Thesis

Acoustic Monitoring System for Frog Population Estimation Using In-Situ Progressive Learning

Submitted by

Adam Aboudan

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2013

Master's Committee:

Advisor: Mahmood R. Azimi-Sadjadi

Kurt Fristrup
Christopher Peterson

Abstract

Acoustic Monitoring System for Frog Population Estimation Using In-Situ Progressive Learning

Frog populations are considered excellent bio-indicators and hence the ability to monitor changes in their populations can be very useful for ecological research and environmental monitoring. This thesis presents a new population estimation approach based on the recognition of individual frogs of the same species, namely the *Pseudacris Regilla (Pacific Chorus Frog)*, which does not rely on the availability of prior training data. An in-situ progressive learning algorithm is developed to determine whether an incoming call belongs to a previously detected individual frog or a newly encountered individual frog. A temporal call overlap detector is also presented as a pre-processing tool to eliminate overlapping calls. This is done to prevent the degrading of the learning process. The approach uses Mel-frequency cepstral coefficients (MFCCs) and multivariate Gaussian models to achieve individual frog recognition.

In the first part of this thesis, the MFCC as well as the related linear predictive cepstral coefficients (LPCC) acoustic feature extraction processes are reviewed. The Gaussian mixture models (GMM) are also reviewed as an extension to the classical Gaussian modeling used in the proposed approach.

In the second part of this thesis, the proposed frog population estimation system is presented and discussed in detail. The proposed system involves several different components including call segmentation, feature extraction, overlap detection, and the in-situ progressive learning process.

In the third part of the thesis, data description and system performance results are provided. The process of synthetically generating test sequences of real frog calls, which are applied to the proposed system for performance analysis, is described. Also, the results of the system performance are presented which show that the system is successful in distinguishing individual frogs, hence capable of providing reasonable estimates of the frog population. The system can readily be transitioned for the purpose of actual field studies.

Acknowledgements

I would first like to thank my adviser, Dr. Mahmood R. Azimi-Sadjadi, for his invaluable support and considerate guidance throughout the course of this research. His guidance and time is greatly appreciated throughout the course of my graduate education.

I would like to thank my committee members, Dr. Kurt Fristrup and Dr. Christopher Peterson, for their time and assistance.

I would like to thank the National Park Service (NPS), for providing the funding for this research under cooperative agreement # H2370094000.

I would like to thank my colleagues in the Signal and Image Processing Lab. They have provided a great environment for discussing work and providing help when most needed. Thanks to Nick, Neil, Soheil, Amanda and Jarrod.

Finally, I would like to thank my family for their support and guidance throughout my graduate education.

Dedication

To my parents, Khaled and Colleen.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1. Background and Motivation

Frog populations are considered excellent bio-indicators, and the ability to monitor changes in their populations is of utmost importance to ecological research and environmental monitoring [1]. Frogs are very sensitive to environmental pollutants due to their permeable skin and water-based life cycle [2]. It has been suggested [3] that a stressed environment suffering from poor water quality and an ecosystem lacking in diversity are causes of recent frog population decline. This has triggered interest in monitoring frog populations to fully understand the causes and the resulting impact, with hope to possibly reverse this decline [4].

Call surveys and call count data are the most widely used methods to assess the presence and abundance of different species of frogs [5]. However, it has been shown that there is no clear relationship between trends in frog call activity and the true population of frogs [6], any resulting population estimates are also likely to be biased [5]. The most effective method currently used to assess the population size of frogs is the capture mark re-capture (CMR) method which can provide unbiased and more precise estimates of frog populations [5]. However, this method is labor intensive, time consuming, and intrusive [5].

Little research has been conducted on estimating populations of animals using acoustic recordings though this approach is less expensive, quicker, and has virtually no impact on the animals [7]. The estimation of animal populations through acoustic recordings requires the ability to differentiate vocalizations from different individual animals. Previous work has focused, almost exclusively, on species recognition [1, 8, 9, 10, 11, 12] where different

species of animals were identified from their acoustic recordings. There are a limited number of studies that have focused on the problem of individual recognition of animals from their acoustic recordings [13, 14]. These methods, almost exclusively, rely on the availability of training data for each of the individuals to be recognized, and do not consider the problem of population estimation of the same species as a possible application. An acoustic animal population estimation system will require the ability to recognize new, unknown individuals of the same species.

1.2. Survey of Previous Work

Previous work has focused, almost exclusively, on species recognition [1, 8, 9, 10, 11, 12, 15, 16] where different species of animals are identified from their acoustic recordings. Gary et al. [1] developed a method based on a set of heuristic features derived from the signal spectrogram of frog calls such as bandwidth and length of call to discriminate amongst different frog species. The frog species is identified by a series of filters and groupings that make use of the identified heuristic features. Harma [10] uses sinusoidal modeling to identify different bird species from their songs. Each song is decomposed to a set of amplitude and frequency modulated pulses. The amplitude and frequency trajectories are then used to identify the different species. Lee et al. [8] proposed a method that uses averaged Mel-frequency cepstral coefficients (MFCCs) [17] and linear discriminant analysis (LDA) [18] to automatically identify frog and cricket species from their sounds. MFCCs are extracted from each window of isolated syllables and averaged over the duration of the syllable. The averaged MFCCs are then transformed to a lower dimensional space through LDA. The distance between vectors in the lower dimensional space is then used for species identification. Tyagi et al. [15] proposed a new technique which computes the

Spectral Ensemble Average Voice Print (SEAV) for each bird species considered. The SEAV is computed as the averaged spectrum over all windows of the duration of the bird song. The euclidean distance between the SEAV from different recordings is used to identify the bird species. Somervuo et al. [16] compared three different parametric representations, namely, sinusoidal modeling, Mel-cepstrum parameters and a vector of various descriptive features such as spectral centroid, signal bandwidth, zero cross rate and short time energy for the purpose of bird species recognition. Recognition based on the use of single syllables was done by means of nearest neighbor classification [19]. A series of syllables (song fragments) was also used to identify the species. In this case, Gaussian mixture models (GMMs) [20] and hidden Markov models (HMMs) [21] are used for classification. Fagerlund [12] studied automatic identification of bird species using two different parametric representations and support vector machine (SVM) classifiers [22]. The first parametric representation used was the mel-cepstrum parameters and the second was a set of low-level signal parameters such as spectral flux, spectral flatness, and frequency range. A decision tree with binary SVM classifiers at each node was used to identify the bird species. Huang et al. [9] automatically identifies frog calls using three features namely spectral centroid, signal bandwidth, and threshold-crossing rate. The classification is done using the k-nearest neighbor (kNN) algorithm [19], as well as, SVMs. Chen et al. [11] used a standard feature template which is extracted by analyzing the multi-stage average spectrum (MSAS) of frog calls to perform species classification. A template matching method was used to compare feature templates extracted from test and training data to recognize the unknown frog species.

There is a limited number of studies that have focused on the problem of individual recognition of animals [13, 14, 23]. Fox et al. [23] extracted MFCC features from bird calls

and used a multi-layer perceptron (MLP) neural network [24] to recognize different individual birds. Chang et al. [13] extracted similar MFCC features but used GMMs for individual recognition of birds. Zhang et al. [14] applied a similar method to the individual recognition of insects using a more sophisticated α-GMM classifier.

All the above-mentioned methods rely on the availability of training data for each of the individuals to be recognized, and do not consider the problem of population estimation of the same species as a possible application. An acoustic animal population estimation system will require the ability to recognize new, unknown individuals of the same species.

Acoustic features that can capture inter-individual variations as well as classification models that can effectively model these variations are required to be able to successfully differentiate among individuals and perform population estimation. Such features and classification models have already been developed, which has demonstrated promising results in achieving individual recognition of birds [13] and insects [14]. Among these features are the mel-frequency cepstral coefficients (MFCCs) which are typically used in many speech and speaker recognition systems [25]. Additionally, among the probabilistic-based models used for species classification are the Gaussian mixture models (GMMs) [26] which are well-suited in dealing with noise.

1.3. Proposed Method

In this thesis, a new population estimation method based on the recognition of individual frogs of the same species, namely the *Pseudacris Regilla (Pacific Chorus Frog)* which are found in the West Coast region of the United States and Canada at various elevations from sea level and upto 10,000 feet, is introduced. The proposed method is based on a progressive learning algorithm which attempts to learn to recognize individual frogs by grouping calls

produced by the individuals in an in-situ manner without prior training. This makes the application of the system in real settings possible. A realistic frog call recording typically contains several temporally overlapping calls from the same frog species. However, since the call signatures are very similar, separation of the calls in order to process each individual call may only be possible with multi-channel recording, i.e. multiple microphone systems, which were not available for this study. Therefore, we also introduced a call overlap detector which serves as a pre-processing tool to exclude temporally overlapping calls of two or more frogs to avoid performance degradation. The frog calls are first detected by means of a segmentation algorithm which exploits the spectral signature of the incoming signal [10]. MFCC feature vectors [17] are then extracted from each detected call, and the overlap detector disregards any calls with detected overlap. The remaining non-overlapping calls are subsequently applied to progressively build models to represent the individual frogs producing the calls in which multivariate Gaussian distributions are used as the probabilistic classification model. The progressive learning algorithm essentially performs a series of association tests on incoming detected calls in which each incoming call is determined as to whether it belongs to a previously detected individual or to a newly encountered individual.

Synthetically generated test sequences, using multiple individual frog calls, are used to evaluate the performance of the system. Several test sequences are generated by inserting single frog calls of known identity into a test signal which is then applied to the system as input data. Two different sets of test sequences are used to test the system's performance. The first set of test sequences contains frog calls none of which are temporally overlapping. These test sequences are used to evaluate the progressive learning ability of the system where in this case the overlap detector is bypassed. The second set of test sequences contain

several temporally overlapping frog calls. These test sequences are used to evaluate the overlap detection ability of the system.

The system performance is measured in terms of the percent correct association of incoming calls for the progressive learning and the probability of detection (P_D) and probability of false alarm (P_{FA}) of the call overlap detection. Simulations have produced promising results with around 97.9% average correct association and $P_D = 0.85$ and $P_{FA} = 0.15$ for the call overlap detection system.

1.4. Organization of the Thesis

This thesis is organized as follows. Chapter 2 reviews two acoustic feature extraction methods and a classification method. The acoustic feature extraction methods reviewed are the MFCC features and LPCC features whereas the classification method reviewed is the GMM. Population estimation using in-situ progressive learning, as well as, the overlap detection are described in detail in Chapter 3. Chapter 4 provides a description of the data set, the synthetic test sequence generation, and experimental results for applying the non-overlapping test sequences, the overlapping test sequences, and a set of extended sequences to the system. Finally, Chapter 5 provides a conclusion and suggestions for future work.

CHAPTER 2

Review of Acoustic Recognition Methods

2.1. Introduction

In this Chapter, two popular acoustic feature extraction methods, namely MFCC [17] and LPCC [27], are reviewed in detail. These methods have been very popular recently in many speaker recognition applications and, hence, have also been applied to animal species/individual recognition applications in at attempt to achieve similar success. In addition, the probabilistic classification method, namely the GMM [20], is reviewed in detail as an extension of the currently applied method of multivariate Gaussian modeling for individual frog recognition in the system proposed in this thesis.

Both the MFCC and the LPCC, which are the linear predictive coefficients (LPC) [27] represented in the cepstrum domain, features are based on computing the cepstrum of the signal which is a warped version of the spectrum that attempts imitate the human auditory perception. The main difference between these two feature extraction methods is in the initial spectrum estimation [17]. The LPCC offers a quicker alternative to the use of Fourier transform for estimating the signal spectrum [28] which could potentially provide a more computationally efficient alternative to the MFCC features currently employed.

The GMMs are used to capture the shape of an arbitrary multivariate distribution. The distribution, in this case, corresponds to the feature vectors, i.e. the MFCC features, extracted from the acoustic recording. The GMM captures the shape of the distribution by finding the Gaussian parameters or several Gaussian component densities as well as their mixing weights. Typically, expectation maximization (EM) [29] algorithm is used to find the unknown parameters. In the current application, only one Gaussian component is used

and so the ability to capture the shape of the distribution is limited. Therefore, the GMM is explored as a possible future extension to the currently applied method to improve the overall modeling capability.

In this chapter, the MFCC and LPCC feature extraction methods are reviewed in detail. Furthermore, the GMM probabilistic classification method is also reviewed.

2.2. Acoustic Features

2.2.1. Mel-Frequency Cepstral Coefficients (MFCC). The Mel-Frequency Cepstral Coefficients (MFCC) are the most popular acoustic features used recently for human speaker recognition applications [17]. Due to their success in speaker recognition applications, they have also been adopted for animal species and individual recognition applications [8, 12, 13, 14, 16, 23]. Several approaches exist for computing MFCC features. The approach detailed in this section is based on cepstral computation of a Mel-scale warped spectral estimate [30]. A diagram illustrating the MFCC feature extraction process in shown in Fig. 2.1.

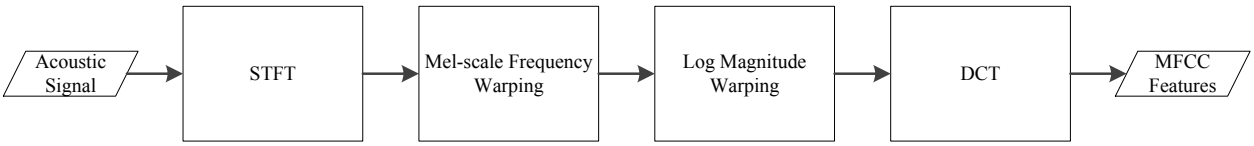


Figure 2.1. Diagram showing MFCC feature extraction process.

2.2.1.1. *Short-Time Fourier Transform (STFT).* The Short Time Fourier Transform (STFT) is first used here to provide spectral content of the signal in localized time windowed regions. The input signal of interest is divided into overlapping windows of length N where each window is shifted from the previous one by Δ. A windowing function is used here for time localization and to reduce discontinuities in the windowed signal. Several types

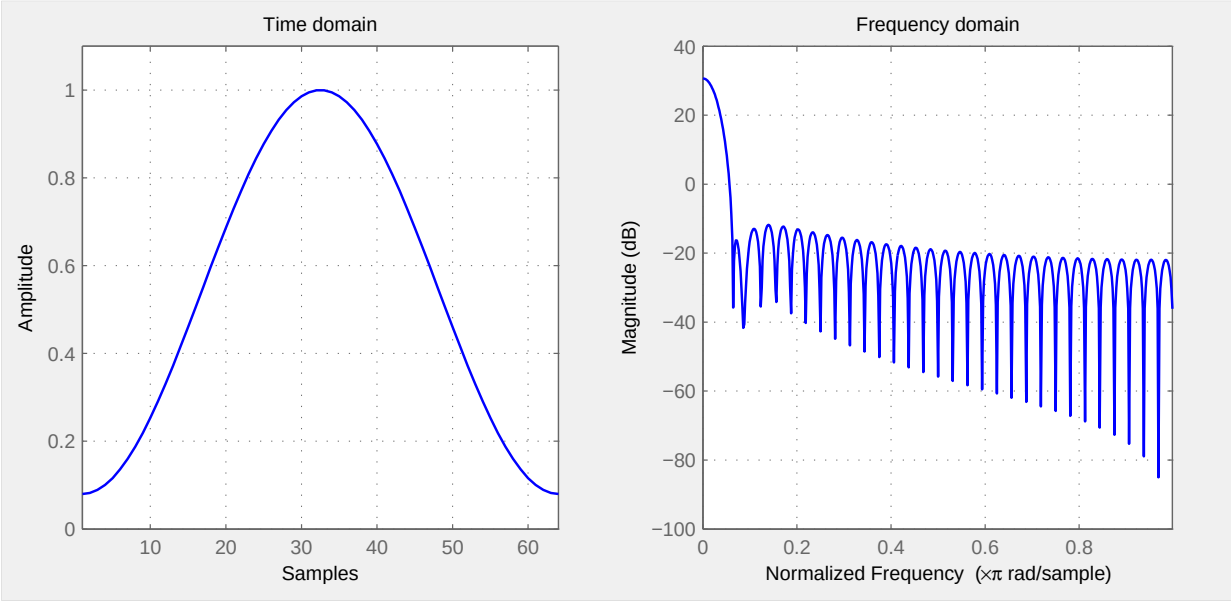


FIGURE 2.2. Hamming window function of length $N = 64$ samples in both time and frequency domains.

of windows exist such as the Hamming, Hanning, Welch, Triangular, Gauss, Blackman and Bartlett [17]. Each type of window has a different shape and localization (time-frequency) characteristic. However, the most popular window used in speech processing is the Hamming window, shown in Fig. 2.2, given by:

$$w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right) \tag{1}$$

where N is the length of the window.

Let $s(n)$ denote the signal and $s_m(n)$ denote the m^{th} window of the signal given by:

$$s_m(n) = s(n)w(n - m\Delta) \tag{2}$$

where Δ is the shift between each window and m is the window index.

The spectrum of the m^{th} window of the signal is given by:

$$S_m(k) = \sum_{n=m\Delta-\frac{N}{2}}^{m\Delta+\frac{N}{2}-1} s_m(n)e^{-2jk\pi n/N},\ \forall k \in [1, K] \tag{3}$$

where k is the frequency index.

This process assumes stationarity of the signal over the duration of the window size. Although the computed spectra $S_m(k)$ are complex-valued, it is common for most speech processing systems to consider only the magnitude spectra $|S_m(k)|$.

