Voice Activity Detection in the Tiger Platform

Examensarbete utfört i Reglerteknik

av

Hampus Thorell

LiTH-ISY-EX--06/3817--SE
Linköping 2006
Voice Activity Detection in the Tiger Platform

Examensarbete utfört i Reglerteknik
vid Linköpings tekniska högskola
av

Hampus Thorell

LITH-ISY-EX--06/3817--SE

Handledare:  Mikael Olausson,
Sectra Communications AB

David Törnqvist
ISY, Linköpings universitet

Examinator:  Fredrik Gustafsson,
ISY, Linköpings universitet

Linköping, april 07, 2006
**Sammanfattning**
Sectra Communications AB har utvecklat ett终端 för krypterat kommunikation kallat Tiger-plattformen. Under språkliga kommunikationsstörningar har tidigare upplevts avstånd, vilket orsakat talsvansar och konversationella problem.


I denna avhandling har det teoretiska om språket talningsaktivitetssensorers studerats. En recensering av talningsaktivitetssensorer som finns på marknaden idag följt av en utvärdering av några av dessa skulle skapa en grundläggande förvaltning av en talningsaktivitetssensor för att implementera den på Tiger-plattformen. Denna utvärdering skulle senare bli den grundläggande för val av en talningsaktivitetssensor för implementeringen.

Förlagt, utvärderingen av valen som gjorts, inklusive en s.k. tillflyttfyllningsgenerator, gjordes på plattformen. Denna implementering var baserad på speciella krav på plattformen. Test av implementeringen i kontorsmiljö visade att möjliga tidsförluster var stabilt reducerade under perioder av talningsaktivitet, medan aktiv talingskvalitet är bevarad.

**Nyckelord**
Abstract

Sectra Communications AB has developed a terminal for encrypted communication called the Tiger platform. During voice communication delays have sometimes been experienced resulting in conversational complications.

A solution to this problem, as was proposed by Sectra, would be to introduce voice activity detection, which means a separation of speech parts and non-speech parts of the input signal, to the Tiger platform. By only transferring the speech parts to the receiver, the bandwidth needed should be dramatically decreased. A lower bandwidth needed implies that the delays slowly should disappear. The problem is then to come up with a method that manages to distinguish the speech parts from the input signal. Fortunately a lot of theory on the subject has been done and numerous voice activity methods exist today.

In this thesis the theory of voice activity detection has been studied. A review of voice activity detectors that exist on the market today followed by an evaluation of some of these was performed in order to select a suitable candidate for the Tiger platform. This evaluation would later become the foundation for the selection of a voice activity detector for implementation.

Finally the implementation of the chosen voice activity detector including a comfort noise generator was done on the platform. This implementation was based on the special requirements of the platform. Tests of the implementation in office environments show that possible delays are steadily being reduced during periods of speech inactivity, while the active speech quality is preserved.
Preface

This master thesis has been performed at Sectra Communications AB and it is the final part of my Master of Science Degree in applied physics and electrical engineering at the Linköping Institute of Technology. The work was done during autumn 2005 and spring 2006.

I wish to thank the following people for making this possible.

My mentor at Sectra, Mikael Olausson, who has always been very supportive and has taken a lot of his time to help me.

Robin von Post and Mikael Bertilsson for letting me do this thesis work at Sectra and for always showing great interest and being supportive in my work.

David Törnqvist and Fredrik Gustafsson at the Department of Electrical Engineering (ISY) for all help.
# Table of Contents

1 Introduction ............................................................. 1  
  1.1 Background .......................................................... 1  
  1.2 Problem description ................................................ 2  
  1.3 Objective and requirements ................................. 3  
  1.4 Method ................................................................. 4  
  1.5 Structure ............................................................... 4  

2 The Tiger platform .................................................. 6  
  2.1 The hardware ......................................................... 6  
  2.2 Speech coders ........................................................ 6  
    2.2.1 G.729D .......................................................... 6  
    2.2.2 MELPe .......................................................... 7  

3 Linear predictive coding .......................................... 7  
  3.1 Introduction .......................................................... 7  
  3.2 The computational components .......................... 7  
  3.3 The process .......................................................... 8  

4 Voice activity detection ........................................... 10  
  4.1 Background .......................................................... 10  
  4.2 Hangover .............................................................. 13  
  4.3 Comfort noise ....................................................... 13  
  4.4 Silence insertion descriptor frames ...................... 13  
  4.5 Methods for detecting speech ............................... 14  
    4.5.1 Energy detection ............................................ 14  
    4.5.2 Zero crossing rate ........................................... 15  
    4.5.3 The spectral shape ........................................... 15  

5 Voice activity detectors: An overview .................... 16  
  5.1 G.729B ............................................................... 16  
  5.2 GSM FR/HR/EFR .................................................. 17  
  5.3 Fuzzy VAD .......................................................... 18  
  5.4 GSM AMR1/2 ........................................................ 18  
  5.5 The Energy VAD ................................................... 19  

6 A published VAD evaluation ................................. 20  
  6.1 Description .......................................................... 20  
  6.2 Results ................................................................. 22
7 Evaluation of VADs .............................................................. 24
  7.1.1 G.729B .......................................................................... 25
  7.1.2 The Energy VAD .......................................................... 25
  7.1.3 GSM AMR1 ................................................................. 25
  7.1.4 GSM AMR2 ................................................................. 26
  7.2 The evaluation ................................................................. 26
    7.2.1 The sound quality ....................................................... 26
    7.2.2 Bandwidth savings ..................................................... 26
    7.2.3 Performance needs .................................................... 27
    7.2.4 Implementation complexity ......................................... 28
  7.3 Results ........................................................................... 28
    7.3.1 The sound quality ....................................................... 28
    7.3.2 Bandwidth savings ..................................................... 30
    7.3.3 Performance needs .................................................... 32
    7.3.4 Implementation complexity ......................................... 32
  7.4 Conclusion .................................................................. 33

8 Descriptions of the chosen VADs .................................... 35
  8.1 The Energy VAD ............................................................. 35
  8.2 G.729B ........................................................................... 36
  8.3 Parameter extraction ....................................................... 36
    8.3.1 Full band energy ....................................................... 37
    8.3.2 Low band energy .................................................... 37
    8.3.3 Zero crossing rate ................................................... 37
  8.4 Differential parameter calculations .................................. 37
    8.4.1 The spectral distortion .............................................. 37
    8.4.2 The full band energy difference ................................... 38
    8.4.3 The low band energy difference ................................... 38
    8.4.4 The zero crossing rate difference ................................ 38

9 Simulations with the selected VADs ......................... 39
  9.1 The reference codes ....................................................... 39
  9.2 Tests with the G.729D speech coder .......................... 40
    9.2.1 G.729D Test case 1 - G.729B without SID frames ...... 40
    9.2.2 G.729D Test case 2 - G.729B with Bad Frame Indication..... 41
    9.2.3 G.729D Test case 3 - G.729B with only initiating SID frames ...... 41
    9.2.4 G.729D Test case 4 - G.729B using the Energy VAD ...... 42
    9.2.5 G.729D Test case 5 - The Energy VAD with no CNG ...... 42
    9.2.6 G.729D Test case 6 - The Energy VAD with static white noise CNG .................................................. 43
    9.2.7 G.729D Test case 7 - The Energy VAD with adaptive white noise CNG .................................................. 43
  9.3 Tests with the MELPe speech coder ............................ 44
    9.3.1 MELPe Test case 1 - The Energy VAD with no CNG and the NPP off... 44
9.3.2 MELPe Test case 2 – The Energy VAD with no CNG and the NPP on ................................................................. 44
9.3.3 MELPe Test case 3 - The Energy VAD utilizing the frame erasure procedure with the NPP off .......................... 45
9.3.4 MELPe Test case 4 - The Energy VAD utilizing the frame erasure procedure with the NPP on .......................... 45

9.4 Conclusion ........................................................................... 46

10 The implementation ................................................................ 47
10.1 The development process ..................................................... 47
10.1.1 The chosen VAD ................................................................ 47
10.1.2 The chosen CNG ............................................................... 47
10.1.3 Object code recycling ....................................................... 48
10.1.4 The implementation verification process ............................. 48

10.2 Evaluation of the implementation ........................................... 51
10.2.1 Performance numbers ...................................................... 51
10.2.2 Memory requirements ..................................................... 52
10.2.3 Bandwidth savings ......................................................... 52
10.2.4 Sound quality testing ....................................................... 53

10.3 Problems encountered ......................................................... 53
10.3.1 Object converter ............................................................. 53
10.3.2 Interrupt routine in C-code ............................................... 53

11 Summary ............................................................................... 55
11.1 Continued work ................................................................. 55
11.1.1 A more thorough evaluation ............................................ 55
11.1.2 Performance optimization .............................................. 55
11.1.3 MELPe ........................................................................ 55

11.2 Conclusions ........................................................................ 56
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMR</td>
<td>Adaptive Multi-Rate</td>
</tr>
<tr>
<td>AR</td>
<td>Auto-Regressive</td>
</tr>
<tr>
<td>BFI</td>
<td>Bad Frame Indication</td>
</tr>
<tr>
<td>CCR</td>
<td>Comparison Category Rating</td>
</tr>
<tr>
<td>CMOS</td>
<td>Comparison Mean Opinion Scores</td>
</tr>
<tr>
<td>CNG</td>
<td>Comfort Noise Generator</td>
</tr>
<tr>
<td>CS-ACELP</td>
<td>Conjugate Structure Algebraic Code-Excited Linear Prediction</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DTX</td>
<td>Discontinuous Transmission</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunication Standardization Institute</td>
</tr>
<tr>
<td>FVAD</td>
<td>Fuzzy VAD</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Prediction</td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Predictive Coding</td>
</tr>
<tr>
<td>LSF</td>
<td>Line Spectral Frequency</td>
</tr>
<tr>
<td>LSP</td>
<td>Line Spectral Pair</td>
</tr>
<tr>
<td>MELPe</td>
<td>Enhanced Mixed-Excitation Linear Predictive</td>
</tr>
<tr>
<td>NPP</td>
<td>Noise Pre-Processor</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulated</td>
</tr>
<tr>
<td>SID</td>
<td>Silence Insertion Descriptor</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>VAD</td>
<td>Voice Activity Detector</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>WMIPS</td>
<td>Weighted Million Instructions Per Second</td>
</tr>
<tr>
<td>WMOPS</td>
<td>Weighted Million Operations Per Second</td>
</tr>
<tr>
<td>ZC</td>
<td>Zero Crossing</td>
</tr>
</tbody>
</table>
1 Introduction
This chapter explains background information, a problem description, the objective and requirements, the method that was used to solve the problem, and the structure of the report.

1.1 Background
Sectra Communications AB is developing systems for secure communication. The products are usually developed together with the customers to meet the high security demands set by the authorities within EU or NATO. The products are utilized by both civil and defense authorities.

Since the middle of the 90s Sectra has been working with a family of products for personal communication called Tiger. These are handheld units running on battery that offer encrypted speech and data services on a high security level, see Figure 1.

![The Tiger XS terminal.](image)

As a part of the ambition to constantly improve performance and the user’s possibilities to utilize the Tiger unit, Sectra is now interested in evaluating the advantages of integrating support for voice activity detection. Voice activity detection means that data is only transmitted to the receiver when speech is present. In other words only the unit belonging to the currently talking person should be transmitting information.

This thesis work involves analyzing and evaluating existing algorithms to decide whether there is any speech activity implying that the data should be transmitted to the receiver. The environments of Sectra’s customers as well as the
implementation and protocols of the Tiger products have to be taken in consideration.

