NONLINEAR DYNAMIC INVARIANTS FOR

CONTINUOUS SPEECH RECOGNITION

By

Daniel May

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

August 2008

NONLINEAR DYNAMIC INVARIANTS FOR

CONTINUOUS SPEECH RECOGNITION

By

Daniel May

Approved:

Joseph Picone
Professor of Electrical and
Computer Engineering
(Major Advisor and Director of Thesis)

Georgios Lazarou
Adjunct Professor of Electrical and
Computer Engineering
(Committee Member)

Julie Baca
Research Professor, Center for
Advanced Vehicular Systems
(Committee Member)

James E. Fowler
Professor of Electrical and
Computer Engineering
(Interim Graduate Coordinator)

Sarah A. Rajala
Dean of the Bagely College of
Engineering

Name: Daniel May

Date of Degree: August 9, 2008

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor: Dr. Joseph Picone

Title of Study: NONLINEAR DYNAMIC INVARIANTS FOR CONTINUOUS SPEECH RECOGNITION

Pages in Study: 57

Candidate for Degree of Master of Science

In this work, nonlinear acoustic information is combined with traditional linear acoustic information to produce a noise-robust feature set for speech recognition. Classical acoustic modeling has relied on the assumption of linear acoustics where signal processing is performed in the signal's frequency domain. However, the performance of these systems suffers significant degradations when the acoustic data is contaminated with previously unseen noise. The objective of this thesis was to determine whether nonlinear dynamic invariants can boost speech recognition performance when combined with traditional acoustic features. Several experiments evaluate both clean and noisy speech data. The invariants resulted in a maximum relative increase of 11.1% for the clean evaluation set. However, an average relative decrease of 7.6% was observed for the noise-contaminated evaluation sets. The decrease in recognition performance with the use

of dynamic invariants suggests that additional research is required for the filtering of phase spaces constructed from noisy time-series.

DEDICATION

Dedicated to my parents, brother, and sisters for their never-ending support, encouragement, and love.

ACKNOWLEDGEMENTS

I owe a great deal of gratitude to my major advisor, Dr. Joe Picone, who has provided much encouragement, motivation, and support during my years as a student and continues to open doors to many opportunities as I begin my career. I began working with Dr. Picone during my undergraduate freshman year as a web programmer for ISIP. Since then, I have developed a great interest in signal processing and speech recognition. Dr. Picone’s teaching and guidance has been an invaluable part of my college experience.

I would also like to thank all of the past and current students who have been a part of ISIP. Each member of the group has had an impact on my education and has played a part in making my experience with ISIP an enjoyable one. During my early years working with the group, Naveen Parihar and Jon Hamaker were extremely patient with me as I asked them endless series of questions about our technology and software. I am extremely thankful for their guidance.

Finally, I would like to thank my parents, Michael and Marcia May, and my siblings, Greg, Erin, and Gwen. They have been a continuous source of encouragement throughout my life. Their love and support have been essential to my success.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER I

INTRODUCTION

For the past several decades, acoustic modeling for speech recognition has been based on the source-filter model and the assumption of one-dimensional wave propagation in the vocal tract [1]. The signal processing techniques used to parameterize acoustic speech data into features operate primarily in the signal's frequency domain. This approach models the vocal tract as a linear filter and captures the lower-order characteristics of the speech production process. Recent theoretical and experimental evidence has suggested the existence of nonlinear mechanisms in the production of speech [2]. It has also been suggested that the characteristics resulting from these mechanisms, which are called nonlinear dynamic invariants, contain a significant amount of discriminatory information between different types of speech [3]. While the traditional linear representation of speech has shown to be a reasonable means of acoustic modeling, it fails to capture this higher-order, nonlinear acoustic information.

Acoustic modeling techniques that exploit the linear characteristics of speech have dominated the speech recognition community for the past 50 years, and the success of systems that employ these techniques has been well documented [4][5][6]. However, the performance of these systems degrades significantly when exposing these systems to conditions previously unseen in the training data. Furthermore, these techniques fail to represent the underlying nonlinear properties the signal [7]. Since dynamic invariants contain nonlinear information not exploited by linear acoustic characteristics, they can be combined with traditional linear acoustic features to form a much more accurate acoustical representation.

The remainder of this chapter provides a discussion of the nonlinear properties of speech and an overview of nonlinear dynamic invariants, as well as a discussion of some of the recent research involving the use of dynamic invariants for speech recognition. The overall thesis structure and contributions are also discussed.

1.1 Nonlinearity of Speech

The earliest studies of the vocal tract by Helmholtz in the late 1800's suggested that it was a passive, linear acoustic system. Evidence to the contrary of Helmholtz's work was not introduced until the 1960's by Teager and his colleagues [2]. These findings were based on experiments which measured airflow rates in different parts of the mouth during sustained phonation [8]. The experiments revealed characteristics which violated the vocal tract's assumed linear acoustic model. For instance, observations of the air jet, which is the air current expelled by the lungs through the vocal tract, showed that it was unstable, attaching and detaching itself from the vocal tract's walls. This unpredictable, oscillatory behavior changes the cross-sectional areas of the vocal tract resulting in modulations of the air-pressure and velocity fields [9]. This phenomenon is known as airflow separation.

These experiments also measured the location and characteristics of vortices [2][8], which are turbulent, swirling flows of air. The generation and propagation of airflow vortices, which have been experimentally found above the glottis in the vocal tract, can modulate the energy of the air jet. These airflow characteristics cause frequency and amplitude modulations in the speech signal and result in instantaneous variations of frequency and amplitude within the signal's pitch period [9]. Formants, which are the primary frequency components within a speech signal, are also affected by these variations. This evidence suggests that nonlinear mechanisms might be among the primary contributors to the speech production process, and could have a major impact on the signal.

1.2 Nonlinear Dynamic Invariants

The discovery of nonlinear speech production mechanisms in the vocal tract paved the way for new research into estimating and representing the nonlinear characteristics of speech [1][3][10][11][12]. While linear acoustic properties are computed from the signal's frequency domain, nonlinear dynamic properties are computed from the signal’s time domain. The speech production system, or vocal tract, is composed of many different dynamic mechanisms, each of which modifies the signal until the final speech signal exits the vocal tract. Dynamic properties of the speech production system are related to these individual vocal tract mechanisms.

Unfortunately, detailed information about vocal tract mechanisms is not observable during speech production. The only available observable is the final speech signal. However, since the speech signal is dependent on each of the vocal tract mechanisms, the relevant properties of each of these mechanisms are embedded within the signal. It is possible to reconstruct some of this information from the final speech [13][14], and dynamic properties of the system can then be estimated from this reconstructed information.

Nonlinear systems can best be represented by their phase space which defines every possible state of the system [15]. The dimensions of the phase space correspond to the system's dynamic variables, and each point in the space corresponds to a unique state of the system. As the state of the system evolves over time, a path is created within the phase space. This path is called the trajectory of the system. After a long period of time, the system may settle down to a consistent set of states known as the system's attractor. Properties of the system's attractor are able to characterize the most important aspects of the system. These properties include shape [16], amount of chaos [9], and entropy [3].

For a speech signal, the only observable is the sound pressure wave that exits the speech production apparatus – typically the speaker’s mouth or nose depending on the nature of the sound. This is transduced into a one-dimensional electrical signal – voltage as a function of time – using a microphone. We will not discuss the transduction process in this thesis, even though that further distorts and transforms the speech signal.

The phase space is not immediately available. Using phase space reconstruction techniques, the time series can be embedded into a multidimensional phase space which retains the properties of the original phase space [15]. Figure 1 illustrates an example of a reconstructed phase space for the sustained phone /*ah*/ uttered by a single speaker. This figure clearly shows that there is structure within the attractor for this phone, and the structural properties are necessary to classify this attractor. Interestingly, some properties of the dynamic system are invariant between the original and reconstructed phase space.

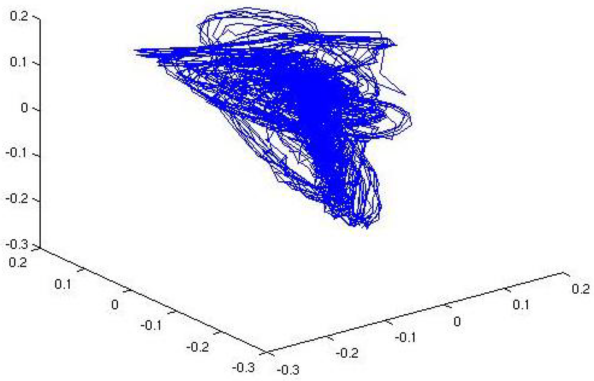

Figure 1. Attractor for phoneme */ah/*

These are the properties which will be most useful since the only available representation of the phase space is the reconstructed version.

Three nonlinear dynamic invariants are explored in this work:

- **Fractal Dimension** [16]: quantifies the geometrical complexity of the attractor;
- **Lyapunov Exponents** [17]: measures the level of chaos in the attractor;
- **Kolmogorov-Sinai Entropy** [3]: measures the average rate of information production in a system.

These invariants are combined with the traditional MFCC features to produce new feature vectors that exploit both the traditional linear properties of the signal and the underlying nonlinear dynamic information.

1.3 Recent Work With Dynamic Invariants

Shortly after evidence was presented that suggested the presence of nonlinear speech production mechanisms in the vocal tract, a significant amount of new research emerged related to nonlinear analysis of speech [3][9][10]. The first of this research focused on bridging the gap between the physical observations mentioned in Section 1.1

and the mathematical aspects of nonlinear dynamic systems. The fractal dimension of speech signals became the primary focus as it was necessary to establish a geometrical description of airflow in the vocal tract. Initial analysis of speech signals revealed that the fractal dimension values for fricative sounds were consistently higher than those computed from vowel sounds [16]. Further experiments showed that fractal dimension could be used to roughly distinguish between unvoiced fricatives, voiced fricatives, and vowels [19].

Soon, researchers investigated the chaotic nature of speech, and nonlinear analysis of speech was broadened to include Lyapunov spectra. Experiments suggested that speech signals had positive Lyapunov exponents indicating that the speech production system was chaotic [1][17]. As the algorithms used to compute Lyapunov spectra were improved, it was shown that Lyapunov exponents could distinguish between different types of phonemes [9]. Kolmogorov entropy, or metric entropy, was also explored as a means of quantifying the level of chaos in speech signals since it is able to determine how much new information is introduced as the attractor evolves. Entropy was also found to be able to distinguish between different phonemes [1][3].

The studies that followed the initial research shifted focus toward the classification of speech signals using dynamic invariants. Many of these studies explored the possibility of using dynamic invariants as features to classify speech segments as phonemes. Those which attempted this arrived at a similar conclusion: dynamic invariants could distinguish between different classes of phonemes, but by themselves, could not always classify phonemes of the same type [1][3][10]. It was then suggested that combining the invariants with traditional linear acoustic features might result in better classification accuracy [20]. The first continuous speech recognition experiments using invariants combined MFCCs with the fractal dimension invariant resulted in improved recognition performance [21]. The work presented in this thesis extends this work and its application to continuous speech recognition problems of scale.

1.4 Thesis Organization and Contribution

The structure of this thesis is outlined below. The current chapter has introduced the primary motivation behind this work. The following chapters cover the theoretical concepts in more detail and discuss the experiments used to explore the application of these concepts to real-world speech recognition systems.

Chapter II discusses the theory behind nonlinear dynamic invariants. This discussion includes the mathematical definition of these invariants, and how they are derived from the reconstructed phase space of the system. This chapter also discusses which characteristics of the nonlinear system the different invariants exploit, and how these characteristics are applicable to speech.

