Master's Thesis

Quantification of audio quality loss after wireless transfer

Frida Hedlund
Ylva Jonasson

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, June 2015.

Master's Thesis

Quantification of audio quality loss after wireless transfer

By

Frida Hedlund and Ylva Jonasson

ael10fhe@student.lu.se ael10yjo@student.lu.se

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

The report describes a quality measurement for audio, both the theoretical background and implementation. It begins by describing the unlicensed methods the implementation is based on, Segmental SNR, Frequency Weighted Segmental SNR, Log-Likelihood Ratio, Cepstral Distance and Weighted Slope Spectral distance, and the commercial methods used as reference, PEAQ and PESQ. It also mentions the problems present in wireless transfer and the concept of sound quality assessment.

It concludes by describing the suggested analysis method and implemented software together with the results when compared to PEAQ and PESQ.

Foreword

This thesis was made during the spring of 2015 at Cybercom Group in Malmö by Frida Hedlund and Ylva Jonasson. The test center at Cybercom needed an objective way of testing sound quality in, primarily, Bluetooth and phone links without investing in licenced methods. This thesis tries to solve that problem by combining unlicenced signal evaluation methods. Most of the work was made at the Cybercom office, but some testing was carried out in the antenna laboratory at LTH in Lund.

The authors would like to thank their examiner, Nedelko Grbic, their advisor from LTH, Mikael Swartling, and their advisor from Cybercom, Joakim Rydh.

Table of contents

CHAPTER 1

1 Introduction

In the Cybercom test center, a wide variety of consumer electronic devices that transmit sound over a radio link are tested. Today, the tests focus on protocol adherence, and not on the actual quality of the sound being transmitted.

The only tests Cybercom perform today of sound quality is the tester subjectively determining if the sound is “good enough”. This is not a result that can be quantified and compared between releases of the product, and when several testers are involved with different opinions it can be very uncertain what the result is. The company therefore wants a computer based program that can make an objective evaluation instead.

The purpose and goal of this master thesis is to research available methods for objectively determining sound quality, and to create an easy to use, unlicensed method that can be used for testing cell phones and Bluetooth products with an acceptable accuracy. This method is only meant for in-house and development support, and should not be sold as a qualifying service for a products sound quality. That should be done by a licensed sound testing lab. This report begins with a theoretical background, which presents the concept of sound quality, problems with wireless links, two licensed and five unlicensed methods for objective sound quality testing and how to do subjective tests of sound quality. It then continues with a chapter on how the research into the unlicensed method was conducted and then the suggested software packet is described. The next chapter presents the results of the tests made during the creation of the software packet and this is followed by an analysis of the results and a conclusion.

CHAPTER 2

2 Theoretical base

2.1 What is sound quality?

It is important to be able to determine the quality of the sound in a product or transmission, for instance when developing or improving products, or for companies to compare their products with the competitors in a quantifiable way. It also makes it possible to set a standard for the lowest acceptable quality in services.

When talking about speech, Denisowski [1] says that there are three main factors that determine how a human will perceive the quality of the voice transmission, delay, echo and clarity. All three have to be within certain levels for the speech signal to be deemed acceptable by an individual.

Delay is the easiest factor to quantify, since it is basically the time it takes for sound to get from the speaker to the listener. Usually, in the public phone network, the delay is simply a function of distance and mostly stays within the order of a few tens of milliseconds. There are, however, circumstances when delay can go into the hundreds of milliseconds, and then the delay starts to become a factor in the quality of the conversation. This can for example occur when the call is satellite-carried, because of the distance to reach a geostationary satellite. A signal travelling in the speed of light will still take somewhere in the region of 250ms to make the trip up and back. If the end-to-end delay exceeds 150ms, it will start to be an impairment to the conversation with talkers not realizing in time that the other has started to speak, and therefore risk talking simultaneously. With really high delays, conversation starts to only be possible in one direction at a time.