1.2 Problem description

The basic flow of voice communication in the Tiger platform and the problem that can appear can be seen in Figure 2.

![Diagram of voice communication in the Tiger platform](image)

Figure 2. Voice communication in the Tiger platform.

The transmitting Tiger unit takes the incoming speech, performs speech encoding, encrypts it, and transmits the encrypted data through, for example, a Bluetooth channel to a GSM unit. The GSM unit is then set up to pass on the data through the circuit-switched GSM data channel to the receiving GSM unit. It is during this transmission through the data channel that the problem is likely to occur. Since the GSM data channel is adapted to data and not speech there can be delays which can be time critical in voice communication but not in data communication. For example if a data packet becomes corrupted or is lost there is a protocol, which forces a new transmission of this packet while the following packets are buffered up. This protocol is not possible to turn off since the GSM unit is separated from the Tiger. It is not possible to use the voice channel in GSM either, at least not at any higher bit rates, since it is encrypted data and not simply encoded speech that is being transmitted.

When using the Tiger in voice communication the buffering can be very evident. The phenomenon is very similar to when talking to a person very far away, for
example on the other side of the planet, on a regular phone. The natural delays introduced because of the distances can lead to complications in the conversation such as when the two parties speak at the same time. With the current configuration in the platform there is no chance to catch up in the data flow once a delay has been introduced which means that this delay will be present until the end of the transmission (conversation).

The problem described here should be possible to solve by introducing voice activity detection since this makes it possible to reduce the amount of data that needs to be transferred. By doing so it should be possible to catch up once a delay has arisen. A new problem then appears, namely how to determine the voice parts of the signal from the rest of it. This leads into the objectives of this thesis.

### 1.3 Objective and requirements

The main objective of this thesis work is to reduce the amount of transferred data by introducing voice activity detection.

More specific the goal of this thesis work can be divided into the following requirements:

- Gain knowledge in the area of voice activity detection.
- Evaluate existing voice activity detectors (VADs).
- Choose one or more VAD for implementation. When selecting one the following has to be taken in consideration:
  - The quality of the sound. Distortions of the synthesized (reconstructed) speech that are critical to the comprehension should be avoided. It is primary that the existing sound quality of the speech should be preserved.
  - Performance and memory use. Resources on the platform needed by the VAD should be minimized. This is important since the voice activity detection is not the primary function.
  - Bandwidth savings. Since the main goal of the introduction of a VAD is to reduce the amount of exchanged information this is one of the most central parts of this thesis work.
  - The implementation complexity. This means taking into account the structure of the algorithms and the number of functions needed to realize them. A high complexity results in high memory use. This is of importance since the time to implement the VAD functionality and later maintain it should be minimized.
• The implementation should be done primarily for the G.729D speech coder. If there is time left an implementation for the MELPe speech coder should also be done.

• The primary programming language of the implementation is C and should be used where it is possible. The secondary language is Assembler.

1.4 Method
This thesis work can be divided into four parts.

The first part involves studies on voice activity detection theory. During this period information on the subject was collected and reviewed. This was done during the first three weeks.

The second part was to analyze and evaluate existing VADs on the market. This was done mainly with the help of reference ANSI C-code published by different standard organizations. This took approximately five weeks.

The third part, which was the greater part of this thesis work, was to implement a selected silence compression scheme consisting of a VAD and a comfort noise generator in the Tiger platform. This implementation took around nine weeks to finalize.

The last part was to document the results and write this report, which took the rest of the time.

1.5 Structure
The structure of the report is as follows.

Chapter 2 gives information on the Tiger platform that is relevant for the implementation.

Chapter 3 gives a simplified explanation of linear predictive coding, a very common speech coding technique that is important for the understanding of voice activity detection theory.

Chapter 4 deals with voice activity detection theory, going into subjects such as discontinuous transmission and comfort noise generation.

Chapter 5 goes through some of the most common VADs on the market today, explaining briefly how they work and their differences.

Chapter 6 describes one of the many VAD evaluation papers that have been published. This should give a hint of the quality of VADs and how to perform such an evaluation.
Chapter 7 covers a VAD evaluation that was performed as a part of this thesis work. This evaluation is more adapted to the requirements in this thesis than the one described in chapter 6.

Chapter 8 gives a more thorough description of the VADs that were chosen for implementation.

Chapter 9 contains test information provided from various simulations on the chosen VADs.

Chapter 10 describes the implementation, how it was developed, evaluated and some problems encountered. It also proposes future work.
2 The Tiger platform

This chapter presents information on the Tiger platform relevant for the implementation. The parts that deal with this are the hardware and the speech coders that are implemented.

2.1 The hardware

In the Tiger platform there is a chip with an integrated digital signal processor (DSP). This DSP uses fixed-point arithmetic which is important to keep in mind before and during implementation.

2.2 Speech coders

Speech coding can be described as a compression of speech into code for transmission purposes. The compression is performed with special consideration taken to the characteristics of speech. This becomes especially important at low bit rates since only the most essential information about the speech must be included in the transmitted bit stream.

In the Tiger platform the following two speech coders are implemented:

- G.729D
- MELPe

The choice of which speech coder to use during the voice communication can be done manually. Usually this selection is made automatically depending on the allocated bandwidth during connection setup.

A closer look on these two speech coders follows.

2.2.1 G.729D

G.729 is an ITU-T standardized speech coder [1] commonly used in Voice over IP (VoIP) contexts and other low bandwidth communication techniques. G.729 was originally specified for a bit rate of 8 kbit/s but there are a couple of annexes specifying modifications to the original standard. Common to all these annexes is the linear prediction technique called conjugate structure algebraic code-excited linear prediction (CS-ACELP) that is used for the speech coding.

The version of G.729 implemented in the Tiger platform is G.729 annex D, commonly known as G.729D, which is specified for a bit rate of 6.4 kbit/s and speech frames of 10 milliseconds [2].

The implementation of G.729D in the Tiger platform consists of object-code.
2.2.2 MELPe

The second speech coder implemented is called MELPe (Enhanced Mixed-Excitation Linear Predictive) [3]. MELPe is a military standard adopted by NATO also known as MIL-STD-3005 and STANAG 4591.

The technique behind MELPe is also based on linear predictive coding and it is specified for extremely low bit rates such as 600 bit/s, 1200 bit/s and 2400 bit/s where the latter works with speech frames of 22.5 milliseconds. There is also a so-called Noise Pre-Processor (NPP) integrated with the speech coder to suppress background noise.

The version implemented in the Tiger platform is the one specified for 2400 bit/s and it is implemented in Assembler code.

3 Linear predictive coding

The speech coding technique called linear predictive coding (LPC) is explained in this chapter. Knowledge of this will be helpful to understand some of the basics of voice activity detection and other terms belonging to this area.

3.1 Introduction

The idea of LPC coding is to build a model of the speech that is based on the strong correlation that exists between adjacent samples. Instead of transferring the waveform itself only the parameters of the model will be transferred to the receiver (decoder). The decoder then rebuilds the model and generates speech very similar to the original. In this way only the essential information of the sound has been transferred, optimizing the synthesized speech.

3.2 The computational components

The method tries to predict the sample of an input signal based on several previous samples. In (3.1) the sample $\tilde{s}[n]$ is estimated as a linear combination of $N$ previous samples. The equation is called an Auto-Regressive model (AR model) [5].

$$\tilde{s}[n] = \sum_{k=1}^{N} a_k \cdot s[n-k]$$

(3.1)

The number of previous samples decides what is called the order of the model and the higher number the more correct becomes the prediction. This will however also mean that the computation complexity increases [6].

The terms $a_k$ are called the LPC coefficients or sometimes the LP (linear predictive) coefficients and they are chosen in such a way that the squared error
between the real input sample and its predictive value is minimized [6]. The error, \( e[n] \), is called the predictive error or the residual [5], see (3.2).

\[
e[n] = \tilde{s}[n] - \sum_{k=1}^{N} a_k \cdot s[n-k]
\]  

(3.2)

By transferring this to the frequency plane with the z-transform we get (3.3) [4, 5, 6].

\[
E(z) = S(z) - \sum_{k=1}^{N} a_k \cdot S(z) \cdot z^{-k} = S(z) \cdot \left(1 - \sum_{k=1}^{N} a_k \cdot z^{-k}\right) = S(z) \cdot A(z)
\]  

(3.3)

The error signal is now represented as the product of the original input signal \( S(z) \) and the transfer function \( A(z) \) which is also called the analysis filter [5], see (3.4).

\[
A(z) = 1 - \sum_{k=1}^{N} a_k \cdot z^{-k}
\]  

(3.4)

Finally, the inverse of the analysis filter is called the synthesis filter [5], see (3.5).

\[
\frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^{N} a_k \cdot z^{-k}}
\]  

(3.5)

### 3.3 The process

In the speech encoding process, see Figure 3, the first step is to compute the LPC coefficients, the \( a_k \) values. It is then possible to create the \( A(z) \) filter which produces the excitation, \( e[n] \), from the original signal \( s[n] \).

![Figure 3. The LPC encoding process.](image)

In the decoding process, see Figure 4, the LPC coefficients are used for constructing the synthesis filter which is fed with the excitation. This will then give the synthesized version of the original signal.
In simplified terms, what needs to be transferred to the decoder is the LPC coefficients and the excitation. The speech coders G.729D and MELPe are however much more sophisticated than that. For example, in G.729D an alternative representation of the LPC coefficients [7], called line spectral pairs (LSP) and excitations are stored in codebooks in which the values have been calculated in advance to fit every speech structure. The LPC to LSP conversion is done since the structure of these coefficients are better suited for interpolation and quantization. The information transferred from the encoder to the decoder is then indexes to the proper coefficients and excitations in the codebooks which of course are the same for both the encoder and the decoder.

This is as mentioned earlier a very simplified view of the LPC and there are many more functions added to the speech coders in the Tiger platform. This explanation should however be enough for the comprehension of the terms used in this thesis.
4 Voice activity detection

In this chapter, the theory of voice activity detection and some of the terms and methods used in the area will be dealt with.

4.1 Background

The theory of voice activity detection was developed since it was discovered that during a conversation between two persons the average time in which speech exists ranges from 40-50% [8]. This is very easy to realize if one considers the opposite situation since it would mean that the two persons would speak at the same time.

By using voice activity detection in mobile communications and other bandwidth limited situations and only transmitting information when speech is present, the amount of data needed to be transferred can be dramatically decreased. There are many benefits of doing this; some of them are described here.

- The bandwidth needed in packet switched networks decreases. In, for example VoIP situations the bandwidth required would be reduced since not all the information present during a voice communication needs to be transmitted.

- The power consumption in mobile terminals decreases. When looking at mobile phones the standby time is often many hundreds of hours while the talk time is usually below 10 h. It is obvious here that the power consumption goes up when transmitting data. By only transmitting during speech periods, the power consumption can be lowered which can give longer battery time.

- In cellular mobile phone systems there is always a problem called co-channel interference. Since voice activity detection implies less transmission time, the interference of nearby cells will also be lowered.

There are however some drawbacks which are important to consider.

- The complexity increases. By introducing voice activity detection, there will be a need to add extra features, which will increase the overall complexity and memory requirements.