Chapter III discusses the initial set of experiments with nonlinear dynamic invariants which use feature sets consisting of traditional acoustic features combined with nonlinear dynamic invariants to classify signal frames as phonemes. These experiments were designed to provide an idea of what kind of effect the invariants will have on the performance of a speech recognition system.

Chapter IV presents two sets of large vocabulary continuous speech recognition experiments which use the new features. These features are evaluated on the

Aurora 4 Corpus which contains speech recorded under quiet recording conditions mixed with a variety of digitally-added noise. The first set evaluates the data recorded in a clean environment and tests the recognition performance effects of adding the different invariants to the traditional acoustic features. The second set of experiments evaluates performance on noisy speech and tests the invariants' robustness to noisy acoustic environments.

Chapter V summarizes experimental results presented in this thesis and briefly discusses potential future research directions to overcome the limitation of this work.

CHAPTER II

COMPUTING NONLINEAR DYNAMIC INVARIANTS

Unlike traditional acoustic modeling techniques which extract acoustic properties from the speech signal's frequency domain, dynamic invariants are estimated from the signal's time domain. This chapter describes the process of computing dynamic invariants, including reconstruction of the system's phase space, and estimating the individual invariants from this reconstructed phase space.

2.1 Reconstructing the Phase Space

Dynamic systems are best represented by their phase space. The dynamic system's phase space describes the behavior and relationship between the system's dynamic variables as time evolves. Each dimension of the phase space corresponds to one of the system's degrees of freedom, and the solutions of the system as it evolves over time form the system's trajectory. For many systems, the trajectory is drawn to a subset of the phase space after a long period of time. This subset is known as the attractor, and its characteristics are the basis for nonlinear dynamic invariants [1][15].

Dynamic systems can be defined by a set of first-order ordinary differential equations. Given a set of initial conditions, it is possible to numerically evaluate these equations. A discrete-time dynamic system can be defined as:

$$\vec{x}_{n+1} = f(\vec{x}_n), \tag{1}$$

The offset of solutions to this equation can then be plotted in the system's phase space to form the trajectory. For example, the Lorenz [15] system is defined by the following set of equations:

$$\begin{gathered} x_{n+1} = \sigma(y_n - x_n) \\ y_{n+1} = rx_n - y_n - x_n z_n \\ z_{n+1} = -bz_n + x_n y_n \end{gathered} \tag{2}$$

The symbols σ, r and b are the system parameters and define different aspects of the shape of the attractor. These parameters are analogous to different physical characteristics of the vocal tract, such as vocal tract length, cross-sectional area, size of vocal cords, etc. One major difference, however, is that the vocal tract characteristics require much more than three parameters for an accurate model.

Figure 3 shows the resulting Lorenz attractor in the system's phase space after numerically integrating the Lorenz system. Only the x and y components are plotted for visualization simplicity. The attractor of this system is clearly seen as the two spirals between which the trajectory alternates. Figure 2 shows the solutions for the Lorenz system’s x variable, and plotting these solutions versus time further illustrates the bimodal behavior of the system as the solutions alternate between an upper and lower range of values.

The equations in (2) provide a complete definition of the Lorenz dynamic system. In practice, however, this complete description is not accessible [15]. Natural systems, such as speech production, have many unobservable mechanisms [14]. For example, it is not practical to measure the dimensions of a speaker's vocal tract, air flow speeds and pressure near the vocal cords, and to account for all other mechanisms which impact the speech signal as it is being generated. Only the final speech signal is available for observation. However, since the speech signal was modified by each of the mechanisms in the vocal tract, it contains a certain amount of information about them. Before dynamic invariants can be computed, some of this hidden information needs to be estimated by reconstructing the attractor. This is achieved using phase space reconstruction methods. The reconstructed attractor must closely resemble the attractor of the original dynamic system in order for the dynamic invariants to accurately reflect the properties of the system.

The simplest method of embedding is called time-delay embedding. In the

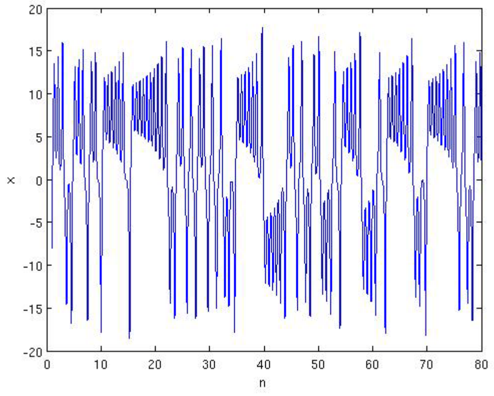

Figure 2. Solutions for a single variable, *x*, of the Lorenz system

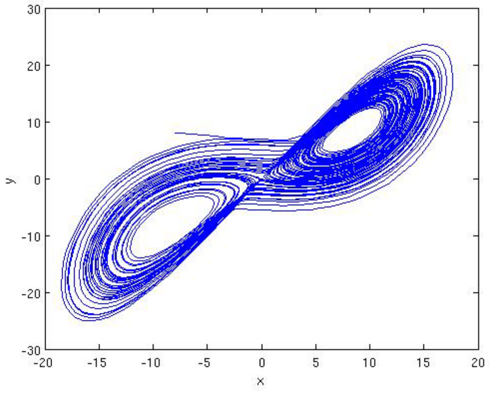

Figure 3. Trajectory plot of *x* and *y* variables of the Lorenz system

reconstructed phase space achieved using this method, the phase space elements are composed of time-lagged versions of the original time-series. Each phase space element is defined as:

$$\bar{s}_n = \left(s_n, s_{n+\tau}, \ldots, s_{n+(m-1)\tau}\right), \tag{3}$$

where τ is the number of delay samples to use, and m is the number of embedding dimensions. In (3), s_n is an observation of one of the system's variables and is defined by:

$$s_n = s\left(x\left(n\Delta t\right)\right), \tag{4}$$

where the observed variable is x. The collection of these elements composes the entire reconstructed phase space, which can be represented in matrix form by:

$$S = \begin{pmatrix} s_0 & s_\tau & \cdots & s_{(m-1)\tau} \\ s_1 & s_{1+\tau} & \cdots & s_{1+(m-1)\tau} \\ s_2 & s_{2+\tau} & \cdots & s_{2+(m-1)\tau} \\ \vdots & & & \vdots \end{pmatrix}. \tag{5}$$

To illustrate this method, solutions to the x variable from the Lorenz system in (2) are used for observations as in (4). Using three embedding dimensions and a time delay

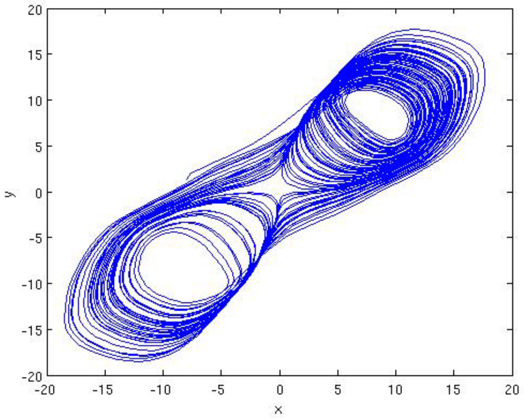

Figure 4. Reconstructed Lorenz attractor from the *x* component

of five samples, the reconstructed phase space in Figure 4 is achieved. Only the *x* and *y* components are plotted for visualization simplicity. The overall, two-loop structure of the original attractor shown in Figure 3 is preserved. More importantly, the nonlinear dynamic invariants computed from this reconstructed attractor will be consistent with those computed from the original attractor, hence the name invariants [15][22].

For effective time-delay embedding, the choices for the time delay length, τ, and number embedding dimensions, *m*, are determined experimentally [23]. The choice for τ is based on the correlation between the original and the time-delayed sets of observations. If the time-delay is chosen too small, there will be a high correlation between the sets, and the resulting attractor will be distorted. On the other hand, if the chosen time-delay is too high, the correlation between the observation sets will be low, which will also result in a distorted reconstructed attractor [15][24].

These two cases are illustrated in Figure 6 and Figure 6 below. In the case where the time delay is too small, the high correlation between the original observation set and

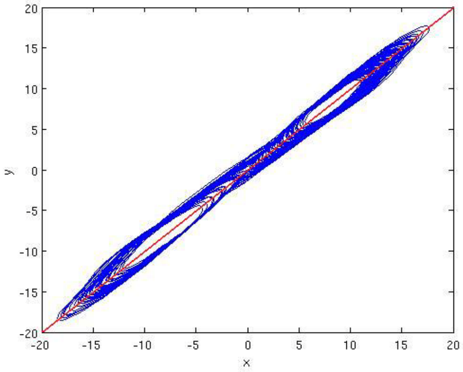

Figure 5. Reconstructed Lorenz attractor where time delay is too small

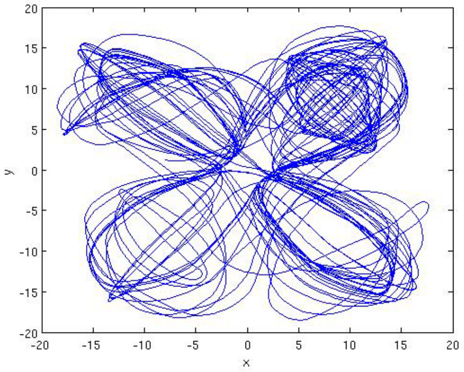

Figure 6. Reconstructed Lorenz attractor where time delay is too large

the time-delayed set causes the reconstructed trajectory to cling to a line described by the equation $x = y$. In the other case, when the time-delay is too large, the low correlation between the different sets causes the reconstructed trajectory to become much more chaotic, distorting the attractor enough that the computed invariant properties will be inaccurate. In general, a good choice for τ is the first zero of the autocorrelation function of the observed samples [15][24].

Choosing the correct number of embedding dimensions requires the detection of false nearest neighbors within the phase space [15]. If m is chosen to be large enough, neighboring states should still remain near to each other as their corresponding trajectories evolve over a short time period. For example, suppose a time series is embedded in m dimensions, and a point $\overline{s}$ is on a given trajectory in the reconstructed phase space as in Figure 7 where $\overline{s}$ is shown in red. The neighbors of $\overline{s}$ are points on neighboring trajectories which fall within a certain radius centered around $\overline{s}$. In Figure 7,

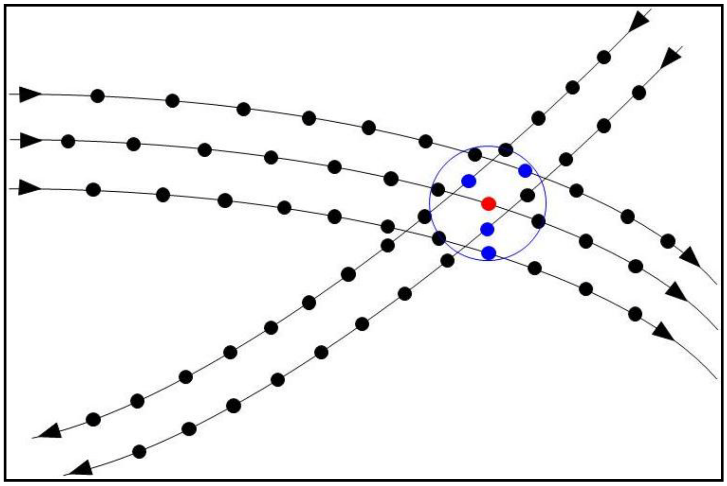

Figure 7. A point $\overline{s}$ in a reconstructed phase space and its nearest neighbors on neighboring trajectories

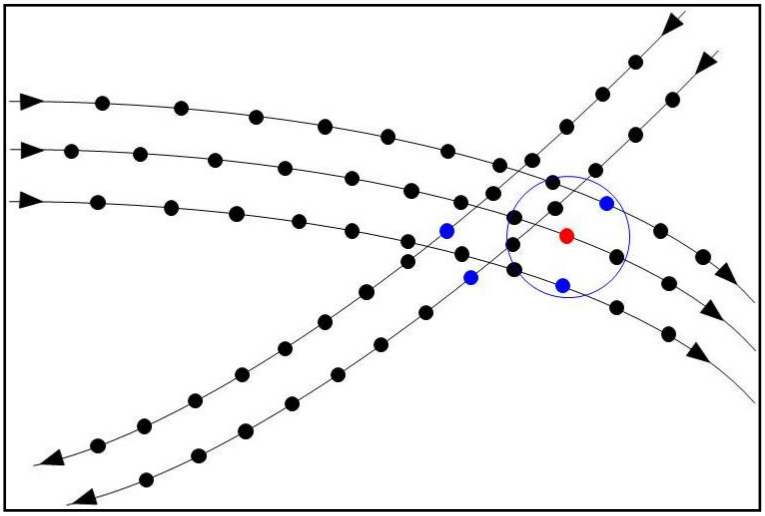

Figure 8. A point $\bar{s}$ in a reconstructed phase space and its neighboring trajectory positions after one time step

this radius is shown as a blue circle, and the nearest neighbors are shown as blue points.

