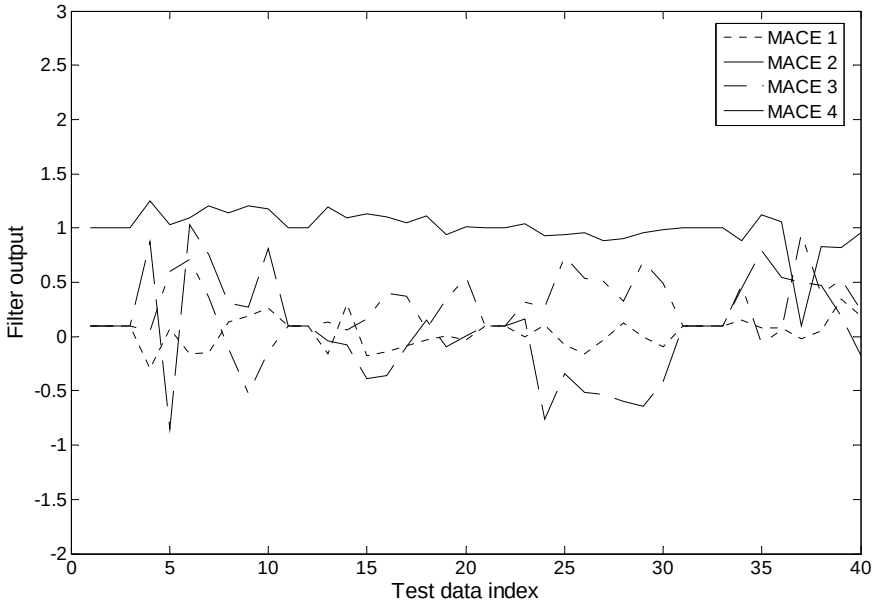
The addition of out-of-class data for training of the MACE improves the rejection of the out-of class patterns and hence decreases the false alarm rate. However, it also converts the problem to a classification problem (closed set classification) where the number of classes has to be known a priori. It is desirable, whenever possible, to avoid the use of out-of-class data in the training data for detection of signals where the out-of-class is either not available a priori or can be any possible signal shape excluding the in-class signal shape (the entire signal space excluding the in-class signal space).

Alternatively, we trained MACE filters in the input space instead of the state space of the ESN. In the input space, the images are of dimension 32x80. The procedure of training MACE filters for each class is repeated in the 32 dimensional input space. Out of 160 trials, the correct classification rate is found to be 85.6%. The confusion matrix of the MACE filter is

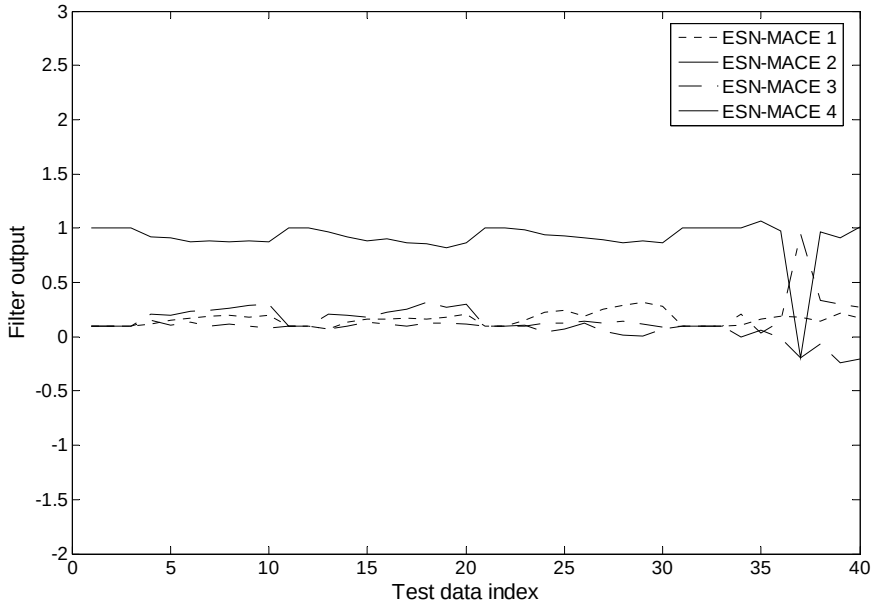
$$\begin{bmatrix} 34 & 5 & 1 & 0 \\ 3 & 32 & 5 & 0 \\ 1 & 7 & 32 & 0 \\ 0 & 0 & 1 & 39 \end{bmatrix}$$

In order to understand the increase in classification rate by the ESNs, let us compare the outputs of ESN-MACE filter with the MACE filter trained directly at the input space for the rosemary class (Figure 4-4).

As observed from the figures, the ESN-MACE filter output corresponding to the rosemary class displays values around one consistently for the in-class data except for the 37^{th} testing sample.



A



B

Figure 4-4. Comparison of the ESN-MACE filter output and MACE filter output trained on the input space for the rosemary class. A) Outputs of the 4 MACE filters over the test data index. B) Outputs of the 4 ESN-MACE filters over the test data index.

The ESN-MACE filter outputs corresponding to the other classes are very close to 0.1 as specified by the training procedure. However, the MACE filter outputs trained directly in the *input space* have very inconsistent outputs for the out of class samples. The margin between the in-class and out-of-class outputs of the MACE filter is increased by the ESN reservoir. There are mainly two reasons for the increase in the distance between the images with the addition of ESN. The first reason is the projection of the signals to a higher dimension by the ESN which increases the separation between the images. Secondly, ESN functions as a nonlinear filter that helps separate the responses to different input classes.

We also compared the results obtained to the TDNN trained in discriminative mode. 4 TDNNs, one for each class, are used where the desired target signal is +1 for in-class data and 0 otherwise. Each TDNN has a delay line of 81 samples (equal to the signal duration) for each one of the 32 sensors, 3 hidden layer PEs and a single output unit. The same data (10 in-class and 30 out-of-class) for discriminative ESN-MACE is used to train the weights of TDNN using the Levenberg-Marquardt algorithm with a step size of 0.01. Training is stopped after 100 epochs. A cross-validation set cannot be used due to limited number of training data. During testing, an input pattern is fed to each one of the 4 TDNNs and the label of the TDNN with the largest output is assigned as the class of the input pattern. Out of 160 trials, the correct classification rate is found to be 65.7%. The confusion matrix of the filter is

$$\begin{bmatrix} 25 & 13 & 2 & 0 \\ 3 & 29 & 7 & 1 \\ 3 & 10 & 21 & 6 \\ 6 & 2 & 2 & 30 \end{bmatrix}$$

The performance of the TDNN is very poor compared to discriminative ESN-MACE and even the static classifier. The main reason for the poor performace is the lack of sufficient data

for training the TDNN. The number of parameters of each TDNN is 32*81*3+3=7779 whereas only 40 training patterns are available for training.

Spike train classification with LSM

The associative memory readout presented for ESN can in fact also be utilized as readout for a LSM. Similar to the ESN case, the liquid states are interpreted as 2-D images, one dimension time and the other space. An optimal MACE filter, which associates LSM states with class labels, can be analytically computed for each class using training data. The value of the liquid state is either 1 or 0 depending on the existence or absence of a spike. Note that the MACE filter readout does not require low-pass filtering the spike trains in order to get continuous amplitude signals for further processing.

We use a spike train classification experiment, where the goal is to classify a spike train presented to the LSM as one of the two classes (Maass et al., 2002). Two classes are represented by randomly generated Poisson distributed spike trains over 100ms time interval. Other members of each class are generated by moving each spike of the representative spike train by an amount drawn from a zero mean Gaussian distribution with a variance of 4ms.

A randomly connected LSM with 50 integrate-and-fire neurons with 20% inhibitory (I) and 80% excitatory (E) connections is used. The membrane time constant, reset voltage, threshold voltage, background current and input resistance of the neurons are chosen to be 30ms, 13.5mV, 15mV, 13.5nA, and 1 MΩ, respectively. Absolute refractory periods for excitatory and inhibitory neurons are 3ms and 2ms, respectively (Maass et al., 2002). The probability of a synaptic connection between neurons *a* and *b* is given by $Ce^{-(\frac{D(a,b)}{\lambda})^2}$, where D(a,b) denotes the distance between two neurons and λ is a parameter which controls both the average number of connections and the average distance between neurons that are synaptically connected. The

neurons are placed on a 10x5x1 grid. *C* is set to 0.3 (*EE*), 0.2 (*EI*), 0.4 (*IE*), 0.1 (*II*). Dynamic synapses are used based on the model proposed in (Markram et al., 1998), with the synaptic parameters *U* (use), *D* (depression time constant), *F* (facilitation time constant) randomly chosen from Gaussian distributions. Mean values of *U*, *D*, *F* are set to .5, 1.1, .05 (*EE*), .05, .125, 1.2 (*EI*), .25, .7, .02 (*IE*), .32, .144, .06 (*II*) and SD of each parameter was chosen to be 50% of its mean. The mean of the scaling parameter *W* (in nA) is chosen from a Gamma distribution with mean values 30 nA (EE), 60 nA (EI), -19 nA (IE), -19 nA (II) (Maass et al., 2002). For input synapses, the parameter *W* has a value of 18 nA or 9.0 nA for destinations excitatory and inhibitory neuron, respectively. SD of *W* is 100% of its mean. The postsynaptic current was modeled as an exponential decay $\exp(-t/ts)$ where t*s* is 3ms and 6ms for excitatory and inhibitory synapses, respectively. The transmission delays between neurons are 1.5 ms (EE), and 0.8 for the other connections. The initial membrane voltage of each neuron at the start of each simulation is drawn from a uniform distribution between 13.5mV and 15.0mV (Maass et al., 2002).

Fifty different realizations of spike trains for each class are generated and the liquid states are calculated for each realization. One readout MACE filter is trained for each class using only data from the corresponding class. Output correlation peak amplitude is assigned to be 1.0 for the training data. Notice that the state matrix is very sparse resulting in singular matrices in the MACE filter computation. Therefore, the matrices have to be regularized in order to compute inverses. Here the regularization is done by adding zero mean Gaussian noise with a variance of 0.02 to the states before computing the MACE solution. The two MACE filters are synthesized in the frequency domain using eq 4.1 and the corresponding image plane filters are obtained by an inverse discrete Fourier transform. For a given test signal, the output of each filter is calculated by correlating the liquid states with the filter coefficients.

Spike train classification with LSMs is conventionally done by first low-pass filtering the spike output of the liquid and using a linear regressor network (Maass et al., 2002). Here, we used a low-pass filter with a time constant of 30ms. A linear readout is trained with the constant target signal of amplitude +1 for one of classes and the constant signal of amplitude -1 for the other class using the same data as MACE filter training. During testing, the echo states, the output signal and two error signals one between the output and the constant signal of amplitude +1 and another between the output and the constant signal of amplitude -1 are computed for a given input signal. Then, two MSE values, integrated over the duration of the input pattern, are compared to assign the class of the input pattern.

Figure 4-5 shows the correct classification rate for both methods as the parameter λ, which controls the synaptic connections in the LSM, varies from 0.2 to 4. The results are averaged over 100 trials. As observed from the figure, the MACE filter readout gives slightly better accuracy compared to the linear readout for all λ values. This is due to the fact that the linear readout tries to approximate a constant signal independent of the liquid state values. This creates difficulties for the simple linear readout especially when the input spike train is sparse. In fact, under closer analysis, the output of the linear readout fluctuates. Therefore, the decision made by the linear readout has to be carefully averaged over time in order to reach a final decision about the class of the input spike train. For example, if the decision is based solely on the output of the readout at the final time instant, the classification rate is very close to theoretical minimum of 0.5. On the other hand, the MACE filter readout provides the ability to operate in the spike domain without the need to low-pass filter the liquid state spike output. However, we note that the MACE filter computation in Equation 4-1 requires extra caution since the liquid state has many zero values mostly leading to singular matrices.

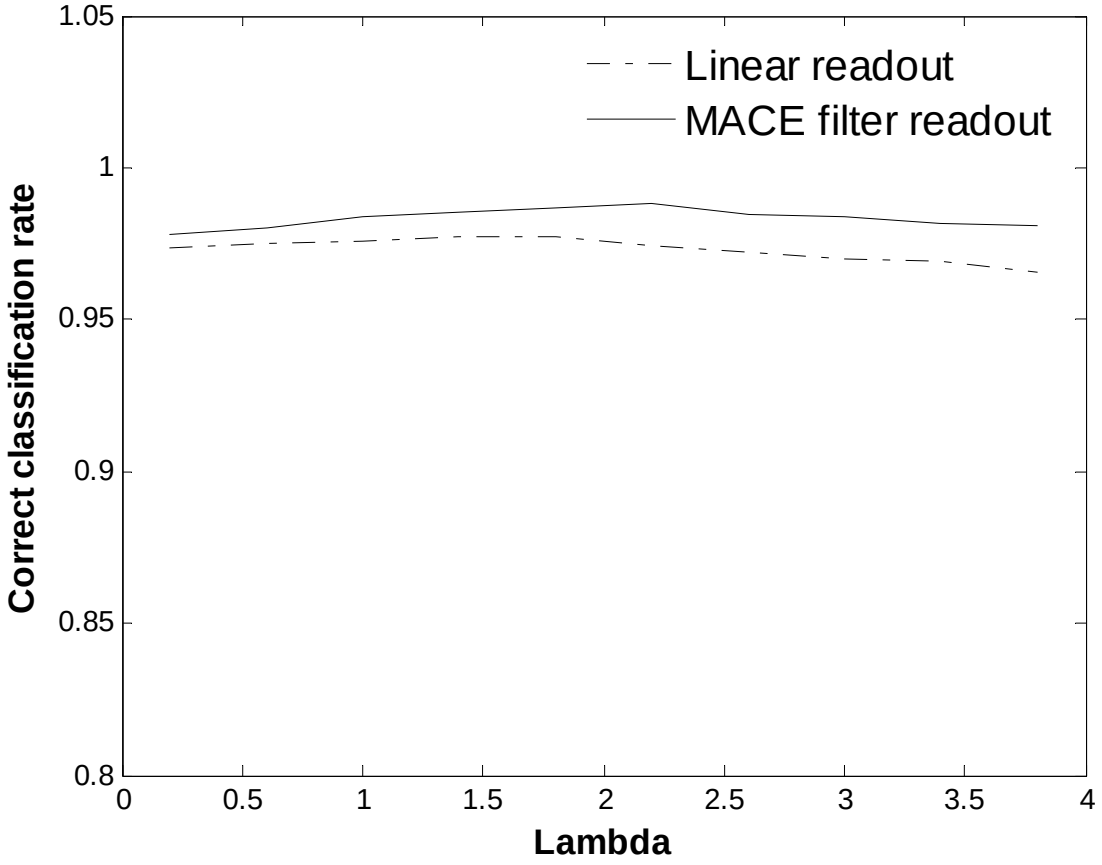


Figure 4-5. Comparison of the correct classification rates of LSM-MACE and LSM with linear readout trained in the spike train classification problem as the parameter λ varies. The MACE filter readout gives slightly better accuracy compared to the linear readout for all λ values. This is due to the fact that the linear readout tries to approximate a constant signal independent of the liquid state values. Moreover, MACE readout operates in the spike domain eliminating the need to convert spike trains to continuous valued signals by low-pass filtering.

This experiment demonstrates the use of MACE filter as a readout for LSMs. A more detailed study has to be carried out to explore the advantages and disadvantages of MACE readout compared to the standard techniques.

CHAPTER 5
IMPLICATIONS OF ESN ON THE DESIGN OF OTHER NETWORKS

Echo state networks introduced a new type of computation that has been proven to be very powerful. In this chapter, we study how the ESN idea can be utilized for the design of other networks. First, we investigate the echo state condition ($|\mathbf{W}|<1$) and approach the echo state condition in terms of the effect of system stability on the power of dynamical system for computation. In particular, we relax the echo state condition and allow some of the poles to be larger than 1. This introduces a new dynamical regime, called "transiently stable computation", where function approximation with a readout is still possible eventhough the system is not globally stable. Second, we investigate the biologically plausible model of the olfactory cortex, Freeman Model (FM), and propose to use a readout for FM to be able to use it for useful computation. Without the proposed readout, the use of FM is limited to simple digit recognition. An interesting property of FM is the nonlinear function used in the model. FM nonlinearity does not have its largest slope at zero operating point unlike the sigmoidal nonlinearity used in ESNs. We will demonstrate with experiments that FM coupled with a readout can process continuous valued signals.

Transiently Stable Computation

Linear filters are the simplest dynamical systems employed to process signals with temporal structure as discussed in chapter 1. They can be classified according to the impulse response, which fully characterizes the filter, as finite impulse response (FIR) and infinite impulse response (IIR) filters (Haykin, 2001). For the IIR case, stability becomes an essential constraint in the design. Without stability, the system response will diverge independent of the input signal. Global asymptotic stability of linear filters can simply be guaranteed by selecting the poles of the system in the proper region of the frequency domain, open left half plane for the

continuous-time systems and inside the unit circle for the discrete-time systems (Kailath, 1980). TDNN and RNN can be considered as the nonlinear counterparts of the FIR and IIR filters, respectively. Similar to the linear case, stability becomes an issue for RNNs, although the definition of stability is no longer BIBO (bounded input bounded output) stability. In the ESNs, stability issue is addressed with the echo state condition (Jaeger, 2001) and has been investigated in detail in chapter 2. Echo state condition is derived from the stability of the linearization of the system around zero equilibrium and guarantees that the states are strongly coupled with the input signal allowing the system to compute an input-output map. In this chapter, we introduce a new computational mode that emerges when we relax the echo state condition and allow the spectral radius to be slightly greater than 1. First, let us consider an ESN that obeys the echo state condition.

Conventional Echo State Networks

A simple demonstration of the typical response of an ESN with echo state condition is given in Figure 5-1. A randomly connected 100-unit ESN without output feedback connections is constructed. The entries of the internal connection weight matrix **W** are set to 0.4, -0.4 and 0 with probabilities of 0.025, 0.025 and 0.95 resulting in a spectral radius of 0.9. The input weight matrix, $\mathbf{W}^{in}$ has values +1 or –1 with equal probabilities. The input to the system (Figure 5-1 A) has 3 different regions, a zero signal for the first 100 steps, a ramp signal for the following 100 steps and again a zero signal of length 100. The system is initialized randomly and run with the given input.

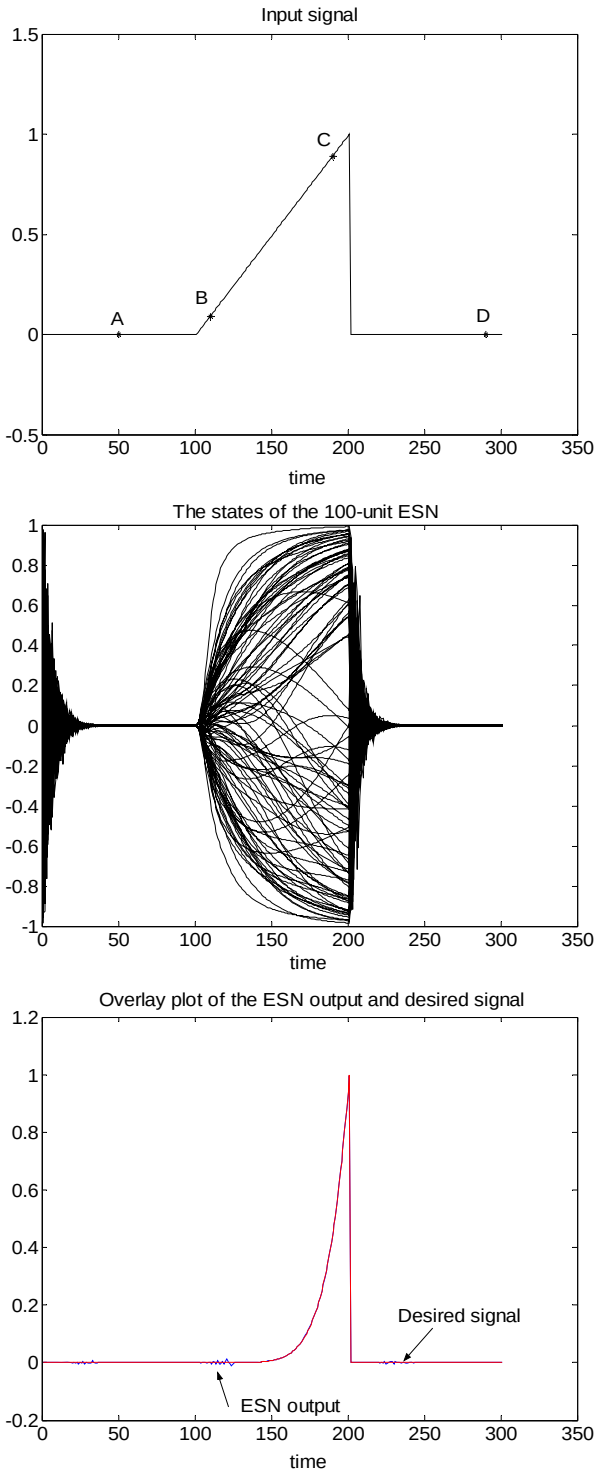
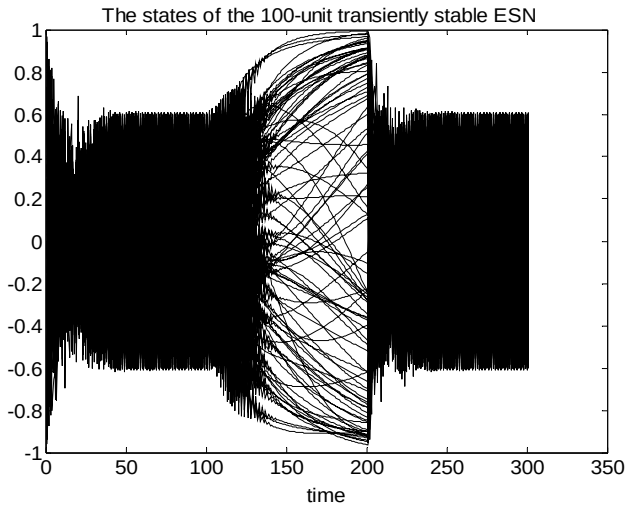


Figure 5-1. Demonstration of a typical response of ESN with a spectral radius of 0.9 and application to function approximation. A) The input signal. B) 100 echo states of ESN. C) The overlay plot of the ESN output and the desired signal.

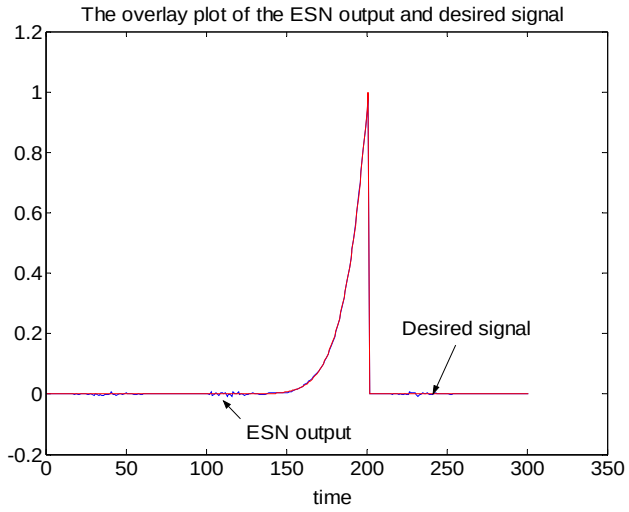
Figure 5-1 B depicts all the 100 echo states on top of each other over the given time interval. As seen from the figure, during the first 100 steps, the system state converged to zero (after an initial transient of length about 30 steps caused by the random initial condition) since the input to the system is zero and the spectral radius is less than 1. Then, the echo states are constructed by the ramp signal between the time interval 100 and 200. The states again converge to zero due to the echo state condition, when the input is removed. The resulting echo states can be used to generate any desired response related to the input signal. Here, desired signal is chosen to be the seventh power of the input signal and The readout weights are calculated using Equation 2-4. Figure 5-1 C displays the overlay plot of the optimal ESN output and the desired signal. As observed from the figure, the system output very accurately estimates the desired signal.