- The sound quality is reduced. One can not assume that all VADs work in an ideal manner. Now and then they will all do incorrect decisions and this can lead to clippings of the speech. Clippings mean that speech is marked as noise and is therefore clipped and not transmitted. The clippings might then lead to a reduced intelligibility since parts of the conversation are removed.
A simplified voice activity detection model can be seen in Figure 5. The incoming signal, which is composed by speech and background noise, has first been divided into smaller units called speech frames. The speech frames usually have duration of 10-20 milliseconds. Before any speech coding is performed the frames are sent on to the VAD. The VAD then extracts one or more parameters from the sound, for example the energy. In the next step each parameter is compared to a threshold value which can be adaptively updated. If the value of the parameter is lower than the threshold value the current frame is marked as having non-active voice contents (it is said that the frame is inactive). If the value is higher than the threshold the frame is on the other hand marked as a frame with active voice contents (the frame is active). The voice activity decision can of course be based on several parameter comparisons and, for example, if all parameters are lower than their respective thresholds the current frame is marked as inactive.

The active voice frames are transmitted to the speech coder in a regular behavior as if there was no voice activity detection. The non-active voice frames on the other hand are coded in such a way that the receiver will understand that no voice is present in the current incoming frames. For instance, when an inactive period is detected a special frame containing information about this could be transmitted. The information should then differ from the regular active frames containing speech-coding data. The only purpose of the information would in other words be to tell the receiver that no active frames will be transmitted. If this information is not transmitted there is a risk that the receiver will believe that the connection is lost since no more frames are received. If the transmitted later detects speech the transmission of active frames will start again. This behavior where only some of the frames are transmitted is commonly known as discontinuous transmission (DTX).
Figure 5. A basic voice activity detection model.

In Figure 6 a typical speech sequence can be seen. The speech comes in spurts and there is also some noise in the background. In Figure 7 the output signal of an ideal VAD can be seen. In the figure the signal’s value equals 1 in the presence of speech while it is 0 otherwise.
4.2 Hangover

If the level of a certain parameter in the current frame is lower than a chosen threshold value this could mean, as mentioned earlier, that this frame should be marked as inactive. One alternative is then to not transmit this frame to the receiver. However, it is very common in many VADs to wait for several frames in a row to be below the threshold level before actually marking the current frame as inactive, and commencing a period of voice inactivity. This methodology is called hangover and the reason for doing so is to prevent the clipping of the end of sequences of speech. This could very easily happen otherwise, especially in energy based VADs, since it is a fact that the energy often is very low in this region.

4.3 Comfort noise

When there is no speech being detected at the transmitter’s side of a communication link the transmission is halted. At least this is the scenario that is wanted when using voice activity detection. The question is then what the receiver should hear or what the decoder should decode since there is no data present on the communication channel. The easiest way to solve this would be to simply not playback anything at all, meaning that the receiving person would only get silence from the decoder during the inactive periods. However this is not to be recommended. The reason for this is that it becomes very hard for the receiving person to determine if it actually is silence that is heard. It could might as well appear as though the sender has hung up or as if the communication link has been broken. The silence will appear the same either way for the receiver.

The solution is to introduce something called comfort noise, which is noise that is added at the receiver’s side in inactive periods the for receiving person to believe that it actually is the original background noise that is heard. The behavior is carried out with the help of a comfort noise generator (CNG) which is activated at the receiver’s side when no active speech frames are being received. The noise that is generated can either be simple random Gaussian white noise or noise generated with a technique similar to the one used for speech coding which is based on information from the actual background noise at the transmitter’s side.

4.4 Silence insertion descriptor frames

As earlier mentioned the frames are divided into active voice frames and non-active voice frames. The non-active voice frames can be further divided into silence insertion descriptor (SID) frames and empty frames. The information in a SID frame consists of parameters extracted from the background noise at the transmitter’s side and is used for comfort noise generation at the receiver’s side. The difference between the SID frames and the empty frames is in other words
that the SID frames will be transferred to the receiver even though they do not contain any speech while the empty frames will not be transmitted at all. Further it can be said that the size of the SID frames is usually much smaller than the active voice frames.

The SID frames are always created and transmitted in the beginning of periods of inactivity. They are however also generated and transmitted if a sudden big change in the background noise at the transmitter’s side occurs. This means that a frame can be transmitted in the middle of inactivity even though no speech has been present. The reason for this behavior is that a change in the background noise at the transmitter’s side might be permanent when going into sequence of speech. If the information being sent to the CNG has not been updated at the receiver’s side there might be rough transitions when going from inactivity to activity in the background sound that also is apparent in the speech.

4.5 Methods for detecting speech

There are many different techniques for detecting speech. The techniques differ in efficiency and complexity. Usually a high complexity involves a more correct behavior of the VAD while a low complexity introduces more clippings of the speech.

The following techniques will be discussed here:

- Energy detection
- Zero Crossing Rate
- Spectral shape

These techniques can be applied to either the whole signal or just portions of it.

4.5.1 Energy detection

The simplest way to detect speech is probably by measuring the energy of the signal. Active voice contents usually result in a higher energy than the background noise does. The drawback is that when the background noise reaches high intensities and the signal-to-noise ratio (SNR) therefore drops the energy of the background noise can be very similar to the energy of the speech.

One way to avoid this problem is to divide the signal into sub-bands and measure the energy in each band with the purpose of calculating the SNR for each band. The dividing up into sub-band can be achieved by for example the use of filter banks or frequency domain transformations.
4.5.2 Zero crossing rate
Another quantity often measured and used in VADs is the zero crossing rate. This means how often the amplitude of the signal changes from positive to negative. The advantage of this parameter is that the zero crossing rate of signal only consisting of white noise which, including all frequencies with the same probability is significantly higher than a signal composed of both background white noise and speech [4].

The use of the zero crossing rate can be very effective in high SNR environments while it becomes less reliable when the SNR decreases [9].

4.5.3 The spectral shape
Another method is to look at the spectral shape of the input signal. The reason for this is that the energy of the predictive coding error, the residual, increases when there is a mismatch between the shapes of the background and the input signal.
5 Voice activity detectors: An overview

In this chapter some of the most common VADs will be discussed. These are the following:

- G.729B
- GSM FR/HR/EFR
- Fuzzy VAD
- GSM AMR1/2

Finally a simple energy based VAD proposed, but not tested, by Sectra was included in the discussion. This will further on simply be called the “The Energy VAD”.

- The Energy VAD

5.1 G.729B

G.729B is an extension for the G.729 speech coder standardized by ITU-T, which specifies what is called a silence compression scheme [10]. The scheme consists of a VAD and a CNG.

The VAD algorithm of G.729B makes selections for every 10 milliseconds on speech frames consisting of 80 samples. From the speech frames the following parameters are extracted:

- The full band energy.
- The low band energy (0 – 1 kHz).
- The line spectral frequency (LSF) coefficients (another representation of the LPC coefficients which is based on the LSPs)
- The zero crossing rate.

Differences between the four parameters extracted from the current frame and running averages of the noise are calculated for every frame. The differences will represent the noise characteristics. Large differences will then imply that the current frame is voice while the opposite implies that there is no voice present. The decision made by the VAD is based on a complex multi-boundary algorithm.

In the standard there is also a recommendation for comfort noise generation where the information of the original background noise is transmitted in SID frames. The size of the SID frames is only 16 bits while it is 80 bits for the speech frames.
The original G.729 speech coder and the annex B both were specified for a bit rate of 8 kbit/s but there is also an annex F specifying reference C-code for running G.729B in 6.4 kbit/s [11]. A summary displaying the differences of the different G.729 annexes can be found in Table 1.

<table>
<thead>
<tr>
<th>Speech coder</th>
<th>Bit rate</th>
<th>DTX</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729</td>
<td>8.0 Kbit/s</td>
<td>No</td>
</tr>
<tr>
<td>G.729B</td>
<td>8.0 Kbit/s</td>
<td>Yes</td>
</tr>
<tr>
<td>G.729D</td>
<td>6.4 Kbit/s</td>
<td>No</td>
</tr>
<tr>
<td>G.729F</td>
<td>6.4 Kbit/s</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Table 1. Some of the different G.729 variants.*

### 5.2 GSM FR/HR/EFR

In the GSM system there are three speech coders all working in a very similar way. They are called half rate (HR), full rate (FR) and enhanced full rate (EFR) and are all standardized by ETSI [12, 13, 14].

The biggest difference between the three speech coders is the bit rate which is 13 kbit/s, 5.6 kbit/s and 12.2 kbit/s for FR, HR and EFR respectively. Integrated to these speech coders are VADs which all are specified in a very similar way.

The voice activity decision is based on speech frames of 20 milliseconds and it compares the predictive residual energy, the energy of the LPC analysis filter, with a threshold. The energy is computed using the current and autocorrelation values of past frames which gives a good description of the spectral contents of the signal. It is here assumed that if the signal only contains background noise the average spectral shape will result in smaller residual energy since noise is considered stationary. The threshold is adaptive and is updated during periods of noise. Finally it can be mentioned that there is also comfort noise and SID frame generation included in the standard.

A summary of the three speech coders is found in Table 2.
<table>
<thead>
<tr>
<th>Speech coder</th>
<th>Bit rate</th>
<th>DTX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM FR</td>
<td>13.0 Kbit/s</td>
<td>Yes</td>
</tr>
<tr>
<td>GSM HR</td>
<td>5.6 Kbit/s</td>
<td>Yes</td>
</tr>
<tr>
<td>GSM EFR</td>
<td>12.2 Kbit/s</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Table 2. The three GSM speech coders.*

5.3 Fuzzy VAD

Fuzzy VAD or FVAD is a further development of G.729B which has not yet been standardized by any organization [15]. The algorithm is in many ways the same as G.729B which means that the same parameters are extracted and the same differential calculations are done. The rules that decide whether a frame is to be considered as speech or not are somewhat different and are based on a fuzzy system. The fuzzy systems imply approximations rather than preciseness. In this VAD method it results in a continuous voice activity output instead of a discrete one as can be found in G.729B and most other VADs.

5.4 GSM AMR1/2

Besides the previously mentioned speech coders for GSM there is also one called Adaptive Multi-Rate (AMR), which is also standardized by ETSI. This speech coder is developed for real-time transitions with the following bit rates: 4.75 kbit/s, 5.15 kbit/s, 5.9 kbit/s, 6.7 kbit/s, 7.4 kbit/s, 7.95 kbit/s, 10.2 kbit/s and 12.2 kbit/s. However there also two VADs simply called AMR1 and AMR2 specified for this speech coder.

AMR1 works with speech frames of 20 milliseconds and decomposes the signal into nine sub bands. A filter bank, where low frequencies are given low bandwidths while the higher frequencies are given higher bandwidths, is then used. The algorithm then calculates the SNR in all the different bands. The energy of the background noise is used for calculations of the SNR and is calculated from an adaptive model.

AMR2 is also specified for speech frames of 20 milliseconds but the VAD decision is made based on every 10 milliseconds. In other words every frame is divided into two parts.

The algorithm divides the signal into sub bands, similar to the AMR1 case, where SNRs are calculated for every band and the VAD decision is based on these ratios. A big difference between AMR1 and AMR2 is the fact that AMR2 uses the discrete Fourier transform (DFT) to transform the signal to the frequency domain. This transformation is equivalent to the filter banks used in AMR1 but the difference is that the SNRs are calculated in 16 frequency bands.
instead of nine. The SNRs are calculated based on the spectra of the signal and the background noise. The energy of the background noise is calculated for every band with an AR model.

As with earlier techniques there is also a CNG, which uses information from the background noise of the transmitter’s side, included in these VADs. The parameters of the backgrounds noise are transmitted in SID frames.

