A FACTORIAL HMM APPROACH TO ROBUST ISOLATED DIGIT RECOGNITION IN NON-STATIONARY NOISE

by

Ameya Nitin Deoras and Dr. Mark Hasegawa-Johnson

UNDERGRADUATE THESIS

Submitted in partial fulfillment of the requirements of
ECE 298/299 at the
University of Illinois at Urbana-Champaign
December 9, 2003

Urbana, Illinois

ABSTRACT

This paper presents a novel solution to the problem of isolated digit recognition in non-stationary noise. A Factorial Hidden Markov Model (FHMM) architecture is proposed that accurately models the simultaneous occurrence of two independent processes, i.e. an utterance of a digit and a clip of non-stationary noise. The FHMM is implemented with its equivalent HMM by extending Nadas' MIXMAX algorithm to a mixture of Gaussians PDF. The proposed system is tested on the recognition of two digits spoken simultaneously by different talkers as well as on the recognition of isolated digits mixed with background music. At 0 dB SNR, the simultaneous recognition accuracy is improved by 105% over baseline. The system also shows a relative improvement of up to 170% over baseline in the recognition of isolated digits immersed in music at 0 dB SNR.

TABLE OF CONTENTS

1. INTRODUCTION

The problem of automatic speech recognition has been effectively solved by the use of Hidden Markov Models (HMM). Quick and efficient algorithms have been developed that perform the necessary recognition tasks accurately. However, these algorithms are only effective for clean speech in very quiet environments. Most methods for clean speech recognition fail even at moderate signal-to-noise ratios (SNR).

Figure 1.1 illustrates the drop in performance of an HMM-based isolated digit recognizer on spoken digits mixed with classical music at different levels of SNR. The HMM system performs perfectly down to about 35 dB SNR. Any decrease in SNR beyond that point dramatically decreases its performance (recognition accuracy.) At 0 dB SNR, i.e. when both speech and noise have equal power, the recognition accuracy is only about 24%.

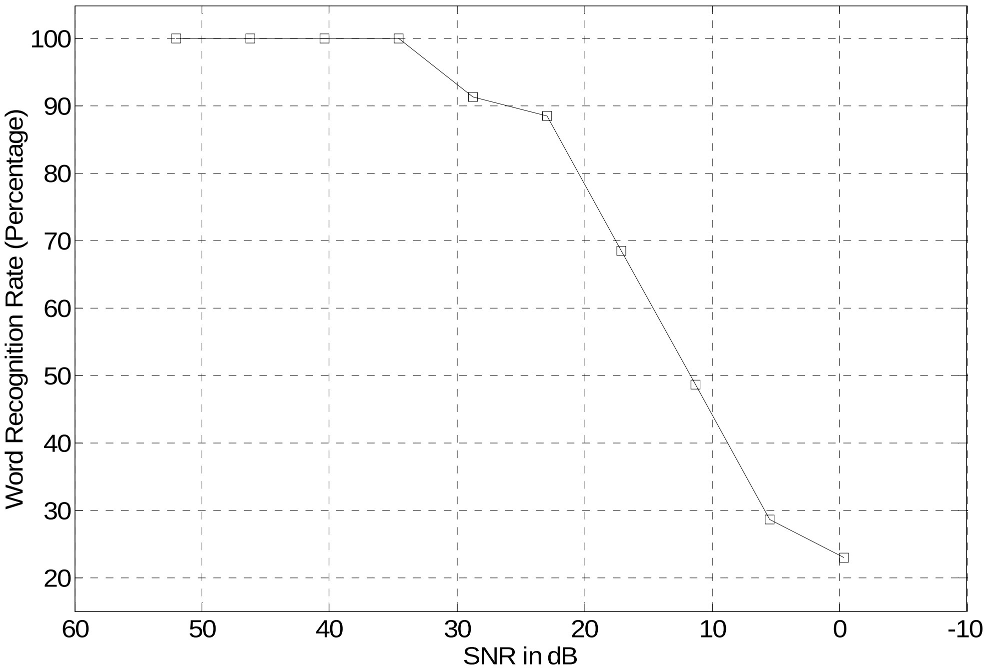

Figure 1.1. Drop in recognition accuracy of an HMM isolated digit recognizer with decrease in signal-to-noise ratio. Five utterances of each of the digits 0 through 6 were mixed with clips from Mozart's second Divertimento (third movement) at different SNRs to create the test set.

Because of such sensitivity to noise, a large number of techniques, such as spectral subtraction and Weiner filtering, have been developed to make speech recognition robust in noisy environments [1]. However, almost all of these techniques can only be applied to stationary noise, such as white noise, talking crowds or noisy machines. They do not work for noise such as music or interfering speech which is highly non-stationary. Figure 1.2 illustrates the fundamental difference between stationary noise and non-stationary noise and why techniques applicable to the former cannot be used on the latter.

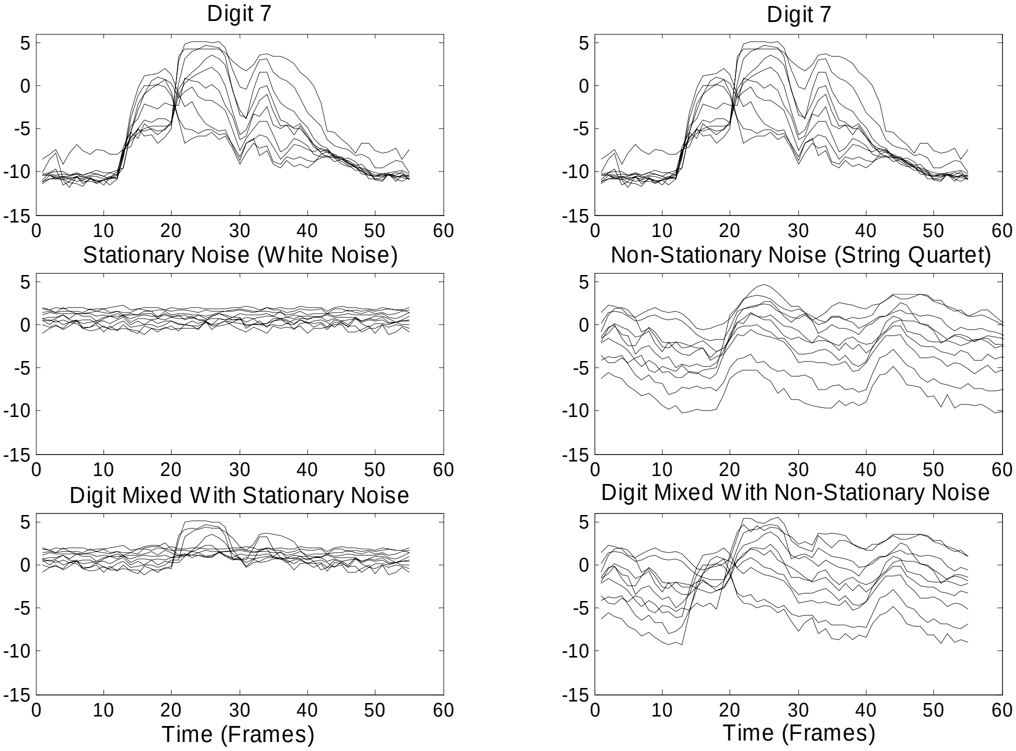

Figure 1.2. An illustration of the differences between speech in stationary noise (white noise) and speech in non-stationary noise (background string quartet). Each diagram represents an MFSC observation sequence of the process. The bottom two figures represent the digit 'seven' spoken in a stationary and a non-stationary noise environment respectively.