The set of points including $\bar{s}$ and its neighbors should remain relatively close as the trajectories evolve over a single time step. If any neighbors of $\bar{s}$ do not follow this trend, they are labeled false nearest neighbors, as illustrated in Figure 8.

As the trajectories evolve over one time step, two of the points remain close to the original trajectory while two do not. The two that fall outside of the radius after the time evolution are labeled as false nearest neighbors, and this indicates that the chosen number of embedding dimensions is too low. In general, false nearest neighbors are the result of trajectories which appear to be close to each other in *m* dimensions, but are actually far from each other in *m*+1 dimensions. Choosing the correct number of embedding dimensions is a matter of minimizing the number of false nearest neighbors in a reconstructed phase space [15][23].

The choice of *m* is not as constrained as the choice of the time delay. It has been shown that if the chosen value of *m* is larger than necessary, the resulting invariant values will not be negatively affected [15][23]. This seems to suggest that overestimating *m* would be effective. However, the computational complexity of the invariants increases with a higher number of embedding dimensions, so this prevents the use of an excessively large number of embedding dimensions. Additional criteria for choosing values for these parameters exist for specific algorithms and are discussed in subsequent sections.

Time-delay embedding is extremely effective when the observed time series is not contaminated with large amounts of noise [24]. However, this is hardly the case in actual voice applications which often involve noisy ambient environments. A more accurate phase space reconstruction can be achieved using singular value decomposition (SVD) embedding [1][22][24]. The SVD embedding method involves two steps, the first of which is similar to time-delay embedding. For the first step, the original time series is embedded into a high dimensional space with a time delay of one sample. The number of embedding dimensions in this step is referred to as the window size in the context of SVD embedding. The window size is generally chosen to be high in the presence of significant noise. Next, a projection based on the singular vectors of the embedded data is applied to the phase space. The dimensionality is then reduced by identifying components which correspond to noise and removing them.

Both methods accomplish the same task of reconstructing the phase space from a single observed time-series. However, the SVD embedding method results in a smoother estimated attractor than time-delay embedding when reconstructing the phase space from noisy data.

2.2 Lyapunov Exponents

The dynamic behavior of the trajectories within a phase space is an important property of dynamic systems [1][15][17]. Lyapunov exponents are used to quantify this property by describing the relative behavior of neighboring trajectories within an attractor. More specifically, they help determine the level of predictability of the system by analyzing trajectories that are in close proximity to each other, and measuring the change in this proximity as time evolves. The separation between two trajectories with close initial points after *N* evolution steps can be represented by:

$$\Delta x(N) \approx \Delta x(0)\,\frac{d\,(f^{N} x(0))}{dx}, \tag{6}$$

where *f* defines the evolution function of the system. Lyapunov exponents provide a global analysis of this separation behavior between the trajectories within the attractor.

Figure 9 illustrates three basic behaviors that neighboring trajectories may exhibit. A group of trajectories may converge, moving closer together as time evolves. They may

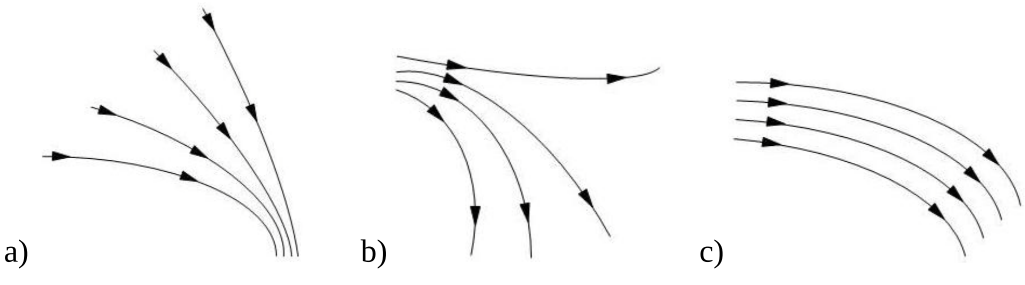

Figure 9. Neighboring trajectories with a) convergent, b) divergent, and c) steady relative behavior.

also diverge, separating from each other over time. They may also neither converge nor diverge, but maintain steady distance between each other in a stable limit cycle [15]. In general, Lyapunov exponents quantify the level of chaos, or sensitivity of the system to initial conditions, within an attractor. A dissipative attractor, completely composed of trajectories which converge to a fixed point will have a negative Lyapunov exponent. An attractor composed of trajectories which both exponentially converge and diverge over time with little predictability will have a positive Lyapunov exponent indicating chaotic behavior, and attractors with trajectories exhibiting stable relative behavior usually have a Lyapunov exponent close to zero.

The computation of a Lyapunov exponent that describes the global chaotic behavior of the attractor requires the averaging of many local behaviors. Trajectories are first examined locally as small subsets of the global attractor, and the behaviors for the local component are averaged to describe the behavior of the attractor as a whole. The following is a high-level description of the algorithm used to compute Lyapunov exponents.

1. Reconstruct phase space from the original time-series data.
2. Select a point $\vec{s}_r$ on the reconstructed attractor.
3. Find a set of nearest neighbors to $\vec{s}_r$.
4. Measure the separation between $\vec{s}_r$ and its neighbors after as time evolves.
5. Compute the local Lyapunov exponent from separation measurements.
6. Repeat 2 though 5 for each $\vec{s}_r$ of the reconstructed attractor.
7. Compute average Lyapunov exponent from local exponents.

Mathematically, the Lyapunov exponent is represented by:

$$\lambda_i = \lim_{n\to\infty} \frac{1}{n} \ln(\mathrm{eig}_{\mathrm{i}} \prod_{\mathrm{p=0}}^{\mathrm{n}} \mathrm{J}(\vec{s})), \tag{7}$$

where J is the Jacobian of the system as the point $\vec{s}$ moves along the attractor. The value *n* is the number evolution steps, and *i* refers to an index in the Lyapunov spectrum which has a number of elements equal to the number of embedding dimensions [3]. Typically, values for all spectral elements are computed, and highest value is chosen to be the Lyapunov exponent [17].

The parameters which must be chosen for this algorithm include the size of the neighborhood, the number of time evolution steps, and the number of embedding dimensions for SVD embedding. For the most part, the number of neighbors should be found experimentally. However, it has been shown that a good starting point is to use 2*m*+1 neighbors where *m* is the number of embedding dimensions. In general, the neighborhood size should be large enough to capture local dynamics around a given point in the phase space, but constrained enough to maintain localization of the dynamics within the neighborhood [15][17].

The choice of the number of evolution steps, *n*, is limited by computation time. Ideally, this value should be very large as seen in (7), but larger values increase computational complexity. Observing the Lyapunov exponents as a function of evolution steps, however, usually indicates that the value of the exponent begins to level off asymptotically for a relatively low number of evolution steps. As with the size of the neighborhood, the optimal value for this parameter should be tuned and chosen experimentally.

The choice of embedding dimension is, again, based on experimentation. It is usually a good idea to choose an initial embedding dimension using techniques described in Section 2.1. By extracting Lyapunov exponents using this value and subsequently increasing values, the exponent should level off asymptotically. As mentioned previously, invariant computations are not adversely affected by an embedding dimension which is too high, but the computation time will increase significantly as the number of embedding dimensions increases.

2.3 Fractal Dimension

Some objects with geometric symmetry exhibit a property called self-similarity. An object is characterized as self-similar if it is composed of smaller versions of itself [15]. A simple example of such an object is a square in a two-dimensional plane, as illustrated in Figure 10. A square can be continuously subdivided into smaller squares where a close-up view of one of the smaller squares appears identical to the original. These special geometrical structures are called fractals.

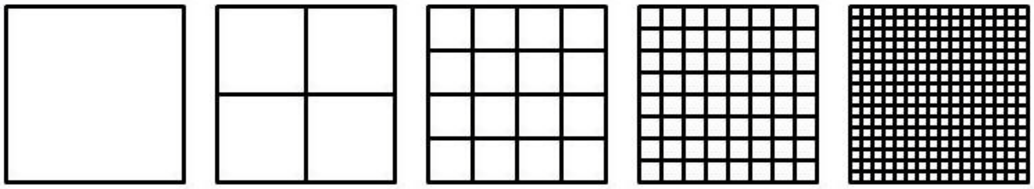

Figure 10. A simple illustration of self-similarity using subdivisions of a square

The dimension of a fractal is used to quantify the degree to which it occupies a space. The term ‘fractal’ comes from the fact that these geometrical structures are not always described as having an integer number of dimensions, but rather a fractional dimension. It is a well known fact that the square in the Figure 10 is a two dimensional structure, but the derivation of this number may be less obvious. The first square in Figure 10 is subdivided into four smaller squares, each of which is a factor of two smaller than the original. In the second square, each smaller square is subdivided into four smaller squares where each of the new squares is, again, smaller than its preceding original by a factor of two. Subsequent divisions of the square follow this trend, and the subdivisions can continue indefinitely. The dimension of this object can be computed by the simple formula:

$$D = \frac{\log M}{\log N}, \tag{8}$$

where *M* is the number of self-similar structures resulting from a division of the original structure, and *N* is the factor of size difference between the original structure and the smaller subdivided structures. For the square, these values are 4 and 2, respectively. Therefore from (8):

$$D = \frac{\log 4}{\log 2} = 2. \tag{9}$$

As mentioned previously, many fractal structures have fractional dimensions. One simple example of such a structure is the Sierpinski triangle which is composed of copies of a simple equilateral triangle. This structure is illustrated in Figure 11. Each subdivision of a

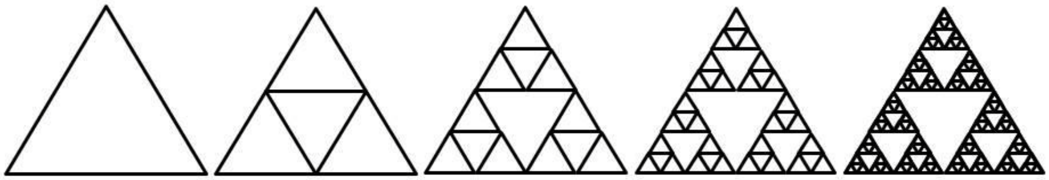

Figure 11. Fractal structure of a Sierpinski triangle for several subdivisions.

triangle results in three triangles, each smaller than the original by a factor of two. From (8), the fractal dimension of the Sierpinski triangle is:

$$D = \frac{\log 3}{\log 2} \approx 1.585. \tag{10}$$

The explanation above illustrates the concept of fractal dimension for geometrical structures with self-similarity, and this dimension is simple to compute when the structures are simple. However, fractal structures observed in nature require more sophisticated calculation techniques since they are much more complex and can be contaminated with noise [16][19][21]. Also, the self-similarity of an object observed in nature is not always immediately apparent. In this thesis, the fractal dimension is estimated from a reconstructed attractor. In the specific case of attractor geometry, this estimated value is called correlation dimension and relies on an important measure of the attractor called the correlation integral [15].