Transiently Stable Computation with Echo State Networks

Engineering systems are mostly designed with a global stability restriction. We have seen a similar constraint, echo state condition, in the design of echo state networks. We have also demonstrated with a function approximation experiment that ESNs with the echo state condition can do useful computation. On the other hand, biological computation may not possess a similar stability constraint (Freeman, 1975; Freeman, 1993; Yao and Freeman, 1990). Indeed, biological systems do not have fixed point dynamics but rather wide variety of collective behavior in the form of oscillations and even chaos (Freeman, 1975; Yao and Freeman, 1990). Inspiring from principles of biological computation, we would like to explore the network's response without the restriction on the spectral radius of **W**. Hence, we will remove the echo state condition and let the spectral radius to be slightly greater than 1. Obviously, this will introduce instability for the autonomous system whereas the response of the system with an applied input is yet not obvious.



A



B

Figure 5-2. Demonstration of a typical response of transiently stable ESN with a spectral radius of 1.1 and application to function approximation. A) 100 echo states of the transiently stable ESN. B) The overlay plot of the ESN output and the desired signal.

In order to observe the ESN response without the echo state condition, we will use a similar experimental set up that we used in previous section. We use the same **W** matrix and scaled it to obtain a spectral radius of 1.1. The same input signal (Figure 5-1 A) is fed to the ESN

that is initialized to a random initial condition. We plot the resulting echo states in Figure 5-2 A. There are a few observations to be made at this point. First of all, we see that the system does not converge to zero during the first 100 steps although the input is zero. Instead, the echo states exhibit a nonconvergent dynamical response that differs from the ESN with the echo state condition. In fact, this was the expected response, since the system is designed to be unstable (spectral radius is greater than one) around zero equilibrium. A similar response can also be observed during the last 100 time steps when the input is again zero. The second and more important observation is the response of the system between time steps 100 and 200 where the ramp signal is applied as the input. The echo states during this interval become more regular (after some initial transient between time steps 100 and 130) compared to the states when there is no input. In fact, after the transient, the echo states look similar to the ones in Figure 5-1 B where the ESN satisfies the echo state condition. Thirdly, there is a transition period between time steps 100 and 130 where the states switch from a disordered mode to a more regular mode. In summary, the system responds according to its own dynamics when there is no input. As the input amplitude gradually increases, the system response is determined by a competition between the system dynamics and the input amplitude. When the input amplitude is sufficiently large, the system dynamics becomes "transiently stable" and is determined by the input signal. We would like to find out if we can utilize the transiently stable ESN for the same function approximation problem we had in the previous section. The weights of the linear readout network are again computed using Equation 2-4 and the corresponding output is generated in Figure 5-2 B. As seen from the figure, we got a good match between the output and the desired signal. Similar results can be obtained even if the system is initialized to a different initial condition or the time instant at which the ramp signal is applied (this will change the state value when the ramp signal is

applied) changes. This experiment clearly demonstrates that "transiently stable" ESN can do useful computation.

Understanding Transiently Stable Computation

We will utilize the linearization analysis proposed in chapter 2.2 in order to quantify the *local dynamics* of the system. The pole movement of ESN with spectral radius of 0.9 is illustrated in Figure 5-3. According to the figure, poles always stay inside the unit circle even though the input signal changes. As the input signal strengthens, the poles move towards the origin of the z-plane (decreases the spectral radius), which results in a more stable system. This is due to the fact that the higher input signals saturate the nonlinear function reducing the slope at which the system is operating. Similar observations have been already made in chapter 2.2.

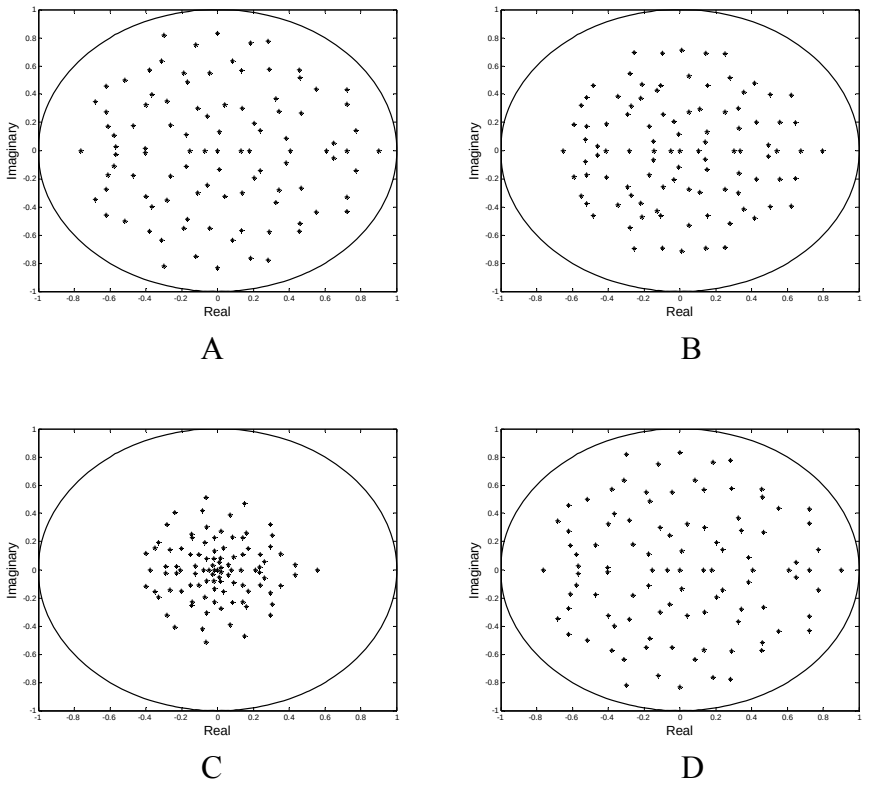


Figure 5-3. Movement of the poles for ESN with spectral radius 0.9 are plotted in figures A, B, C, D , when the input is at points labeled by A, B, C, D in Figure 5-1 A, respectively.

Now, let us examine the movement of poles of the transiently stable ESNof the previous section. Figure 5-4 shows the movement of the poles for the same input signal. First of all, the poles of the system move in and out of the unit circle during the first and last 100 time steps (hence zero state is not asymptotically stable). This explains the complex echo state signal when there is no input to the system (Figure 5-2 B). The poles of the ESN with spectral radius 0.9 stays inside the unit circle during the same time interval. Secondly, when the ramp input is

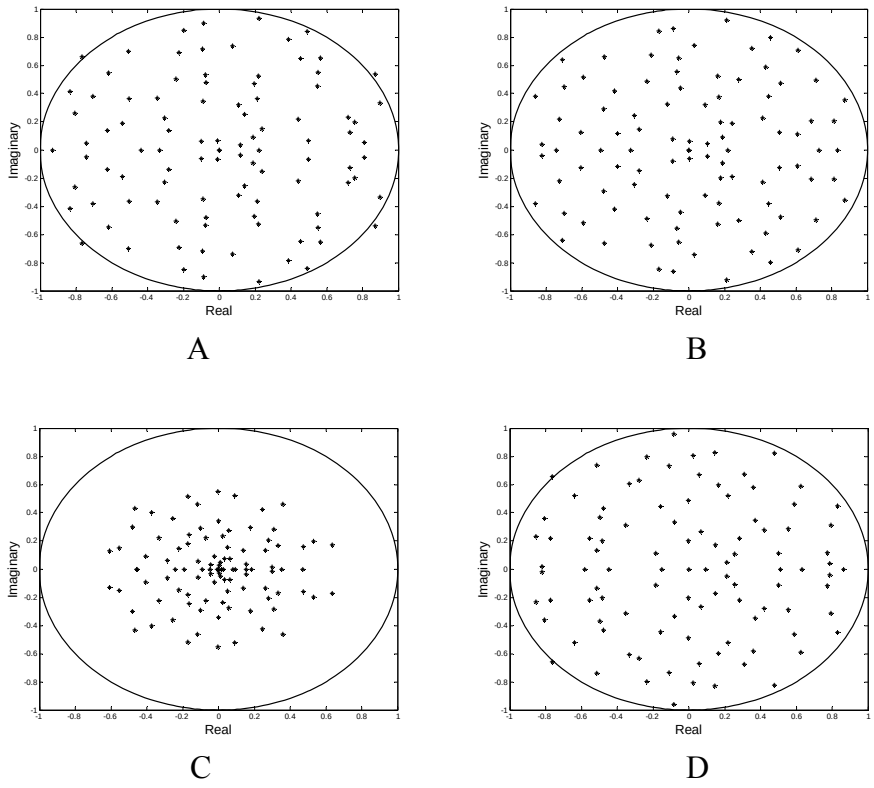


Figure 5-4. Movement of the poles for ESN with spectral radius 1.1 are plotted in figures A, B, C, D, when the input is at points labeled by A, B, C, D in Figure 5-1 A, respectively.

applied, the poles start to shrink towards the unit circle (after some transient) as is observed in the stable ESN. This phenomenon is again due to the sigmoid nature of nonlinearity, which saturates with large input signals reducing the slope of the operating point in Equation 2-5. With the movement of the poles towards the origin, the system stabilized transiently allowing the system state to be controlled by the input. This is critical for the use of the system for

computation (function approximation). Finally, we observe a transient time, when the ramp signal is applied, before the poles start shrinking. In this period, there is a competition between the system dynamics and the input signal. There are a few extra comments about the selection of the spectral radius to be made at this point. First of all, the transient response time after the ramp is applied is determined by two factors, the spectral radius and the slope of the ramp signal. The transient time will increase if the spectral radius of **W** is increased or the slope of the ramp signal is reduced. Because as the spectral radius is increased, the force required to bring the poles inside the unit circle is increased, hence a larger signal is required. Therefore, there is a balance between the system dynamics determined by the spectral radius and the input signal required to stabilize the system. Secondly, it is important to quantify the unstable behavior of the signals present in the transiently stable ESN when the input is absent. We have observed that when the spectral radius is slightly greater than 1, the signals are periodic oscillations whereas for larger values of the spectral radius, the signals are more irregular and even chaotic.

Freeman's K Sets for Dynamical Computation

Understanding information processing in brains remains a challenge. Many different hypotheses on how the brain might process the massive bombardment of information brought by sensory systems have been advanced, ranging from artificial intelligence, neural networks and more recently machine learning and statistics (Freeman, 1975; Yao and Freeman, 1990; Chua and Yang, 1988; Wang, 1999).Walter Freeman developed a biologically realistic mathematical model for the olfactory cortex that captures at a cell assembly level (mesoscopic level), some of the physiological properties of the olfactory system. Freeman model (FM) is based on nonlinear recurrent dynamical interactions representing the response of thousands of cells, the neural assemblies, instead of single neuron responses. In order to derive his model, Freeman used neurophysiological recordings and neuroscience concepts and the K set hierarchy (Freeman,

1975) to build a model that mimics the behavior of olfaction (Freeman, 1975). This model is a drastic departure of other neural network models such as Hopfield's because system response is produced by interactions of second order nonlinear sets.

Freeman's major drive is to understand how olfactory system works; while here we aim at using FM as a computational framework for the interaction of animats with the real world. Real worlds are complex dynamical environments that unfold unpredictably with lots of redundancy but also with many unrelated cues. In this set up, the fundamental operations are the recognition of prior situations that will induce prototypical responses dependent upon the animat's goals. The constraints of the experimental setting are also rather important. The operation is intrinsically real-time in the sense that only the present value of the continuous stream of information from the sensors is available and must be processed at each time step, while the full understanding of the situation is most often related to the input history. So we favor signal processing methodologies instead of statistics where the behavior in time is always rather constrained. The animat may also need to learn how to recognize new situations when they become important for its function, therefore it must have internal representational structures that expect outcomes and change behaviors. Brains process information for the sake of the animal's survival, while animats have external goals preprogrammed by humans, and as such they must possess an extra level of functionality that translates internal representations back and forth to the outside, i.e. it is necessary to define input and outputs (in the animal, brain's only output is the motor system). Therefore we have to emphasize this translation between distributed dynamical representations and outputs in the form of design principles for optimal readouts.

The framework we are pursuing is based on distributed dynamics instead of machine learning. Since Hopfield, the role of dynamics for associative memories is well understood

(dynamical systems with fixed point attractors). In fact, FM has also been utilized as a dynamical auto-associative memory (Tavares, 2001; Xu et al., Ozturk et al., 2004; Xu et al., 2004). FM weights are trained with Hebbian learning while two binary patterns to be stored are presented to the system successively. During testing, a noise corrupted version of one of the two stored patterns is presented to the system. Then, the energy of the oscillatory system response is computed and compared to a threshold to choose the correct stored pattern. We see two major shortcomings with the energy based readout. First, utilizing just the energy is wasteful since it ignores the information embedded in the dynamics of the signals. Second, it does not exploit the global information in the states since it is a local method computing the energy of individual processing elements.

In this chapter, we demonstrate that FM can be considered in the same framework of ESN/LSM. In fact, the KI and KII networks of FM are conceptually similar to the reservoir/liquid in an ESN/LSM with a proper selection of the parameters. The big difference is that the KII layer is highly structured with a coupled set of nonlinear oscillators We will derive conditions on the system parameters for the FM similar to the echo state property of ESNs. With the proposed framework, it becomes evident that the OB layer alone lacks a readout in order to be used as a universal computing machine for time series. Therefore, we propose to use an adaptive linear network to implement the readout from the OB layer. We present experimental results to show the power of this framework for the FM of the olfactory cortex. In particular, we show that it is not necessary to drive Freeman's model with 0 and 1s, as is presently done.

An Overview of Freeman Model

Freeman Model is a biologically plausible mathematical model of the olfactory cortex. The dynamical behavior of the FM mimics the physiological signals obtained from the cortex. It is a

hierarchical model of different levels (*K0*, *KI*, *KII* and *KIII*), where simpler structures at the bottom of the hierarchy combine to form the more complex structures at the top of the hierarchy.

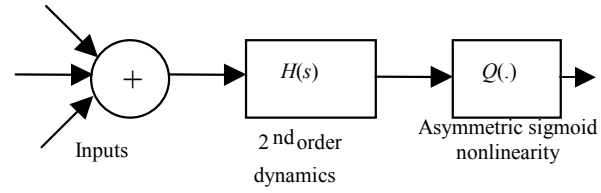
The basic building block of the model is the *K0* set, which is depicted in Figure 5-5 A. Every *K0* unit can accept several spatial inputs that are weighted and summed, and then convolved with a linear time invariant system defined by the second order dynamics whose transfer function, *H(s)*, is given by

$$H(s) = \frac{ab}{(s+a)(s+b)} \tag{5-1}$$

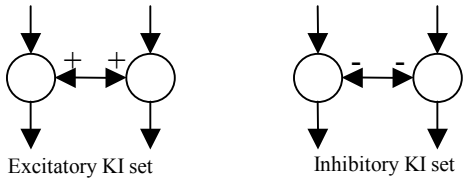
where 1/a and 1/b are real time constants determined experimentally (Freeman, 1975). The output of the linear dynamics is then shaped by the nonlinear sigmoid function, which is experimentally determined to be

$$Q(x, Q_m) = \begin{cases} Q_m(1 - e^{(-\frac{(e^x - 1)}{Q_m})}) & \text{if } x \geq x_0 \\ -1 & \text{if } x < x_0 \end{cases} \tag{5-2}$$

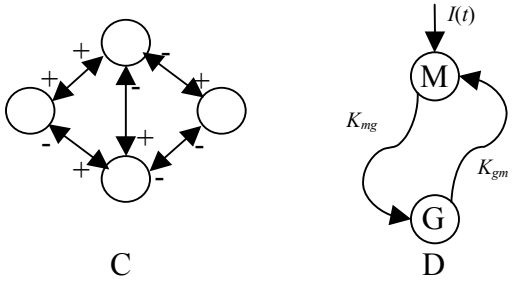
The sign of the connection strength from a *K0* set will define the type of the *K0* set (excitatory for positive weights and inhibitory for negative weights). The second level in the hierarchy is the *KI* network. K0 sets with a common sign (either excitatory or inhibitory) are connected through forward lateral feedback to construct a KI network (Figure 5-5 B). No auto-feedback is allowed in the network. KI network is generally used as an input layer for the higher levels of Freeman model. It preprocesses the input to transform it into a space which is compatible with OB dynamics. In this study, we will use KI network as an echo state network containing a representation of the input history.



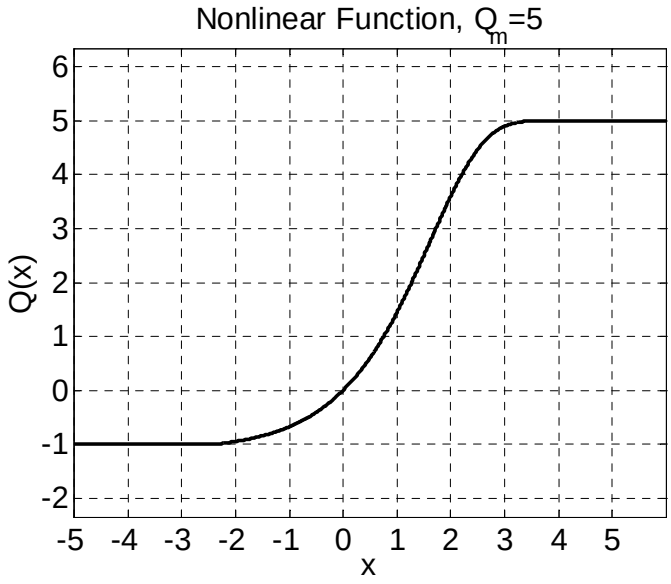
A



B



C D



E

Figure 5-5. The building blocks of Freeman Model. A) The K0 set. B) KI network. C) KII Set. D) Reduced KII set. E) The asymmetric nonlinear function.

The third level in the hierarchy, the *KII* set (Figure 5-5 C), is the most interesting and important building block of the olfactory system, since it is an oscillator controlled by the input. The response of the *KII* set to an impulse input is a damped oscillation whereas with a sustained input, the output oscillation is maintained as long as the input remains. The architecture of *KII* set is shown in Figure 5-5 C, where the circles denote *K0* sets and the sign indicates the type of connection. Figure 5-5 D shows a reduced *KII* set, where the two *K0* units are denoted by M (for mitral cell) and G (for granular cell). In this model, the mitral cell takes the external input P(t) and the coupling strengths between M and G are controlled by the two weights $K_{mg} > 0$ (excitatory) and $K_{gm} < 0$ (inhibitory). If the coupling weights are selected properly, the RKII set is, similar to the KII set, an oscillator controlled by the input (Freeman, 1975).

A *KII* network can be formed by interconnecting a number of *KII* sets with excitatory connections between the excitatory cells (top circles in Figure 5-6) and inhibitory connections between inhibitory cells (bottom circles in Figure 5-6). An RKII network is formed similar to KII network by connecting a number of RKII sets. We will use RKII network in our simulations instead of KII networks since various analytic tools are available for RKII and it has been demosntrated that RKII is functionally very similar to the KII (Freeman 1975, Xu et al., 2004). This interconnected structure represents a key stage of learning and memory in the olfactory system. Input patterns through M cells are mapped into spatially distributed outputs. Excitatory and inhibitory interconnections enable cooperative and competitive behaviors, respectively, in this network. The RKII network functions as an associative memory (Tavares, 2001; Xu et al., 2004; Principe et al., 2001; Ozturk et al., 2004).

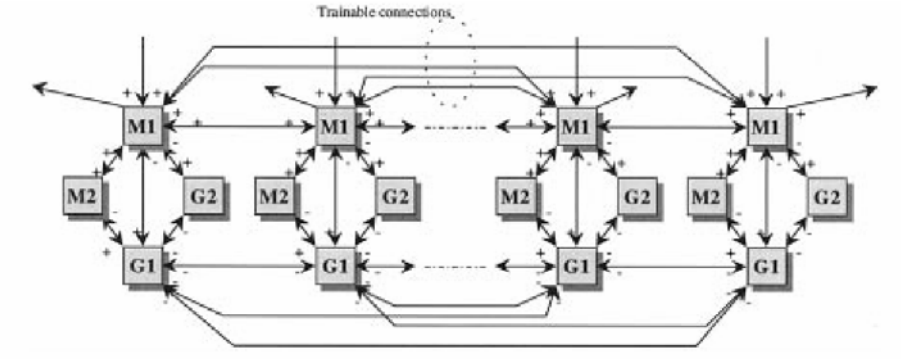


Figure 5-6. A full KII network.

The final Katchalsky level is the *KIII* network and represents the olfactory cortex. In a *KIII* network, two/three *KII* sets and a *KII* network are tightly coupled through dispersive connections, mimicking the different lengths and thicknesses of axons. Since the intrinsic oscillating frequencies of each one of the *KII* sets in different layers are incommensurate among themselves, this network of coupled oscillators will present chaotic behavior. For a detailed description of the *KIII* network, we refer the reader to (Freeman, 1975; Xu et al, 2004).

Now, we will formulate a standard form for the state (outputs of KO sets) equations of the Freeman model valid for all hierarchical sets of FM). Consider a Freeman network with M input units, N *KO* sets and L output units. State equations for this Freeman model can be expressed as

$$\ddot{\mathbf{x}}(t) + (a+b)\cdot\dot{\mathbf{x}}(t) + ab\cdot\mathbf{x}(t) = ab(W\cdot Q(\mathbf{x}(t)) + W^{in}\mathbf{u}(t)) \tag{5-3}$$

Here, $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))^T$ is the state vector where each entry is the output of a *KO set* at time *t* and $\mathbf{u}(t) = (u_1(t), u_2(t), \ldots, u_M(t))^T$ is the input vector. $\boldsymbol{W}$ is an N*N matrix defining the interconnection weight values between the *KO sets* and $\boldsymbol{W^{in}}$ is an N*M matrix defining the interconnection weights between the inputs and the KO sets.

For example, consider the RKII set depicted in Figure 5-5 D. The governing equations for the RKII set are given by

$$\ddot{m}(t) + (a+b)\dot{m}(t) + ab \cdot m(t) = ab(K_{gm}Q(g(t)) + p(t))$$
$$\ddot{g}(t) + (a+b)\dot{g}(t) + ab \cdot g(t) = abK_{mg}Q(m(t)) \quad (5\text{-}4)$$

In a RKII set, the number of *K0* sets is two and the number of inputs is one. Hence, in the standard form, M is one and N is two. With this information, Equation 5-4 can be restated in the standard form of Equation 5-4 with the state vector $\mathbf{x}(t) = [m(t), g(t)]^T$, input vector $u(t) = [p(t)]$, the input weight vector $W^{in} = [1, 0]^T$ and the weight matrix

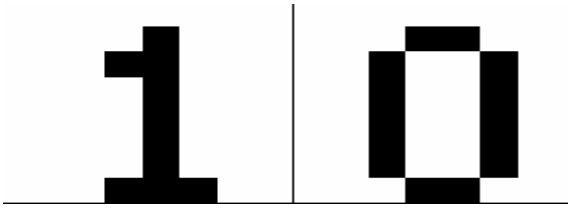
$$\mathbf{W} = \begin{bmatrix} 0 & K_{gm} \\ K_{mg} & 0 \end{bmatrix}$$

Dynamical Computation with Freeman Model with a Readout

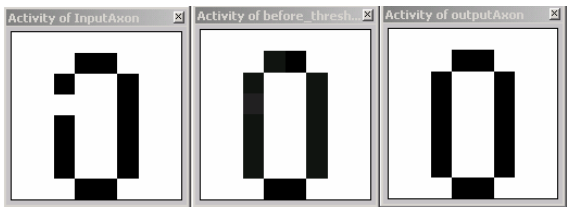
In order to be able to compute with a dynamical system, one has to be aware of how the information is represented in the system states. Once the form of representation is known, desired information can be extracted from the system states with an appropriate readout mechanism. For instance, in Deliang Wang's network (Wang, 1999), the phase difference between the coupled oscillators is the source of information. For FM, it has been argued the information is encoded as amplitude modulated signals and hence energy of the KO output was used as the relevant information from the FM in practical application. For example, the RKII network has been proposed as an auto-associative memory that can store static binary patterns (Tavares, 2001; Xu et al., 2004; Ozturk et al., 2004). In (Xu et al., 2004) an RKII network of 64 reduced KII sets is constructed with fixed K_{mg}, K_{gm} and K_{ii} values (determined experimentally from biological data). Excitatory weights of RKII network are trained with Oja's unsupervised learning rule (a stable version of the Hebbian learning rule) while two binary patterns to be stored are presented to the system successively. During testing, a noise corrupted version of one of the two stored patterns is presented to the system. Each RKII set creates limit cycle oscillations of low or high energy depending on the stored and the input pattern applied. Then,

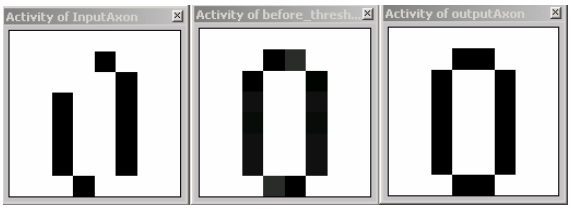
the energy (averaged over a time interval) of the mitral KO set of each RKII set is computed and compared to a threshold to decide the binary system output. We see two major shortcomings with the energy based readout for FM. First, utilizing just the energy is wasteful since it ignores the information embedded in the dynamics of the system state. Second, it does not exploit the global interdependencies among the RKII sets since it is a local method computing the energy of individual processing elements. Hence, the energy based readout can not be optimal and can not be used for continuous dynamic patterns.