This paper proposes a Factorial Hidden Markov Model (Factorial HMM or FHMM) approach to solve the problem of recognition of isolated digits in non-stationary noise. The FHMM accurately models both the desired speech and undesired speech or music simultaneously to create an effective speech recognizer in non-stationary noise, even at low SNR.

The concept of the Factorial HHMM was first developed by Ghahramani and Jordan [2] as an alternative to traditional HMMs. It has been shown that factorial HMMs are better suited to model loosely coupled random processes than HMMs [2], [3]. Furthermore, efficient algorithms for the estimation of parameters of FHMMs have also been developed [2]. The approach presented in this paper is, however, a little different. The FHMM architecture is used to combine two existing HMMs of independent random processes, like the simultaneous utterance of two digits or an utterance of a digit and a piece of music. Since the isolated digit HMMs have already been trained, no additional training of the FHMM is required. Roweis [4] has shown that an FHMM can be used in such a way to model audio signals from different sources for a computational auditory scene analysis application. I build on Roweis' method for the simultaneous recognition of two digits as well as for the recognition of isolated digits with background music.

The motivation behind this model arose from the observed interaction of the log spectra of two signals that are additive in the time domain. Nadas et al. [5] have shown that an additive combination of two sound signals $\bar{Y}(j\omega)=\bar{X}(j\omega)+\bar{Z}(j\omega)$ can be accurately modeled by the element-wise maximum of their log magnitude spectra. This is referred to as the MIXMAX approximation.

$$\log|\bar{Y}(j\omega)| \approx \max(\log|\bar{X}(j\omega)|, \log|\bar{Z}(j\omega)|) \qquad (1.1)$$

The MIXMAX approximation also holds for Mel Frequency Spectral Coefficients (MFSC) [6] as shown in Figure 1.3. Using Equation (1.1), a complete model of the FHMM can be derived and the system can be implemented.

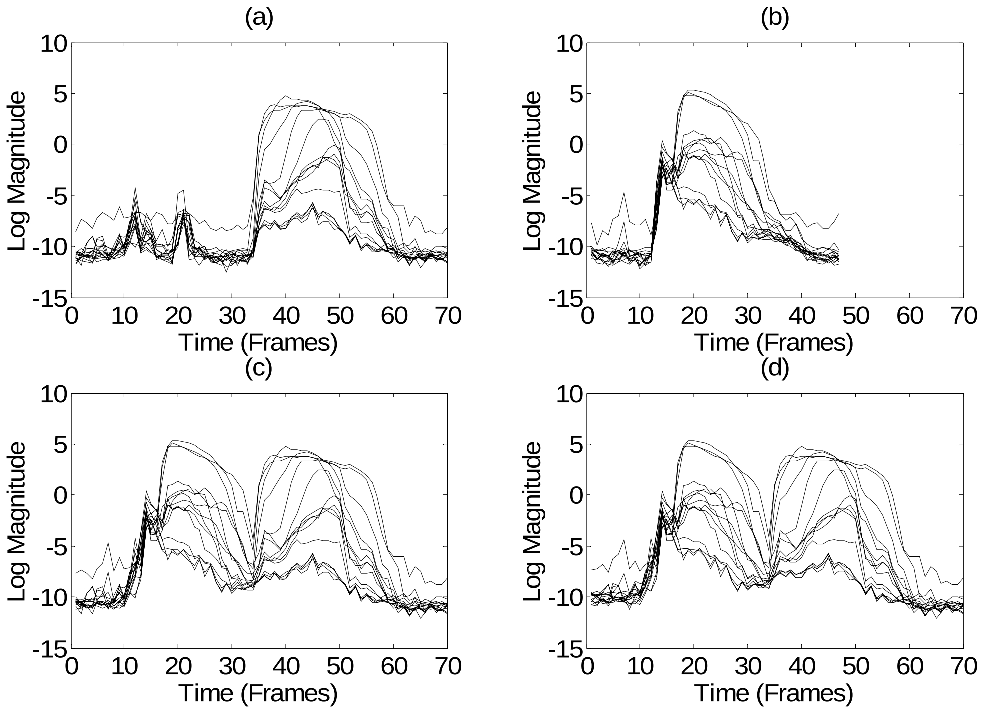

Figure 1.3. Log spectrum MIXMAX approximation; (a) MFSC observation sequence of the digit ‘ONE’; (b) MFSC observation sequence of the digit ‘TWO’; (c) the element-wise maximum of sequences (a) and (b); (d) the MFSC observation sequence generated from the addition of the utterances of the digits ‘ONE’ and ‘TWO’ in the time domain.

2. THEORETICAL BACKGROUND

2.1 Hidden Markov Models

Hidden Markov Models (HMM) are currently the predominant methodology for state-of-the-art speech recognition. Since its inception over thirty years ago, the theory of HMMs has been extensively developed to create efficient algorithms for training (Expectation-Maximization, Baum Welch re-estimation) and recognition (Viterbi, Forward-Backward) [7]. In this section, I present a basic overview of HMMs and the details relevant to the derivation of the proposed FHMM model.

An HMM can be defined as "a doubly embedded stochastic process with an underlying stochastic process that is not directly observable but can be observed only through another set of stochastic processes that produce the resulting sequence of observations" [7]. It can be visualized as a state machine with hidden discrete states but observable outputs. At each time step, the system undergoes a state transition specified by a transition probability matrix **A** and proposes an output (MFSC) vector $\overline{x}_t$ based on an output probability distribution function $b_q(\overline{x}_t)$ for that state. In the case of speech, the PDF is assumed to be a mixture of Gaussians observation PDF.

$$\begin{aligned} b_q(\overline{x}_t) &= \sum_{m=1}^{M} c_{q,m} N(\overline{x}_t \mid \overline{\mu}_{q,m}, \Sigma_{q,m}) \\ N(\overline{o} \mid \overline{\mu}_{j,m}, \Sigma_{j,m}) &= \frac{1}{\sqrt{(2\pi)^n \mid \Sigma_{j,m} \mid}} e^{-\frac{1}{2}(\overline{o}-\overline{\mu}_{j,m})' \Sigma_{j,m}{}^{-1} (\overline{o}-\overline{\mu}_{j,m})} \end{aligned} \tag{2.1}$$

where $b_q(\overline{x}_t)$ is the probability of observing the vector $\overline{x}_t$ in state *q*.

