Automatic Song Identification in Noisy Broadcast Audio

Eloi Batlle, Jaume Masip, Enric Guaus
Audiovisual Institute and Dept. of Technology
Pompeu Fabra University
Pg. Circumvalació, 8
E-08003 Barcelona. Catalunya-Spain
email: eloi,jmasip,eguaus}@iua.upf.es

ABSTRACT
Automatic identification of music titles and copyright enforcement of audio material has become a topic of great interest. One of the main problems with broadcast audio is that the received audio suffers several transformations before reaching the listener (equalizations, noise, speaker over the audio, parts of the songs are changed or removed, etc.) and, therefore, the *original* and the *broadcast* songs are very different from the signal point of view. In this paper, we present a new method to minimize the effects of audio manipulation (i.e. radio edits) and distortions due to broadcast transmissions. With this method, the identification system is able to correctly recognize small fragments of music embedded in continuous audio streams (radio broadcast as well as Internet radio) and therefore generate full play-lists. Since the main goal of this system is copyright enforcement, the system has been designed to give almost no false positives and achieve very high accuracy.

KEY WORDS
Broadcast audio identification, mel-cepstrum coefficients, hidden Markov models, Viterbi, music information retrival.

1 Introduction

Systems that are able to automatically identify file songs have received a great deal of recent attention. The main goal of these systems is to associate a singer and and a title to an audio file. That is, given an audio file (WAV, MP3, etc.) the systems analyzes its content and matches it to a database of *originals* (previous observations and analysis of these files or very similar ones).

Among all proposals, content-based identification techniques has proved to be more flexible and robust than file comparison, metadata or watermarking proposals which depend on the integrity of non audible data. Content-based identification techniques are based on the acustic qualities of audio. Different system implementation for this approach have been proposed which contain different audio feathure extraction mechanisms and database matching algorithms [1, 2]. Nevertheless, none of these system proposals explicitly face the challenges derived from broadcast audio identification.

As a matter of fact, commercial radio stations modify the songs before broadcast to increase their impact on casual listeners and, therefore, it is very common to find some parts of the song changed (or repeated or deleted). Another common situation is the broadcasting of only a few seconds of the song. From a copyright point of view, it is very important to detect these situations because copyrights should be taken into account not only for the whole song but also for small parts of it.

Another problem in broadcast environments is the fact that the system has no access to isolated songs, but to a continuous stream of unlabeled audio that contains not only songs but also news, commercials and other unknown material. And all these audio events mix together with often fuzzy transitions between them.

In the next sections, we present an audio identification system that is able to correctly identify songs in a continuous stream of unknown audio material (song spotting) and to generate a play-list finding the beginning and end points of each song.

This paper is structured as follows. It starts with an overview of the global system and how it works. Section 3 introduces the feature extraction front-end that discriminates relevant information from the whole audio signal. Then, section 4 presents the channel estimation technique used to counterbalance the effects of signal editing and broadcasting. Since the identification system is based on a stochastic approach, section 5 sketches the training algorithm for the system, while Section 6 describes the whole matching process. Finally, section 7 shows the system performance results under different identification conditions.

2 System overview

The identification system is build on a well known stochastic pattern matching technique known as Hidden Markov Models (HMM). HMMs have proven to be a very powerful tool to statistically model a process that varies in time [3]. The idea behind them is very simple. Consider a stochastic process from an unknown source and consider also that we only have access to its output in time. Then, HMMs are well suited to model this kind of events. From this point of view, HMMs can be seen as a doubly embedded stochastic process with a process that is not observable (hidden pro-

Roku EX1014
IPR2024-01053
U.S. Patent No. 8,296,791

cess) and can only be observed through another stochastic process (observable process) that produces the time set of observations.

We can see music as a sequence of audio events. The simplest way to show an example of this this is in a monophonic piece of music. Each note can be seen as an acoustic event and, therefore, from this point of view the piece is a sequence of events. However, polyphonic music is much more complicated since several events occur simultaneously. In this case we can define a set of abstract events that do not have any physical meaning but it mathematically describes the sequence of complex music. In section 5, we describe how we deal in our system with this kind of complex music. With this approach, we can build a database with the sequences of audio events of all the music we want to identity.

