UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

RECORDING AND AUTOMATIC DETECTION OF AFRICAN ELEPHANT (*LOXODONTA AFRICANA*) INFRASONIC RUMBLES

by

Petrus Jacobus Venter

Submitted in partial fulfillment of the requirements of the degree
Master of Engineering (Bio-Engineering)
in the
Faculty of Engineering, Built Environment and Information Technology
UNIVERSITY OF PRETORIA

April 2008

SUMMARY

Recording and automatic detection of African elephant (*Loxodonta africana*) infrasonic rumbles

by

Petrus Jacobus Venter

Supervisor	:	Prof JJ Hanekom
Department	:	Electrical, Electronic and Computer Engineering
Degree	:	Master of Engineering (Bio-Engineering)

SUMMARY

The value of studying elephant vocalizations lies in the abundant information that can be retrieved from it. Recordings of elephant rumbles can be used by researchers to determine the size and composition of the herd, the sexual state, as well as the emotional condition of an elephant. It is a difficult task for researchers to obtain large volumes of continuous recordings of elephant vocalizations. Recordings are normally analysed manually to identify the location of rumbles via the tedious and time consuming methods of sped up listening and the visual evaluation of spectrograms. The application of speech processing on elephant vocalizations is a highly unexploited resource. The aim of this study was to contribute to the current body of knowledge and resources of elephant research by developing a tool for recording high volumes of continuous acoustic data in harsh natural conditions as well as examining the possibilities of applying human speech processing techniques to elephant rumbles to achieve automatic detection of these rumbles in recordings. The recording tool was designed and implemented as an elephant recording collar that has an onboard data storage capacity of 128 gigabytes, enough memory to record sound data continuously for a period of nine months. Data is stored in the wave file format and the device has the ability to navigate and control the FAT32 file system so that the files can be read and downloaded to a personal computer. The collar also has the ability to stamp sound files with the time and date, ambient temperature and GPS coordinates. Several different options for microphone placement and protection have been tested experimentally to find an acceptable solution. A relevant voice activity detection algorithm was chosen as a base for the automatic detection of infrasonic elephant rumbles. The chosen algorithm is based on a robust pitch determination algorithm that has been experimentally verified to function correctly under a signal-to-noise ratio as low as -8 dB when more than four harmonic structures exist in a sound. The algorithm was modified to be used for elephant rumbles and was tested with previously recorded elephant vocalization data. The results obtained suggest that the algorithm can accurately detect elephant rumbles from recordings. The number of false alarms and undetected calls increase when recordings are contaminated with unwanted noise that contains harmonic structures or when the harmonic nature of a rumble is lost. Data obtained from the recording collar is less prone to being contaminated than far field recordings and the automatic detection algorithm should provide an accurate tool for detecting any rumbles that appear in the recordings.

KEY WORDS

elephant vocalizations, voice activity detection, infrasonic rumbles, bio acoustics, speech processing, pitch determination

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

OPSOMMING

Opname en outomatiese deteksie van Afrika Olifante (*Loxodonta Africana*) infrasoniese klanke

deur

Petrus Jacobus Venter

Studieleier	:	Prof JJ Hanekom
Departement	:	Elektriese, Elektroniese en Rekenaar-Ingenieurswese
Graad	:	Meester van Ingenieurswese (Bio-Ingenieurswese)

OPSOMMING

Die waarde van die studie van olifantvokalisasies lê in die magdom inligting wat daaruit onttrek kan word. Opnames van infrasoniese olifantklanke kan deur navorsers gebruik word om die grootte en samestelling van 'n trop, die seksuele toestand van die olifant asook die emosionele staat van 'n olifant te bepaal. Dit is 'n moeilike taak vir navorsers om groot volumes aaneenlopende opnames van olifantvokalisasies te bekom. Die vokalisasies in opnames word normaalweg deur die vermoeiende en tydrowende metodes van versnelde luister en visuele evaluasie van spektrogramme geïdentifiseer. Die toepassing van spraakverwerkingstegnieke op olifantvokalisasies is 'n baie onderbenutte hulpbron. Die doel van hierdie studie was om by te dra tot die bestaande kennis en hulpbronne van olifantnavorsing deur 'n instrument daar te stel wat hoë volumes aaneenlopende akoestiese data in ongunstige omstandighede kan opneem asook om die moontlikheid te ondersoek om spraakverwerkingstegnieke aan te wend om outomatiese deteksie van olifantklanke in genoemde opnames te bewerkstellig. Die apparaat wat die opnames maak is ontwerp en geïmplementeer as 'n olifanthalsband met 128 gigagrepe geheue aanboord wat genoeg geheue is om nege maande se aaneenlopende opnames te stoor. Data word in die "wave"-formaat gestoor en die toestel het die vermoë om die FAT32 lêerstelsel te beheer en daardeur te navigeer sodat die klanklêers na 'n rekenaar toe afgelaai kan word. Die halsband kan ook die klanklêers merk met die datum en die tyd, die buitetemperatuur en "GPS"-koördinate. Verskeie opsies vir die plasing en beskerming van die mikrofoon is eksperimenteel getoets om sodoende 'n aanvaarbare oplossing te kry. 'n Relevante stemaktiwiteitsdeteksie algoritme is gekies as 'n basis vir die ontwikkeling van 'n outomatiese olifantklank deteksie algoritme. Die gekose algoritme is gebasseer op 'n robuuste toonhoogtedeterminasie algoritme wat eksperimenteel geverifïeer is om te werk teen 'n sein tot ruis verhouding van tot so laag as -8 dB mits 4 of meer harmonieke in klank teenwoordig is. Die algoritme is aangepas sodat dit vir olifantklanke gebruik kan word en is getoets met 'n stel vooraf opgeneemde olifantvokaliseringsdata. Volgens die resultate wat verkry is wil dit voorkom of die algoritme infrasoniese olifantklanke akkuraat kan opspoor. Die hoeveelheid vals alarms en onopgespoorde klanke neem toe as die opnames ongewensde geraas met harmonieke bevat en ook wanneer die botone van die klanke verlore raak in swak gehalte opnames. Data afkomstig van die olifanthalsband is minder vatbaar vir ongewensde geraas as vêrveld opnames en die outomatiese olifantklank deteksie algoritme behoort 'n akkurate stuk gereedskap te wees vir die opsporing van infrasoniese olifantklanke wat in opnames voorkom.

SLEUTELWOORDE

olifantklanke, stemaktiwiteitsdeteksie, infrasoniese klanke, bioakoestiek, spraakverwerking, toonhoogte bepaling

Contents

List of abbreviations

Chapter 1

INTRODUCTION

1.1 PROBLEM DEFINITION

The study of elephant vocalizations is an important part of elephant research (Garstang, 2004; Langbauer Jr, 2000). One type of elephant vocalization, the infrasonic rumble, is especially important because of the abundance of information that it contains. Recordings of elephant rumbles can be used by researchers to determine the size and composition of the herd, the sexual state of an elephant as well as the emotional condition of an elephant (Clemins, Johnson, Leong and Savage, 2005; Garstang, 2004; McComb, Reby, Baker, Moss and Sayialel, 2003; Langbauer Jr, 2000).

Some of the practical issues that researchers are faced with include the difficulty of acquiring high volumes of continuous high quality acoustic data in adverse environments as well as the tedious and time consuming task of identifying elephant rumbles within recordings (Clemins *et al.*, 2005; Langbauer Jr, 2000). The identification of rumbles in a recording is done manually by experts who inspect spectrograms and listen to sped-up versions of the recordings (Poole 1999). If, however, high volumes of data were to become available, research time spent on manual detection of rumbles would be hard to justify.

This dissertation investigates methods to assist elephant researchers by applying engineering principles to develop research tools to solve these problems. The aim of this study is to provide solutions to both the problems that have been identified. Firstly, an instrument that can record high volumes of continuous acoustic data in unfavourable circumstances needed to be developed. This included the design of a robust but powerful electronic system that has very low power consumption and high volumes of permanent memory resources as well as a sound mechanical design that ensures that the instrument will survive for the required period of time. Secondly, an algorithm to automatically detect infrasonic rumbles from recordings needed to be developed. Speech processing techniques that can automatically detect the presence of human speech in a recording were modified to achieve the automatic detection of these elephant rumbles.

1.2 APPROACH

Obtaining high quality, continuous recordings of elephant vocalizations in the wild is a difficult but necessary task. One way in which recordings have been made in previous work is via a RF (Radio Frequency) transmitting collar which contains a microphone and a RF transmitter (Clemins *et al.*, 2005; Clemins and Johnson, 2003). The sound received by the microphone is then modulated onto the RF signal and transmitted to a receiver station. This method works well for recording vocalizations from captive elephants. In the wild, where elephants can move hundreds of kilometres in a few days, this method is unreliable. If the elephant moves too far away from the receiver the RF signal will be lost. Also, even if the elephant that wears the collar does not move very far from the receiver station, the topography of the area may result in degradation or loss of the RF signal. This can typically occur if the elephant walks into a deep valley and could result in sporadic loss of data as the elephants move through their territory.

An alternative would be to design a recording collar that stores the data on the collar itself. The concept of a recording collar that stores data onboard is attractive because the unreliable RF link is not used. Except for the problems that could occur when using an RF signal as was discussed above, another potential problem that may occur with a collar is clogging of the microphone with dirt. Elephants continuously cover their bodies with mud and dust. If the microphone gets obstructed it could cause loss of sound input to the system. The immense size and strength of an elephant, as well as their fondness of pushing trees out of the ground (Ogada, Gadd, Ostfeld, Young and Keesing, 2008; Sankaran, Ratnam and Hanan, 2008), mean that the collar needs to be immune to intense physical abuse. Elephants also swim, so the collar needs to be completely watertight. It should also be remembered that wild elephants need to be tranquilized in order to mount a collar. This is an expensive procedure which needs approval from an ethics committee and is a stressful experience for the elephants. The collar should be able to store data for periods of time long enough to justify tranquilising of elephants.

Another vocalization recording method that has been used in the past is a hand-held recorder (McComb *et al.*, 2003; Wood, McCowan, Langbauer Jr, Viljoen and Hart, 2005). A specific herd of elephants was followed and recorded from a safe distance. The person who did the recording was notified of the herd's position each morning via a GPS (Global Positioning System) collar worn by the matriarch of the herd (Wood *et al.*, 2005). He then drove to the site indicated by the GPS coordinates after which he tracked the exact position of the elephants using an RF locator (also worn in a collar by an elephant in the herd).

The use of hand held microphones have several clear disadvantages. Firstly, hand-held recordings are not continuous because the person that operates the recorder needs to stop the recording every time that the herd moves along and then start recording from a new position. Secondly, these recordings are prone to noise since the microphone is located relatively far from the source of the elephant noises. In addition, this method requires the constant presence of a researcher in the field who conducts the recordings, and a GPS collar and RF collar need to be mounted on two of the elephants in the herd for them to be located.

The benefits of using an elephant recording collar therefore include the possibility of continuous recordings, a shorter distance between the source of the vocalizations and the microphone, as well as a reduction in unwanted sounds, as reported with far field hand-held recordings. The main choices that have to be made regarding the design of the collar are the kind of data storage device to use, how to ensure that the device is mechanically robust and watertight and how to mount the microphone so as to prevent damage or blockage without compromising sensitivity. The details of the design and motivations for design choices will be discussed in Chapter 3.

It will be shown in Chapter 2 that sound production of elephants resemble those of humans and that this justifies the use of speech processing techniques used in human voice detection on elephant vocalizations. The problem of detecting elephant rumbles in recordings is similar to that of detecting human speech in recordings. Thus, identification of a suitable VAD (Voice Activity Detector) from this literature may assist in solving this problem.

VAD techniques that are usually used in a telecommunication context are not very robust (Wu, Wang and Brown, 2003). These techniques mostly use the energy level in a signal to decide whether a certain segment of sound was speech or non-speech since speech is normally the only source of energy in the signal. VAD of speech in noisy circumstances are more complex and some of the characteristics of speech need to be exploited in order to distinguish it from other high energy components present in the signal (Wu *et al.*, 2003). The harmonic nature of speech is one such characteristic that may be exploited by using a pitch detection algorithm. There are three basic types of pitch detection algorithms in use, namely frequency domain, time domain and time-frequency domain algorithms.

Just like human speech, infrasonic elephant rumbles also have a harmonic nature but in a lower frequency range (Clemins *et al.*, 2005; Clemins and Johnson, 2003; Langbauer Jr, Payne, Charif and Thomas, 1989; Langbauer Jr, 2000; McComb *et al.*, 2003). The elephant vocalizations itself will not be the only energetic components of a recorded signal. Various noises that occur in the wild will also be recorded, including birds, footsteps, breaking of branches and wind. Therefore, the simple energy threshold VAD technique will probably not suffice. The approach that has been taken for the automatic detection of the elephant rumbles is based on a robust time-frequency domain pitch determination algorithm which has been used specifically for VAD in personal audio recordings which also contains various kinds of noise. The details of the development of the automatic rumble detector will be discussed in Chapter 3.

1.3 HYPOTHESIS AND RESEARCH QUESTIONS

The hypothesis of this study is that speech processing techniques used for human voice can be adapted effectively to detect infrasonic elephant rumbles from noisy recordings.

All mammals produce sounds in a similar way. Specifically, the fact that elephants, just like humans, use vocal cords to produce vocalizations suggests that techniques used to detect human speech in sound data can be modified to detect elephant rumbles from recordings. A suitable technique needs to be identified from the wide range of VAD algorithms available in the literature. One should be able to modify the chosen technique so that it can detect elephant rumbles.

To test this hypothesis, the development of a tool that can record substantial amounts of data is needed. It becomes part of the problem addressed in this study to develop such a tool. With the swift advances in flash card technology it should be possible to develop a physically robust recording collar that can store vast amounts of high quality acoustic data safely onboard the device. The equally impressive advances in low power electronics should ensure that the device can be powered for a long enough period of time from a relatively small and lightweight battery.

Specifically, the research questions investigated in this dissertation are:

1. What are the components needed for developing a recording tool that can withstand the harsh conditions on an elephant for months at a time?
2. What is the optimal way of mounting the microphone so that sensitivity is maximized, an even pass band is maintained and the chances of microphone blockage or damage is minimized while ensuring that the memory cards can be retrieved after a recording session?
3. What is a suitable speech processing technique that can be used as a basis for automatically detecting infrasonic elephant rumbles?
4. What modifications should be made to the chosen speech processing technique so that it may be applicable to elephant rumbles?
5. What is the accuracy with which the algorithm can detect rumbles and under which circumstances will it fail?

1.4 OBJECTIVES

The primary objective of this study is fourfold. Firstly, a tool that can record high volumes of continuous high quality acoustic data had to be developed. Secondly, this instrument needed to be tested on actual elephants to evaluate its performance. Subsequently, an algorithm needed to be developed that can be used to automatically detect elephant rumbles within recordings. The last goal of the primary objective was to evaluate the algorithms developed and identify circumstances under which they would fail.

The primary objective mentioned above was achieved by completing the following steps:

1. Do a concept design of the electronic system
2. Focus on low power usage and robustness
3. Test each sub unit individually
4. Do a final electronic design integrating all sub units
5. Implement the electronic design in hardware
6. Mould the electronics hardware unit into a collar
7. Ensure that weight remains under the maximum level
8. Run tests to find an acceptable way in which to mount the microphone
9. Test correct operation under a range of temperatures
10. Test the collar on an elephant for extended periods of time
11. Identify a suitable VAD technique for use in this application
12. Adapt the chosen technique to detect elephant rumbles instead of human voice
13. Test the level of success of the algorithm using actual recordings from an elephant collar
14. Identify the conditions under which the detector will not produce correct results

The blocks enclosed by the dotted line in Figure 1.1 shows where the work done in this study fits in. The long term objective of this study is to provide a way of recording high volumes of elephant vocalizations and finding a way to automatically detect elephant rumbles within these recordings to enable researchers to do further signal processing on these isolated rumbles.

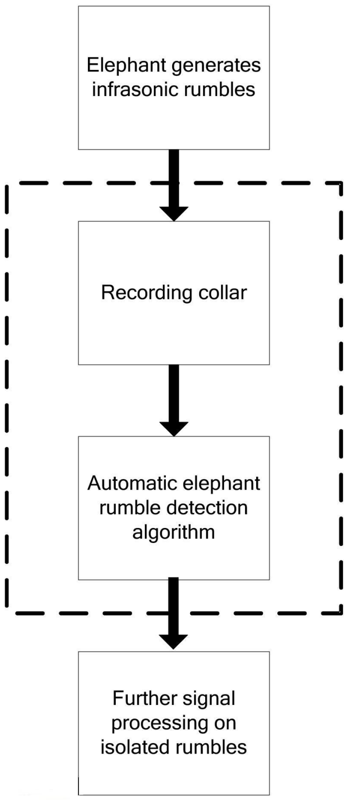

Figure 1.1: A diagram indicating where the current study fits into acoustic elephant research.

1.5 OVERVIEW OF THE STUDY

In the chapters that follow, the details of the development processes for the hardware tools as well as the development of the elephant rumble detection algorithm are discussed. An overview of the relevant literature is given in Chapter 2. This literature study motivates the importance of research on elephant rumbles and describes some of the methods currently used to obtain recordings of elephant vocalizations and the way in which these calls are isolated. The one known instance where speech processing has been applied to elephant vocalizations is also discussed. A short overview of different VAD techniques are given and the choice of technique for the current study is motivated from the literature.

Chapter 3 describes the methods used for the development of the research tools. The design and implementation aspect of the elephant recording collar are explained. The combination of electronic, mechanical, acoustic and ergonomical factors that need to be addressed are discussed in sequence. The electronic design is discussed and motivated after which the software that controls the instrument is explained on a functional level. The mechanical design of the collar is explained. The automatic detection of infrasonic elephant rumbles will then be considered. The details of every processing step of the detection technique are explained and the modifications that were necessary to apply the techniques to elephants are described.

The results from the elephant recording collar and the automatic elephant rumble detection (and pitch determination) algorithm are given in Chapter 4. The results of two field tests that were done with the elephant collar are discussed and the results of optimum microphone placement experiments are given. Further experiments that verify correct operation under certain environmental circumstances are presented. The performance of the automatic elephant rumble detection algorithm is evaluated and situations where the algorithm might not function correctly are identified.

The relevance of this study in context of the current literature is discussed in Chapter 5. The findings and implications of the different aspects in the study are reported and discussed. The contribution of this study to the current body of literature is pointed out. Chapter 6 states the conclusion that is drawn from the results that have been observed in this study. A summary of the work that has been done is given. Possible improvements to both the software and hardware that were developed in this study are proposed and recommendations for further research are made.

Chapter 2

LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

In the outline of the study provided in the previous chapter, the research problem was defined as the development of an elephant recording collar, as well as methods of achieving automatic elephant rumble detection by adapting existing speech processing techniques. This chapter aims to give a background and critical discussion of the relevance of elephant vocalizations (Section 2.3) and discuss the history of elephant vocalization research as well as speech processing methods used to process and analyse vocalization recordings (Section 2.4). The wide range of speech processing techniques for VAD and the motivation for the chosen algorithm will be discussed in Section 2.5.

2.2 INTRODUCTION

Despite the fact that the vocalizations of African elephants have been extensively researched, the use of speech processing as an analysis tool for elephant sounds has not been widely explored or documented. The nature of this study necessitates a comprehensive overview of existing knowledge on both elephant vocalization research and certain speech processing techniques, and these aspects will subsequently be expounded in the following sections.

2.3 ELEPHANT VOCALIZATIONS

Elephants are highly social animals and have a complex social structure in which both long distance and short distance communication plays an important role (Langbauer Jr, 2000). They communicate using taste, smell, touch, sound and visual signs. As an example, an elephant bull can go into a state called musth where the testosterone levels rise, causing increased aggression and dominance. The bull will communicate this state to other elephants using specific odours, bodily positions and vocalizations (Poole and Moss, 1981; Rasmussen, 1988). Vocalizations are easier to observe over long distances than the other methods of communication and can be used by researchers to gain information about individual elephants as well as information about the group.

The best known elephant vocalizations are low frequency rumbles or higher frequency trumpets (McComb *et al.*, 2003), but researchers agree that elephants can produce at least 10 different sound types (Clemins and Johnson, 2003; Leong, Ortolani, Burks, Mellen and Savage, 2002; Soltis, Leong and Savage, 2005b). Most vocalizations are produced in the form of infrasonic rumbles which are too low in pitch to be easily perceived by humans. These infrasonic rumbles have a fundamental frequency of between 15 and 25 Hz and harmonics ranging several hundred Hz (Langbauer Jr, 2000). The harmonic frequency at the approximate level of 125 Hz has been shown to be the most important frequency needed for an elephant in the group to correctly establish the identity of the caller (Langbauer Jr, 2000). Figure 2.1 shows a spectrogram of a typical elephant rumble. It can be seen from the figure that almost all the spectral energy of the rumble occurs below 250 Hz. The harmonic nature of elephant rumbles can also be seen in the spectrogram as the yellow stripes that indicate higher spectral energy at multiples of the fundamental frequency. The harmonics of the elephant rumble are indicated by black lines.

Infrasonic elephant rumbles are an effective way of communicating over long distances. Calls can have a sound intensity of 117 dB SPL (Sound Pressure Level) at a distance of one metre (Langbauer Jr, 2000) with a reference sound pressure of 20 uPa. Low frequency rumbles are much less vulnerable to degradation due to the effects of deflection, refraction and absorption. This is due to the fact that low frequency sounds have long wavelengths which allow them to travel past objects relatively smaller than the wavelength itself (Hartmann, 1998). This implicates that these subsonic vocalizations are to a large extent immune to degradation and can travel distances far

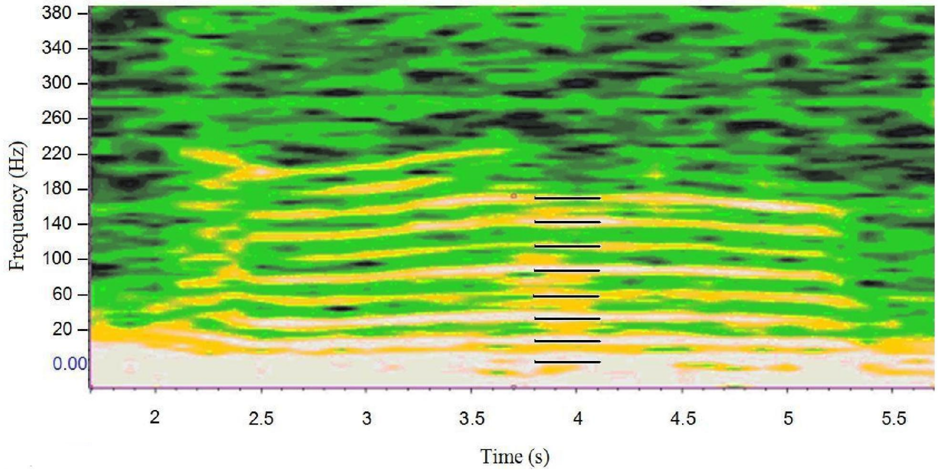

Figure 2.1: A spectrogram of a typical elephant rumble. The harmonics present in the rumble are indicated by black stripes.

exceeding sounds that consist of higher frequencies (like human speech). Experiments have shown that these rumbles can be recognized by elephants separated by a distance of 4 kilometres (Langbauer Jr, Payne, Charif, Rapaport and Osborn, 1991). Other reports have suggested that subsonic communication could even take place over distances of 10 kilometres if optimum conditions exist (Larom, Garstang, Payne, Raspet and Lindeque, 1997). As can be expected the sensitivity of hearing at low frequencies is very well developed in elephants. In fact, elephants have the best hearing at low frequencies of any mammal ever tested (up to 100 times better than humans) (Heffner and Heffner, 1982).

The importance of vocalizations to researchers in the field of elephant behaviour is largely due to the abundance of information that can be retrieved from it (Garstang, 2004; Langbauer Jr *et al.*, 1989; Langbauer Jr, 2000; McComb *et al.*, 2003; O'Connell-Rodwell, Arnason and Hart, 2000; Poole, Tyack, Stoeger-Horwath and Watwood, 2005; Wood *et al.*, 2005). The rate at which vocalizations are observed from an unseen group of elephants can be used to determine the size of a group as well as the number of males, females and calves in the group (Payne, Thompson and Kramer, 2003). Each elephant has specific voice characteristics which means that individuals may be recognized by their vocalizations (Clemins *et al.*, 2005; Clemins and Johnson, 2003; Soltis *et al.*, 2005b). Information about the sexual state of individual elephants can also be determined by analyzing their rumbles (Leong, Burks, Rizkalla and Savage, 2005; Poole, 1999; Soltis, Leong and Savage, 2005a). As is the case with humans, some parameters of vocalizations can be used to determine the emotional state of an elephant (Clemins *et al.*, 2005; Soltis *et al.*, 2005b).

A large proportion of past research on elephant vocalizations involves the analysis of a collection of recordings. Two methods for recording elephant vocalizations have been identified in the literature. The first recording method entails the use of an RF transmitting collar, as reported by several researchers (Clemins *et al.*, 2005; Leighty, Soltis, Leong and Savage, 2008; Leong, Ortolani, Graham and Savage, 2003; Leong *et al.*, 2002; Soltis *et al.*, 2005a). These collars were all built by Walt Disney World Company Instrumentation Support (Division of Ride and Show Engineering) and were based on an earlier design by William Langbauer, Jr and Steven Powell. The collars consist of a condenser microphone, a radio transmitter and a battery pack. The sound picked up by the microphone gets transmitted to a radio receiver station where the sounds are recorded on a data tape. The recordings were done on captive elephants in Disney's Animal Kingdom. On days that recording sessions were scheduled the collars were mounted in the mornings. At a specific time of the day, the data tapes were activated to initialize a recording session with a duration of one hour. The collars were removed in the evenings so that the batteries could be recharged. This method appears to be effective for recording vocalizations from captive elephants since the elephants cannot move out of range of the receiver station and the batteries can be recharged daily. If a problem develops, for example if the microphone gets clogged with mud, the problem can be solved overnight and no more than a one hour recording session will be lost. The implementation of RF collars for recording wild elephant vocalizations could pose a number of problems. Firstly, wild elephants can travel vast distances in a single day. Elephants have been observed to walk up to 38 km in a 24-hour period (Viljoen and Bothma, 1990). Elephants that wear RF collars could walk out of range of the receiver station resulting in loss of data. To expand the range of an RF collar, a more powerful transmitter would have to be used, resulting in higher power consumption. In addition, wild elephants need to be tranquillized before a collar can be fitted. This is a costly procedure and requires ethical clearance. Therefore, the batteries that power the recording system should have a much greater capacity than those used for captive elephants, since frequent recharging of the batteries, requiring removal and refitting of the collar, is not a viable option in wild elephants.

The second method that has previously been used for elephant recordings is hand-held recordings. This method has been used to obtain vocalizations from wild elephants in the Kruger National Park as reported by Wood *et al.* (2005). These researchers tracked a single herd of elephants in order to obtain regular recordings of their vocalizations. One of the female elephants in the herd was fitted with an GPS collar that sent the location of the herd to a cellular phone every morning. The researcher responsible for the recordings drove as close to the reported location as the road would permit and then proceeded on foot. A radio locator was used to find the exact position of the elephants, and a sound recorder with an external dynamic microphone was placed as close to the elephants as was safely possible. The movements of the herd had to be followed and recordings were stopped and restarted in the new location every time the herd moved (J.J. Viljoen, personal communication, September 7, 2005). The recorder saved the sound in the wave file format on a CompactFlash memory card. Under good recording conditions, the combined duration of all the recordings made in a day amounted to approximately an hour and a half. On extremely windy or rainy days no recordings could be made.

Besides the intensive field-work required by this method, an additional factor reduces the efficacy of this method. Because of the fact that the recording device needs to be adjusted to a high sensitivity to clearly record elephant vocalizations from a substantial distance, unwanted noises from sources much closer to the microphone can contaminate the recordings. This method of recording has not been successful in obtaining high quality, continuous recordings of elephant vocalizations in the wild.

For both of these recording methods (RF transmitter as well as hand-held recordings), the elephant vocalizations within recordings have to be identified by experts who manually examine spectrogram plots and listen to sped up versions of the recordings (Leong *et al.*, 2002; Poole, Payne, Langbauer Jr and Moss, 1988). When a recording is played at multiple speeds of its normal rate the inaudible low frequency elephant sounds get shifted up into the audible range of frequencies. This is a time consuming procedure, especially if long duration recordings have to be analysed. In addition, the recordings are often contaminated with noise in the infrasonic bandwidth range (Leong *et al.*, 2002), which makes the analysis of these recordings even more complex and time consuming.

In conclusion it can be said that elephants are social animals that communicate with one another in various ways, including vocalizations. Infrasonic vocalization is a very effective way of communicating over both short and long distances. The wealth of information present in vocalizations as well as the fact that vocalizations are easier to observe than other means of elephant communication render infrasonic vocalization a valuable and useful tool in the study of elephant behaviour. However, previously documented methods used to record these vocalizations have been found to have several limitations, and novel methods should be explored in this regard.

2.4 ELEPHANT VOCALIZATIONS AND SPEECH PROCESSING

In view of the limitations of manual analysis of elephant vocalization recordings, the possibility of using speech processing techniques on these recordings should be considered. Infrasonic elephant rumbles are produced by vocal cords (Garstang, 2004; McComb *et al.*, 2003; Soltis *et al.*, 2005b) so that it may be expected that the resulting sound would have characteristics similar to voiced human speech. In fact, studies have shown that most mammalian vocal production and reception systems are extremely similar (Bradbury and Vehrencamp, 1998; Titze, 1994). This idea is supported by the harmonic nature of the elephant rumbles and justifies the application of speech processing techniques to bioacoustics.

The only known scientific publication where speech processing techniques were used on elephant vocalizations is that of Clemins *et al.* (2005) and Clemins and Johnson (2003). Automatic classification and speaker identification were conducted on a collection of vocalizations and rendered promising results. These vocalizations were recorded from captive elephants by the RF elephant collar discussed previously in this chapter. The classification of elephant vocalizations was similar to speech recognition done on human speech. Five basic elephant vocalizations were classified in the experiment. This was achieved with by using 12 Mel-Frequency Cepstral Coefficients as features and computing log energy by using a shifted filter bank to compensate for the infrasonic range in which elephant vocalizations occur. For the speaker recognition experiments, Hidden Markov Models (HMMs) were used for the modelling of the different speakers. The success rate of the call type classification experiment was 83.8% and that of the speaker recognition was 88.1%. It was noted that in most bioacoustics studies, vocalizations are divided into groups of varying quality, and ultimately only the categories with the best quality recordings are used. In the study reported by Clemins *et al.* (2005), however, the number of recorded vocalizations available were too small to allow exclusion of low quality recordings and these were therefore included in the analyses.

The shortage of a sufficient number of vocalizations in the study of Clemins *et al.* (2005) indicates the need for an elephant vocalization recording system that can gather a large amount of continuous acoustic data. A further limitation in the method of these researchers was the need for manual location and isolation of the vocalizations from the recordings. This is a time consuming process, especially for studies where much larger numbers of vocalizations are needed. These limitations underscore the need for the automatic detection of infrasonic elephant vocalizations.

2.5 VAD TECHNIQUES

The challenge of detecting elephant rumbles from recordings appears to be similar to that of identifying voiced human speech from personal audio recordings. Specifically it suggests that the techniques used for Voice Activity Detection (VAD) of human speech may also be suitable for elephant rumble detection. Because of the periodic nature of voiced speech, one way of detecting the presence of speech during a particular interval may be to use a pitch detection (or determination) algorithm. In order to achieve automatic detection of infrasonic elephant rumbles, techniques should be used that capitalize on the specific characteristics of these rumbles that set them apart from background noise, such as the harmonic content of the rumbles.

Most existing applications of pitch detection algorithms used in voice activity detectors as published in literature are limited to clean speech (noiseless speech) in a telecommunications context (Wu *et al.*, 2003). It is, however, more difficult to extract pitch from a recording where numerous other sources of noise are present, as would typically be the case for the elephant call recordings. Elephant recordings typically contain unwanted sounds including bird calls, motor vehicles, wind, walking elephants and other sounds that occur in the wild. A noise robust pitch detection algorithm will be needed to effectively detect periodicity in the noisy recordings.

Pitch detection algorithms are generally classified into three categories namely time-domain (Hung, 2002; Kunieda, Shimamura and Suzuki, 2000; Shimamura and Kobayashi, 2001; Takagi, Seiyama and Miyasaka, 2000), frequency-domain (Davis, Nordholm and Togneri, 2006; Li, Zhang, Cui and Tang, 2005; Woo, Yang, Park and Lee, 2000; Zhang, Zhang, Lin and Quan, 2006) and time-frequency domain algorithms (Wu *et al.*, 2003; Zhao and Ogunfunmi, 1999). Time-domain pitch detection algorithms consider the temporal structure of the waveform. Peak and valley positions, zero-crossings and autocorrelations are used for detecting the pitch period. The simplest and computationally most inexpensive technique for determining pitch would be a simple count of the number of times that the signal crosses the zero reference. This technique is highly inaccurate when the signal contains noise or in the case of a harmonic signal when the fundamental frequency is less energetic than any of the higher harmonics. For this reason, such a technique will not be suitable for pitch detection of elephant vocalizations.

The purpose of autocorrelation routines used as part of time-domain pitch detection is to find the similarity between a signal and a shifted version of the same signal. This is based on the premise that a periodic signal will thus have a periodic autocorrelation function, and a harmonic signal will have an autocorrelation function with peaks at multiples of the fundamental frequency. This technique works well with lower frequencies and is popular in speech processing techniques where the pitch range is limited. The ability of the technique to extract the pitch of a sound with a harmonic structure is attractive, but the addition of noise to a signal degrades the definition of the peaks of the autocorrelation function and diminishes the accuracy of the technique.

Frequency domain pitch detection algorithms typically detect the fundamental frequency by examining the harmonic structure in the short-term spectrum. The fundamental frequency can be determined by computing the greatest common divisor of the frequencies of the higher harmonic components. The greatest common divisor is determined by filling in a frequency histogram for each harmonic frequency and at integer divisions of the harmonic frequency. The greatest frequency peak of the histogram represents the greatest common divisor, and thus the fundamental frequency. Even though this technique is computationally inexpensive, the addition of narrow band noise to the signal or the evaluation of a sound with a changing number of harmonics degrades its performance. The cepstrum, a second order transform of the power spectrum, is also widely used for pitch determination (Ahmadi and Spanias, 1999; Kim and Chung, 2004; Nadeu, Pascual and Hernando, 1991; Noll, 1967; Seiyama, Tohru, Tetsuo and Eiichi, 1992; Zhao and Ogunfunmi, 1999). The term “cepstrum” is formed by reversing the first four letters of "spectrum". The Fourier transform of a signal is taken to the log-magnitude Fourier spectrum. By implication, if the original spectrum comes from a harmonic signal, the frequency representation will be periodic, so when the FFT (Fast Fourier Transform) is taken again it will result in a peak corresponding to the fundamental frequency. The cepstrum technique can also be seen as a de-convolution procedure. If the original signal is seen as an impulse train that has been convolved with a filter, it results in multiplication in the frequency domain. Applying the log operation translates the multiplication to an addition operation. When the FFT is then applied once more it results in the de-convolution of the original signal which gives us the fundamental frequency. This technique is used with great success for determining the pitch of noiseless speech, but once again, when noise is added to the original signal, the peak indicating the fundamental frequency fades away.