2.2.1.2. *Frequency and Magnitude Warping .* The Mel (abbreviation of melody) is a unit of pitch which is defined to be equal to one thousandth of the pitch ($\wp$) of a simple tone of frequency $1000Hz$ with an amplitude of $40dB$ above the auditory threshold [17]. This definition of pitch in the Mel-scale was motivated by the fact that the human auditory perception is approximately linear upto the frequency of $1000Hz$ and then becomes more logarithmic for higher frequencies. Another motivation was the experiments conducted by Zwicker [31] where he modeled the human auditory perception system using a 24-band filter-bank of critical bands whose center frequencies are positioned according to the so-called Bark scale where they are non-linearly spaced in the frequency domain. The relationship between frequency and pitch in the Mel-scale is given by [32, 33]:

$$\wp = \frac{1000}{ln(1+\frac{1000}{700})} ln(1+\frac{f}{700}) \tag{4}$$

where f is the frequency in Hz and $\wp$ is the pitch in Mels.

Because of the way in which the human auditory perception works, the magnitude coefficients of the spectrum $|S_m(k)|$ computed in Section 2.2.1.1 are modified to have less

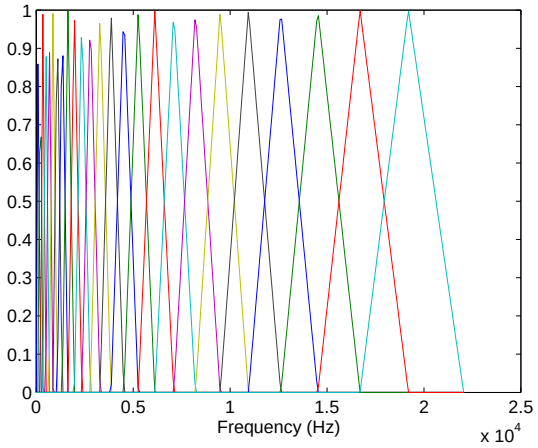


FIGURE 2.3. Band-pass filter bank in frequency domain.

coefficients that are related to the critical band center frequencies using the Mel-scale. This can be achieved by building a filter bank of triangular pass-band filters, as shown in Fig. 2.3, to convert the magnitude spectral coefficients in $|S_m(k)|$ by computing a weighted sum of spectral coefficients to obtain the Mel-frequency magnitude spectrum coefficients denoted by $|\breve{S}_m(l)|$, which is the l^{th} coefficient of the m^{th} window of the signal and is given by:

$$|\breve{S}_m(l)| = \sum_{k=1}^{K} W_l(k)|S_m(k)|, \ \forall \ l \in [1, L] \tag{5}$$

where $W_l(k)$ is the l^{th} band-pass filter weights and L is the number of filters in the filter bank.

Up till now, the magnitude spectral coefficients of the window of the signal have been warped to the Mel-frequency domain. Next, these magnitude coefficients need to be warped so that they may be logarithmic and resemble human auditory perception. This can be achieved by simply taking the logarithm of $|\breve{S}_m(l)|$ [30],

$$Y_m^l = ln(|\breve{S}_m(l)|) \tag{6}$$

2.2.1.3. *Mel-frequency Cepstral Coefficients (MFCC).* We now have a representation of the m^{th} window of the signal in Y_m^l that closely resembles how the human auditory system perceives the sound. The next step is to compress the information provided in this representation and then take only the most useful part of the compressed representation. Several ways to achieve the sought compression exist but the most prevalent method is to compute the discrete cosine transform (DCT) of the coefficients in Y_m^l [30]. The MFCC features for the m^{th} window of the signal, y_m^d are computed as the DCT of the derived logarithm of the magnitude spectrum of the windowed signal in the Mel-scale Y_m^l,

$$y_m^d = \sqrt{2/L} \sum_{l=1}^{L} Y_m^l cos\left(\frac{\pi}{L}(l-0.5)d\right), \ \forall \ d \in [1, D] \tag{7}$$

where $\sqrt{2/L}$ is a normalization factor and D is the number of MFCC features computed.

Although the number of MFCC features can be set to equal the number of filters (i.e. $D = L$), usually only the first few MFCC coefficients are taken (i.e. $D < L$) since they capture most of the useful information.

Remark: In order to capture the dynamics of the MFCC features, first and second order derivatives of the MFCC features known as $Delta$ and $Delta-Delta$ MFCC, respectively, are often used [14]. These features are, in general, independent of the original MFCC features, i.e. contain new information, and can be used to extract information on the local dynamics of the sound. However, adding too many features will increase dimension of the extracted acoustic feature vector which will increase the amount of data required to estimate the statistical parameters of a model which is not well-suited for the system proposed in this thesis. Furthermore, the standard MFCC features alone inherently capture some of the local

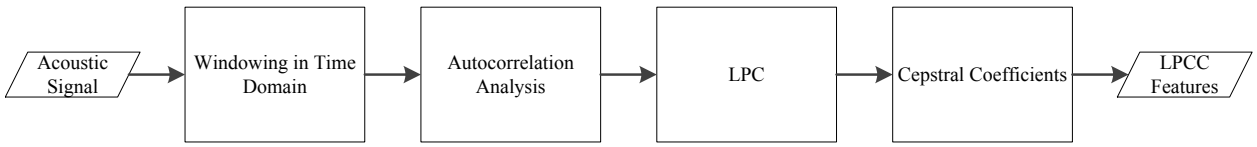


FIGURE 2.4. Diagram showing LPCC feature extraction process.

dynamics due to the windowing process [17]. Therefore, for the proposed system, only the standard MFCC features are considered.

2.2.2. LINEAR PREDICTIVE CEPSTRAL COEFFICIENTS (LPCC). The LPCC features will be reviewed here as an alternative to the currently used MFCC features which could increase the proposed system's overall computational efficiency by eliminating the initial Fourier transform computation required to compute the MFCC features [28]. The LPCC features are the cepstral version of the linear predictive coding (LPC) [17] which is a parametric, spectral, source-filter modeling scheme [34]. LPC coefficients are the autoregressive (AR) model [35] coefficients that minimize the error between the predicted values and the actual values of a given window of data [36]. A diagram illustrating the LPCC feature extraction process in shown in Fig. 2.4.

The first step of time domain framing and windowing to achieve short time analysis of the signal is the same as that in the MFCC feature extraction explained in Section 2.2.1.1. We will begin our analysis here with the m^{th} window of the signal $s_m(n)$ defined in (2). The method we will use for computing the LPC will be based upon an $AR(P)$ process [17]. Performing error minimization of the P^{th} order AR model results in the Yule-Walker [35] equations:

$$\sum_{p=1}^{P} a_p r_m(|i-p|) = r_m(i),\ 1 \leq i \leq P \tag{8}$$

where $r_m(i)$ is the autocorrelation of the signal in the m^{th} window and is given by:

$$r_m(i) = \sum_{n=m\Delta-\frac{N}{2}}^{m\Delta+\frac{N}{2}-1-i} s_m(n)s_m(n-i) \tag{9}$$

Let us define the autocorrelation matrix $\mathbf{R}_m$ as:

$$\mathbf{R}_m = \begin{bmatrix} r_m(0) & r_m(1) & r_m(2) & \cdots & r_m(P-1) \\ r_m(1) & r_m(0) & r_m(1) & \cdots & r_m(P-2) \\ r_m(2) & r_m(1) & r_m(0) & \cdots & r_m(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_m(P-1) & r_m(P-2) & r_m(P-3) & \cdots & r_m(0) \end{bmatrix} \tag{10}$$

which is a non-singular Toeplitz matrix and the autocorrelation vector $\mathbf{r}_m$ as:

$$\mathbf{r}_m = \begin{bmatrix} r_m(1) \\ r_m(2) \\ \vdots \\ r_m(P) \end{bmatrix} \tag{11}$$

Then, we can rewrite (8) in matrix form as:

$$\mathbf{R}_m\mathbf{a}_m = \mathbf{r}_m \tag{12}$$

where $\mathbf{a}_m = [a_1, a_2, ..., a_P]^\top$ is the AR coefficients vector for the m^{th} window.

The Toeplitz structure of the matrix $\mathbf{R}_m$ makes it easy to solve for $\mathbf{a}_m$ using some efficient algorithms that do not rely on matrix inversion. Alternatively, one can use the *Levinson-Durbin* recursive algorithm [35] to solve for $\mathbf{a}_m$.

Once the LPC coefficients $\mathbf{a}_m$ are computed they must be converted into cepstral coefficients. This conversion can be achieved by using a recursive algorithm [37] which will result in the linear predictive cepstral coefficient (LPCC) features. This recursive algorithm allows for the computation of the cepstral coefficients without needing to compute the Fourier transform of the signal. Let us denote the LPCC feature vector as $\boldsymbol{\alpha}_m = [\alpha_0, ..., \alpha_d, ..., \alpha_D]^\top$. The LPCC are computed as follows:

(1) For $d = 0$,

$$\alpha_0 = r_m(0) \tag{13}$$

(2) For $1 \leq d \leq P$,

$$\alpha_d = a_d + \sum_{i=1}^{d-1} \frac{i}{d} \alpha_i a_{d-i} \tag{14}$$

(3) For $d > P$,

$$\alpha_d = \sum_{i=1}^{d-1} \frac{i}{d} \alpha_i a_{d-i} \tag{15}$$

These LPCC features are not based on the perceptual Mel-scale like the MFCC features. However, a warping can be applied to make them based on such a scale. Also, it is usually the case that the dimension of the LPCC feature vector D is greater than the dimension of the linear predictor coefficients P [34].

2.3. Classification Methods

2.3.1. Gaussian Mixture Models (GMM). Gaussian mixture models (GMMs) are used to model measurements or feature data (e.g., MFCC or LPCC features), by representing the probability density function of the data as a weighted sum of multivariate Gaussian densities with unknown parameters [20]. This allows GMMs to smoothly capture the shape of an arbitrary density representing the data. The characteristics of the data captured by

the GMM can be used for classification purposes where a GMM can be trained to recognize a certain phenomenon by capturing the shape of the distribution of the data produced by such a phenomenon. An $\mathcal{M}$ component GMM is given by:

$$p(\mathbf{y}|\lambda) = \sum_{i=1}^{\mathcal{M}} w_i p(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i),\ i = 1, 2, ..., \mathcal{M} \tag{16}$$

where $\mathbf{y}$ is a D-dimensional observation or feature vector, $p(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i)$ are the Gaussian component densities with mean $\boldsymbol{\mu}_i$ and covariance Σ_i and w_i are the unknown mixture weights. The mixture weights w_i must satisfy the constraint:

$$\sum_{i=1}^{\mathcal{M}} w_i = 1 \tag{17}$$

Each component density is a D-variate Gaussian function defined as:

$$p(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i) = \frac{1}{((2\pi)^{D/2}|\Sigma_i|^{1/2})} e^{\frac{-1}{2}(\mathbf{y}-\boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{y}-\boldsymbol{\mu}_i)} \tag{18}$$

where $|\cdot|$ stands for determinant of the matrix inside.

The GMM is completely parametrized by the mean vectors $\boldsymbol{\mu}_i$, covariance matrices Σ_i and mixture weights w_i from each component density. The combined GMM parameters, λ are denoted as $\lambda = \{p_i, \boldsymbol{\mu}_i, \Sigma_i\},\ i = 1, 2, ..., \mathcal{M}$.

The structure of the GMM in terms of the number of components, $\mathcal{M}$, to use, type of covariance to impose (i.e. full vs diagonal) and how parameters are tied usually depends on the nature of the application. For example, for the proposed system in this thesis, the nature of the problem imposes a limit on the amount of data available since the system is designed to work in-situ with possibly limited data. Therefore, for this system it would be

more reasonable to use a smaller number of component densities and parameters. Otherwise, there may not be a sufficient amount of data to estimate all the parameters of the model.

Let $\Gamma = [\mathbf{y}_1, ..., \mathbf{y}_v, ..., \mathbf{y}_V]$ be the matrix of training observation/feature vectors where V is the total number of vectors available. To model this data using a GMM is to estimate the parameters of the density components of the GMM for a certain structure. This is essentially an attempt to choose the GMM parameters that will capture the shape of the training data distribution. There are several methods for estimating the parameters of a GMM [29, 38, 39, 40]. The most popular of which is the maximum likelihood (ML) estimation which estimates the model parameters from the training data using an iterative expectation-maximization (EM) algorithm [29].

2.3.1.1. *Maximum Likelihood (ML) Parameter Estimation .* In ML estimation, GMM parameters are determined by maximizing the likelihood of the GMM given the training data. The likelihood of the GMM given the data in Γ is given by:

$$\ell(\Gamma|\lambda) = p(\Gamma|\lambda) \tag{19}$$

where if we assume that the observations in Γ are conditionally independent, which may not be a necessarily correct assumption but a necessary one nonetheless to make this problem solvable, we can write the likelihood as:

$$\ell(\Gamma|\lambda) = \prod_{v=1}^{V} p(\mathbf{y}_v|\lambda) \tag{20}$$

Alternatively, working with the log-likelihood function and using the definition of the GMM in (16), (20) becomes:

$$\begin{aligned} L(\Gamma|\lambda) =& ln(\prod_{v=1}^{V} p(\mathbf{y}_v|\lambda)) \\ =& \sum_{v=1}^{V} ln(p(\mathbf{y}_v|\lambda)) \\ =& \sum_{v=1}^{V} ln(\sum_{i=1}^{\mathcal{M}} w_i p(\mathbf{y}_v|\boldsymbol{\mu}_i, \Sigma_i)) \end{aligned} \tag{21}$$

It is not possible to maximize log-likelihood function in (21) directly, however, it can be maximized using an incomplete data approach used in the EM algorithm [29]. The general idea here is to use an initial model parameter set, which we will denote as λ^{old}, to estimate a new parameter set, which we will denote as λ^{new}, such that $\ell(\Gamma|\lambda^{new}) \geq \ell(\Gamma|\lambda^{old})$, i.e. a higher likelihood is achieved with the new parameter set. Once the new parameter set is determined, it is used to find the next set and so on until a convergence criteria is met [20]. A two step process is used to estimate the new model parameters, λ^{new}, that will guarantee a monotonic increase in the likelihood of the model.

In the *expectation* step, the maximization function $Q(\lambda)$ is formulated based on the a-posteriori probability of the i^{th} component density, $p(i|\mathbf{y}_v)$ given by:

$$\begin{aligned} p(i|\mathbf{y}_v) =& \frac{p(i, \mathbf{y}_v)}{p(\mathbf{y}_v)} \\ =& \frac{p_i p(\mathbf{y}_v|\boldsymbol{\mu}_i, \Sigma_i)}{\sum_{i'=1}^{\mathcal{M}} p_{i'} p(\mathbf{y}_v|\boldsymbol{\mu}_{i'}, \Sigma_{i'})} \end{aligned} \tag{22}$$

and maximization function, $Q(\lambda)$, which is the expected value of the log-likelihood of the joint event of of the data, Γ, and the model parameters, λ, is then given by [21]:

$$Q(\lambda) = \sum_{v=1}^{V} \sum_{i=1}^{\mathcal{M}} p(i|\mathbf{y}_v) ln(p(\mathbf{y}_v|i) w_i) \tag{23}$$

In the *maximization* step, the function $Q(\lambda)$ is maximized using the following parameter update equations [21]:

$$p_i^{new} = \frac{1}{V} \sum_{v=1}^{V} p(i|\mathbf{y}_v) \tag{24}$$

$$\boldsymbol{\mu}_i^{new} = \frac{\sum_{v=1}^{V} p(i|\mathbf{y}_v)\mathbf{y}_v}{\sum_{v=1}^{V} p(i|\mathbf{y}_v)} \tag{25}$$

$$\Sigma_i^{new} = \frac{\sum_{v=1}^{V} p(i|\mathbf{y}_v)(\mathbf{y}_v - \boldsymbol{\mu}_i^{new})(\mathbf{y}_v - \boldsymbol{\mu}_i^{new})^\top}{\sum_{v=1}^{V} p(i|\mathbf{y}_v)} \tag{26}$$

where the new model parameter set λ^{new} is given by:

$$\lambda^{new} = \{p_i^{new}, \boldsymbol{\mu}_i^{new}, \Sigma_i^{new}\} \tag{27}$$

2.4. Conclusion

This Chapter reviewed two popular acoustic feature extraction methods, namely the MFCC and LPCC, as well as the GMM probabilistic-based classification method. The reviewed feature extraction methods are two of the most frequently used algorithms in speaker recognition applications and recently have been applied to the problem of animal species and individual recognition [13, 14]. These methods attempt to imitate the human auditory perception by warping the acoustic signal both in frequency and magnitude. The LPCC features were shown [28] to be more computationally efficient by avoiding a Fourier transform

computation and, hence, can be used in the proposed system as an alternative to MFCC features to increase the overall performance.

The GMM was shown to offer better capability than the classical unimodal multivariate Gaussian modeling, used in the proposed system, in terms of capturing the unique characteristics of a certain frog call by being better equipped to capture the distribution of the feature vectors. However, in order to be able to accurately estimate the many parameters of a GMM, a much larger number of training samples must be used. This limitation prevents the use of GMM in the conventional sense, however, it could be possible to incrementally move from the simple multivariate Gaussian modeling, currently used in the proposed system, to more sophisticated GMM by increasing the number of Gaussian components used in the GMM only after a certain amount of data has become available through the learning process. This can be accomplished, for example, by establishing data thresholds where if the amount of data accumulated for a certain class/model exceeds the established threshold, then a new model with a more sophisticated structure can be trained to replace the existing model.

