2018

Robust Spectral Classification

Andrew Tucker
University of Rhode Island, andrewt081@yahoo.com

ROBUST SPECTRAL CLASSIFICATION

BY

ANDREW TUCKER

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2018

MASTER OF SCIENCE THESIS

OF

ANDREW TUCKER

APPROVED:

Thesis Committee:

Major Professor: Steven Kay

Ramdas Kumaresan

Steffen Ventz

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018

Abstract

Spectral classification is a commonly used technique for discriminating between two or more signals. The first step in the classification process is to sample a signal with an analog-to-digital converter. Then the power spectral density is estimated. To classify the data, the estimated power spectral density of the unknown signal is compared to power spectral densities from two or more known templates using a classifier. Despite the substantial prior research effort put into developing a robust classifier, the results are not great and in some instances are not even satisfactory.

The topic of this thesis is to evaluate a classifier that may be more robust than those currently used; the realizable Poisson likelihood function. Robustness is determined by the probability of correct classification when there are differences between training data and observed data. Taking the familiar form of the Kullback-Leiber divergence, the realizable Poisson likelihood function is mathematically tractable since it is derived from an alternative model for the power spectral density of a non-homogeneous Poisson process.

The realizable Poisson likelihood function was compared to other popular classifiers. Monte Carlo simulations were done using autoregressive processes with and without distortions added to the observed data. Then a more thorough analysis was done using actual data. Results are presented that show the realizable Poisson likelihood function to be a robust classifier. The performance

of the realizable Poisson likelihood function decreases only very slightly with moderate signal-to-noise ratios and in the presence of channel distortions. This is compared to significant performance reduction of other classifiers.

Acknowledgments

There are so many people that helped me complete this work and I owe a debt of gratitude to all of them. First of all, I would thank my adviser and mentor Dr. Steven Kay. It has been such an honor to study with Dr. Kay, I feel extremely lucky to have had the opportunity that was given to me. He has in depth knowledge of not only statistical signal processing but also research in general. I will always think fondly of the chats we had in his office. After leaving URI I take comfort in knowing that when I have a challenging problem I can think to myself "what would Dr. Kay do?".

Next, I would like to thank Dr. Kumaresan. Dr. Kumaresan has been amazingly supportive, without him I wouldn't be where I am. Also, a heartfelt thank you to Dr. Ventz for being on my committee.

Lastly, I would like to thank my family. My wife Jane has been so amazing throughout this whole process. She has been so supportive and understanding, without her I don't know where I would be. She gave me the time when I needed it and believed in me when I had doubts. My children Lily and Sam are the greatest two things that have ever happened to me. They have been so understanding; there have been many times when they wanted to play but I needed to study.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Objective

The performance of modern spectral classification techniques is severely diminished when there are differences in the training spectra and the spectra under test. The objective of this work is to compare the robustness of the newly developed realizable Poisson likelihood function (RPLF) classifier to other classifiers typically used in research or industry.

1.2 Motivation

The Merriam-Webster dictionary defines "robust" as "capable of performing without failure under a wide range of conditions" [20]. While most people are aware of how changing physical conditions like temperature and vibration test the robustness of hardware, an analogy can be made to software. Software is used in almost every industry now, with applications ranging from medical to economics, engineering to sports and everything in between. Let us consider a well-known example, the modern smartphone. The electronics must work in a wide range of environmental conditions, input power conditions, and conditions with high electromagnetic interference. The software must adjust with different users in different conditions. One example is the touch algorithm that is an essential part of the user interface. One user may have small fingers

and another quite large. Varying finger size will create two different responses from the electronics but the software is expected to react the same.

Software algorithms must be able to adjust to accommodate these classification problems. A classification problem is one in which there are two or more known items and one unknown. The unknown item is determined to be associated with one of the known items. In the above touch example, the known items are: 1) the response of the electronics to a touch, and 2) a response to a no touch. The unknown is the current signal coming from the electronics. This touch example is an example of a simple binary classifier; however most classifiers are much more complicated, one such example is speech recognition software.

According to Markel and Gray (1982), though many different signal models have been postulated, no single model has been developed which can account for all of the observed characteristics of human speech. One of the most widely used models of speech is the linear prediction speech model which uses a mathematical technique called linear prediction [19]. This technique is not just used for speech but for many applications where the signal to be modeled is very complex.

Many times the input to the linear prediction model is assumed to be a Gaussian random process [15]. There is a good reason for this. It has been well studied, is mathematically tractable, and results from the central limit theorem [19]. The Gaussian input is then processed with a linear filter. With

a Gaussian input and a linear filter the output will then also take the form of a Gaussian random process.

With this representation of a linear prediction model two separate likelihood functions can be derived. These likelihood functions can be used as classifiers when trying to associate an unknown signal to known signals. The first likelihood function is the exact Gaussian. The exact Gaussian can produce great results but includes the inverse of a matrix, which is a computationally extensive process even with moderate size matrices. The other likelihood function is asymptotically equivalent to the exact Gaussian. This asymptotic equivalent requires fewer calculations, due to the efficiency of Fourier transform properties. In this thesis the asymptotic equivalent will take the form of a normalized Itakura-Saito distance measure, and will be referred to as just the Itakura-Saito.

The Itakura-Saito is not very robust in nature; noise, channel deformation, and shape of the spectrum are a few issues that can lead to errors [15], [25]. A more robust classifier would reduce these errors and lead to more reliable solutions. Kay [15] derived a likelihood function that bases the frequency spectrum on a nonhomogeneous Poisson process that appears to be more robust. This likelihood function is termed the *Realizable Poisson likelihood function* or RPLF and, in this thesis will be referred to as both the RPLF or Poisson. This thesis will test this theory using simulations, data analysis and analytical

derivations.

1.3 Contributions

The two main contributions to state of the art statistical signal processing that this thesis provides are:

- An indepth look at the RPLF, an alternative classifier that will prove to be more robust when used in simulations and on data.
- An analytical expression for the divergence between the exact Gaussian likelihood function and the asymptotic Gaussian for an autoregressive process with a single pole.

1.4 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 presents some background on spectral classification. Chapter 3 includes the methods used to determine the robustness of the derived classifier. This includes the plan for simulations, and a discussion of the data used and the method of data analysis. Chapter 4 will give results of the simulations and the data analysis. The final chapter summarizes conclusions and further research.

CHAPTER 2
BACKGROUND

2.1 Summary

The intent of this chapter is to give enough information so the reader understands the subsequent chapters, but in no way does it attempt to be a complete treatment of the subject of parametric modeling or Poisson processes. There are a number of good books the interested reader could reference. These include *Modern Spectral Estimation* by Kay [14], *Linear Prediction of Speech* by Markel and Gray [19] and, *Random Point Processes* by Snyder [23]. First, the all-pole model for signal representation and some of the issues that arise while using this model are described. Next a seldom-used model for the Fourier spectrum of a non-homogenous Poisson process is discussed, which leads to a new classifier. Next, an example of a classification problem is presented. Lastly, the chapter concludes with a brief background on speech.

2.2 Autoregressive Process

There seems to be many ways to represent a signal. Two of the most common are the time domain representation and the frequency domain representation. The well-known continuous-time Fourier transform is the method used to convert from the time domain representation to the frequency domain representation

and vice versa.

$$X(f) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-j2\pi ft}dt \tag{2.1}$$

$$x(t) = \int_{-\infty}^{\infty} X(f)\mathrm{e}^{j2\pi ft}df \tag{2.2}$$

The continuous time transform consists of the analysis(2.1) and synthesis(2.2) equations, leading to the Fourier transform pair,

$$x(t) \overset{\mathscr{F}}{\longleftrightarrow} X(f) \tag{2.3}$$

When analyzing signals, the power in each of the frequencies is of particular interest, not just in the frequency content. A classical technique for estimating the power in the signal was developed by Schuster in 1898, called the periodogram [14], which is the magnitude squared of the Fourier transform. Based on the discrete-time Fourier transform the periodogram takes the form of,

$$\hat{P}_X(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j2\pi fn}\right|^2$$

The periodogram has been shown not to be a suitable measure of the distribution of power with frequency [13]. If an expectation operator and an infinite length

realization are used the power spectral density(PSD) is realized, then

$$P_X(f) = \lim_{M\to\infty} \frac{1}{2M+1} E\left[\left|\sum_{n=-M}^{M} x[n]\mathrm{e}^{-j2\pi fn}\right|^2\right]$$

Kay (2016) states that, the fact that the PSD is completely analogous to a probability density function (PDF). The average power of the random process in the frequency band $f_1 \leq f \leq f_2$ is equivalent to the area under the PSD curve in that band. The probability of an event is equivalent to the area under the PDF in between the desired points. If the PSD is normalized, all the same mathematical tools may be used [15], [17]. Conveniently, the PSD is also simply the Fourier transform of the autocorrelation function.

$$r_X[n] \overset{\mathscr{F}}{\longleftrightarrow} P_X(f)$$

Thus, the average power of the signal can be found by estimating the autocorrelation function at zero lag $r_X[0]$.

Parametric modeling is the technique used to model the PSD of a random process. While there are different types of parametric models it is frequently modeled as a zero mean wide sense stationary (WSS) white Gaussian random process put through a linear all-pole filter, also known as an autoregressive ($AR(p)$) process. The $AR(p)$ process is widely used because it models the PSD well and is mathematically tractable [21].

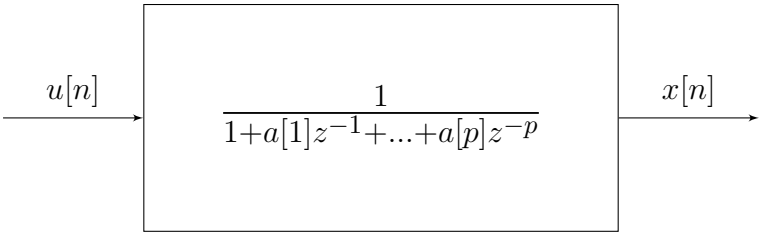


Figure 2.1: $AR(p)$ process

Figure 2.1 is a diagram of an $AR(p)$ process with order p. The input $u[n]$ is the WSS Gaussian random process with a PDF of:

$$p_{\boldsymbol{X}}(x) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N} x_i^2\right)$$

Since the $AR(p)$ is linear, the output $x[n]$ can also be shown to be Gaussian and takes the form of:

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{N/2} det^{1/2}(\boldsymbol{C})} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{C}^{-1}\boldsymbol{x}\right) \tag{2.4}$$

The covariance matrix $\boldsymbol{C}$ is determined by the $AR(p)$ process, and is the same as the autocorrelation matrix $\boldsymbol{R}$ when $u[n]$ is zero mean.

The two most common ways of estimating the $a[k]$ coefficients are the covariance method of linear prediction and the autocorrelation method of linear prediction. The autocorrelation method, sometimes called the Yule-Walker method, uses the Levinson recursion to solve a system of linear equations. Using the autocorrelation method, the resulting estimated poles are guaranteed to be within

the unit circle [14]. The set of autocorrelation equations is:

$$\begin{bmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \dots & \hat{r}_{xx}[-(p-1)] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}(0) & \dots & \hat{r}_{xx}[-(p-2)] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{xx}[p-1] & \hat{r}_{xx}[p-2] & \dots & \hat{r}_{xx}[0] \end{bmatrix} \begin{bmatrix} \hat{a}[1] \\ \hat{a}[2] \\ \vdots \\ \hat{a}[p] \end{bmatrix} = - \begin{bmatrix} \hat{r}_x[1] \\ \hat{r}_x[2] \\ \vdots \\ \hat{r}_x[p] \end{bmatrix} \tag{2.5}$$

Figure 2.2 is an example of a signal modeled with an $AR(p)$ process. The signal is of the letter "A" being spoken by a male. Figure 2.2a is the waveform of the entire utterance, Figure 2.2b is a $30ms$ sample from the middle of the utterance. Figures 2.2c and 2.2d are the periodograms, in blue, and power spectral densities with the model order of $p = 8$ and $p = 14$ $AR(p)$ process, in red-dashed.

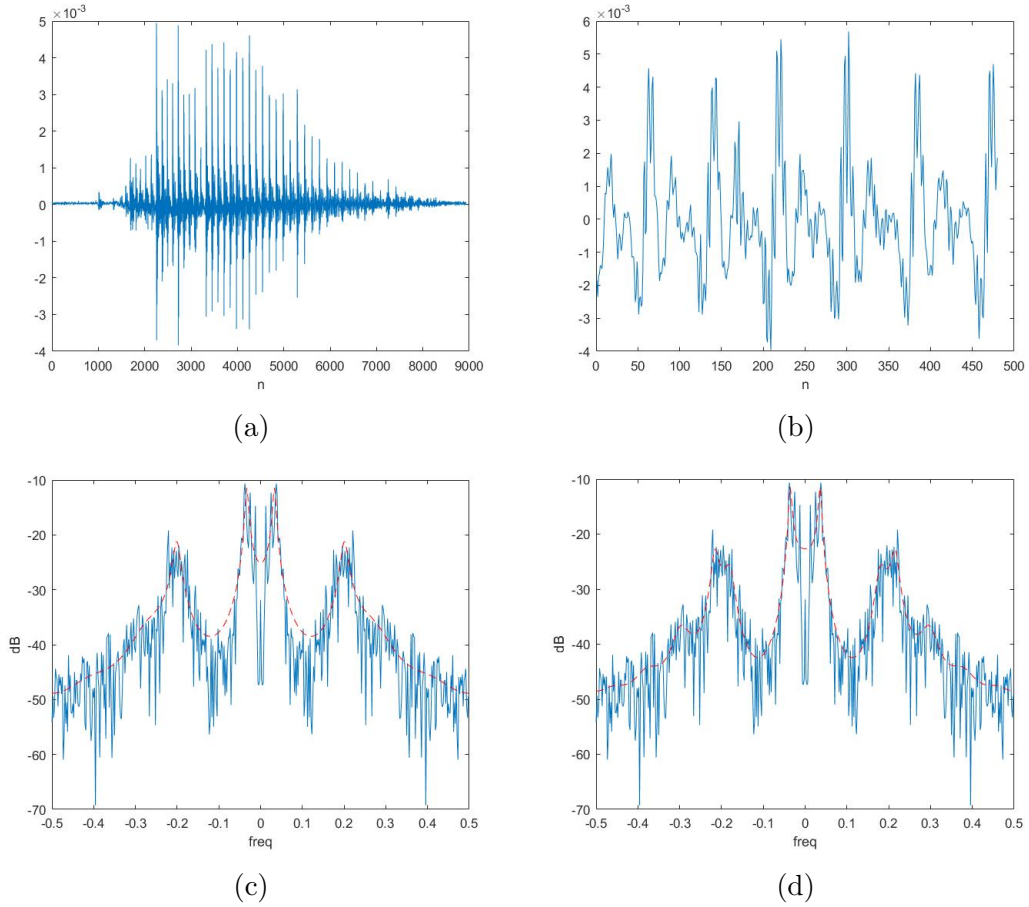


Figure 2.2: Example of a signal being modeled by an $AR(p)$ process

The PDF in equation (2.4) is also a likelihood function; the parameters that make it most likely will maximize it. As mentioned before, the computation of this likelihood function requires a lot of processing time due to the large number of computations needed to invert the covariance matrix. An asymptotic log form can be derived from equation (2.4) [12], and takes the form of

$$\ln(p_{\boldsymbol{X}}(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(P_X(f)) + \frac{I(f)}{P_X(f)}\right) df$$

Where $I(f)$ is the PSD. Since this thesis is concerned with classification, a constant not affecting the PSD will not affect the results. Therefore, this log-likelihood function is shown to be equivalent to [15],

$$\ln(p_{\boldsymbol{X}}(\boldsymbol{x})) = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{I(f)}{P_X(f)} - \ln \frac{I(f)}{P_X(f)} - 1 \right) df$$

This takes the form of the Itakura-Saito distance [6], a distance measure between two spectra. For this thesis we will assume the PSD $P_X(f)$ is normalized and the periodogram $\bar{I}(f)$ is also normalized leading to the final form of the test statistic,

$$\ln(p_{\boldsymbol{X}}(\boldsymbol{x})) = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{\bar{I}(f)}{P_X(f)} - \ln \frac{\bar{I}(f)}{P_X(f)} - 1 \right) df \tag{2.6}$$

where $\bar{I}(f)$ is given by,

$$\bar{I}(f) = \frac{\frac{1}{N} \left| \sum_{m=0}^{N-1} x[m] e^{(-j2\pi fm)} \right|^2}{\int_{-\frac{1}{2}}^{\frac{1}{2}} I(f) df} \tag{2.7}$$

The classifier is defined to be the maximum value calculated by the set of test statistics,

$$max(\ln(p_{\boldsymbol{X}}(\boldsymbol{x}))) \tag{2.8}$$

This classifier will be referred to as the Itakura-Saito, as to not create confusion

between it and the exact Gaussian. One last thing to notice is the Itakura-Saito(2.6) is a convex function taking the form,

$$-\left(\frac{x}{y} - \ln\frac{x}{y} - 1\right)$$

In this form the result is always less then or equal to zero, with equality only when $x = y$.

Since the Itakura-Saito is equivalent to the asymptotic form of the Gaussian likelihood function it should be the one that will give the best results when assuming a Gaussian noise distribution [15]. However, the performance of the Itakura-Saito is severely diminished when there are differences between the spectra obtained for the training data and the spectra obtained for the operational data [24]. Major differences occur in environments or when there are differences in the production of the data. A production difference in speech recognition will occur even with the same speaker. In this example, when the templates are made, the user is healthy with a nice clear voice. But if the user gets sick with a cold, then the voice changes drastically, due to a blockage of the nasal cavity. Environmental differences between the training data and observed data are largely due to noise and channel distortion. Noise can be introduced by the addition of unwanted signals, while a major source of channel distortion is multipath [5], [16], [25], [9]. To a lesser extent, pole placement has also been shown to affect the robustness of the Itakura-Saito classifier, [15].

2.3 Poisson Random Process

Another model for the PSD is a model based on a non-homogeneous Poisson point process [15]. It is not the intention of this thesis to study the Poisson process, that in itself has been the topic of many thesis's. However, some basic definitions are necessary in order to understand the rest of this thesis. The book *Random Point Processes* by Snyder gives a simple explanation of a Poisson process [23].

"A Poisson process is the simplest process with counting a random number of points."

"A random point process is a mathematical model for a physical phenomenon characterized by highly localized events distributed randomly in a continuum."

Some examples include lightning discharges, radioactivity and, seismic events. The Poisson model is also often described as arrivals entering a system [4], such as trains arriving at a station or people forming a line. These points/arrivals in time are an easy concept to understand. An analogy can be made for points/arrivals in frequency. This model was developed by Kay

in [15] to take the form of,

$$X[n] = \frac{1}{\sqrt{\lambda_0/2}} \sum_{k=-1}^{N_p} A_k cos(2\pi F_k n + \Phi_k) \qquad -\infty < n < \infty$$

where, A_K,Φ_k are IID random variables, the amplitudes are independent of phases, the number of sinusoids N_p is a Poisson random variable with mean λ_0, and F_k are the point events in frequency of a non-homogeneous Poisson random process where, $0 \leq A_K < \infty$, $0 \leq f \leq \frac{1}{2}$ and, $0 \leq \Phi_k < 2\pi$. Figure 2.3 provides an example.

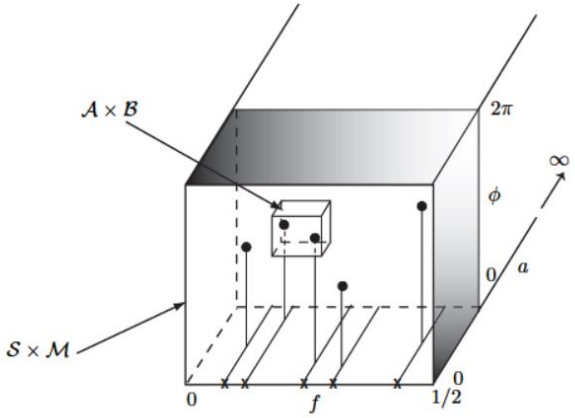


Figure 2.3: Illustration of an outcome of a marked Poisson process. [15]

For a more detailed look at this process see [15]. From this spectral representation Kay derived a new classifier called the *realizable Poisson likelihood function* or RPLF, which takes the form of,

$$max(l_R^{'}) = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_X(f)}\right) df \tag{2.9}$$

Mostly used to find the divergence between two PDFs, the Kullbeck-Liebler divergence [18] is commonly used by statisticians. As previously stated, the only difference between a PDF and a PSD/periodogram is that the PDF integrates to one. If the PSD/periodogram are normalized then Kullbeck-Liebler divergence becomes a good measure of the divergence between spectra [15]. Much like the Itakura-Saito likelihood function the Kullbeck-Liebler divergence is greater then or equal to zero, with equality only when $\bar{I}(f) = P_X(f)$.