Therefore, an HMM can be completely defined by its initial state distribution vector $\boldsymbol{\pi}$, its transition matrix ***A***, and its output PDF matrix ***B*** (which contains the mixture weights, means and variances of each mixture Gaussian for each state). A simple HMM with four-states is shown in Figure 2.1.

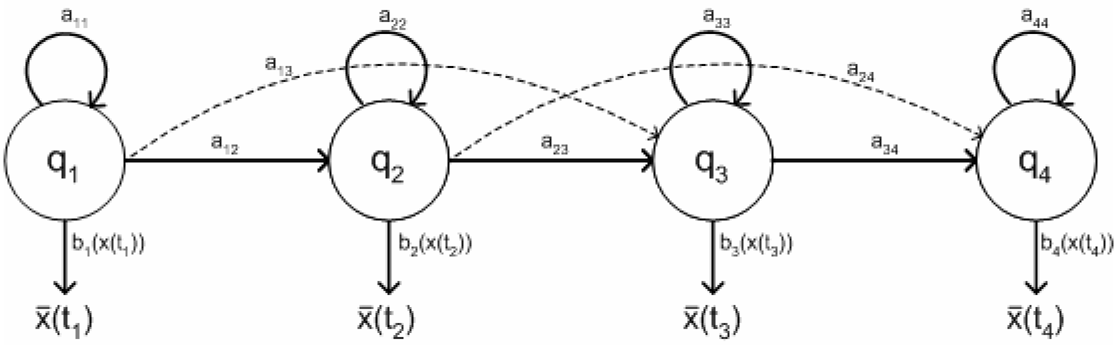

Figure 2.1. A four-state, left-right HMM. The variable a_{ij} is the transition probability from state *i* to state *j* and $b_i\ (\bar{x})$ is as defined in Equation (1.1).

To train the model, an initial guess is made for the parameters $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ using a K-means clustering algorithm [7]. Then, given a large number of observations **X,** the parameters are adjusted using the Baum-Welch re-estimation algorithm to maximize $P(\mathbf{X} \mid \lambda)$, where λ represents the HMM [7], [8].

Recognition is performed by using the Viterbi algorithm to find the optimal state sequence and the Forward-Backward algorithm to score the model by determining the probability $P(\mathbf{X} \mid \lambda)$ [7], [8].

2.2 Front-End Speech Processor

In addition to an HMM, most speech recognition systems have some kind of front-end speech processor which extracts useful phonetic information from the speech and discards all other irrelevant information. The front-end achieves this by performing a vector parameterization of the signal, which is motivated by human speech perception [6]. It has been shown that because of the differences in the ear's critical bandwidths at different frequencies, filters spaced linearly at low frequencies and logarithmically at high frequencies (a mel frequency scale) accurately capture the phonetically relevant characteristics of speech [6].

Mel Frequency Spectral Coefficients (MFSCs) are defined as the log-energy outputs of the speech signal after it is filtered by a bank of triangular bandpass filters on a mel-frequency scale (linear below 1 kHz and logarithmic above 1 kHz) [6]. Figure 2.2 illustrates a sample mel filter bank with 10 filters.

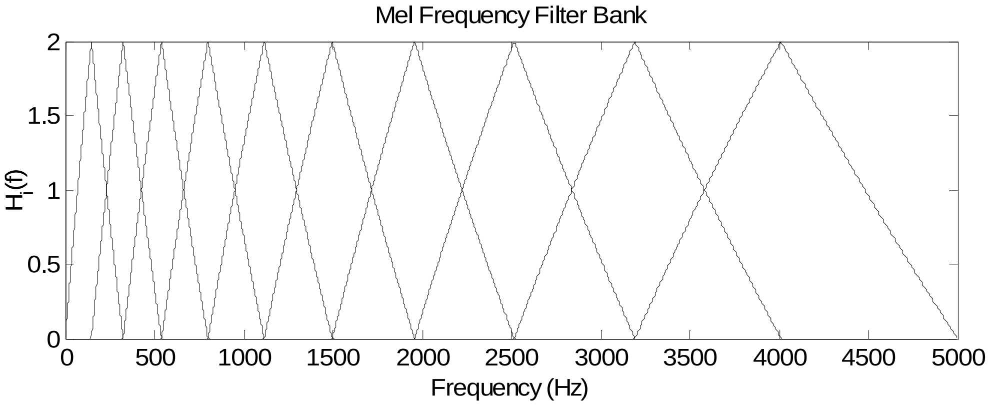

Figure 2.2. A 10-filter mel filter bank for a signal bandlimited to 5000 Hz.

Mathematically, MFSC observation vectors are calculated in the following way,

$$MFSC_i = \log\left(\sum_k \left|X_t(k)H_i(k)\right|^2\right) \qquad (1 < i < M) \qquad (2.2)$$

where $X_t(k)$ and $H_i(k)$ are FFT representations of the signal and the i^{th} filter response respectively.

MFSCs were specifically chosen for this project for two reasons. First, since they are log-spectral signals (unlike Mel Frequency Cepstral Coefficients which are temporal), the MIXMAX approximation can be applied to speech represented by MFSCs as described in Equation (1.1). Second, it has been shown that MFSCs are the best parameterized representation for music signals [9]. Since one of the primary objectives of this paper was the recognition of mixtures of speech and music, MFSC observation vectors were, therefore, a natural choice.

2.3 Factorial Hidden Markov Models

The Factorial HMM recognizer presented in this paper is based on a system proposed by Sam T. Roweis [4]. However, unlike Roweis' source separation system, the objective of this design is to successfully recognize two signals (either two digits or a digit and music) simultaneously without separating them.

A two-chain Factorial HMM is shown in Figure 2.3. It consists of two underlying HMM chains that evolve independently of each other. The output of the FHMM in every frame is the element-wise maximum of the output vectors proposed by each chain independently as expressed in Equation (1.1).