CHAPTER 3

Population Estimation Using In-Situ Progressive Learning

3.1. Introduction

The system presented in this chapter is designed to process an acoustic recording which contains frog calls produced by several different individual frogs. The calls are detected and used by the in-situ learning process to count how many individuals are present and to learn to recognize these individuals. Occasional temporal overlaps between different calls are also taken into consideration by designing an overlap detection process to detect and eliminate such overlapping calls in order to avoid performance degradation which would have resulted by including these calls.

In this chapter, the proposed system is presented and it's different components are discussed in detail. First, an overview is given of the proposed overall system structure. Next, the call segmentation and feature extraction stages of the system are detailed. Then, the proposed in-situ progressive learning system is discussed in detail. Finally, the details of the overlap detection process are discussed.

3.2. Proposed System Structure

The proposed system structure is depicted in Fig. 3.1. In this system, the frog calls are initially detected by means of a segmentation algorithm, which exploits the spectral signature of the incoming signal [10]. MFCC feature vectors are then extracted from each detected call and the overlap detector then identifies and disregards overlapping calls. The overlap detector relies on previously generated models to detect association ambiguities,

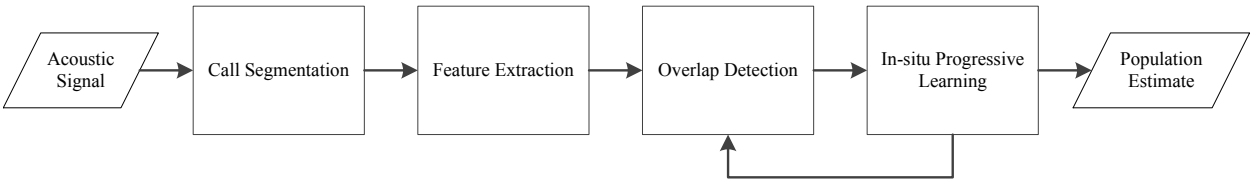


FIGURE 3.1. Proposed system architecture.

which are caused by overlapping calls. The non-overlapping calls are subsequently applied to the progressively learning models to represent the individual frogs producing the calls and to perform in-situ learning of the corresponding parameters. Multivariate unimodal Gaussian distributions are used as the probabilistic-based classification model. The progressive learning algorithm essentially performs a series of association tests on incoming detected calls to determine whether they belong to the previously detected individuals or to newly encountered individuals. The subsystems and processes in Fig. 3.1 are described in detail in the following Sections.

3.3. Call Segmentation and Feature Extraction

3.3.1. Spectral-Based Call Segmentation. The segmentation stage is responsible for detecting and isolating calls in the recorded acoustic time series to essentially provide the start and stop times for each detected call. In this stage, the acoustic recording is partitioned into a set of segments which are then grouped together to form complete frog calls. For the frog species considered here, each frog call consists of two parts separated by a short silent interval. Fig. 2(a) illustrates a typical call from a *Pseudacris Regilla* frog. The detected segments are grouped together based on the interval separating them, i.e. if two segments are separated by less than a pre-specified interval, then the two segments are combined and together make up a complete call. The silent interval was not considered since it does not

contain acoustic characteristics that are specific to the frog individual considered and hence would cause more ambiguity between the calls of different individual frogs.

The segmentation process [10] is based on the signal energy in certain frequency subbands associated with frog calls. Thus, it can reject many transient sources and interference, e.g. high frequency insect sounds and low frequency clutter. Transients or interference sources which exist in the same spectral bands as with the frog calls cannot be rejected by this segmentation process. However, other features could also be exploited to reject such unlikely transients or interference sources.

The segmentation process begins by normalizing the amplitude of the recorded signal $s(n)$ and computing the Short-Time Fourier Transform (STFT), $S_m(k)$, where m is the window index and k is the frequency index. Next, the peak magnitude for each window is computed as:

$$\mathcal{P}[m] = \max_k (20log_{10}|S_m(k)|),\ k_{min} < k < k_{max} \tag{28}$$

where k_{min} and k_{max} are frequency indices that correspond to $f_{min} = 300Hz$ and $f_{max} = 4000Hz$, respectively, which contain most of the spectral energy of the frog calls of interest.

In order to perform call detection and isolation, a moving average filter is applied to sequence $\mathcal{P}[m]$ to generate $\tilde{\mathcal{P}}[m]$ as,

$$\tilde{\mathcal{P}}[m] = \frac{1}{\Delta_1} \sum_{n=m-\frac{\Delta_1-1}{2}}^{m+\frac{\Delta_1-1}{2}} \mathcal{P}[n] \tag{29}$$

where Δ_1 is the span of the filter. This filtering is done to smooth out the peak magnitude function for easier call detection and isolation. The call segments are then detected and isolated using the following procedure:

(1) Set $i = 1$.

(2) Find window index m^i of the i^{th} segment's peak by computing $m^i = \arg\max_m(\tilde{\mathcal{P}}[m])$.

(3) Check that the detected segment is valid, i.e. $\tilde{\mathcal{P}}[m^i] > \beta$, where β a validity threshold experimentally found to be 25. If not valid, terminate.

(4) Find window index m_s^i of the i^{th} segment's start by tracing $\tilde{\mathcal{P}}$ until $\tilde{\mathcal{P}}[m_s^i] < \tilde{\mathcal{P}}[m^i] - \phi$ for $m_s^i < m^i$, where ϕ is a roll-off threshold experimentally found to be 6.

(5) Find window index m_e^i of the i^{th} segment's end by tracing $\tilde{\mathcal{P}}$ until $\tilde{\mathcal{P}}[m_e^i] < \tilde{\mathcal{P}}[m^i] - \phi$ for $m_e^i > m^i$.

(6) Delete region of detected segment in $\tilde{\mathcal{P}}$, i.e. set $\tilde{\mathcal{P}}[m] = 0$ for $m_s^i < m < m_e^i$ to allow for the detection of the next valid segment.

(7) Repeat steps $1 - 5$ until all valid segments are detected and isolated.

Once all segments have been detected and isolated, the segments which are close enough to each other are combined to form a complete call. This process is specific to the frog species of interest since the calls in this case usually consist of two parts. The observation vectors from two adjacent segments are combined if the number of windows separating them is less than a certain threshold, i.e. if $m_s^{i+1} - m_e^i < \theta$ where θ is experimentally found to be 16. The windows included in each complete call are collected to form a data matrix Ψ_j, where j is the detected call index., i.e. $\Psi_j = [\mathbf{S}_1, ..., \mathbf{S}_v, ..., \mathbf{S}_{V_j}]$ where $\mathbf{S}_v = [S_v(1), ..., S_v(k), ..., S_v(K)]^\top$ is the discrete Fourier transforms (DFT) of the v^{th} included window of the signal, K is the

number of DFT points computed, v is the observation index, V_j is the number of observations collected for the j^{th} call, and operator $\top$ denotes matrix transpose.

The segmentation process is depicted in Fig. 3.2 for a particular frog call. Fig. 2(a) shows the acoustic time series of a typical *Pseudacris Regilla* frog call. Fig. 2(b) shows the STFT of the time series with the frequency bands of interest indicated. Here the window was Hamming window of size 440 with 95% overlap. Figs. 2(c) and 2(d) show the segmentation process using the magnitude peak function $\tilde{\mathcal{P}}$. Figs. 2(e) and 2(f) show segment combination and the formation of the data matrix Ψ_j.

3.3.2. Feature Extraction. To distinguish between different frogs of the same species, features must be capable of capturing the potentially subtle, inter-individual differences. The MFCC features [17] described in Chapter 2 are applied to each column of the data matrix $\Psi_j = [\mathbf{S}_1, ..., \mathbf{S}_v, ..., \mathbf{S}_{V_j}]$ computed in the call segmentation stage. Each detected and isolated call, j, will therefore result in V_j extracted feature vectors.

The observations of the data matrix Ψ_j, which are the DFT of the windows of the original time series, are warped from the frequency domain to the mel-frequency domain [32, 33]. This warping is achieved by applying a set of triangular band-pass filters to the amplitude spectrum [17]. The filters are half overlapping with center frequencies spaced equally apart in the mel-frequency domain but not in the frequency domain. The final step is to take the discrete cosine transform (DCT) of the logarithm of the magnitude of the warped spectrum. The result is the observation matrix $\Gamma_j = [\mathbf{y}_1, ..., \mathbf{y}_v, ...\mathbf{y}_{V_j}]$ where $\mathbf{y}_v = [y_v^2, ..., y_v^d, ..., y_v^D]^\top$ is the vector of $(D-1)$ MFCC features computed for the v^{th} observation of the data matrix Ψ_j, d is the MFCC index, $D-1$ is the number of MFCC features computed. The first MFCC, y_v^1, is ignored since it represents the average value of the spectrum which does not provide

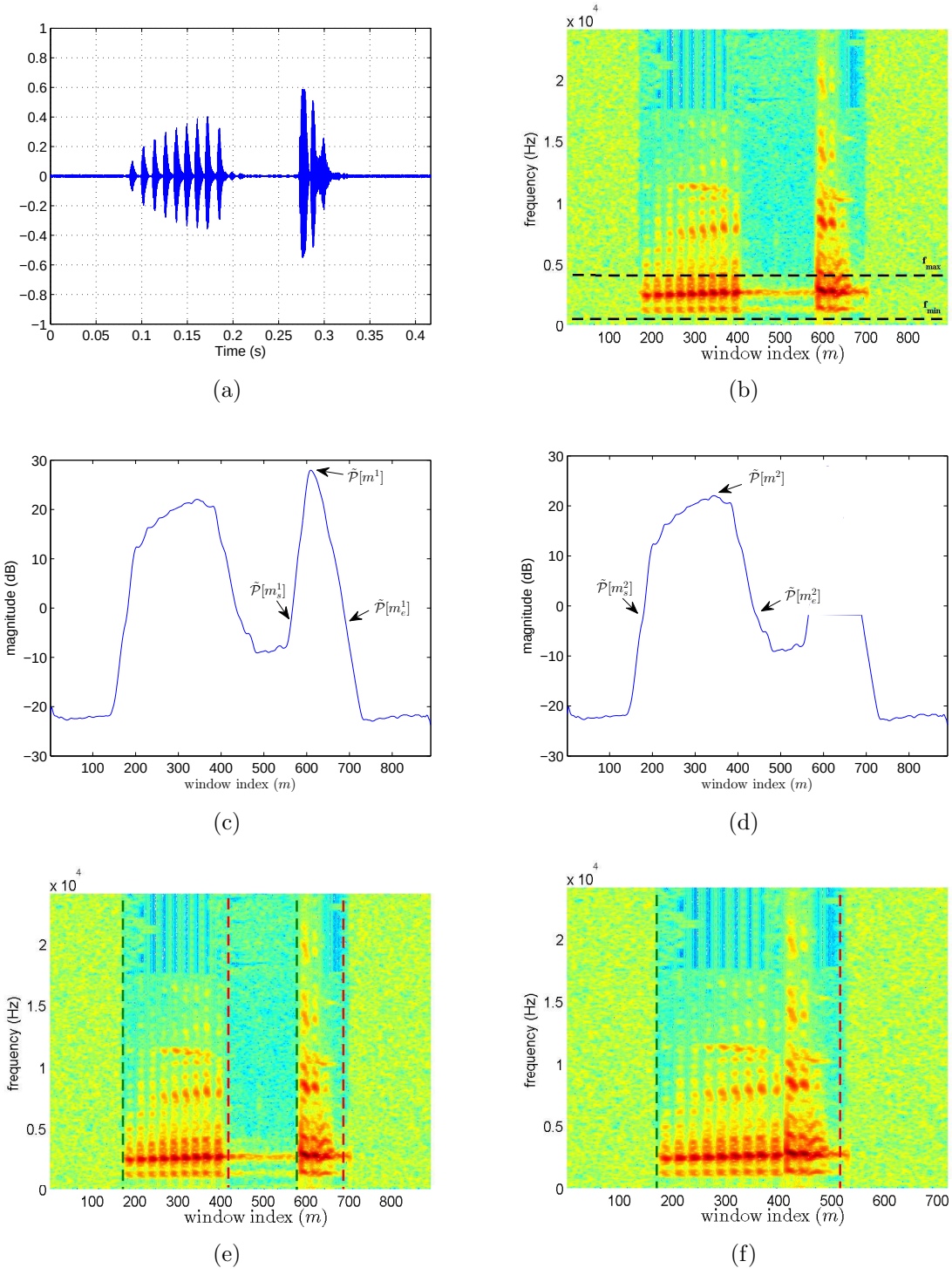


FIGURE 3.2. (a) Frog call acoustic time series (b) STFT of (a) (c) corresponding peak magnitude function showing first segment detection and isolation (d) second segment detection and isolation (e) corresponding start and end locations for segments in signal spectrum and (f) data matrix of the detected and isolated frog call.

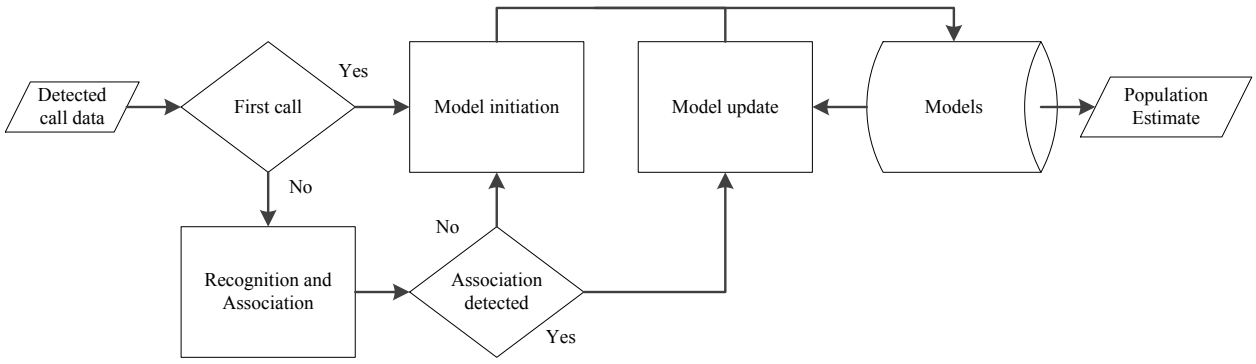


FIGURE 3.3. In-situ progressive learning diagram.

any useful information in this case [41]. The reader is referred to Section 2.2.1 of Chapter 2 for more detailed treatment of MFCC feature extraction method.

3.4. In-situ Progressive Learning

The in-situ progressive learning algorithm, as shown in Fig. 3.3, follows a special learning procedure which allows the system to recognize the individual frogs as it is exposed to their calls sequentially in time. The algorithm operates as follows:

For the first detected call a new model is initiated using the data in the computed observation matrix, Γ_1. For subsequent detected calls the data, i.e. Γ_j for $j > 1$, is tested against all available models to determine any possible association. If an association with a previously initiated model is declared, the current data is combined with data previously used to update the corresponding model parameters. If, however, no association is found, the current data is used to initiate a new model. This process continues until all detected calls are processed and associated. The population can then be estimated as the number of distinct models that have been initiated. Detailed explanations of the model generation, recognition and association in the in-situ progressive learning process are provided next.

3.4.1. Model Generation. A multivariate unimodal Gaussian is used to model the inter-individual characteristics of the detected and isolated calls i.e.

$$p(\mathbf{y}|\lambda_q) = \frac{1}{((2\pi)^{\frac{D-1}{2}}|\Sigma_q|^{1/2})} e^{\frac{-1}{2}(\mathbf{y}-\boldsymbol{\mu}_q)^\top \Sigma_q^{-1}(\mathbf{y}-\boldsymbol{\mu}_q)} \tag{30}$$

where $\mathbf{y}$ is an $(D-1)$-dimensional observation vector, $\boldsymbol{\mu}_q$ is the mean vector, Σ_q is the covariance matrix and q is the model index. The multivariate Gaussian is completely parametrized by the mean vector $\boldsymbol{\mu}_q$ and covariance matrix Σ_q or parameter set $\lambda_q = \{\boldsymbol{\mu}_q, \Sigma_q\}$ for model q.

The multivariate unimodal Gaussian modeling used here is similar to GMM, described in Section 2.3.1, for the case where there is only one component density, i.e. $\mathcal{M} = 1$. The nature of the learning process proposed here imposes a restriction on the number of components that can be used in the mixture. The amount of data available to generate new models is not sufficient enough to estimate the parameters of a more sophisticated model with more than one mixture component.

When a new model is to be initiated, a model data matrix M_q is initialized as $M_q = \Gamma_j$ and then used to estimate model q parameters. When an association is declared, the data of the call being processed is combined with the data of the associated model subsequently used to update the model parameters. This is done by augmenting the model data matrix with the new data as $M_q^{new} = [M_q^{old}\ \Gamma_j]$, where q is the associated model index being updated and Γ_j is the observation matrix of the associated call, and computing the new sample mean and sample covariance matrix using:

$$\hat{\boldsymbol{\mu}_q} = \frac{1}{V_q}\sum_{v=1}^{V_q}\mathbf{y}_v \tag{31}$$

and

$$\hat{\Sigma}_q = \frac{1}{V_q} \sum_{v=1}^{V_q} (\mathbf{y}_v - \hat{\boldsymbol{\mu}}_q)(\mathbf{y}_v - \hat{\boldsymbol{\mu}}_q)^\top \tag{32}$$

which correspond to the maximum likelihood (ML) [42] estimates of these parameters where V_q is the number of observations in the q^{th} model data matrix M_q.