2.4 Classification

Classifiers that have been defined in equations (2.4) ,(2.6), and (2.9) are the exact Gaussian likelihood function, the Itakura-Saito (modified asymptotic Gaussian likelihood function) and the RPLF, respectively. Figure 2.4 is an example of a binary classification problem where there is a spectrum belonging to an unknown class that needs to be classified as one of two known classes of spectra. In practice the spectra of the known classes are estimated from template data and the spectrum of the unknown class is estimated from operational data. In this example, the known spectra are generated as the PSD of two second-order all-pole filters. The unknown data is generated by filtering a Gaussian random process with one of the filters. The purpose of the classifier is to correctly identify which filter generated the process.

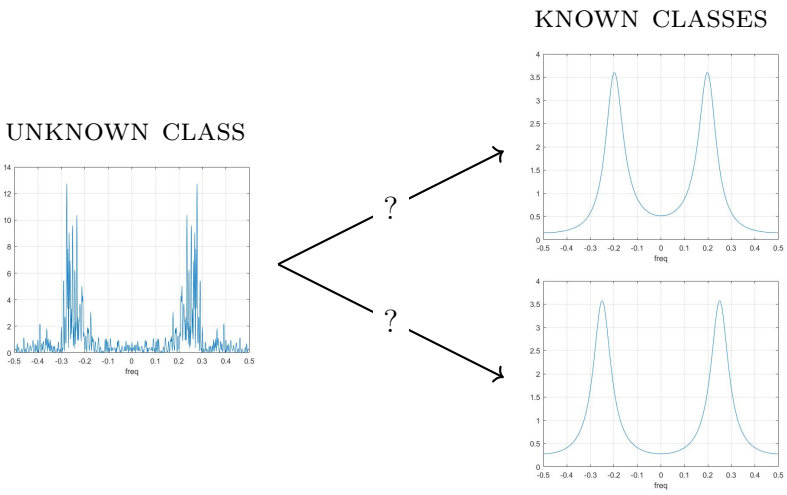


Figure 2.4: Classification of an spectrum

In Figure 2.4, if the filters that generated the process are different enough any classifier would be able to successfully identify which filter generated the observed process. Difficulties arise, when the distance between spectra is small. So how exactly does a classifier work? We start by labeling the known classes as seen in Figure 2.4. Class one represents the top spectrum and class two represents the bottom spectrum. Next we calculate the two normalized PSDs associated with those spectra, $P_{X1}(f)$ and $P_{X2}(f)$. To identify the spectrum in the unknown class belonging to the input data we employ the normalized periodogram $\bar{I}(f)$ equation (2.7). If the RPLF is the algorithm used for classification, the following two test statistics are calculated.

$$l'_{R1} = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_{X1}(f)}\right) df$$

$$l'_{R2} = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_{X2}(f)}\right) df$$

Once we have these results we will need to decide which one is the correct one. When both classes are equally probable, the spectrum under test is said to be classified according to the class associated with $max(l'_{R1}, l'_{R2})$.

2.5 Speech Data

Spectral analysis is a useful tool in a wide range of applications, including but not limited to medical, engineering, economics, and environmental data [3], [13]. One important area of study that has been getting much attention lately is automatic speech recognition. While the work presented in this thesis can be applied beyond the field of automatic speech recognition, speech data was chosen to be used. This is for three reasons,

1. There is a large amount of readily available data.

2. There is an established base of research and prior work in the field.

3. The $AR(p)$ process is one of the most successful models for speech data [19] [9].

There are many good resource texts on speech production and the linear speech production model such as *Linear Prediction of Speech* by Markel and Gray [19] and *Spoken Language Processing, A Guide to Theory, Algorithm, and System*

Development by Huang, Acero, and Hon [9]. A brief overview of this work is presented in the next few paragraphs.

Pioneers in the field, Davis, Biddulph, and Balashek built an isolated digit recognition system for a single speaker at Bell labs in 1952. Since that time there has been incredible advancements in the technology so automatic speech recognition is no longer limited to Sci-Fi movies. It can be seen in everyday life. From phones to cars to personal assistants automatic speech recognition gives people an easy interface with computer systems. However, achieving a robust machine is still something that has not been realized [22], [10].

Speech is divided into two categories, voiced and unvoiced. Voiced sounds are produced by the vibration of the vocal cords and have a roughly regular pattern in their time and frequency structure [9], [11]. In contrast the vocal cords do not vibrate while producing unvoiced sounds. The smallest unit of speech is a phoneme. These are the perceptually distinct units of sound in language that distinguish one word from another. There are forty-four phonemes in the English language that can be divided into two types, consonants and vowels. A consonant is a phoneme that is articulated with the complete or partial closure of the vocal tract. A vowel is a phoneme articulated without major constrictions and obstructions.

What is heard by the ear is an acoustic pressure wave that starts with the contraction of the lungs. Referring to Figure 2.5 the air is pushed between the

vocal folds, through an area called the glottis and out through the vocal tract [9]. The vocal tract is a non-uniform, time varying acoustic tube. Changes in the vocal tract are mainly due to the lips, jaw, tongue, and velum; with the nasal cavity as an additional acoustic tube which generates sounds [11], [22]. The fundamental frequency of the voice originates in the vocal folds. The greater the vocal fold tension, the higher the pitch. As the time-varying components of the vocal tract are manipulated, speech is produced. The vocal tract is essentially an all-pole model consisting of a cascade of a small number of two pole resonators; with each resonance defined as a formant [19].

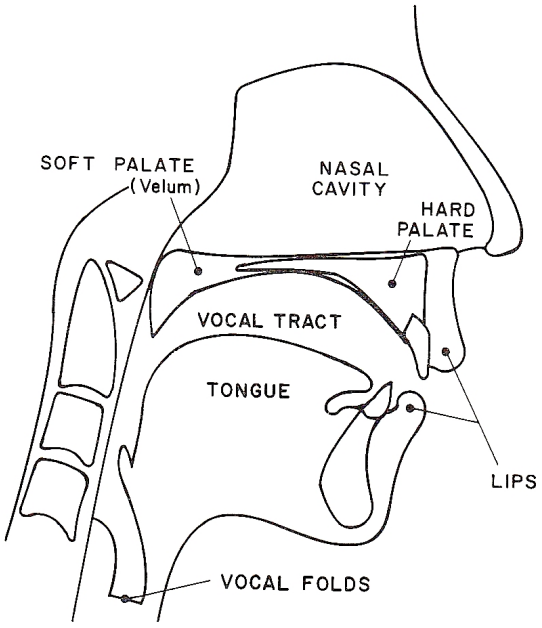


Figure 2.5: Cross sectional view of the human vocal tract showing the major anatomical structures used in speech production [19]

CHAPTER 3

SIMULATION AND ANALYSIS PLAN AND SETUP, METHODOLOGIES

3.1 Summary

Chapter 3 focuses on the plan and methodology employed to demonstrate that the RPLF is a robust classifier compared to other classifiers. As stated in chapter two, the large number of computations required to calculate the inverse covariance matrix of the exact Gaussian likelihood function renders it impractical. Therefore, this work compares the RPLF and the Itakura-Saito. During simulation a comparison to the exact Gaussian likelihood function is done, and this will be presented in chapter 4. The first part of this chapter describes the plan for simulations and how these simulations were performed. The final portion takes a look at the data used and provides a plan for the analyses using the data.

3.2 Simulation

In research the outcome usually cannot be anticipated. With this in mind, the initial simulations are simple binary classifications using $AR(2)$ processes. Simple simulations like these may be analytically explained if the results appear to be incorrect or unexpected. Initially, the simulations have no added noise or channel distortion. A flow chart is presented first with each step described in

detail after.

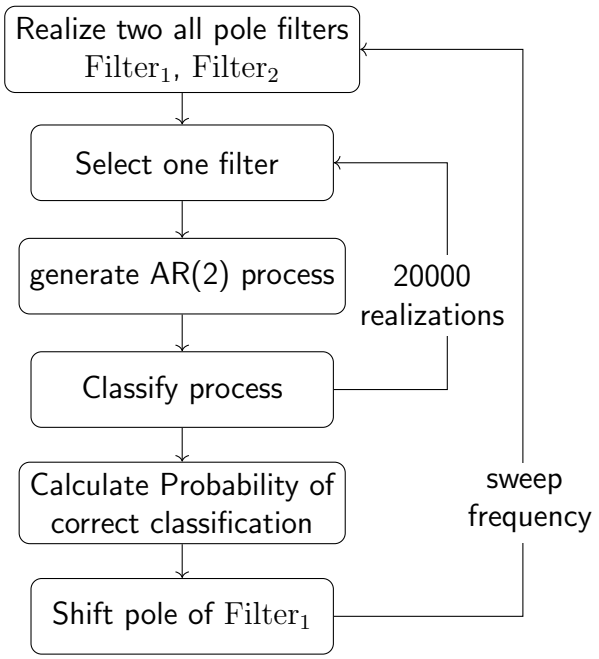


Figure 3.1: Flow chart for initial simulations

An $AR(2)$ process uses a second order infinite impulse response (IIR) filter with coefficients $a[1]$ and $a[2]$ and frequency respose $H(f)$,

$$\begin{aligned} x[n] &= u[n] - a[1]x[n-1] - a[2]x[n-2]; \\ H(f) &= \frac{1}{1 + a[1]e^{-j2\pi f} + a[2]e^{-j4\pi f}} \\ P_X(f) &= \frac{\sigma_u^2}{|1 + a[1]e^{-j2\pi f} + a[2]e^{-j4\pi f}|^2} \end{aligned} \tag{3.1}$$

The frequency has been normalized, resulting in a range from 0 to 1, around the unit circle. Figure 3.2 is a pole-zero map for a second order all-pole filter, with poles at radius r and angles θ where $\theta = \pm 2\pi f$.

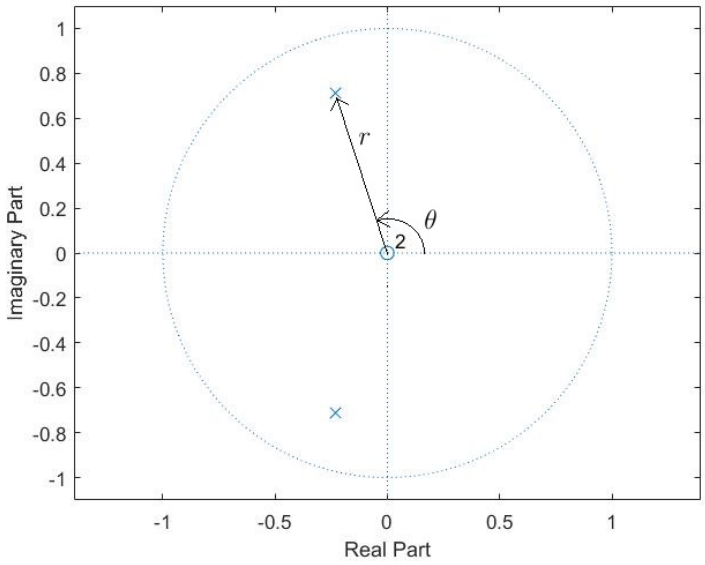


Figure 3.2: Pole-Zero plot for an AR(2) process. $f = 0.30$, $r = .7$

The coefficients are found using,

$$\begin{aligned} a[1] &= -2rcos(2\pi f) \\ a[2] &= r^2 \end{aligned} \tag{3.2}$$

Two all-pole filters were designed with the same radius and a frequency difference of $f_1 - f_2 = 0.05$. One of the filters was selected to generate an AR process using the first equation in (3.1), with each filter having an equal probability of being selected, $p(f_i) = \frac{1}{2}$, $i = 1, 2$. The first 200 samples were discarded, allowing the process to get past the correlation time and become WSS. The classifiers were calculated using equations (2.6) and (2.9). If the correct spectrum was selected by the Itakura-Saito, a counter was incremented. The same was done

for the RPLF using a separate counter. The inner loop of selecting filters and classifying AR processes was repeated for twenty thousand realizations. After the twenty thousand realizations were completed, the probability of correct classification was calculated by dividing the total number each classifier correctly classified by the number of realizations. Then the frequency of the second filter was shifted by $\Delta f = 0.01$ and the entire process was repeated. This procedure was completed for a total frequency shift of 0.1, so $(f_1 - 0.05) \leq f_2 \leq (f_1 + 0.05)$. The next step was to compare the performance of the classifiers as the signal-to-noise (SNR) was decreased. In practical applications, there are many types of noises which affect the system in different ways. Examples include: audible noise, electromagnetic interference and light noise. In an automatic speech recognition (ASR) system the template data may be collected in a quiet setting such as one's home but the ASR system is used in a noisy environment such as in a car or restaurant. In this thesis, noise is modeled by a zero mean independent WSS Gaussian random process $\sim N(0, \sigma^2)$, where "$\sim$" specifies, distributed according to.

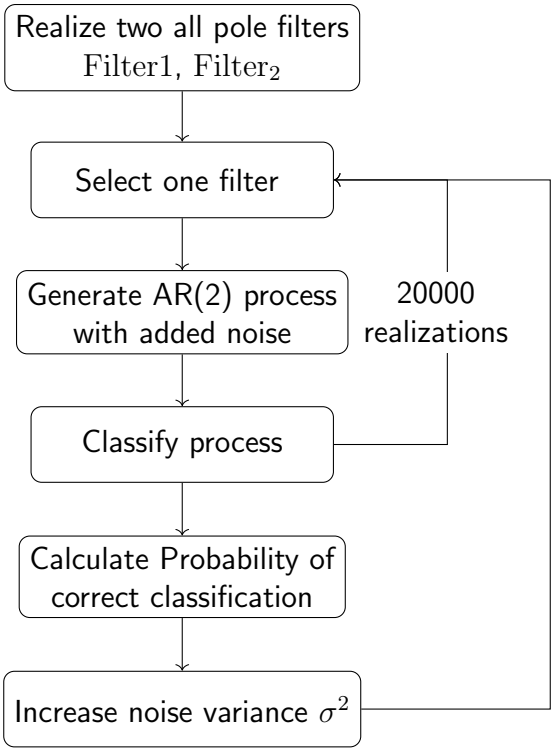


Figure 3.3: Flow chart for added noise simulations

Since the signal is zero mean the average power in a signal can be defined as the expected value of $x^2[n]$. Therefore, the average power in the signal can be found by computing the autocorrelation sequence at lag zero. Another method that leads to the same result can be seen by noticing that the inverse Fourier transform of the PSD is the autocorrelation sequence. This highlights the important relationship between the autocorrelation sequence, the average power, and the PSD. Since the noise is zero mean WSS Gaussian, the power in

the noise is simply the variance or σ^2. The signal to noise ratio is defined as,

$$SNR = \frac{Signal\ Power}{Noise\ Power} = \frac{r[0]}{\sigma^2}$$

in dB

$$SNR = 10log_{10}(\frac{r[0]}{\sigma^2})$$

This simulation was done much like the previous simulation. Modifications include.

1. The filters were designed and fixed.

2. The noise variance was increased, decreasing the SNR.

The final simulations were performed to evaluate the addition of channel distortion to the data. This channel distortion takes the form of multiple paths or multipath. Multipath occurs when the signal has more than one path to the sensor. Under these conditions, the signal is attenuated and time delayed. When it is thought of as a filter, it has an impulse response of,

$$h[n] = \sum_{k=0}^{\infty} \frac{\rho_k}{r_k} \delta[n - T_k] \tag{3.3}$$

where r_k is the distance to travel and ρ_k is the combined attenuation of the kth reflected sound wave [9]. Upon examining equation (3.3), multipath is simulated by adding zeros. A source of multipath is the sound emanating from the human

mouth, combined with the delayed sound emanating from the nose. This type of multipath adds pair of zeros to the model.

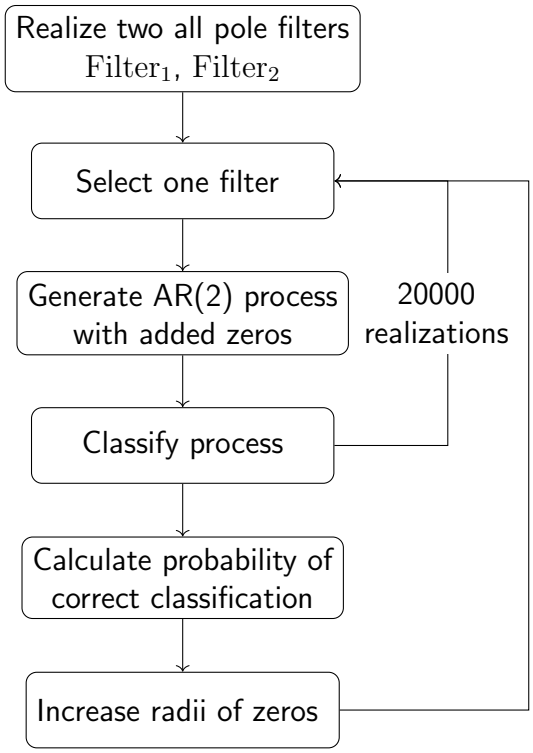


Figure 3.4: Flow chart for added multipath simulations

The PSD $\hat{P}_X(f)$ for the templates was calculated in the same way as previous simulations. Next, two all-pole filters were designed with two sets of zeros; these zeros were designed to have the same frequency as the poles of the filters. One of the sets was selected at random with a probability of $\frac{1}{2}$ for each. An $AR(p)$ process is a process that uses an all-pole filter, the added zeros create an autoregressive - moving average process ($ARMA$). An $ARMA$ process has

a z-domain representation of,

$$H(z) = \frac{X(z)}{U(z)} = \frac{1 + b[1]z^{-1} + b[2]z^{-2}}{1 + a[1]z^{-1} + a[2]z^{-2}}$$

Taking the inverse z-transform gives the result,

$$x[n] = -a[1]x[n-1] - a[2]x[n-2] + u[n] + b[1]u[n-1] + b[2]u[n-2]$$

where $a[1]$ and $a[2]$ are the all-pole filter coefficients designed in step one and $b[1]$ and $b[2]$ are the coefficients designed to place the zeros. These zero coefficients were also designed using equations (3.2). The radii of the zeros was increased the same way the noise power was increased in the previous simulation. Each set of zeros starts at the origin and increases radius at the same frequency until it reaches the pole. Figure 3.5 shows one set of the pole-zero combination.

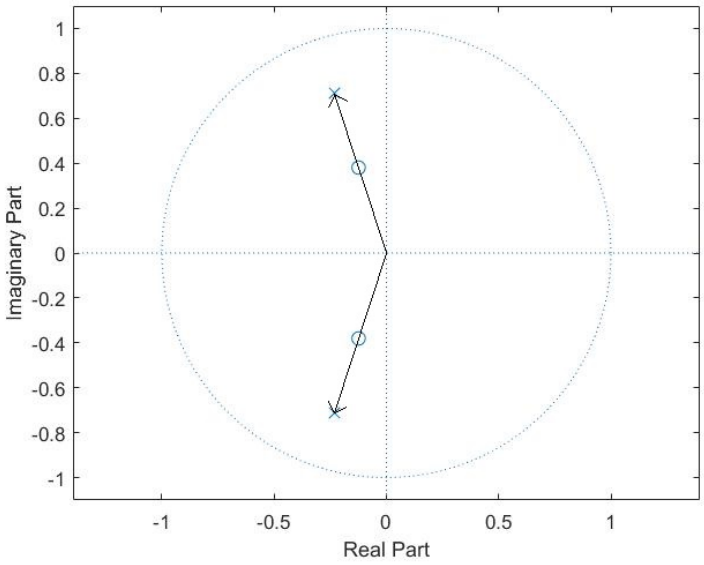


Figure 3.5: Pole-Zero plot for an $AR(2)$ process with added zeros

3.3 Data Analysis

Once the simulations were completed and the results looked reasonable it was time to introduce real world data: data that is simple and easy to divide up. There are many types of data that are well suited to demonstrate the potential of the RPLF, medical, economic and chemical to name a few. Speech data also has the desired characteristics and is readily available. The ISOLET database consists of spoken letters, or more precisely the names of the letters. In the database there are 150 people, 75 female, and 75 male, saying each letter two times. At the top level, the database is broken up into 5 folders, isolet1 - isolet5, Figure 3.6 top. Each one of these 5 top-level folders itself contains 30 folders, 15 male and 15 female speakers, Figure 3.6 middle. Each of the people folders

contain two utterances of each letter, Figure 3.6 botton. The structure of the database is outlined in Figure 3.6.