An additional technique used as part of frequency domain pitch detection algorithms is the use of statistical properties of the communication channel and the expected speech signal (Chang, Kim and Mitra, 2006; Chang, Shin and Kim, 2004; Davis *et al.*, 2006; Sohn, 1999). The success of these pitch detection techniques are to a great extent dependant on a priori knowledge of the statistical behaviour of the communication channel as well as the expected speech signal. Often, these systems need to be trained prior to operation. For example, it might be required that a number of both male and female participants repeat certain words in different tones of voice to acquire the statistical models needed for correct operation of the algorithm. The fact that the elephant vocalizations will be recorded in free field and that system training data is not easily obtainable renders this technique unsuitable for use with elephant vocalizations.

Time-frequency domain algorithms first filter the original signal into sub-bands and then perform time-domain analysis on the band-filtered signals (Wu *et al.*, 2003). By filtering sound into sub-bands this technique resembles the way that humans perceive pitch. The pitch of complex harmonic and inharmonic signals can be identified correctly, and is robust in the face of noise and phase changes. Although this technique is computationally expensive, its characteristics makes this type of pitch determination algorithm ideal for use in elephant recordings.

A robust pitch detection algorithm proposed by Wu *et al.* (2003) for use in human voice activity detection in various background noise conditions is an example of a time-frequency pitch detection algorithm. This technique will be discussed in more detail since it has already been implemented successfully to detect human speech from real life recordings with various background noises (Lee and Ellis, 2006). The reported success of this method despite the presence of background noise makes it an instinctive favourite to use for extracting elephant vocalizations in similar circumstances. The technique features the desired characteristics of previously discussed methods, but outperforms those techniques when a noisy signal is used.

The noise robustness of this algorithm is primarily due to the input signal being divided into a number of sub-bands and only the sub-bands with good signal to noise ratio being used in pitch determination. This perceptual pitch detector combines a cochlear model with a bank of autocorrelators. The algorithm uses the steps shown in Figure 2.2 for determining a pitch track.

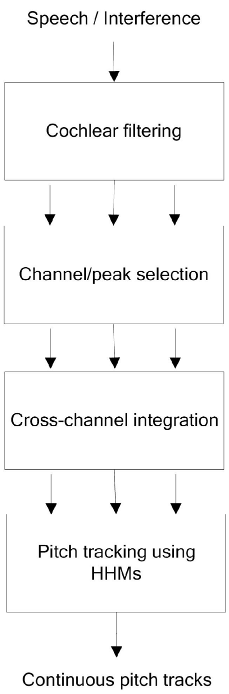

Figure 2.2: The schematic diagram of the model proposed in Wu *et al.* (2003).

The speech signal enters the system where it is firstly filtered into sub-bands by a cochlear type filter. This filter bank comprises an array of fourth order gammatone filters, which is a standard model for cochlear filtering. In fact, the particular sensitivity of the human cochlea to pitch was used as a template for the development of this filter.

A normalized autocorrelation of each of the resulting channels is then calculated. Noisy channels are discarded and the remaining channels are integrated and used to find an estimation of the pitch present in the original signal. An HMM is subsequently used to form continuous pitch tracks. Some of the same principles used in this algorithm were used as a starting point for the automatic detection of infrasonic elephant rumbles, as will be described in Chapter 3.

2.6 SUMMARY

Chapter 2 presents the literature on which this study is based. The importance of vocalizations and specifically infrasonic rumbles to elephant research were highlighted. Existing methods for recording elephant vocalizations were evaluated. The critical discussion of existing literature demonstrated the potential value of applying speech processing techniques to elephant vocalizations. In addition, the limited application of this technique underscores the need for research that will explore the value thereof in the study of elephant vocalizations. In light of this need, the study at hand will aim to develop an automatic elephant rumble detector based on an existing VAD algorithm that has been successfully used to extract human speech from noisy recordings.

Chapter 3

METHODS

3.1 CHAPTER OBJECTIVES

This chapter discusses the methods that have been used to develop the elephant recording collar as well as the automatic elephant rumble detection and pitch tracking algorithm. The details of the electronic design of the collar will be discussed in Section 3.3. The mechanical design of the collar will be examined in Section 3.4 while the development of a rumble detection and pitch tracking algorithm follows in Section 3.5.

3.2 INTRODUCTION

One of the primary aims of this study was the development of a instrument that can record high volumes of continuous acoustic data in unfavourable circumstances. It has been decided that the recording device will be mounted within an elephant collar. The aim behind the development of the collar is to assist zoologists in obtaining large quantities of acoustic data from wild elephants that can be used in further research.

Before a design was made, thought had to be given to the specific mechanical and functional requirements that such a collar would have to satisfy. The requirements were identified by communication with elephant researchers and from scientific literature.

The electronic and mechanical designs were made with these requirements in mind.

The development of the automatic rumble and pitch tracking algorithm was done by first gaining insight into the nature of elephant rumbles by studying elephant vocalization recordings and then using the newly gained knowledge to devise a rumble detection algorithm based on speech detection techniques.

3.3 ELECTRONIC DESIGN

The requirements of the electronic design will be discussed first (3.3.1). Important specifications were determined from this section and were used to devise a concept design for the electronic system (3.3.2). A detailed discussion is given on the design considerations and choices that had to be made when implementing each of the elements shown in the concept design into a realistic electronic design (3.3.3). A functional description is given for each electronic module with a brief discussion of standard electronic design theory where applicable. The way in which the chosen components satisfy the desirable design characteristics of each module is discussed for every module. The operation of the CompactFlash (CF) memory cards is discussed in more detail as these devices are central to the design and has a standard interface that masks the operation of the internal flash technology.

The functional considerations regaring the printed circuit board are discussed (3.3.4). A functional description of the microcontroller software are given (3.3.5). A detailed discussion on the operation of the FAT32 file system is given since it plays a central role in the software design.

3.3.1 Design requirements

A collar that can record high volumes of continuous elephant acoustic data in unfavourable circumstances needed to be developed. The implication on the electronics of such an elephant collar will now be considered.

Firstly it is important to remember that a wild elephant would have to be tranquillized when the collar is fitted and again when the collar is removed. This is an expensive exercise that needs clearance from an ethics committee. Once a recording collar is fitted, it would be desirable that it keeps on recording continuously for a long enough period to make the experiment worthwhile. A minimum recording period of 90 days should ensure that enough data can be gathered.

The device should record the bandwidth of sound that an elephant produces. Elephant rumbles have frequency components as low 8 Hz while their trumpet sound has frequencies of up to 1.150 kHz. All the frequencies between 8 Hz and 1.150 kHz should be recorded. The sampling rate should, however, be kept as low as possible to ensure that memory space is not wasted. A sampling frequency of 3 kHz should be in order, giving an anti-aliasing filter enough spectral space to suppress any frequencies above the Nyquist frequency (1.5 kHz).

Sound should be recorded at a good enough resolution to ensure that its quality is acceptable for further processing. On the other hand, recording at a higher than required resolution would waste memory space. A resolution of 16 bits provides high quality recordings for human speech, so it would be a good choice for the recording of elephant sounds.

Enough memory storage should be provided within the collar to ensure that the collar can keep recording for the required period of time. A minimum capacity of 43 GB is required to record the sound. (This was calculated using Equation 3.7 given in Section 3.3.3.5). It should also be remembered that the collar should not cause discomfort to the elephant wearing it. This implies that the weight and size of the collar should be restricted.

If a recording device is going to be fitted on an elephant, the possibilities of recording extra information could be explored. Any information regarding the circumstances in which a sound recording is made might prove valuable. A GPS and a thermometer could be incorporated into the device to record the outside temperature and the location of the elephant.

Table 3.1 gives a summary of the most important requirements of the collar electronics. These requirements were considered before a specific electronic configuration were chosen.

Table 3.1: Specifications derived for the recording collar.

Parameter	**Consideration**	**Specifications**
Recording duration	The elephant needs to be tranquilized before fitting or taking off the collar	90 days (or more)
Sound bandwidth	Infrasonic rumbles as well as trumpet vocalizations should be recorded	8 – 1150 Hz (3000 Hz sampling)
Sound resolution	The recorded sound will need to be clearly defined in order to be processed	16 bits
Memory storage	There should be enough memory to last for the duration of the experiment and it should be physically robust	42 Gigabytes (or more) shock resistant memory
Power dissipation	The battery should last for the duration of the experiment	33 mA (or less)
Weight	The collar should not hinder nor injure the elephant	10 kg (or less)
Size	The collar should not obstruct the elephant's movements	25x25x30 cm (or smaller)
Added features	The more information that accompanies the sound files the better	GPS, and digital thermometer

3.3.2 Concept design

With the specifications shown in Table 3.1 in mind a concept design of the electronic system were done. The concept design is shown in Figure 3.1. A high level functional description of the concept design will now be given. The details of each of the elements will be given under the corresponding heading.

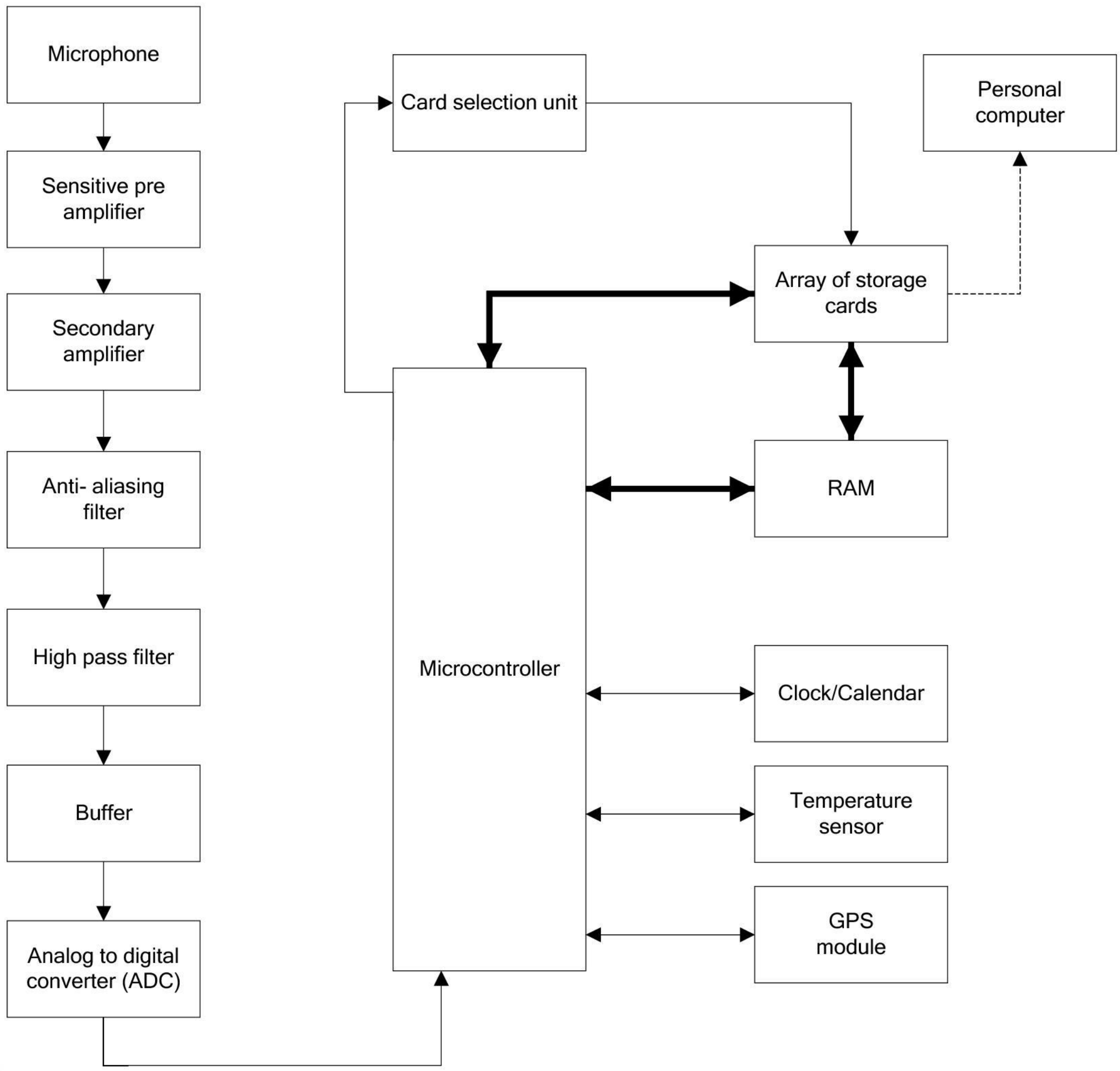

Figure 3.1: Hardware concept design block diagram.

A microphone converts sound waves from the surrounding area into an analogue electrical signal. The signal coming from the microphone needs to be amplified to a voltage signal with useful amplitude. This is done by using a sensitive preamplifier to effectively amplify the small signal directly at the output of the microphone. Another variable amplifier is then used to tune the gain to the desired value.

A sixth order Butterworth low pass filter was used as an anti-aliasing filter to filter out all frequency components higher than half of the Nyquist frequency. A high pass filter and a buffer were used to centre the filtered signal around 0 V for the bipolar ADC (Analogue to Digital Converter) and to provide a low input impedance to the analogue input of the ADC respectively. The ADC converts the analogue sound signal to a sixteen bit digital word. The timer module in the microcontroller generates a conversion pulse at a frequency of 3 kHz to the ADC to regulate its frequency of conversion. The microcontroller controls the flow of data from the ADC to the storage media and controls the timing and interaction of all other digital components in the system.

An array of solid state memory cards are used for data storage. The microcontroller controls the hardware responsible for selecting the memory cards one at a time so that data may be written to a single card until its capacity has been reached and then move on to the next card. A digital thermometer is used to measure the external temperature and a GPS module determines the positional coordinates as well as time and date information.

3.3.3 Detailed discussion

In order to realize the hardware concept design, every block of the block diagram in Figure 3.1 were designed and implemented in hardware. The hardware was designed to use as little power as possible and the idea was to design an appropriate power supply (set of batteries) only after the power consumption of the final design had been tested.

3.3.3.1 Microphone

The microphone that was used for this system should have the following characteristics:

1. A dynamic range of at least 16 Hz to 1.15 kHz (the frequency range of elephant vocalizations).
2. Rugged microphones should be chosen that has good resistance to physical impacts.
3. The microphone housing should be water and mud resistant.

A microphone manufactured by Knowles (model MR-23793) was chosen. The microphone was specified by the manufacturer to have good resistance to physical shocks, be waterproof for up to 15 metres and to resist the effects of mud, sand and salt encrustation. It also had a flat frequency response throughout the applicable bandwidth which made it a good choice for use in the elephant collar.

A circuit was designed to generate the microphone's biasing voltage and to centre the output signal of the microphone at the ground potential. A simple voltage divider circuit was used to provide the biasing voltage for the microphone. Equation 3.1 was used to determine the values of the resistors to be used for the voltage divider.

$$V_M = V_{CC}\left(\frac{R_2}{R_1 + R_2}\right). \tag{3.1}$$

A simple RC high-pass filter with a very low 3 dB cut-off frequency of 1 Hz was used to remove the DC voltage component from the output of the microphone. Equation 3.2 was used to determine the value of the resistor for the high-pass filter. In the equation, f_c is the 3 dB cut-off frequency, R is the parallel resistor and C is the series capacitor of the high-pass filter.

$$f_c = \frac{1}{2\pi RC}. \tag{3.2}$$

A cut-off frequency of 1 Hz was chosen to ensure that none of the spectral information from the elephant vocalizations would get lost. A capacitor value of 100 nF was arbitrarily chosen which gives a resistor value of approximately 159 kΩ. As can be seen from the circuit diagram shown in Figure 3.2, the voltage divider is buffered by an operational amplifier (more information on the choice of a specific operation amplifier will be given with the amplifier discussion). This is done so that the microphone may see a low input impedance on the biasing pin and will thus be able to draw current without effecting the voltage divider network. The capacitor between the biasing output at the voltage divider and ground will filter out any ripple that may be present on the signal.

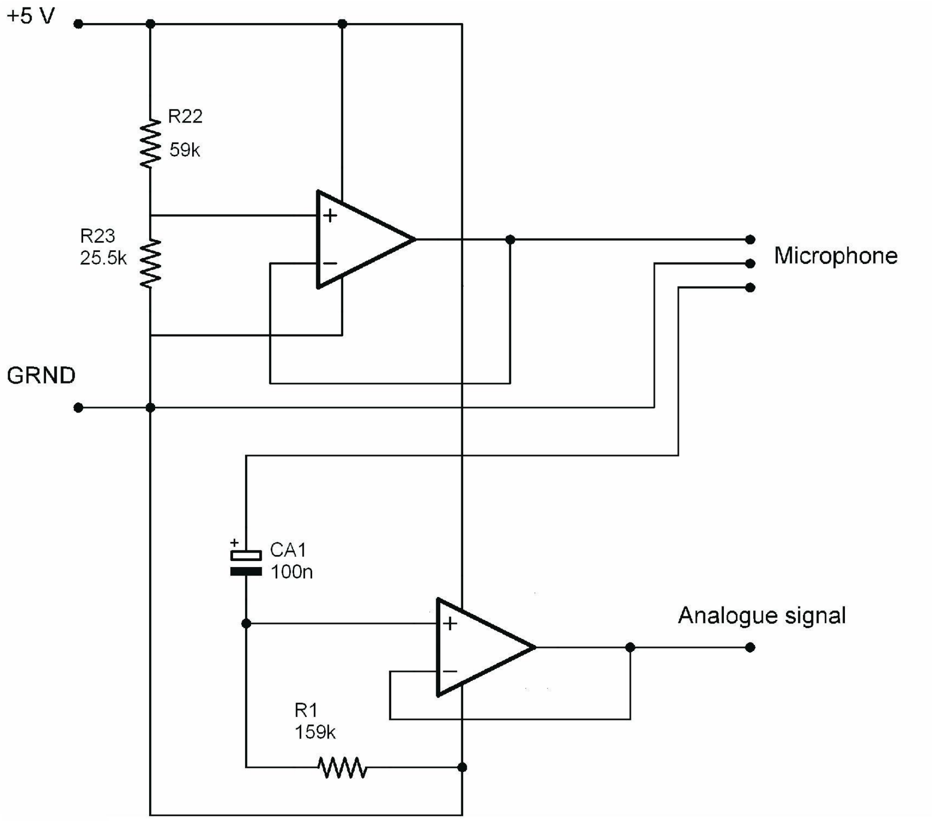

Figure 3.2: A schematic of the microphone conditioning circuit.

3.3.3.2 Amplifier

The analogue signal received from the microphone is in the tens of millivolts range, which requires that a sensitive amplifier was designed to amplify the signal to a useful magnitude.

An amplifier with AGC (Automatic Gain Control) was considered. AGC has the advantage of being able to record both soft sounds and loud sounds by automatically adjusting the gain level of the amplifier. However, the sound intensity is an important characteristic of the elephant vocalizations (e.g. this may be used to estimate the distance of a different elephant). This means that if an AGC is used, the gain level should be continuously recorded with the sound data to ensure that the sound intensity at a given instant can be established. Therefore, the limited storage capacity onboard the recording collar prohibits the use of AGC.

The second option is to set the amplifier to an optimal sensitivity level for the vocalizations of the particular elephant wearing the collar. This option has been chosen, because the microphone will always be at a relatively constant distance from the source of the sound.

A design choice was made to use an operational amplifier for the amplification of the signal from the microphone. This choice was made because of the ease of implementation and the ideal input and output impedances that an operational amplifier provides. A general purpose operational amplifier has a drain current of up to 5 mA. The power consumption of the amplifier that is used in this application has to be as low as possible to conserve battery power. MC33171 micropower operational amplifiers were selected for the design. These operational amplifiers are specified to have a drain current of less than 100 μA which is at present the operational amplifier with the lowest power consumption of all those available that can operate at the desired voltage.

Two inverting amplifiers were cascaded with the first one being a very sensitive amplifier and the second one adding gain to the total amplifying circuit. Equation 3.3 was used to determine the gain of each amplifier with G being the gain of the amplifier, R_1 the resistor connected to the input signal and R_2 the feedback resistor.

$$G = \frac{R_2}{R_1}. \tag{3.3}$$

Each amplifier was designed to have a gain adjustable from 0 to -20 V/V by choosing R_1 10 kΩ and using a 200 kΩ potentiometer for R_2. This results in a total achievable gain of 400 V/V. The resistors connecting the non-inverting inputs of the operational amplifiers to ground were determined by Equation 3.4.

$$R_x = R_1||R_2. \tag{3.4}$$

This resistor value optimizes the offset voltage of the amplifier. Since the value of the feedback resistor will almost certainly always be much larger than the input resistor, a value close the input resistor's value of 10 kΩ has been chosen for R_x. A value off 10 kΩ has been chosen for R_x in both amplifiers.

3.3.3.3 Anti-aliasing filter

The anti-aliasing filter is a low pass filter that should ensure that no spectral overlap occurs in the recorded signal. All frequencies with a value higher than half of the sampling frequency should be suppressed to avoid aliasing when the signal is sampled. The sampling rate is 3 kHz, thus the highest frequency in the sampled signal is limited to 1.5 kHz. It will be desirable to suppress frequencies higher than 1.5 kHz with at least 20 dB. The highest frequency that needs to be recorded is 1 kHz. This means that the filter can start to cut off at 1 kHz and should reach 20 dB attenuation within 500 Hz.

The first design choice that had to be made was to use either a digital or an analogue filter. A digital filter provides an excellent anti-aliasing filter and needs only a small resistor-capacitor filter after its output. Such a filter can be found inside an audio codec chip that also houses an ADC and gives a digitized sound signal as output. It does however need a high frequency clock source to drive it and the ones that are available drains 15 mA of current which makes its power consumption too high (remembering that the total current drain of the system should remain less than 33 mA).

Both Butterworth and Chebychev analogue filters were considered as alternatives. A Butterworth filter has the advantage of having a maximally flat pass band, but its cut-off rate is slower than that of an equal order Chebychev filter. A Chebychev filter has a much steeper cut off-rate but has ripple in the pass band.

A sixth order Butterworth filter was chosen. The flatness of its pass band and the ease of implementation were the main reasons for the choice. Even small tolerances in values of components may result in a higher order Chebychev filter becoming unstable. In the initial realization, three almost identical second order sections were implemented. Table 3.2 shows the frequency and gain values for each of these second order sections needed to realize a sixth order Butterworth filter with a 3dB cut off at 1 kHz.

Table 3.2: Cut-off frequency and gain of each second order section of the filter.

Section no	f_c **(Cut off frequency)**	G **(Gain)**
1	1 kHz	1.068
2	1 kHz	1.586
3	1 kHz	2.483

Equation 3.5 is used to calculate the frequency dependent component values while Equation 3.6 calculates the gain of the section.

$$f_c = \frac{1}{2\pi RC}. \tag{3.5}$$

$$G = \frac{R_2}{R_1}. \tag{3.6}$$

Figure 3.3 shows one second order section of the Butterworth filter. Three of these sections, each with a different gain (as shown in Table 3.2), were connected in series to realize the Butterworth filter.

3.3.3.4 Analogue to digital converter

The function of the ADC is to convert the amplified analogue signal received from the microphone into a digital signal. A wide variety of ADC's are available in IC (integrated circuit) packages. The main difference between the available ADC's are the data output method, input range, speed of conversion, and resolution and power consumption.

The speed of conversion needs to be fast enough so that a complete conversion can be done and the digital data can be read and stored during a single sampling period. Since the sampling frequency is 3 kHz, the rate of conversion should be greater than 3000 conversions per second. The resolution of the ADC determines its SNR (Signal-to-Noise Ratio). A 16 bit ADC should be used, which gives a SNR of 98 dB.

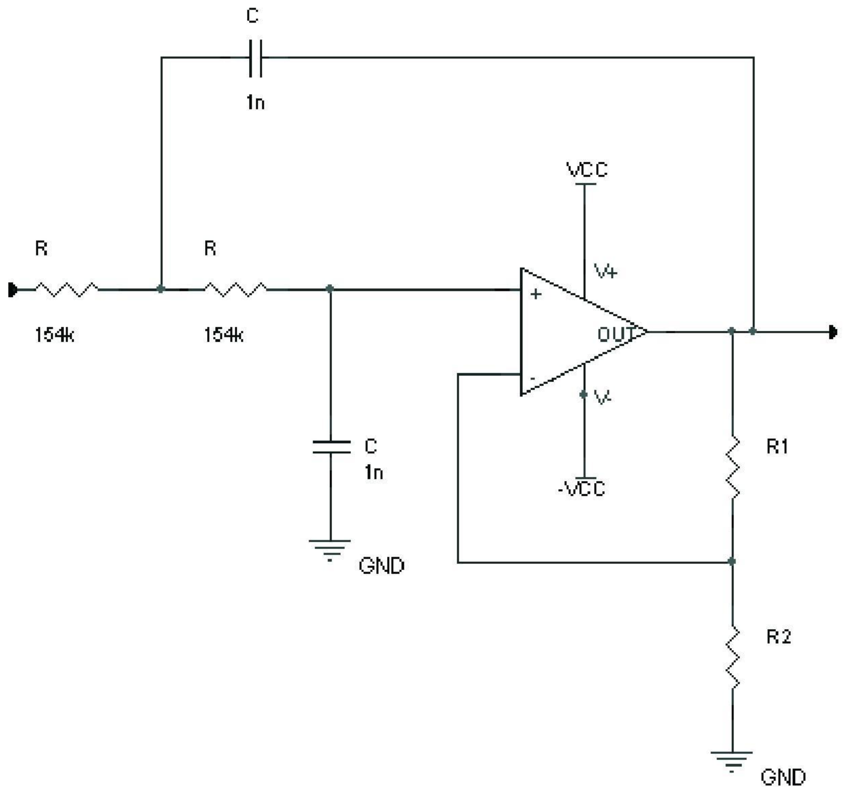

Figure 3.3: A single second order section of the Butterworth filter.

3.3.3.5 Data storage

The digital data received from the ADC have to be stored on permanent storage media. The permanent storage media should have the following characteristics:

1. A storage capacity of more than 43 GB is needed to store data continuously for three months.
2. Resistance against physical impacts is a necessity.
3. The storage device should be as power efficient as possible.
4. Memory with small physical size is required.

The sampling frequency is 3 kHz and 2 bytes of data will be stored 3000 times per second. Using Equation 3.7, this results in approximately 43 GB of memory required for continuously recording over a 90 day period.

$$\text{Capacity} = Bytes/s \times 3000Hz \times 60s \times 60min \times 24h \times days. \tag{3.7}$$

A notebook hard drive was considered as one alternative. It is relatively small in size and will have ample storage capacity. However, because a hard drive has moving parts it could be the weakest link when considering the robustness of the system. It is unclear whether a hard drive would be able to survive the physical impacts that the collar would be exposed to while on an elephant. The power usage of a hard drive is also very high in comparison to some solid-state memory cards.

Different solid-state memory options were available. RamSan is super fast RAM (Random Access Memory) memory that has reached capacities of more than 1 Terabytes. Even though it is solid state memory and has more than enough storage space, the size and high power usage of these devices makes it unsuitable for this application. Flash memory seems like the best choice for the application at hand, since it is mechanically robust, small in size and uses little power. The memory capacity on a singe flash memory device will not be sufficient. It has therefore been decided that an array of flash memory devices would be used to provide enough storage memory to the system.

Universal Serial Bus (USB) flash drives, SD memory cards and CF memory cards were but a few of the other available options.

After some consideration, CF cards have been chosen as the storage media that would be most suitable for the recording system. Mini hard drives with a CF interface are also available, so using CF cards would keep the option open for experimenting with (less expensive) hard drives to see if they will survive for the duration of the recording period. An array of four CF cards will be used to ensure that required amount of memory space is provided.

The CompactFlash card specification was formulated by the CompactFlash Association in 1995. CF cards are compact, removable non-volatile memory cards. It has 50 interface pins and is roughly the size of a matchbook. Internally, the CF card consists of an interfacing microcontroller, a buffer and varying amounts of non-volatile memory. The memory capacity of CF cards is ever increasing and 32 GB CF cards are currently available. The fact that CF cards have the highest storage capacity of all flash memory cards and this together with its low power consumption is the main reason why it has been chosen for this design.

The internal microcontroller of the CF card does the work of storing and accessing the data onto and from the physical memory according to commands given to it through the interface pins. This makes it possible to access CF cards with different memory capacities and technologies with the same instructions so that future CF cards that use new technology can still be accessed in the same way as before. The CF card's internal microcontroller also makes it possible to access the card in a number of different modes, including Common Memory mode and true IDE mode. When the card operates in true IDE mode it can be connected to an IDE bus and a PC will see it as a hard drive. When configured to operate in the Common Memory mode (as is the case for the elephant collar), the CF card operates with an 8 bit data bus instead of the 16 bit data bus when used in IDE mode. This enables an 8-bit microcontroller to access the card.

The CF card can be completely controlled through eight control registers by loading them with appropriate data. Table 3.3 shows the address and name of each of the eight addressable registers. As can be seen from the table, only three address lines (A0 to A2) are needed to access the control registers and thus control the whole CF card.

Table 3.3: The eight control registers of a CF card and their three-bit addresses.

-REG	A10	A9-A4	A3	A2	A1	A0	Offset	-OE=0	-WE=0
1	0	X	0	0	0	0	0	Even RD Data	Even WR Data
1	0	X	0	0	0	1	1	Error	Features
1	0	X	0	0	1	0	2	Sector Count	Sector Count
1	0	X	0	0	1	1	3	Sector No.	Sector No.
1	0	X	0	1	0	0	4	Cylinder Low	Cylinder Low
1	0	X	0	1	0	1	5	Cylinder High	Cylinder High
1	0	X	0	1	1	0	6	Select Card/Head	Select Card/Head
1	0	X	0	1	1	1	7	Status	Command

To load these registers with data, the correct address is placed on the 3-bit address line and a read strobe is given to read the data into the register. After the first seven bytes have been loaded with sector address information, the Command register is loaded with a certain command (usually either a Read Buffer or Write Buffer command). Address zero (the Data register address) is then selected by the three address lines to enable a card buffer to be read or written to. If a Read command is written to the Command register, the CF card's memory buffer is loaded with the contents of the sector to which the address registers is pointing and the information contained in that sector can be read byte by byte by giving a Read pulse until the desired information is reached. If a Write command is written to the Command register, an empty 512 bit buffer is presented and one byte of data is read into the buffer every time the Write pulse is given. When the buffer is completely filled with written data, the internal microcontroller of the CF card automatically moves the written data from the buffer to the non-volatile memory of the card. The next sector address then needs to be loaded into the address registers and another command written into the Command register.

Although the register names imply the SCH (Sector Cylinder Head) sector addressing method, LBA (Logical Block Address) addressing can be selected by setting the three most significant bits of the Head register. LBA is preferred to SCH because with LBA each sector of the disk gets its own specific number starting at zero and counting up the card capacity. With a 27 bit LBA, more than 64 GB of memory may be addressed. Using LBA addressing, the Sector No. register becomes the LBA first byte register, the Cylinder Low register becomes the LBA second byte register and the Cylinder high becomes the LBA third byte register.

When in Common Memory mode, the CF card can operate using only 17 of its 50 pins: the eight data pins, three address pins, read, write, ready/busy, reset, card detect and card select pins. Address lines 3 to 10 are connected to ground and the second cards select pin and the Reg pin are connected to Vcc. All the other pins are left unconnected.

The ready/busy pin is tested to determine whether the card is ready to accept new data. Read and write pin are used to give either read or write pulses. The reset pin can be pulsed to reset the internal microcontroller of the CF card. The card detection pin is internally routed to ground. It can be connected to Vcc through a pull up resistor and the value at the pin can be read to determine if a card is present. If a card is present, the resistor will be grounded and a low signal will be read on the pin. If a card is not inserted, the resistor has one end connected to Vcc and the other open end is read as a high signal, which indicates to the system's controller that no card is present. The card select pin is used to activate a CF card and connect the data bus to the card. If the card is not enabled, the data bus is disconnected and in a state of high impedance. This pin plays a very important role in the designed system since all the CF cards and the RAM shares the same 8-bit data bus.

A problem that had to be addressed was the fact that a single CF card would not have enough storage capacity to store all the data for the desired period of at least 90 days. One CF card has a maximum storage capacity of 16 GB while around 40 GB of storage space is needed. Connecting multiple CF cards onto a shared 8-bit data bus can solve this problem by selecting only one card at a time using the card select pin. When operating in this way all the pins of each card can be connected to each other accept for the card select pin, the card detect pin and the ready/busy pin which cannot be shared. When using 4 CF cards, a total memory capacity of 128 GB can be achieved.

3.3.3.6 Card detection and selection logic

As mentioned in the previous section, each CF card needs a dedicated card select, card detect and ready/busy pin. This results in an extra 12 I/O pins needed on the microcontroller. To save I/O pins on the microcontroller, a card selection module was developed. The card enable pins are input pins that need to be written to and the ready/busy and card detect pins are output type pins that need to be read. The card enable pin on a CF card allows a single card in an array of cards to be selected for reading and writing operations. If a low signal is applied to the active low card enable pin, the card is active and ready to send or receive data. If on the other hand a high signal is applied to the card enable pin the card disconnects its data and address busses and goes into an inactive low power state.

All the card enable pins were connected to the first section of a 74139 2 to 4 line decoder with active low outputs. Two address lines from the microprocessor control the 4 output pins. The output pin to which the applied address corresponds to goes low while all the other pins remain high. In this way each of the 4 CF cards can be selected using only two I/O pins from the microcontroller. If the chip enable input of the 74HCT139 is deactivated, all its output pins goes to the high state.

Now that each card can be selected as desired, the dedicated output signals from each card needs to be linked with the microprocessor using the minimum amount of I/O pins. This has been done using a 74153 dual 4-bit multiplexer. When a card is selected, it's ready/busy and card select pin is automatically routed to the respective I/O pins of the microcontroller and the microcontroller can function in the same way as it would if only a single card were present.

The card detect pin of a CF card is internally routed to ground. When a card is inserted the pin is routed to ground and the voltage on the pin changes from high to low so that it may be detected that a card is present in that slot. A high resistor value of 159 kΩ was chosen to put in series with the card detection pin to ensure minimum current drain. Using Ohm's law, a current of only 32 uA is drained through each resistor when a card is present and has virtually no voltage drop over it when a card is not present.