The correlation integral quantifies how completely the attractor fills the phase space by measuring the density of the points close to the attractor's trajectory, and averaging this density over the entire attractor. The correlation integral of a reconstructed attractor is computed using the following steps:

1. Choose a neighborhood radius, ε, and center a hyper-sphere with this radius on the initial point of the attractor.
2. Count the number of points within the hyper-sphere.
3. Move the center of the hyper-sphere to the next point along the trajectory of the attractor and repeat Step 2.
4. Take the average of the number of points falling within the hyper-sphere over the entire attractor.

This average is the attractor's correlation integral. Mathematically, this is expressed by:

$$C(\varepsilon, N) = \frac{2}{(N - n_{\min})*(N - n_{\min} - 1)} \sum_{i=1}^{N} \sum_{j=i+1+n_{\min}}^{N} \Theta(\varepsilon - \left\| \vec{s}_i - \vec{s}_j \right\|), \tag{11}$$

where *ε* is the neighborhood radius and *N* is the number of points composing the attractor.

The step function, Θ, determines the number of points within the neighborhood radius [15]. The $n_{\min}$ parameter is a correction factor proposed by Theiler which reduces the negative effects of temporal correlations by skipping points which are temporally close to the center of the neighborhood [25]. This temporal correlation can result in significantly misleading correlation integral values. The value of this parameter should be large enough to minimize the temporal correlation distortions but small enough to prevent a significant number of points from being skipped in the summation. The neighborhood radius should be chosen small enough to capture only the local space filling properties along the attractor's trajectory, but large enough to ensure the neighborhoods contain a sufficient number of neighbors. Ultimately, both of these parameters should be chosen according to experimentation results.

This correlation integral is used to compute the correlation dimension of the attractor. It is also used to compute the Kolmogorov entropy which will be discussed later in Section 2.4. Computing the correlation dimension can be accomplished by:

$$D(N,\varepsilon) = \lim_{N\to\infty} \lim_{\varepsilon\to 0} \frac{\partial \ln C(\varepsilon, N)}{\partial \ln \varepsilon}, \tag{12}$$

which captures the power-law relation between the correlation integral of the attractor and the neighborhood radius of the hyper-sphere as the number of points on the attractor approaches infinity and ε becomes very small [15].

2.4 Kolmogorov Entropy

Another important measure of dynamic systems is the rate at which new information is being produced as a function of time [1]. Each new observation of a dynamic system potentially contributes new information to this system, and the average quantity of this new information is referred to as the metric, or Kolmogorov entropy [15][26]. For example, a system with an attractor which is limited to a single, periodic attractor would have an entropy of K=0, since the trajectory does not deviate from the limit cycle with each new observation. For complex attractors which exhibit some level of chaos, the entropy is expected to be greater than zero since each new observation contributes a significant amount of information about the system.

For reconstructed phase spaces, it is easier to compute the second-order metric entropy, K_2, because it is related to the correlation integral discussed in Section 2.3. This relation is defined in (11) below:

$$C_m(\varepsilon) \sim \lim_{\substack{\varepsilon \to 0 \\ m \to \infty}} \varepsilon^D \exp(-\tau m K_2), \tag{13}$$

where D is the fractal dimension of the reconstructed attractor, and ε is the neighborhood radius. The parameters m and τ are the number of embedding dimensions and time delay, respectively, used for phase space reconstruction [26]. From this relation, an expression for K_2 can be derived:

$$K_2 \sim \frac{1}{\tau} \lim_{\substack{\varepsilon \to 0 \\ m \to \infty}} \ln \frac{C_m(\varepsilon)}{C_{m+1}(\varepsilon)}. \tag{14}$$

The criteria for choosing values for the parameters ε, m, and τ are the same as discussed in previous sections. The choice of these parameters is also restricted by the resolution of the attractor and the length of the time-series data used to reconstruct it [3].

This chapter has provided a detailed definition and explanation of the nonlinear dynamic invariants which are used in this work. Before they can be used for experiments in this work the various parameters discussed above must be tuned to values that are optimal for speech processing. The next chapter discusses this tuning procedure, as well as the set of pilot experiments used to determine how effective these invariants are at modeling speech.

CHAPTER III

PILOT EXPERIMENTS

Before using dynamic invariants for large-scale, continuous speech recognition, a set of low-level phoneme classifications were run in order to verify the effectiveness of the invariants for modeling speech. The results of these initial experiments also provided some expectations for the results of larger-scale experiments. This chapter begins with an overview of the parameters used for each of the invariant computation algorithms, and the methods used to tune these parameters. An overview of the Wall Street Journal (WSJ0) corpus is also presented. Finally, the experimental setup and classification results are discussed.

3.1 Parameter Tuning

Before using the methods discussed in Chapter II to compute dynamic invariants, the parameter values must be tuned so that the algorithms are effective for speech processing. Tuning is an experimental process in which a variety of parameter configurations are explored for various speech signals, and based on an analysis of the results, an optimal set of parameters is chosen. The parameters used in this work were tuned using a small database of phonemes articulated as isolated words (*i.e.*, one phoneme is spoken per audio segment and sustained for several seconds) recorded from seven different speakers [3]. Though this type of data is not a good representation of the continuous speech recognition problem, it is useful to gain some insight into some basic nonlinear modeling issues. We refer to this data as the Sustained Phoneme Corpus (SPC).

The set of phonemes selected for this database provide a coverage of the major sustainable phoneme classes, including vowels (/aa/, /ae/, /eh/), nasals (/m/, /n/), and fricatives (/f/, /sh/, /z/). Figure 12 illustrates a reconstructed attractor for each of these phoneme utterances. For visualization purposes, each phoneme in the figure is time-delay embedded in two dimensions using a time delay of $\tau = 10$.

The embedding of the vowels produce reconstructed attractors for which the periodic nature is clearly visible in the overall loop structure. Neighboring trajectories within these attractors tend to flow together in a relatively stable manor, indicating that the Lyapunov exponents computed from these attractors will be in the lower range. The attractors for /ah/ and /ae/ are nearly symmetrical, demonstrating the self-similarity

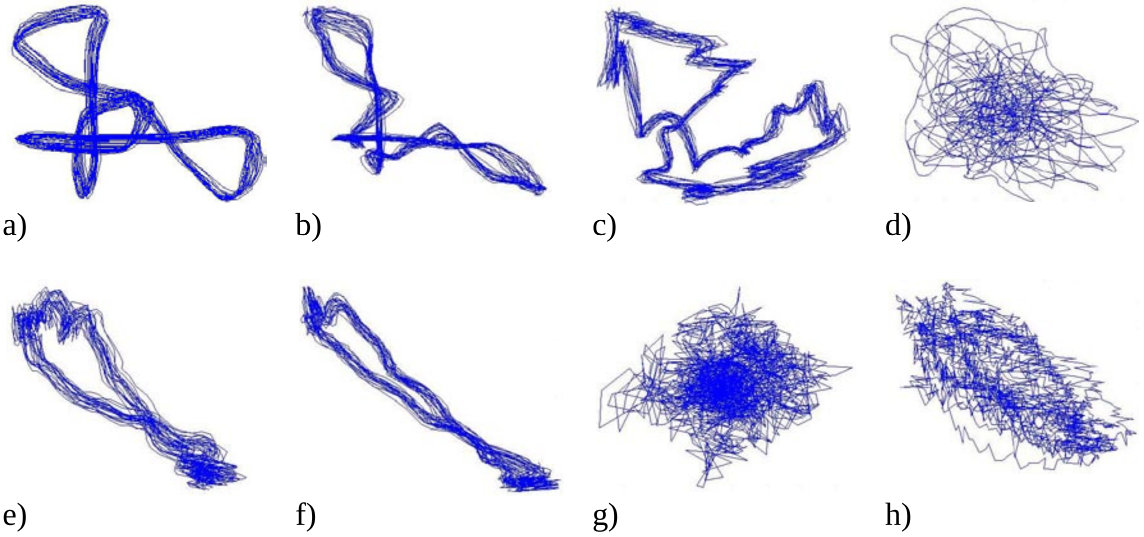

Figure 12. Reconstructed attractors for various phonemes, a) /ah/, b) /ae/, c) /eh/, d) /f/, e) /m/, f) /n/, g) /sh/, h) /z/

principle. Self-similarity is also visible in the attractor for /eh/ where the angle at the top of the attractor has several smaller angles protruding off of it. The attractors for the two nasals also have a loop structure, but the self-similarity attribute is not as obvious. The reconstructed attractors for the two nasals appear to be very similar, suggesting that the estimated invariant values will also be similar.

The reconstructed attractors for the three fricatives appear very different from those of the vowels and nasals. For the reconstructed attractor for the phoneme /sh/, there is very little visible structure. The trajectories do not seem to follow any logical path, and neighboring trajectories do not evolve in a stable manner as they did with the vowels and nasals. In fact, the attractor almost appears to have come from a stochastic process rather than speech. For the most part, the same can be said about the attractor for /f/, but the trajectories of this attractor appear much smoother and less jagged than those of the /sh/ attractor. This can be traced to the fact that the phoneme /sh/ has higher frequency components than /f/. Both of these fricatives are unvoiced, meaning that the vocal cords do not contribute to the generation of the sound, thus removing the periodic behavior seen in the reconstructed attractors of voiced phonemes.

The Lyapunov exponents for unvoiced fricatives will be higher than those for voiced phonemes because of the chaotic behavior of neighboring trajectories. The values of correlation dimension and entropy for the unvoiced fricatives can be expected to be lower than those for voiced phonemes. An accurate prediction is difficult since, for fricatives, these values are highly dependent on the amount of data used for computation [19]. For speech processing, the amount of data is usually determined by the window size.

The reconstructed attractor for the voiced fricative /z/ contains some interesting visual attributes. Although the individual trajectories appear somewhat chaotic, neighboring trajectories appear to cluster around a periodic loop. This is due to that fact that voiced fricatives reintroduce the vocal cords into the speech production process. Since Lyapunov exponents are based on the long-term evolution behavior of neighboring trajectories, the exponent value will most likely be low for this attractor, despite the chaotic appearance of the individual trajectories. The value for correlation dimension for voiced fricatives is expected to be higher than that for unvoiced fricatives due to the existence of periodic behavior. Similarly, the value for voiced fricatives is expected to be lower than that for vowels and nasals since the localized behavior within the attractor resembles that of unvoiced fricatives. The value of the correlation entropy will most likely be closer to that of vowels and nasals than unvoiced fricatives since entropy is based on long-term, global behavior of the attractor instead of local trajectory characteristics.

Visual inspections of the attractor are helpful in understanding the high-level concepts of each of the dynamic invariants. However, tuning the invariant computation parameters requires a more systematic approach. A complete description of the process used to tune these invariants can be found in [3] where the parameters were tuned specifically for speech. The following paragraphs provide an overview of the parameter values found in [3].

For all three invariants, an embedding dimension of m = 5 was used. This value was selected by minimizing the number of false nearest neighbors versus different embedding dimension values. Also, Lyapunov spectra were computed for each utterance for different embedding dimensions, and it was observed that most of the spectra converged around an embedding dimension of 5. For the time delay, a value of τ = 10 was found to work best. This was based on the average of the first minimum of the auto-mutual information [3] versus time-delay function over all phones. Finally, a neighborhood size of 25 is chosen since it was able to sufficiently capture local dynamics.