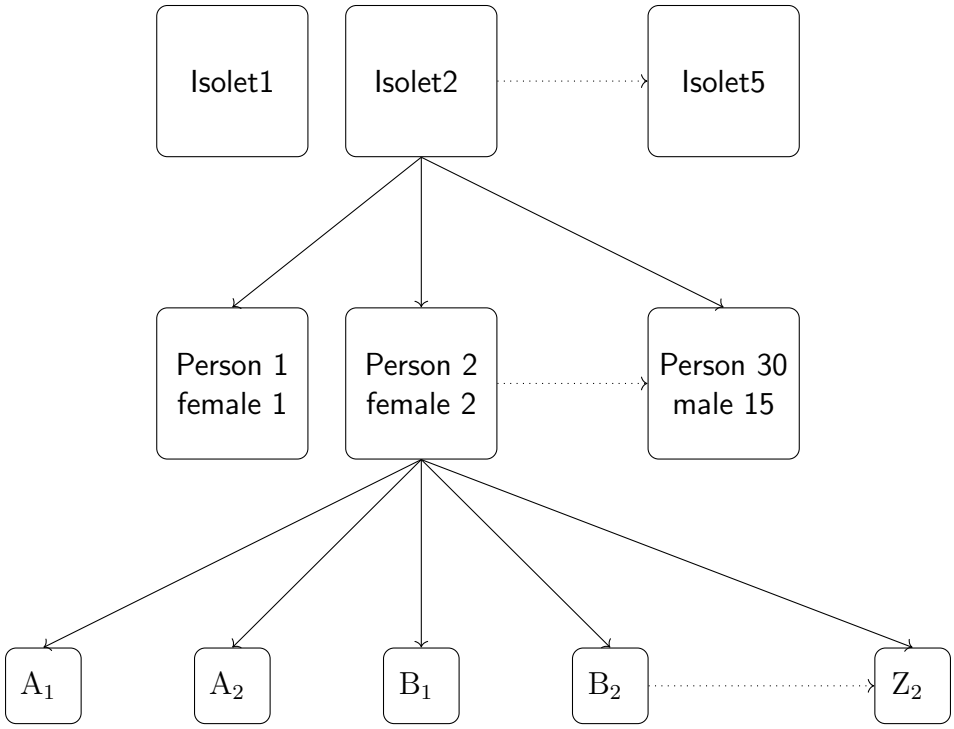


Figure 3.6: Folder structure for ISOLET data base

The write-up that came with the database describing the data is in appendix B [1].

The folders containing the speech are named in the following manner. The first letter is the gender, "m" for male and "f" for female. Next are the persons initials, either 2 or 3 letters. The folder ends with a 0 or 1 depending on whether there is a previous folder with the same beginning name. The names of the files containing the utterances begin with the folder name, followed by the letter spoken, and a 1 or 2 depending on whether it is the first or second

utterance of that letter. All files end with a -t. For example, a female would the have the folder name “fgh0” and the first “A” utterance file for that same female would be “fgh0-A1-t”.

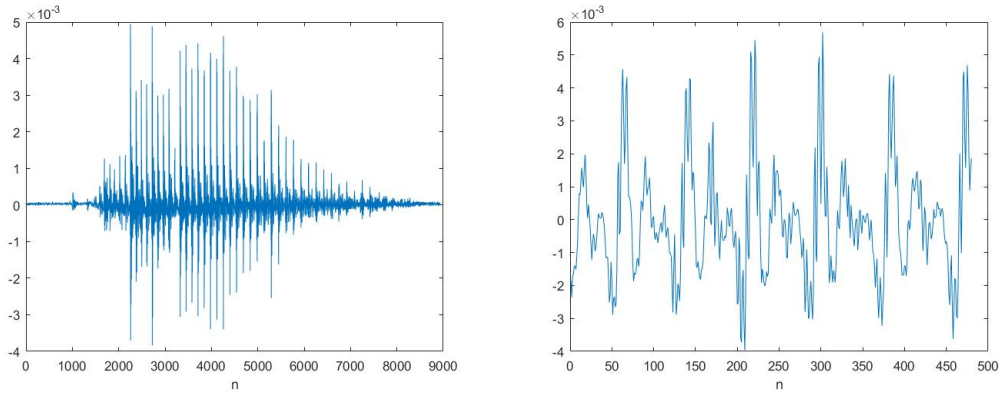
The creators of the database calculated the SNR for the database. An exact description of how it was calculated can be found in the information accompanying the database [2]. The mean SNR is 31.5dB with a standard deviation of 5.6dB.

The analysis of the data for this thesis followed the same basic structure as the simulations. First a baseline analysis was completed. Second, noise was added to the data. Then finally, channel distortion was added to the data. Although the analysis procedure was more complex than the procedure for the simulations, the initial flow chart only consists of four blocks. The first block would be to create templates, then import data, classify vowel and finally calculate the probability of correct classification. To include noise or multipath, two additional blocks were added. The first block adds the noise or multipath to the data. The second block alters the amount of noise or shifting of the zeros.

Before the analysis could begin, the data needed to be divided up into the training data and the observed data and then the templates made. In order to keep the analysis simple five letters were chosen, A, E, I, O and, U. Prior research commonly divides data up into 10-30 ms intervals [8]. After dividing up the data into various lengths up to 50ms and performing the classifications it was

decided to use 30 ms intervals. Lengths longer then 30 ms only minimally improved the classification while shorter lengths did not perform as well. A data length of 30 ms would equate to 480 samples per letter.

In order to extract a WSS sample, the midpoint of the data set was chosen and then a 30ms sample was taken. If the sample $s[n]$ is taken from the data $x[n]$ then $s[n] = x[n], n = (mean-240), (mean-239), ...(mean+239)$. For example, a letter has 7000 data points, with a midpoint of 3500. Samples $3260 \leq n \leq 3739$ would be extracted. Figure 3.7 is an example of the letter "A" spoken by a male for both the entire letter, Figure 3.7a, and the 30ms sample, Figure 3.7b.



(a) (b)

Figure 3.7: Example of the letter "A"

The data in Figure 3.7b is a good example of data that can be used to demonstrate the potential of the RPLF.

Once the analysis data was chosen, the next step was to design the templates. In the simulations the templates were the power spectral densities of all-pole filters, which is a simple task to estimate. Designing the templates with the

speech data was a much more complicated process. The following steps were used,

1. **Choose the data to be used for the templates and the data to be for the observed, or test data** - It was desired to make the analysis speaker independent. Therefore, the first four folders, isolet1 - isolet4, were used for template data and the isolet5 folder data was the observed data. There are 30 people in each folder and 2 utterances of each letter per person which equals 240 utterances of each letter for the templates. This would leave 60 utterances of observed data for each letter to perform the classification with.

2. **Choose the number of $AR(p)$ coefficients** - In order to model the first three to five format peaks, an eighth to fourteenth order model is typically used [11]. Analysis of the data was tried with $p = 8, 10, 12$ and 14. An order of twelve was chosen because fourteen lead to an minimal increase but did moderately better then ten.

3. **Calculate the $AR(p)$ coefficients** - Using the autocorrelation method described above, the autocorrelation sequence was calculated for each utterance, leading to the formulation of the autocorrelation matrix described in equation (2.5). The $AR(p)$ coefficients were solved using the Levinson recursion, which led to 240 AR models, one for each utterance.

4. **Calculate the templates** - The 240 models were averaged together to create the $AR(p)$ model for each letter, equation (3.4). PSD templates were calculated from the averaged $a[k]$ parameters. Figure 3.8 represents the power spectral densities for the five templates.

$$a_t[k] = \frac{1}{M} \sum_{m=0}^{M-1} a_d[m,k] \tag{3.4}$$

where $a_t[k]$ are the averaged linear prediction coefficients for the five letters, $a_d[m,k]$ are the linear prediction coefficients for each of the spoken utterances, and $M = 240$. The PSDs for the five templates are shown in Figure 3.8

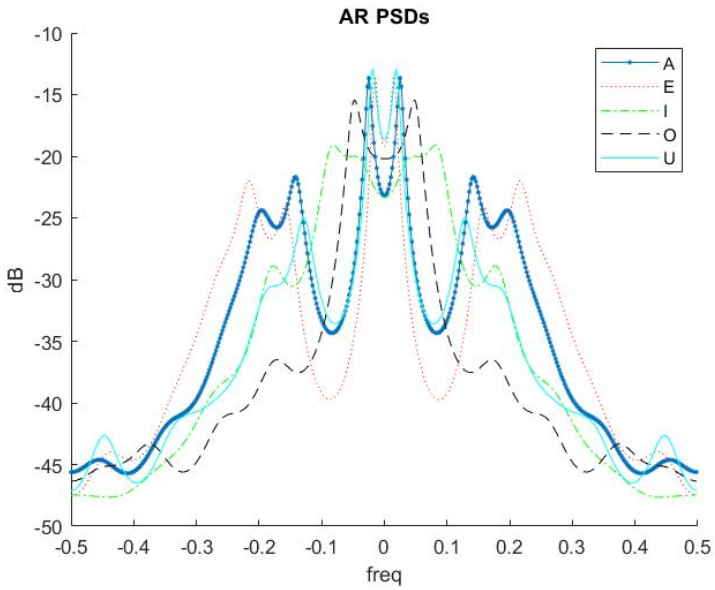


Figure 3.8: Power spectral densities for calculated templates

The initial analysis used equations (2.6) and (2.9) to classify the vowels in the

observed. This included five test statistics each for Itakura-Saito and the RPLF. Then the classifier identified the spoken letter based on the maximum of those test statistics.

$$l'_{Ri} = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_{Xi}(f)}\right) df \quad 1 \leq i \leq 5$$

Where $P_{X1}(f)$ is the PSD for the "A" template, $P_{X2}(f)$ is the PSD for the "E" template and so on. For example, if the observed utterance was an "E" each of the test statistics calculate an output value. The letter is classified as the letter corresponding to the maximum value of the output of those five test statistics, hopefully, an "E".

The analysis kept track of the observed data letter and the result of the classification. The final results are displayed in a confusion matrix. Because the confusion matrix displays the true value against the classified value, the more diagonal the matrix is, the better the results. In this analysis, a perfect matrix would have 60s on the main diagonal because there are 60 utterances of observed data for each letter.

Figure 3.10 is a flow chart that describes the method used to import and processing the observed data. First the data is imported from the file. Next, the imported data is multiplied by a 30ms rectangular window. After windowing, the autocorrelation sequence is estimated from the windowed data. From the autocorrelation sequence the $AR(p)$ linear prediction coefficients are calculated.

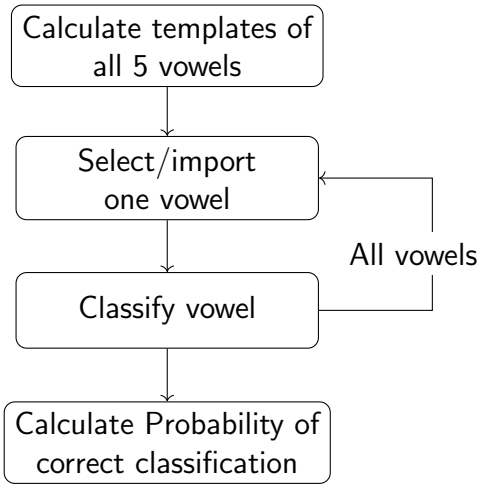


Figure 3.9: Flow chart for vowel classification

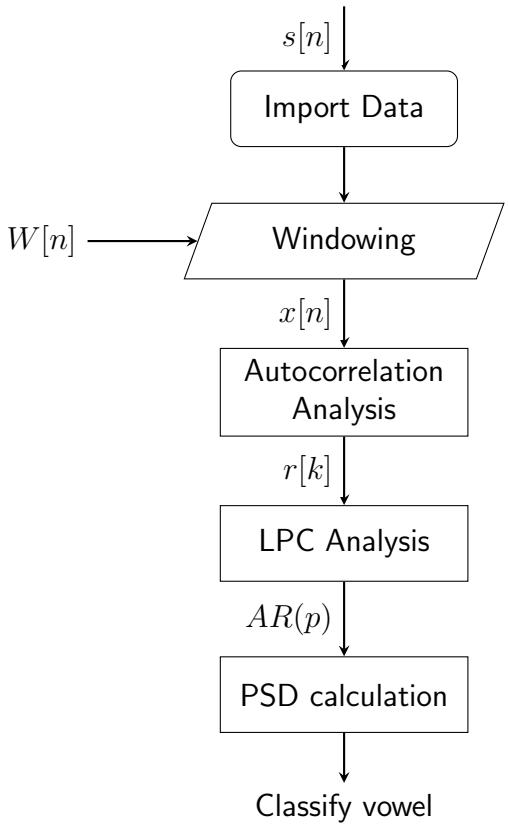


Figure 3.10: expanding Select/Import block of Figure 3.9

The next couple of analyses repeated the above process but added noise or channel distortion to the data. The analysis including noise was accomplished by adding zero-mean independent WSS Gaussian noise with variance σ^2, while keeping the SNR constant. This can be done two ways, leading to the same result. The first is by calculating the signal power for each utterance and then adjusting the noise power. This is done by calculating the autocorrelation sequence, with the average power autocorrelation at zero lag. Then the noise power is adjusted to keep the same SNR. A second option is to normalize the

signal power in each utterance and then calculate the appropriate noise power based on the desired SNR. The normalization is calculated by,

$$x_n[n] = \frac{x[n]}{\sqrt{\frac{1}{N}\sum x^2[n]}}$$

where $x_n[n]$ is the normalized $x[n]$. This analysis used the second option. Channel distortion was added by including a pair of zeros where it had an effect on the PSD's. Referring to Figure 3.8, zeros placed at a frequency of ± 0.2 would have a large effect because it has the effect of diminishing the observed peaks in the frequency response. Depending on the zero radius, the zeros would cancel out some of the power around this frequency. In contrast, if the zeros were placed at a frequency of ± 0.4 they would have very little effect, if any at all. Then the signal power was again normalized because of the added zeros. The data we have selected is speech data and a classifier used in many modern ASR systems that has shown good results is a Euclidean distance measure of the linear prediction cepstral coefficients (LPCC) [22]. So we will compare the RPLF to the LPCC classifier as well. The cepstrum is the inverse Fourier transform of the log magnitude of the Fourier transform of a signal.

$$c[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \ln |X(f)| e^{j2\pi f n} df$$

There are a many reasons the cepstrum is used over other transformations. Two

of the most important are first, the cepstrum provides source-filter separation. Second, cepstral coefficients provide a compact representation of the spectral envelope. [7]. There are other transformations that have these same qualities, but the LPCC better models speech than other models [6]. Given a set of linear prediction coefficients, $a[k]$, the linear prediction cepstrum coefficients are derived from the following equations.

$$\hat{c}[n] = \begin{cases} 0 & n < 0 \\ -ln(a[n]) & n = 0 \\ -a[n] - \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) \hat{c}[k] a[n-k] & 0 < n \leq p \\ -\sum_{k=n-p}^{n-1} \left(\frac{k}{n}\right) \hat{c}[k] a[n-k] & n > p \end{cases} \tag{3.5}$$

This leads to the test statistic,

$$CCTs = \sum_{i=0}^{N-1} (\hat{c}_t[n] - \hat{c}_o[n])^2 \tag{3.6}$$

where $\hat{c}_t[n]$ is the template LPCC and $\hat{c}_o[n]$ is the observed data LPCC. This leads to a classifier $min(CCTs_i)$.

The next analysis was a comparison of the cepstrum classifier, the Itakura-Saito classifier and the RPLF, with noise added to the data. This analysis followed the same procedure as the simulation with added noise, shown in Figure 3.11.

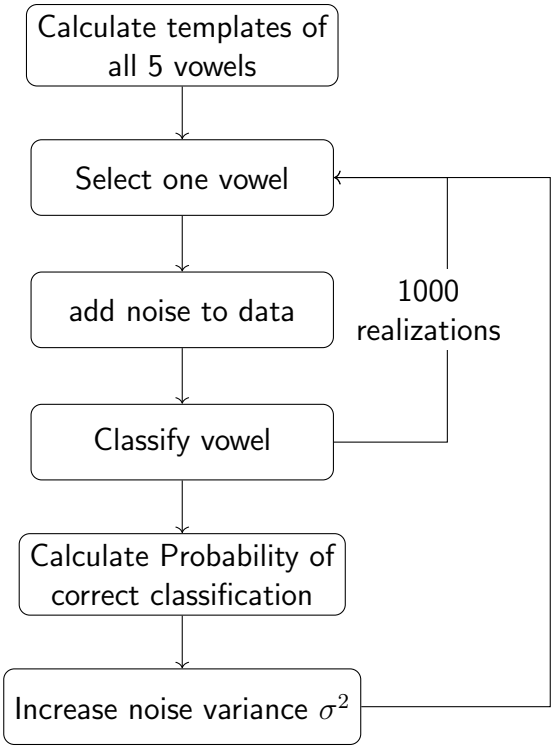


Figure 3.11: Flow chart for vowel classification with added noise

Unlike the first few data analyses, in this analysis the probability of correct classification was not kept for each individual letter, but for all results. Therefore, the results are not presented in a confusion matrix but in a graph of probability of the correct classification versus signal to noise ratio in dB.

In the final analysis a pair of zeros was added to simulate multipath in the data. The procedure combined the flowcharts in Figures 3.4 and 3.11.

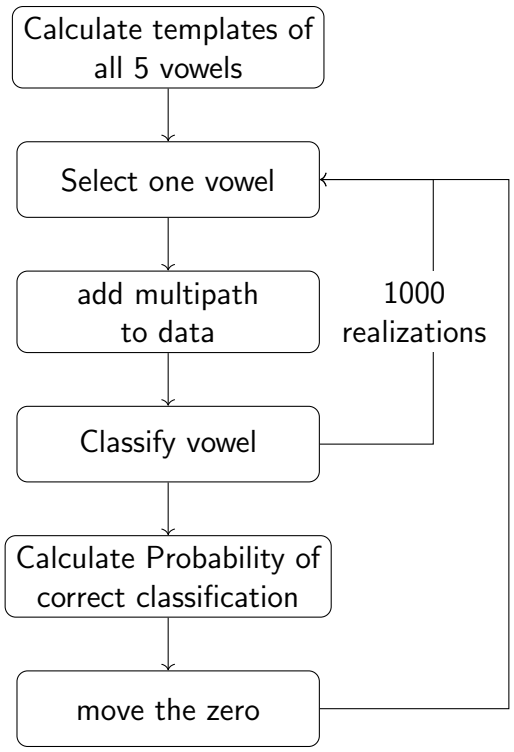


Figure 3.12: Flow chart for vowel classification with added multipath

The analysis of including a zero to simulate multipath was completed two different ways. The first consisted of the zero moving out along a radius with a fixed frequency, the same as in the simulation. In the second the zero was swept through the frequency while keeping the radius constant.

CHAPTER 4
SIMULATION AND ANALYSIS RESULTS

4.1 Summary

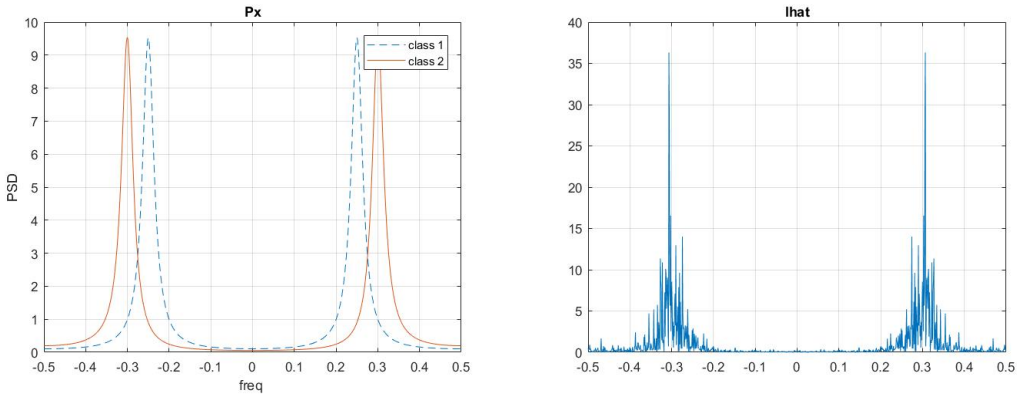
This chapter will begin by presenting the simulation results, and the data analysis will follow. Simulation or analysis checks will also be included to lend credence to the accuracy of the procedure and final results. Detailing this process is important because there is an unexpected result that initially appears to be incorrect. The chapter will conclude with a brief look at the templates created from the known data, and present the results of classifications performed with modified templates that increase the probability of correct classification. For all simulations, the poles of the AR(p) processes will be placed inside the unit circle for stability.