5.5 The Energy VAD

This VAD is basically an energy detector since it for every frame calculates the energy and compares this to an adaptive threshold value. The threshold value is a weighting of the lowest and highest energy detected. If the energy of the current frame is lower than the threshold it is marked as inactive.

There is no CNG specified for this so simple random Gaussian white noise will be used for comfort noise generation.
6 A published VAD evaluation

In this chapter one of the many evaluations of VADs that has been published is presented. The reason for including this is to get a more general opinion of the performance of some of the most common VADs.

The results attained in this evaluation prove to be significant for most evaluations done when it comes to G.729B and the AMR VADs which can also be seen in [4] and [17].

6.1 Description

In [18] F. Beritelli, S. Casale and S. Serrano in the Department of Informatics and Telecommunication Engineering, University of Catania performed an evaluation on the following VADs:

- G.729B
- AMR1
- AMR2
- Fuzzy VAD

The evaluation was performed to compare the Fuzzy VAD, which was developed by the researchers themselves, with some of the standardized VADs. The evaluation was first divided into an objective and a subjective part. In the objective part the goal was to evaluate the amount of speech that was classified as background noise, and also the amount of background noise that was classified as speech by the different VADs. The output signals from the VADs were compared to a database of constructed ideal VAD output signals where the active and non-active voice regions were marked manually. The goal of the subjective part was to let listeners grade the degradation of the sound when using the proposed VADs.

The following parameters where studied during the tests, see also Figure 8:

- FEC (Front End Clipping): clipping introduced in passing from noise to speech activity.
- MSC (Mid Speech Clipping): clipping due to speech misclassified as noise.
- OVER: noise marked as speech due to the VAD output remaining active in hangover periods.
- Noise Detected as Speech (NDS): noise interpreted as speech during a period of silence.
- ABC (Activity Burst Corruption): a way to mathematically measure the sound quality. This is based on the intensity of the sound.

![Diagram of VAD output]

Figure 8. Some of the terms used.

The subjective part of the test was performed with the help of 24 persons with an equal number of men and women. The idea of the test was that the persons were to evaluate the degradation of the sound quality when using the four VADs compared to an ideal VAD. The result was measured with comparison mean opinion scores (CMOS) which is a comparison category rating (CCR) technique proposed by ITU-T. CMOS tests are common in telecommunication contexts and the basic idea is that a number of listeners compare the sound quality of a processed signal with a reference signal and giving it a score. An arithmetic average of all scores are then calculated which gives the CMOS of the current object to be measured.

To be able to do a fair judgment of the VADs separate databases of speech were used for the objective and subjective tests. The database for the objective tests consisted of 192 files of speech each lasting 3 minutes where 40 % of the content was speech. Further, the speech consisted of both male and female voices spoken in languages such as Italian, English, French and German. The speech was mixed with different kinds of background sounds (car, office, train, restaurant and street) with different intensities resulting in different SNRs. The database for the subjective tests was very similar to the one for the objective tests but with the difference that the speech was only in Italian and lasted for ten seconds.

The speech was coded with the AMR speech coder at a constant bit rate of 7.95 kbit/s when evaluating the AMR1 and AMR2 VADs. For G.729B and the FVAD the speech coder G.729 was used with the 8 kbit/s bit rate.
6.2 Results
A summary of the previously mentioned parameters can be seen in Figure 9. The performance is measured in a percentage of error relative to an ideal VAD. The term “total error” is a summation of all the parameters measured.

![Graph showing percentage of error from the entire database](image)

*Figure 9. Percentage of error from the entire database, fig 1 in [18].*

The graph shows that AMR1, AMR2 and the FVAD gave very similar results and also somewhat better than G.729B. This is especially significant when looking at the total error.

It was also discovered that AMR1 and AMR2 works very well in environments of poor sound quality while the FVAD did not work as well but still better than G.729B.

The results from the subjective tests can be seen in Figure 10 and it is evident here that the differences are much lesser when measuring the performance of the VADs subjectively than objectively. When looking at the total error in Figure 9 the difference between AMR1 and G.729B is round 67 %. In Figure 10 this difference is only 26 %. Obviously the high number of errors attained with G.729B consists of errors that actually can not be heard by the average listener. The outcome of this part of the evaluation can also be considered to be more important than the objective part since it measures what a person actually can hear. It is important to remember that it is a very essential part of a VAD that it does not distort the speech in such a way that the listener can not understand the contents.
Figure 10. The results from the subjective tests, fig 6 in [18].

One important thing to mention about this evaluation is that only the correctness in the sense of marking inactive frames and the quality of the synthesized speech was studied. Other important matters when selecting a VAD such as complexity and performance were not studied, or at least not published, in this evaluation. These two factors along with a measurement of the bandwidth saved relative to the degradation of the synthesized speech would also have been very interesting when applying the VAD into and embedded system such as the Tiger platform.
7 Evaluation of VADs

In order to make good choice in selecting a VAD for implementation in the Tiger platform four VADs mentioned in chapter 5 were chosen for further evaluation. The VADs that were selected are the following:

- G.729B
- The Energy VAD
- GSM AMR1
- GSM AMR2

Even though some of these VADs had already been evaluated in [4, 17, 18] the results can not be directly applied to the goal of this thesis. The published evaluations were all concentrating on which VAD was providing the best results in the matter of finding most correct active and non-active voice frames and also the sound quality of the synthesized speech. For the objective of this thesis it is however as important to measure the performance needs and the amount of code that needs to be written in order to realize them.

To motivate the selection of four proposed VADs the following three criteria were studied:

- The possibilities for simulation.
- The possibilities for implementation.
- The subjective opinions.

These criteria were important since the most optimal VAD should be chosen and implemented in a relatively short and limited time.

The criteria for each VAD are handled in the following tables.
### 7.1.1 G.729B

| Simulation: | Since there is ANSI-C code to be used as reference for implementation and for simulation available from ITU-T it will be easy to simulate this VAD. |
| Implementation: | The conditions for implementation are good since the ANSI-C code is developed for the G.729 speech coder and for fixed-point implementations. The code should therefore be relatively simple to use for implementation with the G.729 speech coder. |
| Opinions: | The opinions about the performance of this VAD are pretty good since it is standardized by ITU-T and it is also often used as a reference in VAD evaluations. A drawback is the poor performance in noisy environments mentioned in [18]. |

*Table 3. G.729B.*

### 7.1.2 The Energy VAD

| Simulation: | Since the structure of the VAD is very simple it should be easy to write a fixed-point reference in C-code for simulation. |
| Implementation: | Because of the simplicity of this VAD there should be no problem to implement this VAD for any speech coder. |
| Opinions: | There are no opinions of the performance of this VAD since it was only a suggestion from Sectra. However, the simplicity speaks against it. |

*Table 4. The Energy VAD.*

### 7.1.3 GSM AMR1

| Simulation: | As with the case of G.729B there is ANSI-C code to be used as reference for implementation and for simulation available from ETSI which makes it easy to simulate this VAD. |
| Implementation: | The implementation of this VAD for the speech coders in the Tiger platform would involve a lot of work since it is adapted to a completely different speech coder. |
| Opinions: | The opinions on this VAD are very good when it comes to the quality of the sound and the correctness of the detection. It often gets good results in published tests and it is also an ETSI standard. A drawback is that it is considered to be very complex [18]. |

*Table 5. GSM AMR1.*
7.1.4 GSM AMR2

| Simulation: | The ANSI-C code that is used for AMR1 also includes AMR2, which means that it will be easy to simulate this VAD as well. |
| Implementation: | As with AMR1 this VAD is adapted to the AMR speech coder which will make it complex to implement in the Tiger platform. |
| Opinions: | The opinions concerning the quality of the sound and the correctness of the detection are very good. It often gets good results in published tests, usually even better than AMR1 and it is also an ETSI standard. A drawback is that it is also considered to be very complex [18]. |

Table 6. GSM AMR2.

7.2 The evaluation

In 1.3 the requirements of this thesis were declared, when selecting a suitable VAD the following four criterions where mentioned:

- The sound quality of the synthesized speech.
- The bandwidth savings.
- The performance needs.
- The implementation complexity.

These criterions were studied further in this evaluation with the help of 19 sound files. The sound files consist of male and female speech consisting of different languages and intensities. There was also a variety of background noise consisting of white noise, car noise and babble.

For the Energy VAD and G.729B the G.729D (6.4 kbit/s bit rate) speech coder was used while the AMR speech coder (6.7 kbit/s bit rate) was used for the AMR1 and AMR2 VADs.

7.2.1 The sound quality

The evaluation of the synthesized sound quality is based on purely subjective opinions. The criteria here are the amount of clippings and how well the artificial comfort noise is integrated with the active voice frames.

7.2.2 Bandwidth savings

The evaluation of bandwidth savings is based on a measurement of the number of non-active voice frames relative to the number of active voice frames. This was measured as an average of all speech frames and all sound files. The average is not to be considered as a fixed value that will be attained after an implementation since the result is very much depending on the contents of the
sound. For example, a sound sequence with very little speech and lots of silence will result in lots of non-active voice frames and therefore a lot of bandwidth will be saved.

Since the Energy VAD is not specified for the use of SID frames the evaluation will also include measurements where SID frames are counted as non-transmitted frames. The reason for this is to get a fair comparison.

### 7.2.3 Performance needs

To measure the performance needs, a library of functions called basic_op was used. This library consists of Assembler like functions that perform arithmetic operations. In this library every function is given a number that is proportional to the amount of clock cycles it is supposed to take for it to be executed in the DSP. The library is developed by ETSI and is integrated into the reference codes of G.729B and AMR1/2 and also in the Energy VAD.

The quantities that are measured are the sum of WMOPS (Weighted Million Operations Per Second) and the sum of WMIPS (Weighted Million Instructions Per Second) that the different VADs will require.

The calculation of the WMIPS is described in (7.1). $Cycles$ equals the number of cycles taken per frame and $Framesize$ equals the size of the frame in seconds.

$$WMIPS = \frac{Cycles \cdot \frac{1}{Framesize}}{10^6}$$  \hspace{1cm} (7.1)

Equation (7.2) describes the calculation of the WMOPS. $Operations$ equals the number of operations taken per frame and $Framesize$ equals the size of the frame in seconds.

$$WMOPS = \frac{Operations \cdot \frac{1}{Framesize}}{10^6}$$  \hspace{1cm} (7.2)

It is important to observe that what has been taken in consideration when calculating these are purely arithmetic operations. The resources being used during loops, if-statements and reading from and writing to memories are not taken into account. Because of this the measured WMOPS and WMIPS are not to be considered as realistic results that will be attained after implementation. These measurements are only to be used for comparison between the different VADs where they should give a pretty exact result.

In this evaluation the measuring of the performance has only been done for the detection of speech and not for comfort noise generation or coding of SID
frames since it is the detection of speech that is the significant part in this thesis. The presented result is an average of all sound frames from all the sound files previously mentioned just as with the bandwidth savings measurement.