3.3.3.7 RAM memory

The fact that a CF card drains 30 mA or more when in writing mode, means that the system must be designed so that the time that the card spends in writing mode is minimized.

It had been decided to add 512 kB of static RAM to the system. This enables the system to store 1024 sectors of data in the external RAM. While the digital sound samples are stored in the RAM the CF card may be disabled, putting it in a low power state. When the RAM is filled to capacity the data will be transferred to the CF card at maximal speed. The on time of the CF card will only be a small fraction of the time it takes to accumulate the 512 kB of data samples. CF cards uses 200 mA of current when in write or read mode, so using the low power RAM (100 uA) reduces power consumption considerably.

The 62512 512kB RAM IC was used to supply the external RAM. The RAM chip is very power efficient, since it drains only 100 μA. To save I/O pins on the microcontroller, two 8-bit latches (74HCT377's) and a 4-bit latch (74HCT75) were connected to the common 8-bit data bus shared by the CF cards. The data pins of the RAM chip were also connected to the common data bus. To address the RAM, all the devices on the common data bus was disabled, except the three latches. The data bus was used to clock and hold the first part of the RAM address on the first latch and then clock and hold the second part of the address on the second latch and then on the third latch. Now that an address for the desired bit position on the RAM had been applied, the RAM unit is enabled. To write data into the RAM the data is presented on the data bus and stored to the RAM by pulsing its write pin. To read data out of the RAM, the read pin of the RAM is pulsed to put the data from the RAM onto the data bus.

3.3.3.8 Microprocessor

It was considered to use a single board computer to manage the storage of the data on the CF cards. This option would make the task of storing sound files in on a file system much easier. However, the high power usage and large physical size of a single board computer together with high cost made it an unfeasible option.

Therefore, it was decided that a microprocessor would manage the flow of data from the ADC to the storage media. It is important that the microprocessor consumes as little power as possible and that it has enough I/O pins. An 8-bit data/address bus, a 3-bit instruction bus and at least 6 other I/O pins are needed to establish communication between the microprocessor and the CF card. Another three pins are needed for the communication with the ADC. For accessing multiple CF cards, multiplexers and demultiplexers would be needed which also requires 4 pins. This amounts to 24 I/O pins, but since it is desirable to have a modular design in which other peripherals could later be added to the system if the need arises, it would be wise to have at least 30 I/O pins available. Using the CF card in its common memory mode allows us to use an 8-bit microcontroller, which is less power consuming than 16-bit or 32 bit processors. The PIC16f877 8-bit microcontroller was chosen because of its low power consumption. It has 33 I/O pins, which will allow extra peripherals to be added to the system. The PIC was programmed in assembly language using Microchip's MPLAB software and using the MPASMWIN assembler. The software design and implementation on the microcontroller will be discussed in the Section 3.3.5.

3.3.3.9 Thermometer

The advantages of the addition of a digital thermometer to the system has been discussed in Section 3.3.1. A device with low power consumption, an accuracy and resolution of at least 1 °C in the correct temperature range should be used.

Maxim's Ds18B20 low power digital thermometer was chosen. It communicates with the microcontroller using a one-wire interface, so that only one I/O pin of the microcontroller is used to send and receive data. The device can be configured to operate in a parasitic power mode where the device does not need to be connected to an external power source but charges its internal battery each time a high data pulse is received. The device has a resolution of 0.0625 °C and an accuracy of 0.5 °C. The device can measure temperatures ranging from -50 °C up to 125 °C.

3.3.3.10 GPS module

The U-blox LEA-LA module was chosen used because of its small size and the fact that it had the lowest power consumption of all available modules. The LEA-LA is equipped with two Universal Serial Asynchronous Receive Transmit (USART) ports. The PIC16F877 is also equipped with an USART port and communication is established using the RX and TX pins of both devices.

An U-blox customized GPS protocol called UBX is used for communication with the microcontroller. An UBX command can be sent to request positional coordinates, as well as time and date information. When the GPS is in normal operating mode, it draws a current of approximately 150 mA. It should thus only be used to acquire the necessary information and then be switched into sleeping mode where less than 1mA of current is drawn.

3.3.4 Printed circuit board design

All the main electronics are situated on a main system board while the CF card connectors and the card selection and detection modules are housed on a separate connector board. These two boards both had measurements of 140x180 mm. They were mounted on top of one another to ensure that the size of the collar remained within the desired specifications.

Both printed circuit boards (PCBs) were designed in OrCad 10 using four conducting layers. The top and bottom layers are used as routing layers, while the two layers in the middle are used as a ground layer and a power layer. The four-layered approach was chosen for two reasons. First, having more layers allows components to be placed closer together allowing for a smaller PCB to be manufactured. Second, having a ground and power layer increases the stability of operation of the electronics. Variations on the power lines are counteracted by the capacitive effect of the two power layers. The routing on the top layer is horizontal while the routing on the bottom layer is vertical to ensure minimal signal interference.

3.3.5 Microprocessor software design

All the digital hardware described in the previous section are controlled by the microcontroller. It has the task of coordinating all the different components of the system and to control the flow of data from the microphone to the CF cards. All the software was written for the PIC16F877 microcontroller using assembly language. All programming was done in Microchip's MPLAB.

Even though the software was written specifically for the PIC16F877 microcontroller, only the functional design of the software will be discussed. The same principles that have been used in the software design for this project could later be used even if a different microprocessor was to be selected.

In principle, the software should be designed to regulate the following operations:

1. Data protection
2. File system management
3. File structure control
4. Data flow control

Data protection should be implemented to ensure that previously recorded data can not be overwritten by accident. File system management ensures that the data that is stored on the memory cards will be accessible from any Windows operated computer. It also ensures that the data can be transferred from the cards to a computer's hard drive. In this case the FAT32 file system was implemented. The file structure is used within the file system and provides a way to store the sound data in a standardized manner so that it can be played or processed correctly by other computer programs.

The software controls the operation of the different hardware modules and also determines in what format the data is stored on the memory cards. The following subsections describes each software module in more detail.

3.3.5.1 Protecting the recorded data

As soon as the power to the system is switched on, some microcontroller specific tasks are performed to set it up for correct operation.

After the initialization, the system waits for a record command. If the system simply started to record data as soon as the power has been switched on, previously recorded data that had not yet been downloaded would be overwritten. Also, the power to the system might be switched on in advance, but it will not be desirable to start recording right away. For these reasons recording should only start after the user of the device has given a record command.

Data protection have been done by forcing the user to first press a push button for a certain duration of time before the recording process is started. The flow diagram of Figure 3.4 shows how the user will enable the start of the recording period. After the power has been switched on, a dedicated pin will be configured as an input pin and is pulled low by an external resistor. This pin is connected to the base of a transistor that can drive a LED. When the pushbutton is depressed, the signal on the LED pin will change to the high state. The program will go into a loop and poll the state of the signal on the LED pin continuously to determine whether the button has been depressed. When the button is pressed the change of state on the pin will be detected and the program will set a register with a value of 8 to be used as a counter. It will then go into a subroutine to determine whether the button is pressed for longer than 4 seconds, by calling a delay of a half second and then testing (repeat eight times) whether the button is still being pressed. If the button is still pressed after half a second, the counter is decremented and another half-second delay is called. This process is repeated until the counter reaches zero or if the button is released before the 4 seconds have elapsed. If after any one of the half-second delays it is detected that the button is not pressed anymore the program will return to the loop and wait for the next time the button is pressed where after the whole process will be repeated again. The reason why the button has to be pressed for 4 seconds is to ensure that the device doesn't start recording if the button have accidentally been touched. If the counter reached zero and the button is still being depressed, a high signal will be placed on the pin that will drive the LED.

The shining LED will alert the operator that the device has started recording. The LED is also connected through the switch in order not to waste power, so after the operator has released the button the LED will be deactivated. The recording of data will now commence and the push button will have no further influence until the power has been switched off and on again.

3.3.5.2 Memory card detection and selection

As has been mentioned in the hardware design section of this report, there are eight CF cards connected on a single data bus. The address bus and the read and write pins are also shared by all the cards. Each card, however, needs its own dedicated card detect, card enable and ready/busy pins. This is achieved by using a 3 to 8 bit

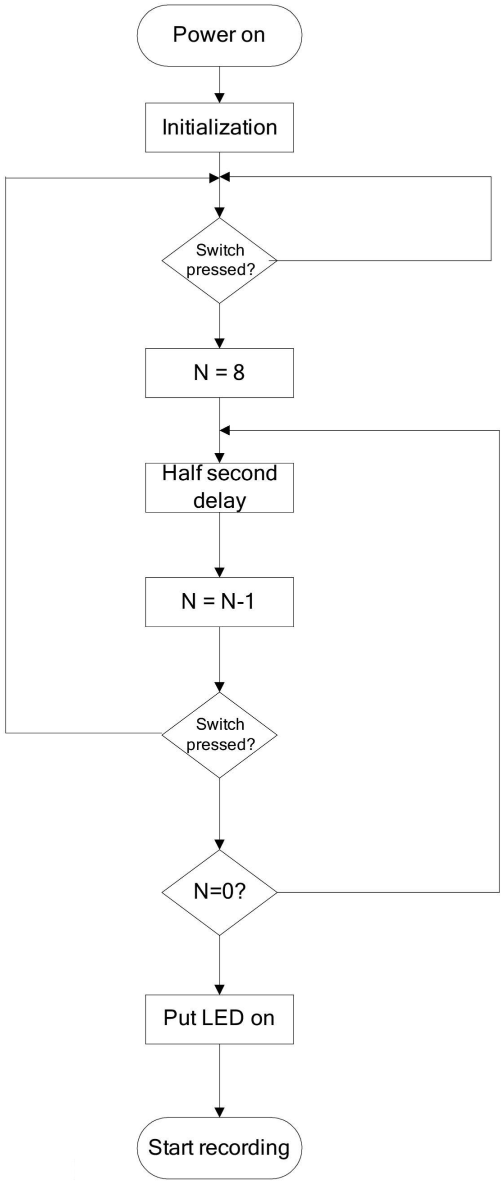

Figure 3.4: A flow diagram describing the program directly after the power to the system has been switched on.

line decoder and two 8-bit multiplexers. Figure 3.5 shows a flow diagram of how the software selects and detects CF cards.

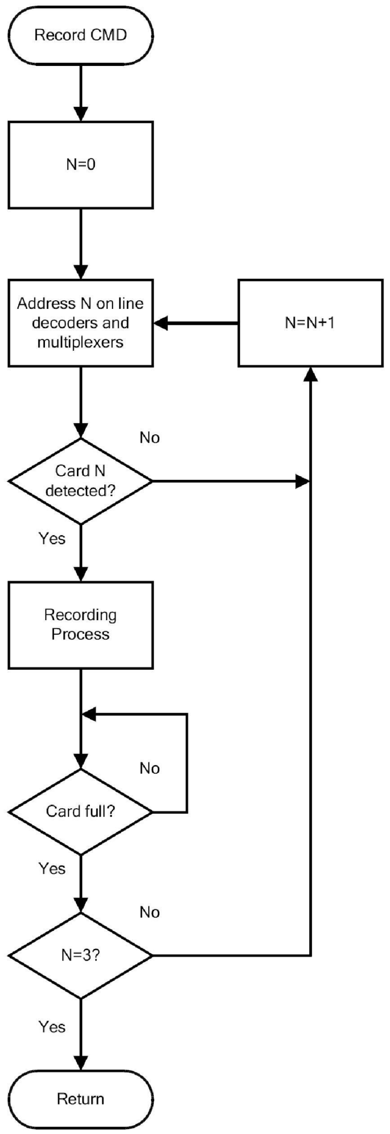

Figure 3.5: A flow diagram describing how specific CF cards are detected and selected for use.

When the record command is given, the first CF card socket is selected by writing zeros to all three address lines of the 3 to 8 decoder. This results in a low signal being sent the card enable pin of the first CF socket and high signals being sent to the other CF socket's card enable pins. The same addresses are sent to the multiplexers to connect the correct socket's ready/busy pin and card detect pin to the microcontroller. After the first CF socket has been selected, the card detect pin is used to determine whether a card is present in that specific socket. If a card is not detected the address of the line decoder is increased and the next CF socket is selected. The card detect pin of this socket is used to detect whether a CF card is present and this process repeats itself until a card is detected or until all eight the CF sockets have been checked. As soon as a card is detected, the program jumps out of the loop and continues with the rest of the prerecording routines. Data will be recorded onto the card until its capacity is reached. The program will then jump back to the card detection loop to find the next socket that contains a CF card. This process will be repeated until the eighth CF socket has been checked.

3.3.5.3 CF card read and write operations

Arguably the most important task of the microprocessor is to write data to the CF cards. As was discussed in the theory section, the entire CF card may be accessed using only eight control registers. The address of the sector that needs to be accessed is loaded into four address registers and either a read or write instruction is loaded into the command register. The data register is then selected and all data transfers take place through this register. Data is clocked into the CF card using a read or write pulse. The process is described by the flow diagram of Figure 3.6. The read and write routines were written based on a piece code published on the Internet by Mark Samuels.

After the LBA address of the desired sector has been entered, either a read or write command is loaded into the command register. At this point, if a read command was given, the CF card will load the addressed sector in to its data buffer. The data may then be read out sequentially until the whole buffer has been read. To access the next sector, a new LBA address has to be loaded into the address registers.

The write procedure is a little bit more complex. After a write command has been loaded into the command register, an empty data buffer is loaded. This buffer can then be filled by sequentially writing data to it. Before a byte can be written the ready/busy pin of the CF card needs to be checked to verify that the card is ready to accept new data. As soon as the 512-byte buffer has been filled with data, the buffer is automatically transferred to the internal memory of the card. No data can be transferred to the card memory if the data buffer had not been completely filled. To write data to the next sector, the LBA address of that sector needs to be loaded and the data buffer needs to be written again.

3.3.5.4 The FAT32 file system

The operation of a FAT32 file system will be given in detail. This is an important aspect of the software design.

3.3.5.4.1 Background information The FAT32 file system has been chosen for this system because it can be read by both Windows and Linux systems. This will enable a user to put a CF card in a card reader, connect it to a PC and drag the files onto the PC. The FAT16 file system was also considered, but it can address only a limited amount of storage space (2 GB) and will not suffice for the large storage cards.

The FAT32 file system consists of the MBR (Master Boot Record), volume ID, FAT tables and the cluster area where the file and directory data are stored. The very first sector on the disk is the MBR. The first 448 bytes of this sector contains boot information and has no use in an imbedded system. This is followed by partition information which holds the starting address of the FAT32 partition (volume ID) and the size of the partition. There is also a file system byte which can be used to test whether the disk is indeed formatted in the FAT32 file system. The LBA address of the volume ID is read from the MBR and the volume ID is accessed. Figure 3.7 shows the structure of a FAT 32 partition. Note that the MBR does not form part of the partition, but simply provides information like the LBA address of the volume ID.

The volume ID contains information about the starting location of the FAT tables, the sector size of the FAT tables, the number of reserved sectors, the cluster size and location of the root directory of the partition. There are a number of reserved sectors after the volume ID and before the start of the FAT tables. The cluster size is given in multiples of the 512 byte sector size and can range from two sectors (1 kB) clusters to 16 sectors (32 kB). Once all the necessary information has been read out of the volume ID, the FAT tables can be accessed. There are two FAT tables so that the operating

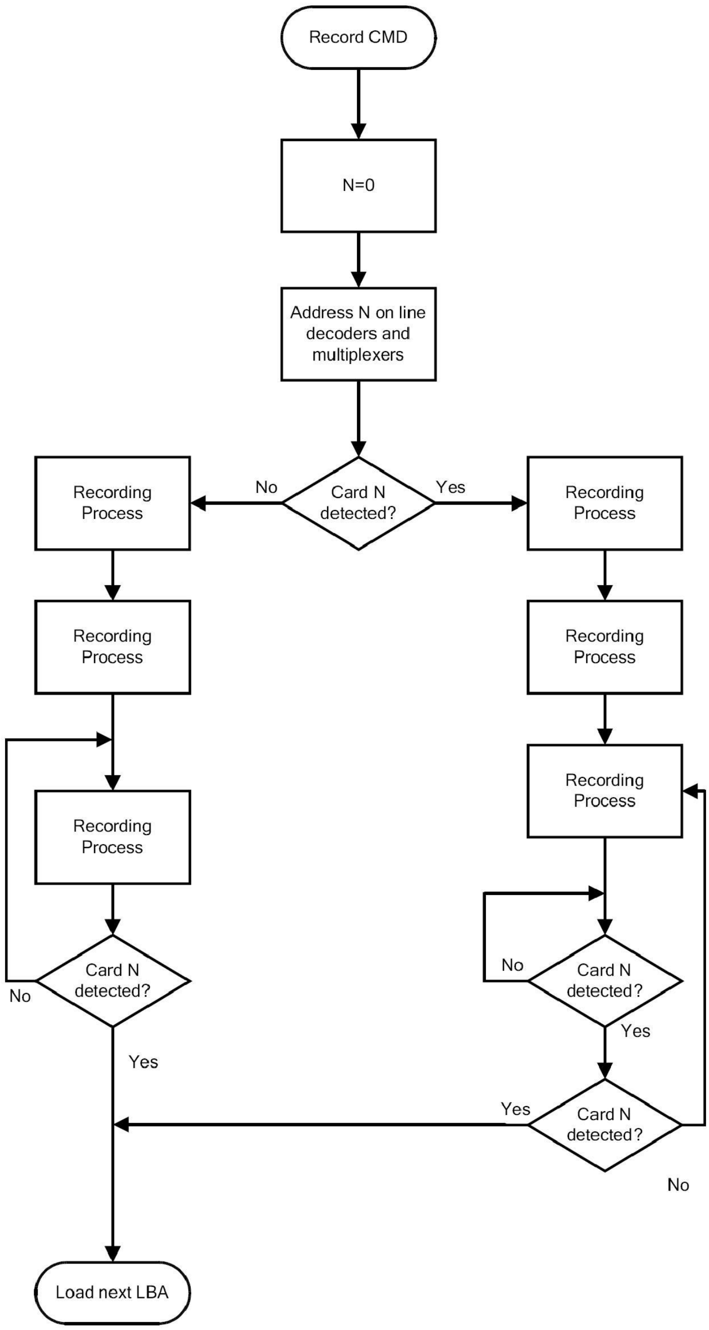

Figure 3.6: A flow diagram describing the data read and write procedure for the CF card.

system of the PC may check for consistency.

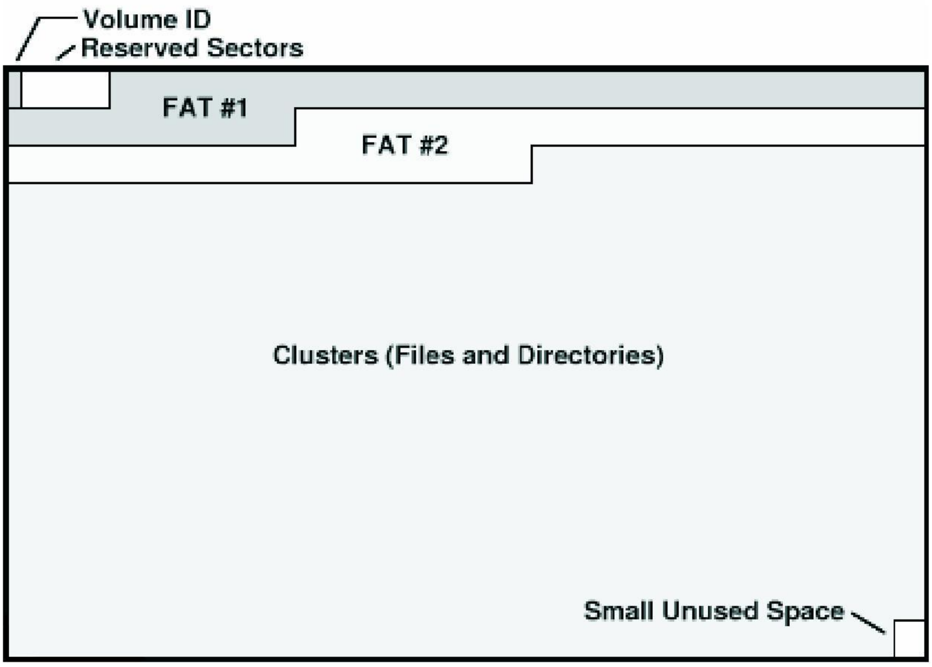

Figure 3.7: A Fat 32 disk partition.

The operation of the FAT tables is straightforward. The FAT table is a map of the cluster area where a four-byte number represents each cluster. This number indicates the location of the next cluster in the chain, or if the bits in that cluster are all set to 1's, it means that the end of the file chain has been reached. Cluster 0 and cluster 1 does not exist, so that the cluster numbering starts at 2. This first cluster is also the start of the root directory. The root directory is the main directory of the partition and all other directories and files are stored under this first directory.

Figure 3.8 shows how a FAT table might look after three small files have been written to the disk. As can be seen from the figure, position 0 and 1 are not applicable and the table starts at position number 2 where the root directory is located. The value stored in position 2 is 9 so the next cluster where the root directory's information will continue is cluster number 9. In cluster number 9, the value of 10 is found so the root directory will continue in cluster 10. This in turn points to cluster 11 and it points to cluster number 17. The bits in position 17 are all set to 1's. This means that the cluster chain ends at this position (the end of the root directory). In a similar fashion, the cluster chains of each of the three files may be followed from their starting point to their finishing point by using the FAT table. The question is now, how does one know at what position of the FAT table a file begins?

XXXXXXXX	XXXXXXXX	00000009	00000004
00000005	00000007	00000000	00000008
FFFFFFFF	0000000A	0000000B	00000011
0000000D	0000000E	FFFFFFFF	00000010
00000012	FFFFFFFF	00000013	00000014
00000015	00000016	FFFFFFFF	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

Root Directory:
2, 9, A, B, 11

File #1:
3, 4, 5, 7, 8

File #2:
C, D, E

File #3:
F, 10, 12, 13, 14, 15, 16

Figure 3.8: A visual representation of how a FAT table operates, showing the hexadecimal values stored in each position.

The starting cluster number of a file is stored together with the file name entry in a directory. The structure of the 32-byte name entries is shown in Figure 3.9. The short filename consists of an eight character filename with a three character extension code to identify the file type. An example of a file name would be "EXAMPLEFWAV" where the file name is EXAMPLEF and the file type is a wave sound file. The next byte is the attribute byte where the attributes of the file are set. A hexadecimal value of 0x08 sets the archive attribute and other attributes like hidden or read only can also be selected by setting the correct bits.

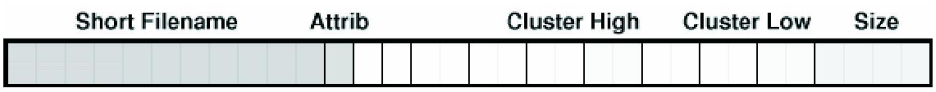

Figure 3.9: The 32-byte directory structure used for short file names in FAT32.

The time and date of file creation and the last time that the file was accessed may also be stored. Four bytes are allocated to store the number of the cluster where the files begin. This value will be used to find the correct position on the FAT table to start the cluster chain for the file. Because a file can end anywhere in a cluster, the cluster information is not enough to correctly determine the file size. For this reason the size of the file (in bytes) is stored in the last four bytes of the file name structure.

The file data is written to the cluster number indicated in its file name structure. If the data is larger than one cluster, the data flows into a next cluster and the FAT tables are updated.

3.3.5.4.2 Reading FAT32 information The very first sector on the disk is known as the master boot record. The first 446 bytes of the MBR contain irrelevant information and are skipped. Then follows a 64-byte partition table which is read. This table contains information on the starting location of the partition and what type of file system is used. The type code is tested to determine whether the file system is indeed FAT32. If an invalid code is detected the program will ignore the card and skip to the next card. The starting LBA of the FAT32 partition is read out of the table and the program loads the sector from the CF card.

This first sector of the FAT32 partition, called the volume ID, contains information on the partition, including the number of sectors per cluster, the number of reserved sectors (before the first FAT table), the number of sectors per FAT table and the cluster location of the root directory. The mentioned parameters are the only ones needed for this application. The program flow diagram for retrieving and calculating important FAT32 parameters are given in Figure 3.10.

After all the necessary information has been retrieved, the start of clusters LBA and the FAT tables' begin LBA needs to be calculated by using Equations 3.8 and 3.9.

$$\text{Fat\_Begin\_LBA} = \text{Partition\_LBA\_Begin} + \text{Number\_of\_Reserved\_Sectors}, \tag{3.8}$$

$$\text{Cluster\_Begin\_LBA} = \text{Fat\_Begin\_LBA} + \text{Number\_of\_FATs} * \text{Sectors\_Per\_FAT}. \tag{3.9}$$

With these values calculated, the program now has the ability to manipulate the FAT tables and all the data in the clusters of the FAT32 partition.

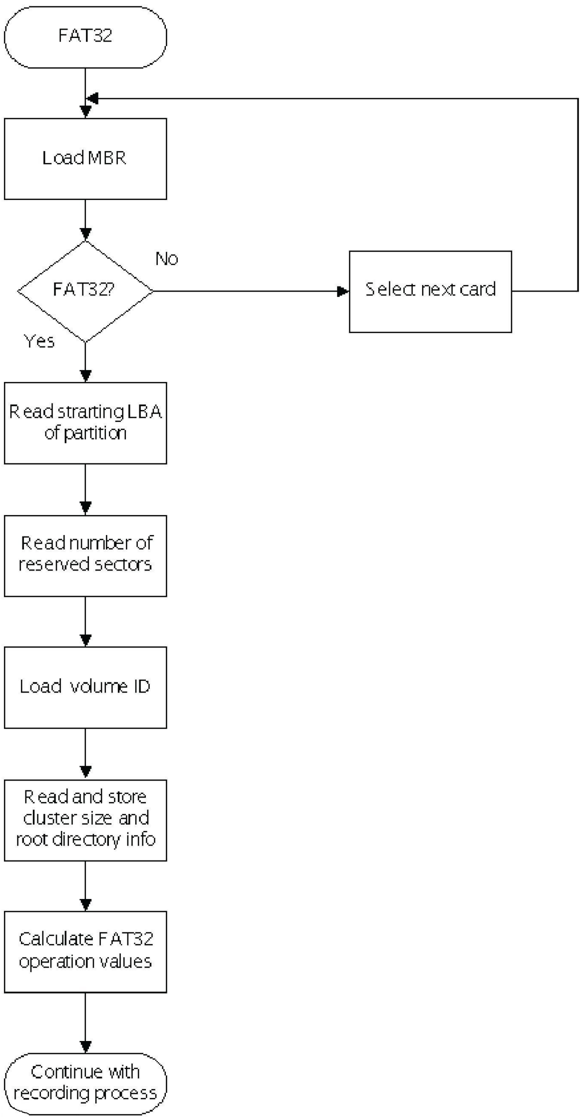

Figure 3.10: A flow diagram describing how important FAT32 information is retrieved and calculated.

3.3.5.4.3 Writing the FAT tables The FAT table is a map of the cluster area where a four-byte number represents each cluster. This number indicates the location of the next cluster in the chain, or, if the hexadecimal value in that cluster is 0xFFFFFFFF it means that the end of the file chain has been reached. Cluster 0 and cluster 1 does not exist so they have a default value of 0xFFFFFFFF and cluster numbering actually only starts at position 2. Cluster 2 is also the start of the root directory. Each FAT table entry contains the number of the cluster position where the data of the current file continues.

Figure 3.11 shows the flow diagram of how the FAT tables were written. A counter N is used to determine the total cluster position. The counter M is used to determine when the data buffer of the CF card is full so that a new sector may be loaded. Since FAT table entries are 4 bytes in size, 128 table entries fits into one sector. There is also a file size counter to keep track of the size of the current file that is being written to the fat. Finally there is a counter K to determine whether the first or the second FAT table is currently being written to.

All the files written to the cards will have an equal length of Fsize. This part of the program begins by first testing the value of K to determine that both tables have not yet been written to and then writing the position of the next cluster position in each Fat table position. After a value has been written N and F are increased and M is decreased. The values of these counters are all tested after each FAT entry. If M is zero a new LBA address needs to be loaded because the CF's data buffer is full. M is then reset to 128 again. If F equals the file size a hexadecimal value of FFFFFFFF is written to the FAT table to indicate that the file had ended and the value of F is reset to 0. If N is at the capacity of the card (the card can only hold a Ncap amount of clusters) the FAT table have been filled and the next table is written in the same way.

3.3.5.4.4 Writing the root directory The root directory is the main directory of the partition and all other directories and files are stored under this first directory.

The file names written in the root directory will be seen in the CF card folder when it is connected to a PC. These files will then be copied by simply dragging them onto the PC. The root directory is written in the way described by the flow diagram of Figure 3.12. The N counter keeps track of the cluster location, where each file starts and increases by the size of a file (in clusters) after every entry and finally causes the

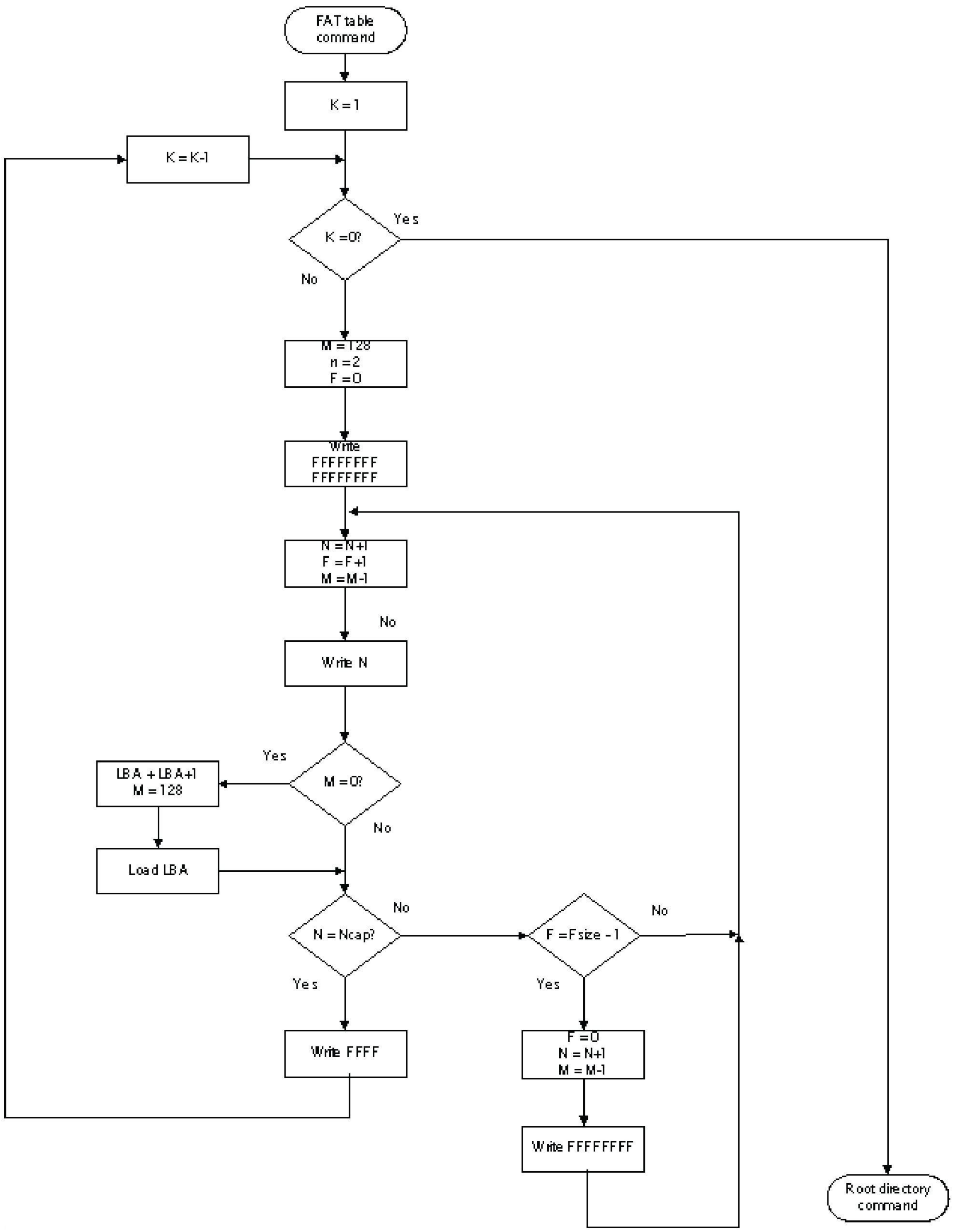

Figure 3.11: A flow diagram describing how the FAT tables are written.

program to go out of the root directory loop when the CF card's capacity has been reached (N equals Ncap). This cluster information is needed so that the operating system of a PC can locate the correct Fat table position for the starting point of the file and follow the cluster chain when copying the file.

The file name structure is 32 bytes in size, which means that 16 directory entries can be made in one 512-byte sector. For this reason the M counter has a starting value of 16 that decreases every time a file name entry has been done. When M reaches zero, the next sector is loaded and the value of M is reset to 16. The Name counter is increased after every entry and has the sole purpose of ensuring that every file name in the root directory is different.

3.3.5.5 Controlling the ADC

The ADC samples a value from the analogue signal from the microphone and converts it to a 16 bit binary number. This binary number needs to be read into two of the microcontroller's registers in order to be processed further. Since the data is output serially on a single pin, a subroutine needed to be written to retrieve the serial data and store it in microcontroller registers. The flow diagram in Figure 3.13 shows how this was achieved.

First, an active low convert pulse is sent to the ADC to initiate a convert cycle. Two clock pulses follow this, the first of which starts before the end of the convert pulse. Sixteen clock pulses are now sent to clock out the data bit by bit. On the rising edge of each clock pulse, the serial output pin from the ADC is read. If the pin is high, the corresponding bit in the ADC data register is set accordingly. If the data pin is low, the corresponding bit in die data register is cleared. The bits are clocked out MSB (most significant bit) first and in the microcontroller two eight bit registers are needed to store the 16-bit binary word.

When storing 16-bit digital sound data in a wave file, the data need to be in the 2's complement form. The last part of the ADC routine converts the binary number to 2's complement by inverting the MSB.