3.4.2. Recognition and Association. In the recognition and association phase, an incoming detected call is recognized and associated with either a new or previously initiated model. If no association is found (i.e. the first call of a new individual is detected), then a new model is initiated. Otherwise, if an association is found, the data from the newly detected call is used to update the associated model as described in the previous section. For the first call detected by the system, this stage is bypassed and a new model is generated directly since no models are available to test for association at that time.

Two methods can be used to determine association, the first method is based on a likelihood score [43] whereas the second method is based on the Kullback-Liebler (KL) divergence measure [44]. Both methods attempt to measure how well the new observations fit the previously generated models. The likelihood-based method does make an assumption on the conditional independence of the observations which is not shared by the KL-divergence method. Furthermore, the KL-divergence method is faster in terms of processing time. Both methods are detailed next.

3.4.2.1. *Likelihood Test.* The likelihood test uses the maximum a posteriori (MAP) method [45] to determine whether or not the incoming call data is associated with any of the already generated models represented by the set of parameters $\{\lambda_1, ..., \lambda_q, ..., \lambda_Q\}$, where Q is the number of models already generated.

As before, let $\Gamma_j = [\mathbf{y}_1, ..., \mathbf{y}_v, ...\mathbf{y}_{V_j}]$ be the observation matrix extracted from the j^{th} detected call to be tested. The model that satisfies the MAP condition [45] for the observations in Γ_j is used to determine association, i.e.

$$q_{MAP} = \underset{q}{\text{argmax}}(p(\lambda_q|\Gamma_j)) \tag{33}$$

where q_{MAP} is the model index corresponding to the MAP model. If we assume equal priors for our observations then using the Bayes rule (33) becomes

$$q_{MAP} = q_{ML} = \underset{q}{\text{argmax}}(p(\Gamma_j|\lambda_q)) \tag{34}$$

which corresponds to the ML estimate of the associated model.

Now, assuming that the observations are conditionally independent we can write

$$p(\Gamma_j|\lambda_q) = \prod_{v=1}^{V_j} p(\mathbf{y}_v|\lambda_q) \tag{35}$$

Alternatively, working with the log-likelihood function and using (35), (34) becomes

$$q_{ML} = \underset{q}{\text{argmax}}(\sum_{v=1}^{V_j} log(p(\mathbf{y}_v|\lambda_q))) \tag{36}$$

Association of the j^{th} detected call with model q_{ML} is then determined by the following rule:

$$\sum_{v=1}^{V_j} log(p(\mathbf{y}_v|\lambda_{q_{ML}})) \underset{not\ associated}{\overset{associated}{\gtrless}} \zeta \tag{37}$$

where ζ is a threshold which is experimentally determined. If an association is found, then the associated model is updated as explained in Section 3.4.1.

3.4.2.2. *Kullback-Liebler Divergence Test.* The KL-divergence test determines association by essentially measuring the distance between the estimated density of the incoming data and the densities associated with the generated models. This is done to determine which density best matches the new data and then to determine whether an association can be made. The KL-divergence, also known as the relative cross-entropy [44], between the two densities $p(\mathbf{y}|\lambda_F)$ and $p(\mathbf{y}|\lambda_G)$ in general is computed as:

$$KL(p(\mathbf{y}|\lambda_F), p(\mathbf{y}|\lambda_G)) = \int p(\mathbf{y}|\lambda_F) log\left(\frac{p(\mathbf{y}|\lambda_F)}{p(\mathbf{y}|\lambda_G)}\right) d\mathbf{y} \tag{38}$$

For the case when $p(\mathbf{y}|\lambda_F)$ and $p(\mathbf{y}|\lambda_G)$ correspond to multivariate Gaussian densities such that $p(\mathbf{y}|\lambda_F) \sim N(\boldsymbol{\mu}_F, \Sigma_F)$ and $p(\mathbf{y}|\lambda_G) \sim N(\boldsymbol{\mu}_G, \Sigma_G)$, the KL-divergence can be written as [46]:

$$\begin{aligned} KL(p(\mathbf{y}|\lambda_F),p(\mathbf{y}|\lambda_G)) = \frac{1}{2}(log\left(\frac{|\Sigma_G|}{|\Sigma_F|}\right) + tr(\Sigma_G^{-1}\Sigma_F) \\ + (\boldsymbol{\mu}_G - \boldsymbol{\mu}_F)^\top \Sigma_G^{-1}(\boldsymbol{\mu}_G - \boldsymbol{\mu}_F) - (D-1)) \end{aligned} \tag{39}$$

where in (39) $|\cdot|$ represents the determinant of the matrix inside.

The KL divergence, however, is not a symmetric measure (i.e. $KL(p(\mathbf{y}|\lambda_F), p(\mathbf{y}|\lambda_G)) \neq KL(p(\mathbf{y}|\lambda_G), p(\mathbf{y}|\lambda_F))$). Therefore, it cannot strictly be used as a distance metric. To achieve the sought symmetry, a $KL2$ distance metric [47] which is defined as

$$\begin{aligned} KL2(p(\mathbf{y}|\lambda_F), p(\mathbf{y}|\lambda_G)) =& KL(p(\mathbf{y}|\lambda_F), p(\mathbf{y}|\lambda_G)) \\ &+ KL(p(\mathbf{y}|\lambda_G), p(\mathbf{y}|\lambda_F)) \end{aligned} \tag{40}$$

is used instead.

Again, let $\Gamma_j = [\mathbf{y}_1, ..., \mathbf{y}_v, ...\mathbf{y}_{V_j}]$ be the observation matrix extracted from the j^{th} detected call to be tested. The observations in Γ_j are used to estimate parameters of a test model, using (31) and (32). This test model, which represents the newly detected call j, is compared against all previously generated models using the metric defined in (40). Let $\tilde{\lambda} = \{\tilde{\boldsymbol{\mu}}, \tilde{\Sigma}\}$ denote the estimated parameters of the test model. We then determine:

$$q_{KL2} = \underset{q}{\operatorname{argmin}}(KL2(p(\mathbf{y}|\lambda_q), p(\mathbf{y}|\tilde{\lambda}))) \tag{41}$$

where q_{KL2} is the model index corresponding to the model that is closest in $KL2$ distance to the test model, $p(\mathbf{y}|\tilde{\lambda})$. Association of the j^{th} detected call with model q_{KL2} is then determined by the following decision rule:

$$KL2(p(\mathbf{y}|\lambda_{q_{KL2}}), p(\mathbf{y}|\tilde{\lambda})) \underset{associated}{\overset{not\ associated}{\gtrless}} \xi \tag{42}$$

where ξ is a threshold which is experimentally determined. Similar to the likelihood test, if an association is found, then the associated model is updated as explained in Section 3.4.1.

3.5. Overlap Detection

In this section, a test is developed to identify temporally overlapping calls of the same frog species and exclude these corrupted calls from the in-situ learning process. This is done to avoid any possible degradation in performance that would otherwise result. The effectiveness of the method is based on the assumption that at least one (single non-overlapping) call belonging to each of the individuals that produced the overlapping calls is already observed by the system before the overlapping call in encountered. Due to this assumption, the overlap

detector can only be used after the system has learned the individual frogs involved in the overlapping calls.

The overlap test is based on sequentially tracking the cumulative sum of log-likelihoods [48] for each available model sequentially, for each observation, in order to gain insight on the composition of the detected call. If any ambiguity in call association is detected, then the call is identified as a potentially overlapping call and hence is discarded.

From Section 3.4.2.1, assuming equal priors and conditional independence of observations, the cumulative log-likelihood $L_q(\ell)$ for model q computed for a sequence of consecutive observations $\mathbf{y}_1, ..., \mathbf{y}_\ell$ is

$$L_q(\ell) = \sum_{v=1}^{\ell} log(p(\mathbf{y}_v|\lambda_q)) \tag{43}$$

An increase in the value of the log-likelihood $L_q(\ell)$, i.e. a positive trend, for a specific model q indicates that the corresponding observations are likely associated with the indicated model. Therefore, the trend information in the log-likelihoods can be used to determine different model associations within the detected call. To this end, the trend (or gradient) of $L_q(\ell)$ denoted by $\nabla L_q(\tilde{\ell})$, where the new index $\tilde{\ell}$ is used to avoid confusion, is computed as:

$$\begin{aligned} \nabla L_q(\tilde{\ell}) =& L_q(\tilde{\ell}\Delta_t + 1) \\ & - L_q(\tilde{\ell}\Delta_t + 1 - \Delta_t),\ \forall \tilde{\ell} \in [1, \mathcal{L}] \end{aligned} \tag{44}$$

where $\mathcal{L} = V_q/\Delta_t$ and Δ_t is the trend interval.

A moving average filter is then applied to $\nabla L_q(\tilde{\ell})$ and the new trend function is denoted as $\bar{\nabla} L_q(\tilde{\ell})$ i.e.

$$\bar{\nabla} L_q(\tilde{\ell}) = \frac{1}{\Delta_2} \sum_{l=\tilde{\ell}-\frac{\Delta_2-1}{2}}^{\tilde{\ell}+\frac{\Delta_2-1}{2}} \nabla L_q(l) \tag{45}$$

where Δ_2 is the span of the filter. This filtering is done to smooth out the resulting trend lines.

For each model q of the available models, a domination ratio $\rho(q)$, which is the ratio of the duration for which a certain model achieves the highest likelihood trend to the to duration of the entire call, is computed as:

$$\rho(q) = \frac{1}{\mathcal{L}} \sum_{\tilde{\ell}=1}^{\mathcal{L}} \mathbf{I}_q(\tilde{\ell}) \tag{46}$$

where

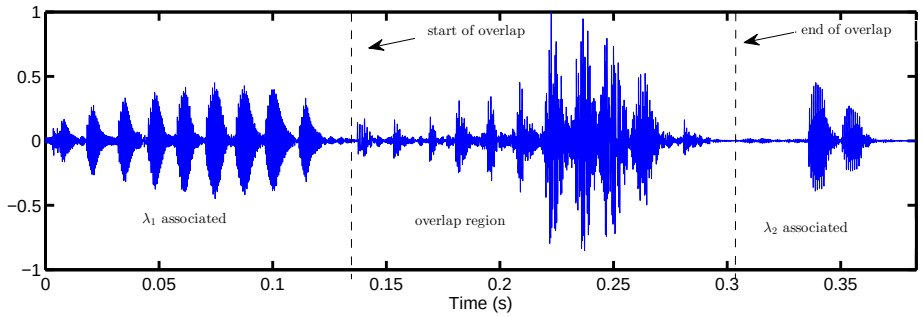
$$\mathbf{I}_q(\tilde{\ell}) = \begin{cases} 1, & \underset{q}{\text{argmax}}(\bar{\nabla} L_q(\tilde{\ell})) = q \\ 0, & otherwise \end{cases} \tag{47}$$

is an indicator function. An overlap is then declared when the domination ratio $\rho(q)$ exceeds a given threshold η for two or more models, which implies that there are regions of the detected call that are more likely associated with atleast two different models, i.e. two or more different individuals have contributed in this detected call.

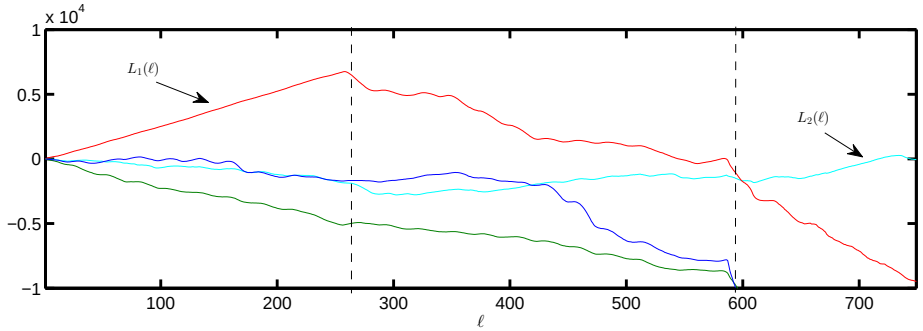
An example illustrating the steps in this process is presented in Fig. 3.4. Fig. 4(a) shows the acoustic time series of two overlapping frog calls with the region of overlap indicated. Fig. 4(b) shows the cumulative log-likelihoods $L_q(\ell)$ for all initiated models. Fig. 4(c) shows the averaged trends $\bar{\nabla} L_q(\tilde{\ell})$ for all initiated models where it is clear that there are two dominant models in different portions of the observed overlapping call. Finally, Fig. 4(d) shows a graph of the domination ratio $\rho(q)$ which is used to make a decision on whether or not an overlap is present. Since the top two dominant models exceed the given threshold in this case, the detected call is declared as potentially overlapping call and is hence discarded.

3.6. Conclusion

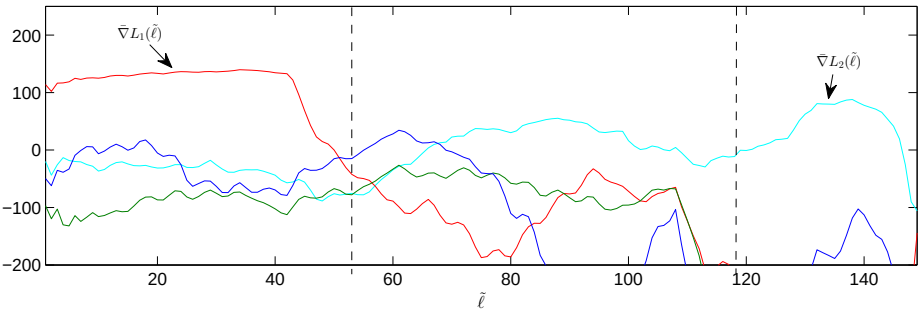
In this chapter the components of the proposed population estimation system were described in detail. The input acoustic signal is first segmented using a spectral-based segmentation algorithm which is responsible for detecting and isolating all frog calls in the signal. Once isolated, the calls are further processed to extract the associated MFCC feature vectors which are then used to form a data matrix of observation vectors, Γ_j, for each detected call j. The observation matrices are then applied to the in-situ progressive learning system where individual frogs progressively become known to the system and are identified by the models which are parametrized by their 1^{st} and 2^{nd} order statistics. These parameters are updated when associations are declared by re-estimating the parameters while taking into account the newly acquired data from the associated call. After the first calls are applied to the system, the overlap detector becomes active and every newly detected call will be subject to the overlap detector to check for temporal overlaps which will disqualify the call from contributing to the system's in-situ learning process. In the next Chapter, the proposed system and it's components will be tested on synthetically generated test sequences which contain frog calls from different individual frogs some of which are temporally overlapping with each other.



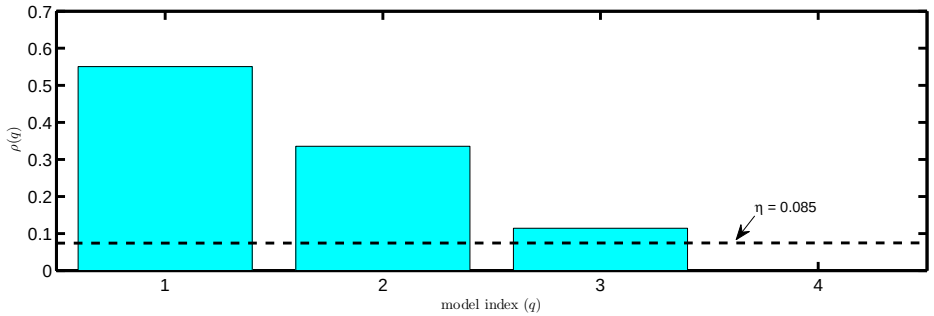
(a)



(b)



(c)



(d)

FIGURE 3.4. (a) Time series of two overlapped calls (b) plot of the cumulative log-likelihood (c) plot of averaged trend of likelihood (d) graph of most dominant models.

CHAPTER 4

Data, Experiments and Results

4.1. Introduction

In order to test the effectiveness of the proposed population estimation system, recordings of frog calls from the *Cornell Lab of Ornithology, Macaulay Library*, were used. However, these recordings are generally of either one individual frog calling or of a chorus of overlapping calls from multiple individual frogs. The proposed system is designed to be applied to a series of non-overlapping calls from several individual frogs. Since no such recordings with labeled calls could be found, they were instead synthetically generated using the available individual frog recordings. The proposed system includes an overlap detector as described in Section 3.5 of Chapter 3. This detector is designed in an attempt to make the system robust to occasional overlaps after the system is exposed to atleast one non-overlapping call of each individual frog. This capability is tested by generating additional synthetic recordings with controlled overlaps inserted.

In this chapter, the data used in the experiments is described. The synthetic sequence generation process is explained. Also, the performance measures used to asses the system are detailed. Finally, the results of three experiments are presented and discussed.

4.2. Data Description

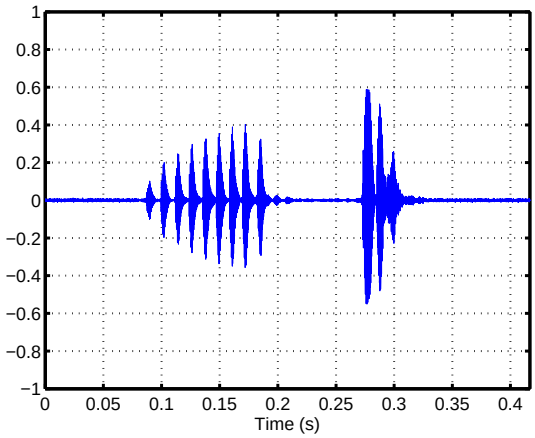
The work in this paper is developed specifically for the *Pseudacris regilla* frog species, also known as the *Hyla regilla*, but can be extended to other frog species or other animals. Male advertisement calls of the *Pseudacris Regilla* are of three distinct types [49]. The most common type (Fig. 1(a)) consists of a two-part burst of sound repeated after a short interval.