To identify a fragment of a piece of music in a stream of audio, the system continuously finds the probability that the events of the pieces of music stored in the database are the generators of this unknown broadcast audio. This is done by using the HMMs as a generators of observations instead of decoding the audio into a sequence of HMMs (see section 6).

3 Feature extraction

The first step in a pattern matching system is the extraction of some features from the raw audio samples. We choose the parameter extraction method depending on the nature of the audio signal as well as the application. Since the aim of our system is to identify music behaving as close as possible to a human being, it is sensible to approximate the human inner ear in the parametrization stage. Therefore, we use a filter-bank based analysis procedure. In speech recognition technology, mel-cepstrum coefficients (MFCC) are well known and their behavior leads to high performance of the systems [4]. It can be also shown that MFCC are also well suited for music analysis [5].

4 Channel estimation

Techniques for dealing with known distortion are straightforward. However, in real radio broadcast, the distortion that affects the audio signal is unknown. To remove some effects of this distortions, we can assume that they are caused by a linear time-invariant (or slowly variant) channel. With this approach we assume that all the distortion can be approximated by a linear filter $\mathcal{H}(\omega)$ that slowly changes in time. Thus, if we define $y(n)$ as the audio signal received, $x(n)$ as the original signal and $\mathcal{F}()$ as the Fourier transform, we can write

$$\mathcal{Y}(\omega) = \mathcal{F}\left[y(n)\right] = \mathcal{H}(\omega)\mathcal{X}(\omega) \tag{1}$$

and in the logarithmic space

$$\ln|\mathcal{Y}(\omega)| = \ln|\mathcal{H}(\omega)| + \ln|\mathcal{X}(\omega)| \tag{2}$$

Since we only have access to the distorted data and due to the nature of the problem we cannot know how the distortion was, we need a method to recover the *original* audio characteristics from the *distorted* signal without having access to the manipulations this audio has suffered. Here we define the channel as a combination of all possible distortions like equalizations, noise sources and DJ manipulations.

If the distorting channel $\mathcal{H}(\omega)$ is slowly varying we can design a filter that, applied to the time sequence of parameters, is able to remove the effects of the channel. The filter we designed for our system is

$$CR(z) = 0.99\frac{1 - z^{-1}}{1 - 0.98z^{-1}} \tag{3}$$

By filtering the paramenters of the distorted audio with this filter, they are converted, as close as possible, to the clean version. By removing this channel effect from the received signal the identification performance is greatly improved because all the distortions caused by any equalization and transmission are removed [6]. Therefore the system will be able to deal with not only clean CD audio but also broadcast noisy audio.

5 Training

In our approach, HMMs represent generic acoustic generators. Each HMM models one generic source of audio. For example, if the audio we model has a piano and a trumpet, we will have one HMM to model the piano and another one to model the trumpet. However, commercial pop music has a very complex variety and mixture of sounds and so it is almost impossible to assign a defined sound source to each HMM. Therefore, each HMM in the system models abstract audio generations, that is, each HMM is calculated to maximize the probability that if it was really a sound generator, it will generate that sound (complex or not). Thus, HMMs are calculated in a way that the probability that a given sequence of them will generate a particular song and, that given all possible songs, we can find a sequence of HMMs for each of them that generates them reasonably well.

To derive the formulas to calculate the paramenters of each HMM we used a modification of the Expectation--Maximization algorithm were the incomplete data (as they are defined in [7]) are not only the parameters of the HMMs but also their correct sequences for each song. If we suppose that a probability density function exists $f(\varphi|\lambda)$ that is related to the probability density function of the incomplete data then we can relate them with

$$g(\mathbf{O}|\lambda) = \int_{\Phi(\mathbf{O})} f(\varphi|\lambda)d\varphi \tag{4}$$

where $\mathbf{O}$ are the samples from the incomplete samples space and Φ are the samples of the complete samples space. We also suppose that there is at least one transformation

from the space of complete samples to the space of incomplete samples.