4.2 Simulation Results

The initial simulation was kept very simple for two reasons. The first reason was to be able to validate the algorithm by checking expected results It is expected that the the Itakura-Saito would outperform the RPLF in the absence of noise or channel distortions. This is because the simulation data is independent and identically distributed WSS Gaussian noise being filtered by a linear all-pole filter. Therefore, the output will be Gaussian. Since

the Itakura-Saito is derived from an asymptotic equivalent of the likelihood function for a Gaussian random process it should outperform the RPLF. It is also expected that when the poles of the two possible filters are right on top of each other, the probability of correct classification is one half, $P_{cc} = 0.5$. This is expected because then both processes are generated from exactly the same filter, so classification is a "50-50 shot". The second reason the initial simulation was kept simple is so there are not any assumptions made that may affect the results, whether known or unknown. Simple simulations utilizing WSS Gaussian noise and a simple filter ensures the exclusion of anything unknown in the data.

The first simulation was performed using the procedure presented in Figure 3.1. Figure 4.1a shows the PSDs used in the test statistics for this simulation with Figure 4.1b an example periodogram for the data to be classified. Figure 4.2 shows the output of the simulation with the number of samples equal to 250. The frequency f_0 is the frequency of the poles for the second filter, the filter where the poles are fixed.



(a) $f_0 = 0.25, 0.3$, radius $= 0.9$ (b) $f_0 = 0.3$,, radius $= 0.9$

Figure 4.1: Power spectral densities for initial simulations, (4.1a) for the all-pole filters, (4.1b) for one instance of the AR(p) process.

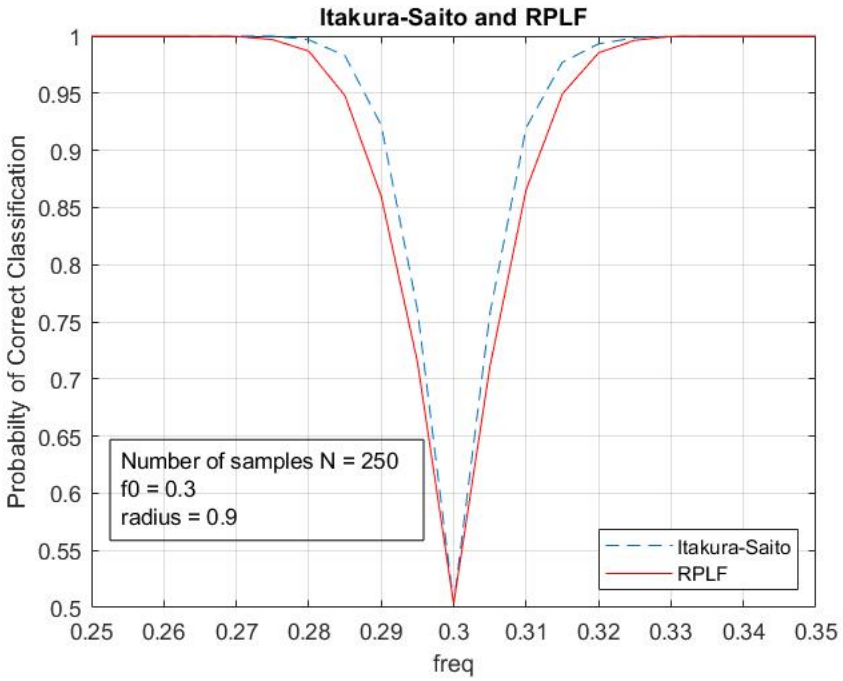


Figure 4.2: Simulation result $AR(2)$, $f_0 = 0.3$, radius $= 0.9$, N$= 250$

This result are exactly as expected, the Itakura-Saito outperformed the RPLF.

Also, the probability of correct classification is one half when the poles are

exactly the same. The RPLF did very well, giving hope that it will outperform the Itakura-Saito when noise or channel distortion is added. However, it is difficult to differentiate between the two classifiers because there is a 100% probability of correct classification by the time the two peaks are separated by a frequency of 0.03. In order to get results that may better distinguish between thee classifiers the number of samples was decreased to 50 and the simulation was run again, results presented in Figure 4.3.

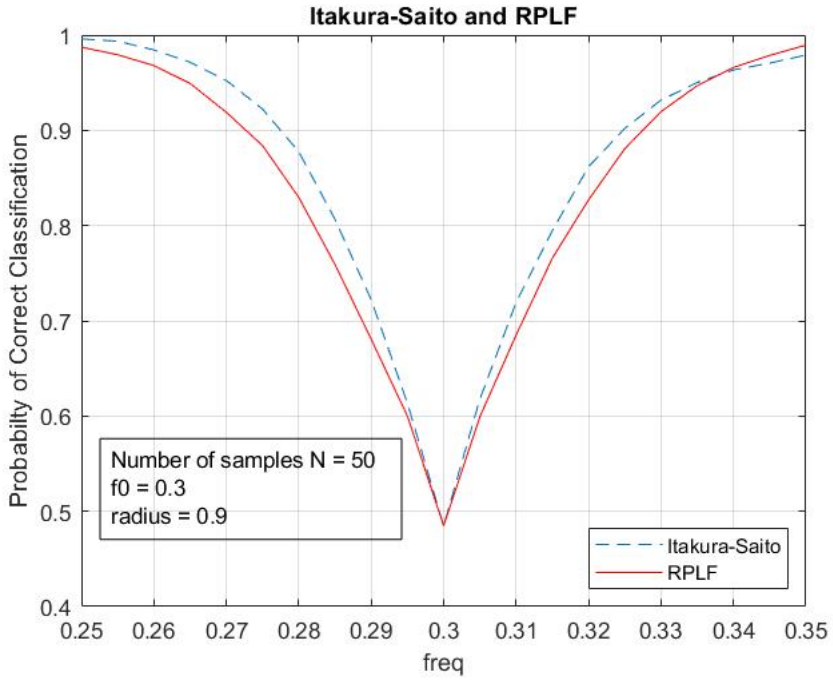


Figure 4.3: Simulation result $AR(2)$, $f_0 = 0.3$, radius $= 0.9$, N$= 50$

Restricting the number of samples to fifty yielded unexpected results: in the top right corner of the figure, when the pole crosses over around 0.34 the RPLF starts to outperform the Itakura-Saito. The next step was to determine if the algorithm or the data was responsible for the unexpected results. The simulation

was repeated and included the exact Gaussian classifier, equation (2.4), since the expected result of the exact Gaussian classifier is the true upper bound. This simulation is presented in Figure 4.4. This result is as expected, the exact Gaussian outperforms both the RPLF and the Itakura-Saito, indicating the data is generated correctly and the algorithm is functioning correctly.

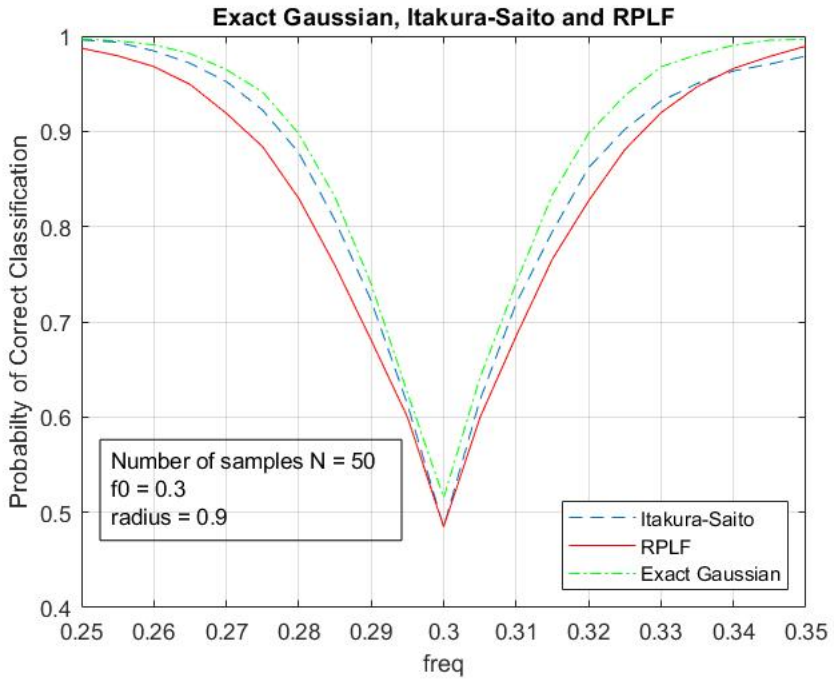


Figure 4.4: Simulation result $AR(2)$, $f_0 = 0.3$, radius $= 0.9$, N$= 50$

To further analyze the problem, if the number of samples was increased would the Itakura-Saito approach the exact Gaussian? For the next simulation the number of samples was set to 500 and results are presented in Figure 4.5.

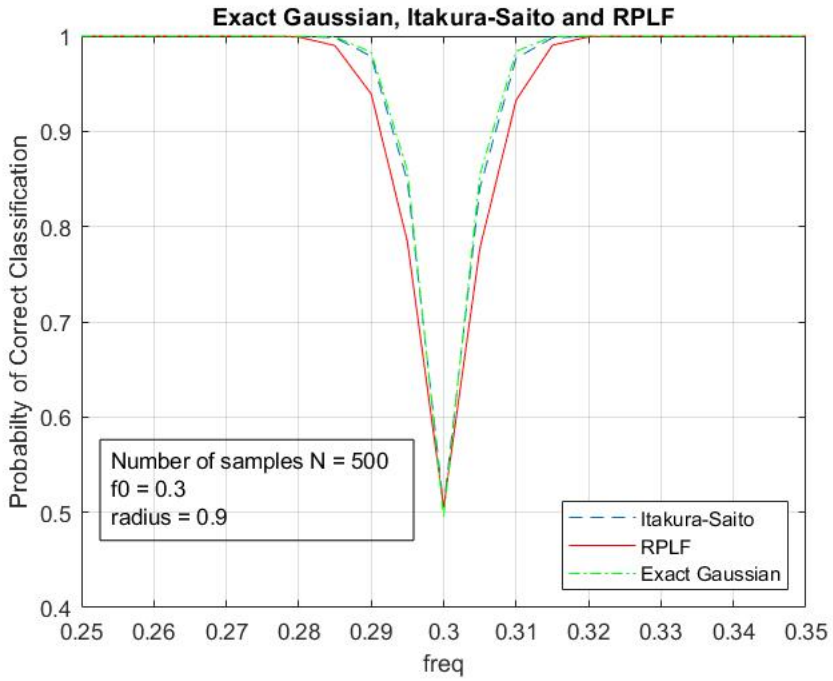


Figure 4.5: Simulation result $AR(2)$, $f_0 = 0.3$, radius $= 0.9$, N$= 500$

As the number of samples increases the Itakura-Saito does approach the exact Gaussian. This result also indicates the algorithm and the data are correct, so why does the RPLF outperform the Itakura-Saito? To investigate this further, a simulation using an $AR(1)$ process was completed since it is even simpler then the $AR(2)$ process employed so far. An $AR(1)$ process has a linear filter with a single pole where,

$$\begin{aligned} x[n] &= u[n] - a[1]x[n-1]; \\ H(f) &= \frac{1}{1 + a[1]e^{-j2\pi f}} \\ P_x(f) &= \frac{\sigma_u^2}{|1 + a[1]e^{-j2\pi f}|^2} \end{aligned} \tag{4.1}$$

The $AR(1)$ filter has a single pole and for a real process, that pole must be on the real axis. In this simulation, the first filter had a fixed pole location at $a[1] = -0.8$ while the pole from the second filter was swept from $-1 < a[1] < 1$. The output of the simulation is shown in Figure 4.6.

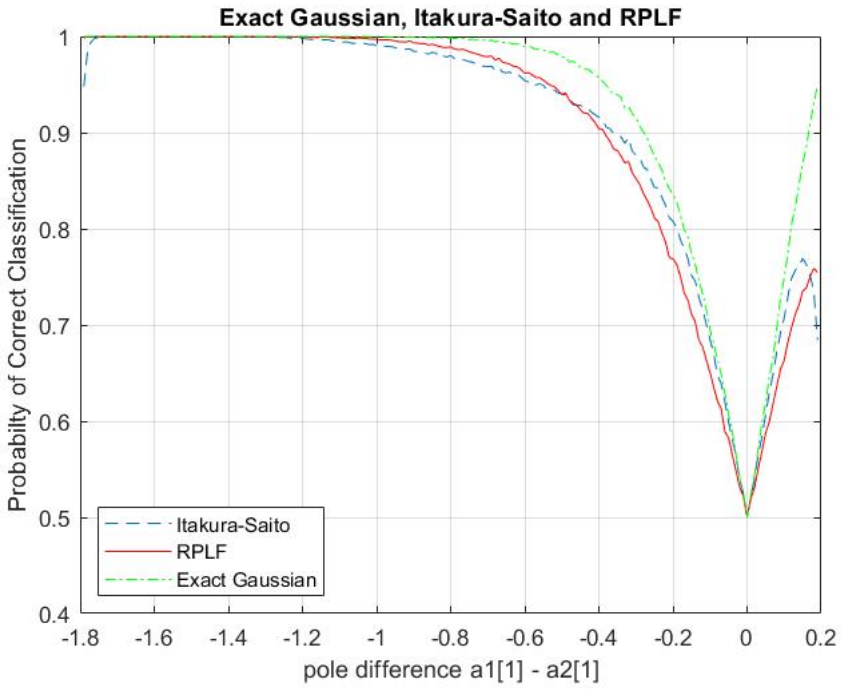


Figure 4.6: Simulation result $AR(1)$, $a[1] = 0.8$, radius $= 0.9$, N$= 50$

It is interesting to note how the Itakura-Saito drops off dramatically as the pole gets closer the unit circle. Due to the simplicity of an $AR(1)$ process, it is possible to go back to first principles and derive the Itakura-Saito and the exact Gaussian based only on pole location, $a[1]$, and the number of samples, N. The difference and a divergence are calculated using this basic information. The derivation for the divergence is in appendix (A), along with results

from [14], [12]. The derivation starts with the log-pdfs version of the exact Gaussian,

$$\ln(p_E(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(det(\boldsymbol{C})) - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{C}^{-1}\boldsymbol{x}$$

and the asymptotic Gaussian,

$$\ln(p_A(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}}\left(\ln(P(f)) + \frac{I(f)}{P(f)}\right)df$$

The PDFs were found in the desired form and are presented below. First the exact Gaussian,

$$\begin{aligned}\ln(p_E(\boldsymbol{x})) = &-\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma_u^2 + \frac{N}{2}\ln(1-a^2[1]) \\ &- \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] \\ &+ \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big]\end{aligned} \tag{4.2}$$

Then the asymptotic Gaussian

$$\begin{aligned}\ln(p_A(\boldsymbol{x})) = &-\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma_u^2 \\ &- \frac{1}{2\sigma_u^2}\Big[x^2[0] + x^2[1] + 2a[1]x[0]x[1] + a^2[1]x^2[0] \\ &+ \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 + a^2[1]x^2[N-1]\Big]\end{aligned} \tag{4.3}$$

And the difference,

$$
\begin{aligned}
&= \ln(p_E(\boldsymbol{x})) - \ln(p_A(\boldsymbol{x})) \\
&= \frac{N}{2}\ln(1-|a[1]|^2) + \frac{1}{2}\Big[a^2[1]x^2[0] + a^2[1]x^2[N-1]\Big]
\end{aligned}
$$

The Kullback-Leibler divergence is the divergence between two PDFs. For this divergence we have used the exact Gaussian and the asymptotic Gaussian as PDFs. While the asymptotic Gaussian is not a true PDF, as it does not integrate to one, the Kullback-Leibler divergence may provide some useful insight. The derivation of the result is presented in appendix (A.2). The Kullback-Leibler divergence takes the form,

$$
\int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \ln\left(\frac{p_E(\boldsymbol{x})}{p_A(\boldsymbol{x})}\right) d\boldsymbol{x} \tag{4.4}
$$

where $p_E(\boldsymbol{x})$ is the PDF of the exact Gaussian and $p_A(\boldsymbol{x})$ is the PDF of the asymptotic Gaussian. After taking the exponential of (4.3) and (4.2) and substituting them into (4.4) the result is,

$$
= \frac{N}{2}\ln(1-a^2[1]) + \frac{a^2[1]}{(1-a^2[1])}
$$

Examining this result as $|a[1]| \longrightarrow 1$ the second term in the equation takes over and the divergence goes to infinity. Figure 4.7 plots the divergence for different number of samples N as $|a[1]|$ goes from $0 \longrightarrow 1$.

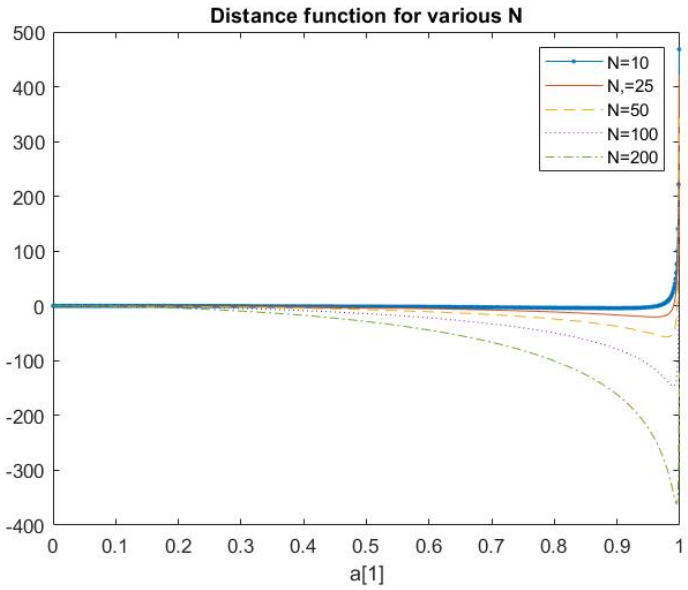


Figure 4.7: Divergence for the difference between asymptotic Gaussian and exact Gaussian

As discussed previously the Kullback-Leibler divergence can not be negative, however, the functions in the above figure go negative. This is explained by the fact that of the asymptotic Gaussian is not really a PDF. In order to be a true PDF, the asymptotic Gaussian would need to be normalized. Despite this limitation, the above figure shows that as the pole radius approaches the unit circle, the distance between the exact Gaussian and asymptotic Gaussian increases, especially for smaller sample sizes. This effect holds true for an $AR(2)$ process, as presented in the following figures. Figures 4.8 - 4.11 number of samples increases, while holding the radius constant. While in Figures 4.12 - 4.14 the radius of the zeros increases while holding the number of samples constant.

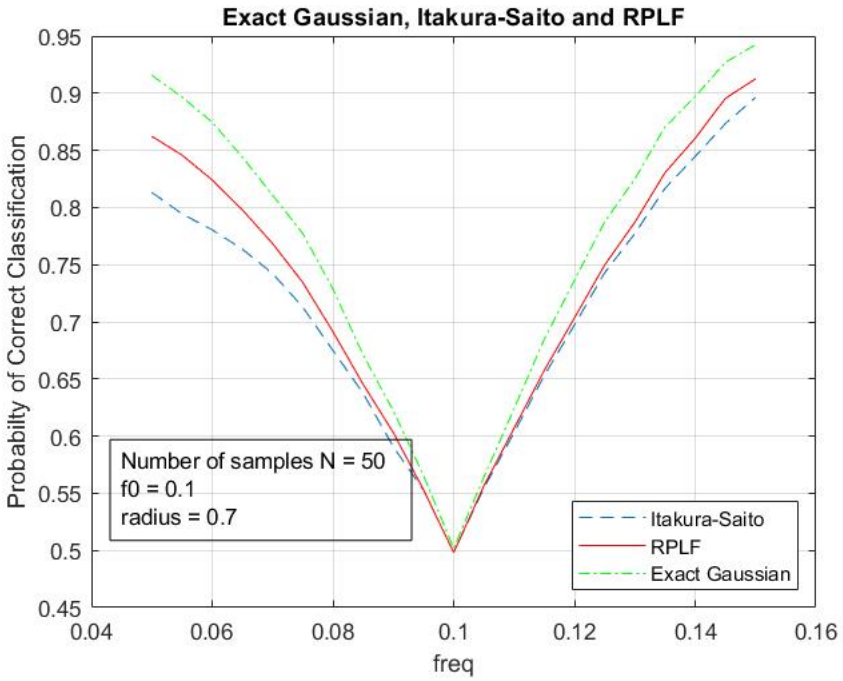


Figure 4.8: Simulation results, N = 50, $f_0 = 0.1$, radius = 0.7.