Echo means that the sound of the talker's voice is reflected back, since some of the transmitted signal is coming back on the receive path. This is desired in a phone conversation, since hearing the own voice in the earpiece of a phone without delay is comforting to the speaker. If it is absent, the line feels dead and like the other person cannot hear either. This form of echo is called sidetone. Since the impact of echo on speech quality is dependent on both delay and loudness, sidetone would have to be very loud to be a problem, as it has virtually zero delay. Echo that has a bigger delay will however not have to be very loud to impair the perception of the conversation. The echo has two possible causes, electrical or acoustic. An electrical echo is usually caused by less than optimal impedance matching or crosstalk, and the acoustic echo can for example be caused by coupling between the microphone and the speaker, e.g. in a speakerphone. To get rid of echo the options are to either suppress it or cancel it. Suppression is the simpler option, since the solution is to simply only let either the receive path or the send path be active, but not both at the same time. The problem there is to determine when the sender has stopped talking so that the other path can open and the receiver has an opportunity to talk. If poorly executed the delays can become long, and the one-way-only conversation mentioned above will be the result. Echo cancellation is a better method, but more complicated, since it remembers the sound that has been transmitted and subtracts any version of it that returns on the receive path. The method

works best with short delays, which makes other strategies such as delay reducing technology important on the network too.

Clarity is the last important factor and it is hard to quantify or measure, since it corresponds to such things as if the listener can make out the words the speaker is saying, identify his or her voice or make out slight nuances in speech that reveals the persons feelings. The way clarity has usually been tested is to set up a group of listeners that will grade a speech sample from 1-5 where 1 is bad, 5 is very good and 4 is defined as normal quality. This is called a MOS test, since the scale is called the MOS (Mean Opinion Score) scale. The problem with such a test is that it is logistically difficult as well as expensive to take in people to do them, especially since they ideally should be experts, so if regular tests are required it is impossible to use this method. To address this problem, objective methods that model the human perception of sound quality, such as PESQ and PEAQ, have been invented.

Jekosch and Blauert [2] talks about another way of defining sound quality in respect to the transmission of all types of sound, and not just speech. Their definition has three parts; authenticity, plausibility and enhancement.

Authenticity is an important factor in communication technology, since it relates directly to transmission. A transmission that is of the highest possible quality is said to have no measurable deviation in the received signal from the original after the transmission is done. It can also be formulated in the context of perception; if the subject can hear no difference between the original and the received signal, it is said to be of the highest possible authenticity.

Authenticity is, however, not always the goal. Since high authenticity is defined as the lowest possible deviation from the original signal, in the cases when enhancement is desired, e.g. in hearing aids or recordings, some parts of the signal is more important than others, so the received signal should be different from the original. Enhancement can for example be the artistic desire to create a certain musical atmosphere, the need to make speech more intelligible or creating certain acoustics in a room. This means that when enhancement is a factor, the simple straight-line transmission is no longer the main quality reference, but it rather has to be considered for each type of transmission and desired effect.

The last factor, plausibility, is quality in regard to the expectation the listener has of the sound. This factor is closely linked with sound engineering; creating sounds for products, movies or other things intended to deliver a message, behavioral or perceptual. The concept is that as long as the sound is deemed to be plausible for the situation, it can very well deviate from the original. Without an original signal, the authenticity cannot be determined, since there is nothing to compare the received signal to. The enhancement is also impossible to measure, for the same reason. The only thing that can be measured in that case is the perceptual factor of plausibility; is the signal reasonable for the expectations of it.

Raake and Blauert [3] says that when talking about stereophonic reproduction, which is the reproduction of sound in a way to give it a sense of direction and perspective, there are two factors that have been shown to be of great importance to the quality experience; spatial and timbral fidelity. Spatial fidelity is how well the space is recreated, and timbral fidelity is the quality of a tone that defines its characteristics, such as the difference between a note that is sung and a note from a piano with the same pitch and loudness. Experiments have shown that the perceived quality of such a setup is explained approximately by 70% timbral fidelity and

30% spatial. Models such as PESQ, PEAQ and POLQA do not use this measurement of quality for their assessments, but rely more on comparing the characteristics of the received signal with the original using perceptual methods.

In conclusion, it can be said that there are many factors that can influence quality, depending on what type of application the sound is used in. Looking only at telecommunications, there can be distortions due to, for example, codecs, packet loss, speech clipping or listener echo, but if there is also speech enhancement algorithms they can introduce disturbances depending on the background noise and suppression functions [4]. Therefore, it is needed to evaluate what type of transmission and sound is to be used, and what type of quality is desired before choosing a method for evaluating.