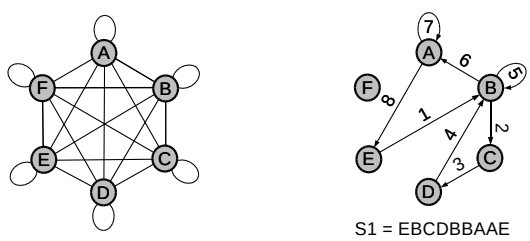
Therefore, the training stage in our system is done in a iterative way similar to the Baum-Welch algorithm [8] widely used in speech recognition system. Speech systems use HMMs to model phonemes (or phonetic derived units) but, unfortunately, in music identification systems we do not have any clear kind of units to use. That is why at each iteration a new set of units is estimated as a part of the incomplete data in order to jointly find the sequence of probabilities and also the set of abstract units that best describes complex music. After some experimental results we found that a good set of units is completely estimated after 25-30 iterations.

6 Audio Identification

HMM training described in the previous section was aimed at obtaining the maximum distance between all possible song models in order to increase speed and reliability during the audio identification phase. Once the HMMs are trained, the next steps toward building the entire system consist in getting the song models and matching them against streaming audio signals.

6.1 Signature generation

Signature generation consists in obtaining a sequence of HMMs for each song that uniquely identifies it among the others. The song signatures are generated using a Viterbi algorithm [9]. The Viterbi algorithm computes the highest probability path between HMMs on a complete HMM graph model as shown in Figure 1.a. This figure is followed by an example of Viterbi signature generation in Figure 1.b. All the song signatures are stored in a signature database.



(a) HMM model (b) Viterbi output

Figure 1. The Viterbi algorithm computes the optimum path travel on a complete HMM graph model.

The time complexity of the Viterbi algorithm that computes the signature of a song with L frames on a complete graph with Q HMMs is $O(L * Q^2)$, while the space complexity required for backtracking the optimal sequence is $O(L*Q)$. Therefore, the implementation of the signature generator is feasible as far as Q is kept under small orders of magnitude.

6.2 Identification algorithm

The identification algorithm is in charge of matching all the signatures against the input streaming audio signals to determine whenever a song section has been detected. The Viterbi algorithm is used again with the purpose of exploiting the observation capabilities of the HMM models contained in the signature sequences. Nevertheless, this time the graph model is not a complete graph but a cyclic HMM model as shown in Figure 2. This model is built linking all song HMM sequences from the identity signature database in a ring structure where each HMM only has two links, one to itself and one toward its immediate neighbor. Nevertheless, the Viterbi algorithm is allowed periodically to use internal ring links in order to allow jump between different song sections. Combining the Viterbi algorithm with the HMM ring model proposal, the identification phase can perform all the following key features:

- *Normal operation*: Identify the song signature and perform continuous time tracking between song start and song end. The optimal path corresponds to consecutive HMM matching where only external links are used.
- *Song mixing*: Identify internal jumps between songs. The optimal path corresponds to consecutive HMM matching using external links and only one internal link.
- *Song interruption*: Identify non modeled sections. The optimal path corresponds to behaviors where the optimal path can not be classified in the previous cases.

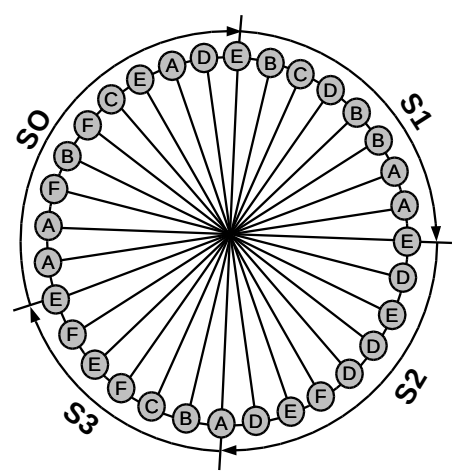


Figure 2. HMM model for a signature database with four songs: S0-S1-S2-S3.

The time complexity of the Viterbi algorithm for the ring graph is $O(L * Q)$, while it only requires a space of $O(Q)$ to process L audio frames with Q HMMs in the

ring graph. Therefore, the identification algorithm scales linearly with the number of songs in the database because each HMM only has two links and the internal link period can be large enough to have small impact on the time complexity while maintaining a reasonable song-mixing capability.

7 Experimental results

The identification algorithm and the song signature database were implemented using the C++ programming language. The innermost time critical loop was developed in assembler code in order to achieve higher optimization. The running process consumed 35Mbytes memory space and achieved real-time performance while processing one streaming audio input. The computing platform was a single Pentium-III CPU with 1GHz clock.