The sound is made up of a series of trills, which is a rapid alternation of two tones, where there are 5 to 11 trills in the first part and 2 to 5 trills in the second part. The second type (Fig. 1(b)) consists of a one-part burst of sound consisting of 4 to 7 trills. The third type (Fig. 1(c)) consists of a long series of short duration notes. This type is not very common and was not present in our data set. The dominant frequencies of all three call types are in the range 0.3-4kHz. In general, only male frogs produce advertisement calls [50] and so the population estimates of our system will only reflect the estimated number of male frogs in the recorded environment. Note that estimating the populations of male frogs is the usual practice and these estimates can give a good indication of the overall frog population [5].

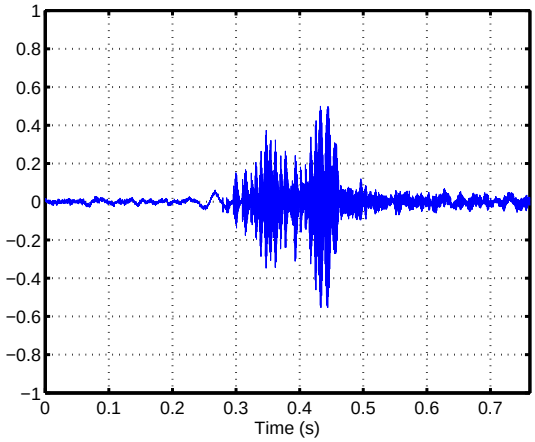
4.3. Synthetic Test Sequences

For the purpose of evaluating the performance of the proposed system, synthesized test signals that contain several calls from 11 individual *Pseudacris Regilla* were constructed. The calls were extracted from recordings of each individual frog from the *Cornell Lab of Ornithology, Macaulay Library*. The calls belonging to each individual frog were manually identified and extracted from the recordings to form a call database containing between 14 and 35 calls per individual.

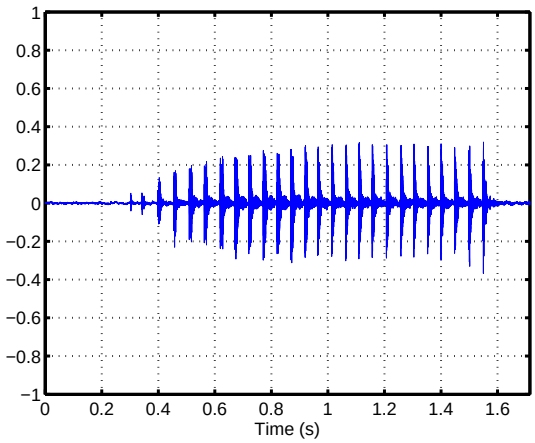
From the call database, ten non-overlapping test sequences of frog calls are synthetically constructed by randomly choosing 14 calls from each of the 11 individual frogs and then randomly ordering them in a sequence. A portion of a synthetically generated non-overlapping test sequence is shown in Fig. 4.2. Each test sequence contains a total of 154 calls which belong to 11 individual frogs none of which are temporally overlapped. These test sequences are used to evaluate the performance of the in-situ progressive learning discussed in Section 3.4 of Chapter 3. Ten additional overlapping test sequences are synthetically generated to



(a)



(b)



(c)

FIGURE 4.1. *Pseudacris regilla* frog species calls (a) type 1 (b) type 2 (c) type 3.

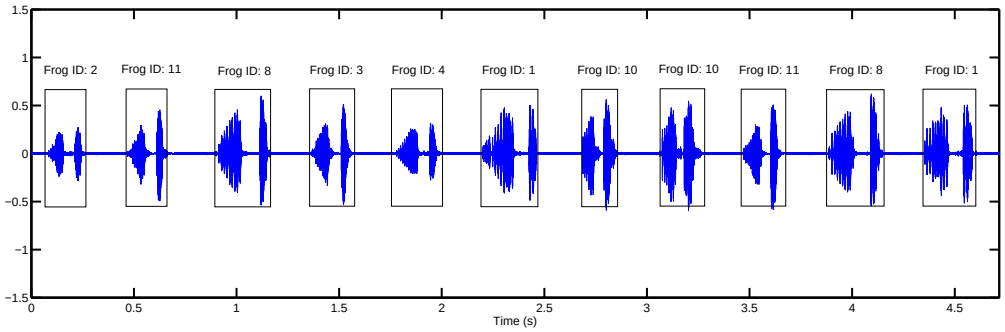


FIGURE 4.2. Portion of a non-overlapping test sequence with labeled calls.

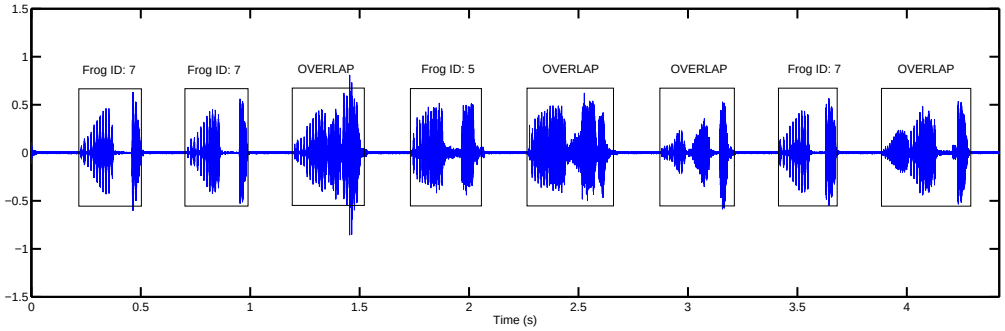


FIGURE 4.3. Portion of a overlapping test sequence with labeled calls and overlaps.

evaluate the performance of the overlap detection method introduced in Chapter 3. These sequences are generated similar to the non-overlapping sequences but include a total of 50 overlapping calls which are inserted subject to the assumption presented in Section 3.5 of Chapter 3. In addition to the mentioned assumption, only two simultaneously overlapping calls are used at a time with upto 50% overlap. A portion of a synthetically generated overlapping test sequence is shown in Fig. 4.3.

4.3.1. TEST SEQUENCE GENERATION. The sequence generation process is divided into two parts. The first part of the process randomly selects 14 calls from each of the 11 individual frogs available. After selecting which calls will be used to construct the test sequence, the non-overlapping sequence is constructed by randomly arranging the selected

calls and arranging them in a sequence where each call is separated by a short buffer of length $200ms$.

For the overlapping case, the overlapping sequence is constructed by again randomly ordering the selected calls and arranging them such that two consecutive calls will be overlapping with a certain probability. A total of 50 overlapping calls are inserted in each overlapping test sequence. This test sequence generation procedure results in a mixture of overlapping and non-overlapping calls in the overlapping test sequences. Furthermore, the insertion of the overlapping calls is governed by the rule that at least one non-overlapping call from each individual contributing to the overlapping call must have been inserted previously in the test sequence. This is done to adhere to the overlap detection assumption discussed in Section 3.5 of Chapter 3.

4.4. Performance Evaluation

The test sequences discussed above are applied to evaluate the association, as well as, the overlap detection performance of the proposed system. For the association, the system processes each non-overlapping test sequence separately and makes a series of association decisions on the detected calls, in each sequence, which ultimately leads to an estimate of the number of individuals that are present in each test sequence, i.e. the population estimate. In order to evaluate the system in this case, it is necessary to define what is meant by an association error in this context. An association error occurs when either of the following scenarios occurs:

(1) An incoming call is not associated with any model, yet at least one call from the individual that produced this incoming call was processed before.

(2) An incoming call is associated with a certain model and one of the following conditions are true:

- The individual that initiated this model and that produced the incoming call are not the same.
- The individual that produced the majority of the calls associated with the model and that produced the incoming call are not the same.

When no association error occurs, the association is considered correct.

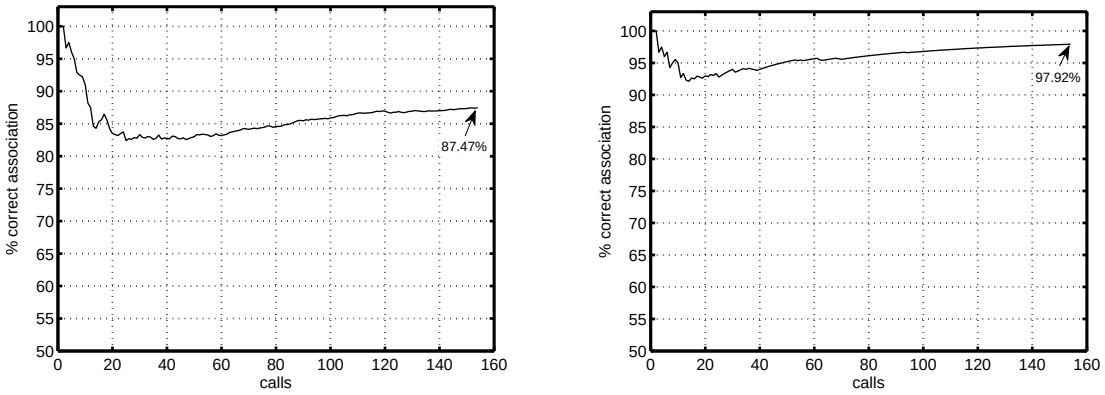
4.4.1. Experiment 1: Non-Overlapping Test Sequences. The first experiment conducted here applied the non-overlapping test sequences to evaluate the association performance of the system. The overlap detection was not used in this case. The parameters chosen in this experiment are shown in Table 4.1. The performance is reported in the form of plots of the percentage of correct association as a function of the number of calls processed for the two methods of Likelihood based and KL-based test methods discussed in Section 3.4.2 of Chapter 3. Fig. 4(a) shows the average correct association using the Likelihood test method for all ten non-overlapping test sequences. This plot shows that after processing all 154 calls in each of these test sequences, on average 87.5% of the detected calls are correctly associated. Fig. 4(b) shows the average correct association using the KL-divergence method over the same set of test sequences. This plot shows that on average 97.9% of the 154 calls are correctly associated. As can be seen from both plots, initially, after the first call is associated, the system exhibits more association errors as expected. Once the learning progresses, the system begins to make lesser and lesser association errors and the overall percentage of correct associations begins to recover. The trend is clearly increasing even towards the end of the learning. However, this recovery is restricted by the number of calls

TABLE 4.1. Simulation Parameters.

parameter	description	value
F_s	sampling frequency of input recording	$44.1kHz$
w	Hamming window	$N = 440$
Δ	window shift	418 (95% overlap)
K	# DFT points computed	512
Δ_1	span of $\mathcal{P}$ filter	51
$D-1$	# of MFCC features	16
ζ	Likelihood test threshold	-6700
ξ	KL-divergence test threshold	37
Δ_t	trend interval	5
Δ_2	span of ∇L_q filter	25

available in the test sequences. Furthermore, comparison of the percent correct association for these methods clearly shows the advantages of the KL-divergence test. This indicates that the KL-divergence test is perhaps a better measure of the similarity between two calls of the same individual frog in this application. This could be attributed to the fact that the KL-divergence method does not assume conditional independence of observations nor does it assume equally likely priors as with the Likelihood method discussed in Section 3.4.2 of Chapter 3. The correct association plots for each non-overlapping test sequence is shown in Figures 4.5 and 4.6 using the Likelihood method and in Figures 4.7 and 4.8 using the KL-divergence method, respectively. These plots show that the KL-divergence method is less sensitive to the order in which the calls are presented since the performance using this method is more consistent over the different test sequences, as apposed to, the Likelihood method which shows large variation in performance over the different test sequences.

Table 4.2 gives the resulting population estimates for each non-overlapping test sequence for the Likelihood and KL-divergence methods. It should be noted that the final population estimate is equal to the number of distinct models that have been initiated and have at least one detected call that has been associated with it, i.e. models that have been initiated but

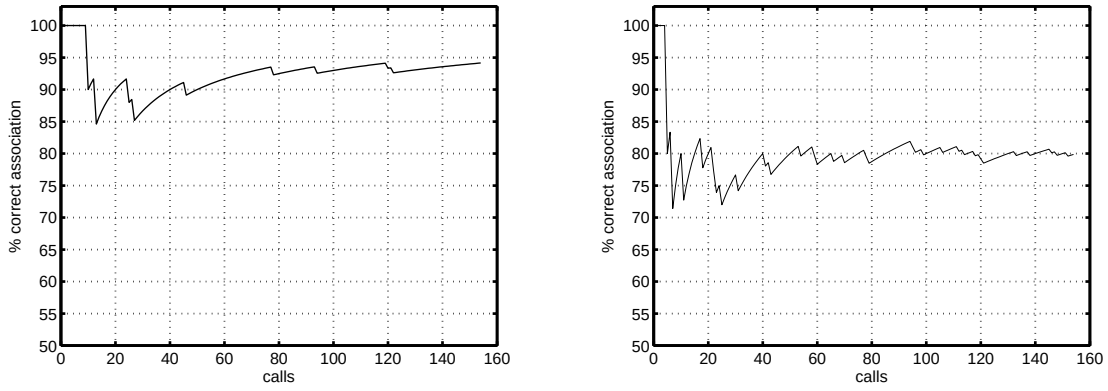


(a) (b)

FIGURE 4.4. The average percent correct association using (a) Likelihood test and (b) KL-divergence test over all ten non-overlap test sequences run.

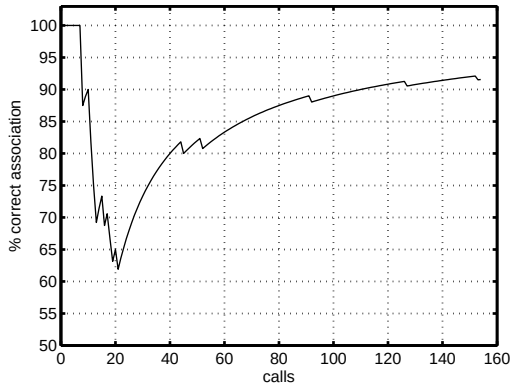
have had no detected calls associated with them are not counted in the population estimate. The table shows that both methods slightly over-estimate the actual population that was 11 individual frogs. However, the over-estimation is less severe for the KL-divergence method as shown by the smaller mean population estimate in Table 4.2 for this method . Additionally, the population estimates generated using the KL-divergence method are found to be more consistent across the different test sequences. Given its superior performance, the KL-divergence method was used as the association method for all remaining simulations.

The association performance is alternatively demonstrated in Tables 4.3 - 4.6 where the true IDs of the associated calls for each model created are shown for each non-overlapping test sequence for the Likelihood and KL-divergence methods. Models that were initiated but never associated with any new calls, i.e. have only one associated call, were disregarded here. From the Tables it is again apparent that the KL-divergence method is better at grouping calls from the same individual frog and is less prone to confusing two individual frogs as one such as in model # 1 of Table 4.3 (b) where frogs 8,9, and 10 are all considered one frog.

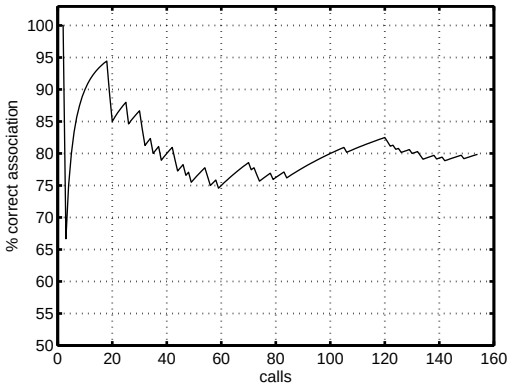


(a)

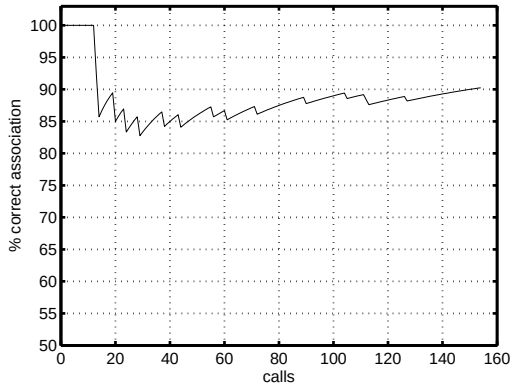
(b)



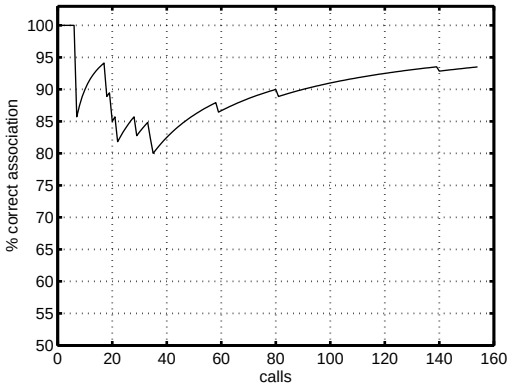
(c)



(d)



(e)



(f)

FIGURE 4.5. The percent correct association using Likelihood test for non-overlap test sequences 1 (a) - 6 (f).