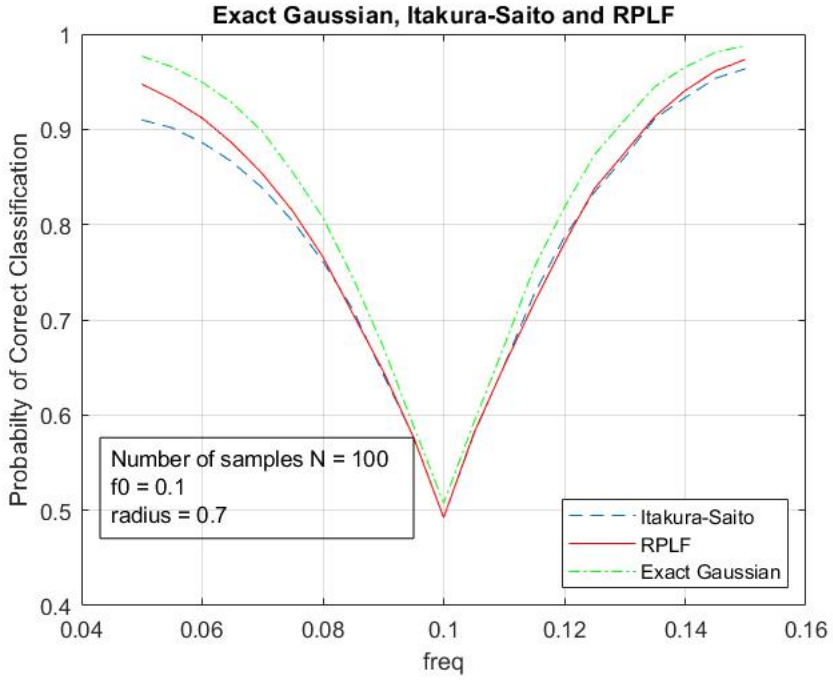


Figure 4.9: Simulation results, N = 100, $f_0 = 0.1$, radius = 0.7.

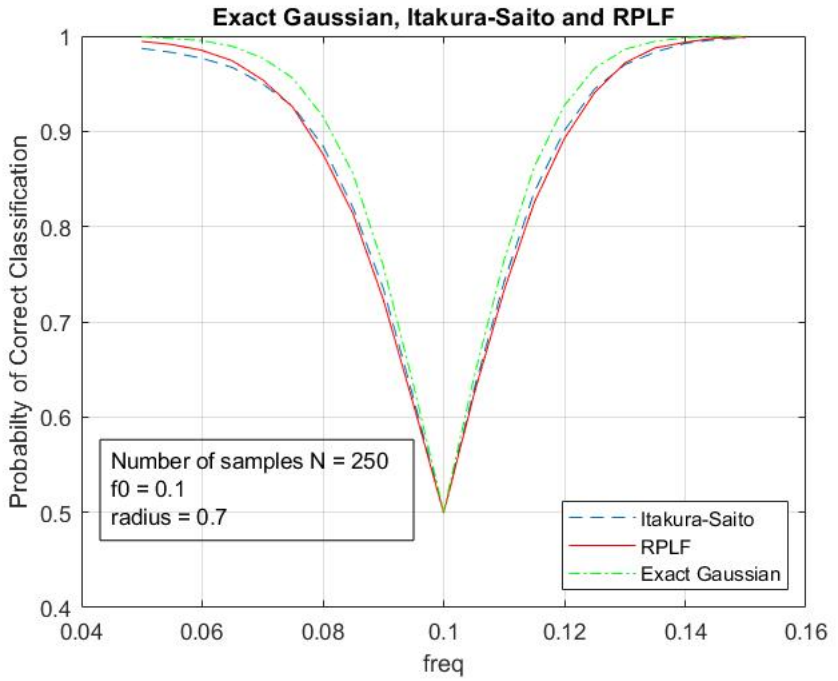


Figure 4.10: Simulation results, N = 250, $f_0 = 0.1$, radius = 0.7.

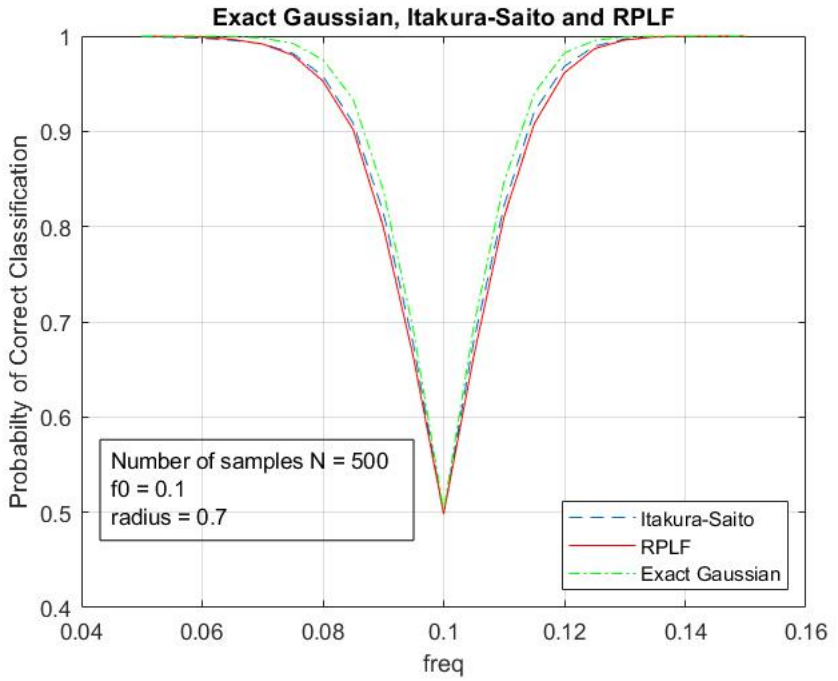


Figure 4.11: Simulation results, N = 500, $f_0 = 0.1$, radius = 0.7.

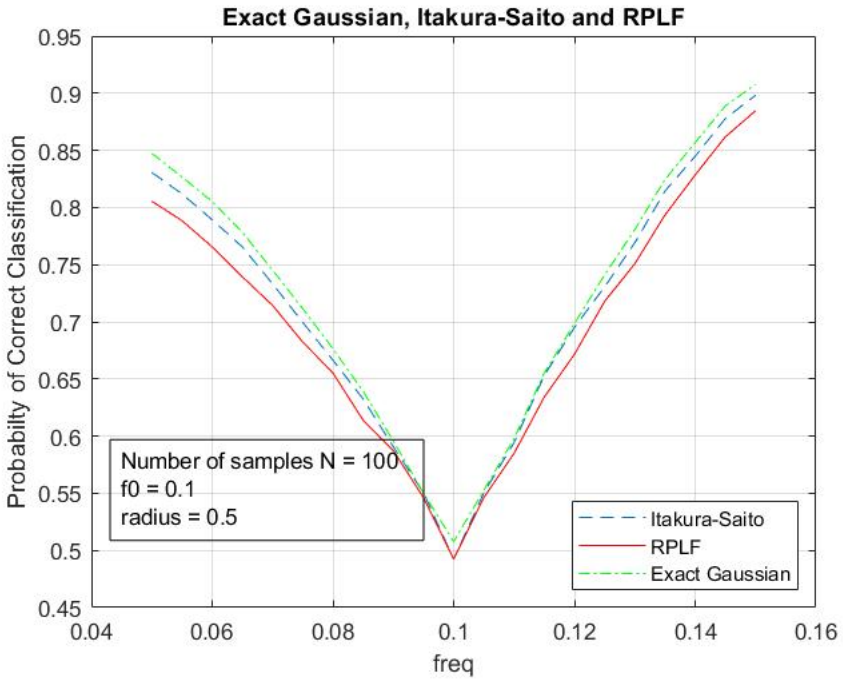


Figure 4.12: Simulation results, N = 100, $f_0 = 0.1$, radius = 0.5.

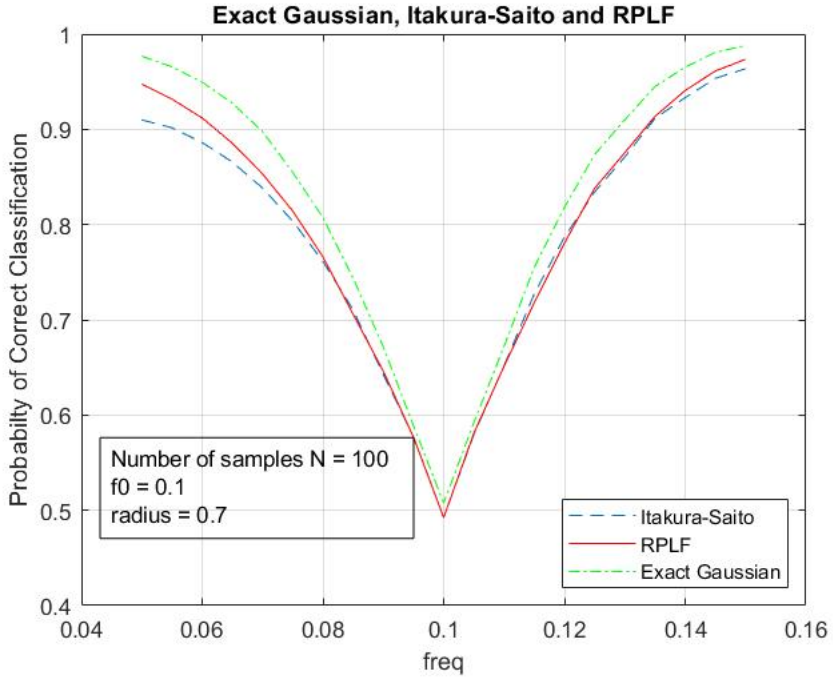


Figure 4.13: Simulation results, N = 100, $f_0 = 0.1$, radius = 0.7.

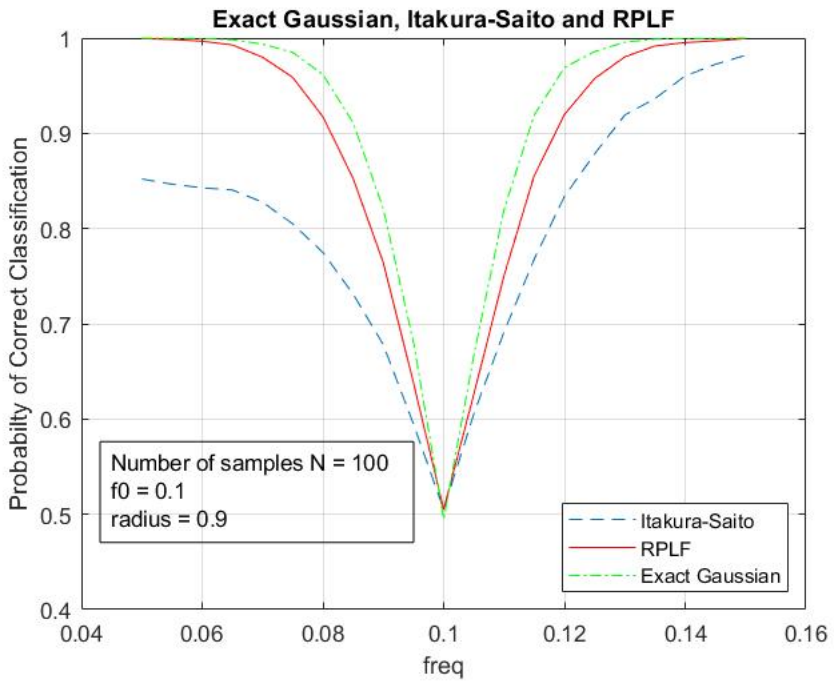


Figure 4.14: Simulation results, N = 100, $f_0 = 0.1$, radius = 0.9.

In order to get reliable results from the simulations utilizing an $AR(2)$ process, the poles were kept in a location where their position did not have an adverse effect on the result. In the initial simulations, placement had a significant effect on the Itakura-Saito but the RPLF was much less affected. This is the first indication of the RPLF being more robust.

Using the procedure in Figure 3.1, many different simulations were performed utilizing different frequencies, radii and number of samples. The Itakura-Saito outperformed the RPLF when the poles were placed away from the unit circle and away from the real axis, $f_0 = 0$ and $f_0 = .5$. Table 4.1 summarizes the results. In Table 4.1 the frequency column is the frequency of the the fixed

pole location filter associated with the second $AR(p)$ processes. Both of the processes had the same fixed radii, presented in the radius column.

Simulation	Frequency	Radius	Summary
1	0.1	0.9	The Itakura-Saito performed much worse then expected, especially at N= 50 and 100 and did not perform as well as the RPLF.
2	0.1	0.7	The Itakura-Saito unperformed compared the RPLF at N = 50 and 100.
3	0.1	0.5	The Itakura-Saito outperformed the RPLF, and got very close to the Exact Gaussian at N = 250 and 500.
4	0.2	0.9	The Itakura-Saito outperformed the RPLF in all cases except a very little at N=50. Both the Itakura-Saito and the RPLF came close to the exact Gaussian
5	0.3	0.9	As originally expected the Itakura-Saito outperformed the RPLF and came very close to the exact Gaussian.
6	0.3	0.5	All methods performed almost exactly the same.
7	0.4	0.9	Almost identical to simulation 1 except for a mirror image.

Table 4.1: Summary of simulations, for a $AR(2)$ processes with no added noise

The next simulations added noise to the $AR(p)$ process using the procedure in Figure 3.3. Table 4.2 and Figure 4.15 summarize the results.

RPLF vs Itakura-Saito with additive noise							
figure	f_0	f_1	radius	realizations	≈dB crossover	≈ DPCC low SNR	≈ DPCC high SNR
4.15a	0.15	0.20	0.45	20000	12	-.05	.02
4.15b	0.28	0.30	0.85	20000	12	-.15	.05
4.15c	0.25	0.30	0.75	20000	8	-.1	.025
4.15d	0.25	0.30	0.75	100000	8	-.1	.025

Table 4.2: Summary of results for simulations

In Table 4.2 the Difference in Probability of Correct Classification (DPCC) is the Itakura-Saito probability of correct classification minus the RPLF probability of correct classification, $p_{ccIS} - p_{ccRPLF}$. This was calculated for the beginning and end of each plot. The first column refers to the plots in Figure 4.15.

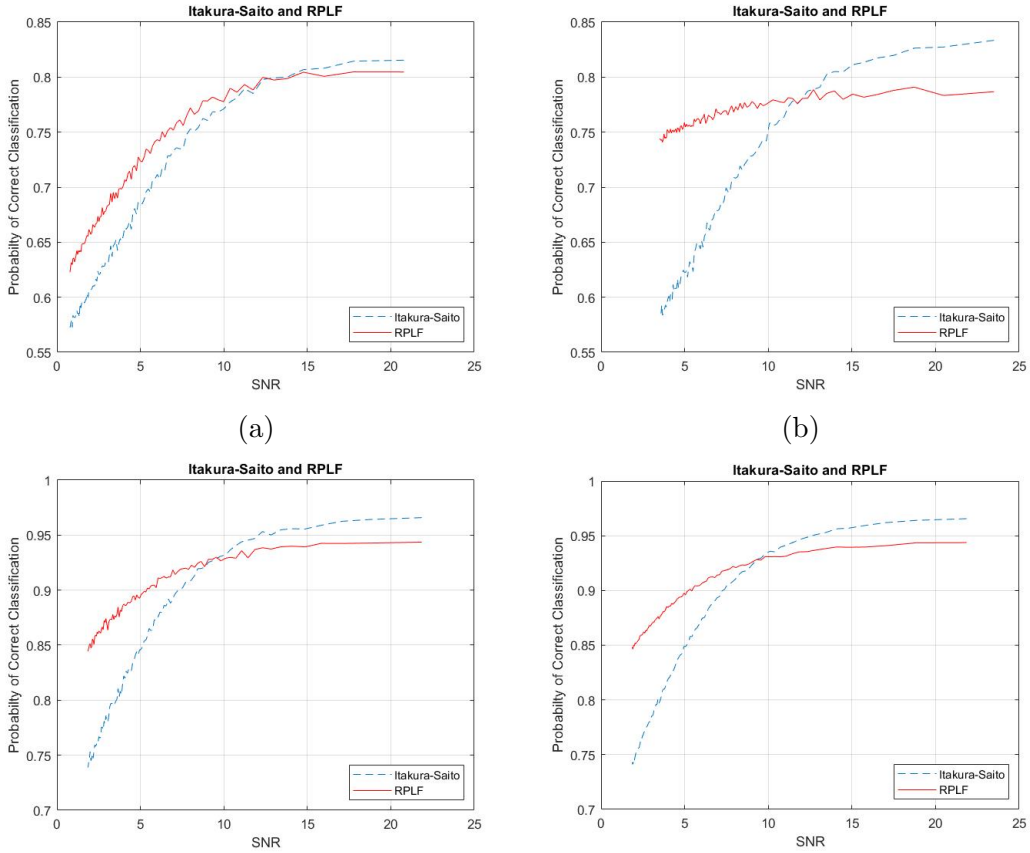
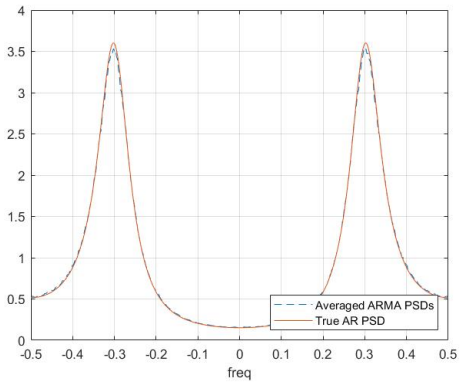


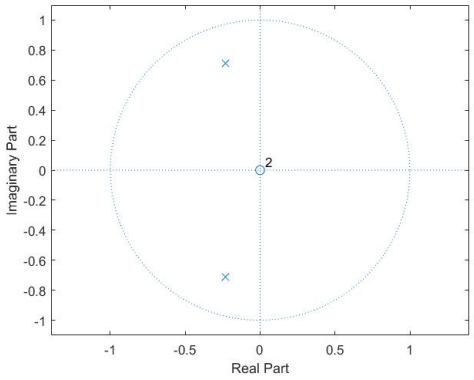
Figure 4.15: Probability of correct classification vs SNR

The Itakura-Saito performed slightly better in environments with a high SNR. The RPLF was more robust in low SNR environments.

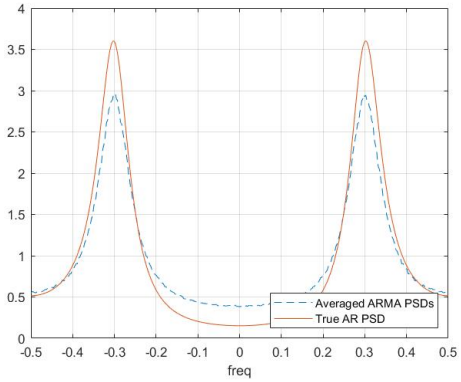
The last set of simulations followed the procedure in Figure 3.4, adding multi-path to the $AR(2)$ processes. Pole-zero plots and plots of the PSD for one of the $ARMA(2)$ processes was checked to validate the algorithms.



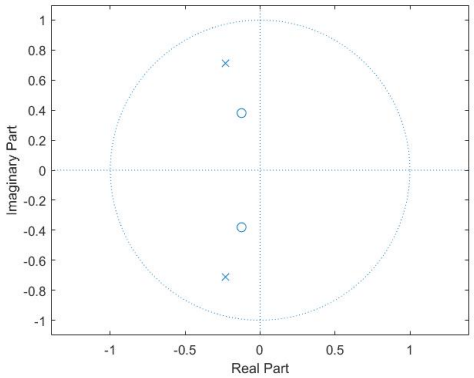
(a) PSD for zero radius = 0.0



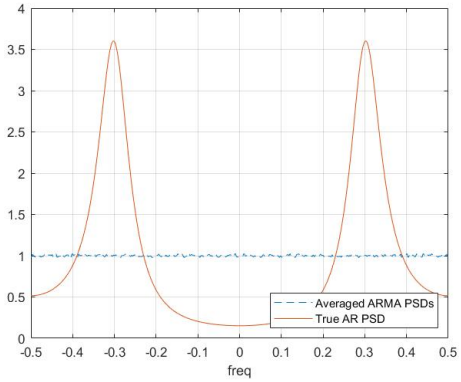
(b) Pole zero plot for zero radius = 0.0



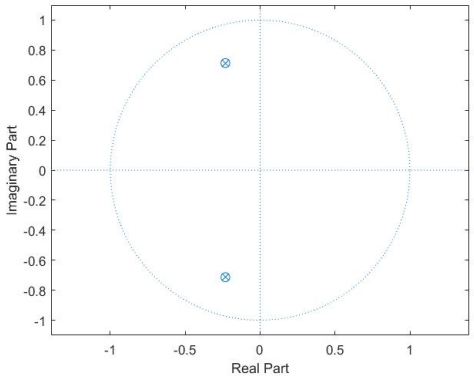
(c) PSD for zero radius = 0.4



(d) Pole zero plot for zero radius = 0.4



(e) PSD for zero radius = 0.75



(f) Pole zero plot for zero radius = 0.75

Figure 4.16: PSD and pole zero plots for $AR(2)$ 'True AR PSD' and and the $ARMA(2)$ 'Averaged $ARMA$ PSD's' processes, frequency = 0.3, 10000 realizations

Upon analyzing the periodograms in Figure 4.16, and the effects of the added

zeros, the algorithm appeared to be correct. Once the algorithm was verified, the simulation was performed, results are presented in Figure 4.17.