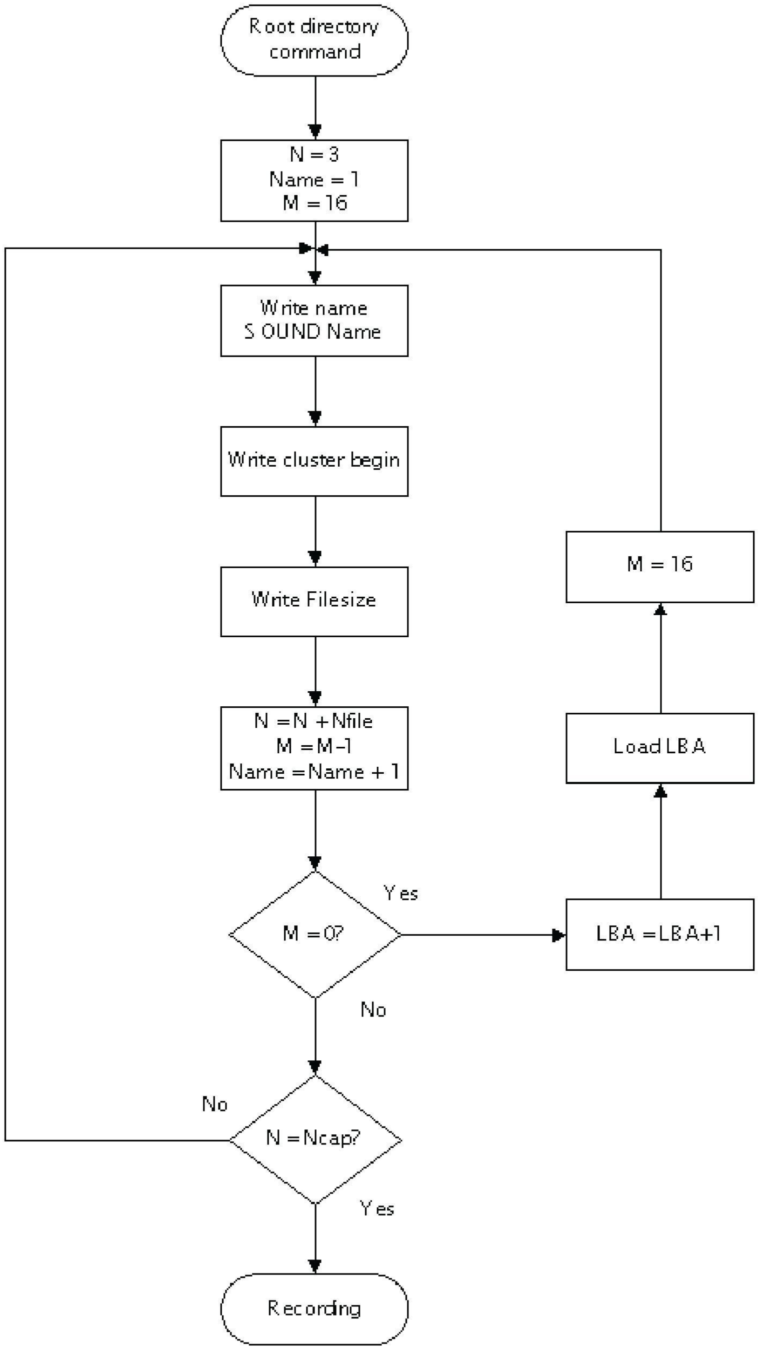

Figure 3.12: A flow diagram describing how the root directory information is written.

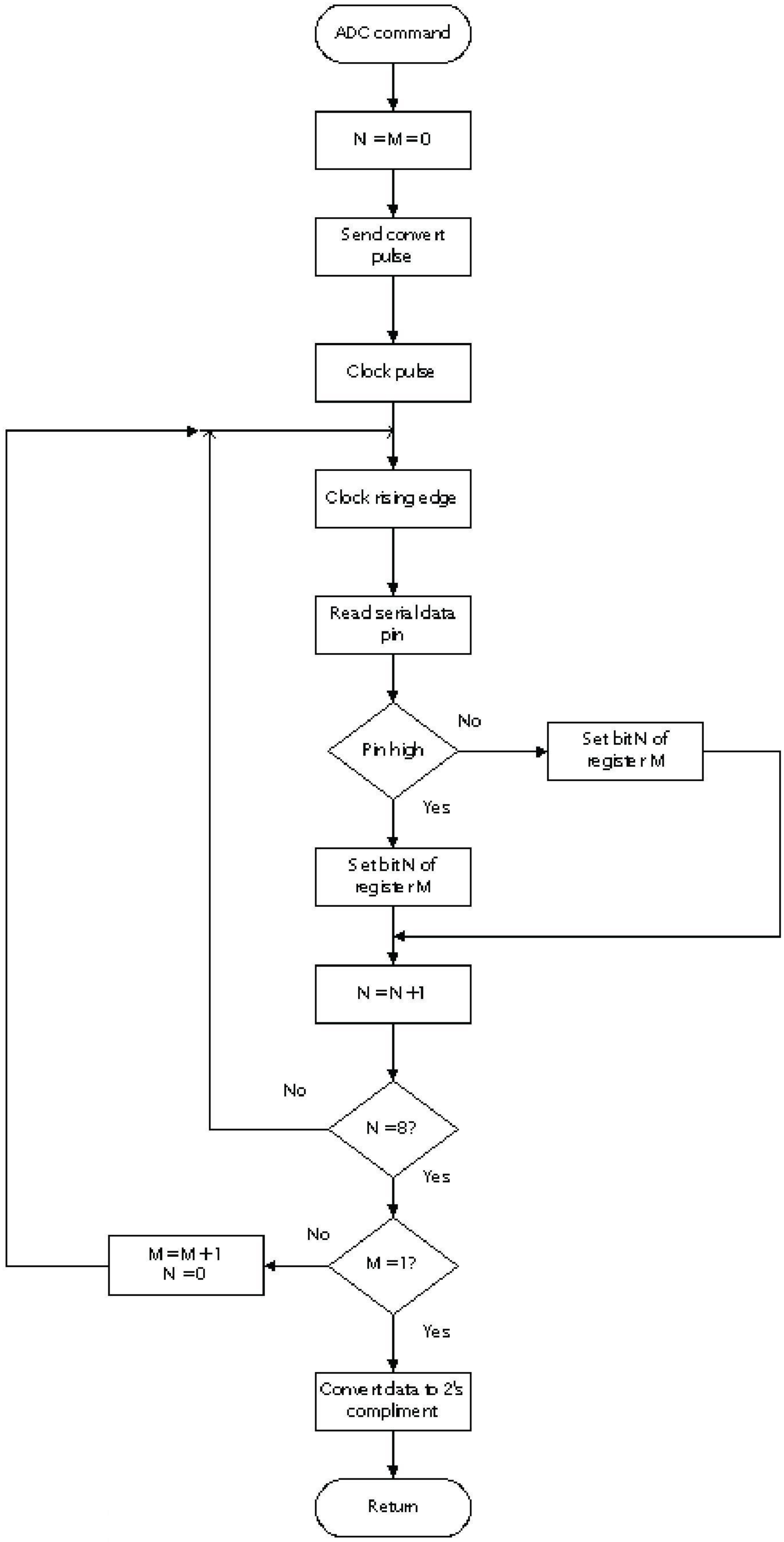

Figure 3.13: A flow diagram describing how the ADC subroutine works.

3.3.5.6 Recording sound

3.3.5.6.1 The wave file format The Wave file format is Windows' original file format for storing digital audio data. It has become one of the most widely supported digital audio file formats on PC's because Windows is so popular and a huge number of programs have been written to handle wave files. Almost every modern program that can open or save digital audio supports this file format and it was the logical choice to use it as the file format in which the sound recorded by the designed device will be stored.

Wave files use a RIFF structure which groups the file contents into separate chunks, each containing its own header and data bytes. The chunk header specifies the type and size of the chunk data bytes. Certain types of chunks can contain sub-chunks. In Figure 3.14 it can be seen that the “fmt” and “data” chunks are actually sub-chunks of the “RIFF” chunk.

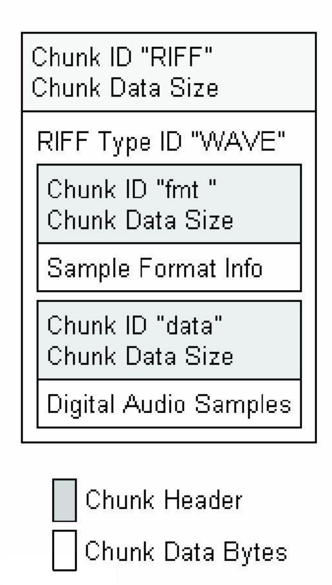

Figure 3.14: The RIFF structure used in the wave file structure.

It is important for correct operation that RIFF chunks should always be word aligned. This means that it should contain an even number of bytes. If this is not the case, an extra byte containing a zero value should be at the end of the chunk.

The RIFF type ID is "WAVE" which identifies this RIFF structure as being a wave file. This is followed by a format chunk containing information like the sample frequency and whether it is mono or stereo sound file. The information stored in the format chunk is displayed in Table 3.3.

Table 3.4: The information bytes stored under the format chunk.

Offset	**Size**	**Description**	**Value**
0x00	4	Chunk ID	"fmt" (0x666D7420)
0x04	4	Chunk Data Size	16 + extra format bytes
0x08	2	Compression code	1 – 65,535
0x0a	2	Number of channels	1 – 65,535
0x0c	4	Sample rate	1 – 0xFFFFFFFF
0x10	4	Average bytes per second	1 – 0xFFFFFFFF
0x14	2	block align	1 – 65,535
0x16	2	Significant bits per sample	2 – 65,535
0x18	2	Extra format bytes	0 – 65,535

The number of channels specifies how many separate signals that are stored in the wave data chunk. If the number of channels is 1 it is a mono signal, while a value of 2 signifies a stereo signal. The designed system will store a mono channel signal. The sample rate indicates how many times ADC samples of the sound signal are taken in one second. Our system will operate at a sampling rate of 3 kHz. The block align value indicates the number of bytes per sample and can be calculated by using Equation 3.10.

$$\mathrm{BlockAlign} = (\mathrm{BitsPerSample} \div 8) \times \mathrm{NumChannels}. \tag{3.10}$$

The average bytes per second value indicate how many bytes of data must be streamed to the digital to analogue converter of the PC per second in order to play the wave file and may be calculated Equation 3.11.

$$\mathrm{BytesPerSecond} = f_s \times \mathrm{BlockAlign}. \tag{3.11}$$

Bits per sample indicate with how many bits the ADC represents the sample taken from the analogue signal. The elephant recording system has a 16-bit ADC and no extra format bytes will be used.

The data chunk contains the actual digital sound data. When samples are represented with 8-bits, they are specified as unsigned values. All other sample bit-sizes are specified as signed values. For example a 16-bit sample can range from -32,768 to +32,767 with a mid-point (silence) at 0. This implies that the value read from the 16-bit ADC in the elephant recording system should be converted to a 2's complement digital number before being stored.

3.3.5.6.2 Recording process After all the file system preparation has been done, the digital sound data may be recorded in sequence with only the wave file format's RIFF structure and time, date, temperature and position in between different files.

The desired sampling frequency of the recording is achieved by triggering an internal interrupt at a frequency of 3 kHz. When this interrupt occurs the program jumps to the interrupt service routine where the ADC flag is set.

As can be seen in Figure 3.15 the main program loops continuously, testing for set flags. When a set ADC flag is detected, the ADC subroutine is called and the new values are stored in the RAM. If the file size is reached, the EOF flag gets set and the program initiates the start of a new file. A new directory entry needs to be written with an unique filename and the FAT tables need to be updated. A temperature measurement and GPS time and coordinates are written and the recording of the next file starts.

When the disk capacity is reached, the capacity flag is set and the recording process will start anew on the next disk. After all disks' capacities have been reached the program goes into a never ending loop waiting for the collar to be retrieved.

3.3.5.7 Thermometer operation

The digital thermometer communicates via a one-wire interface. The I/O pin of the microcontroller that communicates with the thermometer needs to change frequently from an input pin to an output pin to receive data and give commands respectively.

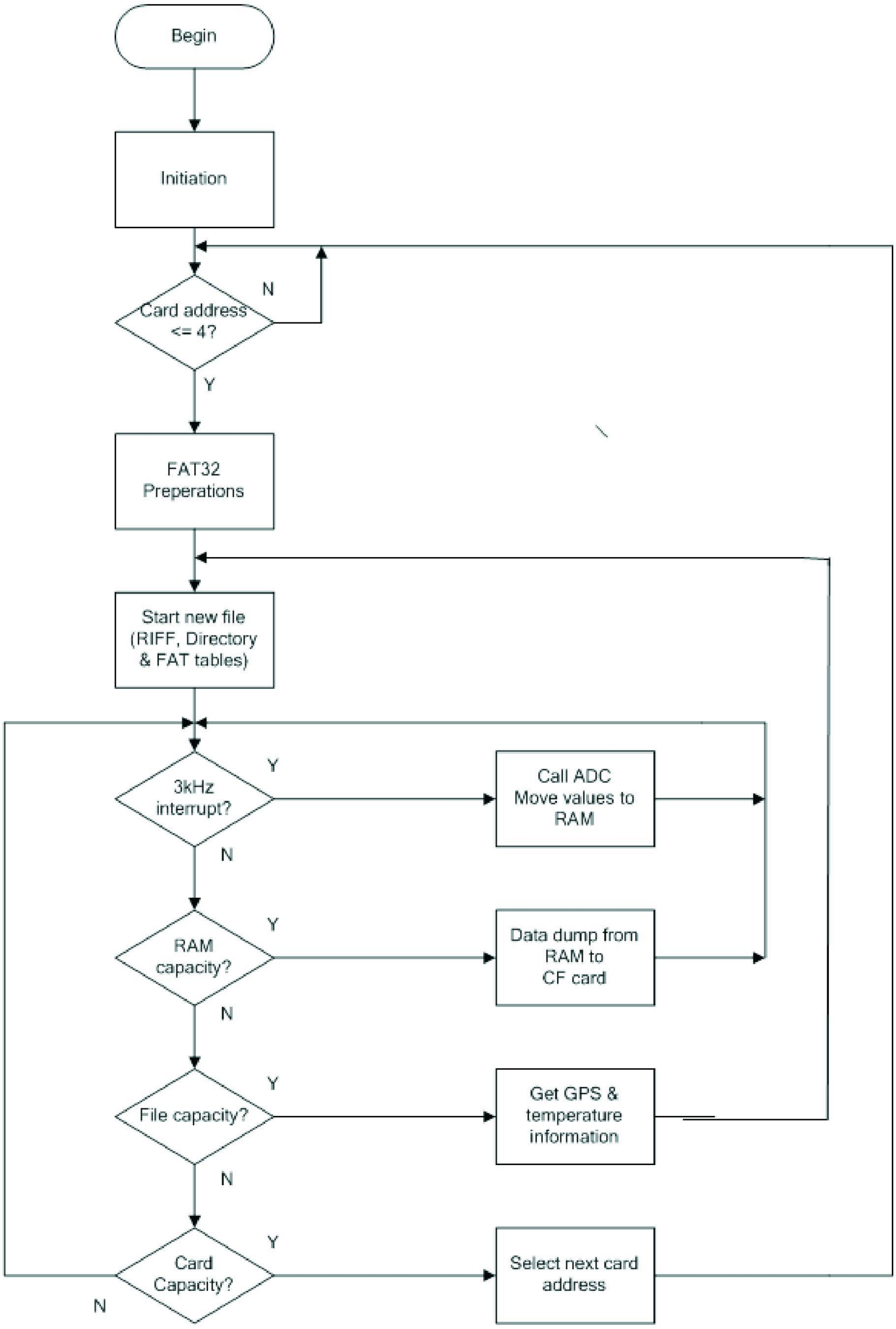

Figure 3.15: The flow diagram on the right describes the interrupt service routine while the diagram on the left is the main program loop reacting on the flags that is set in the interrupt service routine.

Each set of commands begins with a reset pulse after which two instructions are given: CONVERT (convert the outside temperature to a 12-bit digital value) and SEND (send the information) at which point the I/O pin of the microcontroller is set to an input pin to receive the data. Figure 3.16 shows the flow diagram for both the thermometer and the GPS module.

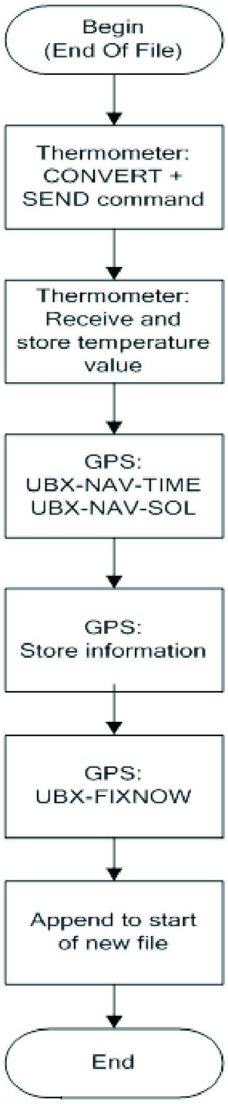

Figure 3.16: A flow diagram describing thesteps in the GPS/Thermometer subroutine.

3.3.5.8 GPS operation

The GPS device is equipped with two USART ports. The PIC16F877 is also equipped with an USART port and communication is established using the RX and TX pins of both devices.

An U-blox customized GPS protocol called UBX is used. An UBX command can be sent to request positional coordinates as well as time and date information (UBX-NAV-SOL and UBX-NAV-TIME respectively). Using the UBX-FIXNOW command the GPS device can be put in a low power mode. As shown in Figure 3.16, the GPS is only waked up for a short while to retrieve time, date and coordinate information and is then put back into a low power mode.

3.4 MECHANICAL DESIGN

The objectives with the mechanical design of the collar can be summarized as follows:

1. The memory cards should be easy to access and remove.
2. The electronics should be protected from physical impact.
3. The electronics should be protected from moisture (watertight).
4. When packaged, the microphone should be nearly as sensitive to sound as before packaging (see Chapter 4).
5. The microphone should not be obstructed by mud for extended periods of time (smooth surface above microphone where mud can dry and peel off).

3.4.1 Packaging of electronics

An enclosure for the electronics, shown in Figure 3.17, was manufactured. The dashed line represents a cover panel of 5 mm thick polycarbonate. Polycarbonate has excellent mechanical strength properties, but additionally its smooth surface made it an attractive option.

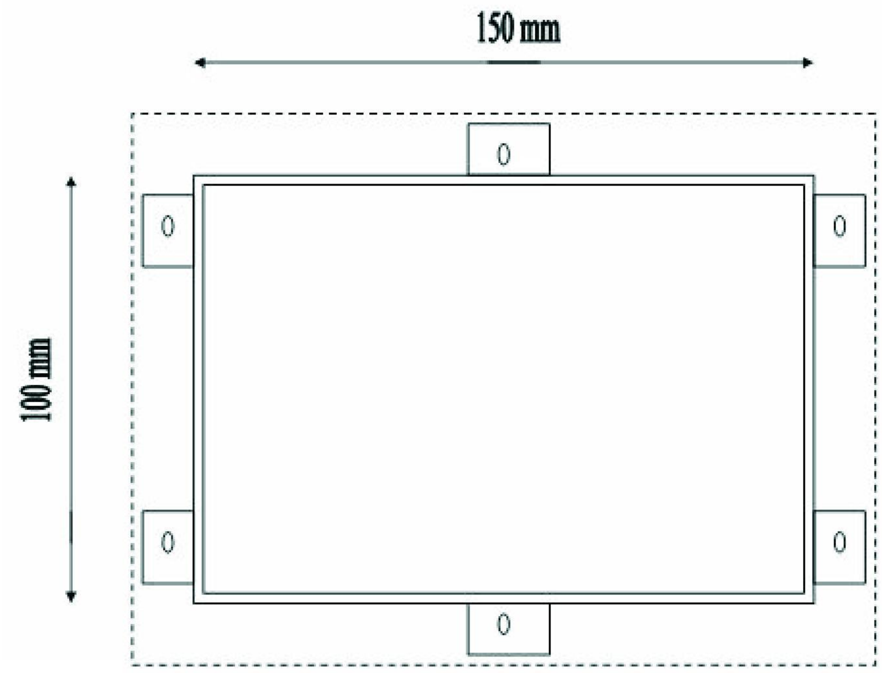

Figure 3.17: Drawing of the enclosure housing the electronics.

The side panels of the enclosure are made of 3 mm Perspex, and six additional Perspex bars were mounted onto the side panels to provide pillars onto which the cover panel can be fastened by screws. Six self-tapping screws are used. Silicon is used to obtain watertight seals all round, including the mounting of the cover panel onto the enclosure. The Perspex pillars also provide a binding surface for the plastic compound that is used to mould the enclosure into the collar.

The electronics boards were encased in silicon within the enclosure to ensure good shock absorbance. The entire enclosure of Figure 3.17 was moulded into the collar with a hard plastic epoxy as shown in Figure 3.18 The plastic epoxy was used instead of dental acrylics, because the acrylics are prone to cracking when rigid objects are moulded within it.

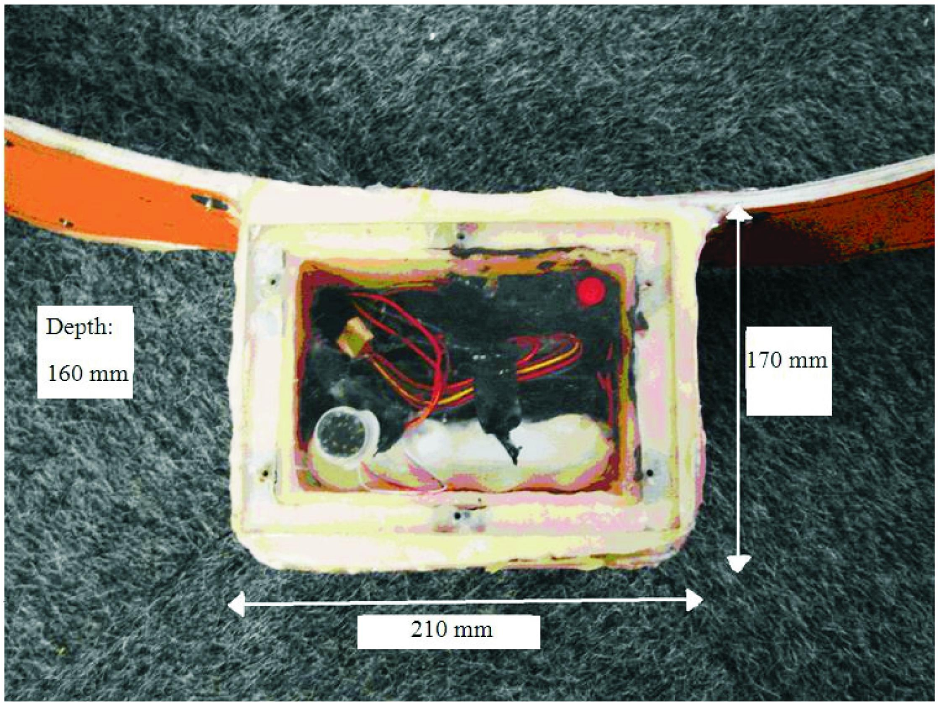

Figure 3.18: A photograph of the collar after the enclosure with the electronics has been mounted into the collar. The weight of the entire collar, including electronics, batteries and packaging is 8.7 kg.

3.4.2 Conclusion

First, the packaging of the electronics was completed to achieve the first three objectives given in 3.4.1. Different methods were experimented with to find an optimal way of achieving the last two objectives. These experiments will be discussed in Chapter 4.

3.5 AUTOMATIC DETECTION AND PITCH TRACKING OF ELEPHANT RUMBLES

This section discusses the development of an automatic elephant rumble detection algorithm. An analysis of recorded elephant vocalizations are made and elephant rumbles are visually detected to acquire an understanding of the nature of elephant rumbles (3.5.1). The development of the algorithm from speech processing techniques is discussed in Section 3.5.2 and finally the functional steps for the implementation of the algorithm in software are discussed in Section 3.5.3.

3.5.1 Analysis of previously recorded data

It was shown in Chapter 2 that elephant vocalizations are identified by experts using the Raven software tool. As a starting point for the development of an automatic elephant rumble detection algorithm, elephant rumbles were first detected visually from previously recorded data. This exercise provided a better insight towards the nature of elephant rumbles which assisted in development of a detection algorithm.

Elephant recordings that had been recorded by a handheld recorder in the Kruger National Park in 2003 by the research group of William Langbauer Jr were provided for use in this study. The recordings were made under a range of different circumstances and in different locations within the southern part of the park. Information about some of the observations that were made while the file were being recorded was also given which helped to locate files that could contain elephant vocalizations. A stepwise method for manually detecting elephant rumbles within recordings are presented in this section.

Raven is a computer program that can be downloaded from the internet (www.brothersoft.com) and is used to draw a spectrogram of a sound file. All the spectrograms shown in this document were created using Raven software.

The exact locations of elephant rumbles within recordings can be identified using the following stepwise method.

1. Open the sound file in Raven to show a spectrogram of the sound.
2. Zoom in to the lower 200 Hz part of the spectrogram.
3. Adjust the program so that 7000 data points are used within the timing window used to calculate the spectrogram. This will ensure that a sharp image of the spectrogram can be seen at low frequencies. A spectrogram now typically appears as shown in Figure 3.19.
4. The spectrogram of a recording with low frequency noise (like the one in the example) may typically appear too bright; the brightness is adjusted to a level at which the information contained in the spectrogram is clearly visible (see Figure 3.20).
5. Areas where low frequency components occurs can be visually identified as horizontal lines on the spectrogram. In this example, a number of areas exist where low frequency energy occurs. The frequency components of both engine noises and elephant rumbles can be observed.
6. The recording is then played back at 10X normal speed. Playing the recording back at faster speeds puts the sounds in the audible range which helps to distinguish between elephant rumbles and other infrasonic sounds such as the engines of motor vehicles which can easily be recognized. In Figure 3.20 the parts of the spectrogram containing the infrasonic elephant rumbles are indicated with red lines.
7. Note that the elephant rumbles has a number of higher harmonics that can be seen as parallel horizontal lines on the spectrogram. This is the main feature that separates the elephant rumbles from other sounds within the infrasonic frequency band.

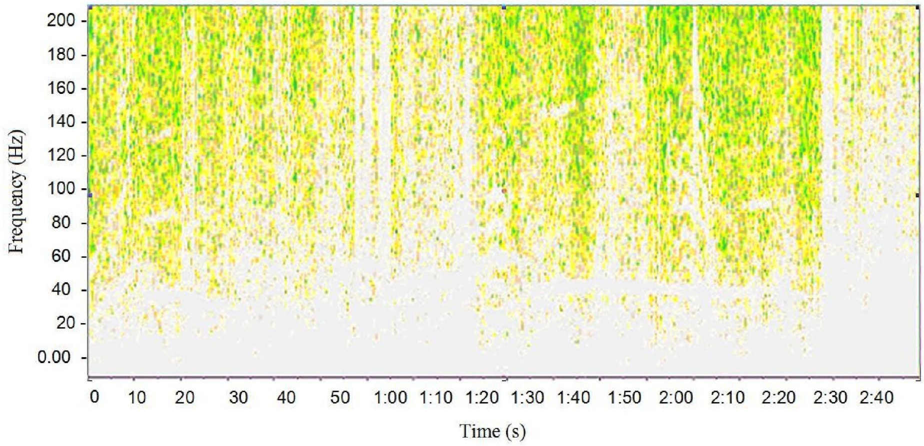

Figure 3.19: The spectrogram of a recording containing subsonic elephant rumbles. This spectrogram is shown at its default brightness.

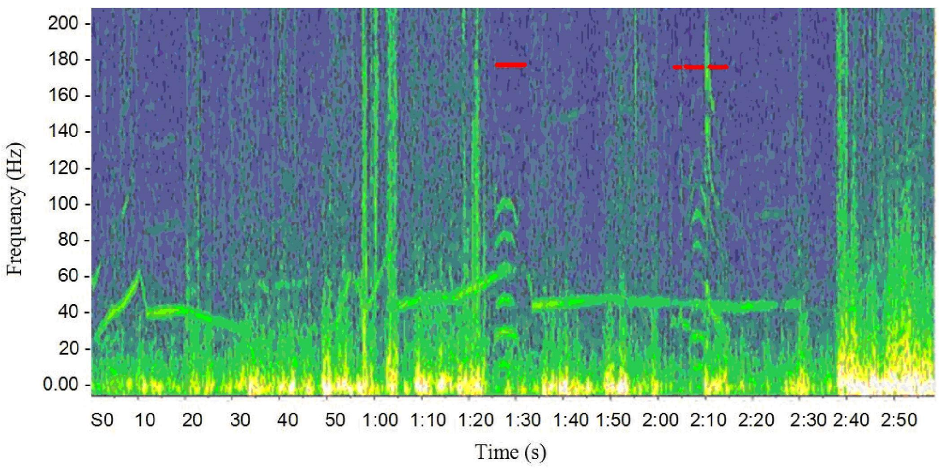

Figure 3.20: The spectrogram of a recording containing subsonic elephant rumbles. The parts of the spectrogram that contains the infrasonic elephant rumbles have been indicated with red lines.

After repeating the procedure mentioned above for a number of different recordings, it became clear that the infrasonic elephant rumbles have some specific characteristics:

1. Rumbles consist of a base frequency with a number of upper harmonics.
2. Rumbles typically have (in the provided recordings) a fundamental frequency of between 14 Hz and 28 Hz.
3. Rumbles have typical durations of between 0.5 to 10 seconds.
4. The majority of rumbles start at a certain pitch that becomes higher in frequency peaking in the middle of the call. The rumble ends at approximately the same pitch that it has started with.

3.5.2 Method

The VAD techniques discussed in the Chapter 2 were developed to identify the presence of human speech in a signal. This section will discuss the details of an algorithm devised to automatically detect the presence of infrasonic elephant rumbles. The proposed algorithm uses some of principles of the model by Wu *et al.* (2003), but some of the techniques have been modified and implemented differently. The final stage, where pitch tracks are determined and the final estimates of the rumble locations are calculated makes use of alternative methods altogether.

Figure 3.21 shows the schematic diagram of the proposed signal processing. A single channel input sound enters the system. This sound is divided into different channels by the filter bank. The sound channels are then windowed and the normalized autocorrelation of each window is calculated. Channels with the best signal-to-noise ration are then selected and these are added together to form a final correlogram with which the pitch of that particular time window is estimated. Finally, all the pitch information for every time window is analysed by the last block to form pitch tracks and estimate the location of possible rumbles.

A detailed description of each of the signal processing stages is given below.

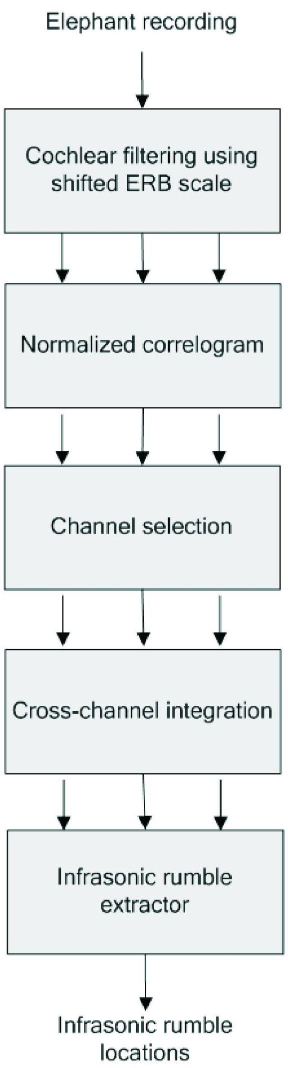

Figure 3.21: The schematic diagram of the adapted model used for elephant rumble identification

3.5.2.1 Input

The input to the system will be single channel recorded sound data from either hand-held recorders or elephant collars.

3.5.2.2 Filter bank

The sound enters the filter bank where it is then filtered into sub-bands. Cochlear filtering, a bank of fourth-order gammatone filters evenly distributed on the ERB (Equivalent Rectangular Bandwidth) scale, has been proposed if human speech had to be identified. The use of gammatone filters is a sensible choice where the determination of pitch is concerned since the particular sensitivity of the human cochlea to pitch was used as a template for the development of this filter. The pitch of human speech is typically close to 100 Hz for a males and 200 Hz for females, while elephant rumbles have pitches of between 14 Hz and 25 Hz. An intuitive modification in designing an appropriate filter bank for elephant vocalizations was to use a shifted ERB scale. The ERB (Equivalent Rectangular Bandwidth) scale was shifted by a factor of 10 and pilot tests showed that the rumble detection algorithm performed better than when using the normal human ERB scale. A bank of 32 filters were used with centre frequencies evenly distributed on the shifted ERB scale between 12 Hz and 30 Hz. Figure 3.22 shows the filter bank with the 32 gammatone filters evenly distributed on the shifted ERB scale. The figure was generated in Matlab. Gammatone filters are defined in Katsiamis, Drakakis and Lyon (2007), but for the present algorithm the implementation of the filters was done using the appropriate function from the Matlab Signal Processing Toolbox.

3.5.2.3 Normalized autocorrelation

Autocorrelation is a statistical concept that expresses the degree to which the value of an attribute at spatially adjacent points varies with the time separating the observations. The normalized autocorrelation of each of the channels, c, is calculated using Equation 3.12, taken from Wu *et al.* (2003), where N is the number of samples used for the calculation of the autocorrelation. A window size of 40 ms was used, which corresponds to 128 samples in a signal sampled at 3 kHz. The choice of N

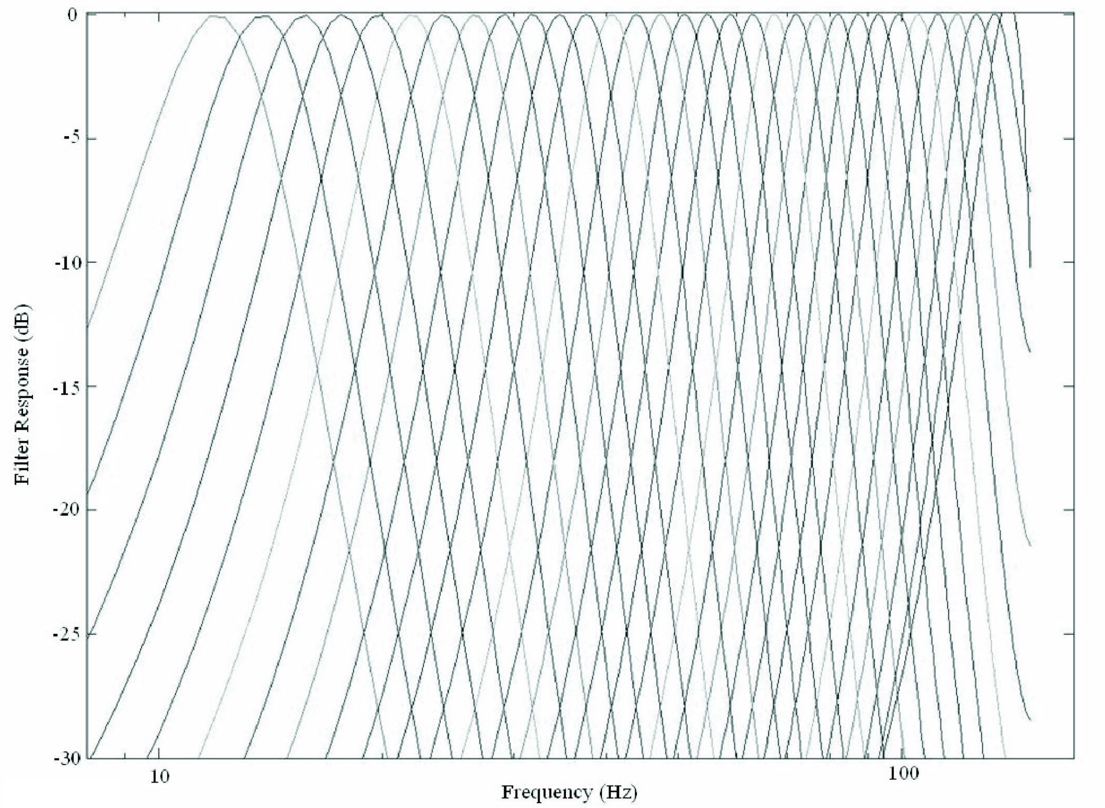

Figure 3.22: Matlab generated response of a filter bank consisting of 32 gammatone filters evenly distributed on the shifted ERB scale.

determines the resolution of the processed data. The output of the filter is denoted as $r(\mathrm{t})$. The number of lag steps, T , used within the autocorrelation determines the lowest frequency that may be detected and needs to be at least 300 samples to detect a frequency of 10 Hz. The position in the centre of the window is denoted as j. The autocorrelation is then

$$A\left(c,j,\tau\right)=\frac{\sum_{n=-N/2}^{N/2}r\left(c,j+n\right)r\left(c,j+n+\tau\right)}{\sqrt{\sum_{n=-N/2}^{N/2}r\left(c,j+n\right)}\sqrt{\sum_{n=-N/2}^{N/2}r^{2}\left(c,j+n+\tau\right)}}. \tag{3.12}$$

The numerator is the autocorrelation, while normalization is performed by the denominator.