2.2 Problems with wireless connections

The goal of this thesis is to determine how much a sound is degraded over a wireless link, such as mobile phones or Bluetooth. The transmission channel is therefore of interest. When a signal is transmitted digitally, deviations in the received signal when compared to the sent is called bit errors. A bit error is when the information in a received bit is translated wrong, and mistaken for another symbol than the one that was sent. There can be several reasons for the distortions. The most common cause is the presence of noise, where the unwanted sounds in the background becomes interpreted as part of the signal, and therefore can push it out of the decision boundaries for the symbol it should be translated as. There is a lot of research on how to counteract the effects of noise, especially in relation to speech.

Noise is not the only thing that can cause errors. In wireless propagation channels, where the signal is sent through the atmosphere such as with radio, TV, mobile phones and satellite communication, different types of signal distortions can also affect the translation. The main types of signal distortions are delay dispersion, where echoes of the transmitted signal arrive with different delays, and frequency dispersion, where signal components arrive with different Doppler shifts. Delay dispersion is the most common cause for errors in channels with high data rates, and frequency dispersion is the most common at low data rates. Delay and frequency dispersion mainly occur due to multipath propagation, which is the fact that a signal can reach the receiver via different paths. When a wireless signal is sent out from a transmitter towards a receiver, it can take a direct route or bounce off objects in the environment, such as houses and mountains. Each of the paths the signal takes gives a distinct amplitude, delay, direction of transmission and phase shift on arrival. When a signal that has taken multiple paths is received by a simple receiver, it just adds up the different multipath components. The components can then interfere with each other. The different phases of the signal can lead to constructive or destructive interference, which affects the amplitude positively or negatively. The phase of the signal components are dependent on the distance from the transmitter to the receiver, so in a wireless situation where one or both of these can move, the effect can vary over the course of the transmission, and sometimes a small change in distance (for example moving one step when talking on a cell phone) can make a great deal of difference on the overall quality. The delays of signals can also interfere when information from more than one bit arrive at the receiver at the same time. The receiver cannot distinguish between the bit that should have been received at that time and the delayed bits, so that leads to a high probability of bit errors [5].

2.3 PESQ

2.3.1 Introduction

The Perceptual Evaluation of Speech Quality (PESQ) is an objective method for assessing quality in narrow-band telephone networks in the range of 300-3400 Hz and speech codecs. The method was created to evaluate end-to-end communication and is based on real world conditions, which means that it considers factors such as packet loss, noise, different types of delay and audio codecs [6].

The PESQ algorithm works by comparing an original signal, x(t) with the degraded signal y(t) that is the result when the original signal is passed through the system to be tested. The method maps the original and the distorted signals to an internal representation based on a perceptual model, and the goal of PESQ is to predict what a real person would think about the signal after it has passed through the system. PESQ can be divided into a couple of key parts: time alignment, auditory transform, disturbance processing and cognitive modeling, aggregation of disturbance in frequency and time, realignment of bad intervals and computation of the PESQ score [7][8]. It also takes into account such things as local gain variations and linear filtering that may not affect how a real human being would perceive the signal, and compensates for that so that inaudible disturbances are not considered when computing the score [6].