The functions measured in this evaluation and the belonging weight numbers can be found in Table 7.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Weight (clock cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sature, add, sub, abs_s, shl, shr, mult, L_mult, negate, extract_h, extract_l, round, L_mac, L_msu, L_macNs, L_msuNs</td>
<td>1</td>
</tr>
<tr>
<td>L_add, L_sub, L_add_c, L_sub_c, L_negate, mult_r, L_shl, L_shr, shr_r, mac_r, msu_r, L_deposit_h, L_deposit_l</td>
<td>2</td>
</tr>
<tr>
<td>L_shr_r, L_abs</td>
<td>3</td>
</tr>
<tr>
<td>L_sat</td>
<td>4</td>
</tr>
<tr>
<td>norm_s</td>
<td>15</td>
</tr>
<tr>
<td>div_s</td>
<td>18</td>
</tr>
<tr>
<td>norm_l</td>
<td>30</td>
</tr>
<tr>
<td>sqrt_Q15</td>
<td>60</td>
</tr>
</tbody>
</table>

*Table 7. Operations used for performance calculations.*

### 7.2.4 Implementation complexity

The complexity has, similar to the sound quality, been measured subjectively and is based on the standard documents and the ANSI-C code published by the respective organizations. The more code used for realization of the VAD the higher the complexity of the VAD is considered to be.

### 7.3 Results

This part covers the results attained from the evaluation. The results are divided by area.

#### 7.3.1 The sound quality

The results from the sound quality evaluation can be found in Table 8. When listening to the sound quality there is no doubt that GSM AMR1 and GSM AMR2 give the best result. They both give very similar results and the quality does not degrade noticeably much when the SNR decreases. Another benefit of these VADs is the fact that the transitions from active speech frames to comfort
noise frames are excellent. It is also very difficult to tell what is comfort noise and real background noise.

The sound quality attained by G.729B is almost as good as with AMR1 and AMR2 although some clippings can be detected when the speech reaches low intensities and the background environments becomes very noisy. In clean background environments it works excellent though. The transitions to and from the active speech frames are perhaps somewhat easier to detect than when using AMR1 and AMR2.

Finally when listening to the result from the Energy VAD it was discovered that this VAD behaves very similar to G.729B when it comes to detecting speech, perhaps slightly more clippings are detected when the background becomes very noisy. What degrades the quality of the synthesized speech when using this VAD is the transitions from active speech to comfort noise frames. The use of Gaussian random white noise proves to give a very poor sound result. The transitions between active and inactive frames become extremely evident which can be really disturbing when the original background noise is something else than white noise which usually is the case.

<table>
<thead>
<tr>
<th>VAD</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729B</td>
<td>The synthesized speech is very good which involves very few clippings, especially in silent background environments. Some clippings can be heard in noisier environments. The comfort noise is integrated very nicely into the active speech frames and it reminds very much of the original background noise.</td>
</tr>
<tr>
<td>The Energy VAD</td>
<td>The detection of speech is very similar to G.729B even though it is only based on measuring the energy. The amount of clippings is somewhat higher than in the case of G.729B and can be discovered in noisy background environments. The integration of comfort noise into the synthesized speech works very poorly. The reason for this is that the comfort noise consists of simple Gaussian white noise. This noise can be very different from the original background noise which leads to big changes in the sound during transitions from active speech to comfort noise.</td>
</tr>
<tr>
<td>GSM AMR1</td>
<td>This VAD gave the best synthesized speech quality. No clippings could be heard unless the level of the background noise intensity was huge. The comfort noise was inserted very nicely giving a good flow in the transitions.</td>
</tr>
<tr>
<td>GSM AMR2</td>
<td>Just as with AMR1 the sound quality here was excellent.</td>
</tr>
</tbody>
</table>
Table 8. The results from the sound quality evaluation.

7.3.2 Bandwidth savings
At the top of Figure 11, a VAD input sound file is plotted. The sound consists of two talk spurts and background with very little noise. The behavior of the four VADs can then be seen when using the sound file as input. It is apparent that all the VADs perform very well in terms of marking the active and inactive areas of the input sound for this situation.

Figure 11. VAD outputs when running a clean background speech file as input.

In Figure 12 the input has been replaced by a sound file consisting of a couple of talk spurts and a much more intensive background noise. The outputs of the VADs now show great differences and it is here evident that the noise level affects the performance very much in a negative way.

By studying the output of the Energy VAD it can be seen that it takes a while for it to find the noise level. It then does a pretty good job determining what is inactivity and not. It however makes some incorrect decisions, mostly in the inactive periods where it marks the noise as speech.

G.729B also seems to need an initial period to detect the noise level. After that it gives a very flickering behavior but still finding a lot of inactive frames.
AMR1 is probably the VAD that stands out the most in this plot. Only for a couple of moments the output goes low and marks some inactivity resulting in a pretty much unaffected synthesized output. The decisions are here incorrect much more often than the Energy VAD and G.729B since it fails to mark the noise as inactivity.

Finally, ARM2 also detects relatively little inactivity though it does not seem to suffer from the initial noise level detection period. It however detects much more inactivity than AMR1 does.

In Table 9, the average bandwidth saved on all frames and sound files by the four VADs is presented. This together with Figure 12 makes it very clear why the sound quality of AMR1 is so good. The difference in saved bandwidth between AMR1 and the other VADs is very significant and since it detects so few inactive frames the synthesized speech reminds very much of the output of a speech coder without a VAD. Since very few alterations of the original sound are done the output sound is basically the same as the input and it is not strange that the output sound becomes good. AMR2 on the other hand manages to detect many inactive frames and also attains a good sound quality.

G.729B and the Energy VAD behave very analogous and they detect most inactive frames which probably is the reason for the higher amount of clippings detected in the synthesized speech in the first test.

Figure 12. VAD output when running a noisy background speech file.
<table>
<thead>
<tr>
<th>VAD</th>
<th>Result (saved bandwidth)</th>
<th>With SID-frames</th>
<th>Without SID frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729B</td>
<td></td>
<td>22.75 %</td>
<td>30.23 %</td>
</tr>
<tr>
<td>The Energy VAD</td>
<td></td>
<td>33.42 %</td>
<td>33.42 %</td>
</tr>
<tr>
<td>AMR1</td>
<td></td>
<td>14.33 %</td>
<td>17.27 %</td>
</tr>
<tr>
<td>AMR2</td>
<td></td>
<td>21.80 %</td>
<td>26.23 %</td>
</tr>
</tbody>
</table>

*Table 9. Bandwidth savings results.*

### 7.3.3 Performance needs

The results from the performance needs evaluation can be seen in Table 10. It now becomes clear why the synthesized speech from AMR2 VAD becomes so good while it also manages to reduce a lot of bandwidth. The reason for it having these two qualities originates from the fact it is extremely demanding in terms of performance. A comparison between this VAD and for example G.729B shows that it requires 25 times the performance counted in WMOPS and 24 times counted in WMIPS more than G.729B. AMR1 being slightly less demanding than AMR2 still gave results that were way higher than G.729B and the Energy VAD.

It can also be seen that the Energy VAD and G.729B gave very similar results being the lowest demanding in terms of performance.

<table>
<thead>
<tr>
<th>VAD</th>
<th>Result</th>
<th>WMOPS in average</th>
<th>WMIPS in average</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729B</td>
<td></td>
<td>0.024</td>
<td>0.032</td>
</tr>
<tr>
<td>The Energy VAD</td>
<td></td>
<td>0.026</td>
<td>0.032</td>
</tr>
<tr>
<td>AMR1</td>
<td></td>
<td>0.167</td>
<td>0.192</td>
</tr>
<tr>
<td>AMR2</td>
<td></td>
<td>0.614</td>
<td>0.785</td>
</tr>
</tbody>
</table>

*Table 10. Performance needs results.*

### 7.3.4 Implementation complexity

The results of the final evaluation part, the implementation complexity, can be seen in Table 11. The results shows that it will be time-consuming and memory demanding to implement the AMR1/2 VADs because of the differences in
structure of the AMR and G.729D speech coders. Almost all the VAD and CNG functions in AMR1/2 would involve writing new code.

Since the VAD and CNG modules of G.729B are developed to be used together with the G.729 speech coders many of the functions used by these modules can be recycled by equivalent functions in the speech coder.

Finally, the Energy VAD would be pretty simple to implement because of the simplicity of its structure.

<table>
<thead>
<tr>
<th>VAD</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729B</td>
<td>The amount of functions needed to be added is relatively high even though some of them might be possible to be recycled. The functions for generating comfort noise and SID frame coding and decoding is the part that would involve the highest number of new functions.</td>
</tr>
<tr>
<td>The Energy VAD</td>
<td>Since this VAD is so simple in its structure it will involve very few functions and therefore a very low complexity.</td>
</tr>
<tr>
<td>AMR1</td>
<td>The technique has a great deal of modules not defined in the G.729D or MELPe speech coders which will lead to a large amount of functions needed to be written. Very few parts of the implemented speech coders in the Tiger platform can be recycled. This would require that much memory needs to be allocated.</td>
</tr>
<tr>
<td>AMR2</td>
<td>As with AMR1 it would be very complex to implement this VAD because of the differences in the AMR speech coder and the Tiger platform speech coders.</td>
</tr>
</tbody>
</table>

Table 11. Implementation complexity results.

7.4 Conclusion

During this evaluation a lot of the results given in [4, 17, 18] were confirmed. It is however essential to remember where the focus is when selecting the optimal VAD solution. AMR1 and AMR2 clearly gave the best results when only looking at the sound quality of the synthesized speech that was attained. However, these two VADs both suffered greatly when it came to the performance needs and the implementation complexity. AMR1 did not appear to fulfil the main objective of this thesis, namely to reduce the bandwidth. It was interesting how clear it was that establishing a good sound quality resulted in very high performance needs, as was the case with AMR2.

The Energy VAD that was studied in this evaluation proved to be the total opposite of the AMR1/2 VADs. It got very good results in terms of performance
needs, bandwidth savings and complexity. However, the sound quality that was produced when using the random Gaussian white noise for comfort noise generation proved to be too poor to be tolerable.

Finally the G.729B VAD that is developed for the G.729 speech coder family seemed to be a good middle course as it got good results in all four evaluation parameters.

A comparison between the G.729B VAD and the Energy VAD showed that the somewhat higher complexity of the G.729B VAD did not appear to be needed since the Energy VAD got just as good results in all tests involving the VAD part. What drags down the overall grade of the Energy VAD is the comfort noise generation.

A conclusion based on the results of this evaluation is that further work will be based on G.729B and the Energy VAD.
8 Descriptions of the chosen VADs

Since the Energy VAD and G.729B were selected as the most suitable VAD candidates for implementation they will be given a more in depth description in this chapter.

8.1 The Energy VAD

This description will depict the basics of this VAD as proposed by Sectra.

The Energy VAD is as mentioned before basically an energy detector. The energy of the input speech is calculated using (8.1).

\[
E = \sqrt{\frac{x^T \cdot x}{N}}
\]  

(8.1)

The vectors \(x^T\) and \(x\) consist of the input sample from one frame while \(N\) is the number of samples in the current frame. The minimum and maximum energy, \(Min\) and \(Max\) levels detected in the sequence of frames are then stored and a threshold value based on these are calculated with (8.2).

\[
T = (K_1 \cdot Max) + (K_2 \cdot Min)
\]  

(8.2)

\(K_1\) and \(K_2\) are constants, which are used to interpolate the threshold value \(T\) to an optimal performance. If the current frame’s energy is less than the threshold value it is marked as non-active. However this does not mean that the transmission immediately will be halted. There is a hangover period that should consist of more than four non-active frames before the transmission is to be stopped. If the energy increases above the threshold value the communication will resume.