The correlogram of a channel containing a pure tone is shown in Figure 3.23. It can be seen from the figure that the correlogram of a pure tone is also a periodic waveform with amplitude of one. The number of lag steps at which the first positive peak is located can be used to determine the frequency of the input signal using Equation 3.13.

$$f=\frac{F_s}{p}, \tag{3.13}$$

with F_s the sampling frequency of the input signal and p the number of lag steps before the first positive peak occurs. This equation shows that as the samples used in the autocorrelation are shifted over each other, a maximum peak will occur when the original signal and the shifted signal overlaps.

3.5.2.4 Channel selection

This subsection will explain how the amplitude of the correlograms of each sub band is used to select only sub bands with a good SNR. It should be remembered that the filter bank creating the sub bands are spaced close enough that only a single harmonic component of a harmonic sound can be present in any given sub band.

Figure 3.23 illustrated that a perfectly sinusoidal signal produces a normalised auto-correlogram with peak values of 1. A correlogram of a pure tone signal with added white Gaussian noise was generated in Matlab and is shown in Figure 3.24. This figure shows that the correlogram is still periodic, but with a lower amplitude. As more noise

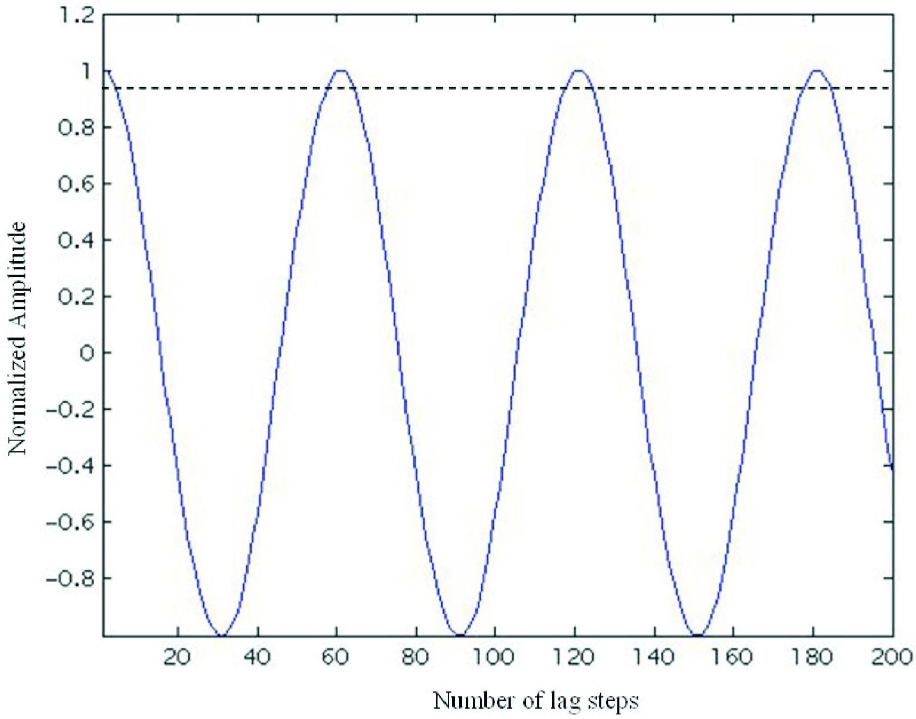

Figure 3.23: The correlogram of a pure tone signal.

is added to the signal, the amplitude of the waveform in the correlogram decreases. The correlogram of a signal consisting of white Gaussian noise is shown in Figure 3.25. It can be seen that the waveform in this figure is non-periodic and at no point has amplitude of more than 0.2 which demonstrates the relationship between the peak values of the correlogram and the periodicity of the applied signal.

The autocorrelation calculation is performed on each of the channels originating from the filter bank. The amplitude of the first positive peak in the correlogram of each channel gives an indication of the frequency component present in that channel and the maximum amplitude of the positive peaks in the correlogram gives an indication of the amount of noise present in the channel for the analyzed segment. Only channels with maximum normalized amplitude of more than 0.945 is selected (this threshold was established empirically in Wu *et al.*, 2003). This method of channel selection enables the algorithm to detect faint calls in recordings with severe broadband disturbances, since only channels with energy at specific pure frequencies are selected regardless of the strength of the component.

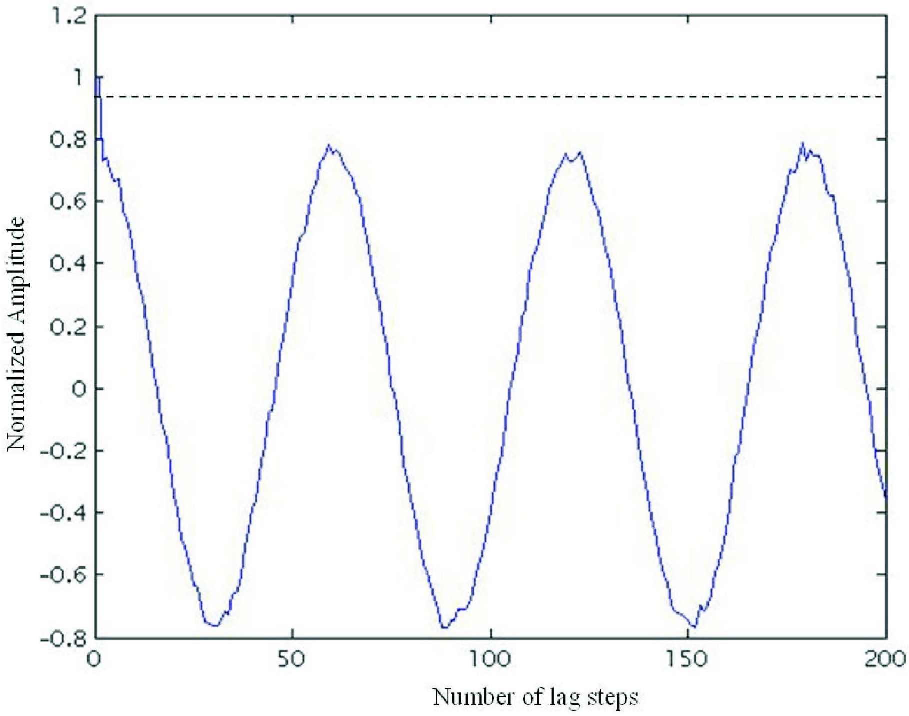

Figure 3.24: The correlogram of a pure tone signal with some white Gaussian noise added.

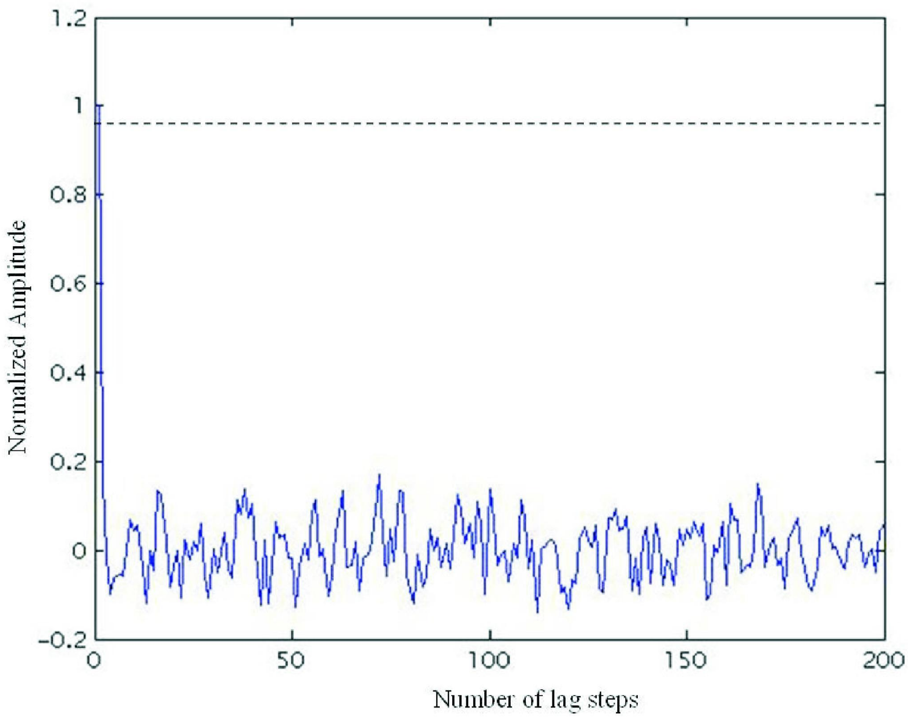

Figure 3.25: The correlogram of white Gaussian noise signal.

3.5.2.5 Channel integration and determination of pitch information

When a segment of sound is analyzed, the sound is divided into sub-bands by the filter bank. Channels with low SNR are selected and all the selected channels are added together to form a summated correlogram. The largest positive peak in this summated correlogram is selected, and the corresponding pitch is calculated using Equation 3.13. The way in which this operation detects the pitch of a harmonic signal will be explained with the aid of graphs that have been created in Matlab.

Figure 3.26 shows the spectrogram of a segment of sound with a pitch of 25 Hz consisting of the fundamental frequency of 25 Hz and two harmonic frequencies at 50 Hz and 75 Hz. All the harmonics in the sound possess equal energy.

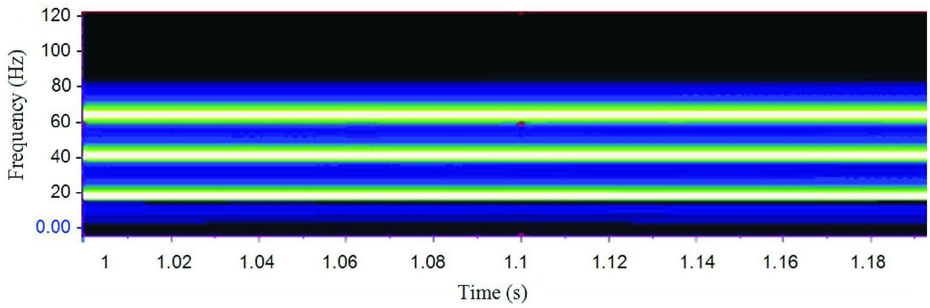

Figure 3.26: Spectrogram of a sound consisting of a 25-Hz fundamental frequency and two harmonics at 50-Hz and 75-Hz.

Figure 3.27(a) shows the normalized correlogram for a channel containing the 25 Hz signal. It can be seen that the first non-zero peak frequency occurs at 120 lag steps, which corresponds to 25 Hz when using Equation 3.13. In the same way Figure 3.27(b) shows the correlogram of a channel containing the 50 Hz signal. In this case the first non-zero peak frequency occurs at 60 lag steps, which correspond to 50 Hz. Figure 3.27(c) shows the normalized correlogram for a channel containing the 75 Hz signal. It can be seen that the first non-zero peak frequency occurs at 40 lag steps which corresponds to 75 Hz according to Equation 3.13. Figure 3.27(d) shows the summed correlogram obtained by summating the correlograms of the 25 Hz, 50 Hz and 75 Hz signals. This figure shows that the positive peak in the summed correlogram with the highest amplitude occurs at 120 lag steps. Equation 3.13 estimates the pitch of the analyzed sound segment as 25 Hz. As can be seen from this example, the presence of higher harmonics in a sound signal assists to strengthen the peak in the summed correlogram that indicates the pitch of the sound.

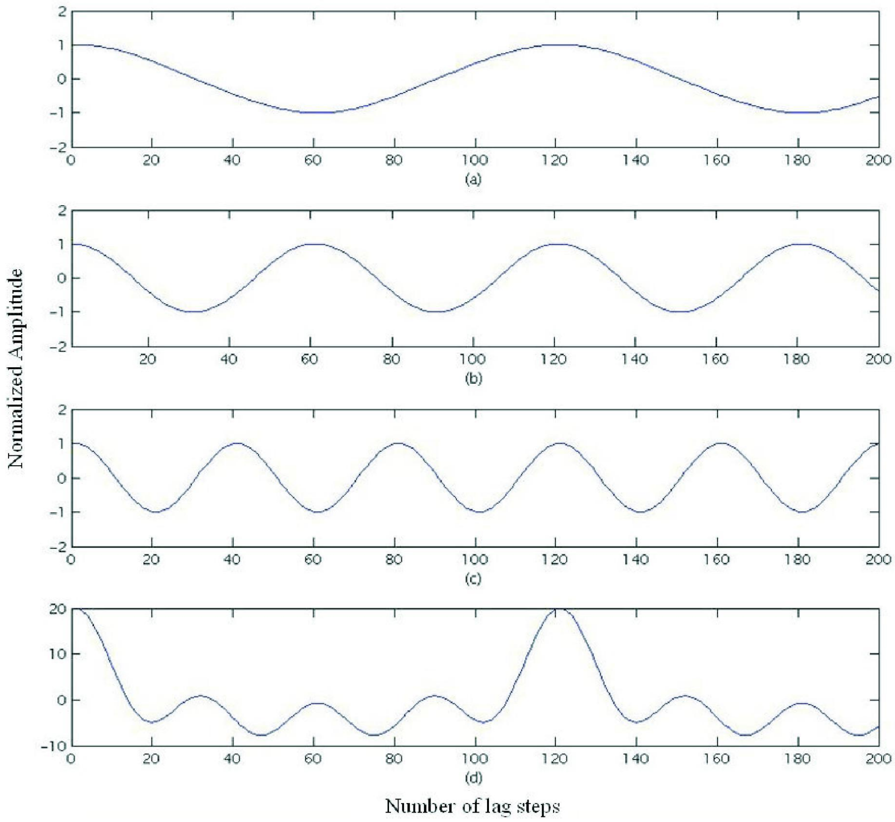

Figure 3.27: (a) Correlogram of a channel containing the 25-Hz fundamental frequency; (b) Correlogram of a channel containing the 50-Hz harmonic frequency; (c) Correlogram of a channel containing the 75-Hz harmonic frequency; and (d) Summed correlogram with the maximum positive peak indicating the pitch of the sound.

Figure 3.28 shows the spectrogram of a segment of sound with a pitch of 25 Hz consisting of only the first three harmonics namely 50 Hz, 75 Hz and 100 Hz.

Figure 3.29(a) shows the normalized correlogram for a channel containing the 50 Hz signal. It can be seen that the first non-zero peak frequency occurs at 60 lag steps, which corresponds to 50 Hz when using Equation 3.13. In the same way Figure 3.29(b) shows the correlogram of a channel containing the 75 Hz signal. In this case the first non-zero peak frequency occurs at 40 lag steps which correspond to 75 Hz. Figure 3.29(c) shows the normalized correlogram for a channel containing the 100 Hz signal. It can be seen that the first non-zero peak frequency occurs at 30 lag steps,

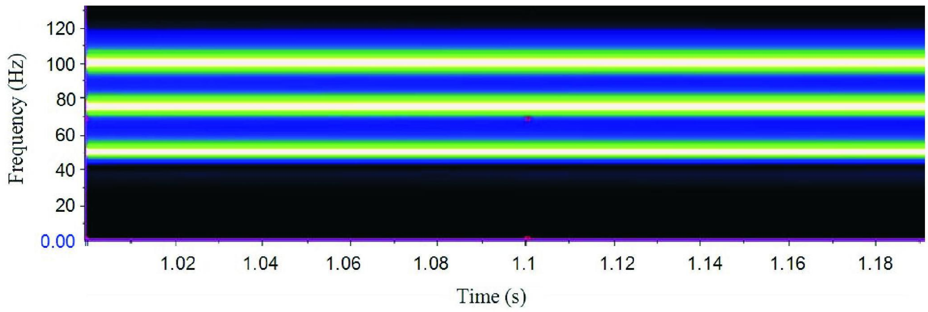

Figure 3.28: Spectrogram of a sound with 25-Hz pitch but with only the first three harmonics present while the fundamental frequency is not present.

which corresponds to 100 Hz according to Equation 3.13. Figure 3.29(d) shows the summed correlogram obtained by summating the correlograms of the 50 Hz, 75 Hz and 100 Hz signals. As can be seen in the figure, the positive peak in the summed correlogram with the highest amplitude occurs at 120 lag steps. The pitch of the analyzed sound segment can be calculated as 25 Hz using Equation 3.13. Note that the correct pitch can still be determined from the summed correlogram even though the fundamental frequency is not present in the signal.

3.5.2.6 Determination of rumble locations and tracking of pitch

Consider the segment of sound shown in the spectrogram of Figure 3.30. It can be seen that an elephant rumble occurs from around 2 seconds up to 5.5 seconds of the recording. After performing the signal processing steps on this sound segment, the pitch information shown in Figure 3.31 is calculated from the maximum peak in each of the summed correlograms. It can be seen from the figure that this pitch information resembles a noisy signal in the area where the rumble is not present, but a smoother pitch track in the area where the rumble occurs. There is however some points of the pitch information that does not follow the pitch track.

The final step of the algorithm should be to extract only the pitch tracks of elephant rumbles while ignoring all the other pitch information. Some additional processing needs to be done to achieve this. When comparing Figure 3.30 and Figure 3.31, note that the section(s) of the estimated pitch where rumbles are present, can be clearly seen as a smooth pitch track, containing perhaps some discontinuities. After

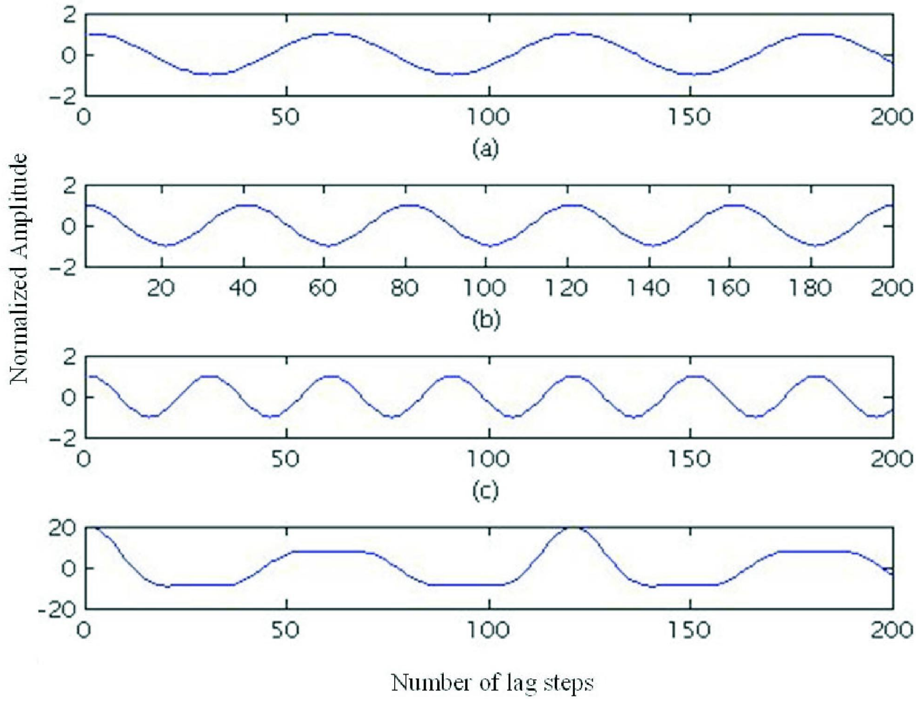

Figure 3.29: (a) Correlogram of a channel containing the 50-Hz harmonic frequency; (b) Correlogram of a channel containing the 75-Hz harmonic frequency; (c) Correlogram of a channel containing the 100-Hz harmonic frequency; and (d) Summed correlogram with the maximum positive peak indicating the pitch of the sound.

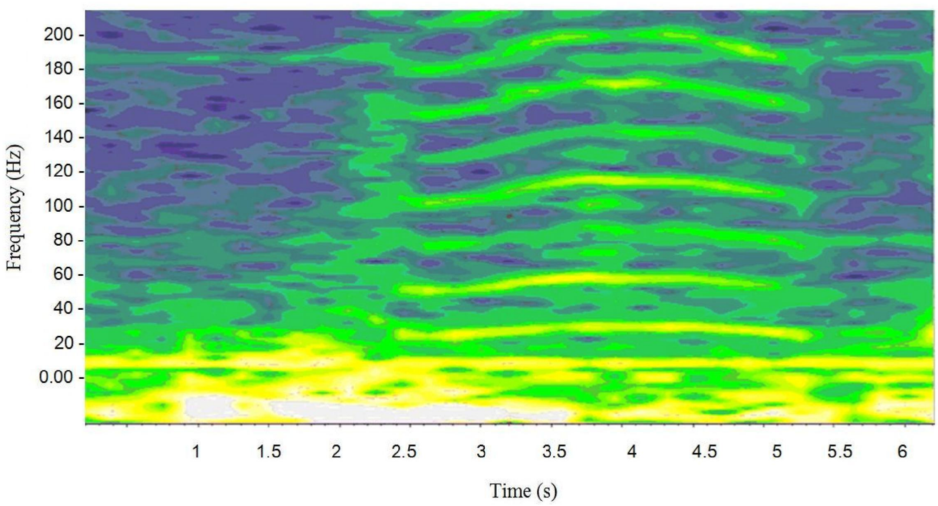

Figure 3.30: Spectrogram of a sound segment containing an infrasonic elephant rumble.

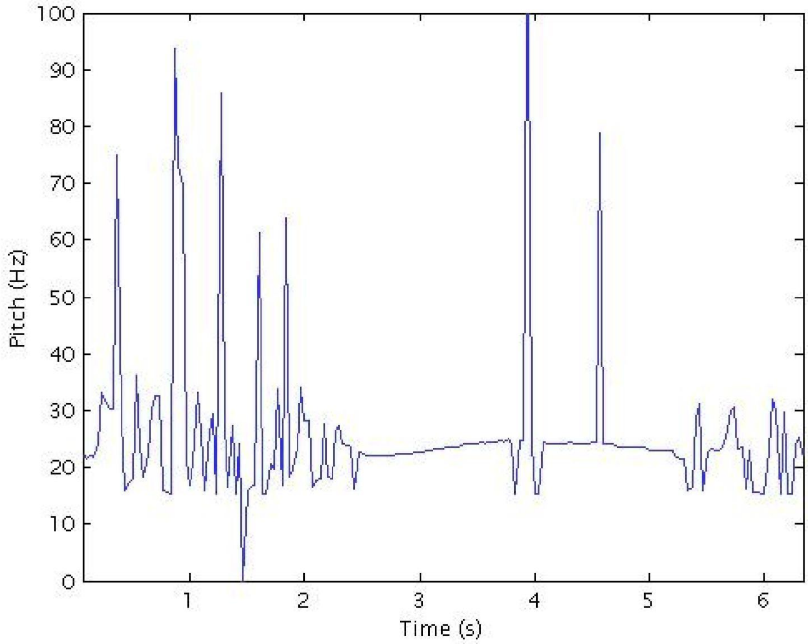

Figure 3.31: Pitch information as calculated for the sound segment shown in Figure 3.34 before pitch track extraction algorithm have been run.

verifying these results with a number of clearly defined elephant rumbles it appears to be a simple matter to determine the locations of the elephant rumbles from the pitch information array by visual inspection of the pitch array.

A number of techniques may be considered to identify the location of elephant rumbles automatically. These include computationally expensive techniques involving the use of HMMs (Paris and Jauffret, 2003; Xie and Evans, 1993) and neural networks (Adams and Evans, 1994).

A simple algorithm (with light computational demands) that mimics the way that one would identify the calls visually were developed. Figure 3.32 describes this algorithm. The algorithm scans through the pitch information array (created in Matlab), looking for smooth parts that indicates the presence of a rumble. If two successive samples differ with less than S Hz it is considered "smooth". If a number of Pf smooth samples occur consecutively, that section of the pitch information array is marked as a pitch track. If a smooth pitch track ends with a discontinuity it is tested whether there is evidence of the same the pitch existing track further on in the pitch information array.

The algorithm will look ahead for F samples in a "beam" with a width defined by θ as shown in the figure. If another point of the pitch track is discovered in that area, it is assumed that the discontinuity can be ignored and the pitch track is extended to that point. When no further evidence of the pitch track can be found, the location of the pitch track is stored and the algorithm continues searching through the pitch information array as before.

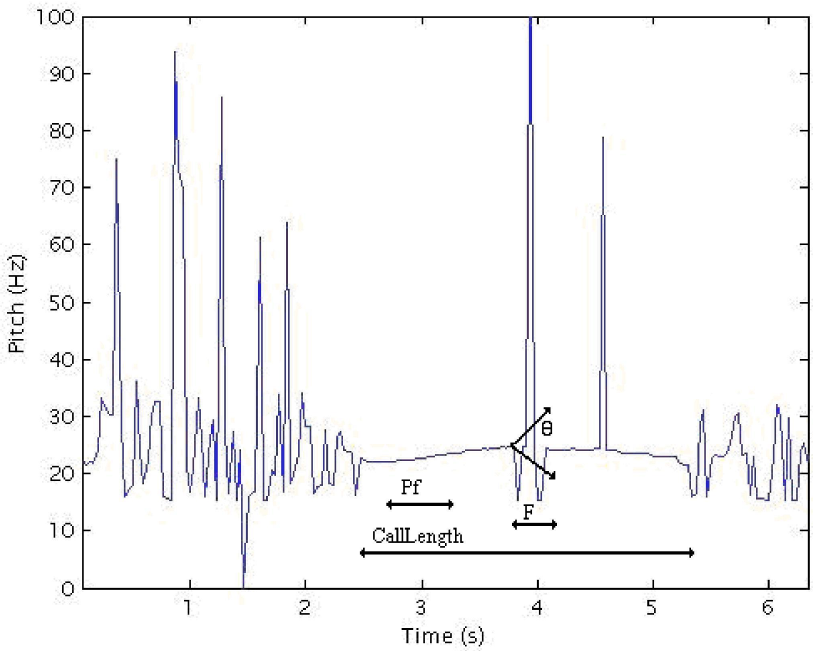

Figure 3.32: Parameters of the pitch track extraction algorithm.

This process is repeated backwards (from the last point of the pitch information array to the first point) to ensure that complete pitch tracks are identified. The pitch tracks from the forward and backward sweeps are combined to estimate the locations of the infrasonic elephant rumbles. Figure 3.33 shows the final output generated after the pitch information array has been processed by the algorithm. In this example, a near perfect estimate of the location of the rumble was obtained.

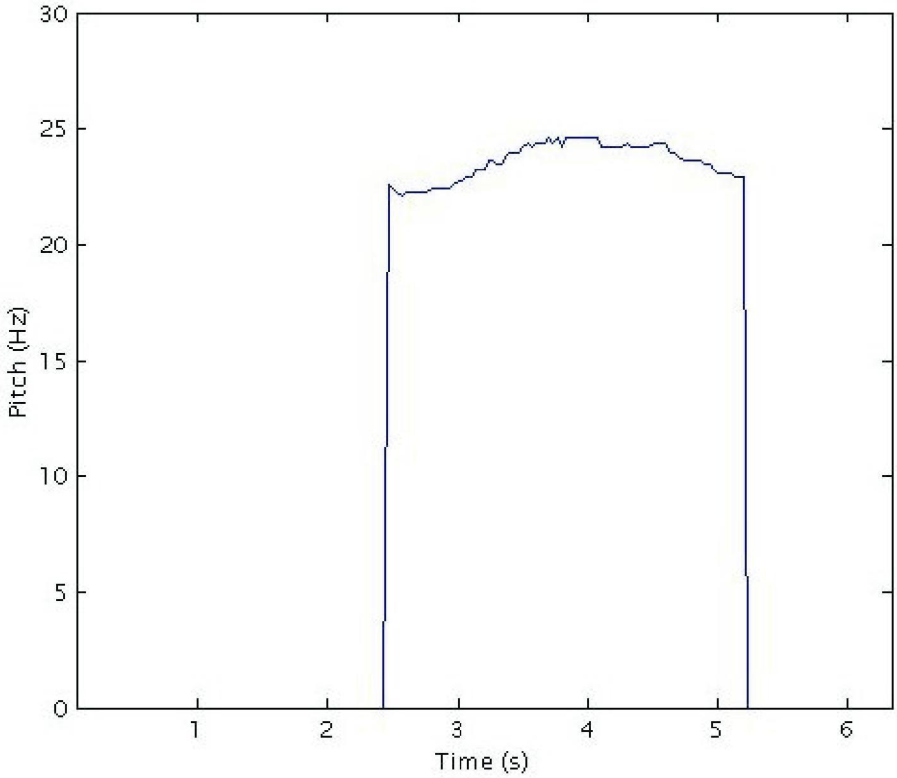

Figure 3.33: Final output of the system when the sound segment shown in Figure 3.34 is analyzed.

3.5.3 Implementing the algorithm in program code

Up to this point, the discussion of the algorithm was from a theoretical viewpoint. Some practical problems needed to be solved when implementing the algorithm in software. The elephant rumble detector and pitch tracking algorithm was implemented in the Matlab programming language. Figure 3.34 shows the basic block diagram describing how the algorithm was implemented in software. A functional description (not related to a specific programming language) of the implementation of the algorithm in software will be given.

Firstly, the wave file containing the recorded sound has to be read into memory. If the sampling frequency of the file is different from 3 kHz it needs to be re-sampled to a rate of 3 kHz. It is important to realize that the single channel sound input is divided into 32 (or more) channels at the filter bank. Each of these channels needs as much memory space as the single channel wave file. If the wave file is relatively large (exactly when a file will be too large depends on the resources available on the specific computer on which the algorithm needs to be implemented), memory management needs to be done to ensure that the estimation process runs at optimal speed and so that the computer does not run out of RAM memory.

The implementation of the memory management is shown in the block diagram of Figure 3.35. A buffer length of *BufferSize* is defined by the user based on the amount of RAM that the computer has access to. If the wave file is smaller than *BufferSize* no memory management is done and the whole file is processed at once. If the wave file is larger than *BufferSize* only the first *BufferSize* samples of the original wave file is processed and the program returns to the starting point of this diagram to process the next segment of the wave file. As soon as the remaining number of samples that needs to be processed is less than the size of the buffer, the value of *LastBuffer* is set to 1 to let the program know that it is now processing the last buffer and it will not be necessary to return to this point of the program again. Managing the computer's memory in this way ensures that the performance of the computer will be the same for large files as for small files.

The filter bank consists of a number of fourth order gammatone filters with centre frequencies distributed evenly on the shifted ERB scale as discussed in the previous section. These filters are implemented by using the Auditory Toolbox in Matlab. This

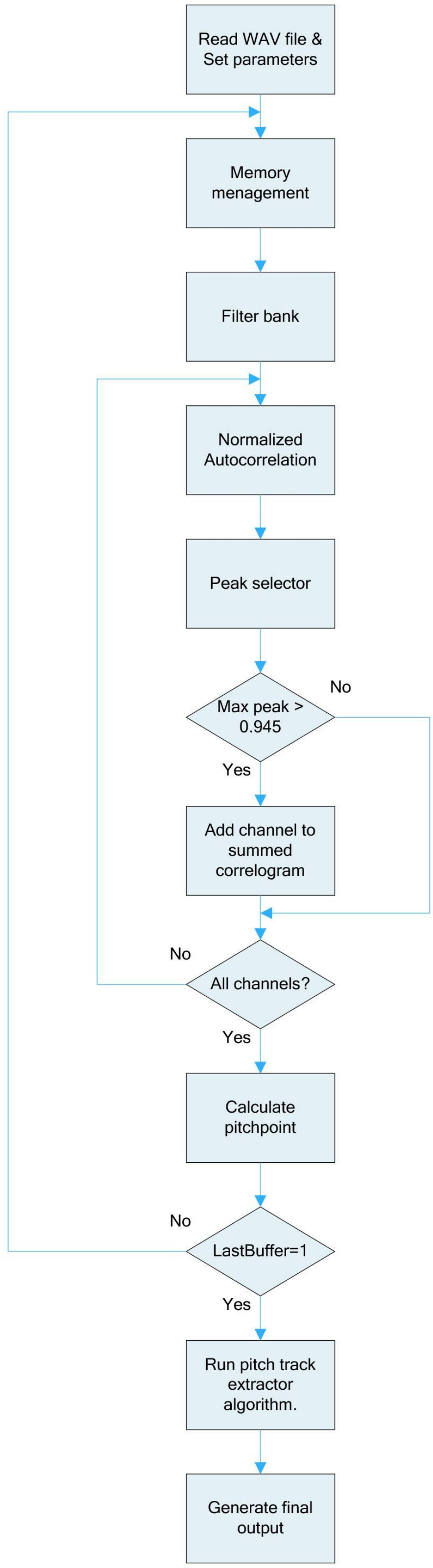

Figure 3.34: Block diagram showing the basic implementation of the algorithm.