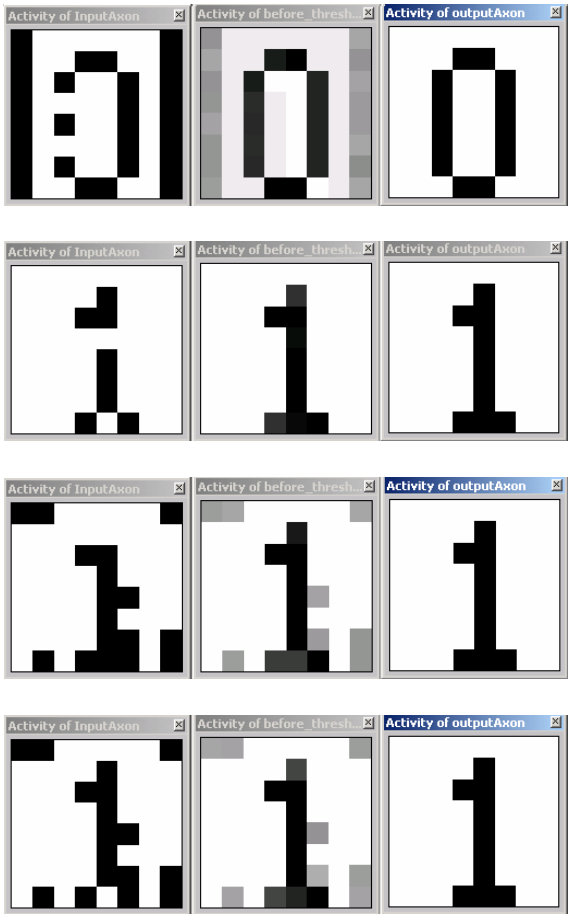
It is clear that a more principled approach is necessary in order to use FM to process signals that are not necessarily binary or static. With the analogy to the LSM/ESN framework, we propose to use a readout which is a linear or nonlinear projection of all the RKII states. The role of the readout is to extract specified desired information from the FM states.



A







B

Figure 5-7. Freeman Model as an associative memory. A) Stored patterns. B) Recall of the corrupted patterns: The first, second and their columns are the input to the KII network, the output before the threshold detector, and the output after the threshold detector, respectively.

We will also demonstrate that Freeman's K sets provides a complete system, that is to say, the readout can also be chosen from the Freeman hierarchy, namely K0 set or the KI network depending on the output layer dimension. When a linear readout is used, the adaptation of the overall system reduces to a linear regression problem, which we can be solved by either Wiener filter equation offline or gradient-based learning rule online. We will use the KI network and KII network architectures analogous to liquid in LSM or the reservoir in ESN. The use of KIII network as the dynamical reservoir is outside the scope of this study. In summary, with the

adoption of LSM/ESN framework, the states of the FM are dynamical basis functionals (representation space) created from nonlinear combinations of the input and the readout simply finds the best projection of the desired response in this representation space.

We will provide two theorems for the echo state property of the Freeman RKII network. The first theorem states the sufficient conditions on the parameter values of a RKII set resulting in echo states. In the second theorem, we will give a sufficient condition for the nonexistence of echo states for all Freeman networks. Unfortunately, we do not have a general theoretical result for the sufficient conditions resulting in echo states for all Freeman networks in standard form given by Equation 5-3.

Theorem 4.1: Consider a reduced KII set with the governing equations given by equation 5-4. Let the parameter values K_{mg} and K_{gm} satisfy $\left|K_{mg}K_{gm}\right| < \frac{ab}{(a+b)^2(\dot{Q}_{\max})^2}$, where $\dot{Q}_{\max}$ denotes the maximum value of the derivative of the nonlinear function Q and a and b are the filter coefficients. Then the network has echo states for all inputs **u** and for all states **x**. The proof follows from the linear stability analysis of the RKII set, which has been extensively studied in (Xu and Principe., 2005).

Proof: Consider the governing equations for the RKII set given in Equation 5-4. An equilibrium point of the nonlinear differential Equation 5-4 satisfies

$$\begin{aligned} m_0 &= K_{gm}Q(g_0) + p \\ g_0 &= K_{mg}Q(m_0) \end{aligned} \tag{5-5}$$

Using the nullclines defined by Equation 5-5, we can see that for a given set of parameters and the input, the system has a unique equilibrium point (Figure 5-8).

Now consider the state space representation given by Equation 5-4.

$$
\begin{aligned}
m_1 &= m_2 \\
m_2 &= -(a+b)m_2 - abm_1 + ab(K_{gm}Q(g_1) + p(t)) \\
g_1 &= g_2 \\
g_2 &= -(a+b)g_2 - abg_1 + abK_{mg}Q(m_1)
\end{aligned}
\tag{5-6}
$$

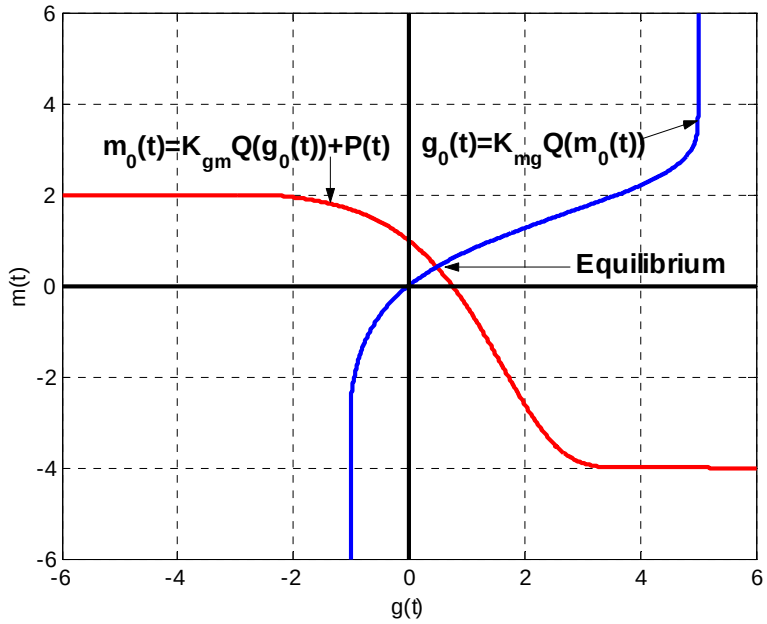


Figure 5-8. Nullcline graph of the reduced KII set.

Then, the linearization of the system around the unique equilibrium point leads to the Jacobian matrix of the form

$$
\begin{bmatrix}
0 & 1 & 0 & 0 \\
-ab & -(a+b) & abK_{mg}\dot{Q}(g_0) & 0 \\
0 & 0 & 0 & 1 \\
abKmg\dot{Q}(m_0) & 0 & -ab & -(a+b)
\end{bmatrix}
\tag{5-7}
$$

Checking the eigenvalues of Equation 5-9, we see that the equilibrium is stable if $\left|K_{mg}K_{gm}\right| < \dfrac{ab}{(a+b)^2\dot{Q}(m_0)\dot{Q}(g_0)}$ and the equilibrium is unstable if

$\left|K_{mg}K_{gm}\right| > \frac{ab}{(a+b)^2\dot{Q}(m_0)\dot{Q}(g_0)}$. The system does have a bifurcation point if

$\left|K_{mg}K_{gm}\right| = \frac{ab}{(a+b)^2\dot{Q}(m_0)\dot{Q}(g_0)}$ (Xu and Principe., 2005).

Now, assume that the parameters K_{mg} and K_{gm} satisfy $\left|K_{mg}K_{gm}\right| < \frac{ab}{(a+b)^2(\dot{Q}_{\max})^2}$. Then the unique equilibrium point of the system will be stable. This means that independent of the initial conditions, the system will approach to the same trajectory for a given set of parameter values and the input sequence. Moreover, the states will converge to zero when there is no input.

Theorem 4.2: Consider a Freeman network containing N *KO* sets with standard form given in Equation 5-3. Define the *effective weight matrix* $\overline{\mathbf{W}}$

$$\overline{\mathbf{W}} = \begin{bmatrix} 0 & I \\ ab(I-W) & -(a+b)I \end{bmatrix} \tag{5-8}$$

where I denotes the N*N identity matrix and 0 denotes N*N zero matrix. Let the effective weight matrix $\overline{\mathbf{W}}$ satisfy real(λ_{max})>0, where λ_{max} is an eigenvalue of $\overline{\mathbf{W}}$ with largest real part. Then the network has an asymptotically unstable null state. This implies that it has no echo states for any input set containing **0**.

Proof: Consider a Freeman network with governing equations in standard form given by Equation 5-4.

In state-space form Equation 5-3 can be expressed as

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \mathbf{x}_2 &= -(a+b)\mathbf{x}_2 - ab\mathbf{x}_1 - ab(\mathbf{W}Q(\mathbf{x}_1) + \mathbf{W}_{\mathbf{in}}\mathbf{u})) \end{aligned} \tag{5-9}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are both N dimensional vectors. For zero input, Equation 5-9 has the trivial null solution. Now, consider the Jacobian matrix of Equation 5-9 from linearization around origin,

$$\overline{\mathbf{W}} = \begin{bmatrix} 0 & I \\ -ab(I+W) & -(a+b)I \end{bmatrix} \tag{5-10}$$

If the matrix $\overline{\mathbf{W}}$ has an eigenvalue with real part greater than zero, the linearized system is not asymptotically stable. This implies the existence of another solution of Equation 5-9 besides the null solution. This implies that the network has no echo states for any input set containing **0.**

This theorem, which is valid for all Freeman networks in standard form, will provide a heuristic way to obtain a useful weight matrix **W** in practical applications. Whenever we create an interconnection weight matrix **W**, we will check the eigenvalues of the effective weight matrix $\overline{\mathbf{W}}$. If the real parts of all eigenvalues of $\overline{\mathbf{W}}$ are less than zero, we will assume the states are controlled by the input.

Experiments

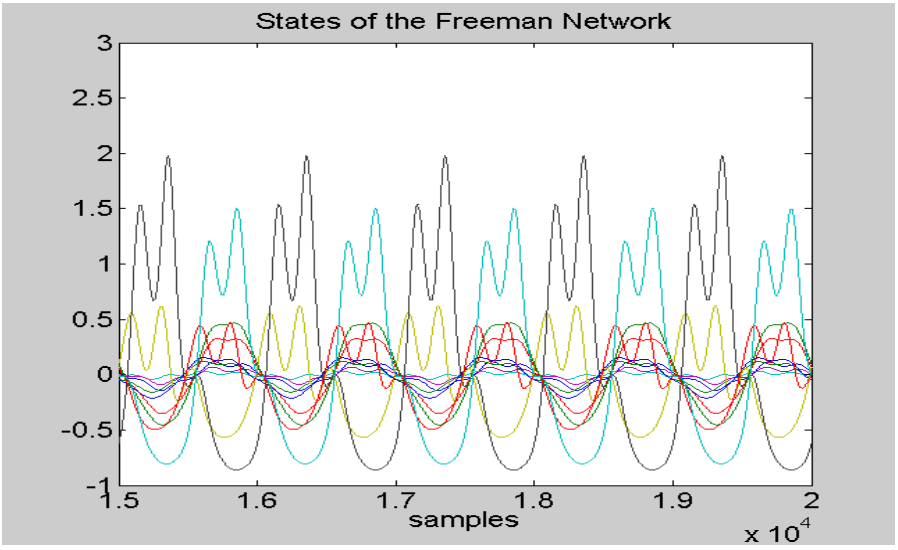
This section demonstrates a variety of experiments in order to demonstrate the proposed framework for Freeman's K sets. In all the experiments, we use the discrete-time Freeman model developed in (Tavares, 2001; Ozturk et al., 2003). The discretization exploits the fact that K0 set is composed of a separate linear dynamical part and a static nonlinear part. Continuous-time linear filter of K0 is discretized by the method called *impulse invariance transformation* resulting in an equivalent discrete-time filter.

Function approximation

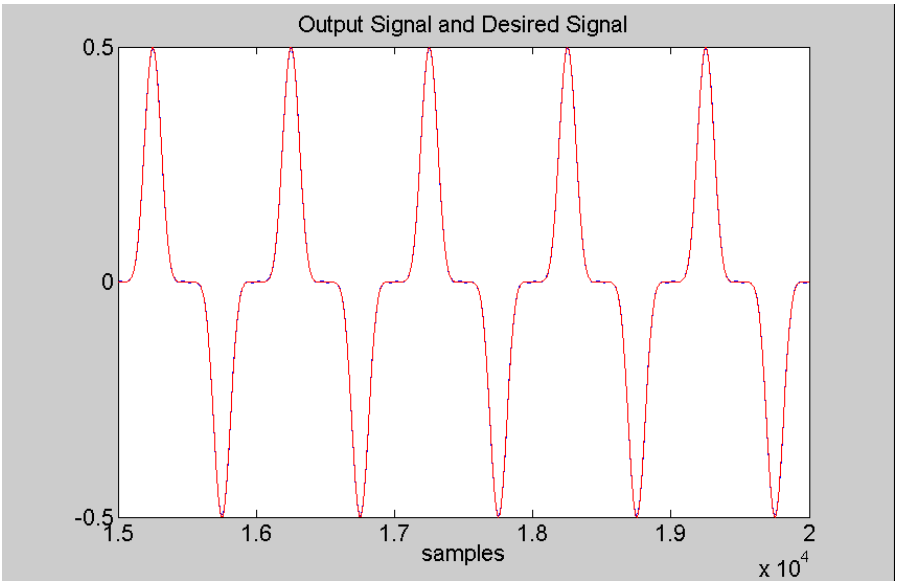
It has been demonstrated that FM can be used as an auto-associative memory (Xu et al., 2004; Tavares, 2001). However, the stored patterns are binary (0's and 1's) and static. We would like to first show that FM can handle continuous valued signals within the proposed framework with a function approximation experiment..

Consider an RKII network of 50 RKII sets and a single input channel connected to all excitatory cells with unity weights. RKII network is driven by the input signal, $\sin(2\pi t/1000\pi)$

and the target signal is $0.5\sin^7(2\pi n/1000\pi)$. The weight values K_{mg} and K_{gm} of all RKII sets and all inhibitory weights, Kii are set to 4, -1 and -0.002, respectively. These values are experimentally determined from biological data and same values are used in (Xu et al., 2004). The excitatory weight values are chosen from a gaussion distribution with zero mean and unity variance.



A



B

Figure 5-9. Function approximation with Freeman Model. A) States of the Freeman model. B) The overlay plot of the output and desired signals

The network is initialized to a random initial condition and the states of the RKII network (K0 outputs) are calculated. At this point, it is interesting to observe the echo states depicted in Figure 5-9 A. This plot shows the signals observed at outputs of some of the KO sets. As seen from the figure, due to the nonhomogenous nature of the system, there is a significant diversity in the signals at the nodes of the KII network creating the rich dynamics required in the echo states. Now, the next step is to construct the desired signal from the echo states using a memoryless readout network. For this example, we do not have direct connections from input to output. We will use a simple linear readout whose optimal weights can be calculated using Equation 2-4. The output of the optimal linear network is depicted in Figure 5-9 B along with the desired response. As seen from the figure, we get an almost perfect match between the desired response and the output of the Freeman model with the linear readout.

K0 set as the readout for function approximation

An alternative to the linear network as the FM readout is to use one of the Freeman's K sets. For systems with only one output, we will use the K0 set as the readout. In this section, we will derive the gradient based update equations for the K0 network using MSE as the cost function. This will allow us to use one of the components of the Freeman network indicating that the Freeman K sets contain all the ingredients necessary for computation in the echo state framework.

In the K0 set, the linear filter and the static nonlinear function are fixed and the input weights will be adaptable. We will derive the gradient-based update equations for the weight vector using the MSE as the cost function. Note that we do not have an analytic solution for the optimal weight vector since K0 comprises a linear dynamical part which is an infinite impulse response filter and also a static nonlinearity. Consider the K0 set depicted in Figure 5-5 A. Let us first fix the notation. The input to the K0, the input weight matrix, the signal at the input of the

linear part, the signal at the output of the linear part and the output of the K0 will be denoted by $\mathbf{x}(n)$, $\mathbf{w}^{out}(n)$, z(n), l(n), y(n), respectively. Then, the governing equations will be given by

$$z(n) = w^{out}(n)x(n)$$
$$l(n) = c_1 l(n-1) + c_2 l(n-2) + c_3 z(n-1)$$
$$y(n) = Q(l(n))$$
$$e(n) = d(n) - y(n)$$

The constants c_1, c_2 and c_3 are the coefficients of the discrete-time filter obtained from the impulse invariance transformation of the original continuous-time filter. The details of the derivation can be found in (Ozturk et al., 2003). The instantaneous cost function is defined as $e^2(n) = (d(n)-y(n))^2$. Then the weights will be optimized according to

$$w^{out}(n+1) = w^{out}(n) - \mu \frac{\partial(e^2)}{\partial w^{out}}.$$

In order to calculate the sensitivity of the cost with respect to the weights, we will use backpropagation over the topology, i.e.

$$\frac{\partial(e^2)}{\partial w} = \frac{\partial(e^2)}{\partial y}\frac{\partial(y)}{\partial l}\frac{\partial(l)}{\partial z}\frac{\partial(z)}{\partial w} = -2\dot{Q}(l)c_3 x.$$

Note that, although K0 includes a recurrent filter, the sensitivity of the cost w.r.t the weights can be calculated instantaneously. Because we do not need to update the filter coefficient but we only need to calculate the sensitivity of the filter output w.r.t the filter input which is just a constant, c_3. With this derivation, K0 can be employed as the readout for one output systems. For multi-output systems, we will use the KI network where the connection weights between the K0 sets of the KI network is fixed and the weights between the echo states and KI network is adaptable as above. The extension of the gradient update rule for the multi output case is trivial.

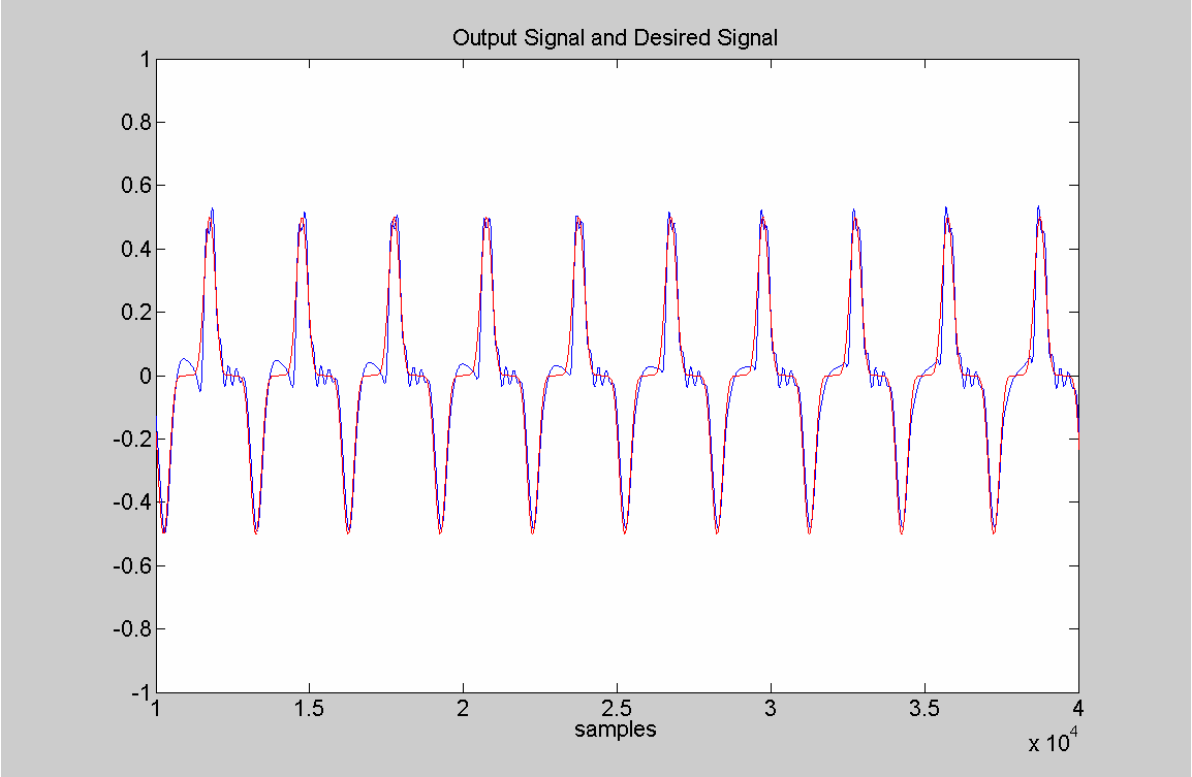


Figure 5-10. Overlay plot of the desired signal and output signal of the Freeman Model with K0 readout.

To demonstrate the idea of using K0 set as the readout network, we will take the same regression experiment. We will use the same parameters as in the previous section. Now, we will use the K0 set as the readout instead of the linear network and adapt the K0 as described above. Figure 5-10 shows the overlay plot of the desired signal and the output of the K0 set. As seen from the plot, we get a good match between the desired signal and the output of the Freeman network. However, the results are not as good as the simple linear readout since we do not have an optimal solution for the K0 readout.

Autonomous squence gneration with output feedback

This experiment demonstrates the ability of the system to autonomously generate a sequence of numbers. We will use a reduced KII network without any input units but with feedback from the output of the readout. The desired signal to be autonomously generated by the

system is $0.5\sin^3(2\pi n/1000\pi)$. An RKII network of 50 K0 sets without input units but with output feedback connections is constructed. The weight values K_{mg} and K_{gm} of all RKII sets and all inhibitory weights, Kii, are the same as the ones in the previous experiment. The excitatory weight values are chosen from a gaussion distribution with zero mean and unity variance. The output feedback connections from the linear readout to the KII network are chosen to be 0.1 and -0.1 with equal probability. During training, the states of the RKII set are driven by the desired response via output feedback.