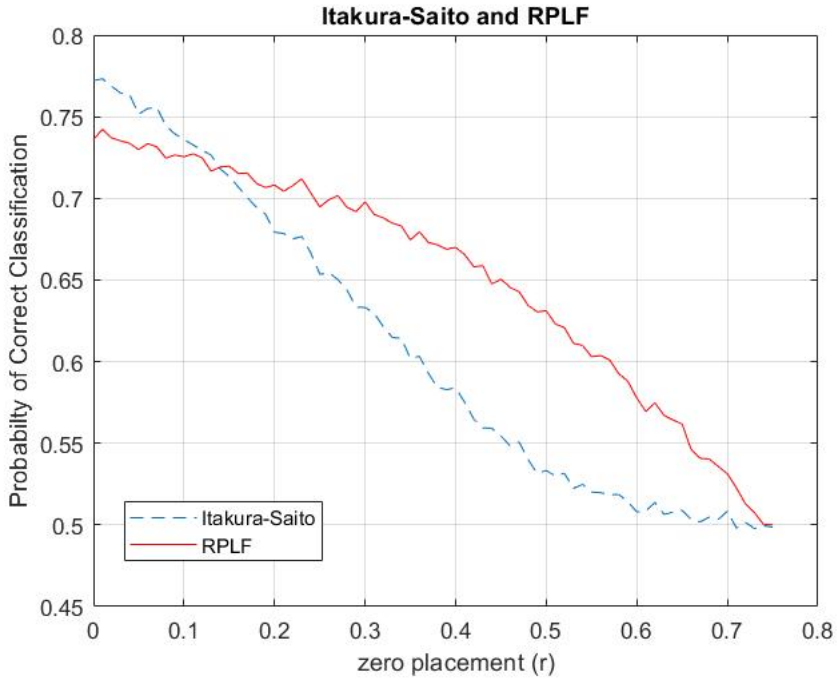


Figure 4.17: Probability of Correct Classification for an $AR(2)$ process with added multipath.

As expected, when the zeros are near the origin and have little effect on the periodogram the Itakura-Saito outperformed the RPLF. As the "multipath" worsened the RPLF performed better. The simulation gave the expected result of $p_{cc} = 0.5$ with a pole-zero cancellation.

4.3 Data Results

The results of the simulations point toward the RPLF having greater robustness to certain spectral deformations than the Itakura-Saito. The next step was to perform the analysis with the ISOLET database to see if real data yielded the

same results.

A baseline analysis was completed with the raw data, as described in chapter 3. Results for this baseline analysis are presented in Tables 4.3 and 4.4.

Classified RPLF, $p_{cc}(L) = .8$					
Classified as / True Letter	A	E	I	O	U
A	40	2	0	3	15
E	2	42	0	0	16
I	3	0	54	3	0
O	2	0	8	49	1
U	4	1	0	0	55

Table 4.3: Confusion matrix for the RPLF test statistic. $AR(p)$, $p = 12$

Classified Itakura-Saito, $p_{cc}(L) = .74$					
Classified as / True Letter	A	E	I	O	U
A	30	12	0	2	16
E	7	45	0	0	8
I	4	0	52	4	0
O	1	0	8	50	1
U	10	0	0	5	45

Table 4.4: Confusion matrix for the Itakura-Saito test statistic. $AR(p)$, $p = 12$

The probability of correct classification in Tables 4.3 and 4.4 is,

$$p_{cc}(L) = \frac{\sum number\ of\ results\ where\ i = j}{\sum number\ of\ results}$$

There are the five vowels being classified in this analysis, resulting in the template letters (L_i) and observed letters (L_j) where $0 \leq i, j \leq 5$.

Even in this baseline analysis the RPLF has a better probability of classification

than the Itakura-Saito. It should be noted, however that several factors could be responsible for this difference. Without a test of significance, it is not clear whether the observed difference between classifiers is due to individual sample variation versus a true difference in performance between the RPLF and the Itakura-Saito. The letter names for "I" and "U" are diphthongs and therefore are non-stationary. The exact location of the transition is unknown, but it is possible it is in sampled window. The RPLF outperformed the Ikaura-Saito classifying both of these letters. This may be an indication of the RPLF being a robust classifier to non-stationary data, further research needs to be done to test this hypothesis.

Other reasons for superior classification performance are, pole location, zero placement or noise. The SNR is high, 31.5dB, which means it is unlikely that noise is causing the difference. The letter "A" performed the worst of all the letters. There may be some zeros that are not accounted for in the model: a pole-zero plot of the AR parameters was completed to investigate this further and is presented in Figure 4.18.

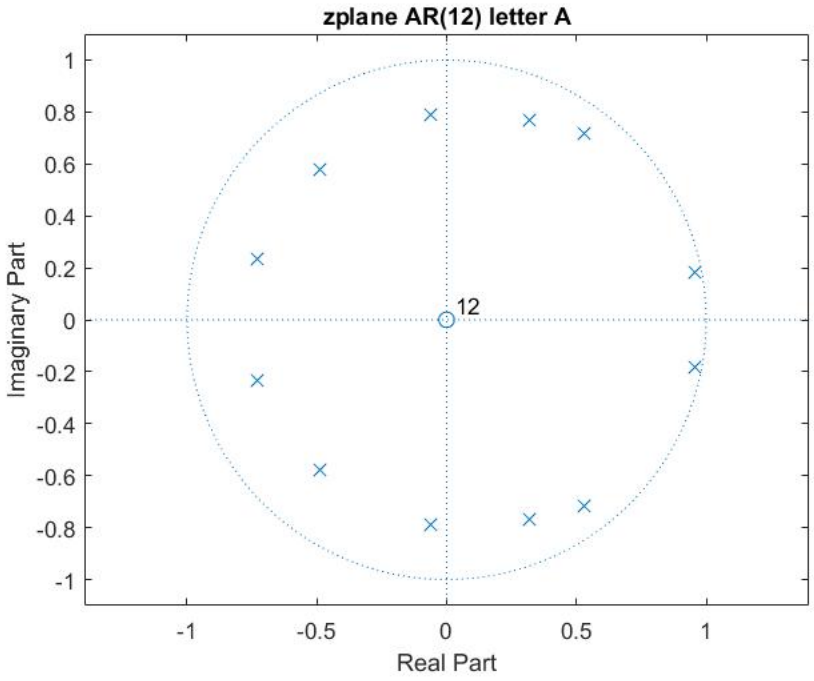


Figure 4.18: Pole Zero plot for the letter "A".

There are two poles very close to the unit circle at around $f \approx 0.025$; however, the next pole did not occur until $f \approx 0.15$. Was the model trying to account for a zero in between these two zeros? Upon examination of the other pole-zero plots, the letter "E" appeared to display the same attribute but did not suffer the same decrease in performance. The reason for the observed decrease in performance of the Itakura-Saito is unknown. Future work, beyond the scope of this thesis, would be required to fully investigate this.

The next analysis repeated the procedure but added independent white Gaussian noise. With the results presented in Tables 4.5 and 4.5.

Classified RPLF, $p_{cc}(L) = .77$					
Classified as / True Letter	A	E	I	O	U
A	41	5	0	1	13
E	3	44	0	0	13
I	3	0	54	3	0
O	4	0	11	44	1
U	8	2	0	0	50

Table 4.5: Confusion matrix for the RPLF test statistic with added noise. $AR(p)$, p=12, SNR =9dB

Classified Itakura-Saito, $p_{cc}(L) = .5$					
Classified as / True Letter	A	E	I	O	U
A	26	28	0	0	6
E	4	56	0	0	0
I	13	1	43	1	2
O	47	1	0	4	8
U	13	25	0	0	22

Table 4.6: Confusion matrix for the Itakura-Saito test statistic with added noise. $AR(p)$, p=12, SNR =9dB

The RPLF does a much better job of classifying when the signal is embedded in white Gaussian noise. The difference between the probability of correct classification for no noise vs noise is represented as, $p_{cc}(L) - p_{cc-n}(n)$. In our data, the difference between the RPLF with noise and without noise was 0.03 $(0.80 - 0.77)$. For the Itakura-Saito it was 0.24 $(0.74 - 0.50)$. Performance with added noise decreased only slightly for the RPLF, approximately 3%, but diminished by approximately 30% for the Itakura-Saito. This analysis was for only a single SNR, a later analysis would include a range of SNR.

The next analysis simulated multipath by adding a set of zeros at a fixed frequency and radius. With the results presented in Tables 4.7 and 4.7.

Classified RPLF, $p_{cc}(L) = .76$					
Classified as / True Letter	A	E	I	O	U
A	5	2	0	4	19
E	3	35	0	0	22
I	3	0	53	4	0
O	3	0	5	51	1
U	4	1	0	0	55

Table 4.7: Confusion matrix for the RPLF test statistic. $AR(p)$, p=12, with zero.

Classified Itakura-Saito, $p_{cc}(L) = .57$					
Classified as / True Letter	A	E	I	O	U
A	8	30	0	1	21
E	0	56	0	0	4
I	2	1	33	11	13
O	1	3	0	31	25
U	1	13	0	3	43

Table 4.8: Confusion matrix for the AG test statistic. $AR(p)$, p=12, with zero.

The Itakura-Saito performance decreased by approximately 23% in this analysis, slightly better than in the last analysis. However it was vastly outperformed by the RPLF, for which diminished performance was nearly the same, decreasing only about 5%. The difference $(p_{cc}(L) - p_{cc-z}(n))$ for the RPLF was 0.04 $(0.8 - 0.76)$ and for the Itakura-Saito was, 0.17 $(0.74 - 0.57)$

Figure 4.19 shows the results of the next analysis, the cepstrum classifier was added and included a range of SNR levels.

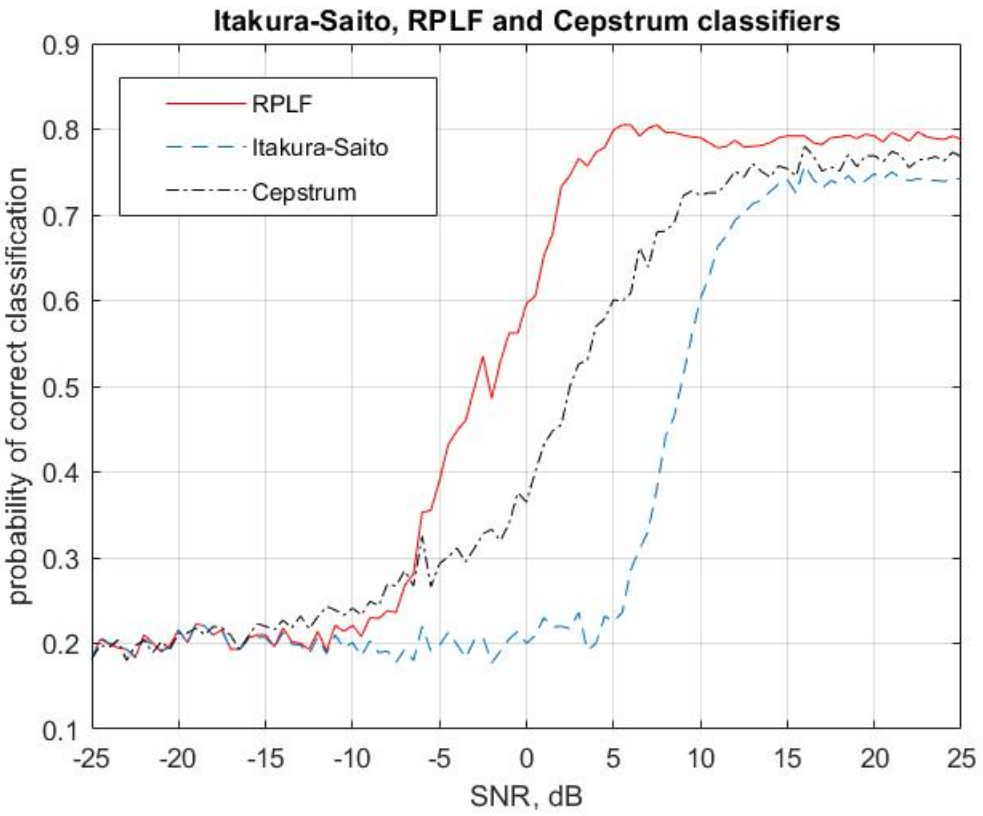


Figure 4.19: Probability of correct classification for RPLF, Itakura-Saito and Cepstrum classifiers, with a range of added noise.

It was expected that the cepstrum classifier would have a better probability of correct classification then the Itakura-Saito with speech data. With other types of data the Itakura-Saito may perform better. The performance of the RPLF exceeded both the Itakura-Saito and the cepstrum. The simulation looks to have run correctly, since the lowest probability of correct classification is 0.2. The 0.2 is the lowest probability of correct classification because there are five different letters and they each have an equal probability of being selected,

therefore 0.2.

The next couple of analyses added zeros, and varied either the radius or the frequency. The results of the simulation with the change in radius are presented first in Figure 4.20.

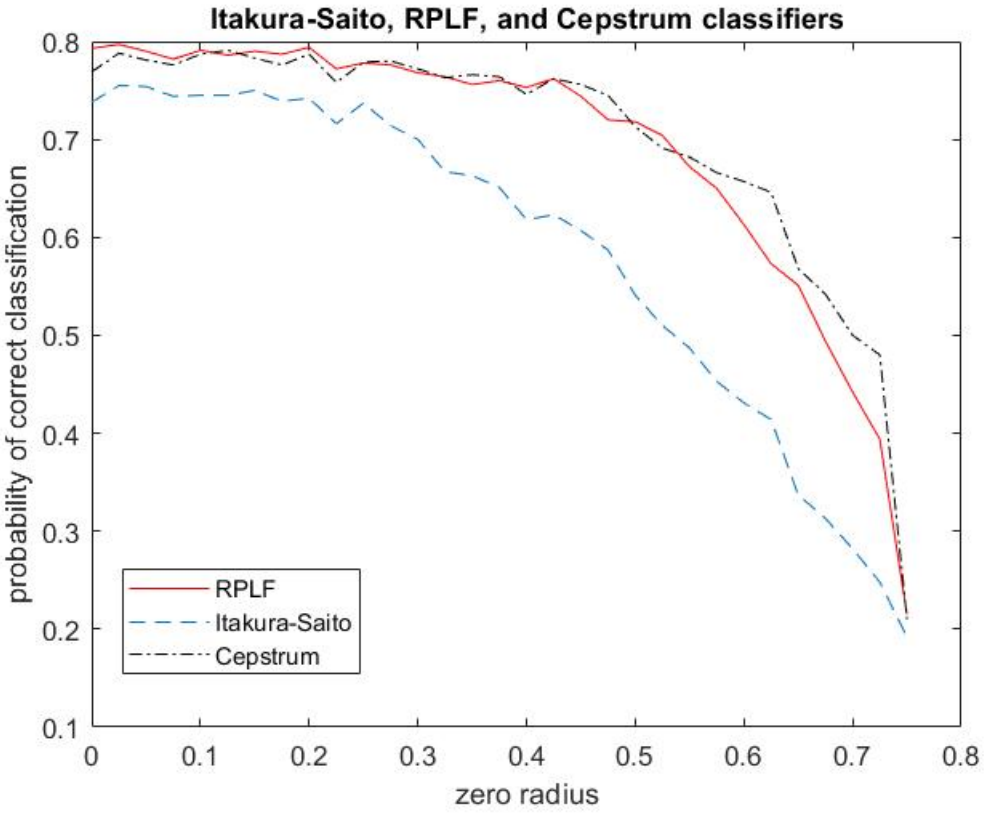


Figure 4.20: Probability of correct classification for RPLF, Itakura-Saito and Cepstrum classifiers, versus the added zero radius.

Figure 4.21 presents the results for the analysis were the frequency of the zeros was changed.

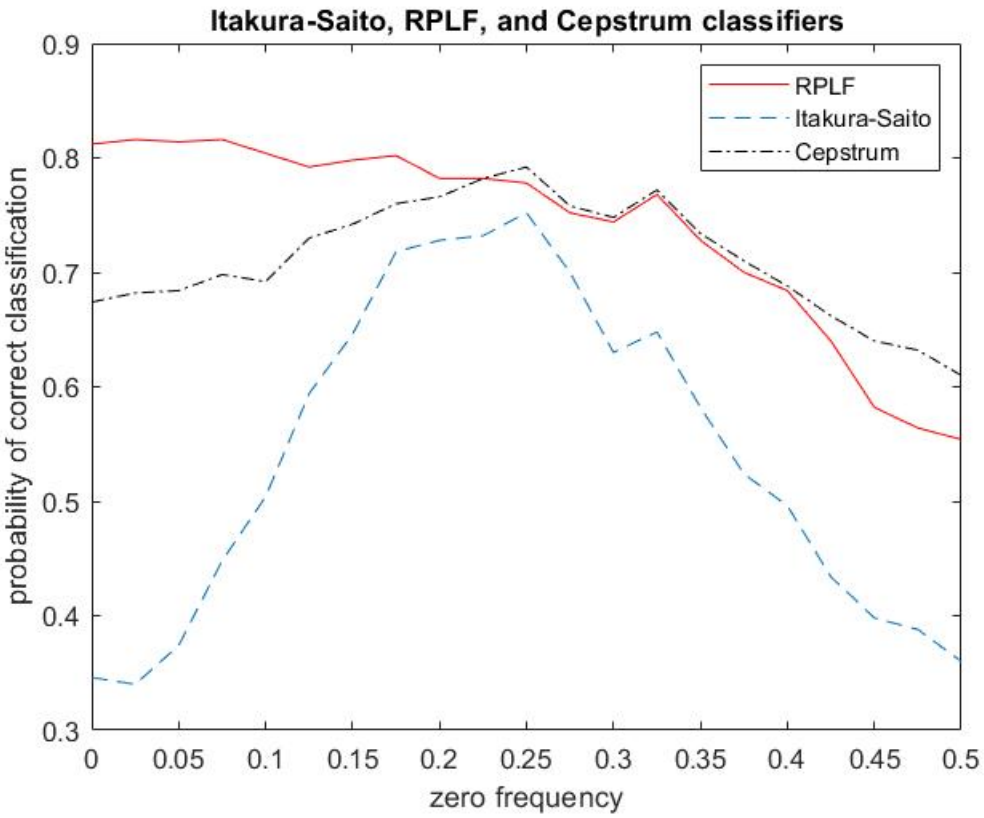


Figure 4.21: Probability of correct classification for RPLF, Itakura-Saito and Cepstrum classifiers versus the added zero frequency.

In Figure 4.20 the RPLF and the cepstrum classifiers performed about equally, however, they both outperform the Itakura-Saito. In Figure 4.21 the RPLF performed as well or better then the cepstrum, except at frequencies around 0.25 and 0.5. Again, they both outperform the Itakura-Saito.