Since low energy anomalies can occur there is a prevention needed for this. The parameter \(Min\) is slowly increased for each frame and this is defined by (8.3).

\[
Min(t) = Min(t-1) \cdot \Delta(t)
\]  

(8.3)

The parameter \(\Delta(t)\) is also increased for each frame and this is defined by (8.4).

\[
\Delta(t) = \Delta(t-1) \cdot 1.0001
\]  

(8.4)

It is from this recommendation easy to realize that the algorithm is very independent and easily can be integrated into most speech coders.
8.2 G.729B

This description covers the basic parts of the ITU-T recommendation in [10]. The whole standard is very complex and is therefore not covered in this thesis in its entirety.

The algorithm of the G.729B VAD starts with a parameter extraction where the parameters are the line spectral frequencies, the full band energy, the low band energy and the zero-crossing rate. A closer description of these extractions will follow in 8.3.

The next step is dependent on whether the current frame number is lower than 31. The first 30 frames are considered as an initialization phase. Background noise characteristics are initialized and updated during this phase. The characteristics are running averages of the frame energy, the LSFs and the zero crossing rate. The first 30 frames equals 0.3 seconds and this should be a period where no voice is present since it is unlikely that a person reacts this fast when answering a call. When making a call it is most likely that one remains silent until the receiver answers by saying his name or for example “Hello”. During the initialization phase it is only the full band energy that constitutes the voice activity detection. If this parameter is higher than 15 dB the current frame is marked as active voice, otherwise it is marked as noise (not transmitted).

When the 31st frame appears the running averages of the full- and low-band energy, which are two of the parameters of the background noise characteristics, are initialized.

During the next phase differential parameters calculations, which will be described more in depth in 8.4, are performed. The full band energy is again checked and if the value is lower than 15 dB the frame is marked as noise, otherwise a multi-boundary voice activity detection decision is made.

In the next step a hangover mechanism called “voice activity decision smoothing” in the recommendation is performed. The hangover mechanism then marks the current frame as voice or noise depending on the energy of the current frame and previous frame decisions. During frames marked as noise the running averages of the background noise characteristics are updated.

8.3 Parameter extraction

The parameters extractions are briefly explained here. Some of the terms used originate from functions in the G.729 speech coder. The formulas in this section are reproduced with the kind permission of ITU-T.
8.3.1 Full band energy
The full band energy $E_f$ is the logarithm of the normalized first autocorrelation coefficient $R(0)$, which is attained from the autocorrelation performed by the speech encoder, and is defined by equation (8.5).

$$E_f = 10 \cdot \log_{10} \left[ \frac{1}{N} \cdot R(0) \right]$$

(8.5)

$N$ is here equal to 240 and originates from the LPC window size in speech samples which equals 3 frames.

8.3.2 Low band energy
The low band energy $E_t$ is measured on the band from 0 to 1 Hz and is defined by equation (8.6).

$$E_t = 10 \cdot \log_{10} \left[ \frac{1}{N} \cdot h^T \cdot R \cdot h \right]$$

(8.6)

$N$ is here the same as with the case of the full band energy while $h$ is the impulse response of an FIR filter with the cutoff frequency at 1 Hz. $R$ is a Toeplitz matrix, which means a matrix that has constant values along the negative-sloping diagonals. The matrix consists of the autocorrelation coefficients on each diagonal.

8.3.3 Zero crossing rate
The normalized zero crossing (ZC) rate is for each frame calculated by the equation in (8.7):

$$ZC = \frac{1}{2M} \cdot \sum_{i=0}^{M-1} \| \text{sgn}[x(i)] - \text{sgn}[x(i-1)] \|$$

(8.7)

The input signal is here represented by $x(i)$ and $M$ is the current frame size which is always 80.

8.4 Differential parameter calculations
In this section the differential parameter calculations are described more thorough. These formulas are also reproduced with the kind permission of ITU-T.

8.4.1 The spectral distortion
The spectral distortion, denoted $\Delta S$, is generated as the sum of squares of the difference between the LSF coefficients of the current frame, denoted
simply $LSF$, and the LSF coefficients of the running averages of the background noise, denoted $\bar{LSF}$. (8.8) is the equation for this.

$$\Delta S = \sum_{i=1}^{10} (LSF_i - \bar{LSF})^2$$

(8.8)

### 8.4.2 The full band energy difference

The measurement of the full band energy difference is generated from the energy of the current frame, denoted $E_f$, and the running average of the background noise full-band energy, denoted $\bar{E}_f$. (8.9) is the equation for this.

$$\Delta E_f = \bar{E}_f - E_f$$

(8.9)

### 8.4.3 The low band energy difference

The measurement of the low band energy is generated from the current frame, denoted $E_i$, and the running average of the background noise low band energy, denoted $\bar{E}_i$. (8.10) is the equation for this.

$$\Delta E_i = \bar{E}_i - E_i$$

(8.10)

### 8.4.4 The zero crossing rate difference

The measurement of the zero crossing rate is generated from the current frame, denoted $Z$, and the running average of the background noise zero crossing rate, denoted $\bar{Z}$. (8.11) is the equation for this.

$$\Delta Z = \bar{Z} - Z$$

(8.11)
9 Simulations with the selected VADs

Since G.729B and the Energy VAD were selected as candidates for the implementation several tests were performed with these. The reason for this was to find the optimal VAD function. Documentation of these tests should also be valuable for possible future enhancements.

The tests were carried out with the same sound files as mentioned before and they were conducted with the G.729D and the MELPe speech coders. A description of the tests and the results will be discussed in this chapter.

9.1 The reference codes

The tests with the G.729 speech coder were performed with the reference code of G.729F, the version of G.729 that uses DTX and a bit rate of 6.4 kbit/s. The reference code consists of a coder and a decoder. The functionality of the reference code can be seen in Figure 13.

![Diagram](image)

**Figure 13.** The flow of the reference code.

A sound file consisting of raw pulse code modulated data (PCM data) serves as input to the encoder. The encoder generates an output file consisting of encoded parameters that act as input to the decoder. The decoder synthesizes the speech from the encoded parameters to an output file with PCM data.

The output file from the encoder consists of data representing three kinds of frames. The frame that represents the active speech frame can be seen in Figure 14. This frame consists of synchronization, the length of the frame and the encoded data.

<table>
<thead>
<tr>
<th>Sync</th>
<th>Length</th>
<th>Encoded active speech parameters</th>
</tr>
</thead>
</table>

**Figure 14.** The active speech frame structure.

When noise is detected a SID-frame is generated and the structure, as can be seen in Figure 15, reminds very much of the active speech frame. The only
difference is that the part representing the data is much smaller. Whenever this frame is encountered by the decoder, comfort noise based on the parameters in the data field of the SID frame is generated.

Figure 15. The SID frame structure.

After a SID frame has arrived, the transmission should be terminated. However, to simulate this behavior, a third frame type has been designed, see Figure 16. This frame is an empty frame without the data field that exists in the active speech and SID frames. This frame tells the decoder to keep on generating comfort noise based on the previous SID frame.

Figure 16. The empty frame structure.

The MELPe reference code works in a very similar way with a coder and a decoder. There is however no DTX functionality implemented and therefore only active speech frames exist.

9.2 Tests with the G.729D speech coder

The following test cases where performed with the G.729D speech coder. The tests where performed with the reference code for G.729F or a modification of it and all the 19 sound files in the evaluation in Chapter 7 were used as input.

9.2.1 G.729D Test case 1 - G.729B without SID frames

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>G.729B (modified)</td>
</tr>
<tr>
<td>CNG:</td>
<td>G.729B CNG</td>
</tr>
<tr>
<td>Objective:</td>
<td>No SID frames.</td>
</tr>
<tr>
<td>Description:</td>
<td>Due to synchronization in protocols of the higher layers in the Tigers communication it was undesirable to transmit sudden single frames in the middle of a period of inactivity. This would lead to problems when using the SID frames since they can appear in the middle of inactivity updating the background noise. The goal of this test was to see how well G.729B behaved when not using the SID frames. At the transmitter’s side, the SID frames where replaced by empty frames and the receiving side was forced to consider</td>
</tr>
<tr>
<td>Result:</td>
<td>these as lost frames.</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
</tr>
<tr>
<td></td>
<td>The result was a clear degeneration of the sound at the receiver’s side. The generation of comfort noise did not work as well as it did with the use of the SID frames as large differences between active and inactive frames were detected. The reason for this was most likely because of incorrect noise amplitudes. The quality of the decoded sound was experienced to be too poor to be acceptable.</td>
</tr>
</tbody>
</table>

**Table 12.** G.729B without SID frames.

**9.2.2 G.729D Test case 2 - G.729B with Bad Frame Indication**

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>G.729B (modified)</td>
</tr>
<tr>
<td>CNG:</td>
<td>G.729B CNG and BFI</td>
</tr>
<tr>
<td>Objective:</td>
<td>No SID frames.</td>
</tr>
<tr>
<td>Description:</td>
<td>In the G.729 standard there is a procedure for dealing with frames that are lost during transmission. Lost frames or errors detected in frames are marked as bad frames by a so-called “bad frame indicator” (BFI), at the receiver’s side. When receiving such a frame the decoder will try to predict the contents of the lost frame and reconstruct it. The SID frames were again removed and the receiver was forced to use the BFI-procedure for all inactive frames.</td>
</tr>
<tr>
<td>Result:</td>
<td>The result was not a success. The amplitude of the reconstructed frames eventually went to zero which lead to a transient as speech eventually appeared again. The quality of the decoded sound was yet again experienced to be too poor to be acceptable.</td>
</tr>
</tbody>
</table>

**Table 13.** G.729B with Bad Frame Indication.

**9.2.3 G.729D Test case 3 - G.729B with only initiating SID frames**

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>G.729B (modified)</td>
</tr>
<tr>
<td>CNG:</td>
<td>G.729B CNG</td>
</tr>
<tr>
<td>Objective:</td>
<td>No SID frames.</td>
</tr>
<tr>
<td>Description:</td>
<td>Since none of the previous tests where the SID frames were totally removed gave successful results it seemed likely that the use of SID frames was needed in order to establish a decent</td>
</tr>
</tbody>
</table>
decoded sound. However, one solution to escape the problem with single frames in periods of inactivity would be to keep only the SID frames that are transmitted immediately after the last speech frame.

Result: The result here was just a slight degradation in the decoded sound at the receiver’s side compared to the case were all SID frames are used. The degradation was however so small that this method was experienced to be totally acceptable.

Table 14. G.729B with only initiating SID frames.

9.2.4 G.729D Test case 4 - G.729B using the Energy VAD

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>G.729B CNG</td>
</tr>
<tr>
<td>Objective:</td>
<td>To test the Energy VAD.</td>
</tr>
<tr>
<td>Description:</td>
<td>To be able to compare the performance of G.729B and the Energy VAD a test case was performed where only the VAD of G.729B was replaced. Other processes such as the parameter extraction for the SID frames, coding of the SID frames and comfort noise generation at the receiver’s side were kept intact.</td>
</tr>
<tr>
<td>Result:</td>
<td>The result of this test was very similar to the original version of G.729B with perhaps slightly more clippings. In this test parameters, such as constants in the algorithm, were attained and used for the rest of the simulations involving the Energy VAD.</td>
</tr>
</tbody>
</table>

Table 15. G.729B using the Energy VAD.

9.2.5 G.729D Test case 5 - The Energy VAD with no CNG

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Silence</td>
</tr>
<tr>
<td>Objective:</td>
<td>To test the importance of CNG.</td>
</tr>
<tr>
<td>Description:</td>
<td>Since the generation of comfort noise in G.729B was discovered to be a very complex process it would be interesting to hear how it would sound if the inactive frames were replaced by mere silence from the decoder.</td>
</tr>
<tr>
<td>Result:</td>
<td>Tests with only silence proved to result in a very poor sound quality unless the sound of the background was just silence. When switching from an active frame where background noise...</td>
</tr>
</tbody>
</table>
consisted of other than pure silence to an inactive frame the changes in the sound appear very abruptly.
This method leads to a behavior where the change from silence to voice with background noise is too obvious to be recommended.