For correlation dimension, the number of embedding dimensions was also chosen to be m = 5. The other two parameters of relevance are the neighborhood radius, ε, and the Theiler correction value. Through experimentation, the optimal neighborhood radius was found to be 2.3. This radius captures enough of the local dynamics to accurately compute the correlation integral, which is the major component of the correlation dimension algorithm. The optimal Theiler correction value was found to be 150 since the distortion-causing temporal correlation effects are minimal after this amount of time. The parameter values used for correlation entropy are the same as those for correlation dimension. This is primarily due to the fact that the parameters apply to the correlation integral computation, and both correlation entropy and correlation dimension are derived from the correlation integral.

The invariants in Table 1 were computed using the parameters discussed above using a window size of 10 ms. Invariants are computed for each window segment within

Table 1. Estimated invariant values for sustained phonemes

	Lyapunov Exponent	**Correlation Dimension**	**Correlation Entropy**
/aa/	-7.7138	0.8831	665.9765
/ae/	59.8887	0.8925	590.1999
/eh/	243.5497	1.0486	729.9142
/f/	566.1099	0.5952	964.4599
/m/	-8.9635	0.8369	343.3732
/n/	39.8994	0.8944	343.5131
/sh/	795.3906	0.3282	622.7224
/z/	83.0456	0.6121	549.4435

the utterance, and then averaged to achieve the values in Table 1. As expected, the Lyapunov exponent values for vowels, nasals, and the voiced fricative /z/ are lower than those for the unvoiced fricatives. For correlation dimension, the values for fricatives are lower than those for vowels and nasals. The values for correlation entropy are less consistent. Entropy values for the two nasal phonemes are low and nearly equal. The low entropy value for the phonemes /m/ and /n/ can be attributed to the fact that the single, periodic loop of the reconstructed attractors contributes very little new information over time. The attractors for the other phonemes are more complex, resulting in higher entropy values.

The set of experiments in the following section classify signal frames from the large vocabulary continuous speech recognition task WSJ0 from a set of 40 phonemes. The next section provides a description of the WSJ0 corpus as well as a description of the experimental setup.

3.2 Phoneme Classification Experimental Setup

Before using dynamic invariants as new features for large scale continuous speech recognition experiments, it is necessary to first show that these invariants are able to distinguish between different phoneme types. In the previous section, a small set of sustained phonemes was used to tune invariant computation parameters for speech. The phoneme segments in continuous speech are much more dynamic than the sustained phones used previously, so it is also necessary to show that accurate invariant estimates can be computed from shorter, more dynamic time series. Overall, these experiments provide an idea of what kinds of improvements can be expected from a large-vocabulary speech recognition experiment using dynamic invariants as additional features.

3.2.1 Corpus Overview

As mentioned previously, the data used for this initial set of experiments is derived from the Wall Street Journal (WSJ0) Corpus. This corpus consists of high-quality recordings of speech read from newspaper articles appearing in the Wall Street Journal. The corpus is divided into a training set and an evaluation set. The training set is referred to as SI-84 [27] and consists of 7,138 utterances from 83 different speakers. Each utterance is sampled at 16 kHz and recorded using a Sennheiser close-talking microphone. The length of each utterance varies, and totals around 14 hours of speech data. The evaluation set consists of 330 utterances from eight different speakers. Both the training set and evaluation set are recorded in the same environmental conditions.

The vocabulary size of this task is about 10,000 words, and all words contained in the evaluation set have been previously seen in the training set. This vocabulary size is small compared to other large-vocabulary speech recognition tasks. However, its modest size and closed-set vocabulary eliminates most of the complex language modeling issues encountered in more complicated tasks. This makes WSJ0 ideal for research which focuses on acoustic modeling because it decouples the acoustic modeling problem from the language modeling problem. This makes the WSJ corpus ideal for this work since goal is determine whether the use of dynamic invariants results in a more robust acoustic model.

3.2.2 Experimental Setup

This set of experiments attempts to classify signal frames within the WSJ corpus as phonemes. The purpose is to gain a low-level understanding of how well dynamic invariants are able to represent speech signals. These experiments use automatic time-alignments of the corpus to extract segments for specific phonemes within each utterance. This is illustrated in Figure 13 below.

The time alignments were achieved using ISIP's Prototype System, a public domain speech recognition system [33]. Traditional 13-dimensional MFCC acoustic features, consisting of 12 cepstral coefficients and absolute energy, were computed from each of the signal frames within the phoneme segments. Each of the three nonlinear

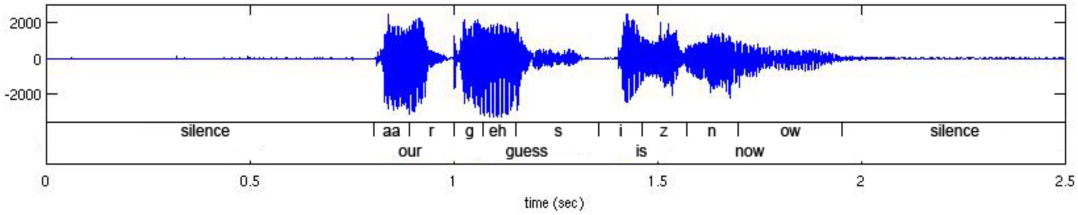

Figure 13. Time alignment for utterance of "our guess is now"

dynamic invariants is computed from the signal frames as well. The 13 MFCCs are combined with the different invariants to create three new 14-dimensional feature vectors. A separate classification experiment is performed using each of the new feature vectors in order to understand the different effects of each invariant on speech representation.

A total of 40 phonemes are used for these classification experiments. These phonemes are broken into several broad phonetic classes. A complete list and description of each class and associated phonemes can be seen in Table 2. A 16-mixture Gaussian Mixture Model (GMM) was estimated for each of the 40 phonemes. These parameters

Table 2. Broad phonetic classes used in our experimentation

Stops			**Glides**		
b	bee	B iy	l	lay	L ey
d	day	D ey	r	ray	R ey
g	gay	G ey	w	way	W ey
p	pea	P iy	y	yatch	Y aa t
t	tea	T iy	hh	hay	HH ey
k	key	K iy	**Vowels**		
Affricates			iy	beet	b IY t
jh	joke	JH ow k	ih	bit	b IH t
ch	choke	CH ow k	eh	bet	b EH t
Fricatives			ey	bait	b EY t
s	sea	S iy	ae	bat	b AE t
sh	she	SH iy	aa	bott	b AA t
z	zone	Z ow n	aw	bout	b AW t
zh	azure	ae ZH er	ay	bite	b AY t
f	fin	F ih n	ah	but	b AH t
th	thin	TH ih n	ao	bought	b AO t
v	van	V ae n	oy	boy	b OY
dh	then	DH e n	ow	boat	b OW t
Nasals			uh	book	b UH k
m	mom	M ah M	uw	boot	b UW t
n	noon	N uw N	er	bird	b ER d
ng	sing	s ih NG			

were estimated using frames from the phoneme segments extracted from the training data set. The same data was used for evaluation. This closed-loop experimental setup is acceptable since these experiments are more focused on determining whether the dynamic invariants can be used to accurately represent acoustics, as opposed to being designed to create generalized acoustic models (in which case closed-loop training cannot be done).

3.3 Phoneme Classification Experimental Results

The detailed classification results are shown in Figure 14 through Figure 17. These graphs show the relative classification improvement in accuracy for our nonlinear

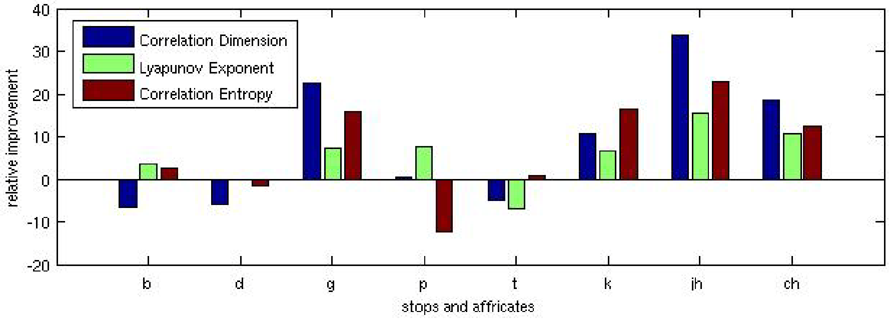

Figure 14. Relative classification accuracy improvement for stops and affricates

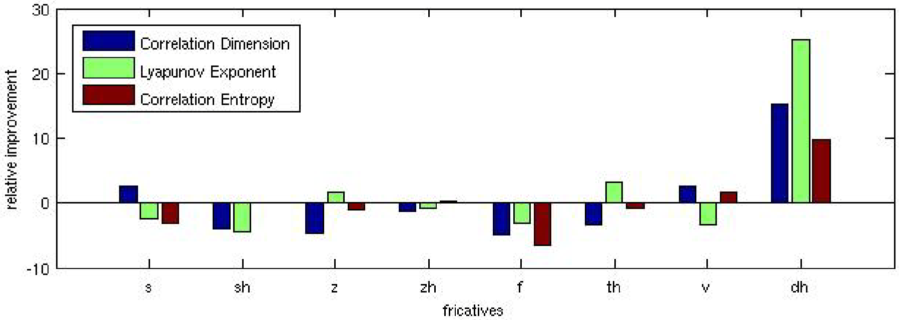

Figure 15. Relative classification accuracy improvement for fricatives

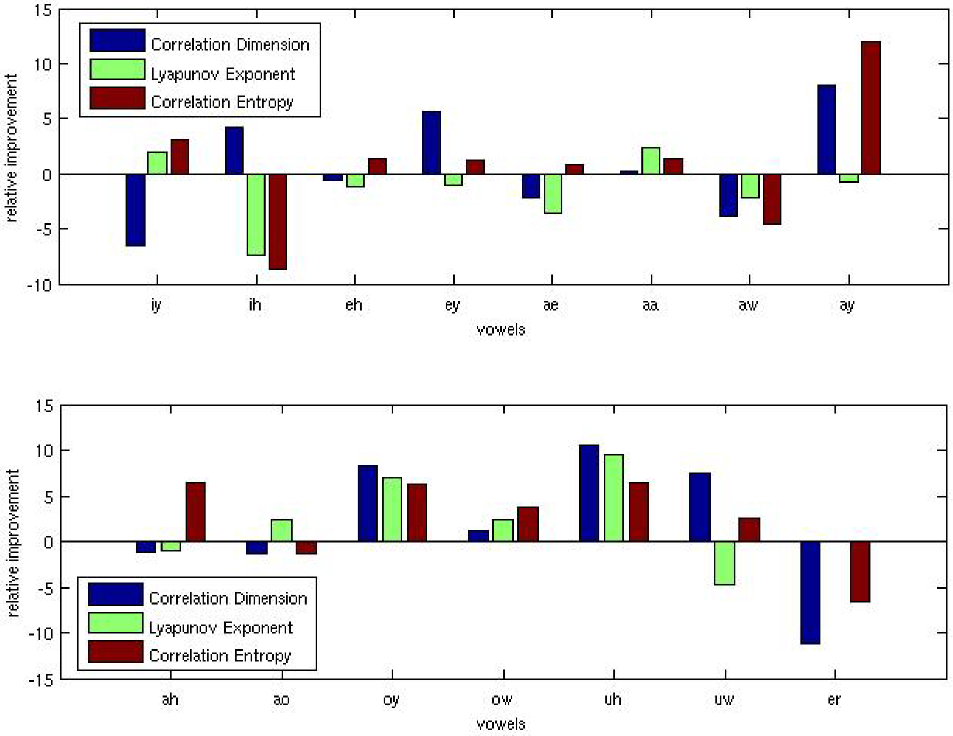

Figure 16. Relative classification accuracy improvement for vowels.