The system parameters used for the real-time implementation were:

- 256 HMMs to generate the song signatures.
- 450 HMMs average per song signature.
- 3852 song signatures in the database.
- 6 seconds periodicity for the internal links.

The first experiment consisted in streaming one song to the audio identification algorithm. Figure 4 shows the Viterbi output from the identified song signature. In this case, the Viterbi algorithm kept running under normal operation since no transitions were performed between songs. The continuous diagonal line corresponds to the end-to-end detection of the main sound track while the small parallel diagonals correspond to sections that were identified multiple times inside the same song.

Three additional experiments were run with the aim of studying the identification system reliability under different broadcast audio distortions. An automated test-bench was built with the aim of performing exhaustive statistical studies of the identification system over the complete song database. The schematic of the complete test-bench used in all the experiments is shown in Figure 3. The first block builds the signature database by processing all the mp3 audio files from original CD albums. An audio tool produces a continuous audio stream and the original audio labels associated with the complete mp3 file database. The audio stream contained a single mono channel coded with signed words at 22050Hz rate. The audio labels combined the song identification number and the time stamp that measured the distance from the beginning. The distortion block is optional and modifies the original audio stream trying to reproduce the main audio editions performed in real radio broadcast studios. The identification block is in charge of observing the audio stream with the HMMs, match them against the signature database and generate detected audio labels. Finally, the monitoring tool verifies the detected audio labels against the original labels and retrieves statistics about the audio identification system reliability.

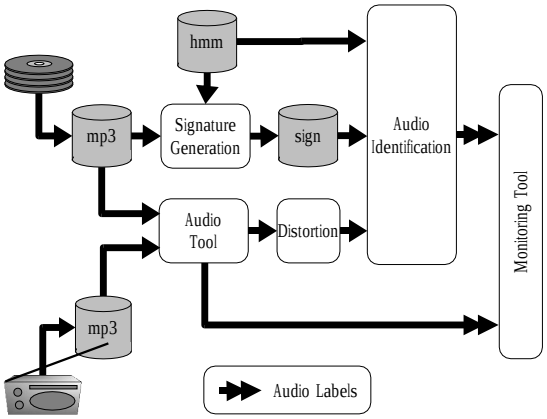


Figure 3. Test-bench schematic.

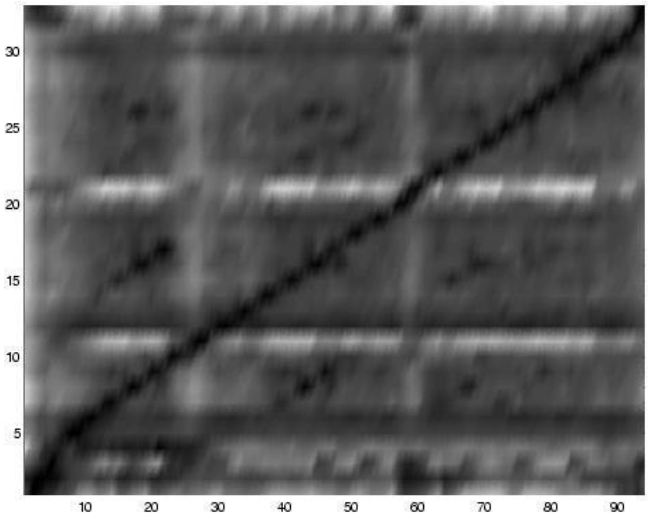


Figure 4. Planar plot of the Viterbi score achieved by the song signature (horizontal axis) over time (vertical axis).

7.1 Matching the complete audio database

This experiment aims at studying the capabilities of the audio identification system to identify original audio streams. This feature is exploited when the content of mp3 or other audio files can be analyzed directly by the application. In a broader sense, these experiments determines the raw identification capabilities of the HMM observers with the Viterbi algorithm.

The input for this experiment was a continuous audio stream generated appending the 3856 songs contained

in the complete mp3 audio database. Aproximately, the length of the complete audio stream was 250 hours (10.5 days). The distortion block was not present during this experiment. The experiment took 8 hours to complete on a 16 computer cluster with dual Pentium-III CPU at 1GHz and running paralellized versions of the audio tool and the audio identification block.