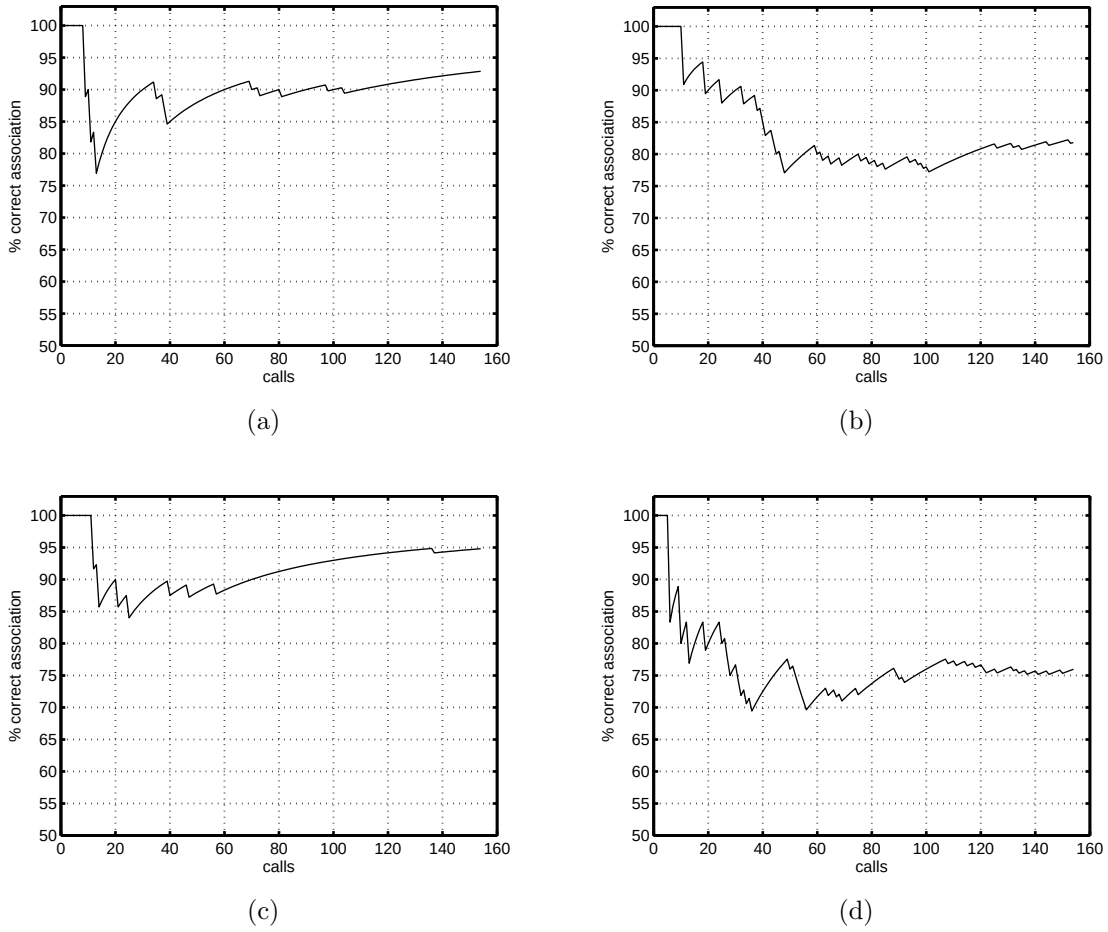
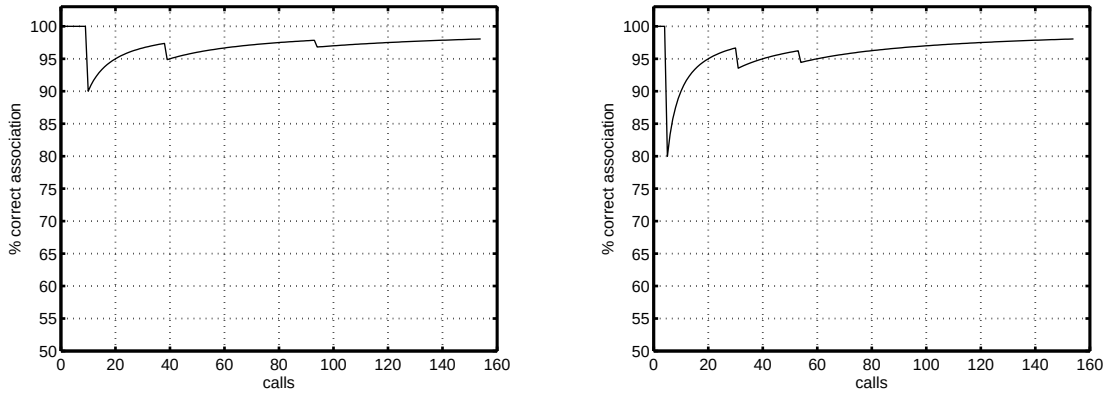


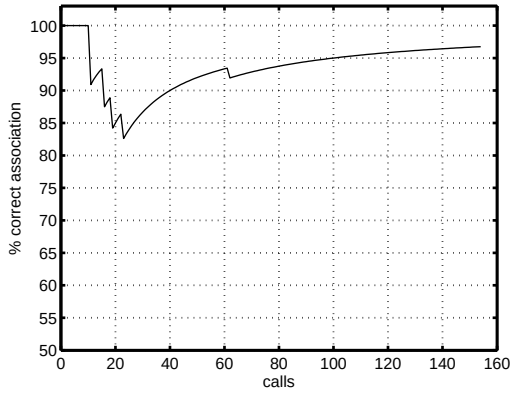
FIGURE 4.6. The percent correct association using Likelihood test for non-overlap test sequences 7 (a) - 10 (d).

4.4.2. EXPERIMENT 2: OVERLAPPING TEST SEQUENCES. As for overlap detection, the overlapping test sequences were similarly processed but each detected call is first applied to the overlap detection to determine whether an overlapping call has occurred. As mentioned before, this is done to avoid performance degradation of the system. The performance of the system in this case is measured by the probability of detection (P_D) and the probability of false alarm (P_{FA}) which are, in this case, defined for each test sequence as:

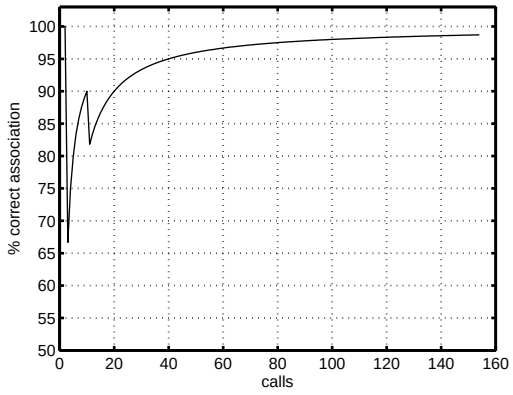


(a)

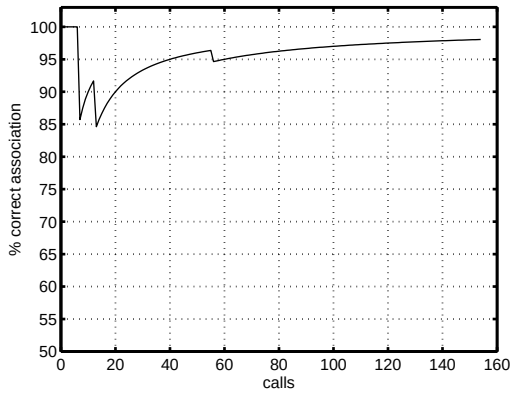
(b)



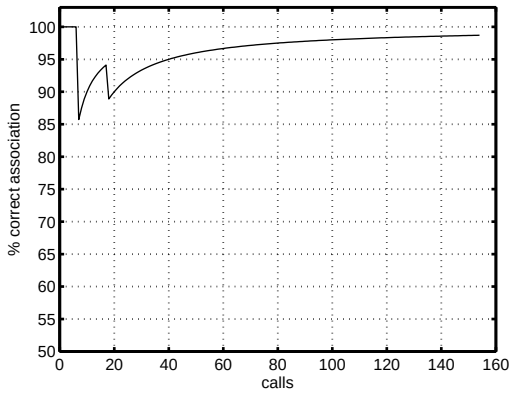
(c)



(d)



(e)



(f)

FIGURE 4.7. The percent correct association using KL-divergence test for non-overlap test sequences 1 (a) - 6 (f).

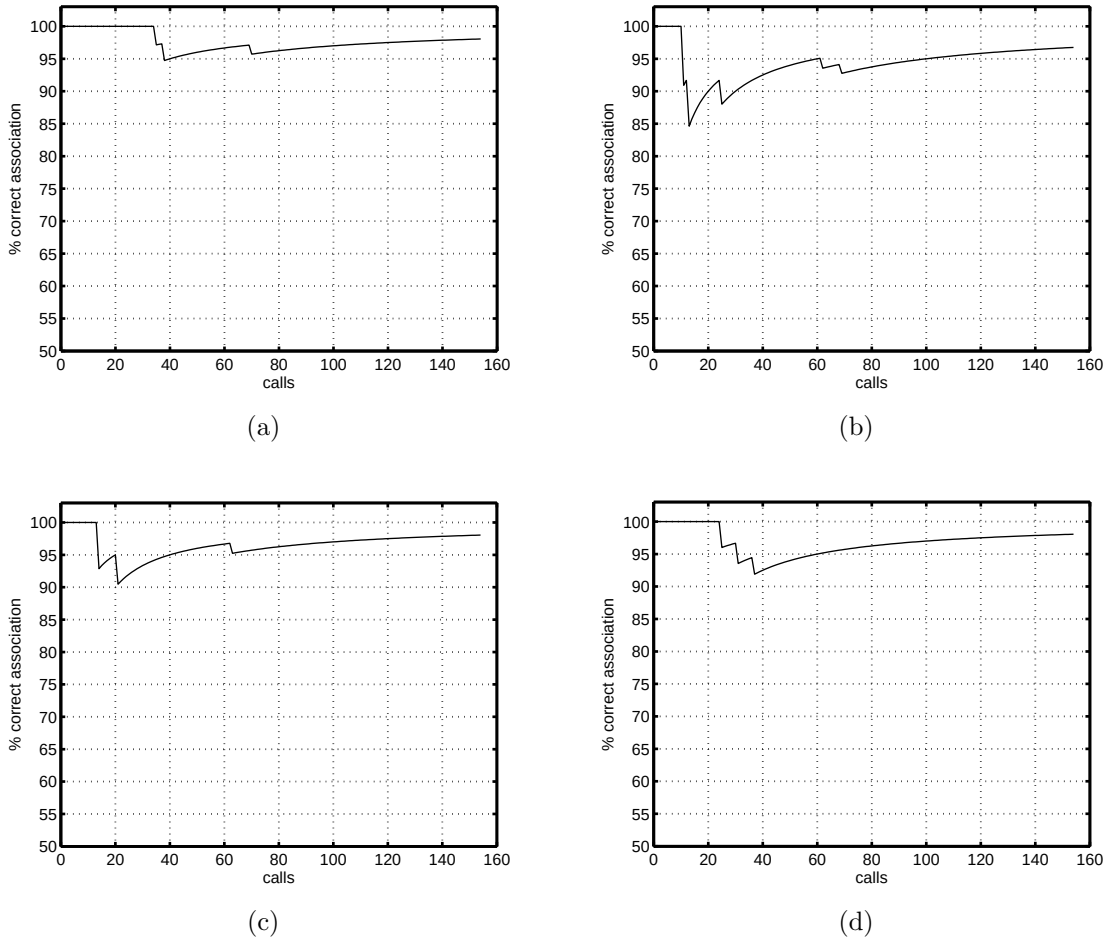


FIGURE 4.8. The percent correct association using KL-divergence test for non-overlap test sequences 7 (a) - 10 (d).

$$P_D = \frac{\#\ correctly\ detected\ overlapping\ calls}{total\ \#\ overlapping\ calls}$$

$$P_{FA} = \frac{\#\ falsely\ detected\ single\ calls\ as\ overlapping\ calls}{total\ \#\ non-overlapping\ calls}$$

The receiver operating characteristic (ROC) curve [45] of the overlap detection is shown in Fig. 4.9 where the threshold η is varied from 0 to 0.25. The knee point of the ROC at

TABLE 4.2. Population estimates for the Liklihood and KL-divergence methods on non-overlapping test sequences.

	population estimate	
sequence #	Likelihood test	KL-divergence test
1	15	14
2	12	13
3	14	15
4	14	13
5	14	13
6	15	13
7	15	14
8	15	13
9	12	14
10	13	13
mean	13.9	13.5
std. deviation	1.20	0.71

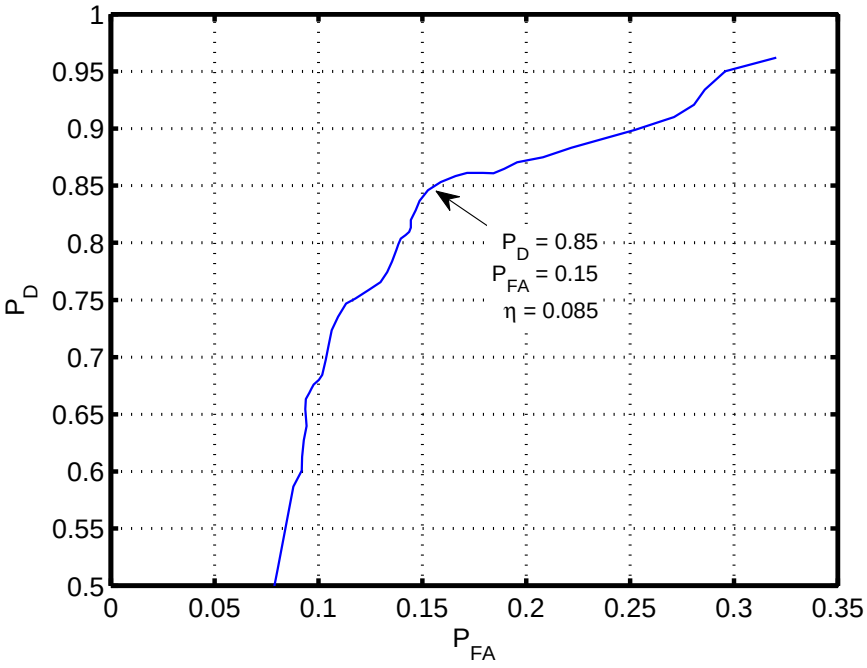


FIGURE 4.9. The ROC of the overlap detection.

which $P_D + P_{FA} = 1$ for this detector (when $\eta = 0.085$) gives $P_D = 0.85$ and $P_{FA} = 0.15$. The P_D can be increased further at the expense of a higher P_{FA} and this will mean that more non-overlapping calls will be mistakenly excluded from the learning process which can

TABLE 4.3. Association performance of Likelihood test for non-overlapping test sequences (a) 1 - (f) 6.

Model #	Associated Call True IDs
1	'5 5 5 5 5 5 5 5 5'
2	'2 2 2 2 2 2 2 2 2 2 2 2'
3	'11 11 11 11 11 11 11 11 11 11 11'
4	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
5	'4 4 4 4 4 4'
6	'1 1 1 1 1 1 1 1 1 1 1 1 1'
7	'10 10 10 10 10 10 10 10 10 10 10 10'
8	'11 11 11'
9	'7 7 7 7 7 7 7 7 7 7 7 7 7'
10	'9 9 9 9 9 9 9 9 9 9 9 9 9'
11	'6 6 6 6'
12	'6 6 6 6 6 6 6 6 6 6'
13	'4 4 4 4 4 4 4'
14	'3 3 3 3 3 3 3 3 3 3 3'
15	'3 3'

(a)

Model #	Associated Call True IDs
1	'8 10 10 10 10 8 9 9 10 8 8 10 10 9 10 8 9 8 10 9 9 10 8 8 10 8 9 9 9 9 10 9 8 8 8 10 9 9 10 9 8 8'
2	'4 4 4 4 4 4 4 4 4'
3	'6 6 6 6 6 6 6 6 6'
4	'6 6 6 6 6'
5	'4 4 4 4 4'
6	'1 1 1 1 1 1 1 1 1 1 1'
7	'11 11 11 11 11 11 11 11 11 11 11'
8	'2 2 2 2 2 2 2 2 2 2 2 2'
9	'5 5 5 5 5 5 5 5 5 5'
10	'3 3 3 3 3 3 3 3 3 3'
11	'3 3 3 3'
12	'7 7 7 7 7 7 7 7 7 7 7 7'

(b)

Model #	Associated Call True IDs
1	'5 5 5 5 5 5 5 5 5 5 5'
2	'9 9 9 9 9 9 9 9 9 9 9 9 9'
3	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
4	'4 4'
5	'3 3 3 3 3'
6	'3 3 3 3 3 3 3 3 3'
7	'6 6 6 6 6 6 6 6'
8	'10 10 10 10 10 10 10 10 10 10 10 '
9	'2 2 2 2 2 2 2 2 2 2 2 2 2'
10	'7 7 7 7 7 7 7 7 7 7 7 7 7'
11	'4 4 4 4 4 4 4 4 4 4 4'
12	'11 11 11 11 11 11 11 11 11 11 11 11'
13	'6 6 6 6'
14	'1 1 1 1 1 1 1 1 1 1 1 1'

(c)

Model #	Associated Call True IDs
1	'11 11 11 11 11 11 11 11 11 11 11 11'
2	'3 3 3 3 3 3 3 3'
3	'2 2 2 2 2 2 2 2 2 2 2 2 2'
4	'4 4 4 4 4 4 4 4 4'
5	'9 9 9 9 9 9 9 9 9 9 9 9 9'
6	'8 8 8 10 10 10 8 10 8 10 8 10 10 8 8 10 8 8 10 10 10 8 10 8 10 8 8 10'
7	'5 5 5 5 5 5 5 5 5 5'
8	'3 3 3 3 3 3'
9	'6 6'
10	'7 7 7 7 7 7 7 7 7 7 7 7'
11	'4 4 4'
12	'1 1 1 1 1 1 1 1 1 1 1 1'
13	'6 6 6 6 6'
14	'6 6 6 6 6'

(d)

Model #	Associated Call True IDs
1	'9 9 9 9 9 9 9 9 9 9 9 9 9'
2	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
3	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
4	'3 3 3 3 3 3 3 3 3 3'
5	'1 1 1 1 1 1 1 1 1 1 1 1 1'
6	'3 3 3 3 3'
7	'4 4 4 4 4 4 4 4'
8	'4 4 4 4'
9	'11 11 11 11 11 11 11 11 11 11 11 11'
10	'5 5 5 5 5 5 5 5 5 5'
11	'2 2 2 2 2 2 2 2 2 2 2 2'
12	'7 7 7 7 7 7 7 7 7 7 7 7'
13	'6 6 6 6 6 6 6 6'
14	'6 6 6'

(e)

Model #	Associated Call True IDs
1	'10 10 10 10 10 10 10 10 10 10 10 10'
2	'2 2 2 2 2 2 2 2 2 2 2 2 2'
3	'9 9 9 9 9 9 9 9 9 9 9 9 9'
4	'3 3 3 3 3 3 3 3 3 3 3'
5	'6 6 6 6 6 6 6 6 6'
6	'3 3 3'
7	'8 8 8 8 8 8 8 8 8 8 8 8 8'
8	'4 4 4 4 4 4 4 4 4 4 4'
9	'1 1 1 1 1 1 1 1 1 1'
10	'6 6 6 6'
11	'5 5 5 5 5 5 5 5 5'
12	'7 7 7 7 7 7 7 7 7 7 7 7'
13	'4 4'
14	'11 11 11'
15	'11 11 11 11 11 11 11 11 11 11 11'

(f)

TABLE 4.4. Association performance of Likelihood test for non-overlapping test sequences (a) 7 - (d) 10.