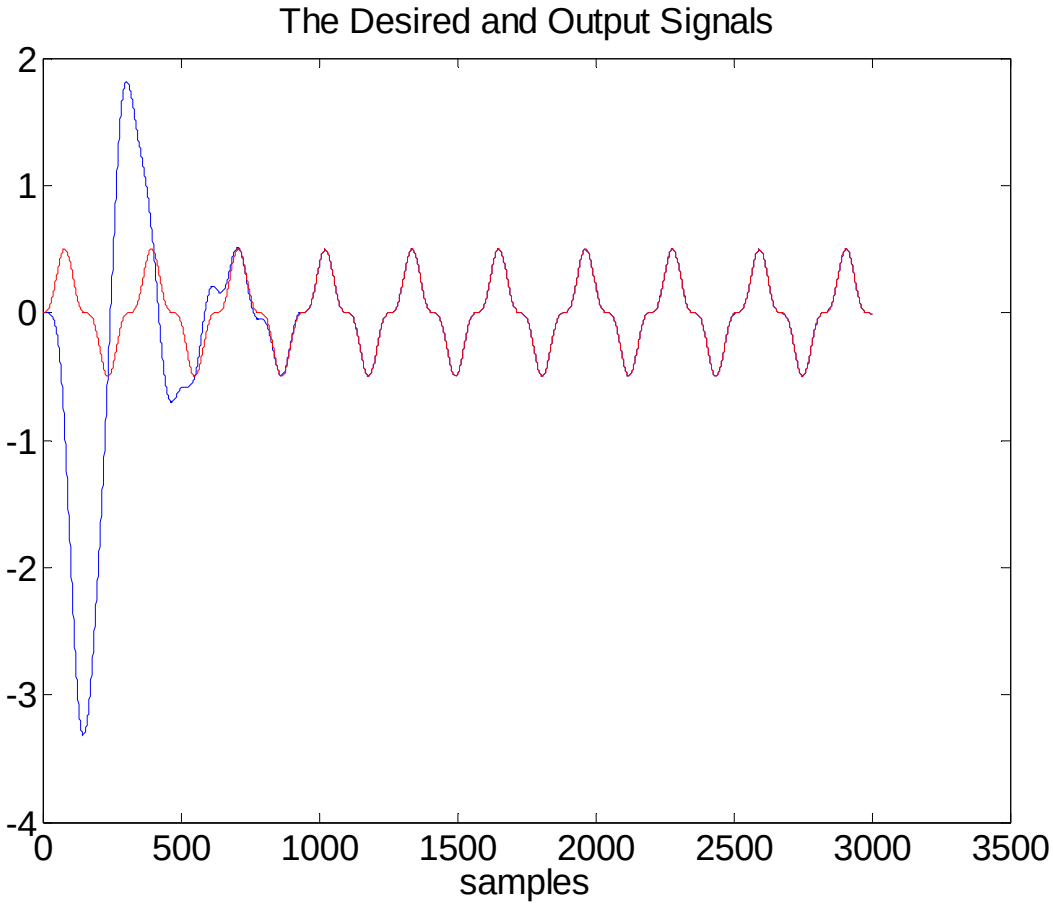


Figure 5-11. Autonomous signal generation with Freeman Model.

The network is initialized to a random initial condition and the states of the RKII network are calculated for 2000 samples of the desired response. The first 1000 samples corresponding to initial transient are eliminated and the optimal output weights are calculated using Equation 2-4.

The MSE value obtained during training is 1.4350×10^{-7}. During testing, the system is first run with for 1000 steps with the correct desired signal. After 1000 steps, the system is let run freely. Figure 5-11 shows the desired and output signals for the first 3000 steps of testing. As observed from the plot, there is a transient time interval for which the system tries to lock into the desired signal. But once it is locked, the system continues generating the signal autonomously. The MSE value for the last 1000 steps of testing is found to be 7.934×10^{-7}.

Binary parity check

This experiment investigates the use of Freeman model for binary logic computation. In particular, a parity check bit calculation will be carried out using a reduced KII set of 100 KO sets. We will attach one input unit and a linear readout with one output unit to the reduced KII network. The input will be a sequence of binary values, '0's and '1's and the desired signal will be the parity bit corresponding to the last 3 binary input values. More precisely, the desired signal will be the addition of the last 3 bit values modulo 2. In our experiment, we will represent each bit by a sequence of 100 samples. In other words, '0' will be represented by a sequence of a hundred 0's, 0..0, and similarly '1'.

We use a reduced KII network of 100 KO set. The weight values Kmg and Kgm of all RKII sets and all inhibitory weights, Kii are set to 0.5, -0.4 and -0.002, respectively. Kee values are set to be 0.1, 0.2 or 0 with probabilities 0.05, 0.05 and 0.90, respectively. We generate a sequence of 300 random binary numbers as the input and the corresponding parity bit sequence as the training data. This corresponds to a training sequence of 30000 data points since each bit comprises 100 samples. Again, the weights of the linear readout are calculated using the Wiener solution after discarding the first 1000 samples. In order to test the system, a random sequence of 300 binary numbers is generated and fed to the FM. Figure 5-12 shows a segment of the desired and output signals during testing. As we observe from the plots, the system output converges to

the desired response for all input bits. If the linear readout output is compared to a threshold value of 0.5 to make binary decisions, the system converges to the true binary value for the whole testing data without an error.

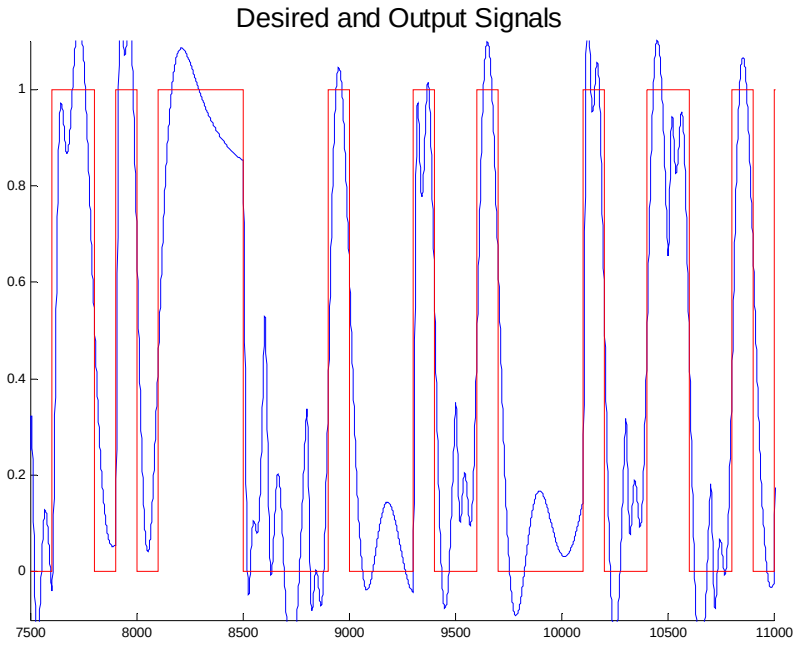


Figure 5-12. Binary parity check with Freeman Model.

Multi-attractor learning

In this experiment, we will train the readout to switch between multiple point attractors depending on the input (Jaeger, 2001). The network will have 20 input units and 20 output units attached. The system will have 20 point attractors that are controlled by the input signal. The input signals are in the form of rare pulses (with values 0 or 1, see Figure 5-13). At any given time, only one of the input channels can have nonzero value (at most one pulse at a time). Each output unit is associated with an input unit and when a pulse arrives at an input unit, the corresponding output unit value becomes 0.5 whereas the other output units take the value -0.5. The values of the output units remain the same until a new pulse arrives at another input unit. At

the arrival of this new pulse to an input unit, the output unit corresponding to that input unit activates to 0.5 and the remaining to -0.5. Figure 5-13 shows sample input signal and the corresponding output signal for a two input-two output system.

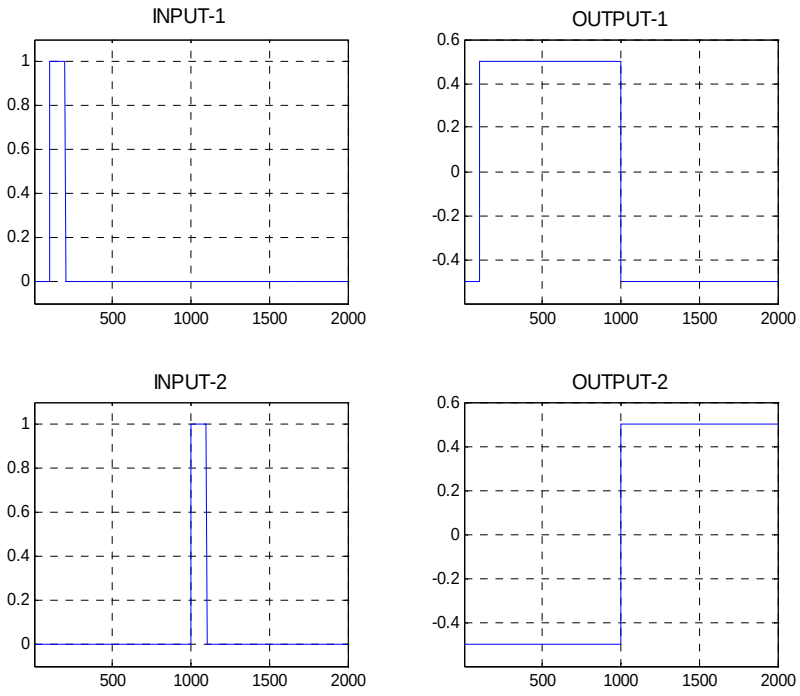


Figure 5-13. Sample input-output signals for multi-attractor learning.

We use a KI network of 100 K0 sets with 20 inputs units. 100 K0 sets are randomly connected having the weight values of 0.2, 0.1 and 0 with probabilities of 0.95, 0.025 and 0.025, respectively. This corresponds to a sparse connectivity of 5%. The input units are connected to the K0 sets with connection strengths of 5 or -5 with equal probability. The output units are connected to the states via output feedback having the weight values of 1, -1 and 0 with probabilites of 0.1, 0.1, 0.8.

We generated a sequence 8100 input samples and the corresponding output sequence for training. Starting from the time step one, a pulse is assigned to input channel 1 to 20 in order at every 400 steps. At time step 8001, a pulse exists in the input channel 1. The pulse width is

chosen to be 100 samples. The output sequence is assigned to 0.5 or -0.5 according to the rule specified above. The system is run with the input data generated while the output is teacher-forced by the output signal generated. The states are calculated for 8100 time steps using the discretization. Then, the output weights are calculated using Equation 2-4 after disgarding the first 1000 samples corresponding to the transient signal.

Figure 5-14 shows the output and desired reponse for 4 channels during training. The system is tested with an input-output sequence of 80000 samples. The system is initialized with the first 500 samples of desired signal forced as the output. Figure 5-15 shows the output and desired response for four channels during testing. The Freeman model is able to model all the switches without any errors during testing.

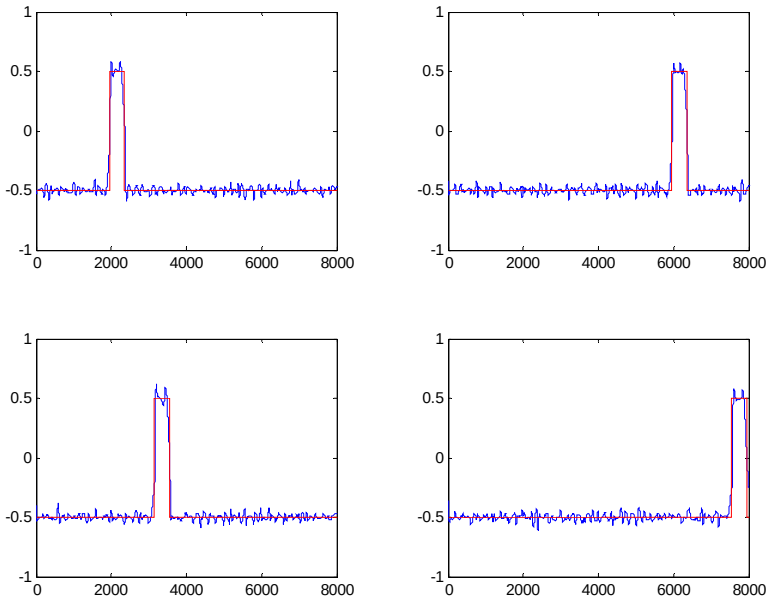


Figure 5-14 Desired and output signals during training for multi-attractor learning with Freeman Model.

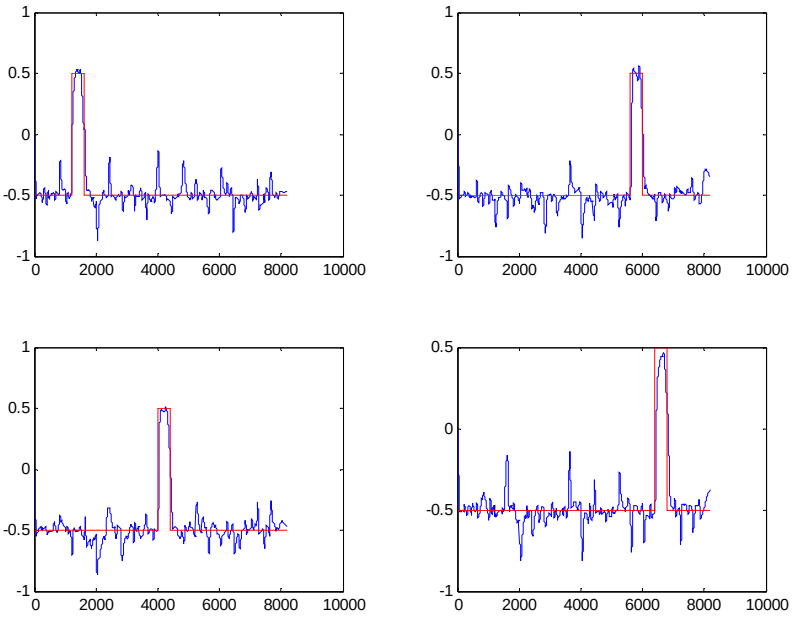


Figure 5-15. Desired and output signals during testing for multi-attractor learning with Freeman Model.

CHAPTER 6
APPLICATIONS

This chapter explores the use of ESN for various real-world applications.

Complex Echo State Networks

Most of the signal processing tools have been developed to operate on signals with real values. However, there exist areas where signal processing in the complex domain is required. For example, in digital communication, the symbols sent through a communication channel are usually implemented as points in complex domain. In phase shift keying (PSK) the binary information is conveyed by changes in the phase of the reference signal (Proakis, 2001). In quadrature PSK, four different phase values are used to encode two bits of data. In quadrature amplitude modulation (QAM), both the amplitude and the phase of the reference signal are utilized to encode information more efficiently with increased data transfer rate. The most commonly used QAM are 32QAM, 64QAM, 128QAM and 256QAM where 5 to 8 bits of data are transferred per symbol. Phase plays an important role in these systems and the symbols are represented by complex numbers. The straightforward method to deal with complex numbers is to process the real and complex parts of the signal separately. However, this is suboptimal because it neglects the interdependency between the real and complex part of the signal (Kim and Adali, 2002). Alternatively, the existing tools of signal processing, which are originally designed to deal with real signals, have been effectively modified to operate in complex domain. In the neural networks literature, the popular back propagation algorithm (BP) for multilayer perceptron (MLP) was originally developed in real domain and in the past two decades it has been effectively used in many diverse areas (Widrow and Lehr, 1990; Haykin, 1998). Kim and Adali have rigorously developed the complex back propagation (CBP) algorithm for complex MLP (CMLP) (Kim and Adali, 2002). CMLP uses fully complex activation functions instead of

split complex functions that treat real and imaginary parts of the signal separately, and achieve consistently better performance. However, the difficulty with the MLP training in complex domain arises mostly from the selection of activation functions that can operate in complex domain. The nonlinear functions have to be both analytic and bounded since the backpropagation algorithm requires derivatives of the nonlinearities to calculate the weight updates. Moreover MLPs are trained in an iterative approach which is usually slow and computationally intensive. The model parameter such as the input dimension and number of hidden units must be selected through exhaustive testing as they play a crucial role in faster convergence and minimal training error.

In this section, we propose ESNs for complex signal processing. The use of ESNs for complex domain is very convenient since system training is equivalent to simple linear regression, which is trivial in the complex domain. The derivatives of the nonlinear activation functions are never necessary since the recurrent part is fixed apriori. The nonlinear activation functions have to be modified similar to (Kim and Adali, 2002) to ensure bounded output. However, being analytic is not anymore required since ESN does not require the computation of derivatives for training. We compare CESNs with CMLP and a linear network trained with complex least mean squares (CLMS) algorithm on a complex channel equalization experiment. We demonstrate that CESN can achieve lower symbol error rates with a simple and fast learning.

Channel Equalization

The performance of the CESN is tested on two benchmark nonlinear channel equalization problems. Fig. 6-1 depicts the block diagram of an additive noise nonlinear communication channel. The modulated signal corrupted by additive noise is passed through a nonlinear channel with memory. An equalizer is used to remove the effects of channel and noise before the signal is demodulated. The equalizer ideally behaves like an inverse system of the nonlinear channel. The

noise is considered to be additive white Gaussian. The goal is to implement the equalizer using adaptive signal processing techniques. Here, both the signals and the channel are complex valued.

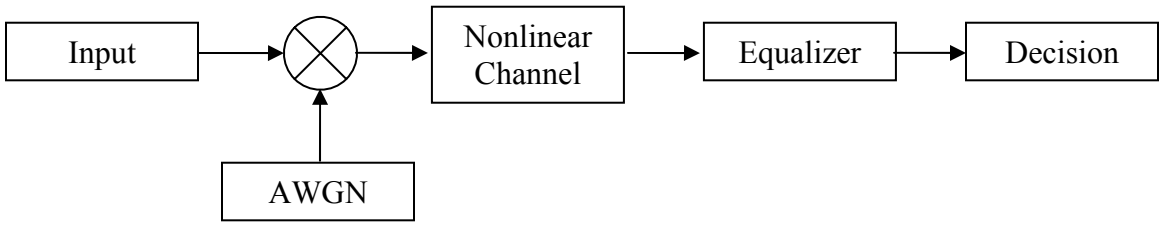


Figure 6-1. Block diagram of a communication system with nonlinear dispersive channel.

First problem is from satellite communication domain and is also used by Kim and Adali [2, 8]. The nonlinear channel is modeled by a Volterra series expansion given by

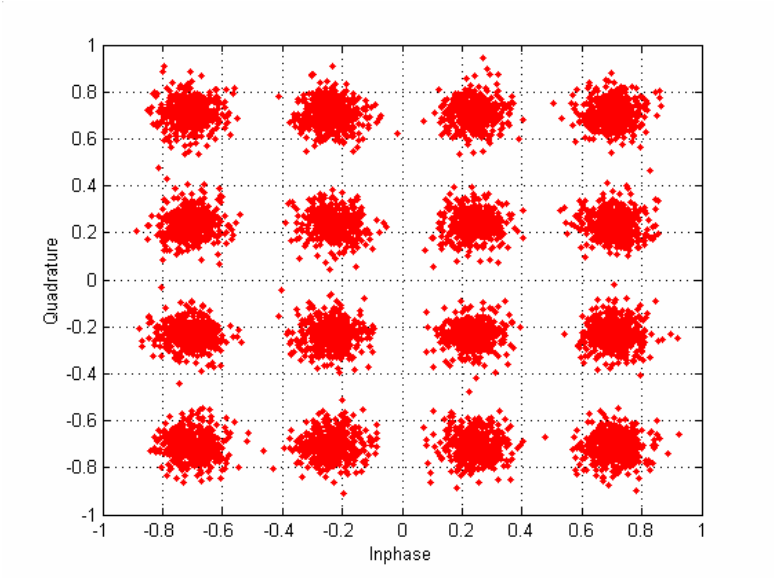
$$x_n = \sum_{k=1}^{\infty} \sum_{n_1} \cdots \sum_{n_{2k-1}} a_{n-n_1} a_{n-n_2} \cdots a_{n-n_k} a^*_{n-n_{k+1}} \cdots a^*_{n-n_{2k-1}} H^{(2k-1)}_{n_1,\ldots,n_{2k-1}} + \nu_0$$

where ν_o is the down-link noise and Gaussian in nature, a_n is the n^{th} information symbol and H is Volterra series coefficient that describes the effect of channel nonlinearity. We used a channel described by the following Volterra series coefficients (Benedetto and Biglieri, 1982; Kim and Adali, 2002).

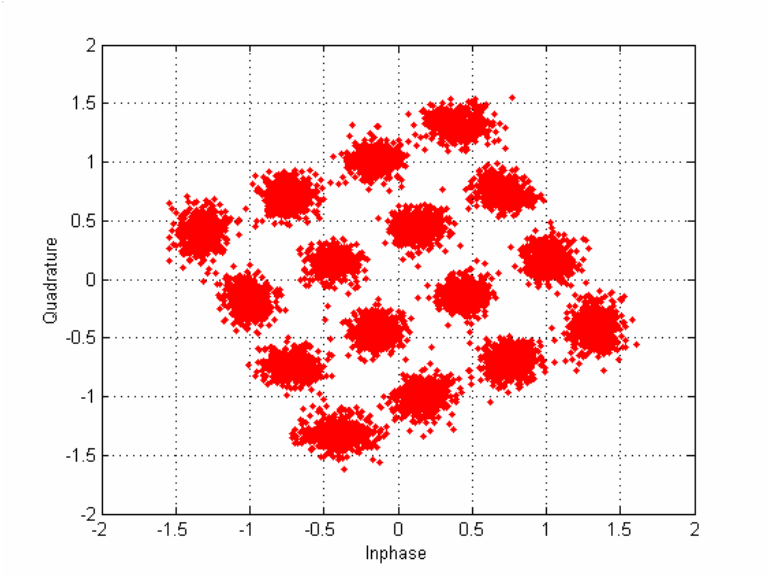
Linear coefficient:	$H_0^{(1)} = 1.22 + i0.646$,	$H_1^{(1)} = 0.063 - i0.001$,
	$H_2^{(1)} = -0.024 - i0.014$,	$H_3^{(1)} = 0.036 + i0.031$
3^{rd} order coefficient:	$H_{002}^{(3)} = 0.039 - i0.022$,	$H_{330}^{(3)} = 0.018 - i0.018$,
	$H_{001}^{(3)} = 0.035 - i0.035$,	$H_{003}^{(3)} = -0.04 - i0.009$,
	$H_{110}^{(3)} = -0.01 - i0.017$	
5^{th} order coefficient:	$H_{00011}^{(5)} = 0.039 - i0.022$.	

We use a rectangular 16QAM modulation scheme where the 16 symbols are equally spaced in the complex domain and the farthest point from origin has a magnitude of 1. The

amplitude and phase of the complex noise are independently generated from Gaussian and uniform distributions, respectively.



A



B

Figure 6-2. Constellation diagram of 16 QAM. A) Input symbols. B) Output symbols.

Figures 6-2 A and 6-2 B show the constellation diagrams of the input symbols (under 12dB signal-to-noise-ratio [SNR]) and the output symbols of the nonlinear channel, respectively. The symbol energy of the 16QAM system is given by $2.5d^2$ where d is the horizontal or vertical distance between two adjacent points. The bit energy is calculated by dividing the symbol energy by 4. SNR is calculated by dividing the bit energy by twice the noise variance.

We compared the performance of three different networks: a linear filter, a CMLP and CESN. All networks are trained with 20000 data points generated from the 16QAM configuration with equal probability. The optimal weights of the 10 tap linear filter are computed using Equation 2-4. We used a CMLP with 10 input units, 5 hidden units and a single input unit. We use arcsinh as activation function. The derivative of the arcsinh(z) is given by $(1+z^2)^{-1}$.

The weights of the CMLP are trained using the complex backpropagation algorithm derived in (Kim and Adali, 2002).We used a 30 unit CESN. The input weights are selected from values -0.1 and +0.1 with equal probability. The internal connection weight matrix is designed with uniform pole distribution principle (Ozturk et al., 2007). ESN designed with this principle is denoted by ASE-CESN while the ESN initialized with random weights is denoted by CESN. The activation function of the internal units is the complex sigmoidal function,tanh(x). For CESN and CMLP the network is trained 10 times for every SNR value and the best performance is recorded. All three networks are tested with 10^6 points input sequence where each point is selected from the 16 points with equal probability. We trained CESN with only 1000 data samples. Figure 6-3 shows variation of the symbol error rate (SER) for different SNR values for four networks. As observed from the figure, ASE-CESN shows the best performance compared to the other networks. The main reason is that ESN avoids backpropagation through the complex nonlinear functions. The CLMS performs very similar to CMLP since the CLMS also uses the

optimal solution and the problem is only mildly nonlinear. SER of ASE-CESN drops to zero after 12dB SNR while the SER for CLMS and CMLP still decreases slowly. CESN that used much fewer points than CLMS and CMLP can still perform as well due to the simple training.