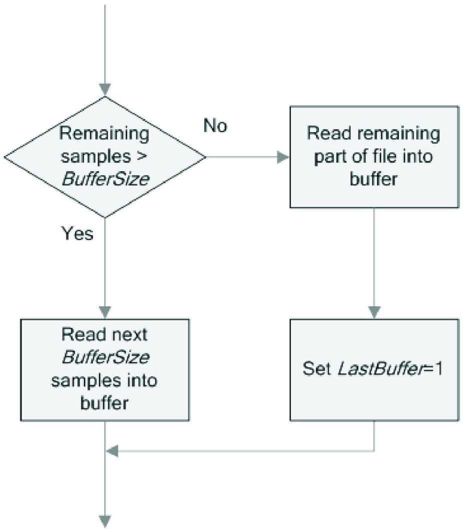

Figure 3.35: Block diagram showing the operation of the memory management stage of the system.

toolbox can be run an any version of Matlab and can be downloaded from the internet without cost. The centre frequencies range from 12 Hz to 150 Hz and 32 filters are used.

Each of the channels resulting from the filter bank is windowed into smaller segments. The windowed segments have a size of 100 samples (corresponding to 33.33 ms). The normalized autocorrelation function given in Equation 3.12 is performed on each windowed segment of each channel. The autocorrelation is done over 128 samples. The number of lag steps calculated in each autocorrelation is 250. According to Equation 3.13, this means that a pitch as low as 12 Hz can be detected. The window length, the number of samples over which the autocorrelation is done and the number of lags steps to be calculated are defined as parameters in the program and can easily be changed by the user of the program.

The previous section indicated how it is determined whether a channel is noisy or clean. The positive peaks of the correlogram are tested and if the peak has amplitude of more than 0.945 it is considered noise-free. A peak detection algorithm was developed to analyze each correlogram. The pseudo code for the peak detector is shown in Figure 3.36.

```
n=1
If n<EndOfFile
        State=Falling
        If C(n)>C(n+1)
        Begin
                If Sate=Rising
                Begin
                        Peak(a)=n
                        State=Falling
                End
        End
        If C(n)<C(n+1)
        Begin
                If Sate=Falling
                Begin
                        State=Rising
                End
        End
End
```

Figure 3.36: Pseudo code showing the operation of the peak detection stage of the system.

The autocorrelation array is imported into the peak detector. Since the normalized autocorrelation always has amplitude of 1 at lag number 0, the autocorrelation curve will initially decrease monotonically at larger lag steps. The difference between successive samples is calculated. If the result is positive and the values are decreasing, it is known that the current position of the marker is still on the downward slope of a peak. If, however, the result is negative while on a downward slope, it is known that a valley has been reached. The state is now changed to rising and the point is marked as the starting point of a positive peak. If another positive peak is reached in this correlogram, this point is also marked as the end of the previous positive peak segment. The difference between the next two samples is now calculated. If the result is negative and the state is increasing, we know that we are still on the upward slope of a peak. If, however, the result is positive while the state is increasing we know that a positive peak has been reached. The state is now changed to falling and the point is marked as the location of a positive peak. The amplitude at this point is recorded (to establish whether the channel is noisy or not). An array with all the peak information stored is sent back to the main program. Using this information, the algorithm decides whether or not to add the channel to the summed correlogram for that specific time window.

The position of the maximum point in the summed correlogram gives the value of the most prominent pitch present in the sound window. This pitch information point is added to the pitch information array. This process repeats itself until the complete memory buffer has been processed. A new memory buffer is then processed and added to the pitch information array. When all the information in the sound file has been processed a complete pitch information array is available.

The theory behind the technique used for identifying infrasonic elephant rumbles from the pitch information array was discussed in the previous section. Figure 3.37 shows the pseudo code for the rumble extractor. Firstly the pitch information array is imported. The algorithm starts at position 1 of the pitch information array. If the absolute value of the difference between the value at the current point and the value at the subsequent point is less then *Thresh* the counter X is increased. If the difference between the value at the current point and the value at the subsequent point is greater than *Thresh* it shows that this part of the pitch information array is not smooth. If the value of X is greater than *Platform*, a long enough segment of the pitch information array was smooth to be considered as a pitch track. The program stores the starting value of this pitch track and now looks ahead into the pitch information array to search for evidence that the pitch track may continue. It will look forward into the array for up to *Flash* points in advance and test if any of the points differs by less *Frange* with the last point of the known pitch track. The value of *Frange* is changed at every data point according to the value of *angle*. If evidence of the continuation of the pitch track is found, the program will once again check *Frange* data points in advance. This process will continue until no further evidence of the pitch track can be found. The location of the identified pitch track is then stored and the program returns to its normal state of scanning through the pitch information array for evidence of a smooth pitch track. When the program has scanned through the complete pitch information array, the process is reversed and the program performs the same operation on the pitch information array, but now starting at the last data point and moving towards the beginning of the array. The information from the forward scan and the backward scan is then combined to give the estimated location of the elephant rumbles.

```
N=Nc
InRumble=1
For c=1 to Nc
Begin
        If A(j)-A(j+c)<=L
        Begin
                N=N-1
        End
End
If  N=0
Begin
        InRumble =0
        P(j…j+Nc)=A(j…j+Nc)
        J=j+Nc
        While InRumble ==0
        Begin
                If  A(j)-A(j+1)<L2
                Begin
                        P(j+1)=A(j+1)
                End
                        Else
                        Begin
                                while c<=Nc2 & A(j)-A(j+c)>L2
                                Begin
                                        c=c+1
                                End
                        End
                If c<> Nc2
                Begin
                        P(j…j+c)=A(j)
                End
                Else
                Begin
                        InRumble=1
                End
        End
End
```

Figure 3.37: Pseudo code showing the operation of the pitch track extractor stage of the system.

3.6 SUMMARY

This chapter describes the electronic design of an elephant recording collar by first giving the basic requirements of such an device and then proposing a concept design. The details of each of the modules shown in the concept design were discussed. A functional description of the microcontroller software needed to orchestrate the dataflow, file structure and file system was given. The objectives of the mechanical part of the collar were stated and the details of the mechanical design were discussed. In short, the development of a mechanically robust and watertight elephant recording collar designed to record infrasonic sound at a resolution of 16 bits and with a sampling rate of 3 kHz were discussed. The collar has a potential memory capacity of 128 GB onboard which allows for continuous data recording for nine months. The incorporation of a GPS device and a thermometer into the device have also been addressed.

The procedure for the visual detection of elephant rumbles from a spectrogram has been explained. The development of an automatic elephant rumble detection algorithm was discussed. Existing speech processing techniques, namely cochlear filtering and normalized autocorrelations were implemented and combined with a newly devised technique that resulted in an algorithm that can detect elephant rumbles from noisy recordings and also track the pitch of the rumble. A functional description of the implementation of the algorithm in software was given.

Chapter 4

RESULTS

4.1 CHAPTER OBJECTIVES

The design and implementation of both the electronic and mechanical parts of the elephant recording collar were discussed in Chapter 3. In this chapter, the results of the evaluation of both the elephant collar and the elephant rumble detection algorithm are presented.

The general tests that have been done on the elephant collar are presented in Section 4.2 while the collar's field tests are discussed in Section 4.3. Section 4.3 discusses the results obtained by the automatic elephant rumble detection algorithm.

The microphone placement experiments were done to find the best possible way to attach the microphone to the collar. The stability of the electronics in extreme temperatures needed to be verified and the power consumption needed to be tested to ensure that it was within the specified range. Field tests were done to test the ability of the collar to operate under physically demanding circumstances.

Several aspects of the automatic elephant rumble detection algorithm were tested. The algorithm's ability to detect harmonic sounds in noisy conditions were quantified which lead to a better understanding of the algorithm's abilities.

4.2 GENERAL ELEPHANT COLLAR TESTS

The microphone placement experiments (4.2.1) and temperature stability and power consumption experiments (4.2.2) are described in this section.

4.2.1 Microphone placement experiments

The microphone of the elephant recording collar (i) needs to be sensitive, but (ii) needs to remain protected against physical abuse. The best solution appears to be to endeavour to mount the microphone in the electronics enclosure (which is strong and rigid), rather than distant from the enclosure, where it would not be protected as well. It was found that infrasonic sound does not propagate well through the 5 mm thick polycarbonate cover sheet. Several experiments were conducted to determine the best way in which to mount the microphone. These are described in this section.

The microphone was mounted in several different ways (described below) inside the enclosure and sound transmission properties were measured. To this end, several cover sheets were made and exchanged for each experiment.

4.2.1.1 Microphone mounting method 1

Seven holes with a diameter of 2.5 mm each were drilled in close proximity to each other (Figure 4.1). These holes do not go completely through the polycarbonate sheet, but end blindly 0.3 mm before the outer edge. This was done to maintain a smooth watertight surface on the outside of the cover.

4.2.1.2 Microphone mounting method 2

The microphone is mounted on the inside of the polycarbonate sheet directly beneath the holes. It is fixed to the cover panel with watertight Pratley putty. The remaining thick parts of the sheet between the holes provide strength in the area above the microphone.

Figure 4.1: Clear polycarbonate cover sheet shown from the outside, showing holes ending blindly.

Three holes with a diameter of 2.5 mm each were drilled through the polycarbonate sheet. A thin layer of softer plastic material (PVC, 1 mm thick) was glued (using Pratley clear glue) on top of the holes to provide a watertight smooth surface. The microphone was mounted directly beneath the holes.

4.2.1.3 Microphone mounting method 3

A single large hole with a diameter of 24 mm (so that the microphone can fit into it) was drilled into the polycarbonate sheet. Once again this hole does not go through, but ends blindly, leaving 0.3 mm of material on the outside of the sheet. The microphone is mounted inside this hole. See Figure 4.2.

Figure 4.2: This figure shows the microphone-diameter blind hole. The smaller holes at the bottom of the larger hole were drilled for method 5, but were not present in method 3.

4.2.1.4 Microphone mounting method 4

At first, sixteen holes with a diameter of 2.5 mm each were drilled through the polycarbonate sheet. These holes were filled with polyurethane foam to ensure watertightness. The microphone was placed directly beneath the foam filled holes. More holes were added in a later experiment (seen in Figure 4.3).

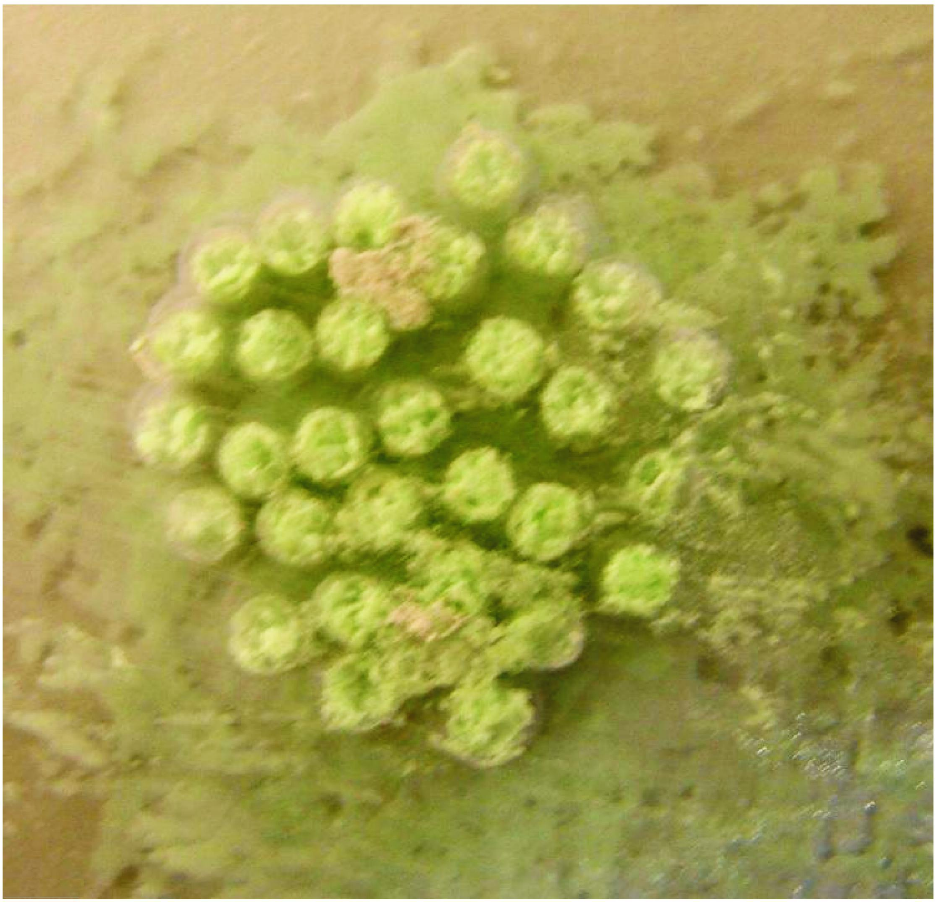

Figure 4.3: In this attempt, the holes were filled with polyurethane foam for a watertight seal.

4.2.1.5 Microphone mounting method 5

A microphone-sized hole as described in method 3 was drilled, again leaving a thin (0.3 mm) layer of material to cover the microphone. The microphone was wrapped in a thin sheet of latex to ensure watertightness and mounted into the hole. Twelve small holes with a diameter of approximately 0.4 mm each were drilled, aligned with the holes of the microphone casing (Figure 4.4).

Figure 4.4: The small holes drilled in the polycarbonate (left) are aligned with the microphone casing holes (right).

In this mounting method, the metallic enclosure of the microphone itself provides mechanical strength to the thin plastic area. The small diameter of the holes ensures that the surface of the polycarbonate sheet remains relatively smooth.

4.2.1.6 Procedure

Each of the mounting methods described above were experimentally tested as illustrated in the setup in Figure 4.5. A laptop running a custom Matlab program connected to a loudspeaker was used to generate a frequency sweep from 0 Hz up to 1250 Hz. The microphone was placed perpendicular to the loudspeaker at a distance of 500 mm. The output signal from the microphone was connected to a spectrum analyzer, which recording the transfer function.

4.2.1.7 Results and discussion

Figure 4.6 shows the results of the experiments. The first test was performed without a cover screwed onto the electronics enclosure, so that the transmission path to the

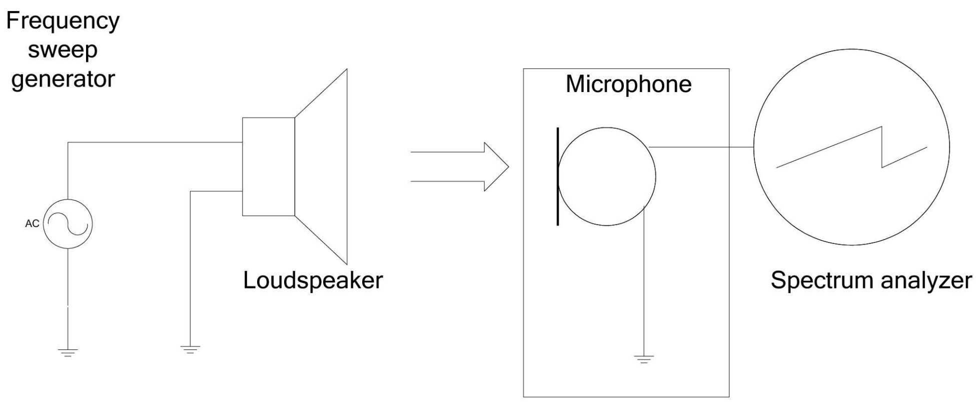

Figure 4.5: Experimental setup for measuring sound transfer functions.

microphone was unobstructed. This result was used as a reference for the various mounting methods tested.

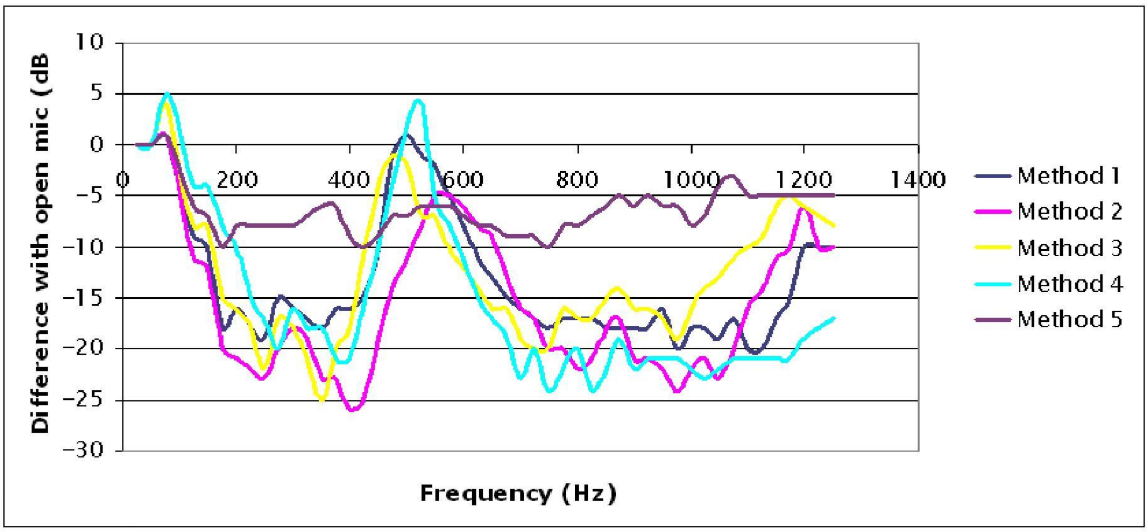

Figure 4.6: Transfer functions obtained from each of the five mounting methods described in the text, along with the reference ("open mic", i.e. unobstructed microphone).

The objective was to find a microphone mounting method that would provide good audio signal transmission, while providing adequate protection for the microphone. It is clear from Figure 4.6 that microphone placement methods 1 to 4 resulted in suppression of spectral components and appeared not to be appropriate choices. Mostly frequency components around 500 Hz conducted well through the thin layer of polycarbonate, but the frequencies lower and higher than 500 Hz were suppressed.

Figure 4.7 shows the difference between the reference microphone placement and each of the methods tested. It can be seen that methods 1-4 all have attenuation of more than 20 dB at some point in the desired band, while method 5 never attenuates more than 10 dB. Among all of the options, method 5 appears to be the most suitable.

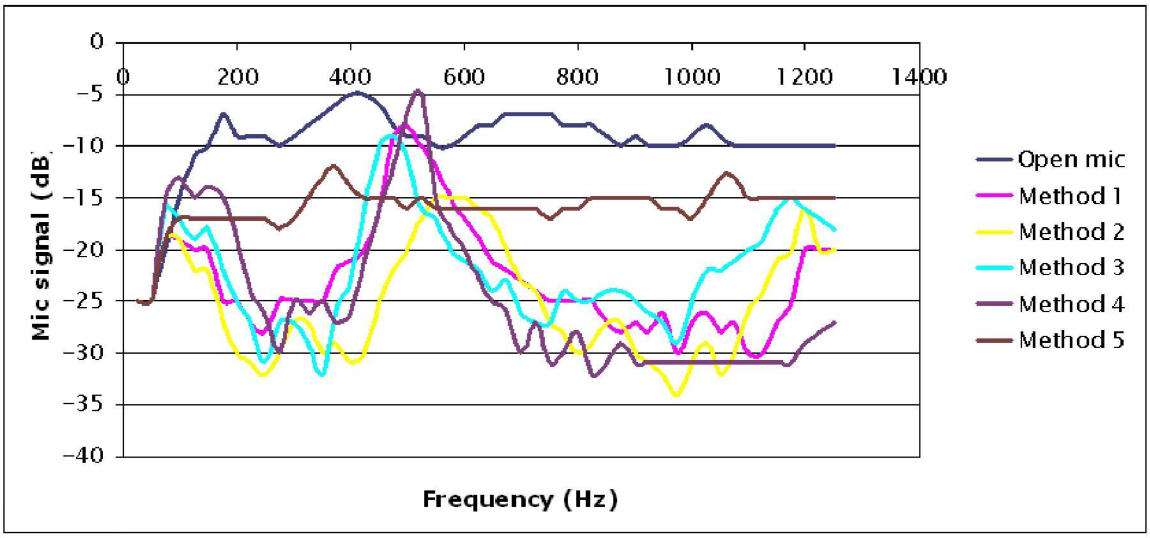

Figure 4.7: Difference (in dB) between the spectrograms of each of the above mentioned methods and that of the open microphone.

The results of the different methods were also compared in Table 4.1 using the average difference between an open microphone and all of the different options as well as the maximum difference.

Table 4.1: Comparison between microphone mounting methods in terms of attenuation

	Method 1	**Method 2**	**Method 3**	**Method 4**	**Method 5**
Average difference between open mic and tested method (dB)	-13.1	-15.32	-12.3	-14.98	-6.34
Max difference (dB)	21	27	29	27	11

Method 5 resulted in a flatter transfer function throughout the part of the frequency spectrum tested and appeared to be the better choice among these methods. Note also (although this result is not shown) that without the latex sheet covering the microphone in method 5, there was almost no attenuation. Thus, with holes in the cover panel carefully aligned with the microphone casing holes, the primary source of attenuation became the latex sheet used to provide a watertight seal.

4.2.2 Temperature stability and power consumption experiment

Simple experiments were conducted to determine the temperature stability of the device where the device (unpackaged and not mounted on the collar) was subjected to a range of different temperatures from -5 °C to 50 °C. These temperatures probably exceed the temperature extremes to which the recording collar would be subjected in the field. The current consumption of the device was measured at each temperature step to determine the influence of temperature variations on power dissipation. Recorded data was verified after each step to determine that correct operation of the device was maintained at every temperature level.

4.2.2.1 Objective of experiments

The objective was to verify that the elephant recording collar would operate correctly at the extreme maximum and minimum temperatures that it might encounter when operating in the field.

4.2.2.2 Equipment

The equipment used in the temperature stability tests is listed below:

1. The elephant recording collar electronics
2. The actual battery pack that would be used in the field
3. A digital multimeter
4. A freezer
5. An oven

6. A personal computer

7. A CompactFlash Card reader

4.2.2.3 Method

The experiment was set up as shown in Figure 4.8. The battery pack was connected to the electronics through an ammeter to measure the current drain of the system. A continuous pure tone at 300 Hz was played through a loudspeaker and recorded (via the microphone) by the elephant recording collar electronics. The device then wrote recorded data to the CF card. After completion of recording, the CF card could be read by a PC using a CF card reader.

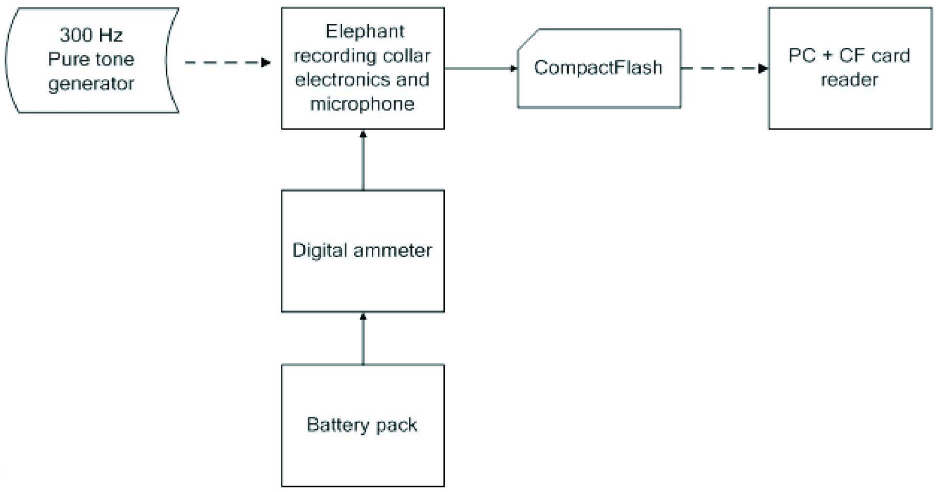

Figure 4.8: Setup of the experiment.

4.2.2.4 Procedure

1. The setup as shown in Figure 4.8 was put into the freezer (set at -10 °C) for 30 minutes.

2. At the end of the 30 minutes the current drain was measured.

3. The files on the CF card were transferred to the PC.

4. It was verified that correct operation had occurred throughout the experiment by checking that the files contained the correct sound data throughout the 30-minute recording. "Correct operation" is defined as no errors having occurred in the recorded 30-minute data file.

5. The same procedure was repeated in an oven that had been pre-heated to 50 °C.

6. These results were then compared to previous measurements at room temperature.

7. The experiment was repeated only once at each temperature.

4.2.2.5 Results

Table 4.2 shows the results obtained from the experiments.

Table 4.2: Operation at different temperatures

Temperature (°C)	**Current consumption**	**Correct operation**
-10	19 mA	Yes
25 (room temperature)	19 mA	Yes
50	21 mA	Yes

4.2.2.6 Discussion

The effects of temperature on the accuracy of the measuring apparatus appears to be negligible. The results of the experiments suggest that the electronics would be able to operate correctly between the two temperature ranges (-10 to 50 °C) as examined in this experiment. The current drain of the device was approximately 20 mA, which was within the range allowed by the specifications.

4.2.3 Field tests

The prototype elephant recording collar was tested on a young elephant bull for one week in August 2007. The Elephant Sanctuary near the Hartebeespoort dam allowed us to test the collar on one of their elephants. These elephants are tame enough to allow handlers to put the collar on and take it off without the need to tranquillize the elephant. This was ideal for the first brief field tests to sort out any obvious problems that might arise. Figure 4.9 shows the handlers commanding the elephant to lie down so that they could attach the collar.

Figure 4.9: The elephant lying down so that the handlers may mount the recording collar.

4.2.3.1 Method

A 4 GB micro hard drive with a CompactFlash interface (from now on called the microdrive) as well as a 256MB CompactFlash card were used as data storage media. The cards were inserted into two of the available CompactFlash sockets housed in the collar. The device was activated approximately 15 minutes before the collar was mounted on the elephant. Figure 4.10 shows the collar fitted onto the elephant.

Figure 4.10: The elephant after the collar have been successfully mounted.

4.2.3.2 Results

After one week the collar was retrieved. The electronics were still in working condition and it appeared that no dust or water entered the sealed part of the collar. This was probably not an adequate test for watertightness, since elephants apparently do not take mud baths in winter. The polycarbonate cover of the collar was still firmly attached and appeared to be an appropriate choice of material for this application.

The data was retrieved from the cards by putting the card into a card reader that was connected to a windows operated computer. The computer detected the card as a removable drive with a FAT32 file system as is shown in Figure 4.11. Sound files are in the wave file format. Each sound file has a duration of one hour and starts where the previous file ended, resulting in the availability of continuous sound data. Each sound file has a corresponding text file that contains the time and date, temperature and GPS coordinates of the collar as it was at the time of file creation.

Inspection of the data on the memory card revealed that the device stopped recording 30 hours after activation and returned to the reset mode. The 4GB microdrive was

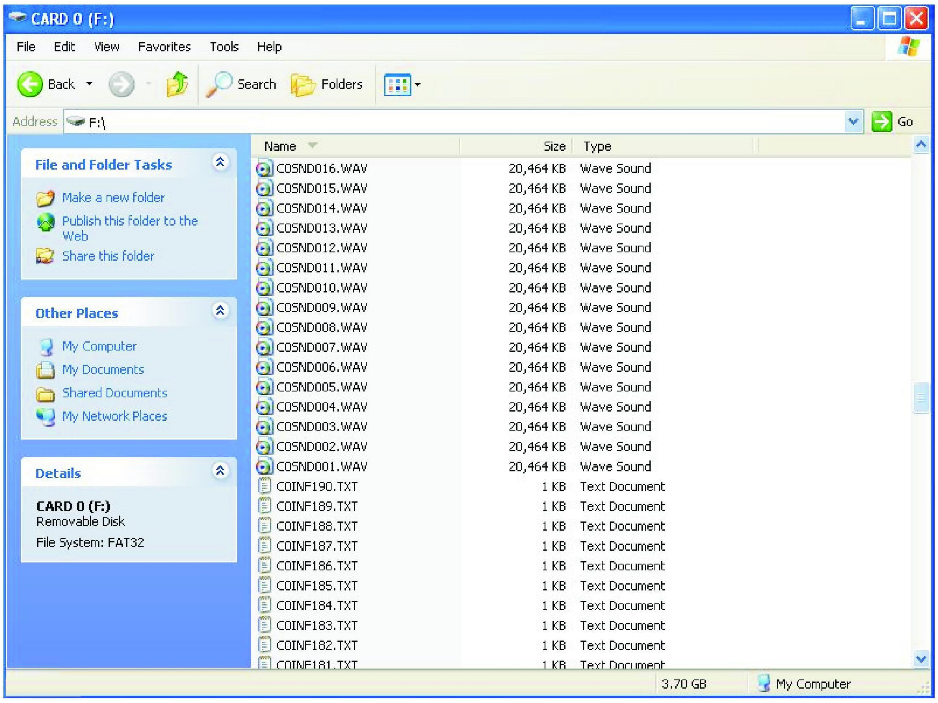

Figure 4.11: The root folder of a CF card containing the recorded sound and information files.

the only card to contain data as the recording process stopped before the 256 MB card could be reached. At first it was thought that power from the battery could have been lost for an instant (as a result of a bump) causing the device to restart and return to reset mode. Further investigation revealed that the specific microdrive that was used consumed more energy than expected. The fact that all previous testing of the collar was done with the same battery pack and microdrive resulted in the battery pack running flat before the calculation recording time had expired. This meant that the device was still entirely operational and only needed new batteries.

This first field test highlighted some problems, but also, despite the disappointingly short recording time, showed that good quality recordings of elephant calls could be made. Some examples are shown here. The rumbles shown in the spectrogram of Figure 4.12 were recorded on file C0SND011.WAV during the field test. Figure 4.13 also shows a spectrogram containing elephant rumbles from file C0SND024.WAV.

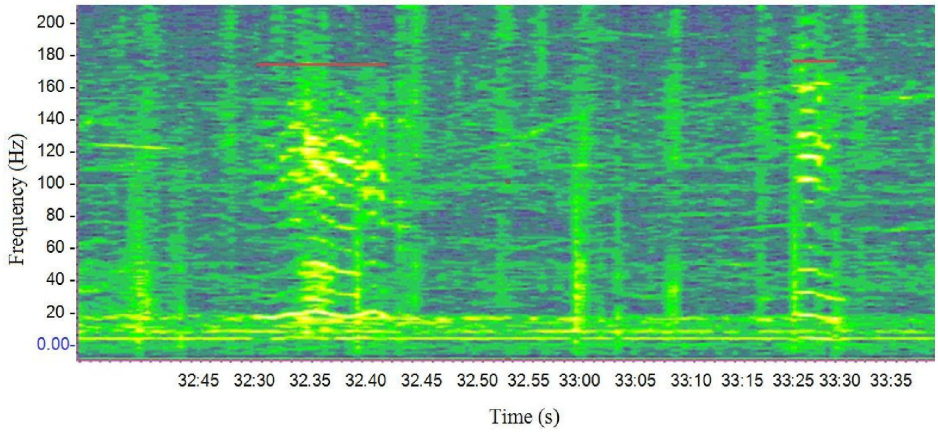

Figure 4.12: A spectrogram from a part of the file C0SND011.WAV recorded during a field test containing elephant rumbles.

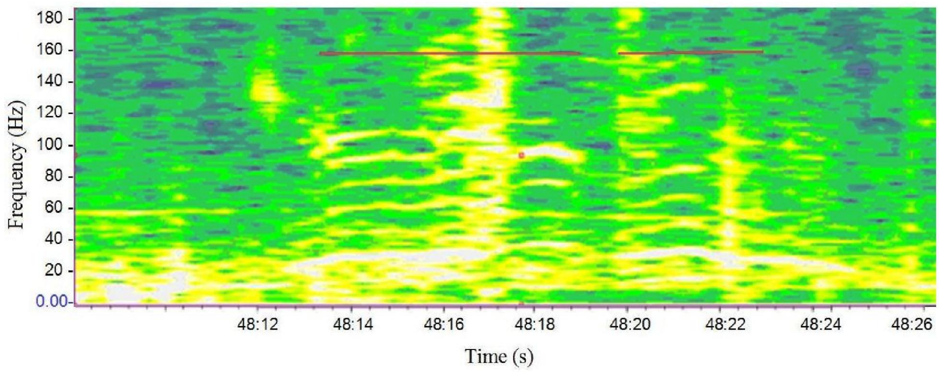

Figure 4.13: A spectrogram from a part of the file C0SND024.WAV recorded during a field test containing elephant rumbles.

4.2.3.3 Apparent errors in the recorded data files

It appeared that sound was not properly recorded in the first few files (where the battery pack was still operational), while recordings in later files were of good quality. So far, we have not found a good explanation for this. Because the microphone was mounted in an airtight container, one possible explanation may be the difference in air pressure between the location where the collar was stored and where it was activated, resulting in the membrane of the microphone being "sucked in" for a while until the pressure could equalize.

A periodic clicking sound occurred in some of the recorded sound files about every 20 seconds as is the case shown in the spectrogram of Figure 4.14. This was not noted in previous tests in the lab. The time interval of the clicking sounds corresponded exactly with the buffer size used in this prototype device. There did not appear to be any previously unnoticed buffer overflow and when using a new power supply the clicking sounds ceased. It appeared that, because the energy levels of the battery were low, the high spike current drawn by the microdrive during the writing cycle (occurring each time that a buffer needed to be written to permanent storage) caused the supply voltage provided by the battery to dip, in turn causing the voltage reference point of the ADC to fluctuate briefly and thus causing the single spike in the sound. Thus, this error was also caused by battery failure.

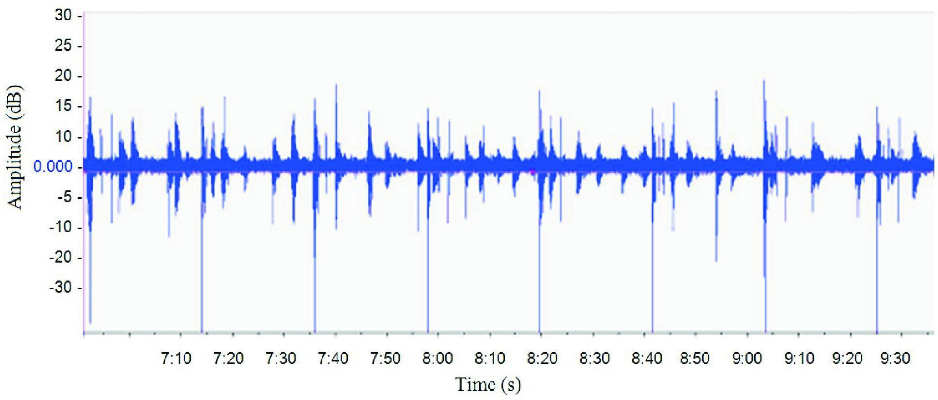

Figure 4.14: The recorded sound contained periodic clicking sounds.

One of the sound files contained a periodic sound that initially appeared to be another error (Figure 4.15).