4.4 Template modifications

In order to increase the probability of correct classification additional templates were examined. The conditional PDF of $x[n]$ for large data records is shown in

[14], for an $AR(p)$ process

$$x[n] = u[n] - \sum_{k=1}^{p} x[k]u[n-k]$$

is,

$$p(\mathbf{x}) = \frac{1}{(2\pi\sigma_u^2)^{(N-p)/2}} e^{-\frac{1}{2\sigma^2}\sum_{n=p}^{N-1}\left(x[n]+\sum_{j=1}^{p} a[j]x[n-j]\right)^2}$$

It was also shown in [14] the maximum likelihood estimate MLE for the $a[k]$ coefficients are found using the covariance method. If it is assumed that we have M independent $AR(p)$ process data, each with the same number of $AR(p)$ parameters, the likelihood function is the product of the individual likelihood functions.

$$p(\mathbf{x}) = \frac{1}{(2\pi\sigma_u^2)^{M(N-p)/2}} e^{-\frac{1}{2\sigma^2}\sum_{m=0}^{M-1}\sum_{n=p}^{N-1}\left(x[m,n]+\sum_{j=1}^{p} a[m,j]x[m,n-j]\right)^2}$$

where n is the number of samples for each utterance, m is the number of utterances, and p is the number of $a[k]$ coefficients. The MLE for the $a[k]$ coefficients of many $AR(p)$ processes is found using the covariance method, which averages the M covariance matrices and covariance vectors. Then from the averaged covariance matrices and covariance vectors the $a[k]$ coefficients can be found. The result of this analysis utilizing these templates is presented is in Figure 4.22.

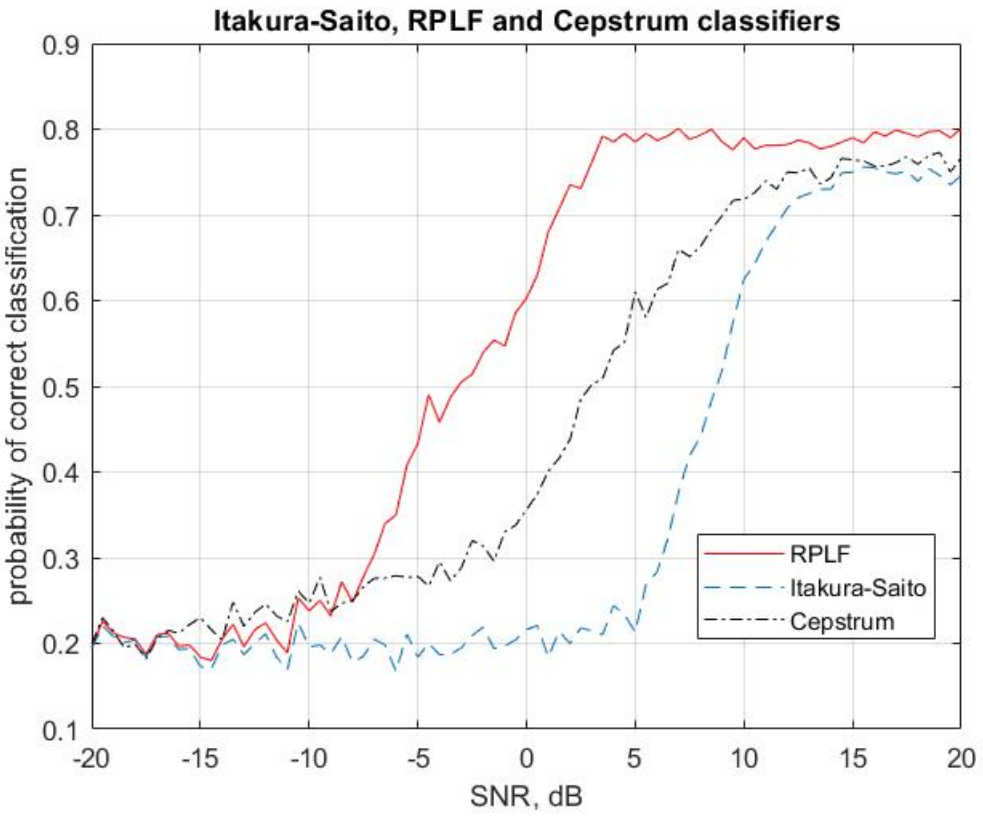


Figure 4.22: Probability of correct classification for RPLF, Itakura-Saito and Cepstrum classifiers with added noise, solving $a[k]$ using covariance method, M = 240.

There was a minimal increase in probability of correct classification when using the covriance method, Figures 4.19 and 4.22.

Next the template data was divided into logical subgroups. Since the gender data is available, this is a natural way to subdivide the groups. Sub-grouping this way resulted in ten calculations for each classifier. The classifier took the maximum value from the ten calculations. For example for the RPLF, when

$P_{Tmale}(f)$ is the normalized PSD male template for the letters.

$$l_{R(j)} = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_{Tmale}(f)}\right) df$$

and when $P_{Tfemale}(f)$ is the normalized power spectral density female template for the letters and $1 \leq j \leq 10$.

$$l_{R(j)} = -\int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{I}(f) \ln\left(\frac{\bar{I}(f)}{P_{Tfemale}(f)}\right) df$$

and $\bar{I}(f)$ is the normalized periodogram of the observed data. The goal is that, for the $AR(p)$ coefficients, the means will converge to the true value for each gender and the variances will decrease. Tables 4.9 and 4.10 present the means and variances for the original estimates, and the tables with the gender specified $AR(p)$ coefficients follow in Tables 4.11 - 4.14.

Estimated mean for letter I, $p = 12$					
$AR(p)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
-1.3615	0.5923	0.0320	0.0105	0.00086	0.0508
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0134	0.0427	-0.0087	0.0378	-0.0756	0.1191

Table 4.9: Estimated means of $AR(12)$ coefficients for the letter I

Estimated variance for letter I, $p = 12$					
$AR(p)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
0.0763	0.2658	0.1417	0.0975	0.0541	0.0610
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0400	0.0282	0.0226	0.0257	0.0308	0.0124

Table 4.10: Estimated variances of $AR(12)$ coefficients for the letter I

Estimated mean for letter I, female voice, $p = 12$					
$AR(12)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
-1.3100	0.5750	0.0072	0.0613	0.0208	0.00017
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0054	0.0820	-0.0223	0.0303	-0.0208	0.0962

Table 4.11: estimated means of $AR(12)$ coefficients for the letter I, spoken by a female

Estimated variance for letter I, female voice, $p = 12$					
$AR(p)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
0.0631	0.2282	0.1149	0.0530	0.0356	0.0391
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0284	0.0206	0.0176	0.0207	0.0218	0.0133

Table 4.12: Estimated variances of $AR(12)$ coefficients for the letter I, spoken by a female

Estimated mean for letter I, male voice, $p = 12$					
$AR(p)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
-1.4086	0.5972	0.0794	-0.0465	-0.0211	0.0934
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0296	0.0030	0.0059	0.0420	-0.1262	0.1407

Table 4.13: Estimated means of $AR(12)$ coefficients for the letter I, spoken by a male

Estimated variance for letter I, male voice, $p = 12$					
$AR(p)-1$	$AR(p)-2$	$AR(p)-3$	$AR(p)-4$	$AR(p)-5$	$AR(p)-6$
0.0710	0.2435	0.1065	0.0966	0.0471	0.0606
$AR(p)-7$	$AR(p)-8$	$AR(p)-9$	$AR(p)-10$	$AR(p)-11$	$AR(p)-12$
0.0345	0.0253	0.0187	0.0220	0.0257	0.0078

Table 4.14: Estimated Variances of $AR(12)$ coefficients for the letter I, spoken by a male

Figure 4.23 presents the results with Table 4.15 summarizing the results

for all classifiers. The probability of correct classification in Table 4.15 is the probability of correct classification with an SNR of $20dB$.

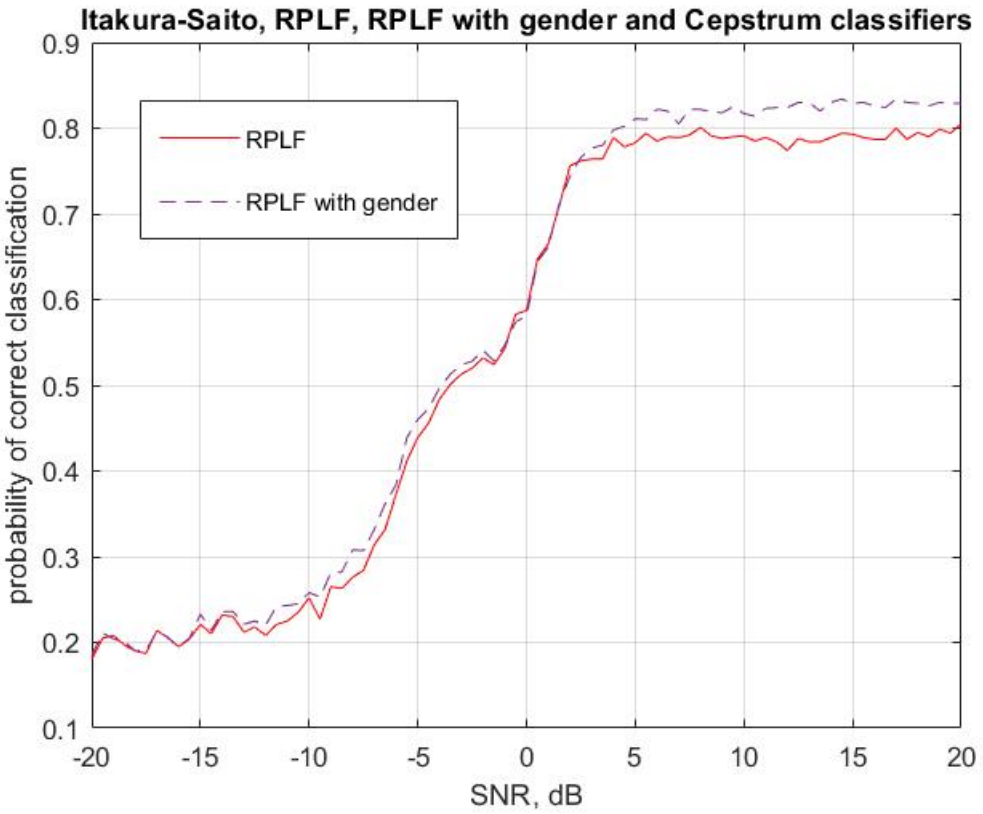


Figure 4.23: Probability of correct classification for RPLF, with and without sub-grouped templates.

Probability of correct classification	
classifier	pcc
Itakura-Saito	0.7460
Cepstrum	0.7690
RPLF	0.8050
RPLF w/gender	0.8290

Table 4.15: Probability of correct classification for RPLF, RPLF with gender sub-group, Itakura-Saito and Cepstrum classifiers at a SNR of 20dB

As expected, the division of the template into sub-groups resulted in an

increase in the probability of correct classification. The RPLF with gender outperformed the RPLF by approximately 3%.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Table (5.1) summarizes the results of all simulations and analyses with a ranking of the performance of each classifier, determined by the probability of correct classification. The classifier with the higher probability of correct classification got a lower number. If no classifier had a higher probability of correct classification for the entire range, as in the analysis with a zero, then the one that appeared to be the most robust over the majority of range was selected as superior. The cepstrum classifier was only used in the data analysis with noise and with channel distortion, so it is only ranked for those two classifications.

Summary of Results with Performance Ranking			
	RPLF	Itakura-Saito	Cepstrum
Simulations	2	1	N/A
Simulations w/noise	1	2	N/A
Simulations w/zero	1	2	N/A
Data analyses	1	2	N/A
Data analyses w/noise	1	3	2
Data analysis w/zero	1	3	2

Table 5.1: Table summary of results, with ranking of results.

The RPLF outperformed the Itakura-Saito and was a more robust classifier overall. The only time the Itakura-Saito performed better was with the simulated perfect $AR(p)$ model. In every other simulation and data analysis test, the RPLF outperformed the Itakura-Saito.

5.2 Future work

The RPLF exhibited superior performance, but this analysis was limited because it was only tested with simulations and with speech data. Further analyses of both the RPLF and the Itakura-Saito classifiers are needed to determine if the RPLF is truly more robust across a variety of data types, including but not limited to medical, economic or environmental data. In order to show the RPLFs full potential the selected data should be both stationary and non-stationary.

The results gathered through out this analysis of speech data is encouraging to the field of development and refinement of signal processing classifiers. As this analysis demonstrated, the more commonly use Itakura-Saito classifier was outperformed by the RPLF classifier on nearly all simulations and all data tests. If it is found that the RPLF classifier is a superior classifier across all data types, this could have far reaching implications for all applications of signal processing classification technology.

REFERENCES

[1] Ronald Cole, Y Muthusamy, and Mark Fanty. *CSLU: ISOLET Spoken Letter Database Version 1.3 LDC2008S07*. https://catalog.ldc.upenn.edu/LDC2008S07. Online, accessed Oct-2017. 2017.

[2] Ronald Cole, Y Muthusamy, and Mark Fanty. *The Isolet Spoken Letter Database*. Tech. rep. University of Oregon, 1994.

[3] K Fokianos. "Spectral estimation". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2010), pp. 165–170.

[4] R.G. Gallager. *Discrete Stochastic Processes*. Boston, Massachusetts: Kluwer, 1996.

[5] B. W. Gillespie and L. E. Atlas. "Acoustic diversity for improved speech recognition in reverberant environments". In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 2002, pp. I–557–I–560. DOI: 10.1109/ICASSP.2002.5743778.

[6] A. Gray and J. Markel. "Distance measures for speech processing". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.5 (1976), pp. 380–391. ISSN: 0096-3518. DOI: 10.1109/TASSP.1976.1162849.

[7] Ricardo Gutierrez-Osuna. *Introduction to Speech Processing*. Tech. rep. Texas A&M University, 2016.

[8] Mark Hall, A.V. Oppenheim, and Alan S. Willsky. "Time-varying parametric modeling of speech". In: *Signal Processing* 5 (Dec. 1977), pp. 267–285.

[9] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing*. Upper Saddle River, New Jersey: Prentice Hall, 2001.

[10] B Juang and Lawrence R. Rabiner. *Automatic Speech Recognition - A Brief History of the Technology Development*. Tech. rep. Georgia Institute of Technology, Atlanta & Rutgers University and the University of California, Santa Barbara, 2004.

[11] Jean-Claude Junqua and Jean-Paul Haton. *Robustness in Automatic Speech Recognition*. Boston, Massachusetts: Kluwer Academic Publishers, 1996.

[12] Steven Kay. *Fundamentals of statistical signal processing, volume I, Estimation Theroy*. Upper Saddle River, New Jersey: Prentice-Hall, 1993.

[13] Steven Kay. *Intuitive Probability and Random Processes using MATLAB®*. New York, New York: Springer, 2006.

[14] Steven Kay. *Modern spectral estimation*. Upper Saddle River, New Jersey: Prentice-Hall, 1988.

[15] Steven Kay. *Poisson Maximum Likelihood Spectral Inference*. Tech. rep. University of Rhode Island, 2016.

[16] K. Kinoshita et al. "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech". In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 2013, pp. 1–4. DOI: 10.1109/WASPAA.2013.6701894.

[17] Esther Klabbers and Raymond N. J. Veldhuis. *On the reduction of concatenation artefacts in diphone synthesis*. Tech. rep. Jan. 1998.

[18] Solomon Kullback. *Information Theory and Statistics*. Mineola, New York: Dover Publications, Inc, 1997.

[19] John D. Markel and Jr Augustine H. Gray. *Linear Prediction of Speech*. 3rd ed. Berlin, Germany: Springer-Verlag, 1982.

[20] Merriam-Webster. *Definition of robust*. https://www.merriam-webster.com/dictionary/robust. Online, accessed Nov-2017. 2017.

[21] Alan Oppenheim and Ronald Schafer. *Discrete-Time Signal Processing*. 3rd ed. Boston, Massachusetts: Pearson, 2009.

[22] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[23] Donald L. Snyder. *Random Point Processes*. Wiley-Interscience Publication, 1975.

[24] O. Viikki, D. Bye, and K. Laurila. "A recursive feature vector normalization approach for robust speech recognition in noise". In: *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. Vol. 2. 1998, 733–736 vol.2. DOI: 10.1109/ICASSP.1998.675369.

[25] Takuya Yoshioka et al. "Making Machines Understand Us in Reverberant Rooms". In: *IEEE Signal Processing Magazine* (Nov 2012), pp. 114–126.

APPENDIX A

DISTANCE BETWEEN EXACT GAUSSIAN AND THE ASYMPTOTIC GAUSSIAN LIKELIHOOD FUNCTION

A.1 Difference

The exact Gaussian takes the form of,

$$\ln(p_E(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(det(\boldsymbol{C})) - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{C}^{-1}\boldsymbol{x}$$

Because the random variable is a zero mean random variable, it takes the form of,

$$= -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln(det(\boldsymbol{R})) - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{R}^{-1}\boldsymbol{x} \tag{A.1}$$

The derivation for the inverse autocorrelation matrix and its determinate can be found in [14].

$$det(\boldsymbol{R}) = \frac{(\sigma^2)^N}{(1-a^2[1])} \tag{A.2}$$

$$\boldsymbol{R}^{-1} = \frac{1}{\sigma^2}\begin{bmatrix} 1 & a[1] & 0 & 0 & \cdots & 0 \\ a[1] & 1+a^2[1] & a[1] & 0 & \cdots & 0 \\ 0 & a[1] & 1+a^2[1] & a[1] & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a[1] & 1+a^2[1] & a[1] \\ 0 & 0 & \cdots & 0 & a[1] & 1 \end{bmatrix}$$

Looking at the last term in the exact gaussian (A.1), $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{R}^{-1}\boldsymbol{x}$, where $\boldsymbol{x}$ is an N-dimensional vector of zero mean WSS Gaussian random variables, from

$1 \leq n \leq N$ and $\boldsymbol{R}^{-1}$ is an N x N matrix.

$$= \frac{1}{2\sigma^2} \begin{bmatrix} x[0] & x[1] & \cdots & x[N-1] \end{bmatrix}$$
$$\begin{bmatrix} 1 & a[1] & 0 & 0 & \cdots & 0 \\ a[1] & 1+a^2[1] & a[1] & 0 & \cdots & 0 \\ 0 & a[1] & 1+a^2[1] & a[1] & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a[1] & 1+a^2[1] & a[1] \\ 0 & 0 & \cdots & 0 & a[1] & 1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix}$$

$$= \frac{1}{2\sigma^2} \begin{bmatrix} x[0] & x[1] & \cdots & x[N-1] \end{bmatrix}$$
$$\begin{bmatrix} x[0]+a[1]x[1] \\ a[1]x[0]+x[1]+a^2[1]x[1]+a[1]x[2] \\ a[1]x[1]+x[2]+a^2[1]x[2]+a[1]x[3] \\ \vdots \\ a[1]x[N-3]+x[N-2]+a^2[1]x[N-2]+a[1]x[N-1] \\ a[1]x[N-2]+x[N-1] \end{bmatrix}$$

$$
\begin{aligned}
= \frac{1}{2\sigma^2} \Big[& \; x[0](x[0] + a[1]x[1]) \\
& + \; x[1](a[1]x[0] + x[1] + a^2[1]x[1] + a[1]x[2]) \\
& + \; x[2](a[1]x[1] + x[2] + a^2[1]x[2] + a[1]x[3]) \\
& + \; \cdots \\
& + \; x[N-2](a[1]x[N-3] + x[N-2] + a^2[1]x[N-2] + a[1]x[N-1]) \\
& + \; x[N-1](a[1]x[N-2] + x[N-1] \Big]
\end{aligned}
$$

$$
\begin{aligned}
= \frac{1}{2\sigma^2} \Big[& \; x^2[0] + a[1]x[0]x[1] \\
& + \; a[1]x[0]x[1] + x^2[1] + a^2[1]x^2[1] + a[1]x[1]x[2] \\
& + \; a[1]x[1]x[2] + x^2[2] + a^2[1]x^2[2] + a[1]x[2]x[3] \\
& + \; \cdots \\
& + \; a[1]x[N-3]x[N-2] + x^2[N-2] \\
& + a^2[1]x^2[N-2] + a[1]x[N-2]x[N-1] \\
& + \; a[1]x[N-2]x[N] + x^2[N-1] \Big]
\end{aligned}
$$

If we regroup the above equations we can form the following,

$$
\begin{aligned}
= \frac{1}{2\sigma^2}\Big[& x^2[0] + a[1]x[0]x[1] + a[1]x[0]x[1] + x^2[1] \\
& + \ a^2[1]x^2[1] + a[1]x[1]x[2] + a[1]x[1]x[2] + x^2[2] \\
& + \ a^2[1]x^2[2] + a[1]x[2]x[3] + a[1]x[2]x[3] + x^2[3] \\
& + \ \cdots \\
& + \ a^2[1]x^2[N-3] + a[1]x[N-3]x[N-2] \\
& + a[1]x[N-3]x[N-2] + x^2[N-2] \\
& + \ a^2[1]x^2[N-2] + a[1]x[N-2]x[N-1] \\
& + a[1]x[N-2]x[N-1] + x^2[N-1] \ \Big]
\end{aligned}
$$

$$
\begin{aligned}
= \frac{1}{2\sigma^2}\Big[& x^2[0] + 2a[1]x[0]x[1] + x^2[1] \\
& + \ \sum_{n=2}^{N-1} (x[n] + a[1]x[n-1])^2 \Big]
\end{aligned}
\tag{A.3}
$$

If we set the initial condition of $x[-1] = 0$, this can be rewritten,

$$
= \frac{1}{2\sigma^2}\left[\sum_{n=0}^{N-1} \Big[(x[n] + a[1]x[n-1])^2 \Big] - |a^2[1]x^2[0]| \right] \tag{A.4}
$$

Now substituting (A.2) and (A.3) into (A.1) will give the form of the exact log Gaussian pdf relative to the pole placement (a) and the length of the process

(N),

$$\begin{aligned}\ln(p_E(\boldsymbol{x})) = &-\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\left(\frac{\sigma^2}{(1-a^2[1])}\right) \\ &-\frac{1}{2\sigma^2}\left[\sum_{n=0}^{N-1}\left[(x[n]+a[1]x[n-1])^2\right] - a^2[1]x^2[0]\right]\end{aligned} \tag{A.5}$$

The asymptotic equivalent of the log of the exact Gaussian is derived in appendix 3D of [12]. We use that process in reverse as a guide through the derivation here. The asymptotic Gaussian takes the form of,

$$\ln(p_A(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}}\left(\ln(P(f)) + \frac{I(f)}{P(f)}\right)df$$

set

$$J = \int_{-\frac{1}{2}}^{\frac{1}{2}}\left(\ln(P(f)) + \frac{I(f)}{P(f)}\right)df$$

Starting with the first term,

$$\int_{-\frac{1}{2}}^{\frac{1}{2}}\left(\ln(P(f))\right)df \tag{A.6}$$

and,

$$P(f) = \frac{\sigma_u^2}{|A(f)|^2}$$

substituting back into (A.6)

$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(\frac{\sigma_u^2}{|A(f)|^2}) \right) df$$
$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(\sigma_u^2) \right) df - \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(|A(f)|^2) \right) df$$

since the integral goes from $-\frac{1}{2}$ to $\frac{1}{2}$ and σ_u^2 does not depend on f,

$$= \ln \sigma_u^2 - \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(|A(f)|^2) \right) df$$

looking at the second part.