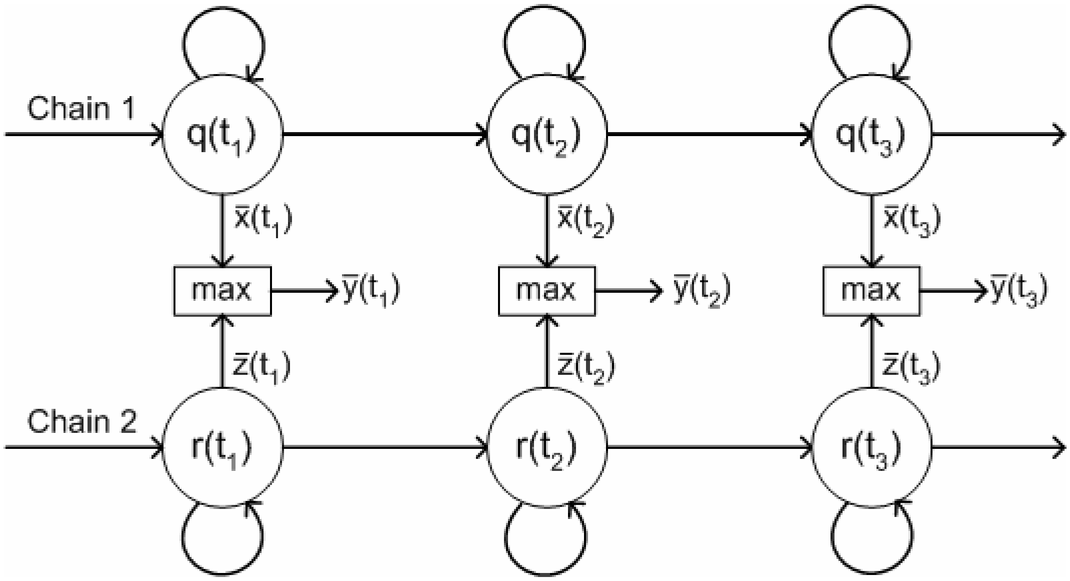

Figure 2.3. A Factorial Hidden Markov Model of two HMM chains with states *q(t)* and *r(t)* and outputs *x(t)* and *z(t)* respectively. The output *y(t)* of the FHMM in each frame is the maximum of the outputs proposed by the HMMs.

Because of the hidden nature of the states of the HMMs composing the FHMM, the recognition task becomes very difficult. Also, because the output of the FHMM is a function of the outputs of the two HMMs, during recognition, a joint PDF and a joint state sequence have to be inferred. Furthermore, the efficient Viterbi and Forward-Backward algorithms for HMM recognition cannot be used. A solution to this problem is presented in Chapter 3 followed by a derivation of the mathematical implementation of the FHMM.

3. MATHEMATICAL IMPLEMENTATION

In the previous chapter, a theoretical model (an FHMM) was described that could model two independent simultaneous processes, like an utterance of speech and an excerpt of non-stationary noise. However, the inference of the best state trajectory and calculation of the posterior probability $P(\lambda \mid \mathbf{X})$ are complicated because of the existence of a joint state sequence and joint PDF of the FHMM. Techniques for calculating such posterior probabilities of Factorial HMMs have been formulated by Logan [3] and Ghahramani [2] to overcome the large computations involved. For the same reason, Roweis [4] uses a log probability upper bound to compute the best joint state trajectory.

I propose a simpler approach that takes advantage of the MIXMAX approximation as well as the efficient recognition algorithms (Viterbi and Forward-Backward algorithms) that are already available for HMMs. I propose an implementation of the FHMM by transforming it into its topologically equivalent HMM. To implement the FHMM in this way, the equivalent HMM's initial state distribution, transition probability matrix and output probability distributions are derived.

The initial state distribution of the FHMM is derived by inspection. As both HMM chains composing the FHMM start in their respective state 1's, the FHMM will start in the state (1,1) .

$$\begin{matrix} \boldsymbol{\pi}_1 = [1, 0, 0, ..., 0] \\ \boldsymbol{\pi}_2 = [1, 0, 0, ..., 0] \end{matrix} \Rightarrow \boldsymbol{\pi}_{\text{eq}} = [1, 0, 0, ..., 0] \tag{3.1}$$

The transition matrix of the equivalent HMM is computed by inspecting all possible state transitions in the individual chains of the FHMM. A discussion on the computation of the transition matrix is presented in Section 3.1.

The output probability distribution for each state of the equivalent HMM is derived in Sections 3.2 and 3.3 by extending the results from Nadas' MIXMAX algorithm [5] to the case of a mixture of Gaussians PDF.

3.1 Computation of the Transition Matrix

Consider a Factorial HMM, as shown in Figure 2.3, with two chains (denoted by a superscript index) containing Q and R states respectively. This FHMM can be shown to be topologically equivalent to an HMM with $Q \times R$ states [3]. The transition matrix for the equivalent HMM can be computed in the following manner:

$$a^{FHMM}(i,j \to k,l) = a^{1}_{i\to k} \times a^{2}_{j\to l} \qquad \begin{array}{l} 1 \le i,k \le Q \\ 1 \le j,l \le R \end{array} \tag{3.2}$$

where the states of the equivalent HMM are indexed by the pair of state indices of chains 1 and 2. This relationship can be better conceptualized by Equation (3.3) which expands Equation (3.2) for the case when $Q = R = 2$.

$$\begin{array}{c} 1 \\ 2 \end{array}\!\begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix} \otimes \begin{array}{c} 1 \\ 2 \end{array}\!\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \to \begin{array}{c} 1,1 \\ 1,2 \\ 2,1 \\ 2,2 \end{array}\!\begin{bmatrix} a_{11}A_{11} & a_{11}A_{12} & a_{12}A_{11} & a_{12}A_{12} \\ 0 & a_{11}A_{22} & 0 & a_{12}A_{22} \\ 0 & 0 & a_{22}A_{11} & a_{22}A_{12} \\ 0 & 0 & 0 & a_{22}A_{22} \end{bmatrix} \tag{3.3}$$

3.2 The Output Probability Distribution

Let the state indices of the two independent HMM chains (denoted by a superscript index) that compose the FHMM be *q*(*t*) and *r*(*t*), and let the proposed MFSC observation vectors be $\overline{x}(t)$ and $\overline{z}(t)$ respectively. The output of the FHMM is given by,

$$\overline{y}(t) \;=\; \max(\overline{x}(t), \overline{z}(t)) \tag{3.4}$$

where $\max(\overline{x}(t), \overline{z}(t))$ is the element-wise maximum. Since the two processes are independent, it follows from Equation (3.4) that,

$$F_{\overline{y}}(\overline{\lambda}) = F_{\overline{x}}(\overline{\lambda}) F_{\overline{z}}(\overline{\lambda}) \tag{3.5}$$

where $F_{\overline{y}}(\overline{\lambda}) = P(\overline{y} < \overline{\lambda}) = \int_{-\infty}^{\overline{\lambda}} p_{\overline{y}}(\overline{y}) d\overline{y}$ is the CDF of $\overline{y}$.
Differentiating Equation (3.5) gives us the PDF of $\overline{y}(t)$.

$$p_{\overline{y}}(\overline{\lambda}) = p_{\overline{x}}(\overline{\lambda}) F_{\overline{z}}(\overline{\lambda}) + p_{\overline{z}}(\overline{\lambda}) F_{\overline{x}}(\overline{\lambda}) \tag{3.6}$$

which is an important result also obtained by Nadas et al. [5].