Table 16. The Energy VAD with no CNG.

9.2.6 G.729D Test case 6 - The Energy VAD with static white noise CNG

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Random Gaussian white noise with static amplitude level</td>
</tr>
<tr>
<td>Objective:</td>
<td>To test random Gaussian white noise CNG.</td>
</tr>
<tr>
<td>Description:</td>
<td>Since silence instead of comfort noise proved to be a bad idea the next choice was to use random Gaussian white noise as a CNG. The amplitude of the noise was set to always range between the same amplitudes for all sound files.</td>
</tr>
<tr>
<td>Result:</td>
<td>The conclusion here is pretty much the same as in the case with the silence. The method worked reasonably well when the original background noise was white noise at the same intensity as the background noise at the transmitter's side. But as long as the background noise consisted of other sounds than white noise, which is the usual case, the switching between inactive and active frames became too evident.</td>
</tr>
</tbody>
</table>

Table 17. The Energy VAD with static white noise CNG.

9.2.7 G.729D Test case 7 - The Energy VAD with adaptive white noise CNG

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>G.729D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Random Gaussian white noise with adaptive amplitude level</td>
</tr>
<tr>
<td>Objective:</td>
<td>To test random Gaussian white noise CNG.</td>
</tr>
<tr>
<td>Description:</td>
<td>Since the previous test was not successful, a test where the amplitude ranges were based on the energy level of the latest received active speech frame was conducted. This would give a more adaptive behavior of the generated comfort noise.</td>
</tr>
<tr>
<td>Result:</td>
<td>This method gave a better result than the previous two methods involving the Energy VAD. In sequences of speech where the background consisted of silence and white noise it worked</td>
</tr>
</tbody>
</table>
pretty well but if it instead was for example babble or car noise
the result was just as bad as the previous two techniques.
The outcome is that this method is not to be recommended
either.

Table 18. The Energy VAD with adaptive white noise CNG.

9.3 Tests with the MELPe speech coder
The following test cases were performed with the reference code for the MELPe
speech coder or a modification of it.

9.3.1 MELPe Test case 1 - The Energy VAD with no CNG and the
NPP off

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>MELPe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Silence</td>
</tr>
<tr>
<td>NPP:</td>
<td>Off</td>
</tr>
<tr>
<td>Objective:</td>
<td>To add DTX functionality.</td>
</tr>
</tbody>
</table>
| Description:  | Since there is no standardized VAD or CNG for the MELPe
speech coder it seemed like a good idea to integrate the Energy
VAD into this speech coder.
At this stage the CNG at the receiver’s side only generated
silence and the NPP at the transmitter’s side was turned off. |
| Result:       | The results were very similar to the results obtained in 9.2.5.
This means that variations from active to inactive frames
resulted in too large differences to be acceptable. |

Table 19. The Energy VAD with no CNG and the NPP off.

9.3.2 MELPe Test case 2 – The Energy VAD with no CNG and the
NPP on

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>MELPe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Silence</td>
</tr>
<tr>
<td>NPP:</td>
<td>On</td>
</tr>
<tr>
<td>Objective:</td>
<td>To add DTX functionality.</td>
</tr>
</tbody>
</table>
| Description:  | This test was exactly the same as 9.3.1 with the only difference
that the NPP was turned on. |
The result was clearly better than the previous test case since the NPP suppresses much of the background noise resulting in a background sound much more alike silence. However, the transitions from active speech frames to CNG frames still became too clear to be acceptable.

Table 20. The Energy VAD with no CNG and the NPP on.

9.3.3 MELPe Test case 3 - The Energy VAD utilizing the frame erasure procedure with the NPP off

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>MELPe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Frame erasures</td>
</tr>
<tr>
<td>NPP:</td>
<td>Off</td>
</tr>
<tr>
<td>Objective:</td>
<td>To add DTX functionality.</td>
</tr>
<tr>
<td>Description:</td>
<td>In MELPe there is a similar method for lost or corrupted frames as the BFI in G.729, this is denoted “frame erasures”. Lost or corrupted frames are replaced by a reconstructed frame based on a correlation of earlier frames. In this test the receiver was forced to use the frame erasure procedure for all empty frames.</td>
</tr>
<tr>
<td>Result:</td>
<td>The result proved to be very different from the case with BFI in G.729. Instead of becoming only silence very fast the synthesized sound was very much alike the last received active voice frame without giving a very stationary sound.</td>
</tr>
</tbody>
</table>

Table 21. The Energy VAD utilizing the frame erasure procedure with the NPP off.

9.3.4 MELPe Test case 4 - The Energy VAD utilizing the frame erasure procedure with the NPP on

<table>
<thead>
<tr>
<th>Speech coder:</th>
<th>MELPe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD:</td>
<td>The Energy VAD</td>
</tr>
<tr>
<td>CNG:</td>
<td>Frame erasures</td>
</tr>
<tr>
<td>NPP:</td>
<td>On</td>
</tr>
<tr>
<td>Objective:</td>
<td>To add DTX functionality.</td>
</tr>
<tr>
<td>Description:</td>
<td>In this test the method of the previous test case was again used but this time with the NPP on. Since the NPP suppresses a lot of the background noise what is left is basically only the speech, not considering extremely noisy background</td>
</tr>
</tbody>
</table>
environments.

| Result: | The background noise that can be heard after filtering is usually only occasional segments of the original background noise. This fact makes it very ideal to use VAD and frame erasures. The difference between using MELPe with NPP and VAD is very little compared to the same scenario but without the VAD. Since this technique includes very little changes of the original MELPe speech coder it is a technique to be recommended. |

Table 22. The Energy VAD utilizing the frame erasure procedure with the NPP on.

9.4 Conclusion

To conclude these tests, there was one variant for the G.729D speech coder that was found where the individual SID frames were removed and the output was acceptable. This variant was obtained from 9.2.3. The SID frames immediately following an active speech frame that always begin the periods of inactivity were however left untouched.

It was also discovered in these tests that a substitution of the VAD function in G.729B with the Energy VAD produced extremely similar results when the others parts of G.729B were left unaffected.

To conclude the G.729 part of these tests either the Energy VAD or the G.729B VAD can be used for voice activity detection. For comfort noise generation it is not recommended to use random Gaussian white noise though. Instead the CNG of G.729B or something similar should be used.

For MELPe it was found that when using a good VAD and a good hangover scheme the frame erasure process should be possible to use for reproducing the inactive frames not transmitted in a comfort noise generation behavior. However, the hangover scheme must make sure that there is no speech being clipped at the end of speech sequences since this method is based on the fact that the last received frame is nothing but noise. More simulations might on the other hand be needed before recommending this technique fully.
10 The implementation
This chapter describes the chosen implementation, how it was done and some of the problems that were encountered.

10.1 The development process
This section describes parts of the implementation and how it was tested.

10.1.1 The chosen VAD
Based on the evaluation and the simulations described earlier in this thesis, it was decided to choose the Energy VAD for implementation. The reason for this was that the Energy VAD clearly reduced the required bandwidth which was the main goal of this thesis work. The quality of the sound in simulations also proved to be sufficient together with the fact that this would involve very little code generation in the Tiger.

A flowchart of the VAD that was implemented can be found in Appendix A.

10.1.2 The chosen CNG
Since the random Gaussian white noise proved to be insufficient the CNG of G.729B was studied more thorough. Initially, the plan was to use the results obtained in 9.2.3. This would involve using the SID frames initializing an inactive period with comfort noise generation. It was however discovered that a lot of new code would be needed to generate the SID frames, this would most likely result in a poor performance and a lot of memory use.

Since the contents of the SID frames only consist of background noise LSPs and a gain factor it should not be too difficult to generate something equal with functions already existing in the code. An idea would for example be to use the gain and the LSPs of the last received active speech frame. The hangover scheme of the Energy VAD should result in a scenario where the last transmitted frame should have very similar contents (similar gain and LSPs) to the SID frame. The reason for this is that the hangover scheme forces five consecutive frames to be marked as noise before turning off the transmission. The last of these five frames should only be noise.

Because of the hangover scheme, including five consecutive frames marked as noise before halting the transmission, it should mean that the gain and LSPs of the last active, but still marked as noise, speech frame should differ very little from the contents of the SID frame.

The reference code was modified for this behavior and simulations where done and it was discovered that the result was surprisingly good. Very little
differences between these results and the results of the test case in 9.2.3 could be heard.

The CNG that was chosen for implementation, see the flowchart in appendix B, basically just adds random white noise to the existing excitation vector generated by previous frames. The synthesis filter that is used originates from the latest received active speech frame and is as a matter of fact exactly the same. The use of this method means that the transition from an active speech frame to the frame consisting of comfort noise becomes very smooth. The noise will also remind very much of the background noise at the transmitter’s side.

A flowchart of the implemented CNG can be found in appendix B

10.1.3 Object code recycling

Since many of the functions that build up the CNG, for example the synthesis filter and post filter functions, already existed in the object code a lot of time was spent on integrating these into the code. Luckily many of the functions were already prepared for this, meaning that some registers just needed to get the correct input values. Running the functions then produced the proper output values into the same or other registers.

Originally there were only encoder and decoder initialization routines and also the main encoder and decoder routines that were supposed to be used. The functions that were needed to be used for the CNG were however undocumented since they were not meant to be used individually. This resulted in a lot of testing and looking for proper data in the memories to understand where to put the input data and where to fetch the output data. Since the object code was not meant to be used this way some registers also had incorrect values. These registers were originally given the proper values in the end of the encoder and decoder routines; this also required a lot of troubleshooting.

By recycling many parts of the object code a lot of code duplication could be avoided. The code used was also much optimized giving these functions a good performance.

10.1.4 The implementation verification process

The development of the addition of a VAD and a CNG, resulting in a DTX behavior to the Tiger platform, was first done merely in the reference code. When moving on to the implementation in the actual platform it was very important to know that the output of the reference encoder and decoder would give exactly the same output as the implemented encoder and decoder.

To verify the correctness of the implementation some of the sound files provided with the G.729F speech coder were used. These files were then used as input to
the reference code which was modified to produce some output test data. The encoder was modified to produce the four files in Table 23.

<table>
<thead>
<tr>
<th>Description</th>
<th>Contents</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input data</td>
<td>Input data converted from binary format to ASCII</td>
<td>Input to implemented encoder</td>
</tr>
<tr>
<td>VAD parameters</td>
<td>The current frame energy, threshold energy and voice activity detection selection.</td>
<td>Comparison with equivalent parameters from implementation.</td>
</tr>
<tr>
<td>Frame type</td>
<td>A marking of the current frame, 1 = voice, 0 = noise.</td>
<td>Input to the implemented decoder for simulation since external signals marking this were missing.</td>
</tr>
<tr>
<td>Encoded parameters</td>
<td>The normal parameters generated by the encoder for active speech frames.</td>
<td>Comparison with equivalent parameters generated by the implemented encoder.</td>
</tr>
</tbody>
</table>

*Table 23. Output files generated by the reference code encoder.*

The reference code decoder was also modified to generate an output file, see Table 24.