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln(|A(f)|^2) \right) df$$
$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln A(f) + \ln A^*(f) \right) df$$
$$= 2\mathrm{Re} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\ln A(f) \right) df$$

Since the linear filter is an $AR(1)$ filter with $a < 1$ it is a stable causal filter it will have its poles inside the unit circle with a region of convergence going outward. Therefore the unit circle will be in the region of convergence. Then going from an integral in frequency to an integral in the z-plane results in a

contour integral around the unit circle.

$$= 2\text{Re} \oint_C lnA(z) \frac{dz}{2\pi jz}$$

the definition of the inverse z-transform is

$$g[n] = \frac{1}{2\pi j} \oint_C G(z) z^{n-1} dz$$

So with $n = 0$

$$= 2\text{Re} \left[\mathcal{Z}^{-1} \{ \ln A(z)|_{n=0} \} \right]$$

Since we now the sequence is causal and stable we will look at the limit of the sequence as it goes to ∞

$$\lim_{x \to \infty} X(z) = \lim_{x \to \infty} \sum_{n=0}^{\infty} x[n] z^{-n} = \lim_{x \to \infty} \sum_{n=0}^{\infty} x[n] \frac{1}{z^n}$$

so as $z \to \infty$ the only term left is when $n = 0$, this is the initial value theorem. Then,

$$\begin{aligned}
&\mathcal{Z}^{-1} \{ \ln H(z) \}|_{n=0} \\
&= \lim_{x \to \infty} \ln H(z) \\
&= \ln \lim_{x \to \infty} H(z) \\
&= \ln(h[0]) \\
&= \ln(1) = 0
\end{aligned}$$

resulting in,

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} (\ln(P(f)))\, df = \ln(\sigma_u^2)$$

and

$$J = \ln(\sigma_u^2) + \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{I(f)}{P(f)}\right) df$$

where

$$P(f) = \frac{\sigma_u^2}{|A(f)|^2}$$

Therefore,

$$= \ln \sigma_u^2 + \frac{1}{N\sigma_u^2} \int_{-\frac{1}{2}}^{\frac{1}{2}} |A(f)|^2 |X(f)|^2 df$$

$$= \ln \sigma_u^2 + \frac{1}{N\sigma_u^2} \int_{-\frac{1}{2}}^{\frac{1}{2}} |A(f)X(f)|^2 df$$

setting,

$$Y(f) = A(f)X(f)$$

$$J = \ln \sigma_u^2 + \frac{1}{N\sigma_u^2} \int_{-\frac{1}{2}}^{\frac{1}{2}} |Y(f)|^2 df$$

Using parsevals theorem,

$$\ln \sigma_u^2 + \frac{1}{N\sigma_u^2} \sum_{n=-\infty}^{\infty} y^2[n] \tag{A.7}$$

Because multiplication in the frequency domain is convolution in the time domain.

$$y[n] = a[n] * x[n]$$

Remember, $a[n]$ is a single zero FIR filter with a tap weight of $a[1]$. For $n < 0$ and $n > (N-1), x[n] = 0$

$$\begin{aligned}
y[0] &= x[0] \\
y[1] &= x[1] + a[1]x[0] \\
y[2] &= x[2] + a[1]x[1] \\
&\vdots \\
y[N-1] &= x[N-1] + a[1]xN - 2 \\
y[N] &= a[1]x[N-1]
\end{aligned}$$

Then,

$$\begin{aligned}
y^2[0] &= x^2[0] \\
y^2[1] &= (x[1] + a[1]x[0])^2 \\
y^2[2] &= (x[2] + a[1]x[1])^2 \\
&\vdots \\
y^2[N-1] &= (x[N-1] + a[1]x[N-2])^2 \\
y^2[N] &= a^2[1]x^2[N-1]
\end{aligned}$$

now recalling the summation in (A.7), and since $y^2[n]$ is zero for $n \leq 0$ and $n \geq N$

$$\sum_{n=0}^{N} y^2[n] = x^2[0] + \sum_{n=1}^{N-1}(x[n] + a[1]x[n-1])^2$$
$$a^2[1]x^2[N-1]$$

From the derivation of the exact form(A.3)

$$\frac{1}{2}\boldsymbol{x}^T\boldsymbol{R}^{-1}\boldsymbol{x} = \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big]$$

following this same format for $y[n]$, by pulling out the $n = 1$ term

$$\sum_{n=0}^{N} y^2[n] = x^2[0] + x^2[1] + 2a[1]x[0]x[1] + a^2[1]x^2[0] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 + a^2[1]x^2[N-1]$$

Now substituting everything back in.

$$\ln(p(\boldsymbol{x})) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\Big(\ln\sigma_u^2 + \frac{1}{N\sigma_u^2}\big(+ x^2[0] + x^2[1] + 2a[1]x[0]x[1] + a^2[1]x^2[0] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 + a^2[1]x^2[N-1]\big)\Big)$$

$$
\begin{aligned}
&= -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma_u^2 \\
&\quad - \frac{1}{2\sigma_u^2}\Big[x^2[0] + x^2[1] + 2a[1]x[0]x[1] + a^2[1]x^2[0] \\
&\quad + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 \\
&\quad + a^2[1]x^2[N-1]\Big]
\end{aligned}
$$

the final form of the asymptotic Gaussian is

$$
\begin{aligned}
\ln(p_A(\boldsymbol{x})) &= -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\left(\frac{\sigma^2}{(1-a^2[1])}\right) \\
&\quad - \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big] \\
&= -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\left(\ln\sigma^2 - \ln(1-a^2[1])\right) \\
&\quad - \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big] \\
&= -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma^2 + \frac{N}{2}\ln(1-a^2[1]) \\
&\quad - \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big]
\end{aligned}
$$

The difference between the exact Gaussian and the asymptotic Gaussian

$$
\begin{aligned}
&= \ln(p^{'}(\boldsymbol{x})) - \ln(p(\boldsymbol{x})) \\
&= \frac{N}{2}\ln(1-a^2[1]) + \frac{1}{2\sigma^2}\Big[a^2[1]x^2[0] + a^2[1]x^2[N-1]\Big]
\end{aligned}
$$

The difference between the likelihood functions is due to two things. The first reason is the effect of the filtering in the asymptotic Gaussian. It creates extra terms at the beginning and the end of the convolution of the input and the impulse response of the filter. The second is the inability of the asymptotic

Gaussian to correctly calculate the determinate of the auto-correlation matrix. Notice if $|a[1]| = 0$ this difference will be equal to 0. However if $a[1]$ is very close to 1 the first term would get large and cause significant errors in the result.

A.2 Distance

Starting with the derived forms of the log of the exact Gaussian $p_E(\boldsymbol{x})$ and the log of the asymptotic Gaussian $p_A(\boldsymbol{x})$.

$$\begin{aligned}\ln(p_E(\boldsymbol{x})) = &-\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma^2 + \frac{N}{2}\ln(1-a^2[1]) \\ &- \frac{1}{2\sigma^2}\Big[\, x^2[0] + 2a[1]x[0]x[1] + x^2[1] \\ &+ \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big]\end{aligned}$$

and

$$\begin{aligned}\ln(p_A(\boldsymbol{x})) = &-\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\sigma_u^2 \\ &- \frac{1}{2\sigma_u^2}\Big[x^2[0] + x^2[1] + 2a[1]x[0]x[1] + a^2[1]x^2[0] \\ &+ \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 \\ &+ a^2[1]x^2[N-1]\Big]\end{aligned}$$

Leading to the pdf's

$$p_E(\boldsymbol{x}) = \left(\frac{(1-a^2[1])}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\Big(-\frac{1}{2\sigma^2}\Big[x^2[0] + 2a[1]x[0]x[1] + x^2[1] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2\Big]\Big) \tag{A.8}$$

and

$$p_A(\boldsymbol{x}) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\Big(-\frac{1}{2\sigma_u^2}\Big[x^2[0] + x^2[1] + 2a[1]x[0]x[1] + |a[1]|^2x^2[0] + \sum_{n=2}^{N-1}(x[n] + a[1]x[n-1])^2 + |a[1]|^2x^2[N-1]\Big]\Big)$$

Therefore

$$\frac{p_E(\boldsymbol{x})}{p_A(\boldsymbol{x})} = (1-a^2[1])^{\frac{N}{2}} \exp\Big(\frac{1}{2\sigma_u^2}\Big[a^2[1]x^2[0] + a^2[1]x^2[N-1]\Big]\Big)$$

leading to

$$\ln\left(\frac{p_E(\boldsymbol{x})}{p_A(\boldsymbol{x})}\right) = \frac{N}{2}\ln(1-a^2[1]) + \frac{1}{2\sigma_u^2}\Big[a^2[1]x^2[0] + a^2[1]x^2[N-1]\Big] \tag{A.9}$$

Now it is easy to combine (A.8) and (A.9) to form the distance measurement,

$$\int_{-\infty}^{\infty}\left(p_E(\boldsymbol{x})\ln\left(\frac{p_E(\boldsymbol{x})}{p_A(\boldsymbol{x})}\right)\right)d\boldsymbol{x} \tag{A.10}$$

Substituting into (A.10)

$$\int_{-\infty}^{\infty} \left(p_E(\boldsymbol{x}) \left(\frac{N}{2} \ln(1 - a^2[1]) + \frac{1}{2\sigma_u^2} \Big[a^2[1]x^2[0] + a^2[1]x^2[N-1] \Big] \right) \right) d\boldsymbol{x}$$

now to split up the integrals,

$$\begin{aligned} &= \int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \left(\frac{N}{2} \ln(1 - a^2[1]) \right) d\boldsymbol{x} \\ &\quad + \int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \left(\frac{1}{2\sigma_u^2} a^2[1]x^2[0] \right) d\boldsymbol{x} \\ &\quad + \int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \left(\frac{1}{2\sigma_u^2} a^2[1]x^2[N-1] \right) d\boldsymbol{x} \end{aligned}$$

Now pulling the terms out that don't rely on x and since the pdf will integrate to one this can be simplified to,

$$\begin{aligned} &= \frac{N}{2} \ln(1 - a^2[1]) \\ &\quad + \frac{a^2[1]}{2\sigma_u^2} \int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \left(x^2[0] \right) d\boldsymbol{x} \\ &\quad + \frac{a^2[1]}{2\sigma_u^2} \int_{-\infty}^{\infty} p_E(\boldsymbol{x}) \left(x^2[N-1] \right) d\boldsymbol{x} \end{aligned}$$

Then realizing the $\mathcal{E}(x[n]x[n])$ is just the autocorrelation function at zero lag, and that $p_E(\boldsymbol{x})$ is the pdf of the $AR(1)$ process where $r[0]$ is defined as [14],

$$r_{xx}[0] = \frac{\sigma^2}{(1 - a^2[1])}$$

Substuting in and simplifing,

$$= \frac{N}{2} \ln(1 - a^2[1]) + \frac{a^2[1]}{(1 - a^2[1])}$$

APPENDIX B
ISOLET INFORMATION

Introduction

CSLU: ISOLET Spoken Letter Database Version 1.3, Linguistic Data Consortium (LDC) catalog number LDC2008S07 and isbn 1-58563-488-3, was created by the Center for Spoken Language Understanding (CSLU) at OGI School of Science and Engineering, Oregon Health and Science University, Beaverton, Oregon.

CSLU: ISOLET Spoken Letter Database Version 1.3 is a database of letters of the English alphabet spoken in isolation under quiet laboratory conditions and associated transcripts. The data was collected in 1990 and consists of two productions of each letter by 150 speakers (7800 spoken letters) for approximately 1.25 hours of speech. The subjects were recruited through advertising and consisted of 75 male speakers and 75 female speakers. Each subject received a free dessert at a local restaurant in exchange for his or her participation in the data collection. All speakers reported English as their native language. Their ages varied from 14 to 72 years; the speakers' average age was 35 years.

Data

Speech was recorded in the OGI speech recognition laboratory. The room measured 15' by 15' with a tile floor, standard office wall board and drop ceiling and contained two Sun workstations and three disk drives.

The recording equipment was selected to mimic the equipment used to collect the TIMIT database as closely as possible. The speech was recorded with a Sennheiser HMD 224 noise-canceling microphone, low pass filtered at 7.6 kHz. Data capture was performed using the AT∧T DSP32 board installed in a Sun 4/110. The data were sampled at 16 kHz and converted to RIFF(.WAV) format.

The subjects were seated in front of a Sun workstation and prompted with letters in random order. After each prompt, the subject would strike the return key and say the letter. Two seconds of speech were recorded and immediately played back for verification. If the subject spoke too soon or too late and missed the two-second buffer, or if the experimenter or subject decided that the letter was misspoken, the recording was repeated. There was no attempt to elicit ideal speech. A letter was judged to be misspoken only if there was a significant departure from normal pronunciation.

After the recording session, each utterance was verified by a human examiner for two determinations. First, the examiner viewed a waveform of the utterance to determine that the speech was padded with silence. The examiner then listened to the speech and noted any ambiguous or misspoken utterances. All utterances noted by the examiner were examined by two additional human examiners. If a majority of the examiners perceived that an utterance was abnormal, that utterance, and the rest of the utterances from that speaker, were removed from the corpus.

The transcriptions of the recorded speech are time-aligned phonetic transcriptions conforming to the CSLU Labeling standards. Time-aligned word transcriptions are represented in a standard orthography or romanization. Speech and non-speech phenomena are distinguished. The transcriptions are aligned to a waveform by placing boundaries to mark the beginning and ending of words. In addition to the specification of boundaries, this level of transcription includes additional commentary on salient speech and non-speech characteristics, such as glottalization, inhalation, and exhalation.

BIBLIOGRAPHY

Michle Basseville. "Distance measures for signal processing and pattern recognition". In: *Signal Processing* vol 18 issue 4, (1989), pp. 349-369.

J.P. Campbell. "Speaker recognition: a tutorial". In: *Proceedings of the IEEE* vol 85 issue 9, (Sept 1997), pp. 1437-1462.

Ronald Cole, Y Muthusamy, and Mark Fanty. *CSLU: ISOLET Spoken Letter Database Version 1.3 LDC2008S07.* https://catalog.ldc.upenn.edu/LDC2008S07. Online, accessed Oct-2017.

Ronald Cole, Y Muthusamy, and Mark Fanty. *The Isolet Spoken Letter Database.* Tech. rep. University of Oregon, 1994.

R. Donovan. "A New Distance Measure for Costing Spectral Discontinuities in Concatenative Speech Synthesisers". In: *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, (Sept 2002), pp. 223-226.

Y. Ephraim. "Statistical-model-based speech enhancement systems". In: *Proceedings of the IEEE* vol 80 issue 10, (1992), pp. 1526-1555.

K Fokianos. "Spectral estimation". In: *Wiley Interdisciplinary Reviews: Computational Statistics 2* (2010), pp. 165-170.

R.G. Gallager. *Discrete Stochastic Processes.* Boston, Massachusetts: Kluwer, 1996.

B. W. Gillespie and L. E. Atlas. "Acoustic diversity for improved speech recognition in reverberant environments". In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vol. 1. 2002, pp. I557-I560. doi: 10.1109/ICASSP.2002.5743778.

A. Gray and J. Markel. "Distance measures for speech processing". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.5 (1976), pp. 380-391. issn: 0096-3518. doi: 10 . 1109 / TASSP . 1976 . 1162849.

Ricardo Gutierrez-Osuna. *Introduction to Speech Processing.* Tech. rep. Texas A&M University, 2016.

Mark G. Hall, A.V. Oppenheim, and Alan S. Willsky. "Time-varying parametric modeling of speech". In: *Signal Processing* 5 (Dec. 1977), pp. 267-285.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing.* Upper Saddle River, New Jersey: Prentice Hall, 2001.

F. Itakura. “Minimum prediction residual principle applied to speech recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* (Feb. 1975), pp. 67-72.

Juang, B and Lawrence R. Rabiner. *Automatic Speech Recognition - A Brief History of the Technology Development.* Tech. rep. Georgia Institute of Technology, Atlanta & Rutgers University and the University of California, Santa Barbara, 2004.

Jean-Claude Junqua and Jean-Paul Haton. *Robustness in Automatic Speech Recognition.* Boston, Massachusetts: Kluwer Academic Publishers, 1996.

Steven Kay. *Intuitive Probability and Random Processes using MATLAB* . New York, New York: Springer, 2006.

Steven Kay. *Fundamentals of statistical signal processing, volume I, Estimation Theroy.* Upper Saddle River, New Jersey: Prentice-Hall, 1993.

Steven Kay. *Modern spectral estimation.* Upper Saddle River, New Jersey: Prentice-Hall, 1998.

Steven Kay. *Poisson Maximum Likelihood Spectral Inference.* Tech. rep. University of Rhode Island, 2016.

K. Kinoshita et al. “The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* 2013, pp. 1-4. doi: 10.1109/WASPAA.2013.6701894.

Esther Klabbers and Raymond N. J. Veldhuis. *On the reduction of concatenation artifacts in diphone synthesis.* Tech. rep. Jan. 1998.

Solomon Kullback. *Information Theory and Statistics.* Mineola, New York: Dover Publications, Inc, 1997.

John D. Markel and Jr Augustine H. Gray. *Linear Prediction of Speech.* 3rd ed. Berlin, Germany: Springer-Verlag, 1982.

Merriam-Webster. *Definition of robust.* https://www.merriamwebster.com/dictionary/robust. Online, accessed Nov-2017. 2017.

Alan Oppenheim and Ronald Schafer. *Discrete-Time Signal Processing.* 3rd ed. Boston, Massachusetts: Pearson, 2009.

Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech* Recognition. Englewood Cliffs, New Jersey: Prentice Hall, 1993.

Donald L. Snyder. *Random Point Processes.* Wiley-Interscience Publication, 1975.

Y. Tohkura. "A Weighted Cepstral Distance Measure for Speech Recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* (Oct. 1987), pp. 1414-1422.

O. Viikki, D. Bye, and K. Laurila. "A recursive feature vector normalization approach for robust speech recognition in noise". In: *Acoustics, Speech and Signal Processing*, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 2. 1998, 733-736 vol.2. doi: 10.1109/ICASSP.1998.675369.

Takuya Yoshioka et al. "Making Machines Understand Us in Reverberant Rooms". In: *IEEE Signal Processing Magazine* vol 29 issue 6, (Nov 2012), pp. 114-126.