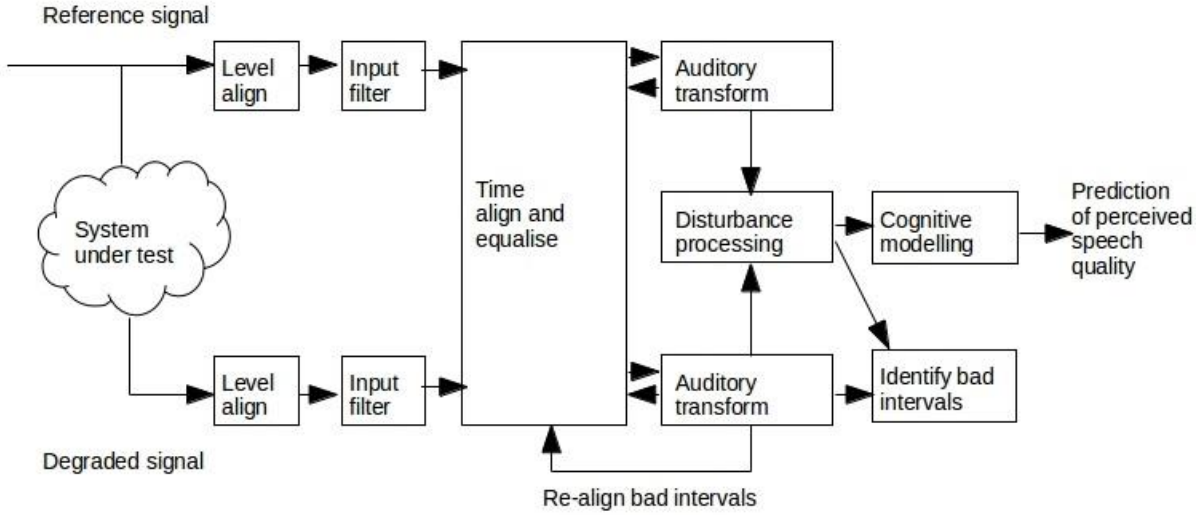

Figure 1: Simple model of the flow of the PESQ algorithm

2.3.2 Background

In the end of the 1990s, new technologies for speech transfer was developing fast. Voice over IP or ATM, mobile phones and new codecs were introducing new types of distortions that could affect the channel, and thus making the old quality measures like signal-to-noise ratio and frequency response functions become outdated. The reason new methods for quality prediction was necessary was that the old systems had been nearly linear and time invariant, something that the new technologies are not. The idea behind the quality measurements that were developed to address this issue was to measure the perceptual quality of the signal and try to predict what a subjective listener would think.

The first method based on the idea of perceptual quality was PSQM. It was standardized by the ITU-T as recommendation P.861 in 1996, and was developed for telephone-band speech signals. Out of five proposals, PSQM showed the highest correlation between its score and the

subjective score given by experts and was therefore chosen. That the scope of the recommendation was limited to only include telephone-band speech codecs was however a drawback to the method. Another problem the PSQM had was that when used with distortions it was not designed to handle, the correlation with the subjective quality scores was very low. The VoIP protocol can, for example, introduce a time warping distortion that makes the delay varied over the received signal, something that the PSQM could not handle. Other problems where PSQM algorithm failed included loud short localized distortions and linear filtering distortions, where the first were underestimated and the second overestimated.

Alternative systems and improvements to PSQM were developed in the following years to try to address the weak points of the method. The upgrade PSQM99 and the appendix to the recommendation MNB were two examples. A system called PAMS that could manage to accurately predict the impact of a wide number of distortions was one of the alternatives that were presented, and in 1999 ITU benchmarked it and four other systems that all could cope with many types of distortions. PSQM99 and PAMS were the two out of these five that had the best overall correlation to subjective quality scores, but none of the five proposals met with all the ITU-T requirements. Because of this, PESQ was created as an integrated method, taking the best features of PSQM99 and PAMS and combining them into one. From the former it took its perceptual model and from the latter the variable delay estimation, creating a method that was versatile, could deal with a large number of distortions and met the ITU-T requirements. It was accepted as the new standard for objective speech quality measurement in 2001, the ITU-T recommendation P.862 [8].

2.3.3 How PESQ works

2.3.3.1 Time alignment

A received signal can be delayed in relation to the original, a delay that does not always have to be constant during the whole duration of the signal. To address this issue a time alignment block is the first part of the PESQ method. This block also includes a gain alignment as well as the parts that calculate the different delays.

The PESQ algorithms perceptual model divides the signals to be compared into short frames, typically of 32ms, that are transformed using the windowed Fast Fourier Transform (FFT) and overlap by 50%. These frames are compared one by one, and the error parameters are calculated. If the frames are even slightly misaligned, however, false errors will be detected due to the FFT algorithm and the very nature of speech, which is time-varying. Therefore the delays must be cancelled out before the FFT is done and analysis can proceed [9].

Before the algorithm can remove the delays in the signals, the system gain is aligned in both the original and the degraded signal to make sure that they have a constant, and same, power level. This is done because the gain of tested transfer is not known beforehand, and can vary a lot. Also, the original signal is not stored at a pre-determined level, which needs to be compensated for. The PESQ algorithm therefore aligns the signals at a suitable target level using a method that works in two steps. The first part applies a filter to the original and the degraded signals to give more importance to parts that are perceptually more significant. The result is then used to calculate the overall system gain. The filter blocks all frequency components below 250 Hz, lets everything between 250 and 2000 Hz through and linearly suppress the signal in a piecewise manner above 2000 Hz through the points that follow:

{2000 Hz, 0 dB}, {2500 Hz, -5 dB}, {3000 Hz, -10 dB}, {3150 Hz, -20 dB}, {3500 Hz, -50 dB}, {4000 Hz and above, -500 dB}.