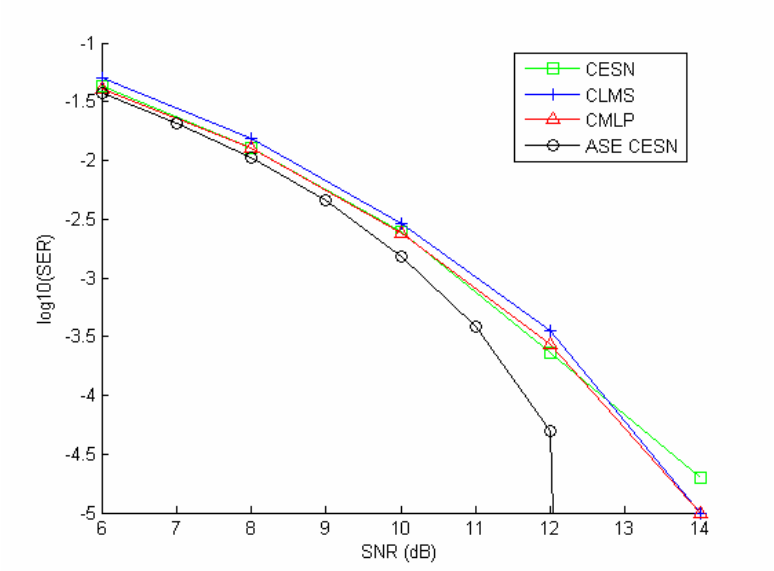


Figure 6-3. SER vs SNR plot for four networks.

The second problem is another channel equalization problem from digital communication domain and is also used in (Cha and Kassam, 1995). Here the nonlinear channel is described by the following equations.

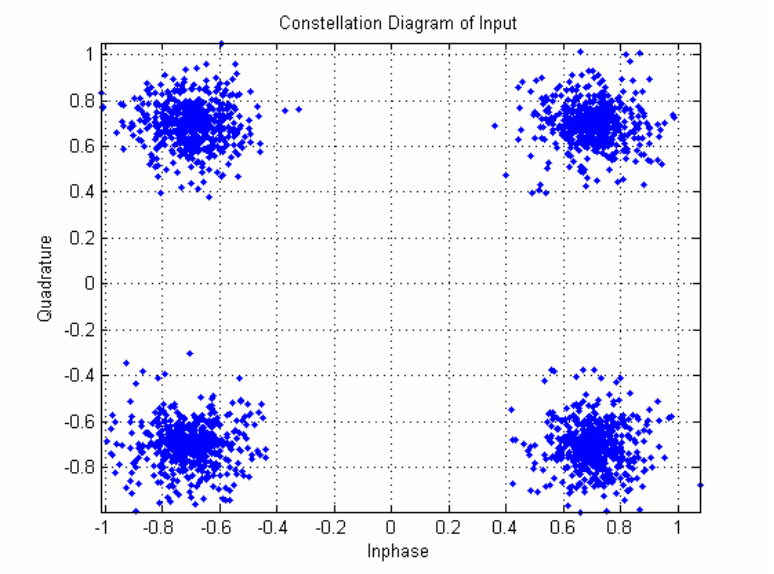
$$y_n = x_n + 0.1x_n^2 + 0.05x_n^3 + v_n$$

$$x_n = (0.34 - i0.27)u_n + (0.87 + i0.43)u_{n-1} + (0.34 - i0.21)u_{n-2}$$

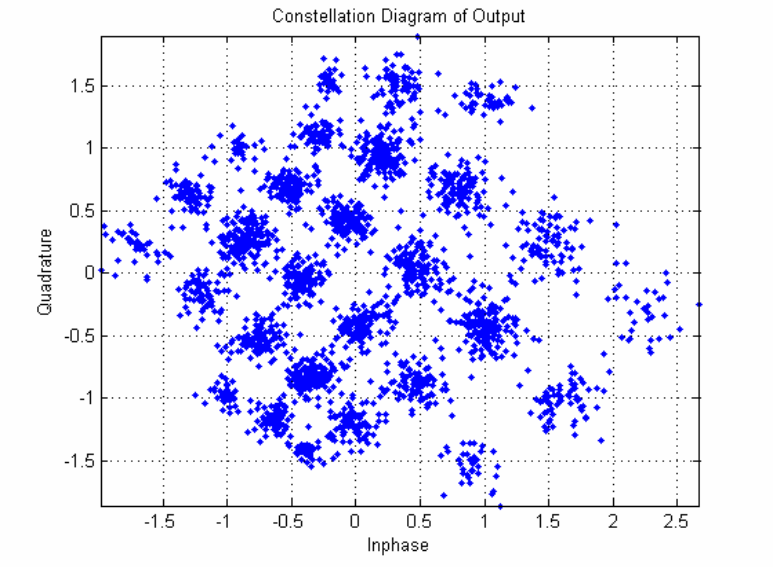
where v_n is independent identically distributed white Gaussian.noise,

We use a rectangular QPSK modulation scheme where the 4 symbols are equally spaced in the complex domain and the farthest point from origin has a magnitude of 1. Figures 6-4 A and 6-4 B show the constellation diagrams of the input symbols (under 12dB signal-to-noise-ratio [SNR]) and the output symbols of the nonlinear channel, respectively. The symbol energy of the QPSK system is given by $0.5d^2$ where d is the horizontal or vertical distance between two

adjacent points. The bit energy is calculated by dividing the symbol energy by 2. SNR is calculated by dividing the bit energy by twice the noise variance.



A



B

Figure 6-4. Constellation diagram rectangular QPSK. A) Input symbols. B) Output symbols.

We compared the performance of three different networks: a linear filter, a CMLP and a CESN. All networks are trained with 10000 data points generated from the QPSK configuration with equal probability. The optimal weights of the 10 tap linear filter are computed using Equation 2-4. We used a CMLP with 3 input units, 10 hidden units and a single input unit. The same ESN models used in the previous sxample are utilized. Here, CESN uses 500 training samples. Fig 6-5 shows variation of the symbol error rate (SER) for different SNR values for three networks. Similar to the previous experiment, ASE-CESN shows the best performance. CESN can still perform very close to CLMS and CMLS eventhough CESN utilizes a small number of training points.

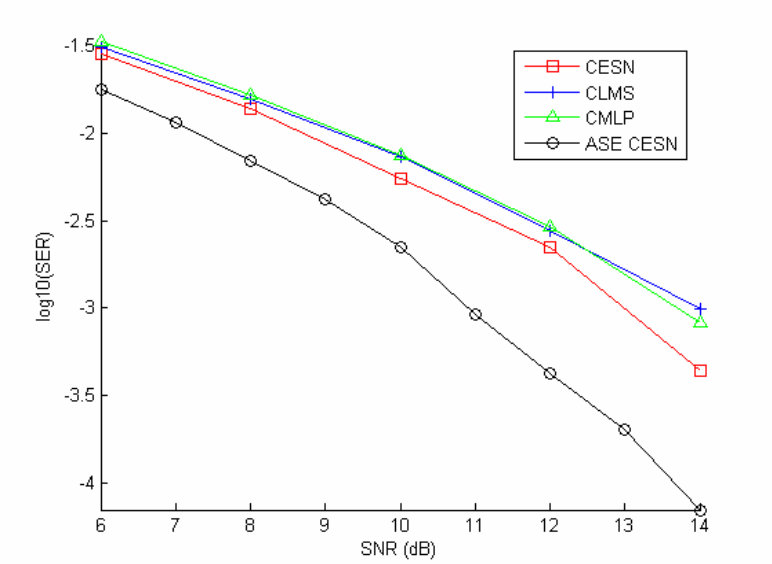


Figure 6-5. SER vs SNR plot for four networks.

Brain Machine Interfaces

This section of the dissertation is a result of the joint study with Aysegul Gunduz (Gunduz et al., 2007). We have provided the expertise in ESNs while they provide the data and expertise in brain machine interfaces (BMI). The results of the study are accepted for publication (Gunduz et al., 2007).

In BMI, the goal is to model the relationship between a kinematic variable such as hand position of a primate and the electrophysiological data collected from its brain. BMIs provide new rehabilitation options to patients who have lost their motor functions due to damage to the nervous system. Therefore, much effort has been put into the research on neuroprosthetics for motor control and BMIs.

In order to model the relationship between brain signals and hand position in BMIs, simultaneous recordings have been collected from the brain and the primates hand position. The brain signals can be from acquired from microarray recordings (action potentials and local field potentials), on the cortex via subdural grids (electrocorticogram [ECoG]) and on the scalp (electroencephalogram [EEG]). The levels of abstractions of these signals are referred to as microscopic, mesoscopic, and macroscopic respectively.

In the BMI literature, most of the studies focus on modeling the hand position of a primate from the activities of individual neurons collected from various sites in the sensorimotor cortex (Chapin et al., 1999; Serruya et al., 2002). However, the main difficulty in the practical use of these systems is the crude invasiveness of microelectrodes, which prevents their use in human subjects. On the other hand, non-invasive scalp EEG contains a summation of dendritic and axonal currents of millions of neurons over the cortical impedance (Sanchez et al., 2007). The macroscopic EEG lacks the spatial resolution of microscopic recordings and have very low signal-to-noise ratio. In this chapter, our focus will be on the mesoscopic level recording called electrocorticogram (ECoG). ECoG when compared with EEG, has the great advantage of being more spatially specific (smaller electrodes), much less contaminated by noise, and can display better frequency resolution because of its closer proximity to the neural tissue (Sanchez et al., 2007).

Current BMIs based on ECoG recordings either require movement related averages of trials for extraction of features (Mehring et al., 2004; Rickert et al., 2005) or imagery association of another task to the kinematic behavior (e.g. protruding tongue) (Leuthard et al., 2004). Movement related potentials, though a valuable analysis tool, are not suitable for building real time BMIs. In our experimental paradigms, we undertake the prediction of the kinematic behavior of the patients without employing any other task.

The parameters that can be extracted from ECoG to predict hand trajectories is still under investigation (Gunduz et al., 2007). Still, since amplitude modulation plays a key role in both neuronal activation and rate coding, this study seeks to explore preprocessing modalities that emphasize amplitude modulation in the ECoG above the level of noise and background fluctuations in order to derive the commands for complex motor control tasks (reaching and grasping) (Sanchez et al., 2006).

Mapping amplitude modulated control features to motor behavior requires models of low orders that are easy to train. Simple linear filters, such as Wiener filters, are very easy to train but have limited performance in accuracy. Time delayed neural networks (TDNNs) require parameters of very high orders especially when the system is a multiple-input multiple output system as in BMIs. Recurrent neural networks (RNNs) which have high accuracy in modeling are very hard to train. In this chapter, we study and compare mappings from amplitude modulated ECoG features to hand movements involving reaching and pointing tasks using linear filters, echo state networks and leaky echo state networks.

The data for this study was obtained from a patient undergoing extraoperative subdural grid electroencephalographic monitoring for the treatment of intractable complex partial epilepsy at Shands Hospital, University of Florida. The experimental paradigms were approved by the

University of Florida Institutional Review Boards.The patient, a right-handed 15 year old female, underwent a presurgical work-up that included scalp EEG, formal neuropsychological testing, and MRI. The patient's IQ and motor functions were verified to be nonfocal by the absence motor or sensory deficits on neurological examination (Sanchez et al., 2007)

The patient was implanted with subdural grid electrodes according to established protocols (Lesser et al., 1990). The grids consisted of a 1.5mm thick silastic sheet embedded with platinumiridium electrodes (4mm diameter with 2.3mm diameter exposed surface) spaced at 1-cm center-to-center distances. The anatomical location of the grids was based upon the medical team's recommendation for epilepsy evaluation. The primary motor cortex was determined by evoked potentials and direct electrical stimulation of the subdural grids (Jasper and Penfield, 1954) and was found to be far from the seizure focus. The 32-channel electrode grid was covering premotor (PMA), primary motor (M1), and somatosensory (S1) cortices based on the patient's cytoarchitecture (Sanchez et al., 2007).

Experimental Setup

Extraction of control features from ECoG recordings for neuroprosthesis is facilitated by continuously time synchronizing neuronal modulation with the well defined behavioral paradigm (Sanchez et al., 2006). The behavioral tasks used in this neuroprosthetic design focus on arm reaching and pointing toward a visual input (Sanchez et al., 2007). The patient was cued to follow with her index finger a predefined cursor trajectory presented on an LCD screen with an active area of (20 x 30cm) while neuronal modulations from the implanted ECoG electrodes were simultaneously being recorded (Sanchez et al., 2007). The trajectory consisted of a widely performed center-out task (Georgopoulos et al, 1982) and a target selection task (Desmurget et al., 1998). This behavior mimics a computer user's movement to select an icon on the screen. In

a single session, the patients were required to repeat the entire task six times and this was repeated for each trial (Sanchez et al., 2006).

Spectrally preprocessed cortical recordings have been shown to correlate with a variety of visual, auditory, and motor tasks in frequency bands comprised of slow potentials (1-60Hz) (Pfurtscheller et al., 2003),the gamma band (60-100Hz) (Crone et al., 1998), fast gamma band (100-300Hz) (Sinai et al., 2005) and ensemble depolarization (300-6kHz) (Engel et al., 2005). Therefore we define the band specific amplitude modulated control features, u(t) as the integrated power of the ECoG voltage V(t) in non-overlapping bins of 100 msec given by

$$u(t_n) = \sum_{t_i=0}^{100ms} V^2(t_n + t_i), \tag{6-1}$$

where $t_{n+1} = t_n + 100ms$.

Methods

The simplest method to map the neuronal activity to hand movement is the linear Wiener filter topology. There are 32 input and 2 output channels. For each input channel, a tap delay line with 25 taps is used. The filter order was optimized by scanning tap delays from 5-30. At each time sample, the filter computes the outputs using 2.5 milliseconds of the past inputs. The model was trained on 3 minutes of recordings and tested on a data segment of 1.5 minutes.

We use an ESN with 500 internal units. The 32x500 input weight matrix $\mathbf{W}^{in}$ is fully connected with values of +1 or -1 with equal probability. A spectral radius of 0.9 is chosen.

In BMI problem, the neuronal activity that is input to the system is a fast changing signal whereas the hand position is at a relatively slow pace. Therefore, we alternatively used a leaky ESN network with a built-in low-pass filter to further increase the memory depth of the system and slow-down the input signal. The update equation of the leaky ESN is given by

$$\mathbf{x}(n+1) = (1 - \mu Ca)\mathbf{x}(n) + \mu C\mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n))$$

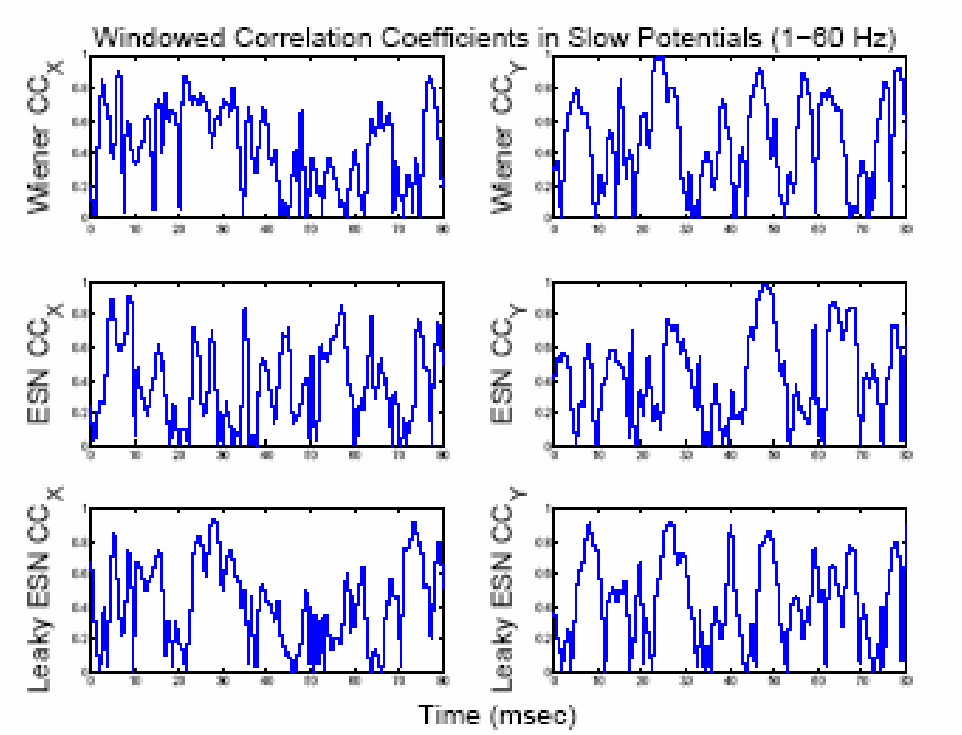
Here μ, C and a are the leakage parameter, time constant and decay factor, respectively. They define the level of low-pass filtering the system introduces. The number of processing elements for the leaky ESN is experimentally determined to be 200. The spectral radius was kept at 0.9 and the leakage parameters μ, C and a were experimentally determined for each band. In both architectures one second of data was discarded as transients.

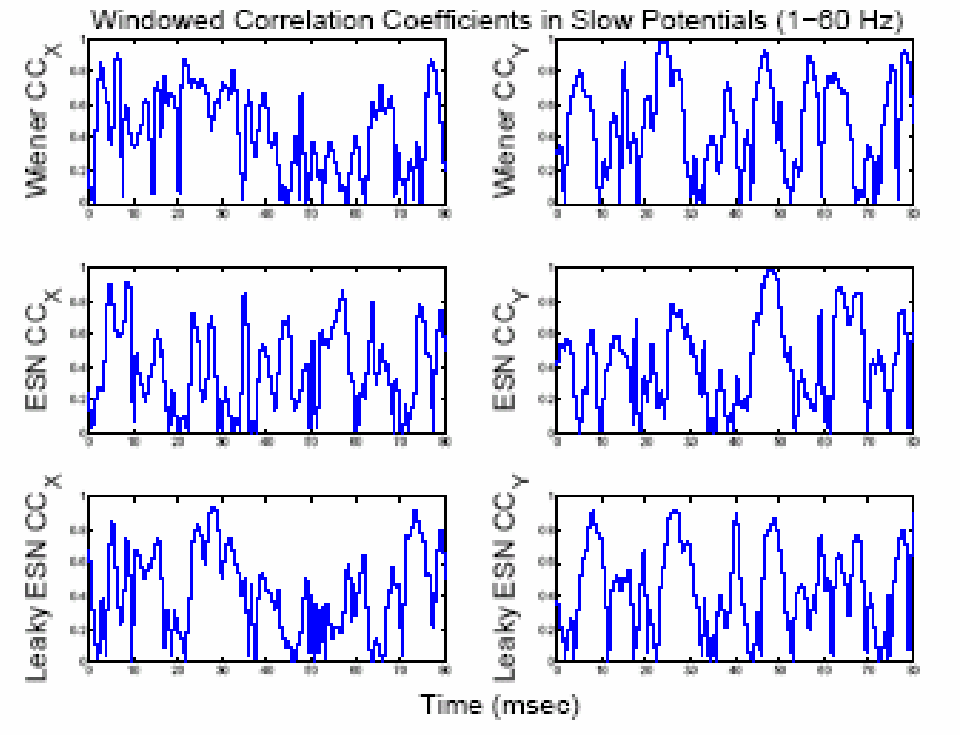
Averaged correlation coefficients (CC) between the actual hand trajectory and the filter outputs computed over non-overlapping windows of 5 seconds are presented in table 6-1. Figure 6-6 shows the variation of CC over time. The performance of each model drops when the patient was switching between tasks. The performance along the y-direction is evidently better for all topologies and all frequency bands compared to x-direction. Across spectral bands the most accurate reconstruction was observed in the ensemble activity (300-6kHz). Overall, the leaky ESN performed better than the basic ESN and Wiener filter, increasing the CC performance of the Wiener filter by 15% in the y-direction in the 300-6kHz band. Improvement in results with ESNs compared to the linear filter decreased as the input spectral activity slowed down. This may be an indication that the slower frequency bands contribute less to the modulation of hand kinematics.

Table 6-1. Averaged correlation coefficients between the actual hand trajectory and the model outputs computed over non-overlapping windows of 5 seconds

TESTING PERFORMANCE

Frequency Bands	Wiener Filter		ESN		Leaky ESN	
	X-pos CC	Y-pos CC	X-pos CC	Y-pos CC	X-pos CC	Y-pos CC
1-60 Hz	0.33 ± 0.16	0.41 ± 0.25	0.39 ± 0.26	0.43 ± 0.26	0.41 ± 0.25	0.44 ± 0.26
60-100 Hz	0.35 ± 0.24	0.41 ± 0.22	0.37 ± 0.25	0.39 ± 0.23	0.43 ± 0.25	0.45 ± 0.26
100-300 Hz	0.34 ± 0.21	0.35 ± 0.25	0.33 ± 0.22	0.50 ± 0.27	0.36 ± 0.26	0.52 ± 0.27
300-6 kHz	0.39 ± 0.26	0.48 ± 0.27	0.50 ± 0.27	0.61 ± 0.29	0.49 ± 0.26	0.63 ± 0.28


Windowed Correlation Coefficients in Slow Potentials (1−80 Hz)
Wiener CC_X
Wiener CC_Y
ESN CC_X
ESN CC_Y
Leaky ESN CC_X
Leaky ESN CC_Y
Time (msec)


Windowed Correlation Coefficients in Slow Potentials (1−80 Hz)
Wiener CC_X
Wiener CC_Y
ESN CC_X
ESN CC_Y
Leaky ESN CC_X
Leaky ESN CC_Y
Time (msec)

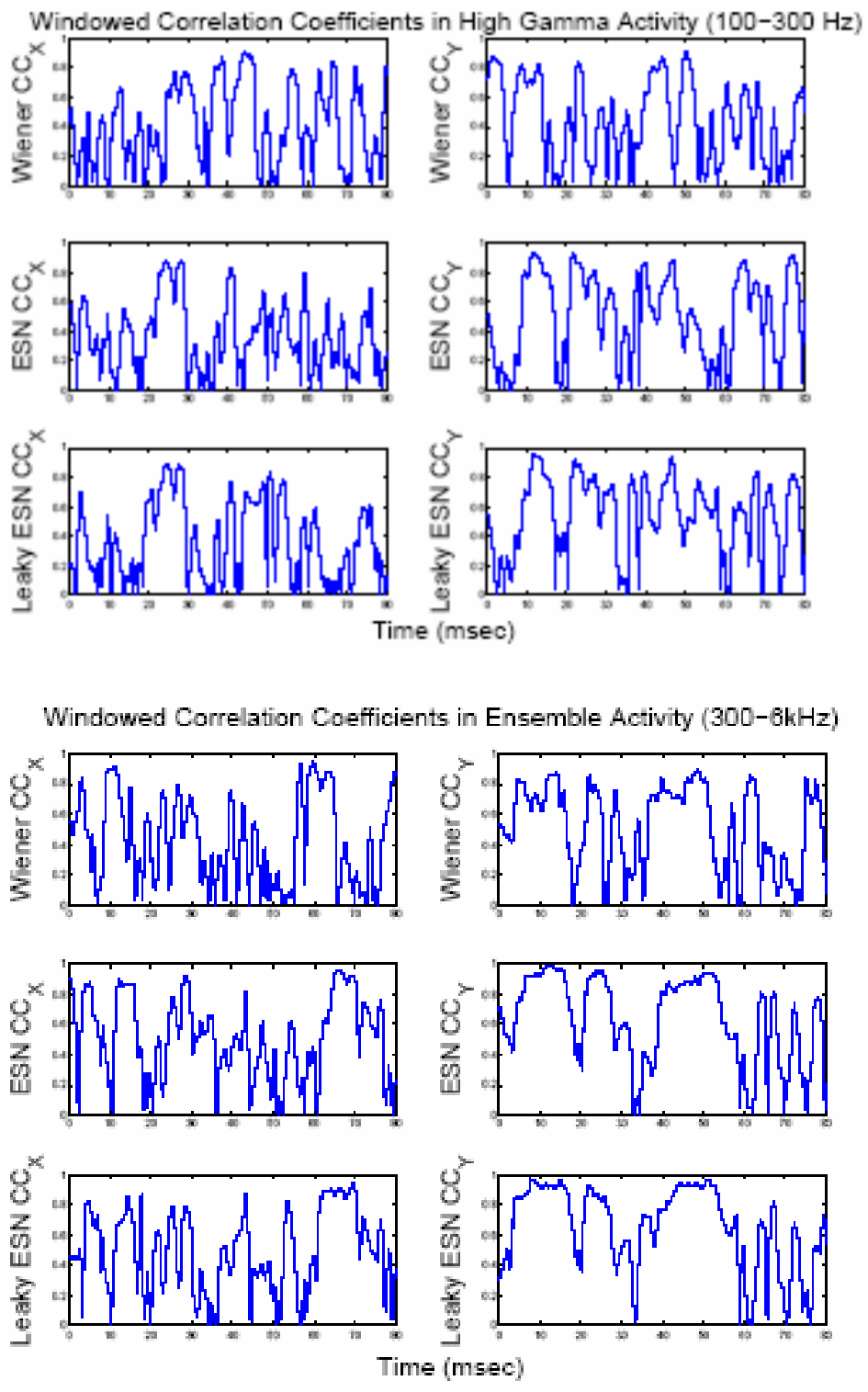


Figure 6-6. Comparison of three methods in brain machine interface modeling. Windowed correlation coefficients in all spectral bands.