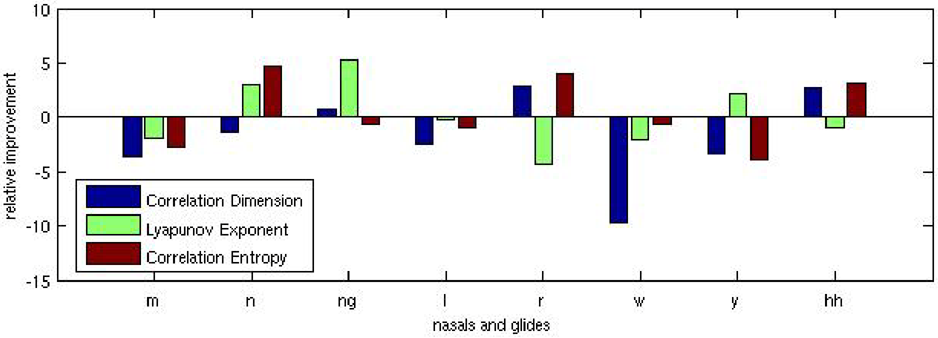

Figure 17. Relative classification accuracy improvement for nasals and glides

dynamic model as compared to the standard MFCC approach. Results for each phoneme class are presented individually in Table 3. The relative differences in accuracy are not consistent among the phonemes. Some phonemes experience a dramatic improvement in

Table 3. Average relative phoneme classification improvements using MFCC/invariant combinations

	Correlation Dimension	**Lyapunov Exponent**	**Correlation Entropy**
Affricates	10.3%	2.9%	3.9%
Stops	3.6%	4.5%	4.2%
Fricatives	-2.2%	-0.6%	-1.1%
Nasals	-1.5%	1.9%	0.2%
Glides	-0.7%	-0.1%	0.2%
Vowels	0.4%	0.4%	1.1%
Overall	1.7%	1.5%	1.4%

accuracy (e.g. /jh/, /ch/, /dh/, /ay/, /oy/, /uh/), some phonemes experience a decrease in accuracy (e.g., /f/, /ih/, /er/, /m/), and there is little change for others.

In Table 3, it can be seen that the classification results for affricates, stops, and vowels benefit the most from the addition of dynamic invariants. Accuracy for nasals decreased for correlation dimension, but accuracy increased for Lyapunov exponents and correlation entropy. There was a small decrease in accuracy for glides for correlation dimension, but glides were more or less unaffected by the other invariants. The only phoneme class that showed a consistent decrease in accuracy for all invariants was fricatives, but these decreases were relatively small. Overall, after averaging the relative increases and decreases for each phoneme class, each dynamic invariant resulted in an increase in classification accuracy.

Based on these results, it is reasonable to expect a recognition accuracy increase for continuous speech recognition experiments. The next chapter discusses these larger-scale experiments and also analyzes the invariants' robustness to noise.

CHAPTER IV

CONTINUOUS SPEECH RECOGNITION EXPERIMENTS

The results of the initial phoneme classification experiments in the previous chapter provide the necessary motivation to extend a set of experiments to a large vocabulary, continuous speech recognition (LVCSR) corpus. The phoneme classification accuracy improvements suggest that dynamic invariants may improve the recognition accuracy for continuous speech recognition tasks. In this chapter, the WSJ-derived Aurora-4 Corpus is evaluated using the different MFCC/invariant feature combinations. Two sets of experiments are performed: evaluation of clean speech data using acoustic models trained on clean speech and an evaluation of speech data with different types of digitally added noise using the same models from the first set. The remainder of this chapter provides a corpus description as well as a description of the experimental setup used for these evaluations. The results of these experiments are then discussed, followed by an interpretation of these results.

4.1 Aurora-4 Corpus Description

The Aurora-4 Corpus (A4C) is derived directly from WSJ0 and consists of the original WSJ0 data with digitally-added noise [29]. A4C is divided into two training sets and 14 evaluation sets [30]. Training Set 1 (TS1) and Training Set 2 (TS2) include the complete WSJ0 training set known as SI-84 [31]. In TS2, however, a subset of the training utterances contains various digitally-added noise conditions including six common ambient noise conditions. The 14 evaluation sets are derived from data defined by the November 1992 NIST evaluation set [32]. Each evaluation set consists of a different microphone or noise combination. The experiments in this thesis use a subset of the overall A4C. The following discussion provides an overview of this subset.

In this work, only TS1 was used to train the acoustic models. This set consists of 7,138 training utterances spoken by 83 speakers. All utterances were recorded with a Sennheiser HMD-414 close-talking microphone. The data comes from WSJ0, but has a P.341 filter applied to simulate the frequency characteristics of a 16 kHz sample rate. The set totals approximately 14 hours of speech data with an average utterance length of 7.6 seconds and an average of 18 words per utterance. There are a total of 128,294 words spoken with 8,914 of these being unique words.

Only seven of the 14 evaluation sets were used in this work due to the limited computational facilities available for these experiments. These sets include the original, noise-free data recorded with the Sennheiser microphone mentioned previously and six versions with different types of digitally-added environmental noise at random levels between 5 and 15 dB. The environments include an airport, random babble, a car, restaurant, street, and a train. Each of the seven evaluation sets consist of 330 utterances spoken by a total of eight speakers, and each utterance was filtered with the P.341 filter mentioned previously. The data for each test set totals around 40 minutes with an average of 16.2 words per utterance.

The vocabulary size of A4C is around 9,000 words, which is smaller than standard LVCSR tasks by today's standards. However, the corpus features a closed-set vocabulary, meaning that all words existing in the evaluation sets have been previously seen in the training set. Like WSJ0, these properties make A4C ideal for acoustic modeling research since a small, closed-set vocabulary decouples the problem of language modeling from the acoustic modeling problem. For a more complete description of the entire A4C, including the portions which were not described in this work, see [27].

4.2 Experimental Setup

The speech recognition experiments discussed in this chapter use a public domain speech recognition system developed at Mississippi State University [28]. This system is referred to as the Prototype System since it was the first conversational speech recognition system developed by this organization and has been used as a test bed for the development of speech recognition technology [28][33]. This system has achieved state-of-the-art performance on many speech recognition tasks [34][35][36] and its modifiable architecture and intuitive interface make it ideal for researching new technology. A toolkit based on the prototype system was developed for Aurora in [27], and the experiments in this thesis are largely based on the experimental setup in this toolkit.

The system uses HMMs with underlying GMMs for context-dependent acoustic modeling and an N-gram language model with back-off probabilities for language modeling. The Baum-Welch algorithm is used for model parameter estimation, and a Viterbi beam search is used for evaluation. For a more detailed description about this system, see [27]. The rest of this section discusses the experimental setup for the baseline system and the set of evaluation systems which test the different dynamic invariants.

4.2.1 Baseline System Setup

Before testing the feature vectors which include the dynamic invariants, it is necessary to establish a set of baseline experiments. The baseline experiments in this work evaluate the seven A4C test sets using the traditional 39-dimension MFCC feature vector without dynamic invariants. The results of these experiments will be compared to the results of the experiments using dynamic invariants to measure the effect of invariants on recognition performance.

A complex training process is used to estimate acoustic model parameters. This process is adapted from the training procedure in [27][30], and has been optimized and tuned for A4C. The explanation below describes the training procedure:

1. **Model Initialization:** Initializes the GMMs of the initial monophone models with the global mean and variance computed from the training data. This step provides a starting point for model parameter estimation.

2. **Initial Monophone Training:** Four iterations of Baum-Welch training are used to re-estimate monophone model parameters based on the training data. This step also allows the models to learn the silence at the beginning and end of the utterance.

3. **Short-Pause (Interword Silence) Model Training:** Four additional iterations of Baum-Welch are used to further re-estimate model parameters. This step also trains the short-pause ('sp') model which models the silence between words.

4. **Forced Alignment:** The training data transcriptions are aligned to the training acoustic data and the most likely pronunciation for each word in the transcription is chosen. A new set of phonetic transcriptions are generated from this process and this set is used throughout the remainder of the training process.

5. **Final Monophone Training:** Five final iterations of Baum-Welch are used to further re-estimate the model parameters using the new transcriptions generated in the forced alignment step.

6. **Cross-Word Triphone Training:** Context-dependent, cross-word triphone models are generated and initialized from the trained monophone models. Only triphones existing in the training data are created. Four iterations of Baum-Welch are used to re-estimate the triphone model parameters.

7. **State-Tying:** To reduce the parameter count and to provide sufficient training data to undertrained states, states that are statistically similar to one another are tied into a single state, and the training data previously attributed to each state is now shared in the single tied state. Four passes of Baum-Welch are then used to re-estimate the parameters of the new state-tied models.

8. **Mixture Training:** The single mixture models are successively split until 4 mixtures are generated using incremental stages of 1, 2, and 4 mixtures. At each stage, four iterations of Baum-Welch are used to re-estimate the parameters of the multi-mixture models.

The acoustic models are trained using TS1 described in Section 4.1. These acoustic models are used to evaluate the clean test set as well as the six noisy test sets. The training data does not contain any instances of the six noise conditions since the goal of this work is to determine whether dynamic invariants can be used to generalize acoustic models to unseen conditions in the training data.

The experiments were designed to balance recognition performance and speed. Due to limited computational resources, acoustic models are not split beyond four Gaussian mixtures. Although better recognition performance could be achieved using a higher number of mixtures, the required CPU time increases as the number of mixtures increases. Since this work requires running a large volume of experiments, the time required to run these experiments must be as short as possible. Using 4-mixture GMMs provides a reasonable balance of computation time and recognition performance [27][30].

4.2.2 Evaluation Setup

The evaluation experiments are used to test the effects of dynamic invariants on speech recognition performance. In this work, four sets of experiments are used to evaluate the A4C data. Each set uses a different combination of MFCCs and dynamic invariants. These feature vectors are described in Table 4 below.

The traditional MFCC feature vector consists of 12 Cepstral coefficients, absolute energy, and the first and second derivatives of these values, which results in a base feature vector totaling 39 dimensions. The three dynamic invariants are extracted from all utterances of both the training and testing sets using the methods discussed in Chapter II. The four new feature vectors are constructed by simply appending the invariants to the existing MFCC features. This results in four new training sets, and 28 new evaluation sets (seven sets per each of the four new feature sets). Each of the seven test sets are evaluated using the four new feature vectors. As mentioned previously, each evaluation uses

Table 4. Description of the different feature sets used for evaluation

Feature Set 1 (FS1)
MFCCs (39)
Correlation Dimension (1)
40 Dimensions Total

Feature Set 2 (FS2)
MFCCs (39)
Lyapunov Exponent (1)
40 Dimensions Total

Feature Set 3 (FS3)
MFCCs (39)
Correlation Entropy (1)
40 Dimensions Total

Feature Set 4 (FS4)
MFCCs (39)
Correlation Dimension (1)
Lyapunov Exponent (1)
Correlation Entropy (1)
42 Dimensions Total

acoustic models trained with the data from TS1. The experimental parameters, including beam pruning and language model parameters, were tuned in [27][30] using the Aurora development set.

4.3 Evaluation Results

This section presents the evaluation results described in the previous section. The word error rate (WER) results were obtained using the standard NIST scoring software. This software quantifies the number of errors within the recognition hypotheses and provides a means of performance comparison between the evaluations of a common test set by two different systems. These errors consist of misrecognized, inserted, and deleted words. The overall WER is the ratio of word recognition errors to the total number of words within the reference data transcriptions.