After the filter, the average values of the squared filtered original and the filtered processed signal are computed. The results are used to align both signals to a constant target level.

The time alignment process is divided into five different parts, each with a separate task to make sure the two samples are not delayed in relation to each other.

The first part is a so-called envelope-based delay estimation where the entire gain-scaled signals, both the original and the processed, are used. The envelopes are based on a function defined as $\log(MAX\left(\frac{E(k)}{E_{thresh}}, 1\right))$, where $E(k)$ is the energy in 4 ms and E_{thresh} is a voice activity detector-determined threshold for speech. The envelopes of the original and the degraded signals are cross-correlated to make a crude estimation of the delay between them, with a resolution of 4 ms.

After the first delay estimation both the samples are divided into smaller subsections known as utterances. Utterances are defined by Beerends, Hekstra, Rix & Hollier [9] as a continuous segment of speech with a duration of at least 300 ms and containing no silent period of more than 200 ms. The identification is made by a voice activity detector that can determine whether there is speech in the considered part of the signal or not.

When the utterances have been identified and divided into their parts, they also get an envelope-based delay estimation, just as was made for the whole signals.

The crude envelope-based alignments are however not enough to make the system secure against false errors due to delays, so a finer histogram based identification of the delays are made in each utterance. To make this finer alignment the signals are divided into frames of 64ms with 75% overlapping. These frames are Hann windowed and the frames of the degraded and the original signals are cross-correlated. The result of this correlation is used to determine the confidence of the alignment of the frames. To do this, the maximum of the correlation to the power of 0.125 is used to estimate the delay in each frame. This estimate, weighted by the confidence measure, is plotted in a histogram that is then smoothed by a convolution of a triangular kernel that has a width of 1 ms. A final delay estimate is given by the maximum of the smoothed histogram combined with the previous result of the delay in each frame, and a final confidence measure is calculated by dividing the maximum of the histogram by the sum of the histogram before smoothing it. This result gives a value of the delay of each utterance, together with a confidence value, in a way that takes into account the possible delay changes during silent periods. Since the start and end of each utterance is known, the delay of each frame can be identified to a high resolution.

The next step is to identify the changes in delay that can occur during periods of speech, since delay changes during silences have been compensated for. This is done by a method called utterance splitting. Each utterance is split into parts at several points, and the split with the highest confidence is found. If the confidence of that part is greater than that of the whole utterance without a split, and the delay of the two parts are significantly different, the utterance is divided according to the split. The test is redone recursively with each split to find all delay changes, even if there should be many in the same utterance.

After these steps, all delays, both within speech and silences, are hopefully accounted for, and a delay per time interval can be calculated, stored and passed on to the later parts of the algorithm. The matched start and stop of the samples are also calculated and passed on. If a delay has been missed by this method, the end of the perceptual model have a realignment scheme that might catch it [10].

2.3.3.2 Perceptual Model

When the time alignment has been done, the PESQ algorithm tries to predict the subjective quality of the speech signal as a human would hear and value it. The basis of this part is to use two measurements called Barks and Sones that are the psychoacoustical equivalents of frequency and intensity, and measure pitch and loudness. These measurements are then used to compare the original with the degraded signal, and determine the difference in internal representation to get a value of the audible changes to the signal and how much it will affect a person's experience of listening to it. This is done using a cognitive model that focuses on two effects that have a great impact on the listening experience, asymmetry effect and different weightings of speech and silence distortions.

The asymmetry effect appears when the codec distorts a signal, since any introduced time-frequency component will have a hard time integrating into the input signal, and will therefore be clearly audible as a separate distorted part of the output signal. When a time-frequency component is left out by the codec, however, the distortion is less audible, since there is no clear added irregularity.

The second effect works on the basis that noise during speech is more audible and affect the overall quality more than noise during silences [8]. The actual perceptual model is divided into several parts, each with its own role in determining how a human would hear the signal.