Forecasting Water Inflow Using the Echo State Network

This section of the dissertation is a result of the joint study with Rodrigo Sacchi (Sacchi et al., 2007). We have provided the expertise in ESNs while they provide the data and expertise in water inflow forecast. The results of the study are accepted for publication (Sacchi et al., 2007).

We use the ESN for the prediction of the hydropower plant reservoir water inflow. Being able to predict the water inflow is fundamental in terms of planning the hydrothermal power system operation. In this study, we used a database of average monthly water inflows of Furnas plant, one of the Brazilian hydropower plants. The goal is to predict the value of the water inflow for the next month using previous data. The performance of the ESN is compared with SONARX network, RBF network and ANFIS model.

Water Inflow Problem

The greatest difficulties to forecast water inflow arises due to the nonstationary, nonGaussian nature of the time series due to changing weather conditions (wet and dry seasons) in different periods of the year across years.

Forecasting models based on Box & Jenkins methodology have been largely used in streamflow forecasting problems (Salas et al., 1982; MecLeod, 1994). However, those parametric auto-regressive models assume linear relationship (Box et al., 1994). This is a major limitation since the nature of the water inflow problem requires nonlinear modeling. Alternatively, ANNs have been proposed for time series analysis due to their ability to deal with nonlinear input-output relationships (Haykin, 1998). MultiLayer Perceptrons (MLP) with a backpropagation algorithm, the most popular of them, have been applied to forecast streamflow with promising results (Atiya, 1999). The examples of the successful application of ANN architectures for water inflow prediction are (Coulibaly and Anctil, 1999;Valenca et al., 2005;Sacchi et al., 2004).

The main problem with ANNs is the number of parameters which is usually big due to the difficulty of the problem. Moreover, a tapped delay line has to be used as a preprocessor to provide short term memory for static architectures such as MLP or RBF. The selection of the optimal embedding that defines the most important previous measured samples which must be considered as input data of the predictor is not trivial.

ESNs provide nonlinearity with training reduced to simple linear regression. Moreover, since the system has internal built-in memory resulting from the feedback connections, it is not necessary to embed the input signal before further processing. The training of the linear readout weights can be done analytically; hence very fast. In this section, we use the ESN as a one-step-ahead predictor of the monthly average hydropower plant reservoir water inflow. The performance of ESN is evaluated and compared to the Self-Organizing Nonlinear Auto-Regressive model with eXogenous input (SONARX) model (Barreto and Araújo, 2001), the Radial Basis Function (SONARX-RBF) network (Sacchi et al., 2004) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) model (Jang, 2003).

Results

In this case study, the data collected from Furnas hydroelectric power plant (HPP) was used since it is without operative influence due its relative position on the cascade. Monthly water inflow data from 1931 to1994 is available. A five year period from 1972 to 1976 is selected as the testing data. The remaining data is used to train the models.

Since this is a dynamical modeling problem, most of the models require embedding a short term history of the time series by using a tapped delay line. The states are formed by delayed versions of the input signal, i.e., $x_1(t) = u(t)$, $x_2(t) = u(t-1)$, $x_3(t) = u(t-2)$, where t denotes the current month and u(t) is the value of the water inflow sample for the month t and $x_1(t)$, $x_2(t)$, $x_3(t)$ are the states which are the inputs of the predictor model. Alternatively, we can use two

short periods, one from the immediate past and the other from the previous year, i.e., $x_1(t) = u(t)$, $x_2(t) = u(t-1)$, $x_3(t) = u(t-2)$, $x_4(t) = u(t-10)$, $x_5(t) = u(t-11)$, $x_6(t) = u(t-12)$. This allows the use of not only the monthly information but also seasonal information from previous year.

The desired response at month t is defined by u(t+1), which is the value of the water inflow for the next month. The performances of the proposed methods are compared w.r.t. 4 different metrics: MSE, root MSE (RMSE), mean absolute error (MAD), and mean percentage error (MPE). MAD and MPE are defined as

$$MAD = \frac{1}{T}\sum_{t=1}^{T}|y(t) - d(t)|$$

$$MPE = \frac{100}{T}\sum_{t=1}^{T}\frac{|y(t) - d(t)|}{d(t)}$$

where T is the total number of samples and y(t) and d(t) are the model output and desired signals, respectively.

Table 6-2 shows the performances of the models (ESN, SONARX, SONARX-RBF and ANFIS) w.r.t. the error criterion specified above.

Table 6-2. Performance comparison of the models for water inflow prediction

Models	*Embedding*	*MSE (x10^4)*	*RMSE*	*MAD*	*MPE (%)*
Echo State Network	built-in	5.97	244.45	177.41	21.04
SONARX	t,t-1,t-2	21.78	466.73	320.13	37.94
	t,t-1,t-2,t-3,t-4,t-5	12.20	349.30	253.18	28.78
	t,t-1,t-2,t-10,t-11,t-12	13.54	367.90	261.04	28.32
SONARX-RBF	t,t-1,t-2	9.24	303.94	233.24	29.35
	t,t-1,t-2,t-3,t-4,t-5	6.35	252.02	187.53	22.41
	t,t-1,t-2,t-10,t-11,t-12	7.13	266.94	202.56	24.35
ANFIS	t,t-1,t-2	9.45	307.47	235.67	29.89
	t,t-1,t-2,t-3,t-4,t-5	7.07	265.97	198.52	24.27
	t,t-1,t-2,t-10,t-11,t-12	6.72	259.25	205.99	25.02

We can see that ESN, SONARX-RBF and ANFIS performed significantly better than SONARX model, which has an intrinsic error since it is based on Vector-Quantized Temporal Associative Memory (VQTAM).

ESN performs slightly better than all other models, even though it presents a significantly simpler, and faster training algorithm. SONARX-RBF with 5 taps performs very close to ESN whereas with other selection of taps, ESN performs significantly better. This shows the importance of the selection of the number of taps for performance.

The prediction results of the water inflow during training are shown in Figure 6-7. The blue line refers to the desired time series, and the red one refers to the "forecasted" values water inflow. Based on the figure, the predictor model was able to learn and capture most of the behavior variability present on this time series. Figure 6-8 shows the water inflow forecast during the testing period of 1972-1976. The red line, representing water inflow forecast, shows a behavior very close to the blue one, the desired time series.

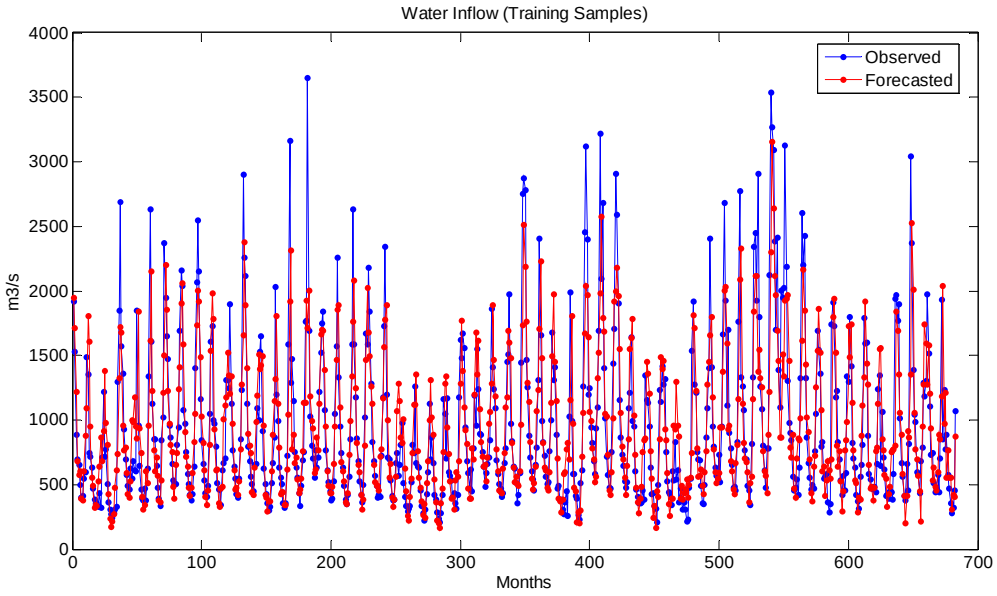


Figure 6-7. Training performance of ESN for the water inflow prediction at Fumas Hydroelectric power plant.

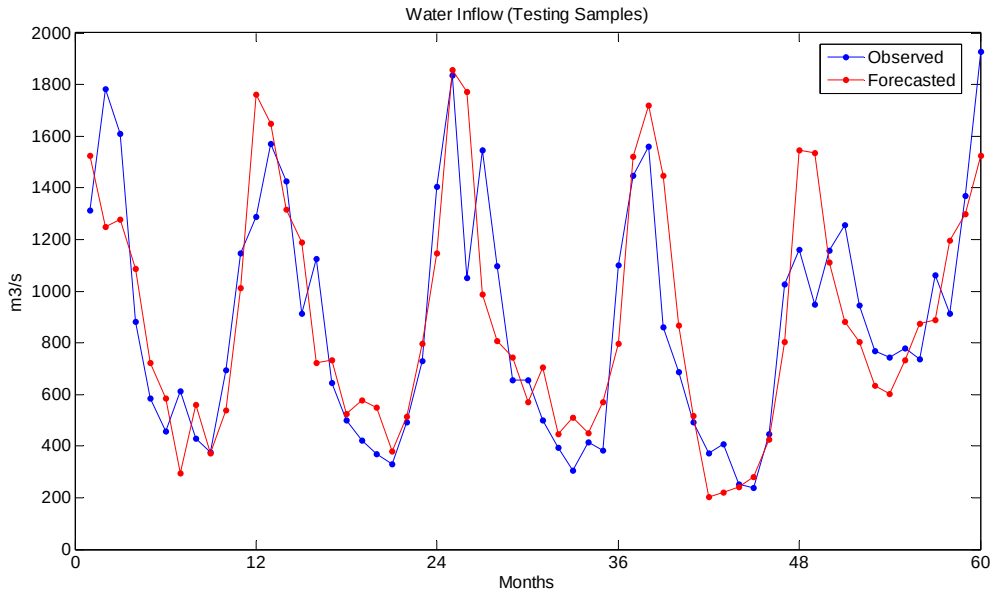


Figure 6-8. Testing performance of the ESN for the water inflow prediction at Fumas Hydroelectric power plant.

Spike Detection with Echo State Networks

In this section, we propose to use ESN-MACE for detecting neural action potentials in neural recording. Detection of action potentials is a very challenging task due to the great variability in the spike shapes due to the difference in spike shapes (both in amplitude and width of the spikes), as well as the intrinsic noise, which is nonstationary and most likely non-Gaussian, collected in neural recordings. The sorting of spikes from single neurons is classically defined as a two-step process. First, the spikes must be detected using either a threshold detector or through power analysis (Obeid and Wolf, 2004). Once the spikes are detected, a variety of algorithms such as template matching with linear matched filter, or Principal Components Analysis (PCA) can be used to isolate the waveforms of individual distinct neurons (Lewicki, 1998; Obeid and Wolf, 2004). However, the performance of this serial process is greatly dependent upon the accuracy of the first step. In threshold detection, the background noise, which can be instantaneously at a high value, is confused with spikes leading to false alarms. In

addition, spikes with low power are hard to detect. The main difficulty with the matched filtering method is the absence of a well-defined template signal, which can represent the spike class, due to great variability in signal shapes. Furthermore, noise statistics in neural recordings is non-Gaussian and nonstationary impairing the optimality of matched filter.

The proposed ESN-MACE aims at addressing the difficulties posed by the spike detection problem by utilizing the high dimensional states of a nonlinear dynamical system to train an associative memory that can represent a class under noise and distortions with a single filter. The advantage of the LAM over the conventional linear readout is that it explicitly utilizes the covariance information of the input training set, unlike linear readouts (Ozturk and Principe, 2007). It also simplifies the training of the ESN read out since there is no need to train for a desired response for the LAM. ESN-MACE is particularly powerful when the problem requires modeling a class of functions since ESN-MACE filter is able to build a single template containing covariance information about the full training set. Such a system will improve the ability to correctly identify neural action potentials because it is able to explicitly utilize the variability seen during the training phase.

Various experiments using in *vivo* neural recordings were used to compare the performance of the ESN-MACE with thresholds and a matched filter detector. The performance metric is the receiver operating characteristics curve (ROC), which is a standard measure for evaluating the performance of a detector by demonstrating the trade-off between probability of detection and probability of false alarms (Helstrom, 1995). The ROC curve for each filter is obtained by varying the threshold value and calculating the detection rate and false alarm rate for each value of the threshold. We demonstrate that the ESN-MACE is able to detect action potentials with lower false alarm rates compared to established methods of spike detection.

Problems in Spike Detection

A neural recording cannot be characterized as a deterministic signal, not only because of the intrinsic noise present in the recording instruments, but also because of the many sources of uncertainty that make a neuron fire in an vivo experimental condition (Sörnmo and Laguna, 2005). The extracellular spikes, recorded from a microelectrode array, have amplitudes ranging from a few microvolts up to a few hundred microvolts. The amplitudes of the spikes, as well their varying shapes, make the process of distinguishing the spikes from instrumentation noise difficult. As such, accurate identification of spikes has been problematic (Lewicki, 1998; Sörnmo and Laguna, 2005). However, the wave shape that is registered in a conventional extracellular neural recording is basically determined by the relative distance and position of the cell body with respect to the microelectrode. We will briefly review below several spike detection algorithms.

Threshold detection

The most common used technique to detect the presence of spike activity is through the use of a threshold. This system decides the existence of a spike whenever the amplitude of the neural recording exceeds a predetermined threshold (Lewicki, 1998). The beauty of this method is that it provides a quick and inexpensive way to identify and discern neural activity.

However, threshold detection is ideally suited only for spikes which are easily discernible from the background noise (Lewicki, 1998). Threshold detection is also prone to missing spikes (false negatives) that are present in a dataset, but do not exceed the established threshold. While it is possible to lower the threshold in order to capture all spikes present, doing so increases the probability of falsely misidentifying noise for an action potential (false positives), thus reducing the overall accuracy of the system.

Matched filter

Another approach to spike detection is through the use of filters, in particular by using a matched filter (MF) (Lewicki, 1998). Matched filters are widely used in the receiving end of the digital communication systems to detect binary signals. Matched filter provide optimum filtering in terms of minimum bit error rates for an additive white Gaussian noise channel (Helstrom, 1995).

The fundamental problem with the matched filter for spike detection is that it assumes that there is a template that represents the spike class. Matched filters were originally developed for radar and communication applications, where the signals of interest were designed by humans, and the difficulty was simply one of discovering them in high noise backgrounds. In fact in these applications the template is known a priori and so it is trivial to design the matched filter. Moreover, it is known that the MF is optimum under very restrictive conditions of white noise backgrounds, uncorrelated with the signal and where the noise can be well characterized by Gaussian statistics (Proakis, 2001).

In neural spike detection, the use of a matched filter is very different, due to the fact that the spike amplitude and shape are variable, the noise is not stationary, and its statistics are most likely not Gaussian. But the determining shortcoming is really the variability in spike shapes. Figure 6-9 shows a very regular spike recording from a single neuron under high signal-to-noise-ratio (SNR) conditions. Figure 6-10 shows a more natural recording of activity collected by a single electrode in the rat motor cortex where several neurons are being recorded. The waveforms are aligned by their maximum slope, and show the normal variability of neural recordings.

The first step in the design of a matched filter for spike detection is the design of the spike template. Each dataset was initially spike sorted using a program designed by Cambridge

Electronic Design called Spike 2 (version 5) in order to generate markers for the instance of time when the spike is first detected, commonly known as "spike times". Using these spike times, each spike was then extracted and sorted according to its variance from the original dataset.

Sorting each spike by its variance allowed us to determine how much variability was present within all spikes, and as such selected a number of spikes that encompassed the variability present in the dataset. Numerous experiments with different datasets showed that we were able to accurately describe the variability of all spikes present by randomly selecting a minimum of 50 spikes. This will be the training set for all the experiments presented in this section. Once the necessary number of spikes has been extracted, an average of all spikes present is generated. In order to ensure uniformity within the template, all spikes were aligned according to their initialization time, as recorded in the dataset's spike times. By doing this, a template is generated that characterizes the spike waveform that is most predominant in the dataset.

This approach is best when the neural spikes detected are easily discernable from the background noise and each spike varies insignificantly from each other. However, when the dataset possesses spikes with high variability, the average proves to be inaccurate, reducing the overall accuracy of the system.

Spike Detection with ESN-MACE

The high dimensional internal states of ESN architecture coupled with the MACE filter creates an excellent medium for dynamical pattern recognition. It has been demonstrated in (Ozturk and Principe, 2007) that ESN-MACE is particularly powerful when the problem requires modeling a class of functions since ESN-MACE filter is able to build a single template containing covariance information about the full training set. In this chapter, we propose to use the ESN-MACE for spike detection problem where the spikes can be considered as a family of signals due to great variability in spike amplitude and width (see Figure 6-10).

An ESN with 28 internal units and a single input unit is used for the simulations in this work. **W** matrix is designed according to the ASE metric presented in chapter 3 and scaled so that its spectral radius is set to 0.9. The entries for $\mathbf{W}^{in}$ are set to -1or 1 with equal probability.

Two distinct neural recordings were used in the experiments. The first one was acquired from a single neuron with high SNR. The second dataset was collected from a rat neural cortex neuron while the animal performed a lever pressing task in order to achieve a water reward. Both datasets were sampled at 24414.1Hz and were pre-filtered between 300–7000Hz. Both datasets were normalized, and spike sorted in order to generate their respective spike-times files.

Fifty representative spike signals were selected as the training data for each one of the neural recordings. Figure 6-9 depicts the selected spike signals for the high SNR neural recording. It is evident from the figure that the spike shapes has a limited variability since they all are generated from a single neuron. On the other hand, the differences in amplitude and width of the depolarization/repolarization phases in the spike shapes in Figure 6-10 clearly shows that the rat dataset has more than one type of neuron present in the dataset.

For each data set, the algorithm described in chapter 4 is used to train the ESN-MACE filter. Fifty representative spike signals (each with length 21) are fed to the ESN and the echo states are calculated using equation 1-1 with zero initial conditions. This creates an image of dimension 28x21, where 28 is the number of PEs and 21 is the signal length. Output correlation peak amplitudes are assigned to be 1.0 for the training data. The MACE filter is synthesized in the frequency domain using equation 4-1 and the corresponding image plane filters are obtained by inverse discrete Fourier transform. For a given test signal, MACE filter output is calculated by correlating the echo states of dimension 28x21 with the filter coefficients at zero lag. MACE

filter output is then compared to a threshold to decide the absence or presence of a spike.

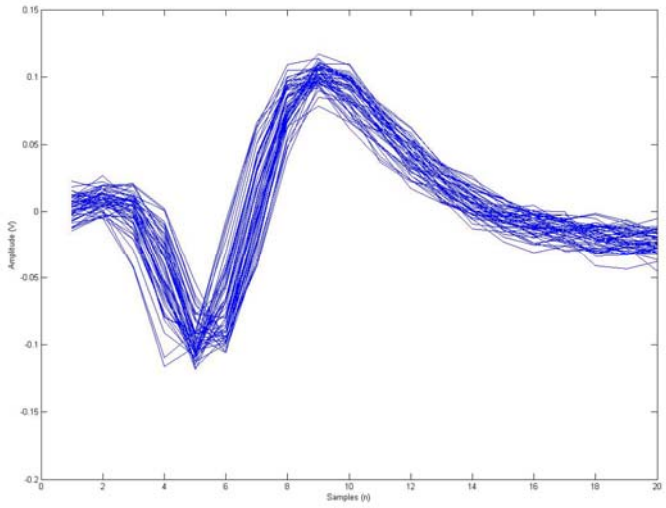


Figure 6-9. Spike shapes recorded with high SNR.

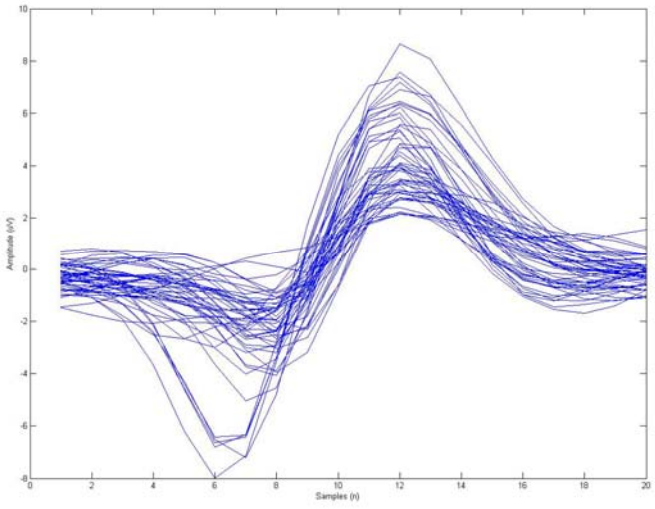


Figure 6-10. Spike shapes in rat dataset. Numerous distinct spike waveforms are present in this dataset.

Different threshold values are used to obtain the ROC figure. Figure 6-11 shows a section of the rat dataset where we used the ESN-MACE filter to identify several spikes. As the ESN-MACE filter is designed to provide a maximum response after a spike has been convolved with

the filter, we shifted our system response by the width of our filter, in order to use the spike times as a means of identifying whether a spike was properly detected, as seen in Figure 6-11.

The results obtained with ESN-MACE are compared to thresholds and matched filter detectors. The matched filter is obtained from the same data set of fifty spike signals according to the method explained in the previous section.

In the first dataset, the ESN-MACE filter, the threshold detector and the matched filter correctly identify 300 spikes present in a 100,000 samples with no missed identifications, demonstrating that the ESN-MACE filter performs as well as both the matched filter system the threshold detector. This was an expected result since the class of spikes present in this dataset is easily discernible from the background noise and come from a single class of spikes. In fact, this is a very simple dataset created by a single neuron with very low noise levels.