The analysis of the preliminary results determined the existence of a large number of identification labels that overlapped and generated false negative detections. As a matter of fact, three sources of false positives were found:

- *Same file*: Two copies of same audio file were found in the mp3 database when it appeared both in the original and in the compilation albums by the same artist. Duplicated copies were also detected in albums from different artists who performed together.
- *Same song*: Two different audio files contain the same song but performed in a slightly different way as may be the case of the original and live concert versions of the same song.
- *Song mix*: A single song is composed by mixing pieces of songs from different albums.

The false positives where corrected by means of label exchange using tables that contained allowed correspondences between songs. The error rate measures obtained before and after extracting false positives are shown in Figure 5. The figure presents three sections clearly differentiated in terms of error rate: the song introduction, the song middle stage and the song end. The higher error rates found at the song introductions and endings is due to a higher mismatch between the MFCC coefficients and the instrumental sections that concentrate at the song introduction and song end. Moreover, as already stated in Section 5, each HMM represents a generic acoustic generator and in average, these sections are simpler in terms of instrumental complexity or even contain significant silence periods.

7.2 Matching the complete audio database with radio distortions

It is well known that radio stations use complex sound processing techniques to get higher loudness and produce the effect of impressive sound broadcast. The use of all these sound processing techniques is not perfectly defined, and it depends on the music style and the legislation of each specific country, between other factors. The most common techniques are signal compressions, enhancements, time stretching and exciters.

The radio distortion model used in the test-bench focuses on the compression technique. Audio compression consists on dynamic range reduction, due an adaptative and variable gain of the input signal, which allows signal amplification without changing the maximum peak level, Therefore, audio compression increases the overall loudness. In

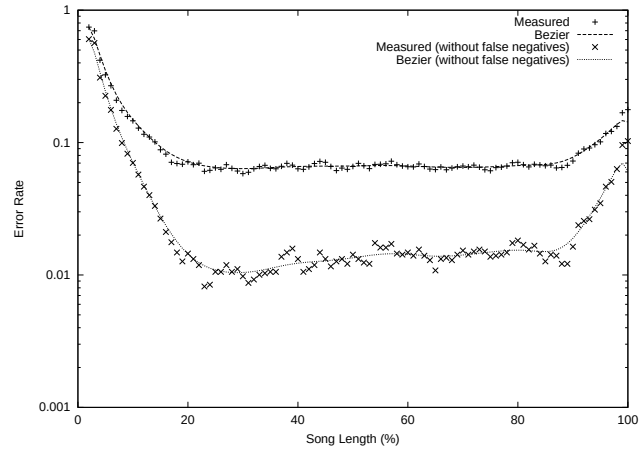
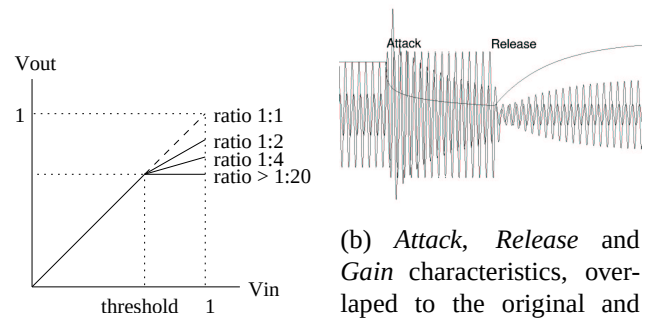


Figure 5. Probability distribution of the error rate over the song length.

fact, the distortion block here defined is a combination of a compressor and a limiter in order to achieve a fixed maximum level. There are four important parameters in the compression process: threshold, ratio, attack and release.



(a) *Threshold* and *Ratio*

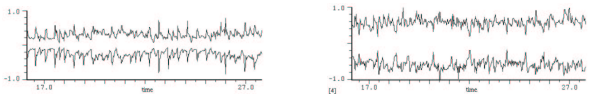
(b) *Attack*, *Release* and *Gain* characteristics, overlaped to the original and modified signals

Figure 6. Compressor parameters

The *threshold* defines the minimum value which the compressor reduces the input signal, according to the *ratio*. This is not an instantaneous process, and we must choose the *attack* and the *release* time in order to define how fast the signal is compressed when its amplitude increases, and how fast the signal leaves compression when its amplitude decreases, respectively. The threshold, ratio, attack and release values used in this experiment are 0.5, 40, 10ms and 2500ms respectively. All these values are experimentaly fixed for the worst case.