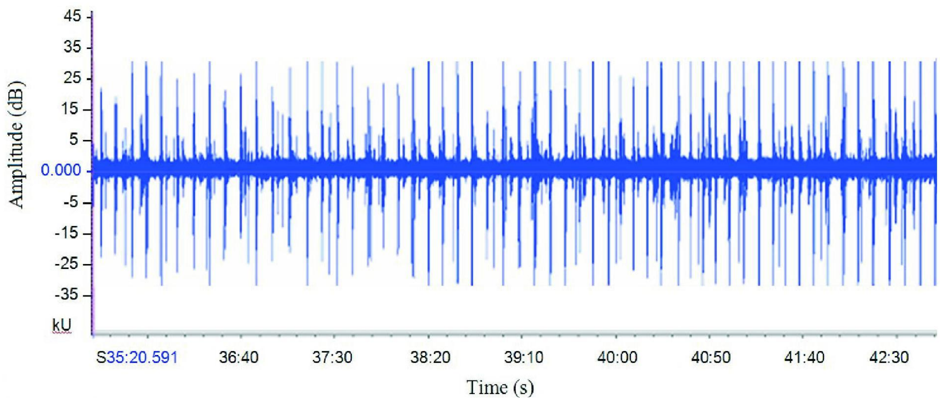

Figure 4.15: (Almost) periodically occurring sounds were recorded.

However, closer investigation suggested that the periodic sounds occurring in wavefile C0SND011.WAV as shown in Figure 4.15 was not a design defect. We could not repeat this effect in the lab, and the almost (but not entirely) periodic sound suggested that this was in fact a sound captured by the microphone. When the recording was played 10 times faster than normal, a sound resembling a person taking quick successive breaths could be heard. Given the time of this recording (middle of the night), it might be possible that a snoring elephant was recorded.

4.2.3.4 Second field test

A second field test was done with a fresh battery pack. The test was done at the same location and with the same elephant that was used for the first test. This time around a 4 GB CompactFlash card was used instead of the microdrive. The test once again had a duration of one week.

The following results were obtained:

1. The test was completed successfully. Both the memory cards were filled to capacity with recorded data.
2. The electronics were still in working condition. Plenty of rain had fallen during the test, but it appeared that no dust or water entered the sealed part of the collar.

3. It was noticed that, as expected, the defect that was observed in the first field test was not present in these recordings.

4. Some of the recordings seemed to be recorded at a softer sound level than the rest of the recordings.

4.2.3.5 Conclusion

The results of the first field test confirmed the ability of the elephant collar to record elephant vocalizations in realistic circumstances. The mechanical design of the collar proved able to withstand the physical conditions that it had been subjected to while fitted on the elephant.

The recording process ended prematurely because of earlier than expected end of battery life. This was the result of the microdrive having had a higher power consumption than expected. A defect was observed in some of the recordings. A phase shift occurred in the sound every time that the information in the RAM buffer was transferred to the microdrive. This happened because the excessive power consumption of the microdrive disturbed the reference voltage of the ADC at low battery capacity.

A second field test was done with a fresh battery pack and a Compactflash card was used instead of the microdrive. This solved the problems that had been encountered in the first field test, since the recording process was completed successfully and without the presence of the clicking sound in the recordings.

The fact that some of the recordings were at a softer sound level was probably due to mud blocking the microphone for the period of time that it needed to dry and peel off.

4.3 SIGNAL PROCESSING RESULTS

Finding a measure of the success of the elephant rumble detection algorithm is a difficult task because it can only be referenced against the subjective findings of a person who did a visual analysis of the spectrograms of the recordings. An experiment that aims to quantify the abilities of the algorithm in an objective way will be presented. This experiment is discussed in Section 4.3.1.

The next step was to test the algorithm on actual elephant vocalization data to see if the results correspond to the findings of the visual identification of the same data (4.3.2). A number of good quality elephant rumbles were isolated from the available elephant recordings by visual inspection of spectrograms. The algorithm was tested on clearly defined rumbles as well as rumbles that occurred together with common background noises. The results were used to do an estimation of the accuracy with which the algorithm could detect elephant rumbles (4.3.3).

Finally, the circumstances under which the algorithm will fail (4.3.4) will be investigated and a method by which the algorithm can detect overlapping rumbles will be discussed (4.3.5).

4.3.1 Quantifying the pitch detection and tracking ability of harmonic sounds in noisy conditions

It was not an easy task to measure the level of success of the elephant rumble detection algorithm since the results of the algorithm had to be compared to the subjective findings of a person who did a visual analysis of the spectrograms of the elephant recordings. It was therefore decided that an experiment should first be done to quantify the performance of the algorithm based on certain parameters that could be measured and controlled.

The algorithm used the fact that elephant rumbles have a harmonic structure to distinguish between rumbles and other noises present in the recording. The following experiment generated sounds with a varying number of upper harmonics. White noise (like light wind induced noise often found in elephant recordings) was added to the signal. The energy of the white noise was increased repeatedly until the algorithm started to fail (tracking of the pitch was lost). This were repeated with other sounds with upper harmonics.

4.3.1.1 Method

This experiment was carried out using the Matlab programming language. A pure tone signal with amplitude 1 and frequency 25 Hz was generated and white Gaussian noise with zero mean was added to the signal. In this case "white" implies noise with a flat PSD (Power Spectral Density) from 0 Hz to 1500 Hz. The power of the noise signal was increased until some gaps in the pitch track started to occur. The SNR at which this happened was recorded.

The same procedure was then repeated, but with a first overtone added to the signal. The signal now consisted of the 25 Hz signal and a 50 Hz signal also with an amplitude of 1. Once again the noise power was increased until some gaps in the pitch track started to occur. The SNR at which this happened was recorded.

The same procedure was repeated, each time with one more harmonic component added to the signal.

4.3.1.2 Results

Table 4.3 shows the results obtained from the experiments.

Table 4.3: Results obtained by the SNR experiment

SNR (dB)	25	6	-5	-8	-8.4	-8.7
Number of overtones	0	1	2	3	4	5

Figure 4.16 shows a plot of the SNR values obtained in the procedure explained above. The figure shows that the pitch of a signal containing more than one overtone can be tracked in noisy conditions.

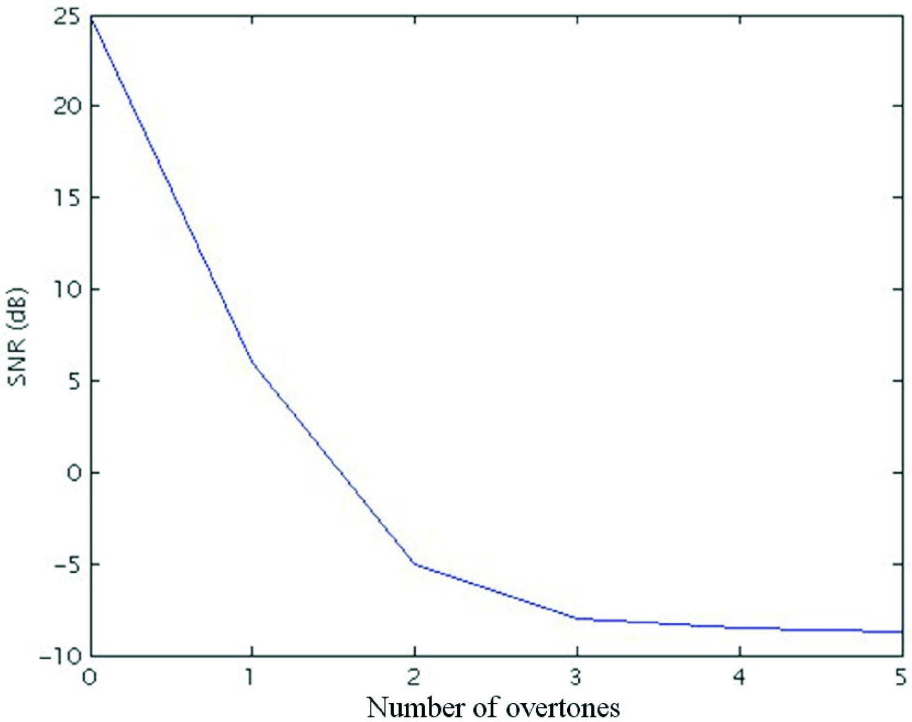

Figure 4.16: A plot showing the minimum SNR values at which a continuous pitch track could be estimated versus the number of harmonic frequencies present in the signal.

4.3.1.3 Discussion

The results indicate that the pitch of a signal which consists of only the fundamental frequency (has no harmonics) will not be tracked if some noise is present in the recording. This is a useful quality, as it can assist in avoiding false alarms generated by narrow band noise that could be produced by motor vehicle engines (as an example see the first 30 seconds of the spectrogram shown in Figure 4.17).

The results also show that the algorithm will work best when the elephant rumbles in a recording have three or more clear overtones. The algorithm can detect the pitch of a sound with harmonic structures in an SNR as low as -8 dB which means that the algorithm has good rejection of broadband noise like that caused by wind. This result also implicates that the algorithm's weakness lies in its inability to reject noises with a harmonic structure.

4.3.2 Tests on recording segments containing single elephant rumbles

As stated in Chapter 3, a number of elephant recordings made in the Kruger National Park were made available for this study. As a first step in testing the rumble detection algorithm on actual elephant recordings, some clearly definable elephant vocalizations were located within the provided recordings. The segments of the recording containing the vocalization were saved as a separate sound file. A spectrogram of one such sound segment (R001.WAV) is shown in Figure 4.17. The part of the spectrogram where the rumble occurred was marked by a red line. It can be seen that the rumble shown on the spectrogram started at approximately two and a half seconds into the recording and ended at approximately 5 seconds.

The sound segments were then used as input to the elephant rumble detection algorithm. It was found that algorithm accurately detected an elephant rumble in a good quality sound segment (if the rumble is clearly definable and little background noise is present). The output produced by the algorithm is shown in Figure 4.18.

Certain rumble parameters were automatically determined by the algorithm and are given in Table 4.4. The time in the sound segment where the presence of an elephant

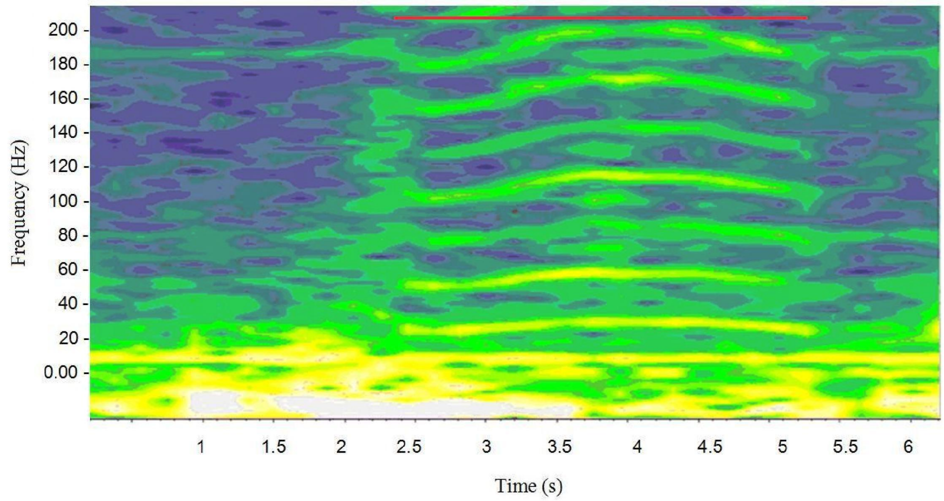

Figure 4.17: Spectrogram of a sound segment containing a clearly definable infrasonic elephant rumble.

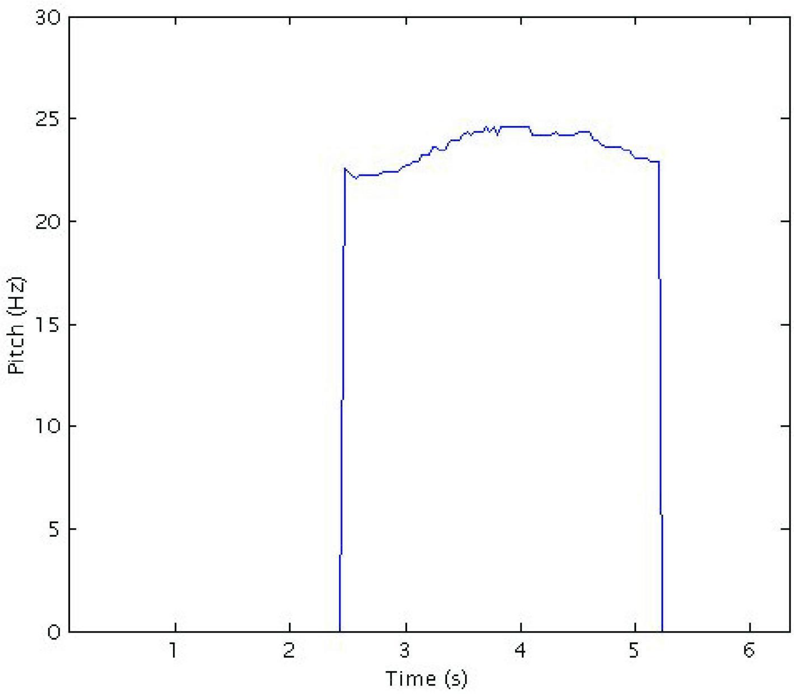

Figure 4.18: Output from the rumble detection algorithm after processing the sound segment of Figure 4.18.

rumble was first detected is denoted as *time begin* and the value is given in seconds. *Time end* indicates the time in the recording where the vocalization stopped. The duration of the call was calculated by subtracting the end time from the starting time of the call. *Pitch high* denotes the highest pitch present in the rumble and is given in Hz. *Pitch low* indicates the lowest pitch that was detected in the rumble. *Pitch average* gives the average pitch of the rumble and is calculated by adding the pitch value of all the time windows within the rumble together and dividing the answer by the number of time windows present in the rumble.

The results obtained by the automatic detection algorithm correspond well with the visual detection of the rumble from the sound segments. The automatic calculation of the pitch values is something that cannot be done by visual inspection of a spectrogram and can be a valuable asset since the pitch of a rumble can communicate information about the emotional state of an elephant (Soltis *et al.*, 2005b).

Table 4.4: Automatically detected parameters of the elephant rumble shown in Figure 4.17

Time (s)		**Call Duration (s)**	**Pitch (Hz)**		
Begin	**End**		**High**	**Low**	**Average**
2.4	5.2	2.8	24.8	22.3	23.6

4.3.3 Tests on recordings with background noise

It was established that the rumble detection algorithm could accurately detect the location of a clearly definable rumble within a sound segment. For the algorithm to be used in a meaningful way, it would have to be able to determine rumble locations within recordings containing common background noises. A few such examples will now be given.

These examples were chosen from approximately 40 hours of available data. Notes (compiled by elephant call experts) were provided with the data. These documented the conditions under which the data were recorded, as well as the distance from the elephant and a rough estimate of when vocalizations were uttered during the recording. After careful analysis of some of the data, it became clear that, in general, no calls could be detected in recordings made from too far away from the elephants. Further, on extremely windy days the data were corrupted. All recordings were made on two channels.The recordings on the second channel could not be used because the sound level was too low. Where recordings were made under the unwanted conditions mentioned above, these were disregarded, resulting in roughly a tenth of the data being analysed.

The data used in the following examples were recorded in typical conditions, defined here to be from a location close to the elephants, relatively far away from roads and not in strong wind. From preliminary work, it was known that the algorithm should perform well if the harmonic structure of the rumbles were well preserved and if the recording was not contaminated with unwanted noises that had strong harmonic structure. Circumstances under which the algorithm failed are discussed in Section 4.3.4.2.

The spectrogram of the sound (S0017.WAV) used to demonstrate the manual identification of elephant rumbles in Chapter 3 is shown in Figure 4.19. The elephant rumbles that were manually identified are indicated with red lines. The horizontal green lines that can be seen in the spectrogram are unwanted narrow band noise produced by a car engine. The vertical green lines are broadband noise typically caused by wind, breaking of brances or disturbance of the microphone itself.

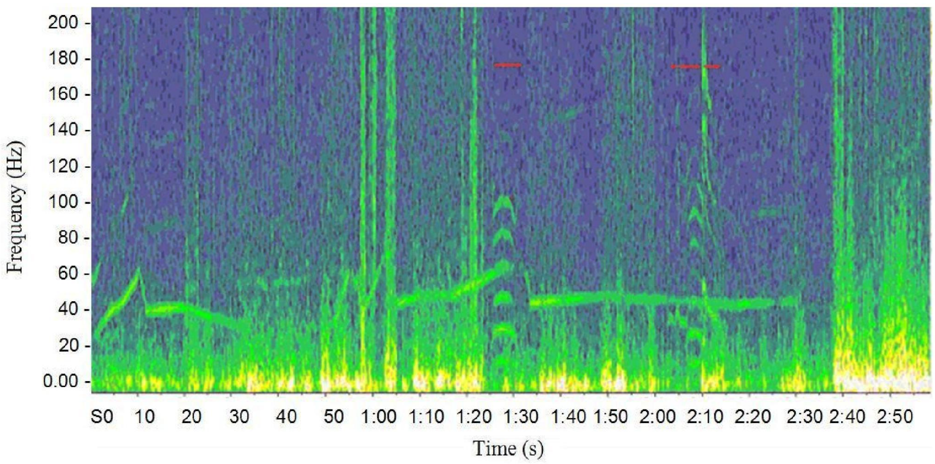

Figure 4.19: A spectrogram of a segment of sound containing narrow band and broadband noise as well as elephant rumbles (which are indicated by red lines).

Figure 4.20 shows the results obtained after processing the recording with the algorithm. It can be seen that unwanted noises are rejected well, while the harmonic structures of the elephant calls are all identified and their pitch extracted.

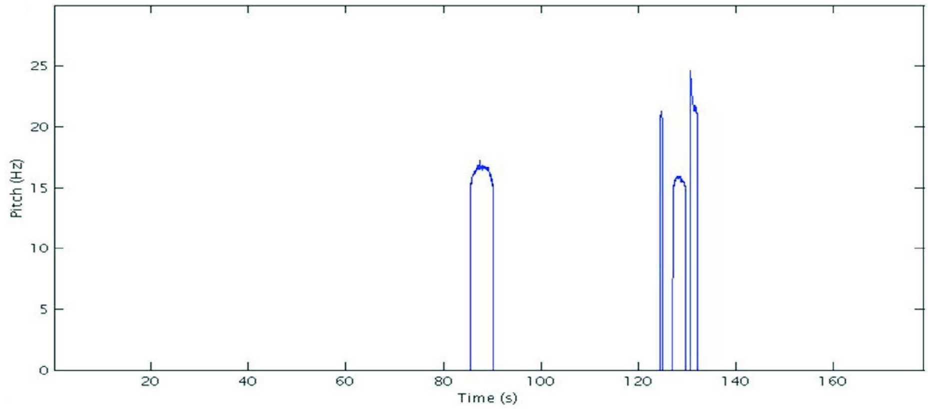

Figure 4.20: Final output of the algorithm after the file in Figure 4.19 was analysed.

The spectrogram shown in Figure 4.21 shows a more difficult example (S0023.WAV). Engine noises can once again be seen in the middle of the recording. Note that certain parts of the engine noises contain upper harmonics. It seems that this recording was made from a considerable distance from the elephants, since the rumbles that were recorded are quite weak. The rumbles that were manually identified are indicated by a red line.

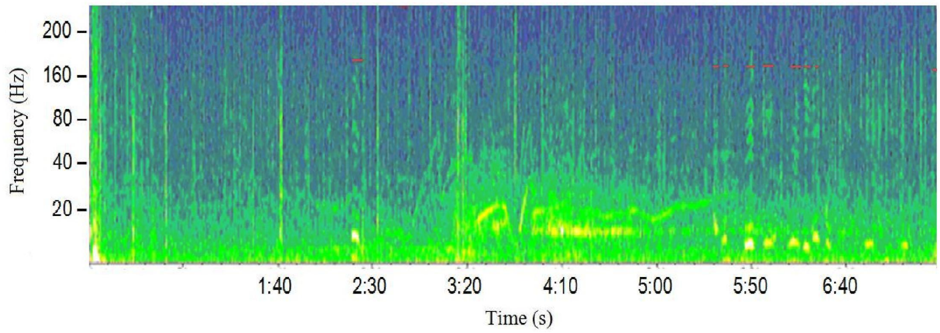

Figure 4.21: A spectrogram of a segment of sound containing engine noise as well as elephant rumbles.

Figure 4.22 shows the results obtained by the algorithm after the processing of the sound in Figure 4.21. The noise was successfully rejected. The harmonic engine sounds were rejected because the rate of change of the pitch was high enough that a pitch track could not be formed. The fifth manually detected rumble was not detected by the algorithm due to the loss of a clear harmonic structure.

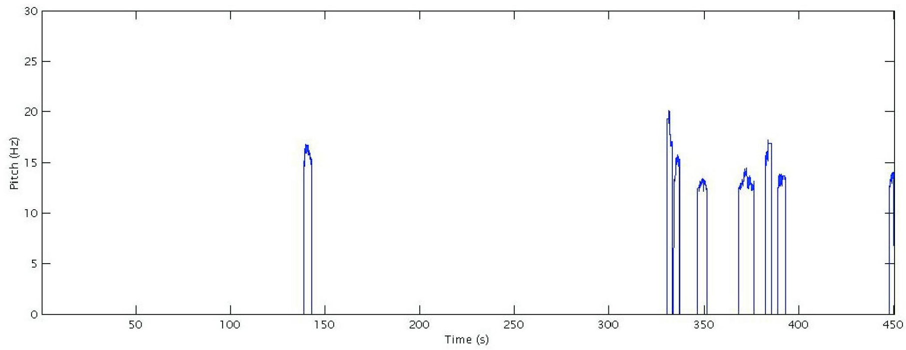

Figure 4.22: Final output of the algorithm after the file in Figure 4.21 was analysed.

Figure 4.23 shows the spectrogram of a recording (S0039.WAV) containing engine noises and very loud screaming type elephant sounds. When listening to the recording speeded up ten times faster than normal, it could be heard that rumbling sounds were also made together with the screaming sounds. These rumbles were very faint and were not easy to identify visually. The rumbling sounds that were manually identified are indicated by red lines.

Figure 4.24 gives the results produced by the algorithm. Noises were rejected successfully, but one of the calls that was identified manually was not detected due to a lack of a sufficient number of higher harmonics.

Table 4.5 shows a list of the calls (presented in this section) that have been identified manually along with the results of the automatic rumble detection algorithm. It indicates whether the algorithm was able to identify the rumble and presents the parameters obtained by the algorithm. A rumble was labelled as correctly identified when the start and end times were within 0.3 seconds of the call identified manually from the spectrogram. The manually identified rumble locations have indistinct starting points and end points and are for this reason not given in the table.

The results that were obtained by the elephant rumble detection algorithm are given in Table 4.5. The recordings that were analysed up to this point were chosen to be representative of the average quality of useful elephant recordings as was discussed in

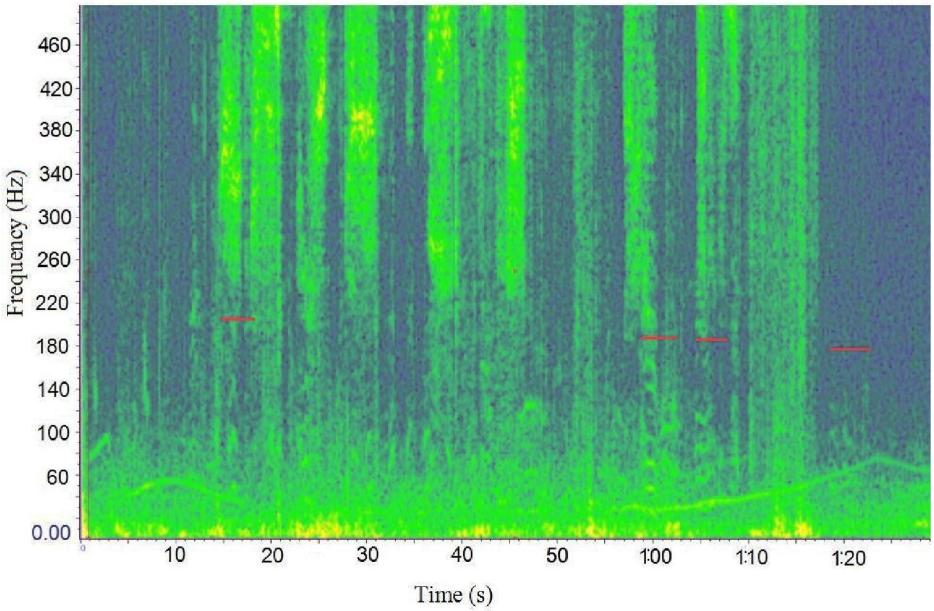

Figure 4.23: A spectrogram of a segment of sound containing scream-like elephant sounds as well as elephant rumbles.

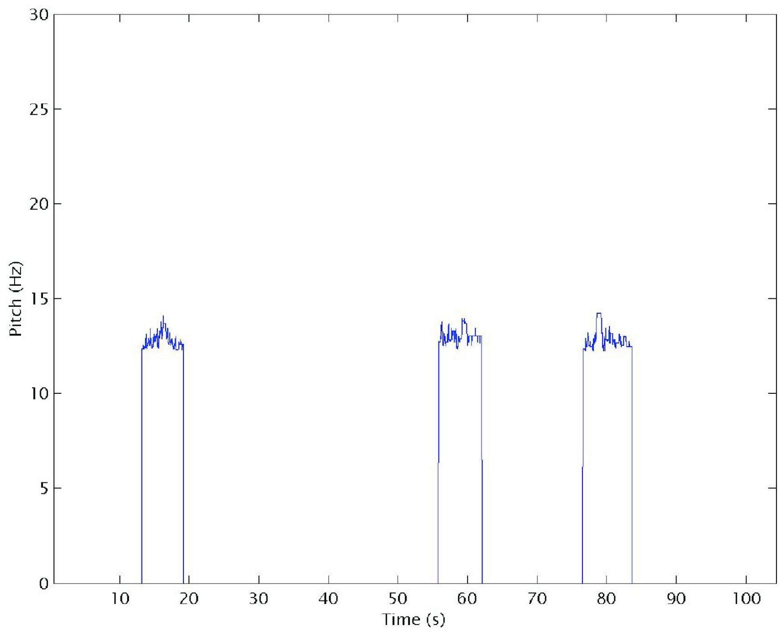

Figure 4.24: Final output of the algorithm after the recording of Figure 4.23 was analysed.

the opening paragraphs of this section. All but two of the calls that were identified by hand were also correctly identified by the algorithm. Four clearly defined elephant rumbles including the one depicted in Figure 4.17 were isolated and were correctly identified by the algorithm. Combining these with the results given in Table 4.5 it can be seen that 19 out of the 21 calls that were manually detected were also detected by the algorithm. This means that the number of calls that were correctly identified by the elephant rumble detection algorithm corresponded with 90.47% of those that were identified manually.

4.3.4 Circumstances under which the algorithm fails

Within some limitations, the algorithm discussed here proves to be a reliable way of detecting infrasonic elephant rumbles in noisy conditions. It also rejected unwanted low frequency sounds that do not have higher harmonics, such as distant motor vehicles. Sounds that contain higher harmonics, but had a rapid rate of change in pitch were also rejected.

Except for the examples shown in the previous subsection, where faint calls were not detected because of the masking of the harmonic structure by noise, there are certain other instances where the algorithm might fail.

4.3.4.1 Overlapping calls

Figure 4.25 shows the spectrogram of a sound (R005.WAV) where a faint rumble appears in the background, while a short louder rumble occurred simultaneously.

Figure 4.26 shows that only the louder call was identified by the algorithm. It will be shown in the next subsection that overlapping calls can be detected by applying a slight modification to the algorithm.

Table 4.5: A summary of the results obtained from the elephant rumble detection algorithm.

Recording: Figure 4.19		**Data produced by the algorithm (S0017.WAV)**					
Call No.	**Algorithm identified?**	**Time (s)**		**Call Duration (s)**	**Pitch (Hz)**		
		Begin	**End**		**High**	**Low**	**Average**
1	Yes	85.6	90.1	4.5	17.6	15.1	16.8
2	Yes	124.4	125	0.6	22.4	21.9	22.1
3	Yes	127.1	129.8	2.7	16.1	14.7	15.3
4	Yes	130.7	132.1	1.4	24.6	21.8	22.3
Recording: Figure 4.21		**Data produced by the algorithm (S0023.WAV)**					
Call No.	**Algorithm identified?**	**Time (s)**		**Call Duration (s)**	**Pitch (Hz)**		
		Begin	**End**		**High**	**Low**	**Average**
1	Yes	139	143.1	4.1	17.0	14.9	15.4
2	Yes	330.8	333.4	2.6	20.0	16.8	17.1
3	Yes	334.4	337.1	2.7	14.9	12.7	13.8
4	Yes	346.3	351.4	5.1	13.3	12.5	12.9
5	No	N/A	N/A	N/A	N/A	N/A	N/A
6	Yes	368.4	376.5	8.1	14.8	12.6	13.0
7	Yes	382.5	385.7	3.2	17.4	14.3	16.2
8	Yes	389	392.9	3.9	14.2	12.3	13.7
9	Yes	447.6	450.2	2.6	14.1	12.7	13.5
Recording: Figure 4.23		**Data produced by the algorithm (S0039.WAV)**					
Call No.	**Algorithm identified?**	**Time (s)**		**Call Duration (s)**	**Pitch (Hz)**		
		Begin	**End**		**High**	**Low**	**Average**
1	Yes	12.6	19.7	7.1	14.3	12.4	12.9
2	Yes	56.8	62.3	5.5	14.1	12.6	13.1
3	No	N/A	N/A	N/A	N/A	N/A	N/A
4	Yes	77.2	83.5	6.3	14.9	12.3	13.2

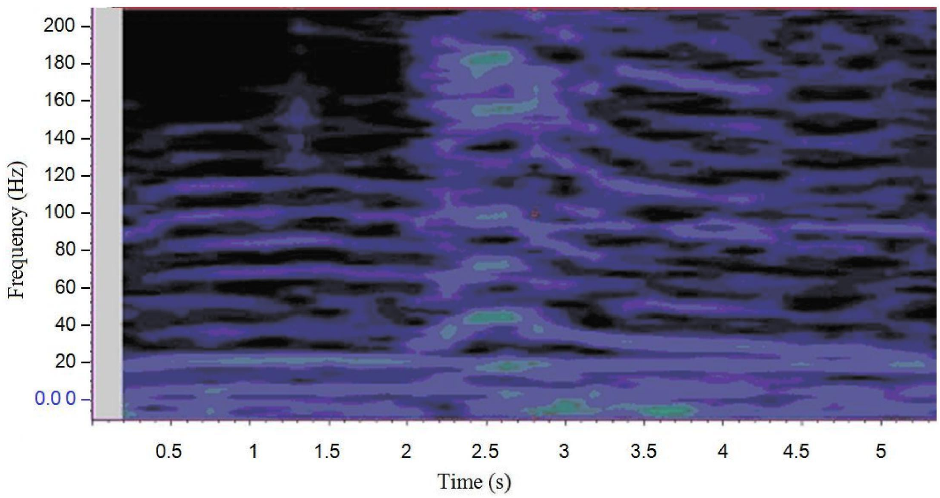

Figure 4.25: Spectrogram showing a faint call in the background with a louder call starting in the middle of the spectrogram.

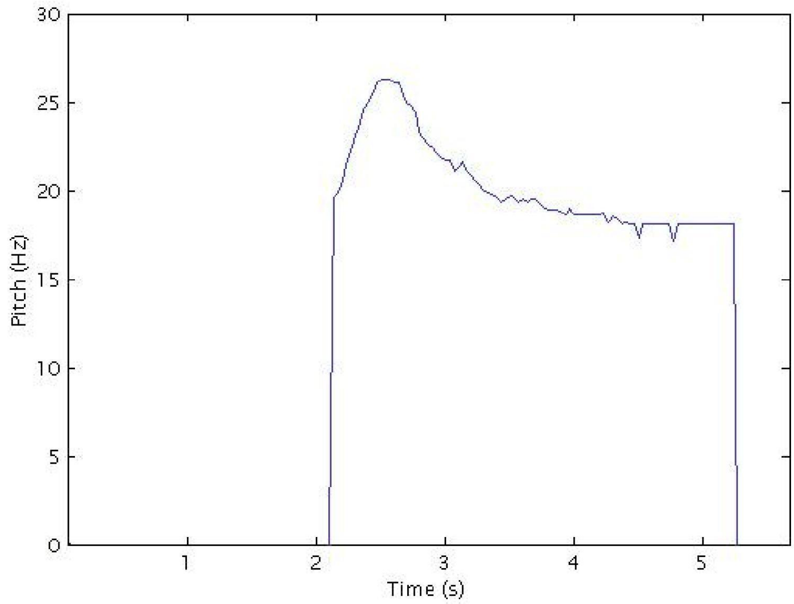

Figure 4.26: The output of the algorithm after the sound in Figure 4.25 was processed.

4.3.4.2 Unwanted harmonic noises

Figure 4.27 shows the spectrogram of a recording (003.1001.6000.WAV) with very energetic unwanted frequency components at 50 Hz and 75 Hz and the engine noises of a motor vehicle. Two elephant rumbles occur at 3:20 and 3:30, but it can be seen from the spectrogram that the energy of the unwanted components are much stronger than those of the rumbles itself. The algorithm falsely detected the engine noises as rumbles (because of the presence of the stronger harmonics) and at some point detected the pitch of the unwanted frequency components for a short while. This can be seen in Figure 4.28 which shows the output of the algorithm when the sound shown in Figure 4.27 was the input. The last of the two elephant rumbles was less faint and was correctly identified.

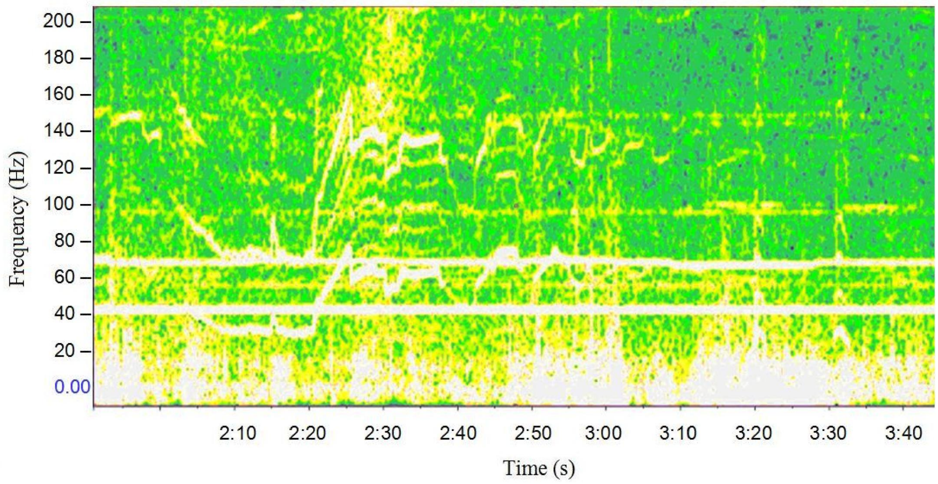

Figure 4.27: Spectrogram of a recording with loud unwanted frequency components.