Model #	Associated Call True IDs
1	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
2	'5 5 5 5 5 5 5 5 5 5'
3	'4 4 4 4 4 4 4 4'
4	'3 3 3 3 3 3 3 3 3 3 3 3 3 3 3'
5	'6 6 6 6 6 6 6 6 6 6 6 6'
6	'7 7 7 7 7 7 7 7'
7	'4 4 4 4 4 4'
8	'11 11 11 11 11 11 11 11 11 11 11 11 11 11 11'
9	'7 7 7 7 7 7'
10	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
11	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
12	'1 1 1 1 1 1 1 1 1 1 1 1 1'
13	'2 2 2 2 2 2 2 2 2 2'
14	'2 2'
15	'6 6'

(a)

Model #	Associated Call True IDs
1	'7 7 7 7 7 7 7 7 7 7 7 7 7'
2	'8 8 8 8 8 10 8 10 8 8 10 10 10 8 10 10 10 8 8 10 8 10 8 10 8 10 10'
3	'3 3 3 3 3 3 3 3'
4	'1 1 1 1 1 1 1 1 1 1 1 1 1'
5	'5 5'
6	'2 2 2 2 2 2 2 2 2 2 2 2 2'
7	'9 9 9 9 9 9 9 9 9 9 9 9 9'
8	'11 11 11 11 11 11 11 11 11 11 11 11 11 11 11'
9	'4 4 4 4 4 4 4 4 4 4 4 4'
10	'5 5 5 5 5 5'
11	'6 6 6 6'
12	'6 6 6 6'
13	'6 6 6'
14	'5 5'
15	'3 3 3 3 3 3'

(b)

Model #	Associated Call True IDs
1	'4 4 4 4 4 4 4 4 4 4 4 4 4 4 4'
2	'8 8 8 8 8 8 8 8 8 8 8 8 8 8 8'
3	'2 2 2 2 2 2 2 2 2 2'
4	'10 10 10 10 10 10 10 10 10 10 10 10 10 10 10'
5	'1 1 1 1 1 1 1 1 1 1 1 1 1 1 1'
6	'7 7 7 7 7 7 7 7 7 7 7 7 7'
7	'3 3 3 3 3 3 3 3 3 3 3 3 3 3 3'
8	'2 2 2 2 2'
9	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
10	'6 6 6 6 6 6 6 6 6 6 6 6 6'
11	'11 11 11 11 11 11 11 11 11 11 11 11 11 11 11'
12	'5 5 5 5 5 5 5 5'

(c)

Model #	Associated Call True IDs
1	'2 2 2 2'
2	'11 11 11 11 11 11 11 11'
3	'5 5 5 5 5 5 5 5 5 5'
4	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
5	'8 9 8 9 9 9 9 8 8 9 9 9 8 8 8 9 8 8 9 9 8 9 8 9 9 8 8 8'
6	'3 3 3 3 3 3 3'
7	'6 6 6 6 6 6 6 6 6 6 6 6'
8	'7 7 7 7 7 7 7 7 7 7 7 7 7'
9	'1 1 1 1 1 1 1 1 1 1 1 1 1'
10	'11 11 11 11 11 11'
11	'3 3 3 3 3 3 3'
12	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
13	'2 2 2 2 2 2 2 2 2'

(d)

have some impact on the overall performance of the system depending on how many calls are detected and available for the learning process. Nonetheless, the effects of excluding some non-overlapping calls is less dramatic than including the false negatives.

A simulation was conducted to asses the affect of using the overlap detector on the overall performance of the system when applying the overlapping test sequences. Similarly, the results here are reported in the form of a comparison of the percent correct association

TABLE 4.5. Association performance of KL-divergence test for non-overlapping test sequences (a) 1 - (f) 6.

Model #	Associated Call True IDs
1	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
2	'2 2 2 2 2 2 2 2 2 2'
3	'11 11 11 11 11 11 11 11 11 11 11'
4	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
5	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
6	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
7	'10 10 10 10 10 10 10 10 10 10 10 10 10'
8	'11 11 11'
9	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
10	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
11	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
12	'2 2 2 2'
13	'3 3 3 3 3 3 3 3 3 3 3 3'
14	'3 3'

(a)

Model #	Associated Call True IDs
1	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
2	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
3	'10 10 10 10 10 10 10 10 10 10 10 10 10 10 10'
4	'6 6 6 6 6'
5	'6 6 6 6 6 6 6 6 6'
6	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
7	'2 2 2 2 2 2 2 2 2 2 2 2 2 2'
8	'11 11 11 11 11 11 11 11 11 11 11 11'
9	'9 9 9 9 9 9 9 9 9 9 9 9 9'
10	'5 5 5 5 5 5 5 5 5 5 5 5 5'
11	'3 3 3 3 3 3 3 3 3 3'
12	'3 3 3 3'
13	'7 7 7 7 7 7 7 7 7 7 7 7 7'

(b)

Model #	Associated Call True IDs
1	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
2	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
3	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
4	'2 2 2 2 2 2'
5	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
6	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
7	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
8	'3 3 3 3 3'
9	'3 3 3 3 3 3 3 3 3'
10	'11 11 11 11 11 11 11 11 11 11 11 11 11'
11	'2 2'
12	'7 7 7 7 7 7 7 7 7 7 7 7 7'
13	'2 2 2 2'
14	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
15	'2 2'

(c)

Model #	Associated Call True IDs
1	'11 11 11 11 11 11 11 11 11'
2	'3 3 3 3 3 3 3 3'
3	'11 11 11 11 11'
4	'2 2 2 2 2 2 2 2 2 2 2 2 2'
5	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
6	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
7	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
8	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
9	'3 3 3 3 3 3'
10	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
11	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
12	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
13	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'

(d)

Model #	Associated Call True IDs
1	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
2	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
3	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
4	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
5	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
6	'3 3 3 3 3 3 3 3 3 3'
7	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
8	'3 3 3 3 3'
9	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
10	'11 11 11 11 11 11 11 11 11 11 11 11 11'
11	'2 2 2 2 2 2 2 2 2 2 2'
12	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
13	'2 2 2 2'

(e)

Model #	Associated Call True IDs
1	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
2	'2 2 2 2 2 2 2 2 2 2 2 2 2 2'
3	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
4	'3 3 3 3 3 3 3 3 3 3 3 3'
5	'6 6 6 6 6 6 6 6 6'
6	'3 3 3'
7	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
8	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
9	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
10	'6 6 6 6 6'
11	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
12	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
13	'11 11 11 11 11 11 11 11 11 11 11 11 11'

(f)

TABLE 4.6. Association performance of KL-divergence test for non-overlapping test sequences (a) 7 - (d) 10.

Model #	Associated Call True IDs
1	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
2	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
3	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
4	'3 3 3 3 3 3 3 3 3 3 3 3 3'
5	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
6	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
7	'11 11 11 11 11 11 11 11 11 11 11 11 11 11'
8	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
9	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
10	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
11	'2 2 2 2 2 2 2 2 2 2'
12	'2 2'
13	'2 2'
14	'3 3'

(a)

Model #	Associated Call True IDs
1	'2 2 2 2 2'
2	'7 7 7 7 7 7 7 7 7 7 7 7 7'
3	'8 8 8 8 8 8 8 8 8 8 8 8 8'
4	'3 3 3 3 3 3 3 3'
5	'1 1 1 1 1 1 1 1 1 1 1 1 1'
6	'5 5 5 5 5 5 5 5 5 5 5 5 5'
7	'2 2 2 2 2 2 2 2'
8	'9 9 9 9 9 9 9 9 9 9 9 9 9'
9	'11 11 11 11 11 11 11 11 11 11 11 11 11 11'
10	'4 4 4 4 4 4 4 4 4 4 4 4 4'
11	'6 6 6 6 6 6 6 6 6 6 6 6 6'
12	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
13	'3 3 3 3 3 3'

(b)

Model #	Associated Call True IDs
1	'4 4 4 4 4 4 4 4 4 4 4 4 4 4'
2	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
3	'2 2 2 2 2 2 2 2 2 2'
4	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
5	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
6	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
7	'3 3 3 3 3 3 3 3 3 3 3'
8	'2 2 2 2 2'
9	'6 6 6 6 6 6'
10	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
11	'6 6 6 6 6 6 6 6'
12	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
13	'11 11 11 11 11 11 11 11 11 11 11 11 11 11'
14	'3 3 3'

(c)

Model #	Associated Call True IDs
1	'2 2 2 2'
2	'11 11 11 11 11 11 11 11 11 11 11 11 11 11'
3	'5 5 5 5 5 5 5 5 5 5 5 5 5 5'
4	'10 10 10 10 10 10 10 10 10 10 10 10 10 10'
5	'8 8 8 8 8 8 8 8 8 8 8 8 8 8'
6	'9 9 9 9 9 9 9 9 9 9 9 9 9 9'
7	'3 3 3 3 3 3 3'
8	'6 6 6 6 6 6 6 6 6 6 6 6 6 6'
9	'7 7 7 7 7 7 7 7 7 7 7 7 7 7'
10	'1 1 1 1 1 1 1 1 1 1 1 1 1 1'
11	'3 3 3 3 3 3 3'
12	'2 2 2 2 2 2 2 2 2 2'
13	'4 4 4 4 4 4 4 4 4 4 4 4 4'

(d)

with and without the overlap detector. However, since the number of calls processed by the system is different over the different test sequences due to the rejection of a different number of calls by the overlap detector, we will only report the final percent correct association for each test sequence. Also, a new error scenario is defined here in addition to the errors defined in the beginning of Section 4.4 of this Chapter. An error will also occur when a call passed to the learning system by the overlap detector is overlapping, i.e. a miss detection has occurred,

TABLE 4.7. Percent correct association and population estimates on overlapping test sequences with and without overlap detection.

	% correct assoc.		population estimate	
sequence #	no overlap det.	overlap det.	no overlap det.	overlap det.
1	35.6	85.4	15	7
2	37.5	66.7	14	6
3	23.1	31.3	15	8
4	38.5	84.4	14	9
5	41.4	65.1	14	8
6	40.4	77.8	12	11
0	42.3	66.7	15	9
8	34.6	81.6	14	9
9	32.7	66.7	17	11
10	40.4	91.9	13	8
mean	36.6	71.8	14.3	8.6
std. deviation	5.7	17.2	1.3	1.6

and both members of the overlapping call satisfy either one of the error scenarios defined in the beginning of Section 4.4 of this Chapter. If at least one individual of the overlapping call does not satisfy either of the error scenarios then the association is considered correct. Table 4.7 gives the resulting percent correct association for each overlapping test sequence, as well as, the resulting population estimates. From Table 4.7 it is apparent that the association performance of the system improves by an average of 35.1% when applying overlap detection.

In order to show the impact of using the overlap detection on the overall system performance, a new indicator, the percentage of correct decisions, is used. For this indicator, miss detections (false negatives) and false alarms from the overlap detector are treated as errors in addition to the association errors defined at the beginning of Section 4.4 of this Chapter. This means that the percent correct decisions shown here not only reflects the learning performance of the system but it also incorporates the overlap detection performance. Fig. 10(a) shows a plot of the average percent correct decisions when no overlap detection was used while Fig. 10(b) shows the same when the overlap detector is operating at the knee

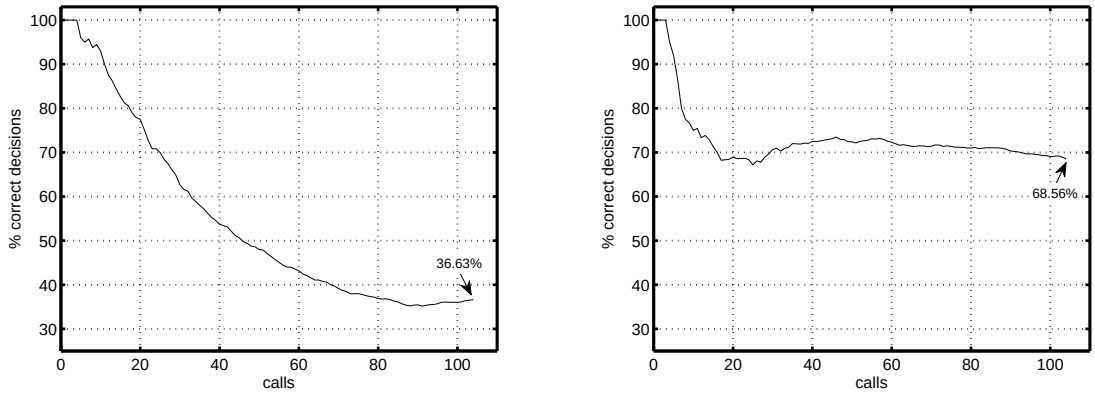


FIGURE 4.10. The average percent correct decisions on overlapping test sequences (a) without overlap detection and (b) with overlap detection.

point threshold (i.e. $\eta = 0.085$) of the ROC in Fig. 4.9. These plots attest to the importance of overlap detection for preventing a large degradation in the system performance. They also show that using the overlap detection can improve the overall system performance by upto 31.9% as can be seen in Fig. 4.10. Therefore, being able to identify and discard overlapping calls is crucial.

4.4.3. EXPERIMENT 3: EXTENDED TEST SEQUENCES. In this experiment, extended test sequences were applied to the system. The extended sequences were generated by concatenating the non-overlapping test sequences with the overlapping test sequences. This experiment was conducted to investigate the effect of having more data available on the performance of the system. Similar to experiment 2, the results here are reported in the form of a comparison of the final percent correct association with and without overlap detection. Table 4.8 gives the resulting percent correct association for each extended test sequence, as well as, the resulting population estimates. From Table 4.8 it is apparent that the association performance of the system improves by an average of 15.4% when applying overlap detection, while the population estimate improves more significantly. The population estimates here

TABLE 4.8. Percent correct association and population estimates on extended test sequences with and without overlap detection.

	% correct assoc.		population estimate	
sequence #	no overlap det.	overlap det.	no overlap det.	overlap det.
1	79.1	97.1	14	10
2	79.5	96.9	17	13
3	68.2	79.2	16	12
4	79.8	95.7	13	14
5	79.8	93.7	13	12
6	79.5	92.3	15	12
0	78.7	94.8	17	11
8	78.7	96.6	14	13
9	79.5	95.3	16	10
10	79.5	94.7	15	13
mean	78.2	93.6	15	12
std. deviation	3.5	5.3	1.5	1.3

are better, on average, than those in experiment 1 when the non-overlapping sequences were used. This is due the fact that more data is available in this case when combining the non-overlapping calls from both parts of the extended sequence.

In order to show the impact of using the overlap detection on the overall system performance in this case, the percentage of correct decisions is generated here as well. Similar to experiment 2, the percent correct decisions not only reflects the learning performance of the system but it also incorporates the overlap detection performance. Fig. 11(a) shows a plot of the average percent correct decisions when no overlap detection was used while Fig. 11(b) shows the same when the overlap detector, operating at the knee point threshold (i.e. $\eta = 0.085$), was used. These plots show that when using overlap detection the overall performance of the system initially suffers due to the errors produced by the overlap detector, i.e false alarms, but eventually recovers and slightly surpasses the overall performance of the system without overlap detection.