Although Equation (3.6) defines the PDF for each state of the equivalent HMM, it is not directly applicable to the multivariate mixture Gaussian PDF defined in Equation (2.1). In the next Section, Nadas' result for the PDF from the MIXMAX approximation, Equation (3.6), is extended to a mixture of Gaussians PDF and the derivation of a fully implementable theoretical model is completed.

3.3 Extension to a Mixture of Gaussians PDF

Since each HMM state has an output probability density function represented by a mixture of Gaussians, we can re-write Equation (2.1) for the case of a Factorial HMM with two chains denoted with an index in the superscript,

$$\begin{aligned} b_q^1(\bar{x}_t) &= \sum_{m=1}^{M} c_{qm}^1 N(\bar{x}_t \mid \bar{\mu}_{qm}^1, \Sigma_{qm}^1) \\ b_r^2(\bar{z}_t) &= \sum_{m=1}^{M} c_{rm}^2 N(\bar{z}_t \mid \bar{\mu}_{rm}^2, \Sigma_{rm}^2) \end{aligned} \tag{3.7}$$

$$N(\bar{o} \mid \bar{\mu}_{jm}, \Sigma_{jm}) = \frac{1}{\sqrt{(2\pi)^n \mid \Sigma_{jm} \mid}} e^{-\frac{1}{2}(\bar{o}-\bar{\mu}_{jm})'\Sigma_{jm}{}^{-1}(\bar{o}-\bar{\mu}_{jm})}$$

where *M* is the number of Gaussians in each mixture, *c* is the mixture weight and *n* is the dimensionality of the output vectors.

Therefore, the FHMM output PDF result from Equation (3.6) can be written for the case of HMM output PDFs defined in Equation (3.7).

$$b_{q,r}(\bar{y}_t) = b_q^1(\bar{x}_t) \int_{-\infty}^{\bar{y}_t} b_r^2(\bar{z}_t) d\bar{z}_t + b_r^2(\bar{z}_t) \int_{-\infty}^{\bar{y}_t} b_q^1(\bar{x}_t) d\bar{x}_t \tag{3.8}$$

where $b_q^i(\bar{x}_t)$ represents the probability of seeing MFSC vector $\bar{x}_t$ in state *q* of HMM chain *i*.

The integrals in Equation (3.8) are the cumulative density functions of $b_q^1(\bar{x}_t)$ and $b_r^2(\bar{z}_t)$. The crux of the problem now becomes the implementation of the CDF of a mixture of Gaussians. To simplify the problem, the d-variate Gaussians are assumed to be diagonal covariance Gaussians. This assumption reduces the order of computation from O(n^d) to O(nd) and enables the representation of a multivariate Gaussian as the product of *d* univariate Gaussians. Equation (3.7) can, therefore, be written as,

$$b_q^1(\bar{x}_t) = \sum_{m=1}^{M} c_{qm}^1 \prod_{p=1}^{n} \frac{1}{\sigma_{qm,p}\sqrt{(2\pi)}} e^{-\frac{1}{2}\left(\frac{x_{t,p}-\mu_{qm,p}}{\sigma_{qm,p}}\right)^2} \tag{3.9}$$

where $\sigma_{qm,i}^2$ is the element at position *(i,i)* on the diagonal of the covariance matrix Σ_{qm}. The integral of the diagonal covariance Gaussian mixture is therefore given by,

$$\int_{-\infty}^{\bar{y}_t} b_q^1(\bar{x}_t) d\bar{x}_t = \int_{-\infty}^{\bar{y}_t} \left(\sum_{m=1}^{M} c_{qm}^1 \prod_{p=1}^{n} \frac{1}{\sigma_{qm,p}\sqrt{(2\pi)}} e^{-\frac{1}{2}\left(\frac{x_{t,p}-\mu_{qm,p}}{\sigma_{qm,p}}\right)^2} \right) d\bar{x}_t \tag{3.10}$$

The integral of the sum of the Gaussians is equivalent to the sum of the integral of the Gaussians. Also, since we are representing each *n*-variate Gaussian as a product of *n* univariate Gaussians, the integral to $\overline{y}_t$ of the product of Gaussians is equal to the product of the integrals of each Gaussian to $y_{t,p}$, where $\overline{y}_t = \left[y_{t,1}, y_{t,2}, ..., y_{t,n} \right]$. Therefore, the CDF of a mixture of *n*-variate Gaussians can be written,

$$\int_{-\infty}^{\overline{y}_t} b_q^1(\overline{x}_t) = \sum_{m=1}^{M} c_{qm}^1 \prod_{p=1}^{n} \left(\int_{-\infty}^{y_{t,p}} \frac{1}{\sigma_{qm,p}\sqrt{(2\pi)}} e^{-1/2\left(\frac{x_{t,p}-\mu_{qm,p}}{\sigma_{qm,p}}\right)^2} dx_{t,p} \right) \tag{3.11}$$

In other words, the CDF of every *n*-variate diagonal covariance Gaussian is simply the product of the CDFs of the *n* univariate Gaussians that it is composed of.

The result in Equation (3.11) is an easily implementable formula for calculating the CDF of a mixture of diagonal covariance Gaussians. Therefore, Equations (3.8), (3.9) and (3.11) alone completely define the output PDF of the FHMM and its equivalent HMM.

3.4 Computational Complexity

While the proposed implementation of the Factorial HMM is intuitive and does not require the implementation of new algorithms, it is computationally challenging. An FHMM consisting two chains with *Q* and *R* states respectively, when transformed, results in an equivalent HMM with $Q \times R$ states. Since the computational cost of the recognition algorithms for HMMs is proportional to the square of the number of states [7], implementation by an equivalent HMM dramatically increases the recognition time for each pair of HMMs. For example, when two isolated digit HMMs are combined factorially, the equivalent HMM requires 16 times the number of computations required by the two digit HMMs together. For this project, however, ease of implementation was the major consideration and, therefore, an equivalent HMM implementation was chosen regardless of computational efficiency.

4. EXPERIMENTAL SYSTEM

This chapter describes the training and testing of the FHMM recognizer for isolated digits in non-stationary noise. The system was designed and tested in Matlab with the help of routines provided by Kevin Murphy in the Hidden Markov Model Toolbox for Matlab [10]. Figure 4.1 summarizes the implementation and design procedure followed in the creation of the FHMM system.