On the other hand, the results for the more interesting and more challenging rat data set are not as perfect since the variability in the spike signals are tremendous. Figure 6-12 depicts the ROC curves for the ESN-MACE, threshold detector and the matched filter. As observed from the figure, ESN-MACE gives the best results out of the three methods presented. ESN-MACE filter identifies 273 spikes within 100,000 data samples of the rat dataset, corresponding to approximately 4 seconds of recorded data. In order to correctly identify all 273 spikes, the ESN-MACE filter has a cost of 42 falsely identified spikes; whereas the matched filter and the threshold detector have a cost 262 and 210 falsely identified spikes, respectively, in order to achieve this level of identification. Obviously, the variability in the rat dataset makes detection much more challenging compared to the first dataset. The ESN-MACE filter handles the variability in the spike shapes better compared to the simple threshold detector and the matched filter detector.

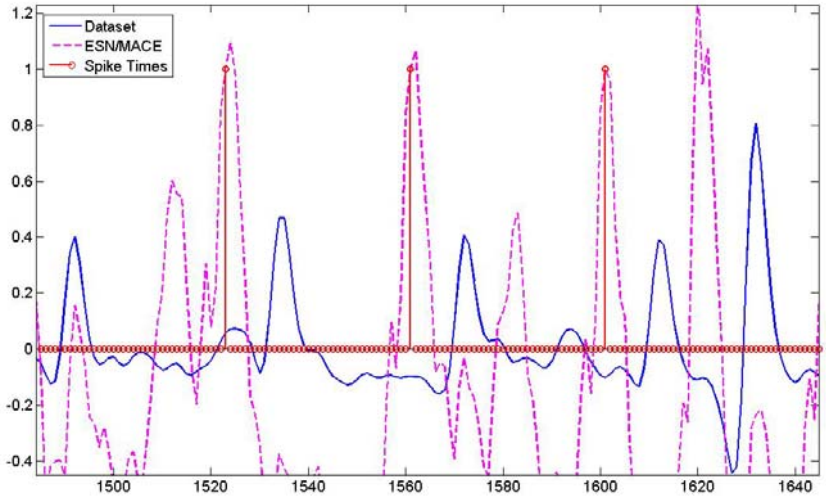


Figure 6-11. Segment of neural recording from rat's brain.

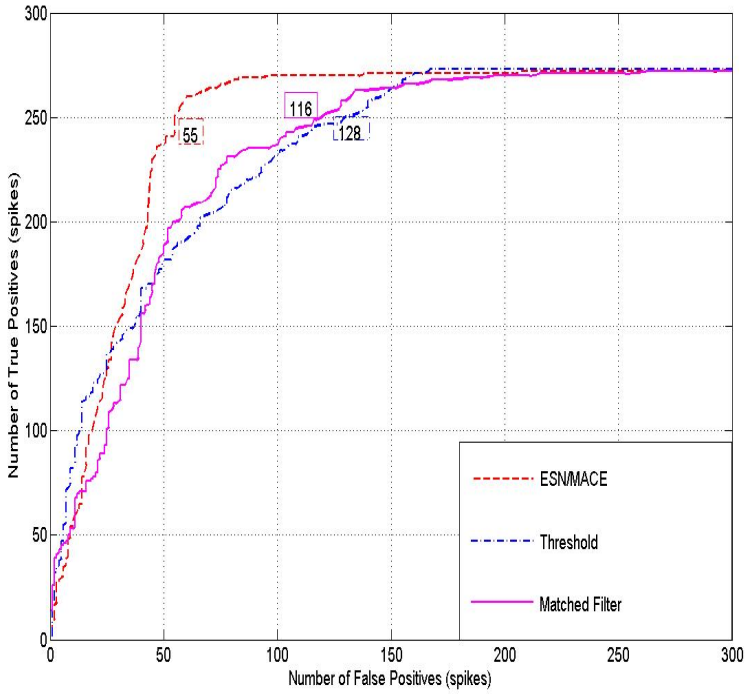


Figure 6-12. Comparison of ROC curves for the rat dataset. Each grid division corresponds to a division of 50 spikes. The leftmost dotted curve corresponds to our ESN-MACE filter system, whereas the solid like corresponds to a matched filter system and the dashed curve corresponds to a threshold detector system.

Robust signal detection in a digital communication channel

The detection of a known waveform transmitted through a channel in a noisy environment is a critical problem in signal processing with applications in areas such as digital communication, radar, pattern recognition and biomedical engineering (Helstrom, 1995; Proakis, 2001). Linear matched filtering is the popular solution to this problem since it is optimal in terms of maximizing signal to noise ratio (SNR) at its output under a linear additive white Gaussian noise (AWGN) channel assumption (Proakis, 2001). However, matched filters lose their optimality if the noise is non-Gaussian or if the channel introduces nonlinear distortions on the transmitted signal. For instance, in the case of an impulsive noise (with infinite variance) in a linear channel, the noise power at the output of matched filter is still infinite, which creates erratic performance due to the finite sample size produced by the filter impulse response.

In mathematical terms, the problem of detecting a known deterministic signal (s(n)) corrupted by a zero mean additive white noise (v(n)) is equivalent to the binary hypothesis testing problem (Helstrom, 1995):

H_0: u(n) = s(n) + v(n) $\quad n = 1,\ldots,T$

H_1: u(n) = v(n) $\quad n = 1,\ldots,T$

The matched filter is defined by the impulse response s(T-n) and the filter output is obtained by convolving the received signal u(n) with the filter impulse response. SNR is maximized if the filter output is sampled at lag T, since there is maximal correlation between s(n) and filter impulse response at this lag. SNR has been defined as the ratio of the total signal energy to the noise variance (Helstrom, 1995):

$$SNR = \frac{\sum_{n=1}^{T} s^2(n)}{\mathrm{var}(v(n))}$$

The output of the matched filter is sampled at the optimal time instant T and compared to a threshold to detect the presence or absence of the signal s(n).

The performance of the matched filter is compared with that of the ESN-MACE and the discriminative TDNN for a linear channel under Gaussian and impulsive noise distributions. The performance metric is the receiver operating characteristics curve (ROC), which is a standard measure for evaluating the performance of a detector by demonstrating the trade-off between probability of detection and probability of false alarms (Helstrom, 1995; Proakis, 2001). The ROC curve for each filter is obtained by varying the threshold value and calculating the detection rate and false alarm rate for each value of the threshold.

The transmitted signal is chosen to be a Gaussian pulse s(n)=exp(-n^2), where n is varied between -5 to 5 with a time step of 0.25 units. The noise distribution is chosen to be either Gaussian or impulsive. Impulsive noise is generated using a Gaussian mixture model with probability distribution function

$$p(n) = (1-\varepsilon)N(0,\sigma_1^2) + \varepsilon N(0,\sigma_2^2)$$

In our simulations, we used $\varepsilon = 0.15$, $\sigma_2^2 = 50\sigma_1^2$ and $\sigma_1^2 = 0.07$. The ASE-ESN with 30 internal units and a single input unit from section 3.1 is used. The TDNN uses a window of 41 samples (equal to the signal duration), 5 hidden layer PEs and a single output unit.

In a communication system, one of two types of operation is possible. We will consider these two cases separately.

a. Synchronous operation

In synchronous operation, the timing of the received signal and the optimal sampling time are known during testing. Therefore, the frames are already available for further processing.

In order to calculate the MACE filter weights for synchronous operation, 100 realizations of the received signal (s(n) + v(n)) are generated and fed to the ESN one sample at a time. For each realization, the echo states are calculated using Equation 2-1 where the initial state is set to zero. The echo state creates an image of dimension 30x41, where 30 is the number of PEs and 41 is the signal length. Output correlation peak amplitudes are assigned to be 1.0 for the training data. The MACE filter is computed in the frequency domain using Equation 4-1 and the corresponding correlator weights are obtained by inverse discrete Fourier transform.

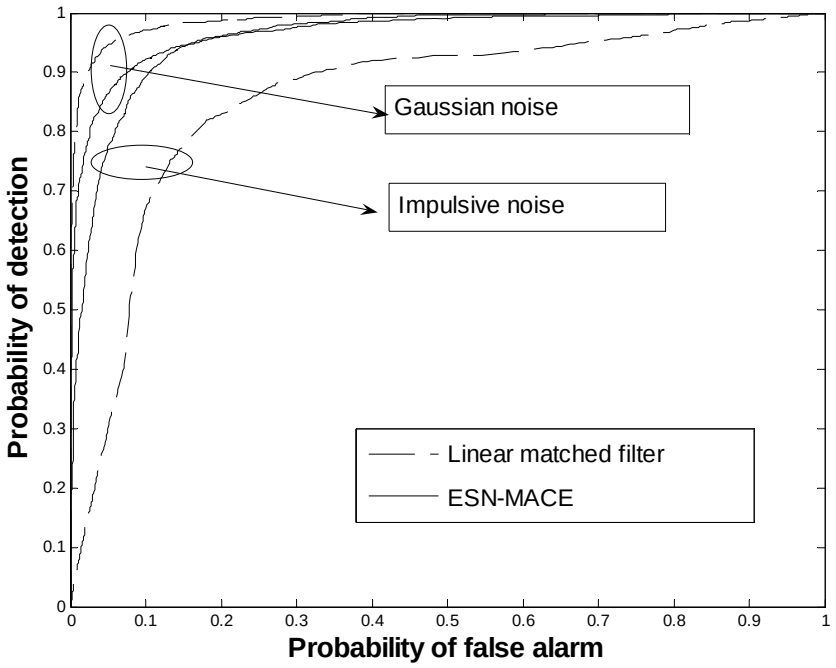
During testing, since the optimal decision time is known, the received signal u(n) corresponding to a transmitted symbol can be processed in a batch. The received signal of length 41 is fed to the ESN with zero initial conditions (as in the training), the echo states are computed, and the decision made once at sample 41. Here, echo states create an image of dimension 30x41. The output of the MACE filter is computed by 2-d correlation of the echo states with the filter coefficients.

The TDNN is used in discriminative mode where the desired target signal is +1 when the received signal is (s(n) + v(n)) and 0 when the received signal is v(n). 500 realizations of the received signal (250 from each class) are generated to train the weights of TDNN using the Levenberg-Marquardt algorithm with a step size of 0.01. A crossvalidation set of 200 realizations is used to stop the training. Notice that we use more data and an extra crossvalidation set for TDNN training compared to ESN-MACE training. During testing, a received signal of length 41 is fed to the TDNN and the output obtained from the TDNN is compared to different threshold values to plot the ROC curve. We used the neural networks toolbox of Matlab© to simulate and train the TDNN.

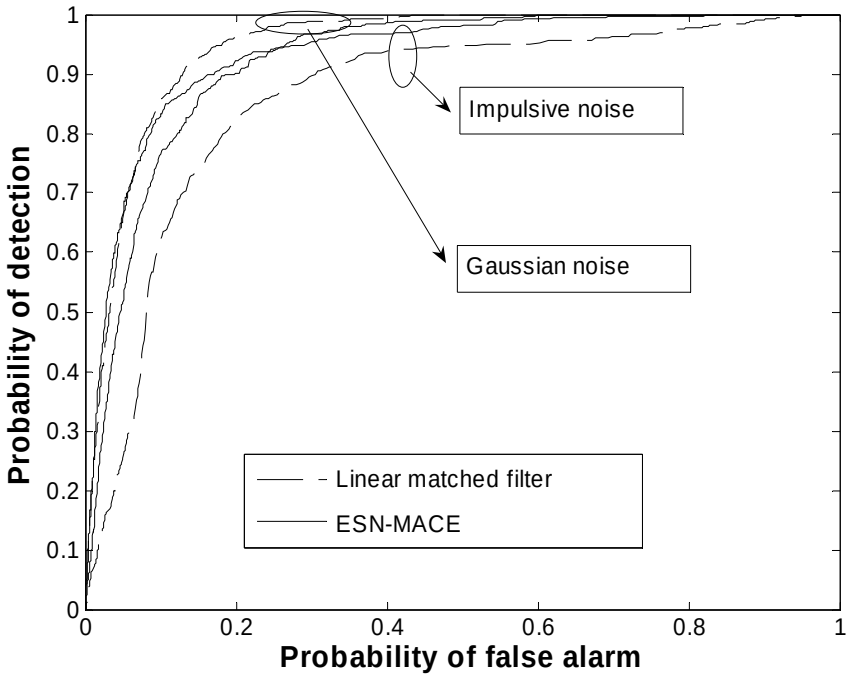
For each noise distribution and a 0.5 probability of signal transmission, 10,000 Monte Carlo simulations are performed to compute theROC curves of the matched filter ESN-MACE and TDNN. Figure 6-13 A shows the ROC curves under Gaussian and impulsive noise with SNR value set to 10dB for synchronous operation. Under Gaussian noise, the matched filter is the optimal detection filter and hence results in the optimal ROC. However, the ROC curve of the ESN-MACE and the TDNN are still close to that of the matched filter. The performance of the matched filter degrades tremendously under impulsive noise whereas the ESN-MACE performance is robust to impulsive noise. The TDNN performance under impulsive noise is better than the matched filter but worse than the ESN-MACE even though TDNN requires more data samples and a cross validation data set for training. If the training of the TDNN is done without a cross-validation set and stopping is done after 50 epochs, the generalization performance suffers tremendously for both noise distributions. We have also tested a TDNN trained in predictive mode, where the desired target signal is the next value of the input. The results are very poor mostly due to the noise level involved in the experiment.

b. Asynchronous operation

In asynchronous operation, the timing of the received signal (and the optimal sampling time) is unknown during testing. Therefore, in asynchronous operation, the frames should be constructed for time n by extracting the echo states from time n-40 to n. Notice that the echo states from time n-40 to n will also be affected from the value of the state at time n-40 (initial state). In synchronous operation, we did not have this problem since the timing of the received signal is known and the frames can be processed separately using zero initial state.



A



B

Figure 6-13. Comparison of ROC curves for ESN-MACE and the linear matched filter under Gaussian and impulsive noise conditions. A) ROC curves for synchronous operation. B) ROC curves for asynchronous operation.

To be able to mimic the testing conditions of asynchronous operation while training the MACE filter, the initial conditions of the ESN are not set to zero but determined by the history of the input. 100 different realizations of the received signal are generated and inserted into the noise signal of 10000 samples at random time instants. The resulting signal is fed to the ESN and echo states are calculated. The echo states during the application of the received signal are extracted. Echo states create an image of dimension 30x41, where 30 is the number of PEs and 41 is the signal length. Output correlation peak amplitudes are assigned to be 1.0 for the training data. The MACE filter is computed in the frequency domain using Equation 4-1 and the corresponding correlator weights are obtained by inverse discrete Fourier transform.

During testing, the received signal is fed to the ESN and the echo states are computed. Then, at every time instant n, the echo states from time n-40 to time n are fed to the MACE filter and its 2-d correlation with the MACE is computed. The correlation result is compared to various threshold values to obtain the ROC curve.

We also trained a TDNN with the specifications from section 3.1.a. We created a data set of 500 realizations of the received signal (250 for each class). The generated data is used to train the weights of TDNN using the Levenberg-Marquardt algorithm with a step size of 0.01. A crossvalidation set of 200 realizations is used to stop the training. The TDNN is trained in the discriminative mode with the same target signal from section 3.2.a. During testing, the received signal is fed to the TDNN. At every time instant n, the input signal from time n-40 to time n is used to produce the TDNN output which is then compared to various threshold values to obtain the ROC curve.

Figure 6-13 B shows the ROC curves for the matched filter, the ESN-MACE and the TDNN under Gaussian and impulsive noise with SNR value set to 10dB. Similar to the

asynchronous case, the matched filter results in the superior ROC curve for Gaussian noise whereas the ESN-MACE and TDNN performance are still robust to impulsive noise. ESN-MACE performs the best under impulsive noise similar to the synchronous operation. The performance of all filters in asynchronous operation degrades slightly compared to synchronous operation since neither the timing of the received signal is available in asynchronous operation, nor the initial condition.

CHAPTER 7
CONCLUSIONS

The great appeal of echo state networks (ESN) is their ability to construct arbitrary mappings of signals with rich and time varying temporal structures without requiring adaptation of the free parameters of the recurrent layer. The echo state condition allows the recurrent connections to be fixed and only a linear output layer needs to be trained. In a way, the difficulty associated with RNN training is avoided yet preserving the well-known power of recurrent topology. However, fixing the recurrent connections imposes important constraints in the overall architecture that have not yet been fully studied. In this dissertation, we addressed some of the issues with the state of the art by proposing a signal processing framework to understand the operation of ESNs, a system-theoretic approach to quantify ESN dynamics and a metric to quantify the richness of reservoir for computation. The proposed framework led to a systematic design procedure for the fixed reservoir weights of ESN. We further investigated the use of ESNs for temporal pattern recognition and compare several approaches to tackle the problem. A novel associative memory readout adopted from image processing literature, called MACE filter, was proposed for ESNs to be used in pattern recognition problems. ESNs with the linear readout were utilized to solve various real-world problems such as brain machine interface design, prediction of water inflow, and channel equalization. ESN combined with the MACE readout led to a detector for action potentials in neural recordings that outperform the standard techniques such as threshold detector and the matched filter. A nonlinear matched filter that can detect signals in a digital communication channel under different noise conditions was also proposed.

The ESN idea has implications on the understanding and design of other architectures. We investigated the use of readouts for Freeman model of the olfactory cortex. With the proposed readouts, we showed that Freeman Model can process continuous valued, time varying signals.

In addition, we investigated the effect of echo state condition for computation. By allowing the spectral radius to be slightly greater than 1, we showed that a system without a global stability constraint can still perform useful computation. The detailed conclusions for each of the topics are given below.

Understanding Echo State Networks

In this dissertation, we first proposed a framework for ESNs, which is a signal processing interpretation of bases and projections in functional spaces to describe and understand the ESN architecture. According to this interpretation, echo states implement a set of functional bases formed by fixed nonlinear combinations of the input. The linear readout at the output stage simply computes the projection of desired output space onto this representation space. We further introduced an information-theoretic criterion, average state entropy (ASE), for better understanding and evaluating the capability of a given ESN to construct such a representation layer. The entropy of the distribution of the echo states at each time step is calculated and averaged over time to obtain the ASE measure. ASE quantifies the volume of the state manifold. As such this volume should be the largest to achieve the smallest correlation among the bases and be able to cope with arbitrary mappings.

We also interpreted the ESN dynamics as coupled linear systems obtained from the linearization of the ESN nonlinear PE in a small local neighborhood of the current state. This system-theoretic approach allowed the visualization of movement of the system dynamics in response to the input signal. A single ESN with fixed parameters implements a combination of many linear systems with varying pole locations, hence many different time constants that modulate the richness of the reservoir of dynamics as a function of input amplitude. Higher-amplitude portions of the signal tend to saturate the nonlinear function and cause the poles to shrink toward the origin of the z-plane (decreases the spectral radius), which results in a system

with a large stability margin. When the input is close to zero, the poles of the linearized ESN are close to the maximal spectral radius chosen, decreasing the stability margin. When compared to their linear counterpart, an ESN with the same number of states results in a detailed coverage of the *z*-plane dynamics, which illustrates the power of nonlinear systems. Moreover, the spectral radius of an ESN can be adjusted using a constant bias signal at the ESN input without changing the recurrent connection matrix,**W**. The application of a nonzero constant bias moves the operating point to regions of the sigmoid function closer to saturation and always decrease the spectral radius due to the shape of the nonlinearity.

Designing Echo State Networks

We proposed a systematic design procedure for ESNs. When the desired response is not accessible for the design of the ESN bases or when the same reservoir is to be used for a number of problems, the default strategy should be to maximize the ASE of the state vector. However, not all function approximation problems require the same memory depth, which is coupled to the spectral radius. The effective spectral radius of an ESN can be optimized for the given problem with the help of an external bias signal that is adapted using the joint input output space information. The interesting property of this method when applied to ESN built from sigmoidal nonlinearitieis is that it allows the fine tuning of the system dynamics for a given problem with a single external adaptive bias input and without changing internal system parameters. Moreover, the bias can be easily trained either with a line search method or a gradient based method since it is one dimensional. We illustrated experimentally that the design of the ESN using the maximization of ASE with the adaptation of the spectral radius by the bias has provided consistently better performance across tasks that require different memory depths. This means that these two parameters design methodology is preferred to the spectral radius criterion proposed by Jaeger, and it is still easily incorporated in the ESN design.

Computation at the Edge of Chaos

Experiments demonstrated that the ASE for ESN with uniform linearized poles is maximized when the spectral radius of the recurrent weight matrix approaches one (instability). It is interesting to relate this observation with the computational properties found in dynamical systems "at the edge of chaos" (Packard, 1988; Langton, 1990; Mitchell et. al., 1993; Bertschinger and Natschlger, 2004). Langton stated that when cellular automata rules are evolved to perform a complex computation, evolution will tend to select rules with "critical" parameter values, which correlate with a phase transition between ordered and chaotic regimes. Recently, similar conclusions were suggested for liquid state machines (LSMs) (Bertschinger and Natschlger, 2004). Langton's interpretation of edge of chaos was questioned by Mitchell *et al* (Mitchell et. al., 1993). Here, we provided a system theoretic view and explain the computational behavior with the diversity of dynamics achieved with linearizations that have poles close to the unit circle, exactly at the "edge of chaos". According to our results, the spectral radius of the optimal ESN in function approximation is problem dependent and in general it is impossible to forecast the computational performance as the system approaches instability (the spectral radius of the recurrent weight matrix approaches one). However, allowing the system to modulate the spectral radius either by the output or internal biasing may allow a system at the edge of chaos to solve various problems requiring different spectral radii.

Transiently Stable Computation

We introduced a new computational mode, "transiently stable computation", that can be observed in nonlinear systems. Linear system theory shows clearly that stability is necessary to obtain useful responses from linear systems. However, this argument does not necessarily apply to nonlinear systems as observed in transiently stable computation. A nonlinear system can be unstable and still be able to have dynamics that are controlled by the input, when it is applied.

We showed that transiently stable systems can still compute input output mappings. From a computation point of view, this regime is interesting because it toggles between internal dynamics and external dynamics. This makes transiently stable computation biologically more realistic compared to conventional echo states, since biological systems do not have fixed point dynamics but rather wide variety of collective behavior in the form of oscillations and even chaos.

Echo State Networks for Temporal Pattern Recognition

We argued that the high dimensional internal states of ESN/LSM architecture create an excellent medium for dynamical pattern recognition. We approached the problem without the use of a desired label as conventionally done, instead by designing a readout for each class based on a specially trained architecture, called the linear associative memory (LAM). Unlike computer memory, LAMs store information globally in system weights and access memory by its content. Among the family of LAMs, we employed the minimum average correlation energy (MACE) filter which is widely used in the recognition of objects in 2-D images due to its much higher rejection to outliers when compared with conventional LAM training. The MACE interprets the states of the ESN/LSM as a two dimensional "image", one dimension being time and the other space. An optimal template image for each class can be analytically computed using training data. During testing, ESN states are correlated with each template image and the class label of the template with the highest correlation is assigned to the input pattern. A single MACE readout is able to represent a *class* of temporal signals because it incorporates explicitly the covariance of the training data class. If the conventional linear readout is utilized it can only represent the mean of the posterior density of the class given the data as is well known both in classification (Bishop, 1995) or in regression/prediction (Haykin, 2001). This was clearly demonstrated in

classification of the triangular versus square waves of different amplitudes (i.e. a class of signals).

The ESN-MACE combination is a nonlinear template matcher that is posed to replace the linear matched filter in digital communication systems with a more robust performance under different noise distributions. The detection performance of the linear matched filter, which is optimal under Gaussian noise, degrades tremendously under impulsive noise, whereas the ESN-MACE filter provided consistent results under both Gaussian and impulsive noise. We applied the ESN-MACE filter for the classification of odors from an electronic nose and compared to the MACE filter trained on the input space. ESN increases the margin between the in-class and out-of-class outputs of the MACE filter due to the user-defined high dimensionality provided by the ESN reservoir.