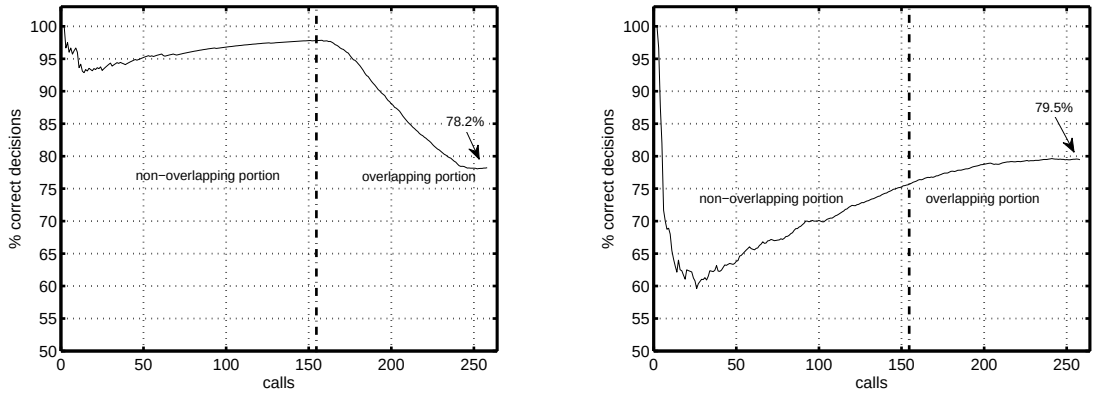


FIGURE 4.11. The average percent correct decisions on extended test sequences (a) without overlap detection and (b) with overlap detection.

4.5. CONCLUSION

In this chapter, the results of evaluating the proposed system's performance were presented and discussed. Several synthetic test sequences were generated and applied to the system and the performance was determined in terms of correct association and overlap detection. Three experiments were conducted. In the first experiment, Only the non-overlapping test sequences were applied to the system to evaluate the learning performance though the percent correct association. The results have shown that the system can achieve an average of 97.9% using the KL-divergence method the non-overlapping sequences without using overlap detection. In the second experiment, the non-overlapping test sequences were applied to the system to evaluate the overlap detection and rejection capability. The ROC of the detector was computed which showed that the overlap detector, operating at the knee point of ROC, can achieve $P_D = 0.85$ and $P_{FA} = 0.15$. Furthermore, the impact of using the overlap detection of the overlapping test sequences was studied. It was found that this detector improves the percent correct association, i.e. the learning performance, by an average of 35.1% and when including the effect of the overlap detector errors, i.e. false alarms and

miss detections, the overall system performance increases by an average of 31.9%. In the third experiment, an extended sequence, including non-overlapping as well as overlapping portions, was generated and applied to the system. In this case, an average of 15.4% increase in percent correct association and a corresponding 1.3% increase in overall system performance was reported when using an overlap detector. Furthermore, the population estimates in this case using the overlap detector were much closer to the actual population of 11 when compared to previous estimates.

CHAPTER 5

Conclusion and Future Work

5.1. Conclusion

Estimating the population of frogs of different species is an important problem for environmental monitoring purposes as well as ecological research [3]. This importance is attributed to the fact that frogs are very sensitive to the environmental condition due to their delicate, permeable skin and water-based life cycle [2]. It has become of great interest to monitor frog populations in order to serve as an indicator of the overall health of the environment [1].

Several methods for estimating the population of frogs in the environment exist including call count and call survey methods, as well as, CMR methods [5]. However, these methods are lacking in both the accuracy of the population estimates produced, and the efficiency of the process. The use of passive acoustic monitoring for frog population estimation or, in general, animal population estimation, is an effective and non-invasive approach which can provide the needed accuracy and efficiency. Many acoustic-based systems have been developed for the purpose of recognizing different species of animals [1, 8, 9, 10, 11, 12, 15, 16]. A fewer number of acoustic systems have been developed for the purpose of recognizing individuals of the same species [13, 14, 23]. None of the existing acoustic systems provide the capability of estimating the population of the animals being observed acoustically. An acoustic system that is capable of estimating the population of the monitored animals requires that the vocalizations of the individuals in the recordings be distinguishable [51]. The proposed system in this thesis was developed with this precise goal in mind i.e. to be able to distinguish individual frogs from

their vocalizations or calls. The system should be able to provide an estimate of how many individuals are present in the recording and, hence, in the population.

The proposed population estimation using the in-situ progressive learning acoustic system was introduced and described in detail in this thesis. The proposed system used a combination of MFCC features extracted on a window-by-window basis, and multivariate Gaussian modeling to achieve individual frog learning. Spectral-based call segmentation was used to detect and isolate frog calls in the input acoustic signal where spectral matrices were computed for each detected call. MFCC features were then extracted from the columns of the spectral matrices for each detected call. These features are used in the progressive learning component of the system where Gaussian models were generated from the data to model the calls from the different individual frogs and newly detected calls were tested for possible associations with previously generated models. After processing an entire acoustic recording, the number of models initiated throughout the process was used to estimate the population. The proposed system was shown to be capable of learning to recognize different individuals by being progressively exposed to their calls. This learning ability eliminates the requirement for a prior training data that would otherwise be required. It was mentioned that this capability allows the system to operate in-situ which makes it more advantageous. The proposed system was shown to be capable of generating models to represent different individual frogs from the detected calls. Also, the system was shown to include overlap detection capability to detect and reject temporally overlapping calls.

The data set comprising of calls of different individual frogs was described and synthetic test sequences were generated for evaluating the performance of the system. The generated sequences included provisions to make them more realistic by including artificially induced

temporally overlapping calls. This was done due to the fact that in any real acoustic recording of frogs, which would be used to estimate the population, many such overlapping calls would occur due to the sometimes very high call rates from multiple frogs in the same proximity. Therefore, it was important to analyze the effect of such overlapping calls to determine the viability of applying such a system in the field. Three experiments were conducted to evaluate the system's performance in terms of it's learning ability and overlap rejection ability. Based on the simulation results presented, it was shown that the proposed system excelled at the task of distinguishing frog calls from the different individual frogs. The system was able to produce high percent correct association, 97.9% correct association, and was robust to changes in the order of calls in the input acoustic signal when using the KL-divergence based test method. Furthermore, the overlap detection component was able to detect a significant number of the overlapping calls, 85% probability of detection, and reject them. The overlap detection was shown to have a significant impact of the learning performance of the system. It should be noted that high percent correct association did not necessarily produce a proportionally accurate population estimate. This is due to the fact that a negligible number of association errors can greatly impact the population estimate by initiating additional models. Additionally, the system may not be capable of dealing with extremely dense choruses where the acoustic recording contains long segments of continuously overlapping calls.

Based on the results presented in this thesis, it is evident that the proposed system provides a viable acoustic frog population estimation method. The system provides a foundation for a new avenue in acoustic-based animal population estimation.

5.2. Suggestions for Future Work

Although the acoustic system proposed in this thesis does demonstrate preliminary success in being able to accurately distinguish between calls of individual frogs, it has not yet been tested on a more realistic set of data that could potentially have more varied conditions such as greater interference, and more spontaneity in the frog calls produced. Therefore, in order to evaluate the true performance applicability of this system, new data sets should be applied to the system. The ideal test data that could be used in this case would be a recording of a certain number of frogs in a lab controlled environment which will provide reliable data along with an accurate ground truth. It would also be of interest to determine the minimum number of non-overlapping calls required to achieve a certain percent correct association. This could be determined by running a Monte Carlo experiment [52] over many test sequences with randomly ordered calls.

This acoustic system is a first attempt at using individual recognition in the application of frog population estimation. It is also, as far as we can tell, the first system to apply an in-situ progressive learning approach which eliminates the need for prior system training. Given the novelty of this approach, several improvements could be explored to potentially increase its accuracy and robustness to noise, interference, and other unfavorable conditions. These improvements include but are not limited to:

- Including additional pre-processing steps such as digital pre-emphases [53] to enhance the high frequency content of the acoustic recording. Such high frequency content can suffer from higher attenuation during propagation and may include additional information on inter-individual variations in the frogs calls which could be exploited to improve system performance.

- Better understanding the inter-individual variations in frog calls and develop acoustic features that can capture these variations. A similar procedure used in developing the MFCC features may be followed where the frog vocalization and auditory perception systems are studied to develop specific features that emphasize inter-individual variations among frog calls.
- Exploring the use of more sophisticated classification methods such the GMM in conjunction with an EM-based parameter estimation method. GMMs can offer better feature vector distribution modeling capability allowing for more efficient capture of the inter-individual variations. Given the challenge of initial data availability posed by the proposed system, such a sophisticated classification method can only be used in a gradual fashion where the number of GMM components, for example, is increased gradually based the amount of data available.

The proposed system can also be incorporated into a larger animal tracking system which could be used to eventually learn and track the individual movements of many different vocalizing animal species. Such a system would rely on strategically placed recording devices over an area of interest. The resulting acoustic recordings can be processed to learn to recognize the different individual animals in the area and, therefore, be able to recognize them if they move to a different locality thus providing tracking capability. This large scale system could be easily implemented as an extension to the system proposed here, and would provide a low cost alternative to existing satellite-based systems , i.e. GPS tracking systems, and could extend tracking capabilities to smaller animals that are currently not supported by existing animal tracking systems [54]. The benefits of such capability are enormous and

will allow researchers to further understand issues such as animal migration patterns, the spread of infectious disease, biological diversity, ecosystem function, and many others.

Bibliography

[1] G. Gary and Q. Fu, "Automatic frog calls monitoring system: a machine learning approach," *International Journal of Computational Intelligence and Applications*, vol. 1, no. 2, pp. 165–186, 2001.

[2] T. Gardner, "Declining amphibian populations: a global phenomenon in conservation biology," *Animal Biodiversity and Conservation*, vol. 2, pp. 25–44, 2001.

[3] L. Vitt, J. Caldwell, H. Wilbur, and D. Smith, "Amphibians as harbingers of decay," *BioScience*, vol. 40, no. 6, 1990.

[4] J. Pechmann and H. Wilbur, "Putting declining amphibian populations in perspective: natural fluctuations and human impacts," *Herpetologica*, 1994.

[5] J. Pellet, V. Helfer, G. Yannic, and C. Romain, "Estimating population size in the European tree frog (Hyla arborea) using individual recognition and chorus counts," *Population Trends*, vol. 28, pp. 287–294, 2007.

[6] B. R. Schmidt, "Declining amphibian populations: The pitfalls of count data in the study of diversity, distributions, dynamics, and demography," *Herpetological Journal*, vol. 14, no. 4, pp. 167–174, 2004.

[7] E. J. Fox, "A new perspective on acoustic individual recognition in animals with limited call sharing or changing repertoires," *Animal Behaviour*, vol. 75, pp. 1187–1194, Mar. 2008.

[8] C.-H. Lee, C.-H. Chou, C.-C. Han, and R.-Z. Huang, "Automatic recognition of animal vocalizations using averaged MFCC and linear discriminant analysis," *Pattern Recognition Letters*, vol. 27, pp. 93–101, Jan. 2006.

[9] C.-J. Huang, Y.-J. Yang, D.-X. Yang, and Y.-J. Chen, "Frog classification using machine learning techniques," *Expert Systems with Applications*, vol. 36, pp. 3737–3743, Mar. 2009.

[10] A. Harma, "Automatic identification of bird species based on sinusoidal modeling of syllables," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. V–545, IEEE, 2003.

[11] W.-P. Chen, S.-S. Chen, C.-C. Lin, Y.-Z. Chen, and W.-C. Lin, "Automatic recognition of frog calls using a multi-stage average spectrum," *Computers & Mathematics with Applications*, vol. 64, pp. 1270–1281, Sept. 2012.

[12] S. Fagerlund, "Bird species recognition using support vector machines," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 64–64, 2007.

[13] J. Cheng and Y. Sun, "A call-independent and automatic acoustic system for the individual recognition of animals: A novel model using four passerines," *Pattern Recognition*, vol. 43, pp. 3846–3852, Nov. 2010.

[14] B. Zhang, J. Cheng, Y. Han, L. Ji, and F. Shi, "An acoustic system for the individual recognition of insects.," *The Journal of the Acoustical Society of America*, vol. 131, pp. 2859–65, Apr. 2012.

[15] H. Tyagi and R. Hegde, "Automatic identification of bird calls using spectral ensemble average voice prints," *Proceedings of the 13th ...*, pp. 1–5, 2006.

[16] P. Somervuo, A. Harma, and S. Fagerlund, "Parametric representations of bird sounds for automatic species recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 6, pp. 2252–2263, 2006.

[17] H. Beigi, "Signal processing of speech and feature extraction," in *Fundamentals of Speaker Recognition*, pp. 143–204, Springer US, 2011.

[18] A. M. Martinez, A. M. Mart'inez, and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 228–233, 2001.

[19] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, 1980.

[20] D. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometric Recognition*, vol. 2, no. 17.36, pp. 14–68, 2008.

[21] H. Beigi, "Hidden markov modeling (hmm)," in *Fundamentals of Speaker Recognition*, pp. 411–463, Springer US, 2011.

[22] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[23] E. J. Fox, J. D. Roberts, and M. Bennamoun, "Call-independent individual identification in birds," *Bioacoustics*, vol. 18, no. 1, pp. 51–67, 2008.

[24] K. R. Farrell, R. J. Mammone, and K. T. Assaleh, "Speaker recognition using neural networks and conventional classifiers," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 1, pp. 194–205, 1994.

[25] A. Lawson, P. Vabishchevich, M. Huggins, P. Ardis, B. Battles, and A. Stauffer, "Survey and evaluation of acoustic features for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5444–5447, IEEE, 2011.

[26] M. a. Roch, M. S. Soldevilla, J. C. Burtenshaw, E. E. Henderson, and J. a. Hildebrand, "Gaussian mixture model classification of odontocetes in the Southern California Bight

and the Gulf of California," *The Journal of the Acoustical Society of America*, vol. 121, no. 3, p. 1737, 2007.

[27] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," 1993.

[28] D. M. G. Watts, "Speaker identification-prototype development and performance," *Faculty of Engineering & Surveying*, p. 116, 2006.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[30] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, 2001.

[31] E. Zwicker, "Subdivision of the audible frequency range into critical bands (Frequenzgruppen)," *The Journal of the Acoustical Society of America*, vol. 365, no. 1956, p. 2013, 1961.

[32] S. S. Stevens and J. Volkmann, "The relation of pitch to frequency: A revised scale," *The American Journal of Psychology*, vol. 53, no. 3, pp. pp. 329–353, 1940.

[33] D. O'Shaughnessy, *Speech communications: human and machine*. Institute of Electrical and Electronics Engineers, 2000.

[34] O. C. Ai, M. Hariharan, S. Yaacob, and L. S. Chee, "Classification of speech dysfluencies with mfcc and lpcc features," *Expert Systems with Applications*, vol. 39, no. 2, pp. 2157 – 2165, 2012.

[35] P. Brockwell and R. Davis, *Introduction to Time Series and Forecasting*. Lecture Notes in Statistics, Springer, 2002.

[36] G. Antoniol, V. F. Rollo, and G. Venturi, "Linear predictive coding and cepstrum coefficients for mining time variant information from software repositories," *SIGSOFT Softw. Eng. Notes*, vol. 30, pp. 1–5, May 2005.

[37] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," *Expert Syst. Appl.*, vol. 36, pp. 6069–6075, Apr. 2009.

[38] G. Casella and E. I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. pp. 167–174, 1992.

[39] C. Canuto, Y. Hussaini, and A. Quarteroni, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Scientific Computation, Springer-Verlag Berlin Heidelberg, 2007.

[40] M. Tarter and M. Lock, *Model-free Curve Estimation*. Monographs on Statistics and Applied Probability, Chapman and Hall, 1993.

[41] J. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, 1993.

[42] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. No. v. 1 in Prentice Hall Signal Processing Series, Prentice Hall, 1993.

[43] J. Buschbom and A. Haeseler, "Introduction to applications of the likelihood function in molecular evolution," in *Statistical Methods in Molecular Evolution*, Statistics for Biology and Health, pp. 25–44, Springer New York, 2005.

[44] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2006.

[45] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. New Jersey: Prentice-Hall Inc, 1993.

[46] S. Roberts and W. Penny, "Variational Bayes for generalized autoregressive models," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2245–2257, Sept. 2002.

[47] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," *Proc. DARPA Broadcast News ...*, pp. 4–6, 1997.

[48] E. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1, pp. 100–115, 1954.

[49] W. Snyder and D. Jameson, "Multivariate geographic variation of mating call in populations of the Pacific tree frog (Hyla regilla)," *Copeia*, vol. 1965, no. 2, pp. 129–142, 1965.

[50] T. W. Friedl and G. M. Klump, "Sexual selection in the lek-breeding European treefrog: body size, chorus attendance, random mating and good genes," *Animal Behaviour*, vol. 70, pp. 1141–1154, Nov. 2005.

[51] D. K. Dawson and M. G. Efford, "Bird population density estimated from acoustic signals," *Journal of Applied Ecology*, vol. 46, pp. 1201–1209, Nov. 2009.

[52] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[53] A. Oppenheim and R. Schafer, *Discrete-time signal processing*. Prentice-Hall signal processing series, Prentice Hall, 2010.

[54] M. Wikelski, R. W. Kays, N. J. Kasdin, K. Thorup, J. A. Smith, and G. W. Swenson, "Going wild: what a global small-animal tracking system could do for experimental biologists," *Journal of Experimental Biology*, vol. 210, no. 2, pp. 181–186, 2007.