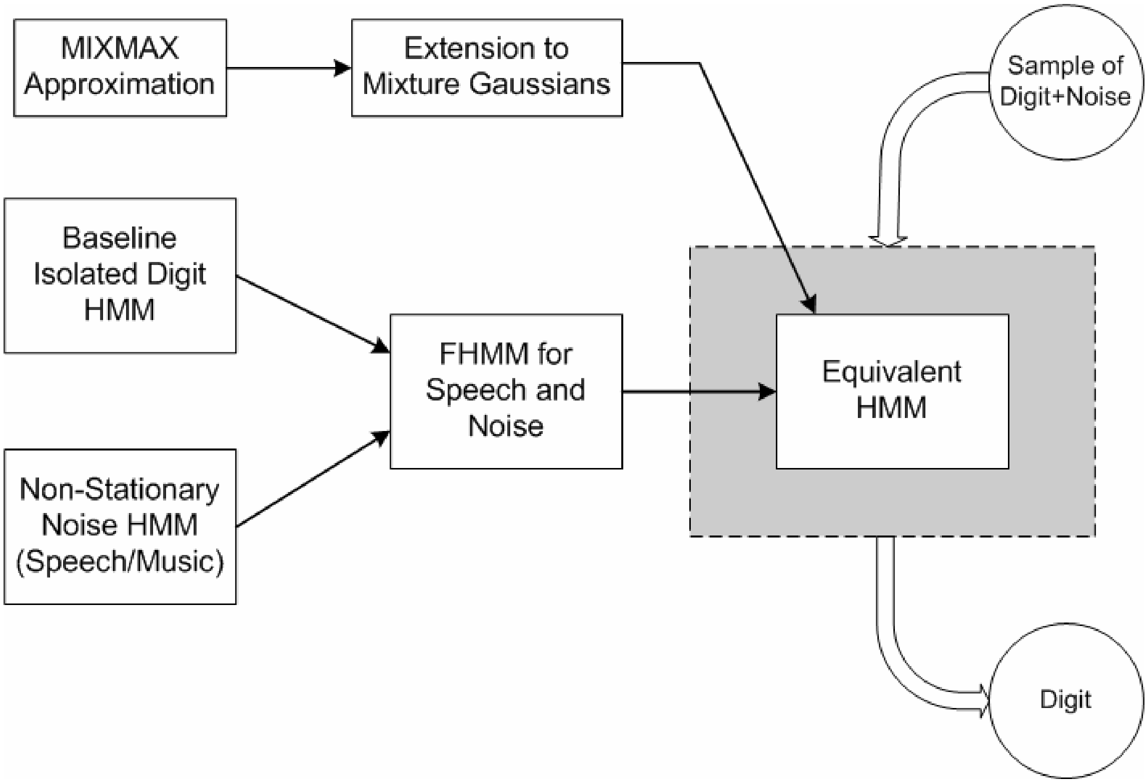
MIXMAX Approximation
Extension to Mixture Gaussians
Sample of Digit+Noise
Baseline Isolated Digit HMM
FHMM for Speech and Noise
Equivalent HMM
Non-Stationary Noise HMM (Speech/Music)
Digit

Figure 4.1. Summary of the implementation of the FHMM recognizer for isolated digits in non-stationary noise.

The system was tested on two different types of non-stationary noise, interfering speech and music. In all tests, two HMMs, one for clean speech and one for non-stationary noise were combined factorially to create an FHMM. The FHMM was then scored on an observation sequence of a digit and noise. This procedure was carried out on all possible digit-noise FHMM combinations to find the best model.

Sections 4.1 and 4.2 describe the training of the speech and music models respectively. The testing and results using each type of non-stationary noise are presented in Sections 4.3 and 4.4 respectively.

4.1 Isolated Digit HMM

The isolated digit speech recognition system was implemented with 10 HMMs, one for each digit from zero to nine. Each HMM was designed with 8 states and 120 Gaussian mixtures per state. In the front-end pre-processor, feature vectors were created by windowing 32ms of data with a Hamming window and computing 20 MFSCs as described in Equation (2.2). Consecutive frames were allowed 16ms of overlap. Adjacent frames of MFSC vectors were then concatenated (to create continuity between frames) to produce the observation sequence. The values of the parameters were optimally chosen based on empirical data presented by Rabiner [7].

Each digit HMM was trained on 100 utterances of that digit spoken by 50 male speakers from the NIST/TIDIGITS speech corpus [11]. The test data used was a subset of the 10 utterances of each digit by 5 speakers not included in the training set. On clean speech, the baseline system performed with a word recognition error rate of 3%.

4.2 Classical Music HMM

It has been shown that HMMs with MFSC feature vectors can be used to accurately model music for genre classification purposes with an accuracy up to 80% [9]. The model of musical ‘noise’ was therefore designed and trained like the isolated digit model, with clips of music used as ‘utterances’[1] instead of spoken digits.

4.2.1 Training Procedure

To reduce the complexity of the recognition task, the music was restricted to classical music, specifically to string quartets. The classical music HMM was trained on Mozart’s Second Divertimento, movement 2 Allegro Molto. The observation sequence was computed by taking consecutive 1.28 s long clips (80 frames) of data from the wave file and generating MFSC observation vectors exactly like the isolated digit front-end. The HMM was then trained using the K-Means clustering algorithm and the Baum-Welch re-estimation algorithm [7], [8], [10]. The front-end parameters used for the music HMM (window size, overlap, length of MFSC vectors) were deliberately chosen to be identical to those of the speech front-end because, during recognition, the combination of speech and music are processed with the same front-end processor.

[1] A music clip is referred to as an ‘utterance’ even though it does not contain any speech. This is done only to simplify the semantics of a digit-music combination.

4.2.2 Power Scaling

The music data had to be scaled down to reduce its power to a level comparable to that of the speech data. This requirement follows directly from the MIMAX approximation. As described in Equation (3.4) and illustrated in Figure 1.3, the output of the FHMM in each frame is equal to the element-wise maximum of the outputs of the two chains it is composed of, i.e. the digit and the music HMM chains.

$$\overline{y}(t) \;=\; \max(\overline{x}(t), \overline{z}(t)) \tag{4.1}$$

where $\overline{x}(t)$ represents the speech and $\overline{z}(t)$ represents the music. If in any frame, the music vector is always greater than the speech vector, the speech is lost completely.

$$\text{If } \overline{z}(t) \geq \overline{x}(t), \text{ then } \overline{y}(t) = \overline{z}(t) \tag{4.2}$$

Therefore, the two observation vectors have to be scaled appropriately so that Equation (4.2) is never true in any frame. Since the music is the undesired signal, it is scaled down to be of comparable power to the desired speech.

The scaling factor was determined by computing the SNR in each frame of a large number of mixtures of speech and music. The minimum SNR over all frames of an observation sequence was regarded as the true SNR for that mixture. The mean of the true SNRs of all combined utterances was used to compute the scaling factor. The resultant power scaling factor was found to be 0.31.