We also provided a preliminary experiment that extends the proposed MACE filter readout for LSMs. The MACE readout does not require the conversion of the liquid state outputs (spike trains) into continuous valued signals by filtering. Instead, the value of the liquid state at each time instant is interpreted by the MACE as 1 or 0 depending on the existence or absence of a spike. The liquid state is usually very sparse leading to singular matrices used in the MACE filter computation. Therefore, proper regularization of the singular matrices should be done before computing the LSM/MACE filter weights in order to avoid numerical errors. Detailed studies have to be performed to better understand the advantages/disadvantages of a MACE readout for LSMs.

Channel Equalization with Complex Echo State Networks

We proposed ESN for signal processing with complex numbers. ESNs are very suitable for complex domain signal processing since training is linear regression, which is very simple both in real and complex domains. ESNs do not require backpropagation of weights through complex

nonlinear functions, which is problematic in the complex domain since it requires the estimation of complex gradients. We compared the complex ESN with the complex least mean squares and complex multilayer perceptron in a complex domain channel equalization problem and achieved superior results in terms of bit error rates.

Freeman Model

We demonstrated that FM can be considered in the same framework with ESN/LSM. In fact, the KI and KII networks of FM are conceptually similar to the reservoir/liquid in an ESN/LSM with a proper selection of the parameters. The big difference is that the KII layer is highly structured with a coupled set of nonlinear oscillators We derived conditions on the system parameters for the FM similar to the echo state property of ESNs. With the proposed framework, it became evident that the OB layer alone lacks a readout in order to be used as a universal computing machine for time series. Therefore, we proposed to use an adaptive linear network to implement the readout from the OB layer. We presented experimental results to show the power of this framework for the FM of the olfactory cortex. With the proposed framework, we showed that it is not necessary to drive Freeman's model with 0 and 1s, as is conventionally done.

Applications

We have also utilized ESNs for various real-world engineering problems.

Brain Machine Interfaces

We used ESNs to model brain machine interfaces. In BMIs, the goal is to predict a kinematic variable such as the hand position from electrophysiological recording that are acquired through microarrays from the brain. Mesoscopic level ECoG signals and simultaneous hand position of a human patient are used to build the models. We compared the performances of a Wiener filter, an ESN and a leaky ESN in terms of modeling accuracy using the correlation coefficient metric. Both ESN and leaky ESN achieved superior performance compared to the

Wiener solution since the task requires nonlinear modeling. The leaky ESN achieved the best performance since the input signals change rapidly whereas the hand position is at a much slower rate. In problems where the desired signal is at a slower rate than the input signal, the built-in low pass filter in the leaky ESN is very useful to match the signal rates.

Prediction of Water Inflow

We utilized ESNs to forecast the water inflow of a hydropower plant reservoir. Water inflow information is fundamental to planning of the hydrothermal power system operation. A database of average monthly water inflows of Furnas plant, one of the Brazilian hydropower plants, was used as source of training and test data.The results of the ESN forecasting were compared with other nonlinear models We have shown that the ESN performed significantly better than the other nonlinear models with the advantage of significantly simpler, and faster, training algorithm.

Spike Detection

Spike detection is a challenging task due to the variability in the neural recordings and nonstationary, non-Gaussian noise embedded in the signal. The conventional techniques such as threshold detector or matched filter fail especially when the neural signal comprises recordings from multiple neurons spaced at different distances relative to the microelectrode. We proposed the ESN-MACE filter to tackle the variability in the spike detection problem. We showed in a challenging neural data recorded from a rat's brain that ESN-MACE combination results in lower false alarm rates for perfect detection. The better results of the ESN-MACE is attributed to the fact that MACE filter can model a class of signals by incorporating the covariance information of the class whereas ESN provides a rich representation space as the input to the MACE.

Matched Filtering

The ESN-MACE combination is a nonlinear template matcher that is posed to replace the linear matched filter in digital communication systems with a more robust performance under different noise distributions. The detection performance of the linear matched filter, which is optimal under Gaussian noise, degrades tremendously under impulsive noise, whereas the ESN-MACE filter provided consistent results under both Gaussian and impulsive noise.

Possible Future Directions

Recurrent neural networks train all the weights in the network using algorithms such as backpropogation through time , real time recurrent learning whereas echo state networks train only the readout network weights. A possibility in the recurrent network learning paradigm is to find an intermediate step where the internal connection weights can be cleverly initialized with a simple algorithm. This initialization algorithm finds a good choice of weights for the internal connection weight matrix using both the input and desired response and optimally computes the output matrix. This intermediate network will address the big drawback in the state-of-the-art of the ESN design that ignores the desired response in the design of internal weight matrix.

A better understanding of the effect of dynamics in terms of computational power has to be established. We have shown that a nonlinear system without global stability constraint can still do useful computation in terms of functional approximation. Similarly, KII network of the Freeman Model can be considered in the same framework. However, we have not yet observed a concrete advantage of this regime compared to conventional fully stable regime. A thorough study that links stability with computational power could be very interesting both from an engineering and biological systems perspective.

The use of multiple linear readouts in the spirit of local linear modeling can considerably improve the performance of ESNs. In our opinion, the main difficulty associated with this

approach would be to divide the input space since the signals in consideration are not static but time-series. A practical solution to this problem either by embedding the input time series or yet better using the instantaneous value of the echo states has to be developed.

LIST OF REFERENCES

Amari, S. I. (1990). *Differential-geometrical methods in Statistics*. New York: Springer.

Atiya, A. F., El-Shoura, S. M., Shaheen, S. I., & El-Sherif, M. S. (1999). A comparison between neural-network forecasting techniques-case study: River flow forecasting. *IEEE Transactions on Neural Networks.* 10, 402–409.

Barreto, G. A., & Araújo, A. F. R. (2001). A self-organizing NARX network and its application to prediction of chaotic time series. In *Proc. of International Joint Conference on Neural Networks* (pp. 2144-2149). Washington D.C., USA.

Benedetto, S., & Biglieri, E. (1983). Nonlinear Equalization of Digital Satellite Channels. *IEEE Journal on Selected Areas in Communication,* 1, 57-62.

Bertschinger, N., & Natschlger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7), 1413–1436.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Box, G. E. P., Jenkins, G., & Reinsel, G. C. (1994).*Time Series Analysis, Forecasting, and Control*. California: Prentice-Hall.

Braitenberg, V. (1990). Reading the structures of brains. *Network: Computation in Neural Systems,* 1(1), 1-12.

Casasent, D., & Ravichandran, G. (1992). Advanced distortion-invariant minimum average correlation energy (MACE) filters. *Applied Optics,* 31 (8), 1109-1116.

Cha, S. I., Kassam A. (1995). Channel equalization using adaptive complex radial basis functionnetworks. *IEEE Journal on Selected Areas in Communication*, 13(1), 122-131.

Chapin, J.K., Moxon, K.A., Markowitz, R.S., & Nicolelis, M.A.L. (1999).Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience*, 2(7), 664-670.

Chua L., Yang L. (1988). Cellular neural networks: Theory. *IEEE Transaction on Circuits and Systems*, 35, 1257–1272.

Coulibaly. P., & Anctil, F. (1999).Real-Time Short-Term Natural Water Inflows Forecasting Using Recurrent Neural Networks. In *Proc. of International Joint Conference on Neural Networks* (pp. 3802-3805). Washington D.C., USA.

Cox, R. T. (1946). Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14(1), 1–13.

Crone, N.E., Miglioretti, D.L., Gordon, B., & Lesser, R.P. (1998). Functional mapping of human sensorimotor cortex with electrocorticographic spectral analysis: II. Event-related synchronization in the gamma band. *Brain*, 121(12), 2301–15.

de Vries, B. (1991). *Temporal processing with neural networks- the development of the gamma model*. Unpublished doctoral dissertation, University of Florida.

Delgado, A., Kambhampati, C., & Warwick, K. (1995). Dynamic recurrent neural network for system identification and control. *IEEE Proceedings of Control Theory and Applications*, 142(4), 307–314.

Desmurget, M., Pelisson, D., Rosetti, Y., & Prablanc, C. (1998). From Eye to Hand: Planning Goal-Directed Movements. *Neuroscience and Biobehavioral Reviews*, 22, 761-788.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211.

Engel, A.K., Moll, C.K.E., Fried, I., & Ojemann, G.A. (2005). Invasive Recordings from the Human Brain: Clinical Insights and Beyond. *Nature Reviews Neuroscience*, 6(1), 35-47.

Erdogmus, D., Hild, K. E., & Principe, J. (2003). Online entropy manipulation: Stochastic information gradient. *Signal Processing Letters*, 10(8), 242–245.

Erdogmus, D. & Principe, J. (2002). Generalized information potential criterion for adaptive system training. *IEEE Transactions on Neural Networks*, 13(5), 1035–1044.

Fancourt, C., & Principe, J. (2000). On the use of Neural Networks in the Generalized Likelihood Ratio Test for Detecting Abrupt Changes in Signals. In *Proc. of International Joint Conference on Neural Networks* (pp. 243-249). Como, Italy.

Feldkamp, L. A., Prokhorov, D. V., Eagen, C., & Yuan, F. (1998). Enhanced multi-stream kalman filter training for recurrent networks. In J. Suykens, and J. Vandewalle (Eds.), *Nonlinear Modeling: Advanced Black-Box Techniques* (pp. 29–53). Dordrecht, Netherlands: Kluwer.

Fisher, J.W. (1997). *Nonlinear Extensions to the Minimum Average Correlation Energy Filter*, Unpublished doctoral dissertation, University of Florida.

Freeman, W. (1975). *Mass Action in the Nervous System.* New York: Academic.

Freeman, W., & Jakubith, S. (1993). Bifurcation analysis of continuous time dynamics of oscillatory neural networks. In A. Aertsen, W. von Seelen (Eds.), *Brain Theory-Spatio-temporal Aspects of Brain Function* (pp.183-208). Amsterdam, Netherlands: Elsevier.

Georgopoulos, A., Kalaska, J., Caminiti, R., & Massey, J. (1982). On the relations between direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(1), 1527-1537.

Gori, M., & Scarselli, F. (1998). Are multilayer perceptrons adequate for pattern recognition and verification?. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(11), 1121-1132.

Gunduz, A., Ozturk, M. C., Sanchez, J. C., & Principe, J. C. (2007). Echo State Networks for Motor Control of Human EcoG Neuroprosthetics. In *Proc.* of *IEEE EMBS conference on Neural Engineering*. Manuscript accepted for publication.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* (2nd ed.). New Jersey: Prentice Hall.

Haykin, S. (2001). *Adaptive Filter Theory* (4th ed.). New Jersey: Prentice Hall.

Helstrom, C.W. (1995). *Elements of Signal Detection and Estimation*. New Jersey: Prentice Hall.

Hinton, G. E., & Anderson J. A. Ed. (1981). *Parallel Models of Associative Memory*. New Jersey: Lawrence Erlbaum Associates.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Inhyok S. C., & Kassam, A. (1995). Channel equalization using adaptive complex radial basisfunction etworks. *IEEE Journal on Selected Areas in Communication*, 13(1), 122-131.

Iso, K., & Watanabe, T. (1991). Large vocabulary speech recognition using neural prediction model. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (*pp. 57-60). Toronto, CA.

Ito, Y. (1996). Nonlinearity creates linear independence. *Advances in Computer Mathematics*, 5(1), 189–203.

Jaeger, H. (2001). *The echo state approach to analyzing and training recurrent neural networks* (Tech. Rep. No. 148). Bremen: German National Research Center for Information Technology.

Jaeger, H. (2002). *Short term memory in echo state networks* (Tech. Rep. No. 152). Bremen: German National Research Center for Information Technology.

Jaeger, H. & Hass, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667), 78–80.

Jang, J.S. R. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference Systems.*IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665-685.

Jasper, H., & Penfield, W. (1954). *Epilepsy and the Functional Anatomy of the Human Brain*. Boston: Little, Brown and Co.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, A(196), 453–461.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *In Proc. of the Eighth Annual conference of the Cognitive Science Society* (pp. 531-546). Hillsdale, NJ.

Kailath, T. (1980). *Linear systems*. New Jersey: Prentice Hall.

Kautz, W. (1954). Transient synthesis in time domain. *IRE Transactions on Circuit Theory*, 1(3), 29–39.

Kechriotis, G., Zervas, E., & Manolakos, E. S. (1994). Using recurrent neural networks for adaptive communication channel equalization. *IEEE Transactions on Neural Networks*, 5(2), 267–278.

Kim, T., Adali, T. (2002). Fully Complex Multi-Layer Perceptron Network For Nonlinear Signal Processing. *Journal of VLSI Signal Processing*, 32(1), 29-43.

Kremer, S. C. (1995). On the computational power of elman-style recurrent networks. *IEEE Transactions on Neural Networks*, 6(5), 1000–10004.

Kumar, B. (1986). Minimum variance synthetic discriminant functions. *Journal of the Optical Society of America,* A3(10), 1579-1584.

Kung, S. Y. (1993). *Digital Neural Networks*. New Jersey: Prentice Hall.

Kuznetsov, Y., Kuznetsov, L., & Marsden, J. (1998). *Elements of Applied Bifurcation Theory* (2nd ed.). New York: Springer-Verlag.

Lang, K. J., & Hinton, G.E. (1998). *The development of time-delay neural network architectures for speech recognition* (Tech. Rep. No. CMU-CS-88-152), Pittsburg: Carnegie-Mellon University.

Langton, C. G. (1990). Computation at the edge of chaos. *Physica D*, 42(1),12–37.

Lesser, R.P., Gordon, B., Fisher, R.S., Vining, E., & Uematsu, S. (1990). Cortical stimulation using subdural electrodes. *Journal of Epilepsy*, 3(1), 103-106.

Leuthardt, E.C., Schalk, G., Wolpaw, J.R., Ojemann, J.G., & Moran, D. W. (2004). A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering*, 1(1), 63-71.

Lewicki, M.S., (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. *Computation in Neural Systems,* 9(4), 53-78.

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

Mahalanobis, A., Vijaya K. B.V.K., & Casasent, D. (1987). Minimum average correlation energy filters. *Applied Optics* 26(17), 3633-3640.

Markram, H., Wang, Y., & Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, 95(1), 5323-5328.

MecLeod, A. I.(1995). Diagnostic checking of periodic autoregression with Application. *Journal of Time Series Analysis,* 15(2), 221–223.

Mehring, C., Nawrot, M. P., Cardoso de Oliveira, S., Vaadia E., Schulze-Bonhage, A., Aertsen, & A., Ball, T. (2004). Comparing information about arm movement direction in single channels of local and epicortical field potentials from monkey and human motor cortex. *Journal of Neurophsiology*, 98(1), 498-506.

Mitchell, M., Hraber, P., & Crutchfield, J. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7(1), 89–130.

Obeid, I. & Wolf, P. (2004). Evaluation of Spike-Detection Algorithms for a Brain-Machine Interface Application. *IEEE Transactions on Biomedical Engineering,* 51(6), 905-11.

Oh, S.-H. & Lee, Y. (1994). Effect of nonlinear transformations on correlation between weighted sums in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 5(3), 508–510.

Ozturk, M. C., & Principe, J. C. (2007). An Associative Memory Readout for Echo State Networks. *Neural Networks.* Manuscript accepted for publication.

Ozturk, M.C, Principe, J.C., Davis, B., & Erdogmus, D. (2003). Simulation of the Freeman model of the olfactory cortex: a quantitative performance analysis for the DSP approach. In *Proc. of International Joint Conference on Neural Networks* (pp. 332-336). Portland.

Ozturk, M.C, Xu, D., & Principe, J.C. (2004). Modified Freeman model: a stability analysis and application to pattern recognition. In *Proc. of International Joint Conference on Neural Networks* (pp. 3207-3212). Budapest, Hungary.

Ozturk, M.C, Xu, D., & Principe, J.C. (2007). Analysis and Design of Echo State Networks. *Neural Computation*, 19(1), 111-138.

Packard, N. (1988). Adaptation towards the edge of chaos. In J. A. S. Kelso, A. J. Mandell, & M. F. Shlesinger (Eds.), *Dynamic Patterns in Complex Systems* (pp. 293–301). Singapore: World Scientific.

Pfurtscheller, G., Graimann, B., Huggins, J.E., Levine, S.P., & Schuh, L.A. (2003). Spatiotemporal patterns of beta desynchronization and gamma synchronization in corticographic data during self-paced movement, *Clinical Neurophysiology*, 114, 1226-1236.

Pouget, A. & Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience*, 9(2), 222–237.

Principe, J. (2001). Dynamic neural networks and optimal signal processing. In Y. Hu, & J. Hwang (Eds.). *Neural Networks for Signal Processing* (pp. 6–28). Boca Raton: CRC Press.

Principe, J. C., de Vries, B., & deOliviera, P. G. (1993). The gamma filter- a new class of adaptive iir filters with restricted feedback. *IEEE Transactions on Signal Processing*, 41(2), 649–656.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). *Neural and Adaptive Systems: Fundamentals through Simulation*. New York: John Wiley and Sons.

Principe, J., Xu, D., & Fisher, J. (2000). Information theoretic learning. In S. Haykin (Ed.), *Unsupervised Adaptive Filtering* (pp. 265–319). New Jersey: Wiley.

Proakis, J.G. (2001). *Digital Communications* (4th ed.). New York: McGraw-Hill.

Prokhorov, D. (2005). Echo state networks: Appeal and challenges. In *Proc. of International Joint Conference on Neural Networks* (pp. 1463–1466). Montreal, Canada.

Prokhorov, D., Feldkamp, L., & Tyukin, I. (1992). Adaptive behavior with fixed weights in recurrent neural networks: An overview. In *Proc. of International Joint Conference on Neural Networks* (pp. 2018–2022). Honolulu, Hawaii.

Puskorius, G. V. & Feldkamp, L. A. (1994). Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, 5(2), 279–297.

Puskorius, G. V. & Feldkamp, L. A. (1996). Dynamic neural network methods applied to on-vehicle idle speed control. *Proceedings of IEEE*, 84(10), 1407–1420.

Rao, Y., Kim, S., Sanchez, J., Erdogmus, D., Principe, J. C., Carmena, J., Lebedev, M., & Nicolelis, M. (2005). Learning mappings in brain machine interfaces with echo state networks.In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp.233-236). Philadelphia.

Renyi, A. (1970). *Probability Theory*. New York: Elsevier.

Rickert, J., Cardoso de Oliveira, S., Vaadia, E., Aertsen, A., Rotter, S., & Mehring C. (2005). Encoding of movement direction in different frequency ranges of motor cortical local field potentials. *Journal of Neuroscience*, 25, 8815-8824.

Sacchi, R., Carneiro. A. A. F. M., & Araújo, A. F. R. (2004). An RBF network trained by the SONARX model and applied to obtain the operation policies of the hydropower systems. In *8th Brazilian Symposium on Artificial Neural Networks*. Sao Luis, Brazil.

Sacchi, R., Ozturk, M. C., Principe, J. C., Carneiro, A. A. F. M., & da Silva, I. N. (2007). Water Inflow Forecasting using the Echo State Network: a Brazilian Case Study. In *Proc. of IEEE International Joint Conference on Neural Networks.* Manuscript accepted for publication.

Salas, J. D. Boes, D. C., & Smith R. A. (1982). Estimation for arma models with seasonal parameters. *Water Resources Research*, 18(4), 1006–1010.

Sanchez, J. C. (2004). *From Cortical Neural Spike Trains to Behavior: Modeling and Analysis*. Unpublished doctoral dissertation, University of Florida.

Sanchez, J.C., Carney, P.R., & Principe, J.C. (2006). Analysis of Amplitude Modulated Control Features for ECoG Neuroprosthetics. In *Proceedings of the 28th IEEE EMBS Annual International Conference* (pp. 5468-5471). New York.

Sanchez, J.C., Gunduz, A., Principe, J.C., & Carney, P.R. (2007). Extraction and Localization of Mesoscopic Motor Control Signals for Human ECoG Neuroprosthetics. *Journal of Neuroscience Methods*. Manuscript submitted for publication.

Sandberg, I. W. & Xu, L. (1997). Uniform approximation of multidimensional myopic maps. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(6), 477–500.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel Networks that learn to pronounce English text. *Complex Systems*, 1(1):, 145-168

Serruya, M.D., Hatsopoulus, N.G., Paninski, L., Fellows, M.R. & Donoghue, J.P. (2002). Instant neural control of a movement signal. *Nature*, 416, 141-142.

Shah, J. V. & Poon, C.-S. (1999). Linear independence of internal representations in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 10(1), 10–18.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Techinical Journal*, 27, 623–656.

Siegelmann, H. T. (1993). *Foundations of Recurrent Neural Networks*. Unpublished doctoral dissertation, Rutgers University.

Siegelmann, H. T. & Sontag, E. (1991). Turing computability with neural nets. *Applied Mathematics Letters*, 4(6), 77–80.

Sinai, A., Bowers, C.W., Crainiceanu, C.M., Boatman, D., Gordon, B., Lesser, R.P., Lenz, F.A., & Crone, N.E.(2005). Electrocorticographic high gamma activity versus electrical cortical stimulation mapping of naming. *Brain*, 128, 1556-1570.

Singhal, S. & Wu, L. (1989). Training multilayer perceptrons with the extended Kalman algorithm. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 1* (pp. 133–140). California: Morgan Kaufmann.

Sörnmo, L., & Laguna, P. (2005). *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. London: Elsevier Academic Press.

Takens, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand, & L. S. Young (Eds.), *Dynamical systems and turbulence*, (pp. 366–381). Berlin: Springer.

Tavares V. (2001) *Design and implementation of a biologically realistic olfactory cortex model*. Unpublished doctoral dissertation, University of Florida.

Tebelskis, J. (1995). *Speech recognition using neural networks.* Unpublished doctoral dissertation, Carnegie Mellon Univerity.

Thogula, R. (2003). Information theoretic self-organization of multiple agents. Unpublished master's thesis, University of Florida.

Valença, M., Lurdermir, T., & Valença, A. (2005). River Flow Forecasting for Reservoir management through Neural Networks. In *Proc. of the Fifth International Conference on Hybrid Intelligent Systems* (pp. 545-547). Rio De Janeiro, Brazil.

Vanderlugt, A.B. (1964). Signal Detection by Complex Matched Spatial Filtering. *IEEE Transacactions on Information Theory,* 2, 139-145.

Wang D. (1999). Relaxation oscillators and networks. In J. G. Webster (Ed.), *Wiley encyclopedia of electrical and electronics engineering* (pp. 396-405). New York: Wiley.

Werbos, P. (1990). Backpropagation through time; what it does and how to do it. *Proceedings of IEEE*, 78(10), 1550–1560.

Werbos, P. (1992). Neurocontrol and supervised learning: An overview and evaluation. In D. White, & D. Sofge (Eds.), *Handbook of Intelligent Control* (pp. 65–89). New York: Van Nostrand Reinhold.

Widrow, B., & Lehr, M. A. (1990). 30 years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of IEEE*, 78(9), 1415-1442.

Wilde, D. J. (1964). *Optimum Seeking Methods*. New Jersey: Prentice Hall.

Williams, R. J. & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1, 270–280.

Xu D., & Principe, J. C. (2004). Dynamical Analysis of neural oscillators in an olfactory cortex model. *IEEE Transactions on Neural Networks,* 15, 1053–1062.

Xu, D., Davis, B., Ozturk, M. C., Deng, L., Skowronski, M., Harris, J. G., Freeman, W., & Principe. J.C. (2004). Computational neural networks**.** In B. He, (Ed.), *Neural Engineering* (chapter 16). New York: Kluwer.

Zahalka, A., & Principe, J. (1993) Transient detection with neural networks: the search for the desired signal. In S. J. Hanson, J. D Kowan, C. L. Giles (Eds.), *Advances Neural Information Processing Systems 5* (pp. 688-695). California: Morgan Kaufmann.

BIOGRAPHICAL SKETCH

Mustafa Can Ozturk was born in Antalya, Turkey, on November 23, 1980. He received his B.S.in electrical and electronics engineering and B.S.in mathematics, both from Middle East Technical University, Ankara, Turkey, in 2002. He has been pursuing his graduate studies in electrical and computer engineering department at the University of Florida under the supervision of Dr. Jose Principe. He received his master's degree in May 2004 and Ph.D. degree in May 2007. His research interests broadly include dynamical neural networks, adaptive systems, and signal processing.