Some Radio Stations apply multi-band compression: the compression applied at different frequency bands is not the same. With this technique, the original sound gets more presence and contrast. In Fig. 7, we can see the effect of the compression techniques mentioned above, applied to an original signal from a CD.

The identification test-bench with the distortion block



(a) Original signal (b) Compressed signal

Figure 7. Compressor effects

produced the error rate measures shown in Figure 8. The figure shows the system performance with and without false positives corrected using the same tables as the first experiment. As can be seen, the distortion block introduces a significant performance penalty in terms of false negative labels while it has a minimal impact on the final error rate when comparing Figure 8 and Figure 5.

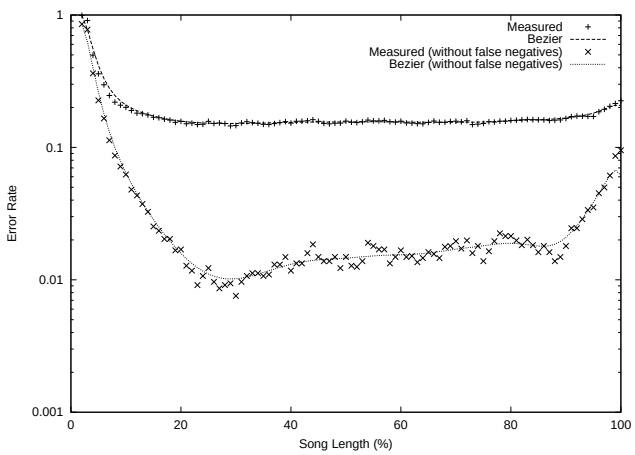


Figure 8. Probability distribution of the error rate over the song length.

7.3 Matching broadcast radio and Mp3 compression

This test used an input of 25 hours of audio captured from a single radio broadcast station which accumulated 217 songs in total interleaved with news and commercials. Table 1 shows the system performance results for the radio capture as well as different mp3 compression rates before and after extracting false negatives. Therefore, the system accuracy can be granted as far as table correspondences are maintained withing the song database.

8 Conclusions

The combination of channel estimation, trained HMM observers and Viterbi sequencing and alignment algorithms results in highly robust audio identification system performance. The system has been characterized extensively in terms of error rate response under original and radio distortion audio databases, radio captures and different mp3 compression rates. The system analysis showed that false positives were due to song copies, versions and remixes. Moreover, the system performance for different song sections has been detemined. Finally, radio distortion and mp3 compression deteriorate the algorithm output but do not impact the audio detection reliability.

Audio Source	Identification with False Positives	Identification with no False Positives
Original	100%	100%
Radio capture	100%	100%
MP3 128 kbps	100%	100%
MP3 32 kbps	99.83%	100%
MP3 24 kbps	99.04%	100%

Table 1. System performance on different environments.

References

[1] T. Kastner, E. Allamanche, J. Herre, O. Hellmuth, M. Cremer, and H. Grossmann, "MPEG-7 Scalable Robust Audio Fingerprinting," in *Proceedings of the AES Convention*, 2002.

[2] J. Haitsma, T. Kalker, and J. Oostveen, "Robust Audio Hashing for Content Identification," in *Proceedings of the Content-Based Multimedia Indexing*, 2001.

[3] L. R. Rabiner, "A Tutorial on HMM and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[4] E. Batlle, C. Nadeu, and J. A. R. Fonollosa, "Feature Decorrelation Methods in Speech Recognition.," in *International Conference on Spoken Language Processing*, Sydney, 1998, vol. 3, pp. 951–954.

[5] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *ISMIR*, 2000.

[6] R. A. Bates, "Reducing the Effects of Linear Channel Distortion on Continuous Speech Recognition," M.S. thesis, Col. of Engineering. Boston University, 1996.

[7] A. P. Dempster and et altri, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[8] L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *BAMS*, pp. 360–363, 1967.

[9] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Identification," *IEEE Trans. Info. Theory*, vol. 13, no. 2, pp. 260–269, 1967.