4.2.3 The Optimal Model

The optimal classical music HMM parameters were determined experimentally. A large number of classical music HMMs were trained by varying the numbers of states, the number of mixture Gaussians per state, the set of allowed transitions (left-right or ergodic) and the number of utterances used in recognition. The performance of the resulting FHMM (when combined with speech) was then compared for all the models. The FHMM was tested on digits 0, 1 and 2 mixed with clips from the third movement, Allegro Assai, of Mozart's second Divertimento. Two utterances of each digit were mixed with two clips from different sections of the piece. The results from the HMM parameter variation tests are presented in Table 4.1. The optimal model was found to be a left-right HMM with 6 states and 1 Gaussian mixture per state, with a training set containing 40 clips of music.

TABLE 4.1.
Variation of Performance of Classical Music HMMs with Variation in Parameters

States	Mixtures	Scaling factor	Training Set Size	Observation Sequence Length (Frames)	Recognition Rate	Average SNR
6	1	0.3146	40	80	6 out of 6	1.13
6	1	0.3146	40	100	4 out of 6	0.33
6	1	0.4116	40	80	4 out of 6	1.13
6	2	0.3146	80	80	4 out of 6	1.13
8	4	0.3146	80	80	4 out of 6	1.13

4.3 Simultaneous Multiple Digit Recognition

In the first set of performance tests, interfering speech was used as non-stationary noise. The interfering speech was restricted to isolated digits because an HMM of isolated digits was already available. Since Factorial HMMs model the simultaneous occurrence of two processes, in this case, two speech signals, the problem of isolated recognition with interfering digits became equivalent to the problem of recognizing two different digits spoken simultaneously. To test the system's performance as a multiple digit recognizer, the following two recognition tasks were devised.

The first task given to the system was that of isolated digit recognition with interfering digits. One of the digits of the double-digit pair was treated as the 'signal' and the other as a known 'interference' at 0db SNR. The system's performance in recognizing the signal digit, given the knowledge of the interference digit, was then tested for different combinations of digits. In the second set of experiments, the system was not given any prior information on either digit and its simultaneous double-digit recognition performance was evaluated.

In each test, two utterances (produced by different speakers) were combined at 0db SNR in the time domain, followed by the computation of an MFSC observation sequence using the same front-end used during training. The recognition system was then presented with the task of finding the Factorial HMM (out of all combinations of allowed digits) that best modeled the mixed utterance.

The details and test results from both tasks are presented in Sections 4.3.1 and 4.3.2.

4.3.1 Recognition of a 'Signal' Digit Given an 'Interference' Digit

In this test, as described above, the system was given a mixed utterance of a known (interference) and an unknown (signal) digit. The recognition system then had to decide which digit's HMM, when combined with the HMM of the 'interference' digit, produced an FHMM that best modeled the mixed utterance.

The baseline system's performance in this task was also evaluated for comparison. This was done by comparing log-likelihoods of the utterance mixture produced by different HMMs (except the interference digit HMM) and then choosing the digit HMM with the highest likelihood as the recognized 'signal' digit.

The results from the tests are presented in Table 4.1. The multiple-digit recognition system showed an average relative improvement in word accuracy of 35% over baseline.

TABLE 4.1
Average recognition rates of the 'signal' digit, given the 'interference' digit. The column *N* represents the number of utterances of each digit. The results presented are percent (number) correct.

Allowed Digits	Interference Digit	*N*	FHMM Recognition Rate		Baseline Recognition Rate
			Allowing FHMMs of pairs of the same digit	Not allowing pairs of the same digit	
0 - 5	4	1	83%	100%	67%
0 - 8	3	5	64%	68%	55%

4.3.2 Simultaneous Double-Digit Recognition

In this set of tests, as described above, the system was given a mixed utterance of two unknown digits. It was then presented with the task of identifying both the digits by comparing log-likelihoods of the mixed utterance from all allowed FHMM combinations. A successful recognition of both digits was labeled a ‘complete success’ (CS) and a recognition of one of the digits in the pair was labeled a ‘partial success, partial failure’ (PSPF). The recognition rate was computed by,

$$\text{Recognition Rate (\%)} = \frac{CS + 0.5 \times PSPF}{N_{trials}} \tag{4.3}$$

where N_{trials} is the total number of recognitions performed.

Table 4.2 lists the results from the double-digit recognition tests. The FHMM system showed an average relative improvement in word accuracy of 105% over baseline.

TABLE 4.2
Average recognition rates for the task of simultaneous double-digit recognition. The *CS*/*N* column represents the ‘complete success’ recognition rate, assuming no models of pairs of the same digit. The column *N* represents the number of utterances of each digit. The results presented are percent (number) correct.

Allowed Digits	N	$\frac{CS}{N}$	FHMM Recognition Rate		Baseline Recognition Rate
			Allowing FHMMs of pairs of the same digit	Not allowing pairs of the same digit	
5 – 8	1	100%	-	100%	67%
4 – 8	4	78%	-	89%	36%
1 – 5	5	70%	79%	84%	38%

4.4 Recognition of Isolated Digits in Music

In the second set of performance tests, classical music was used as the non-stationary noise. An excerpt of music was combined with an utterance of a digit in the time domain, followed by an MFSC observation sequence computation using the same front-end used during training. The system was then presented with the task of determining the digit which when combined with the music HMM produced an FHMM that best modeled the mixed utterance. Different parameters were varied to study the system’s performance at this task. The baseline performance was obtained by performing recognition of the noisy speech using the isolated digit HMMs.

The music sample used in the tests was generated by taking 80 frames of data (around 1.28 s), either sequentially or randomly, from a piece of music in *wav* format. This was followed by scaling by a factor of 0.31 to reduce the power of the music to a level comparable to the power of the speech signal. These parameters had been determined empirically during the testing of the baseline music HMM.

The set of allowed music clips was limited to classical string quartets because the music HMM had been trained on this genre. The two pieces of music used for the generation of the music data were the first (*Andante*) and third (*Allegro Assai*) movements of Mozart's second Divertimento. The second movement was faster and quicker as opposed to the first movement, which was slower and quieter. None of the data from the training set was allowed during recognition.

The results from each test are presented in Table 4.3. For the task of isolated digit recognition in background music, the FHMM system showed an average relative improvement in word accuracy of 170% over baseline.

TABLE 4.3
Average recognition rates for the task of isolated digit recognition in background music. The column *N* represents the number of utterances of each digit. The recognition rates presented are percent (number) correct.