4.3.1 Significance Testing

Although WER provides a reasonable performance comparison, it is not the best way to determine whether one recognition system performs better than another. In this work, the size of the test sets is 330 utterances, and such a small evaluation set can introduce noise in the experimental design in the form of statistical fluctuations which do not truly represent the recognition performance of the system [30]. For example, suppose a first system results in a WER which seems significantly lower than a second system. It would be tempting to label the first system "better" than the second competing system. However, it is possible that a small subset of test utterances evaluated by the second system encountered problematic evaluation issues such as corrupt acoustic data, hardware failures, software failures, etc. Although the resulting hypothesis errors damage the WER for the second system, the errors are not representative of the performance of the system on the entire evaluation set since the errors are not statistically uniform across the entire set of recognition hypotheses [30].

The statistical significance testing method provides a measure of the extent to which one system outperforms another by measuring the distribution of errors within the entire evaluation set. In this work, significance testing was performed using NIST's Matched Pairs Sentence-Segment Word Error (MAPSSWE) method [37]. This method selects random segments of sentences from within the recognition hypotheses of each system and performs a pairwise comparison of the number of errors within these selections. The result of this test is the significance level value, p, which is the probability that the distribution of errors for both systems is the same. If this probability is high, the distributions for both systems are similar which means that the difference in WER is not necessarily a significant indicator of superior performance of one of the systems. A low value of p suggests that the errors for both systems did not likely come from the same distribution, and therefore, the difference in WER is an indicator of significant performance difference.

4.3.2 Evaluation Results for Noise-Free Data

The recognition results for the noise-free evaluation set are presented in Table 5. For easy visualization, these results are also presented graphically in Figure 18. Table 5 provides the WER for each feature set, as well as the relative improvement over the baseline system and the significance level of the results of each system.

Table 5. Recognition performance for different feature sets

Dynamic Invariant	**WER (%)**	**Improvement (%)**	**Significance Level (*p*)**
Baseline (FS0)	13.5	--	--
Feature Set 1 (FS1)	12.2	9.6	0.030
Feature Set 2 (FS2)	12.5	7.4	0.075
Feature Set 3 (FS3)	12.0	11.1	0.001
Feature Set 4 (FS4)	12.8	5.2	0.267

All four feature sets with dynamic invariants resulted in a decreased WER compared to the baseline MFCC features. This reinforces the pilot experiment results in Section 3.3 where an increase in phoneme classification accuracy was seen for each feature set. The most significant WER improvement was seen for FS3 which contains the correlation entropy invariant as an added feature. The relative improvement in this case

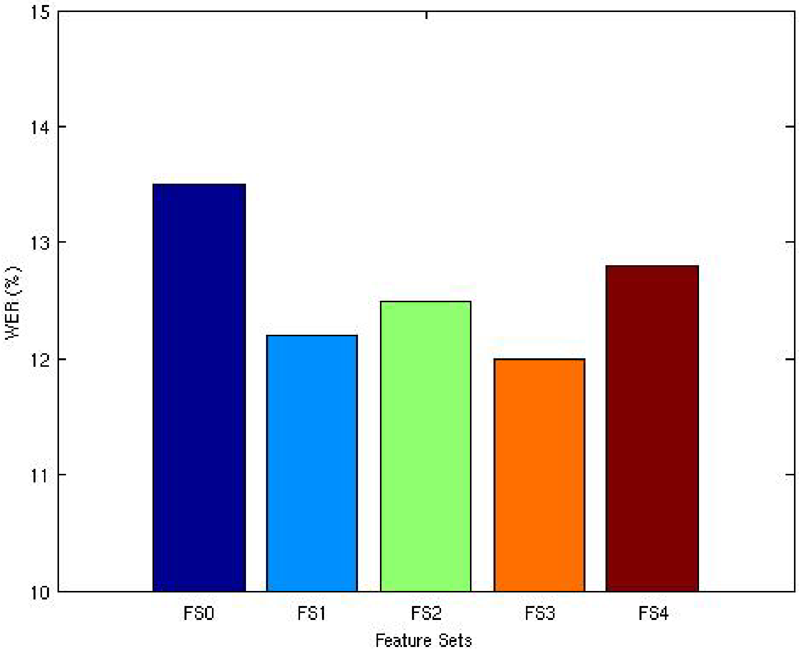

Figure 18. Graph of recognition performance for different feature sets

was 11.1% with a significance level of 0.1%. The results for FS4 which contains all three invariants shows a slight WER improvement, but this improvement is small and insignificant compared to the improvements seen by feature sets containing a single invariant. This suggests that the invariants contribute a certain level of overlapping information about the nonlinear properties of the acoustics. The next section discusses the results for the noisy evaluation sets.

4.3.3 Evaluation Results for Noisy Data

The evaluation results for the six noisy evaluation sets are presented in Table 6 and Table 7, and are also shown graphically in Figure 19. The results for the noisy data are much less encouraging than those for the noise-free data. Most of the evaluations resulted in a higher WER than the baseline. The correlation dimension and Lyapunov

Table 6. WER results for noisy evaluation data using different feature sets

	WER (%)					
	Airport	Babble	Car	Restaurant	Street	Train
Baseline	53.0	55.9	57.3	53.4	61.5	66.1
FS1	57.1	59.1	65.8	55.7	66.3	69.6
FS2	56.8	60.8	60.5	58.0	66.7	69.0
FS3	52.8	56.8	58.8	52.7	63.1	65.7
FS4	58.6	63.3	72.5	60.6	70.8	72.5

Table 7. Relative WER improvements over baseline for noisy evaluation data

	Relative Improvements (%)					
	Airport	Babble	Car	Restaurant	Street	Train
FS1	-7.7	-5.7	-14.8	-4.4	-7.8	-5.3
FS2	-7.2	-8.8	-5.6	-8.6	-8.5	-4.4
FS3	0.4	-1.6	-2.6	1.3	-2.6	0.6
FS4	-10.6	-13.2	-26.5	-13.5	-15.1	-9.7

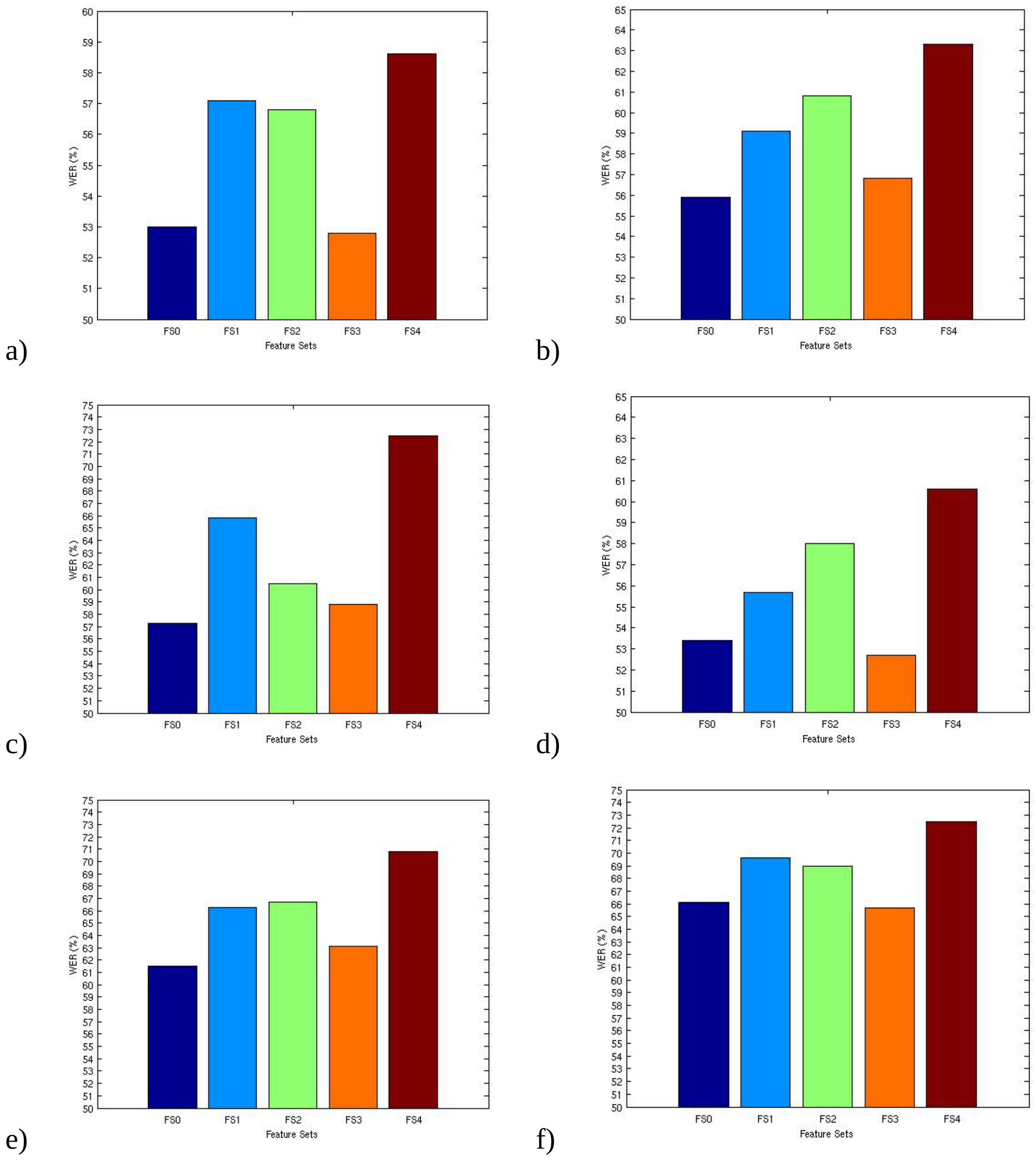

Figure 19. Graphs of recognition performance for the six noisy evaluation sets: a) airplane, b) babble, c) car, d) restaurant, e) street, and f) train.

exponent invariants caused an average WER increase of around 7%. The only invariant that appears somewhat promising is correlation entropy in FS3. Three of the evaluation sets resulted in a slight WER decrease (shaded in gray in Tables 6 and 7), while the other three resulted in increases. The WER increases for FS3, however, were significantly lower than the increases for the other feature sets. Although three sets saw slight improvements for correlation entropy, these improvements were not statistically significant. The use of all three invariants in FS4 had the most damaging effect on performance with an average WER increase of around 14%.

4.3.4 Analysis

The recognition performance improvements for the noise-free data suggest that nonlinear dynamic invariants have a significant contribution to traditional acoustic information and can be used to better model speech. Although the improvements for clean speech data are encouraging, one of the primary purposes of this work was to determine whether nonlinear features can improve recognition accuracy for speech recorded in unseen environments. According to the results of the experiments presented in the previous section, the dynamic invariants used in this work are unable to achieve an improvement.

One reason for this may be that the dynamic invariant computation methods are not conducive to accurate estimation from noisy data. Since frame-based feature extraction estimates features from small segments of the speech signal, the length of the segment may not be long enough to estimate accurate dynamic invariant values when the signal in contaminated with noise [17]. Since noise distorts the phase space, a longer time series is required in order to sufficiently capture the true dynamics of the system. Unfortunately, the dynamic nature of speech signals places a limit on the extent to which the frame length can be extended since an excessively large frame will capture the dynamics of more than one phoneme.

The opportunities for the improvement of nonlinear dynamic invariant techniques lie within the filtering of the reconstructed attractor. While the use of SVD embedding for phase space reconstruction can reduce the effects of noise, this work suggests that it is not an effective method of noise filtering when used alone. Additional filtering techniques are required in order to better reduce the effects of noise on the dynamics of the attractor.