The quality of the recording of Figure 4.27 is very low. The sources of the unwanted frequency components seems to be very close to the microphone while the elephant calls were distant. Although a recording as the one shown in this example is unlikely if an elephant recording collar was used, it is a good example to use for pointing out the weaknesses of the elephant rumble detection algorithm.

An unwanted sound with strong harmonics in the infrasonic bandwidth causes false rumble detections, while distant elephant rumbles which lost their upper harmonics might not be detected.

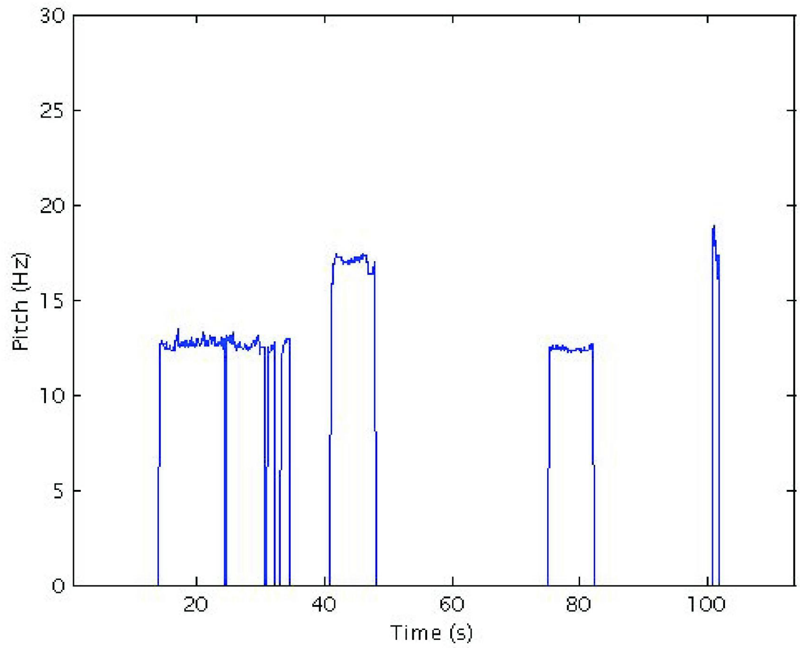

Figure 4.28: The algorithm's output after the sound shown in Figure 4.27 was processed.

4.3.5 Detection of overlapping rumbles

In an attempt to mitigate the problem of detecting overlapping calls, the algorithm was adapted slightly to allow tracking of two pitch tracks simultaneously. In the single-track algorithm, the point in the summed correlogram where the maximum value occurred was used to calculate each data point of the pitch information array. In the adapted version, the two peaks in each summed correlation with the highest values were used to generate two pitch information arrays. The information contained in these two arrays was then processed in the same way as in the case of the single pitch information array with the exception that two pitch tracks were processed. In essence, the procedure described in Section 3.5.2.6 is performed twice.

Figure 4.29 shows an example of the graphical output generated by the dual pitch track algorithm. The sound shown in the spectrogram of Figure 4.25 was used. In this figure multiple calls that overlap were identified.

The tracking of multiple pitches solved one problem, but posed another. The second pitch information array contained weaker pitch information that would normally be

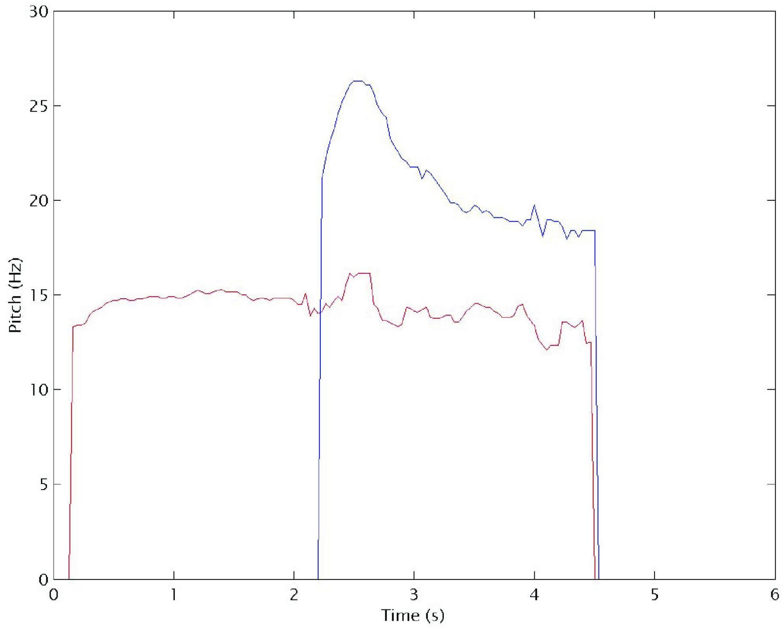

Figure 4.29: Dual pitch track algorithm output for the sound segment shown in Figure 4.25.

ignored, so that unwanted sounds that would not be detected by the single-track algorithm could be detected more easily when using the dual-track algorithm. The dual-track algorithm should generate a far greater number of false alarms.

4.3.6 Conclusion

In summary, the proposed algorithm will work reliably in some conditions, but not in all. Weak calls might not be detected if their harmonic structure is lost, and false alarms will increase when the dual-track algorithm option is selected. Either the single or the double pitch track algorithm can be selected depending on the need of the user.

The rumble detection algorithm identified the positions of rumbles in recordings with common background noises with an accuracy of 90.47%. It should, however, be remembered that both the manual detection of elephant rumbles and the definition of average data are subjective procedures which means that a good idea of the value of this algorithm could only be established by feedback from a number of different elephant researchers over a substantial period of time. The ability of the algorithm to determine the pitch of a rumble cannot be done accurately by inspection and could be a useful attribute.

The elephant recording collar that was developed in this study should solve the above mentioned problems. The collar records vocalizations very close to its source and this ensures that the harmonic nature of rumbles will not get lost. The problem of car engine and aeroplane noise should be solved to a large extent since the field of recording is now much smaller, meaning far-off car noises will have much less effect on the recording.

4.4 SUMMARY

Chapter 4 discussed the general experiments that were done with the recording collar. The first experiment was done to attain the best position for placement of the microphone. A number of different configurations were investigated by measuring the attenuation of sound over the required band of frequencies. The results of these experiments illustrated that one method provided much better results then the rest. An experiment was done to measure the power consumption of the collar and to test if the device remained stable at the extremes of the temperature range in which it was specified to operate. The results of the experiment showed that the device remained stable at the temperature at which it was tested and that 20 mA of electric current was drawn by the device.

The procedure of the field tests was explained and the results of the first field test were presented. Both the pleasing and the unsatisfying results were discussed. The recording process in the first field test ended prematurely because the battery ran down faster than had been expected due to a higher than anticipated power consumption of the microdrive that was used. The low battery power destabilized the reference voltage of the ADC causing a clicking sound at the beginning of every data buffer transfer. The second field test was done with new batteries and the microdrive was replaced with a CF card which solved the battery induced problem of the first field test. A number of elephant vocalizations were recorded during both tests. An unresolved problem was the difference in sensitivity between different recordings which was probably caused by blockages of the microphone at certain stages of the experiment.

The performance of the elephant rumble detection algorithm was tested by processing sounds with a different number of upper harmonics in varying sound levels. The results showed that the algorithm could detect sounds with three or more upper harmonics if the SNR was -8 dB or better. The detection algorithm was then tested on a number of elephant recordings and the results were compared to the manual analysis of the same recordings. The results showed that the algorithm detected 90.47% of the manually identified calls. It was also shown that the algorithm failed to detect rumbles that had lost their harmonic structure while noise sources with strong upper harmonics could cause false alarms. It was shown that the algorithm could be applied in a slightly different manner so that overlapping rumbles could be detected.

Chapter 5

DISCUSSION

5.1 CHAPTER OBJECTIVES

This chapter reviews the scientific contributions of the research (5.2). The research questions posed in the introductory chapter are answered (5.3). Finally, the results as expounded in the previous chapter are critically evaluated, and shortcomings in the present study are elucidated in 5.4.

5.2 CONTRIBUTIONS

The use of elephant recording collars that utilise RF signals to transmit sound to a base station has been documented by previous researchers (Leong *et al.*, 2002; Leong *et al.*, 2003; Soltis *et al.*, 2005a; Soltis *et al.*, 2005b; Clemins *et al.*, 2005; Leighty *et al.*, 2008). These RF collars were used for the recording of vocalizations of captive elephants. The work presented in this study provides a potential improvement of this method throught the use of a collar that stores all the sound information onboard (together with temperature information, GPS coordinates as well as time and date information). The new elephant collar was designed and built specifically for obtaining high quality continuous recordings of vocalizations from elephants in the wild for extended periods of time.

The electronics used in the recording collar were developed from first principles, specifically for use in an elephant collar. The latest available technology with respect to flash memory cards and low power consumption was incorporated into the design to extend recording time by improving memory capacity and prolonging battery life. This ensured that the time in between tranquilization of elephants was maximised. In addition, the low power consumption of the device and the use of lithium thionide battery technology ensured that the weight of the battery pack was less than one kilogram for every six months worth of operating time.

The electronics used in animal tracking collars are normally cemented into a collar using dental acrylics. The acrylics provide sufficient mechanical protection when the electronics are completely emerged within it, but are prone to cracking if an outside element like a microphone is introduced. A new way of cementing the electronics by using a hard plastic epoxy was introduced. One end of the collar could not be cemented as this would prevent access to the microphone and memory cards, and was therefore covered with a polycarbonate sheet instead. The combination of the hard plastic epoxy and the polycarbonate sheet provided sufficient protection of the electronics over a limited time period, without the risk of cracking open. A number of experiments were conducted to find the optimal way of mounting the microphone that would ensure protection against physical assault, water, and wind noise whilst retaining adequate sensitivity to pick up the full band of sound frequencies that had to be recorded. The smooth polycarbonate surface ensured that mud will dry and peel off, thereby preventing clogging of the microphone.

The application of speech processing technology to elephant vocalizations is not widely reported. The only known research on this subject (Clemins and Johnson, 2003; Clemins *et al.*, 2005; Clemins and Johnson, 2006) used speech processing techniques to do accurate voice placements of individual elephants. However, a major challenge in the application of the technique was the absence of a large database of elephant vocalizations. It is believed that recordings made by the newly developed elephant collar can be used to provide a more extensive database of elephant vocalizations to aid in such research.

In this study, an algorithm was developed that can successfully detect elephant rumbles from recordings and determine the pitch of these rumbles. This was achieved by modifying the speech detection algorithm proposed by Wu *et al.* (2003). Different window sizes and filter positions were used to compensate for the difference in the frequency ranges between human speech and elephant vocalizations. A new method was devised for the final detection and pitch tracking phase of rumbles (from the raw combined autocorrelation data produced by the main algorithm). Although this method is not optimal, it is computationally inexpensive. The results have demonstrated that the algorithm can detect and track the pitch of a harmonic sound in the infrasonic frequency range with great accuracy. This method is effective even where the SNR is as low as -8 dB, which is comparable to other noise robust pitch detection algorithms used for speech detection (Shimamura and Kobayashi, 2001; Wu *et al.*, 2003). It should, however, be noted that the accuracy of the algorithm decreases sharply if the harmonic structure of a sound is lost. This feature ensures that periodic sounds that have no harmonic structure, like those produces by car engines, are not mistaken for elephant rumbles.

The results of the current research have indicated that the rumble detection algorithm can be changed to enable the tracking of two pitch tracks at the same time. Thus, if two elephants in close approximation to the collar utter vocalizations that overlap, rumble detection and pitch estimation of both the rumbles can be done simultaneously. Using the algorithm in this manner however, means that there is a greater potential for falsely detecting a faint periodic signal like a car engine as an elephant rumble. This results in a trade off situation, where the user must decide what is more important, detecting overlapping calls, or avoiding false alarms.

It has been noted that elephant vocalizations that are analysed by researchers must first be identified manually by experts (Poole *et al.*, 1988; Leong *et al.*, 2002). The automatic elephant rumble detection algorithm could be used as an aid for the identification of rumbles from recordings to speed up the process of assembling large databases containing elephant rumbles. If relatively good quality elephant recordings can be obtained such a database could be created automatically from a large collection of field recordings.

5.3 DISCUSSION OF RESEARCH QUESTIONS

The following conclusions were made with respect to the research questions posed in the first chapter:

1. The problem of developing a recording tool that can withstand operation in harsh conditions and make continuous, high quality recordings for long periods of time was solved by making certain design choices to ensure the following: low power electronic operation; physically robust electronic design; the use of robust, high density onboard memory; small physical size and weight of final electronic product. This was realized respectively by using electronic components featuring the latest micro power technology; using surface mount components where possible and using CompactFlash memory cards that have excellent resistance to physical shocks. A multilayered PCB housing some of the smallest commercially available electronic components ensured that the size and weight of the electronics were kept to a minimum. The mechanical design of the device also played an important role in ensuring the robustness of the device. The electronics were moulded into an elephant collar using a strong plastic epoxy. A removable polycarbonate cover was used to protect the memory cards without restricting access to it.

2. The positioning of the microphone is one of the most important issues in the design of an elephant rumble recording device. The microphone should be mounted in such a way that its sensitivity is not degraded, as this would compromise the quality of the recordings. However, it is also necessary to protect the microphone from excess noise, physical damage and mud clogging, as clogging or physical insult could render the device useless. To prevent the microphone from being clogged with mud, it should be mounted in such a way that the mud would dry and fall off without affecting it. The polycarbonate sheath used as a cover for the collar provides a strong protective barrier and a smooth surface where mud can dry and peel off. A suitable way needed to be found to mount the microphone behind or within the polycarbonate sheath so that sensitivity and bandwidth are kept at desirable levels. A variety of different microphone mounting methods were experimentally evaluated and the most acceptable solution was chosen. This was attained by covering the microphone with a thin waterproofing layer of latex. In addition, the area of the polycarbonate sheath that covered the micro-

phone was only 0.3 mm thick, and perforated over the corresponding openings of the microphone.

3. VAD techniques used on human speech should provide a good basis for automatic elephant rumble detection. Most VAD techniques was developed for use in the telecommunications industry where it is only required to distinguish between silence or speech. This is usually done by determining whether a certain energy threshold has been breached. In real life recordings there are a lot of other high energy components present in a channel besides the voiced speech. A technique described by Wu (reference) was chosen as basis for the elephant detection algorithm in the current study. This technique was selected because it has previously been used specifically for the detection of speech in noisy recordings. The recording is filtered into a number of sub-bands and an autocorrelation function is used to estimate the pitch present in each sound frame of each sub-band. All the channels containing strong evidence of periodicity are summed together to estimate the dominating pitch in each respective time frame of the recording.

4. Elephant rumbles have a harmonic structure (just like voiced human speech) and Wu's algorithm implements a PDA for detecting voice presence in a recording. The main difference between human speech and elephant rumbles concerning the use of the algorithm is the fact that elephant rumbles have a much lower fundamental frequency than human speech. In addition, the voiced components within elephant rumbles continue much longer than in human speech. The cut-off frequencies of the band pass filters used were therefore in the range of 10 up to 250 Hz instead of the 80 - 2.5 kHz used for human speech.

5. The results of the study have demonstrated that the automatic elephant rumble detection algorithm is able detect the presence of sound with harmonic structures with an SNR of -8 dB or better. In low quality recordings with an SNR of less than -9 dB the algorithm will fail. The harmonic nature of elephant rumbles is an important aspect used in the algorithm. If the upper harmonics of a far off rumble is lost there is a good chance that the rumble will not be detected. The presence of bursts of periodic noise which contain upper harmonics increases the chance of false alarms.

5.4 CRITICAL DISCUSSION OF RESULTS

The elephant recording collar was subjected to two field tests. Some good quality recordings of elephant vocalizations were made in both of the field tests. This indicated that the system was able to function in the conditions experienced while fitted on an elephant. The electronics within the collar remained dry and in working condition, showing that the mechanical design of the collar provided sufficient protection. Some undesired results were also obtained. The first test ended prematurely because of battery failure. The cause of the premature battery drainage was the fact that a micro hard drive (with a CompactFlash form factor) was used instead of an authentic CompactFlash card. The micro hard drive consumed 600% more energy than an average CompactFlash card in a write cycle and also took longer to go into shut down mode after a write cycle. Only CompactFlash memory devices should be used in the recording collar, as the superior power efficiency justifies the additional cost of these devices.

The second test was completed without incident, but the results showed that the microphone was periodically obstructed, resulting in some portions of the recording being recorded at a much lower sound level. There was some rainfall during the time of the field test so it was concluded that the microphone was temporarily obstructed by a layer of mud. The mud eventually dried and fell off the smooth surface of the polycarbonate cover, thereby restoring the recordings to normal sound levels. Although the obstruction of the microphone by mud caused only a temporary problem, it indicates the need for improvement in the design to prevent this problem from re-occurring.

It was also observed that the recordings on the micro hard drive contained small periodic disturbances. This was caused by digital noise due to the excessive power consumption of the hard drive disturbing the reference voltage of the analogue to digital converter. The problem did not occur when recording on a CompactFlash card. Nevertheless, this finding indicated the possibility of the reference voltage becoming unstable, especially when the battery is running low, and the issue should be addressed by using a dedicated voltage reference in a future design.

One of the limitations of the electronic design was the fact that the sensitivity of the microphone could not be easily adjusted by the user. In a reviewed electronic design, the user should be able to set the sensitivity of the recordings to one of a range of predefined values. It would also be beneficial to design the system to operate at 3 V rather than 5 V. This will result in a battery pack half the size of the current one.

There are no existing scientific publications with which the automatic elephant rumble detection results of this study can be compared. The accuracy of the algorithm was verified against elephant rumbles that were manually identified using the method described in Chapter 3. A number of vocalizations were isolated from raw recordings that were made with a handheld recorder in the Kruger National Park. Automatic rumble detection performed on a number of vocalizations recorded under good conditions gave an accuracy of 90.47%. Although this is a promising result it should be remembered that the vocalizations used for the test were chosen to represent the quality of average elephant recordings with common background noises which is a subjective concept. A better understanding of the value of this algorithm could be established by receiving feedback from a number of different elephant researchers using the algorithm over a substantial period of time.

Far field recordings like those that have been made by a handheld recorder are prone to periodic noise within the infrasonic range like car engines and aeroplanes. If these sounds have a harmonic structure it could result in the algorithm falsely detecting the sounds as elephant rumbles. The harmonic nature of the elephant rumbles could also be lost if recordings are made from too far away and this could result in the algorithm missing some of the elephant calls. Although the elephant recording collar that was developed in this study should produce recordings that are better suited for use with the automatic rumble detection algorithm than recordings made by handheld recorders, some of the unwanted effects may still occur under undesirable recording conditions.

Chapter 6

CONCLUSION

The current chapter presents a summary of the procedures of the study and provides recommendations for further research.

6.1 SUMMARY OF THE WORK

1. An exploration of existing literature on elephant vocalizations and the application of speech processing techniques to animal sounds was conducted. This led to the identification of several shortcomings in the current body of knowledge. These include the need for the development of a reliable way to collect large quantities of acoustic data from elephants in the wild and a way to automatically isolate rumbles from these recordings. The different types of existing VAD techniques were invesigated to find a suitable technique that could be used for the automatic detection of elephant rumbles.

2. An elephant recording collar was designed and built in an attempt to provide a reliable means of recording large quantities of acoustic data from elephants in the wild. An electronic design was made from component level focussing on low power consumption, physical robustness, small size and lightweight. A PCB was designed for the physical realization of the electronic design.

3. The resulting device recorded 16-bit encoded sound digitally to onboard memory at a sampling rate of 3000 Hz. The recorded sound was stored on an array of

flash memory cards that have enough capacity to store sound continuously for a period of 133 days. Sound files were saved in the wave file format on an FAT32 file system together with GPS coordinates and temperature information.

4. The electronic recording device was fitted into an elephant collar and protected mechanically by moulding it into a hard plastic epoxy and covering the memory card section with a layer of polycarbonate.

5. Experiments were done to find the most effective way to mount the microphone under the polycarbonate sheet so that it remained physically protected, water-tight and able to pick up the desired bandwidth of sound frequencies.

6. An automatic elephant detection algorithm was developed by using an existing VAD technique as a basis. The algorithm divided the input signal into 32 sub bands and took the autocorrelation of each band in windowed frames. Correlograms with good SNRs were added together to form an estimate of the dominating pitch present in that time window.

7. A method was devised to determine the presence of an elephant rumble from the summed correlation data and then track the pitch within a rumble. A program was written to realize the algorithm using Matlab software. The output was displayed graphically as well as in table format.

8. The automatic rumble detection algorithm was tested on a combination of clearly recorded rumbles and rumbles occurring in common background noises, but under generally good conditions, resulting in 90.47% of the rumbles being correctly detected. Tests were done to find circumstances under which the algorithm would fail.

9. Finally, potential improvements of the algorithm to enable processing of non-ideal recordings were examined. This included modification of the algorithm to enable detection and tracking of overlapping elephant rumbles.

6.2 FUTURE WORK

The objective of the current study was to introduce new research tools to the field of acoustic elephant research. During the course of the study, a number of difficulties were encountered. Solutions to some of these problems were found, while others remained unsolved. As the study progressed and a better understanding of the issues at hand developed, several new ideas were conceived that might prove valuable in future work on the same topics.

It is preferable for the elephant recording collar to remain active for a very long period of time so that the amount of data obtained would justify tranquillizing the elephant. Currently, the collar has a battery pack weighing less than one kilogram. This battery can power the device for six months. The rapid technological advances made in the field of low power electronics should make it possible to redesign the electronics in a manner that would prolong the battery life. High density flash memory technology is also advancing at a steady pace which should make it possible to reduce the number of memory cards needed in the system. Reducing the operating voltage of the system to 3 V would result in a battery pack half the weight and size of the current design.

One of the unsolved problems encountered with the elephant recording collar was the temporary blockage of the microphone by mud. The possibility of replacing the existing microphone with a surface microphone could be explored. A surface microphone is a very sensitive microphone mounted on the backside of a surface like the polycarbonate sheet that can pick up sound waves that collide with the surface. This could provide much better protection against physical damage of the microphone and would prevent the possibility of the microphone getting clogged by mud. However, methods to prevent these highly sensitive microphones from recording unwanted noises if something should brush against the collar would still need to be investigated.

In this study, a number of different algorithms for VAD were considered as a possible base for the automatic detection of elephant rumbles. A technique that has been implemented successfully in the past for the detection of human speech in recordings containing background noise was chosen. This technique worked well, but the bank of autocorrelators used in the technique makes it computationally expensive. In the future, the use of other noise robust algorithms (Barros, Rutkowski, Itakura and Ohnishi, 2002) that are less expensive could be explored and the results compared to those of this study. The method devised in this study for the final detection and pitch tracking phase of the algorithm (from the raw autocorrelation data obtained from the first stages of the algorithm) appears to be less than optimal. The use of an optimal method (Wang and Willett, 2003) for this task could be explored and the trade off between the gained accuracy and reduced speed of the optimal method should be examined.

The elephant detection algorithm developed in this study could be used as a basis for further signal processing on elephant rumbles. Using the algorithm, a system could be developed that would automatically detect rumbles from a large collection of unprocessed recordings (like those obtained from the elephant recording collar), and automatically isolate and place the detected vocalizations in a database. Such a system could help to provide the necessary resources needed for work combining speech processing and elephant acoustics.

References

Adams, G. J. & Evans, R. J. 1994, "Neural networks for frequency line tracking", *IEEE Transactions on Signal Processing*, vol. 42, no. 4, pp. 936-941.

Ahmadi, S. & Spanias, A. S. 1999, "Cepstrum-based pitch detection using a new statistical V/UV classification algorithm", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333-338.

Barros, A. K., Rutkowski, T., Itakura, F., & Ohnishi, N. 2002, "Estimation of speech embedded in a reverberant and noisy environment by independent component analysis and wavelets", *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 888-893.

Bradbury, J. W. & Vehrencamp, S. L. 1998, *Principles of animal communication,* Sunderland: Sinauer Associates.

Chang, J. H., Kim, N. S., & Mitra, S. K. 2006, "Voice activity detection based on multiple statistical models", *IEEE Transactions on Signal Processing*, vol. 54, no. 6 I, pp. 1965-1976.

Chang, J. H., Shin, J. W., & Kim, N. S. 2004, "Voice activity detector employing generalised Gaussian distribution", *Electronics Letters*, vol. 40, no. 24, pp. 1561-1563.

Clemins, P. J. & Johnson, M. T. 2003, "Application of speech recognition to African elephant (Loxodonta Africana) vocalizations", Proceedings of the *IEEE International Conference on Acoustics, Speech and Signal Processing, 6-10 April,* Hong Kong, Volume 1, pp. 484-487.

Clemins, P. J. & Johnson, M. T. 2006, "Generalized perceptual linear prediction features for animal vocalization analysis", *Journal of the Acoustical Society of America*, vol. 120, no. 1, pp. 527-534.

Clemins, P. J., Johnson, M. T., Leong, K. M., & Savage, A. 2005, "Automatic classification and speaker identification of African elephant (Loxodonta africana) vocalizations", *Journal of the Acoustical Society of America*, vol. 117, no. 2, pp. 956-963.

Davis, A., Nordholm, S., & Togneri, R. 2006, "Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 412-423.

Garstang, M. 2004, "Long-distance, low-frequency elephant communication", *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology,* vol. 190, no. 10, pp. 791-805.

Hartmann, W. M. 1997, *Signals, sound and sensation*, New York: Springer.

Heffner, R. S. & Heffner, H. E. 1982, "Hearing in the elephant (Elephas maximus): absolute sensitivity, frequency discrimination, and sound localization.", *Journal of Comparative and Physiological Psychology*, vol. 96, no. 6, pp. 926-944.

Hung, W. W. 2002, "Use of fuzzy weighted autocorrelation function for pitch extraction from noisy speech", *Electronics Letters*, vol. 38, no. 19, pp. 1148-1150.

Katsiamis A. G., Drakakis E. M. & Lyon R. F. 2007, "Practical gammatone-like filters for auditory processing", *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 2007 article no. 63685.

Kim, Y. J. & Chung, J. H. 2004, "Pitch synchronous cepstrum for robust speaker recognition over telephone channels", *Electronics Letters*, vol. 40, no. 3, pp. 207-209.

Kunieda, N., Shimamura, T., & Suzuki, J. 2000, "Pitch extraction by using autocorrelation function on the log spectrum", *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, vol. 83, no. 1, pp. 90-98.

Langbauer Jr, W. R., Payne, K. B., Charif, R. A., & Thomas, E. M. 1989, "Responses of captive African elephants to playback of low-frequency calls", *Canadian Journal of Zoology*, vol. 67, no. 10, pp. 2604-2607.

Langbauer, J. 2000, "Elephant communication", *Zoo Biology*, vol. 19, no. 5, pp. 425-445.

Larom, D., Garstang, M., Payne, K., Raspet, R., & Lindeque, M. 1997, "The influence of surface atmospheric conditions on the range and area reached by animal vocalization", *Journal of Experimental Biology*, vol. 200, no. 3, pp. 421-431.

Lee, K. & Ellis, D. P. W. 2006, "Voice activity detection in personal audio recordings using autocorrelogram compensation", *Proceddings of the 9th International Conference on Spoken Language Processing, September 17-21*, Pittsburgh, USA, *Interspeec-2006*, p. 1753.

Leighty, K. A., Soltis, J., Leong, K., & Savage, A. 2008, "Antiphonal exchanges in African elephants (Loxodonta africana): Collective response to a shared stimulus, social facilitation, or true communicative event?", *Behaviour*, vol. 145, no. 3, pp. 297-312.

Leong, K. M., Burks, K., Rizkalla, C. E., & Savage, A. 2005, "Effects of reproductive and social context on vocal communication in captive female African elephants (Loxodonta africana)", *Zoo Biology*, vol. 24, no. 4, pp. 331-347.

Leong, K. M., Ortolani, A., Burks, K. D., Mellen, J. D., & Savage, A. 2002, "Quantifying acoustic and temporal characteristics of vocalizations for a group of captive African elephants (Loxodonta africana)", *Bioacoustics*, vol. 13, no. 3, pp. 213-231.

Leong, K. M., Ortolani, A., Graham, L. H., & Savage, A. 2003, "The use of low-frequency vocalizations in African elephant (Loxodonta africana) reproductive strategies", *Hormones and Behavior*, vol. 43, no. 4, pp. 433-443.

Li, Y., Zhang, R., Cui, H., & Tang, K. 2005, "Voice activity detection algorithm with low signal-to-noise ratios based on the spectrum entropy", *Qinghua Daxue Xuebao/Journal of Tsinghua University*, vol. 45, no. 10, pp. 1397-1400.

McComb, K., Reby, D., Baker, L., Moss, C., & Sayialel, S. 2003, "Long-distance communication of acoustic cues to social identity in African elephants", *Animal Behaviour*, vol. 65, no. 2, pp. 317-329.

Nadeu, C., Pascual, J., & Hernando, J. 1991, "Pitch determination using the cepstrum of the one-sided autocorrelation sequence", Piscataway, IEEE Press, pp. 3677-3680.

Noll, A. M. 1967, "Cepstrum pitch determination", *Journal of the Acoustical Society of America*, vol. 41, no. 2, pp. 293-309.

O'Connell-Rodwell, C. E., Arnason, B. T., & Hart, L. A. 2000, "Seismic properties of Asian elephant (Elephas maximus) vocalizations and locomotion", *Journal of the Acoustical Society of America*, vol. 108, no. 6, pp. 3066-3072.

Ogada, D. L., Gadd, M. E., Ostfeld, R. S., Young, T. P., & Keesing, F. 2008, "Impacts of large herbivorous mammals on bird diversity and abundance in an African savanna", *Oecologia*, vol. 156, no. 2, pp. 387-397.

Paris, S. & Jauffret, C. 2003, "Frequency line tracking using HMM-based schemes", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 2, pp. 439-449.

Payne, K. B., Thompson, M., & Kramer, L. 2003, "Elephant calling patterns as indicators of group size and composition: The basis for an acoustic monitoring system", *African Journal of Ecology*, vol. 41, no. 1, pp. 99-107.

Poole, J. H. 1999, "Signals and assessment in African elephants: Evidence from playback experiments", *Animal Behaviour*, vol. 58, no. 1, pp. 185-193.

Poole, J. H. & Moss, C. J. 1981, "Musth in the African elephant, Loxodonta africana", *Nature*, vol. 292, no. 5826, pp. 830-831.

Poole, J. H., Payne, K., Langbauer, J., & Moss, C. J. 1988, "The social contexts of some very low frequency calls of African elephants", *Behavioral Ecology and Sociobiology*, vol. 22, no. 6, pp. 385-392.

Poole, J. H., Tyack, P. L., Stoeger-Horwath, A. S., & Watwood, S. 2005, "Elephants are capable of vocal learning", *Nature*, vol. 434, no. 7032, pp. 455-456.

Ramirez, J., Gorriz, J. M., & Segura, J. C. 2007, "Statistical voice activity detection based on integrated bispectrum likelihood ratio tests for robust speech recognition", *Journal of* *the Acoustical Society of America*, vol. 121, no. 5, pp. 2946-2958.

Ramirez, J., Segura, J. C., Benitez, C., Garcia, L., & Rubio, A. 2005, "Statistical voice activity detection using a multiple observation likelihood ratio test", *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 689-692.

Rasmussen LEL. 1988, "Chemosensory responses in two species of elephants to constituents of temporal gland secretion and musth urine. ", *Journal of Chemical Ecolopgy.*, vol. 16, pp. 687-711.

Sankaran, M., Ratnam, J., & Hanan, N. 2008, "Woody cover in African savannas: The role of resources, fire and herbivory", *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 236-245.

Seiyama, N., Tohru, T., Tetsuo, U., & Eiichi, M. 1992, "New method of pitch modification by complex cepstrum analysis-synthesis", *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, vol. 75, no. 11, pp. 102-113.

Shimamura, T. & Kobayashi, H. 2001, "Weighted autocorrelation for pitch extraction of noisy speech", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 7, pp. 727-730.

Sohn, J. 1999, "A statistical model-based voice activity detection", *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1-3.

Soltis, J., Leong, K., & Savage, A. 2005a, "African elephant vocal communication I: Antiphonal calling behaviour among affiliated females", *Animal Behaviour*, vol. 70, no. 3, pp. 579-587.

Soltis, J., Leong, K., & Savage, A. 2005b, "African elephant vocal communication II: Rumble variation reflects the individual identity and emotional state of callers", *Animal Behaviour,* vol. 70, no. 3, pp. 589-599.

Takagi, T., Seiyama, N., & Miyasaka, E. 2000, "Method for pitch extraction of speech signals using autocorrelation functions through multiple window lengths", *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, vol. 83, no. 2, pp. 67-79.

Titze, I. R. 1994, *Principles of voice communication,* Englewood-Cliffs: Prentice-Hall.

Viljoen, P. J. & Bothma, J. D. P. 1990, "Daily movements of desert-dwelling elephants in the northern Namib Desert", *South African Journal of Wildlife Research*, vol. 20, no. 2, pp. 69-72.

Wang, Z. & Willett, P. 2003, "Two algorithms to segment white Gaussian data with piecewise constant variances", *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 373-385.

Woo, K. H., Yang, T. Y., Park, K. J., & Lee, C. 2000, "Robust voice activity detection algorithm for estimating noise spectrum", *Electronics Letters*, vol. 36, no. 2, pp. 180-181.

Wood, J. D., McCowan, B., Langbauer, J., Viljoen, J. J., & Hart, L. A. 2005, "Classification of African elephant Loxodonta africana rumbles using acoustic parameters and cluster analysis", *Bioacoustics*, vol. 15, no. 2, pp. 143-161.

Wu, M., Wang, D. L., & Brown, G. J. 2003, "A multipitch tracking algorithm for noisy speech", *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229-241.

Xie, X. & Evans, R. J. 1993, "Multiple frequency line tracking with hidden Markov models - further results", *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 334-343.

Zhang, T., Zhang, Z., Lin, X., & Quan, J. 2006, "Power spectrum reprocessing algorithm for pitch detection of speech", *Jisuanji Gongcheng/Computer Engineering*, vol. 32, no. 1, pp. 1-3.

Zhao, W. W. & Ogunfunmi, T. 1999, "Formant and pitch detection using time-frequency distribution", *International Journal of Speech Technology*, vol. 3, no. 1, pp. 35-49.