Music File	Clip/ Speaker Sequence	Allowed Digits	*N*	Baseline Recognition		FHMM Recognition	
				Recog. Rate	Ave. SNR	Recog. Rate	Ave. SNR
Allegro Assai	Random	0 - 6	5	17%	-0.39	65%	-0.65
Allegro Assai	Sequential	0 - 6	5	23%	0.55	66%	0.55
Andante	Random	3 - 8	5	30%	4.34	73%	4.77
Andante	Sequential	1 - 8	8	30%	10.43	66%	10.43

5. DISCUSSION

In the previous chapter, the Factorial HMM was tested on three distinct tasks, two dealing with recognition in non-stationary interfering speech and the one with recognition in non-stationary music. The three tasks presented to the system started with the simplest (recognition of a signal digit given an interfering digit) and progressively increased in complexity to the second (double digit recognition) and the third (recognition of speech in music) tasks. The FHMM system performed consistently at all three tasks (with recognition rates from 65% to 85%). However, the baseline system performance dropped consistently from about 55% in task 1 to 38% in task 2 to about 20% in task 3.

5.1 Simultaneous Multiple Digit Recognition

In the first and second tasks, the baseline system performed better than was first indicated by Figure 1.1. At SNRs of around 0db, it produced recognition rates of 40 to 50%. This is due to the nature of the database used for testing. All the samples in the TIDIGITS database [11] contained different durations of silence before and after the utterance of the digit. Because of this, when two utterances were combined, the actual utterances of the digits did not always align perfectly. This effect can even be seen in Figure 1.3, where the digits ‘one’ and ‘two’ can easily be distinguished in their MFSC combination sequence.

Although the baseline system performed better than expected, it is still interesting to note that while the performance of the baseline system dropped in the second task, the performance of the FHMM system improved. This is due to the increase in complexity of the second task. While the FHMM was performing the same operation in both tasks, i.e. the search for the best double digit combination, the baseline system was not. It was carrying out single digit recognition in the first task and double digit recognition in the second. In the second task, there were a larger number of possible choices for the digit pair, which led to a drop in performance. The improvement in the performance of the FHMM system in the second task was probably due to the increase in the allowed number of successes as implied by Equation(4.3). In task 1, the FHMM only had to recognize one digit in the combination, because of which, a success was defined as the correct recognition of the ‘signal’ digit. However, in the second task, recognition of one of the digits in the pair was counted as a partial success. Because of this the FHMM demonstrated a higher overall accuracy in modeling the more diverse set of digit combinations.

Another observation that can be made from the results is that the FHMM system cannot easily model simultaneous utterances of the same digit by different speakers. All the tests show an improvement in performance when pairs of the same digit are not allowed. This is because in the FHMM structure, each chain is competing to explain the same observation sequence, resulting in a drop in performance.

5.2 Isolated Digit Recognition in Music

It is interesting to note that the FHMM system is not as sensitive to changes in SNR as the baseline system. In the first two tests, while the recognition rate of the baseline system increased by 29% due to an increase of 0.9 db in SNR, the FHMM recognition rate increased only by 1.1% (for, in fact, a larger increase in SNR). This is because while the baseline HMMs try to identify the speech despite the music noise, the FHMM tries to identify both. Therefore, as long as the MIXMAX approximation holds, the SNR of the combination of speech and music does not significantly affect the performance.

Another feature to notice is the sensitivity of the FHMM to the size of the set of allowed digits. In the fourth testing Table 4.3, when the number of digits and utterances is increased by two from the third test, the performance of the FHMM drops more than that of the baseline system. This suggests that the FHMM is more susceptible to confusion from an increased number of possible model combinations.

The FHMM solution to the problem of recognizing speech in music has a similar range of recognition accuracy at 0 dB SNR when compared to current highly robust speech recognizers [1]. However, while most standard methods for speech recognition in noise (e.g. spectral subtraction, Wiener filtering) assume stationary or slowly-varying background noise, the FHMM approach is robust for noise that is rapidly varying over a large dynamic range, like speech or music.

5.3 Limitations

As suggested in Section 3.4, the FHMM system was found to require long periods of time to perform recognition tasks. In fact, computational complexity was the chief reason for limiting the size of the set of allowed digits and number of utterances of each digit in the tests described in Sections 4.3 and 4.4. The following equation was empirically found to describe the relationship between test time, the number of allowed digits and the number of utterances for the task of multiple digit recognition in Matlab, on a Pentium 4, 1.6 GHz processor,

$$\text{Time (Hours)} = 0.015 \times N_{trials} \times \left(\frac{N_{dig} \times (N_{dig} - 1)}{2} \right)^2 \qquad (5.1)$$

where N_{trials} is the number of utterances of each digit and N_{dig} is the number of allowed digits

6. CONCLUSION

The primary objective of this project was to design a system to recognize isolated digits in non-stationary noise, specifically interfering speech and music. To accomplish this goal, an isolated digit recognizer was first designed and trained in Matlab [10] for spoken digits at high SNR. Motivated by Petruncio's results from HMM musical genre classification using MFSC feature vectors [9], a classical music HMM was also trained with a front-end MFSC observation sequence. The digit and noise HMMs were combined using Roweis' Factorial Hidden Markov Model [4] with outputs defined by Nadas' MIXMAX algorithm [5]. The FHMM system was implemented as an equivalent HMM by extending the results from the MIXMAX algorithm, Equation (3.6), to a mixture of Gaussians PDF, Equations (3.8), (3.9) and (3.11).

The performance of the FHMM system was evaluated on two types of non-stationary noise, interfering speech and music at 0 dB SNR. The FHMM system was shown to perform multiple digit recognition with an accuracy of up to 89%, an average relative improvement of 105% over baseline. At the task of isolated digit recognition in music, the FHMM system was found to perform with an accuracy of up to 70%, an average relative improvement of 170% over baseline.

REFERENCES

[1] H. K. Kim and R. C. Rose, "Cepstrum-Domain Acoustic Feature Compensation Based on Decomposition of Speech and Noise for ASR in Noisy Environments," *IEEE Transactions on Speech and Audio Processing*, Vol. 11, No. 5, September 2003.

[2] Z. Ghahramani and M.I. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, 29, pp. 245-275, 1997.

[3] B. Logan and P. Moreno, "Factorial HMMs for Acoustic Modeling," *ICASSP*, pp. 813-816, 1998.

[4] S. T. Roweis, "One Microphone Source Separation," *Neural Information Processing Systems* 13, pp. 793-799, 2000.

[5] A. Nadas, D. Nahamoo and M. A. Picheny, "Speech Recognition Using Noise-Adaptive Prototypes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 10, October 1989.

[6] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 4, August 1980.

[7] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, February 1989.

[8] B. H. Juang, S. E. Levinson, M. M. Sondhi, "Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains," *IEEE Transactions on Information Theory*, Vol. IT-32, No. 2, March 1986.

[9] D. Petruncio, "Evaluation of Various Features for Music Genre Classification with Hidden Markov Models," BS Thesis, University of Illinois, March 2002.

[10] K. Murphy, "Hidden Markov Model (HMM) Toolbox for Matlab," May 2003, http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html.

[11] R. G. Leonard, "A Database for Speaker-Independent Digit Recognition", *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, p 42.11, 1984.