CHAPTER V

CONCLUSIONS AND FUTURE DIRECTIONS

This thesis explored a technique for using nonlinear dynamic invariants as features for continuous speech recognition. When combined with traditional MFCC features, dynamic invariants exploit the underlying nonlinear properties of the speech signal resulting in a more accurate acoustic model. The purpose of this work was to determine whether dynamic invariants could improve recognition performance for a large-vocabulary, continuous speech recognition task. Additionally, the question of whether or not these nonlinear features could produce an acoustic model which is more robust to unseen environmental conditions was explored.

For noise-free evaluation data, it was shown that the addition of dynamic invariants to traditional MFCCs could significantly boost recognition performance and result in a lower WER. However, dynamic invariants were not able to improve the performance of recognition for noisy evaluation sets. The use of dynamic invariants had a negative effect on the recognition performance for noisy data.

5.1 Thesis Contribution

In this thesis, a variety of experiments were run in order to determine whether nonlinear dynamic invariants can be used to create a better acoustic model for speech recognition. The first contribution involved a set of pilot experiments which classified frames within utterances from the WSJ corpus as phonemes. Traditional MFCC features were combined with dynamic invariants, and a set of GMMs were trained for each feature combination. The purpose of these initial experiments was to gain a low-level understanding of the effect these invariants have on speech modeling. It was found that the addition of dynamic invariants was able to improve phoneme classification accuracy. For correlation dimension, an overall relative improvement of 1.7% was observed. Lyapunov exponents and correlation entropy saw similar improvements at around 1.5% and 1.4%, respectively. These results suggest that dynamic invariants will be able to improve recognition performance for large-scale continuous speech recognition experiments.

The second contribution of this thesis was the evaluation of the MFCC/invariant feature combinations on the Aurora 4 Corpus (A4C). The data sets evaluated included one noise-free set, and six sets with various noise conditions. For the noise-free data, dynamic invariants were able to significantly improve recognition accuracy. The relative WER improvements seen were: 9.6% for correlation dimension, 7.4% for Lyapunov exponents, and 11.1% for correlation entropy. Combining MFCCs with all three invariants also improved performance, but at 5.2%, the relative improvement was not as significant as those using individual invariants. Overall, these results suggest that nonlinear dynamic invariants can be used to better model acoustics and can improve speech recognition performance when evaluation and training conditions match.

Using the models trained from A4C's clean training set, evaluations were also performed on six noisy data sets. These data sets contain digitally added noise conditions which vary significantly from the training conditions. Unfortunately, the dynamic invariants did not improve recognition performance. A few slight improvements were seen using the correlation entropy invariant, but these improvements were not statistically significant. The negative results suggest that the dynamic invariant computation methods explored in this thesis are not effective for noisy data. This is most likely due to the frame length used for computation results in a time-series which is too short to get an accurate representation of system dynamics. Further research is required to develop advanced phase space filtering techniques.

5.2 Future Work

Although the negative results seen in the evaluation of the noisy data sets were disappointing, they provide some motivation for further research in filtering techniques. In this thesis, the only method of phase space noise reduction was the use of SVD phase space reconstruction. While this method has been shown to reduce the effects of noise, it is not very effective for speech since the time-series used for phase space reconstruction is limited by the short frame length. Therefore, more research is required for the development of advanced phase space filtering techniques which can be used to post-process a reconstructed phase space and reduce the effects of noise on phase space dynamics.

Nonlinear methods for speech recognition provide many new potential research areas. For example, instead of computing values which describe the global behavior of the attractor, such as dynamic invariants, it may be beneficial to model the attractor itself.

This would require a new type of statistical model and would provide a more complete description of the local dynamics within the attractor.

REFERENCES

[1] A. Kumar and S. K. Mullick, “Nonlinear Dynamical Analysis of Speech,” Journal of the Acoustical Society of America, vol. 100, no. 1, pp. 615-629, July 1996.

[2] H. M. Teager and S. M. Teager, “Evidence for Nonlinear Production Mechanisms in the Vocal Tract,” NATO Advanced Study Institute on Speech Production and Speech Modeling, Bonas, France, pp. 241-261, July 1989.

[3] S. Prasad, S. Srinivasan, M. Pannuri, G. Lazarou and J. Picone, “Nonlinear Dynamical Invariants for Speech Recognition,” Proceedings of the International Conference on Spoken Language Processing, pp. 2518-2521, Pittsburgh, Pennsylvania, USA, September 2006.

[4] R. W. Schafer, L. R. Rabiner, “Digital Representations of Speech Signals,” Proceeding of the IEEE, vol. 63, no. 4, April 1975.

[5] A. M. Noll, “Cepstrum Pitch Determination,” Journal of the Acoustical Society of America, vol. 41, pp. 293-309, February 1967.

[6] F. Zheng, G. Zhang, Z. Song, “Comparison of Different Implementations of MFCC,” Journal of Computer Science and Technology, vol. 16, no. 6, pp. 582-589, September 2001.

[7] T. Vivek, C. Wellekens, “On Desensitizing the Mel-Cepstrum to Spurious Spectral Components for Robust Speech Recognition,” Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 529-532, Philadelphia, Pennsylvania, USA, March 2005.

[8] H. M. Teager, “Some Observations on Oral Air Flow During Phonation,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 5, October 1980.

[9] P. Maragos, A. G. Dimakis and I. Kokkinos, “Some Advances in Nonlinear Speech Modeling using Modulations, Fractals, and Chaos,” Proceedings of the IEEE International Conference on Digital Signal Processing, pp. 325-332, Santorini, Greece, July 2002.

[10] M. Banbrook, S. McLaughlin, "Is Speech Chaotic?: Invariant Geometrical Measures for Speech Data," IEE Colloquium on Exploiting Chaos in Signal Processing, Digest No. 1994/193, pp. 8/1-8/10, London, U.K., June 1994.

[11] G. Zhou, J. H. L. Hansen and J. F. Kaiser, "Nonlinear Feature Based Classification of Speech Under Stress," IEEE Transactions on Speech and Audio Processing, vol. 9, no. 3, pp. 201-216, March 2001.

[12] P. E. Rapp, T. A. Watanabe, P. Faure, and C. J. Cellucci, "Nonlinear Signal Classification," International Journal of Bifurcation and Chaos, vol. 12, no. 6, pp. 1273-1293, June 2002.

[13] H. Yehia and F. Itakura, "Determination of Human Vocal-Tract Dynamic Geometry from Formant Trajectories using Spatial and Temporal Fourier Analysis," Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 477-480, Adelaide, Australia, April 1994.

[14] H. Wakita, "Estimation of Vocal-Tract Shapes from Acoustical Analysis of the Speech Wave: The State of the Art," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27. no. 3, pp. 281-285, June 1979.

[15] H. Kantz and T. Schreiber, Nonlinear Time Series Analysis, Cambridge University Press, New York, New York, USA, 2003.

[16] P. Maragos, "Fractal Aspects of Speech Signals: Dimension and Interpolation," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 417-420, Toronto, Canada, May 1991.

[17] M. Banbrook, G. Ushaw, and S. McLaughland, "How to Extract Lyapunov Exponents from Short and Noisy Time Series," IEEE Transactions on Signal Processing, vol. 45, no. 5, pp. 1378-1382, May 1997.

[18] P. Grassberger and I. Procaccia, "Estimation of the Kolmogorov Entropy from a Chaotic Signal," Physical Review A, vol. 28, no. 4, pp. 2591-2594, October 1983.

[19] H. F. V. Boshoff and M. Grotepass, "The Fractal Dimension of Fricative Speech Sounds," Proceedings of the South African Symposium on Communication and Signal Processing, pp. 12-16, Pretoria, South Africa, August 1991.

[20] A. C. Lindgren, M. T. Johnson and J. Povinelli, "Speech Recognition using Reconstructed Phase Space Features," Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. I-60-63, Hong Kong, China, April 2003.

[21] V. Pitsikalis and P. Maragos, “Filtered Dynamics and Fractal Dimensions for Noisy Speech Recognition,” IEEE Signal Processing Letters, vol. 13, no. 11, pp. 711-714, November 2006.

[22] J. F. Gibson, J. Farmer, M. Casdagli, and S. Eubank, “An Analytic Approach to Practical State Space Reconstruction,” Physica D, vol. 57, no. 1-2, pp. 1–30, June 1992.

[23] J. Vitrano and R. J. Povinelli, “Selecting Dimensions and Delay Values for a Time-Delay Embedding Using a Genetic Algorithm,” Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, California, USA, pp. 1423-1430, July 2001.

[24] D. S. Broomhead and G. P. King, “Extracting Qualitative Dynamics from Experimental Data,” Physica D, vol. 20, pp. 217–236, 1986.

[25] J. Theiler, “Spurious Dimension from Correlation Algorithms Applied to Limited Time-Series Data,” The American Physical Society, vol. 34, no. 3, pp. 2437-2432, September 1986.

[26] P. Grassberger and I. Procaccia, “Estimation of the Kolmogorov Entropy from a Chaotic Signal,” Physical Review A, vol. 28, no. 4, pp. 2591-2594, October 1983.

[27] N. Parihar, “Performance Analysis of Advanced Front Ends on the Aurora Large Vocabulary Evaluation,” M.S. Thesis, Department of Electrical and Computer Engineering, Mississippi State University, November 2003.

[28] N. Deshmukh, A. Ganapathiraju, J. Hamaker, J. Picone and M. Ordowski, “A Public Domain Speech-to-Text System,” Proceedings of the 6th European Conference on Speech Communication and Technology, vol. 5, pp. 2127-2130, Budapest, Hungary, September 1999.

[29] D. Pearce and G. Hirsch, “The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition System under Noisy Conditions”, Proceedings of the International Conference on Spoken Language Processing (ICSLP), pp. 29-32, Beijing, China, October 2000.

[30] N. Parihar and J. Picone, "DSR Front End LVCSR Evaluation - AU/384/02," Aurora Working Group, European Telecommunications Standards Institute, December 6, 2002.

[31] D. Paul and J. Baker, “The Design of Wall Street Journal-based CSR Corpus,” Proceedings of International Conference on Spoken Language Processing (ICSLP), pp. 899-902, Banff, Alberta, Canada, October 1992.

[32] D. Pallett, J. G. Fiscus, W. M. Fisher, and J. S. Garofolo, “Benchmark Tests for the DARPA Spoken Language Program,” Proceedings of the Workshop on Human Language Technology, pp. 7-18, Princeton, New Jersey, USA, March 1993.

[33] K. Huang and J.Picone, “Internet-Accessible Speech Recognition Technology,” presented at the IEEE Midwest Symposium on Circuits and Systems, Tulsa, Oklahoma, USA, August 2002.

[34] R. Sundaram, A. Ganapathiraju, J. Hamaker and J. Picone, “ISIP 2000 Conversational Speech Evaluation System,” presented at the Speech Transcription Workshop, College Park, Maryland, USA, May 2000.

[35] R. Sundaram, J. Hamaker, and J. Picone, “TWISTER: The ISIP 2001 Conversational Speech Evaluation System,” presented at the Speech Transcription Workshop, Linthicum Heights, Maryland, USA, May 2001.

[36] B. George, B. Necioglu, J. Picone, G. Shuttic, and R. Sundaram, “The 2000 NRL Evaluation for Recognition of Speech in Noisy Environments,” presented at the SPINE Workshop, Naval Research Laboratory, Alexandria, Virginia, USA, October 2000.

[37] “Benchmark Tests, Matched Pairs Sentence-Segment Word Error (MAPSSWE),” http://www.nist.gov/speech/tests/sigtests/mapsswe.htm, Speech Group, National Institute for Standards and Technology, Gaithersburg, Maryland, USA, January 2001.