<table>
<thead>
<tr>
<th>Description</th>
<th>Contents</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesized output data</td>
<td>The synthesized output data, consisting of a composition of active speech and comfort noise, generated by the reference decoder.</td>
<td>Comparison with equivalent data generated by the implemented decoder and CNG.</td>
</tr>
</tbody>
</table>

*Table 24. Output files generated by the reference code decoder.*

The generation of the test files can be seen in Figure 17 and 18.
Figure 17. The implementation verification process.

When the test files had been generated they were used as input to the implemented encoder and decoder. This can be seen in Figure 18.
Figure 18. *The implementation verification process.*

When the simulations of the implementation gave resulted in a correct behavior for all the three sound files the implementation was considered to be done.

### 10.2 Evaluation of the implementation

This section describes how the implementation was evaluated.

#### 10.2.1 Performance numbers

The first thing to test before actually listening to the result of the implementation was to measure the performance needed.

In Table 23 the performance numbers of the different modules can be found. It would be very time consuming to calculate an average for each of these so instead the numbers will only give a hint of the actual value since some of them varies a great deal depending on the contents of the current speech frame.

In the tables the numbers are presented with optimization of the code, which is a setting in the development tool, both on and off. The goal is however to always have the optimization setting turned on.
As can be seen in the table the VAD uses very little of the performance while the CNG requires very much from the DSP. Further studies of this showed that a big part of this was due to the random number generation used for the white noise addition to the excitation. These numbers were however acceptable to be used.

10.2.2 Memory requirements

The next important thing was to check the amount of memory required by the added functions. In Table 26 it can be seen that the VAD function requires very little memory. The CNG on the other hand required pretty much. Still these functions were easily fitted into the existing memories.

<table>
<thead>
<tr>
<th>Function</th>
<th>Program memory</th>
<th>Data memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimization off</td>
<td>Optimization on</td>
</tr>
<tr>
<td>Voice activity detection</td>
<td>1.2 Kbytes</td>
<td>0.8 Kbytes</td>
</tr>
<tr>
<td>Comfort noise generation</td>
<td>7.7 Kbytes</td>
<td>4.9 Kbytes</td>
</tr>
</tbody>
</table>

Table 26. Memory requirements for the DTX modules.

10.2.3 Bandwidth savings

Since this criterion depends on so many factors, such as background noise and the amount of speech activity, it was not tested very thoroughly. However, a delay of 0.5 minutes was hard coded into the platform. A normal conversation was then performed in an office environment and the delay quickly disappeared.
10.2.4 Sound quality testing
The sound quality was also tested in an office environment conversation. The comfort noise adapted nicely to the real background noise and no clippings or rough transitions when moving from active speech frame to inactivity could be heard.

10.3 Problems encountered
This section describes some of the problems encountered during the implementation process.

10.3.1 Object converter
The first step when implementing the VAD was to get the speech coder G.729D up and running. From Sectra there was however a wish to use a new development tool called. As previously mentioned the speech coder, G.729D, implemented in the Tiger platform consists of object code. However, the format of this object code was not the same as the format handled by the new development tool. It was therefore essential to convert the object code to the structure needed by the development tool. Fortunately it was discovered that an object converter tool was provided with the tool.

Unfortunately it was discovered that the speech coder had lots of flaws resulting in an incorrect behavior after the conversion. After much debugging it was detected that these errors were results of the conversion meaning that the converter worked in an incorrect way. The errors were reported to the developers of the development tool, who after some time returned new versions of the converter. This process went on for a while and after three new versions of the converter no more errors could be detected and the speech coder could be run in the new development tool.

To conclude this problem it can be said that the creators of the development tool Devices were very helpful though it was extremely time consuming to find the actual origin of the errors and the explain them to the company.

10.3.2 Interrupt routine in C-code
When the DSP receives new samples, an interrupt is generated. The routine for this was initially written in Assembler since it needed to be fast and give the best possible performance.

It seemed interesting though to test the performance when having the routine written in C-code instead. This procedure was tested but it was soon discovered that it was not possible. The reason for this was due to one of the registers of the DSP, which is always supposed to be set to –1 when using the C-Runtime. According to the manuals of the development tool, the register is used for
storing data on to the stack for every interrupt. In the object code for the speech coder this register is however used as a temporary storage register, meaning it could have various other values. The result is that when an interrupt occurs when running processes in the object code this register can have an incorrect value when storing away data onto the stack. This data will then be lost when returning from the interrupt routine.
11 Summary
In this chapter possible further work that would be recommended, as well as conclusions from this thesis works, is discussed.

11.1 Continued work
In this section some possible future improvements and additions will be covered.

11.1.1 A more thorough evaluation
The implementation has so far only been tested in an office environment with almost ideal background conditions. Since the users of the Tiger platform operate in various other environments the implementation should of course be tested in these environments. If it is then discovered that it behaves very poor there could be a need to add more voice activity detection techniques such as the ones discussed in 4.5.

11.1.2 Performance optimization
Since the performance of the CNG measured in WMIPS was relatively poor some work should be done on improving this. For example the random number generation could be replaced by something more hardware-based.

11.1.3 MELPe
It was mentioned in the requirements of this thesis that an implementation for the MELPe speech coder also was desired. If the implementation of the VAD function for the G.729D speech coder proves to be a success work should also be done for this speech coder.

A recommendation would be to use the VAD function that was added to G.729D for MELPe as well. For comfort noise generation the proposed technique for G.729D would not be suitable for MELPE. The reason for this is the fact that it is very integrated with the speech coder. Instead, the frame information of Table 27 is recommended.

<table>
<thead>
<tr>
<th>MELPe Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch &amp; Global UV Decisions</td>
<td>msvq[0] Average of earlier frames</td>
</tr>
<tr>
<td></td>
<td>msvq[1] 0</td>
</tr>
<tr>
<td></td>
<td>msvq[2] 0</td>
</tr>
<tr>
<td></td>
<td>msvq[3] 0</td>
</tr>
<tr>
<td>Line Spectral Frequencies</td>
<td>gain[0] Average of earlier frames</td>
</tr>
<tr>
<td>Gains</td>
<td>gain[0] Average of earlier frames</td>
</tr>
<tr>
<td></td>
<td>gain[1]</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Band pass Voicing</td>
<td>0</td>
</tr>
<tr>
<td>Fourier Magnitudes</td>
<td>0</td>
</tr>
<tr>
<td>Jitter</td>
<td>0</td>
</tr>
<tr>
<td>Synchronization</td>
<td>Continue alternations</td>
</tr>
<tr>
<td>Error Protections</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 27. Frame for comfort noise generation in MELPe.

The terms msvq[0] and gain[0] should be an average of values from earlier active speech frames which should be included in the hangover scheme. The rest of the values, except for the synchronization should then be zeroed.

11.2 Conclusions

In this thesis work, voice activity detection theory was first examined to get a good understanding for the subject. Several VADs were then investigated in order to find the most suitable, existing one, for the Tiger platform. Four of them were then chosen for further evaluation were important areas such as sound quality, bandwidth savings, performance and implementation complexity were studied. In order to see if the results from this evaluation were reasonable, they were also compared with the results from an externally performed evaluation. The parameters measured in the evaluation of this thesis work were however chosen with the Tiger platform as the target while the external evaluation primarily was based on the correctness of the VADs. The overlapping areas from these two evaluations gave the same results though. By weighing together all the parameters from the evaluation, G.729B and the Energy VAD were then selected as the most suitable VADs for implementation. During this evaluation, several techniques were studied and measured resulting in a hopefully, objective selection of the most suitable VAD.

After performing simulations it was later discovered that the Energy VAD would be the most suitable VAD. The reason for this was the simplicity of the algorithm and the sound quality that was attained, which was very similar to the one attained with G.729B which was experienced to be more complex.

Since the random Gaussian white noise variant of CNG originally used with the Energy VAD proved to be unsuccessful, because of the bad sound quality, the CNG of G.729B was considered to be the best alternative. The algorithm uses many of the functions already existing in the speech coder G.729D which makes it possible to recycle a lot of the code, reducing the memory requirements. The result was a very efficiently obtained comfort noise, sounding very similar to the
original background noise, since little performance and memory from the platform was used.

The selection of VAD and CNG that was done in this thesis work was performed with the special criteria of the Tiger platform in mind. This resulted in an implementation that should be the most optimal under the circumstances that existed during this thesis work.

After the completion of the implementation, it was integrated into the Tiger platform and it was tested in an office environment. In this special environment it performed very well, fulfilling most important objectives of this thesis works such as a reduction in transferred data as well as maintaining a high sound quality. Since the implementation only has gone through very basic tests it is not easy to comment on the quality of it in general. It might be discovered further on that it is not the best suited implementation for all environments but it should be relatively easy to improve the existing implementation to meet harder requirements. In other words the existing implementation consisting of a VAD and a CNG should be a very good starting point for further work.

The CNG proved to be very demanding which means that more time perhaps should have been spent on comparing different techniques for generating comfort noise. This was originally not the main objective of this thesis work since it was the voice activity detection part that was considered to be the most complex part. It has now been discovered that the comfort noise generation can be just as complex.

To summarize this, the thesis work that was done at Sectra has hopefully resulted in a good starting point for further work in this area.
References


[12] ETSI, ETSI EN 300 730 Digital cellular telecommunications system (Phase 2+); Voice Activity Detector (VAD) for Enhanced Full Rate (EFR) speech traffic channels (GSM 06.82 version 8.0.0 Release 1999), 1999.

[13] ETSI , ETSI EN 300 965 Digital cellular telecommunications system (Phase 2+); Full rate speech; Voice Activity Detector (VAD) for full rate speech traffic channels (GSM 06.32 version 8.0.0 Release 1999), 2000.

[14] ETSI , ETSI EN 300 973 Digital cellular telecommunications system (Phase 2+); Half rate speech; Voice Activity Detector (VAD) for half rate speech traffic channels (GSM 06.42 version 8.0.0 Release 1999), 2000.

[16] ETSI, *ETSI EN 301 708 Digital cellular telecommunications system (Phase 2+); Voice Activity Detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels. General Description (GSM 06.94 version 7.1.1 Release 1998)*, 1999.


Appendix A: Flowchart of the implemented VAD function part 1

Incoming samples.

Calculate the energy of the frame.

First frame?

Yes

Max = current energy.

No

Initialize Min.

Current energy > Max?

Yes

Max = current energy.

No

Current energy < Min?

Yes

Energy = 0?

No

Min = Current energy.

Reset Min to initial value.

Increase Δ(t).

Reset Δ(t).

No

To be continued
Appendix A: Flowchart of the implemented VAD function part 2

1. Calculate the threshold energy.
2. Current energy > Threshold energy?
   - Yes
     - Mark the current frame as voice.
   - No
     - Mark the current frame as noise.
     - Inactive frames counter = Hangover threshold?
       - Yes
         - Mark the current frame as voice.
       - No
         - Mark the current frame as noise.
         - Increase the inactive frame counter.
         - Increase Min.
         - Return the frame marker.
         - Reset the frame counter.
Appendix B: Flowchart of the implemented comfort noise generator

Generate the new excitation vector from the excitation of the previous frame.

Generate the new synthesis buffer from the synthesis of the previous frame.

Copy the unvoiced gain of the previous frame.

Copy the LSPs from the previous frame.

Add a random Gaussian excitation to the excitation vector.

Interpolate the LSPs.

Do the synthesis filtering, \(1/A(z)\), for each sub frame of 40 samples.

Store the current excitation vector for the next frame.

Post filter the synthesized signal in two steps.

Store parts of the synthesized buffer for the next frame.