Before the perceptual algorithm can start, it is necessary to model the effects of an IRS (Intermediate Reference System) receive characteristic, since it is assumed that the listening is done using a handset with a frequency response according to this standard, and therefore the method need to model the signals the subjects hear. To achieve this a filter is applied to the original and the degraded signal to give them the IRS-like characteristics. The filter is always the same independent of if the listening test used regular or modified IRS, since there is usually no way of knowing the exact filtering or other handset parameters used, so the ITU-T required that the model should be relatively insensitive to what filtering the handset used, and therefore the method that gave the best results in the most conditions was chosen. PESQ does this by applying an FFT to the file and filter in the frequency domain with a piecewise linear response that match the IRS receiver, and then transform the file back again with an inverse FFT. The results $x_{IRSS}(t)$ and $y_{IRSS}(t)$ are the signals that are passed to the perceptual part of the model.

The first step is to make a time-frequency transformation, since the ear does just that. PESQ uses a short term FFT with a Hann window over 32ms frames, and an overlap between the frames of 50%. The model is only interested in the power spectra, which is defined as the sum of the squared real and the squared imaginary parts of the FFT components, and saves these in the variables $PX_{WIRSS}(f)_n$ and $PY_{WIRSS}(f)_n$. The phase information in the FFT is ignored. In this step the delays of the degraded signal as calculated in the previous section are also compensated for. The starting points of the frames are shifted to match the original signal and repetition or omission of parts of signals are made to compensate for delays within the frame.

The human ear has a higher frequency resolution in the low spectra than for high frequencies, and this has to be compensated for. The Bark measurement is made for this purpose and in this part of the PESQ algorithm a warping function mapping is made from the frequency scale in Hertz to the pitch scale in Barks, resulting in something called the pitch power density. A binning function in combination with the absolute hearing threshold is also used to calculate the FFT bands that are present in the ear.

The next step in the algorithm is to compensate for the linear frequency response, so that filtering in the system to be tested does not affect the outcome of the method. This is important because low or mild filtering effects hardly detracts from the overall quality, especially when an original is not available for comparison, but high effects have a larger impact. To achieve this the pitch power densities of both the original and the degraded signal are averaged over time. The average is calculated using only speech-active frames and time-frequency cells with a power of 30 dB more than the absolute hearing threshold. For each bin calculated in the previous step, a compensation factor is found by making a ratio of the spectrum of the degraded signal and the spectrum of the original, with a maximum of 20 dB The pitch power density of each frame of the original signal is then multiplied with the compensation factor of the frame to equalize it to the degraded signal and stored in the variable $PPX'_{WIRSS}(f)_n$. The compensation is done this way, with aligning the original to the degraded signal instead of the more intuitive other way around, because the degraded signal is the one being judged in a listening test, and therefore the one to be evaluated by PESQ.

Gain variations that only last a short time are compensated for by creating a ratio between the original and the degraded pitch power densities. The ratio is bound to the range $\{3 \cdot 10^{-4},\ 5\}$. This ratio is then modified using a first order low-pass filter along the time-axis with a time constant of about 16ms. The pitch power densities of the original and the degraded signal are then used to calculate the sum of the values in each frame that exceed the absolute hearing threshold. This result is then multiplied by the ratio to get a partially gain compensated distorted pitch power density. The resulting variable is called $PPY'_{WIRSS}(f)_n$.

The next step is to transform the compensated pitch power densities to a Sone loudness scale with the use of Zwicker's law. The formula is:

$$LX(f)_n = S_t \cdot \left(\frac{P_0(f)}{0.5}\right)^{\gamma} \cdot \left[\left(0.5 + 0.5 \cdot \frac{PPX'_{WIRSS}(f)_n}{P_o(f)}\right)^{\gamma} - 1\right]$$

Where $P_0(f)$ is the absolute hearing threshold and S_t the loudness scaling factor.

The constant γ is called the Zwicker power, and is defined as 0.23 when above 4 Bark. Below 4 Bark the power is slightly increased. The result of this calculation is called loudness density.

The signed difference in loudness density is the next thing to be computed. If the resulting difference is positive, noise components have been added to the signal when it passed through the system to be tested, and if the difference is negative, components have been left out from the original signal. The result is called the raw disturbance density.

Because small distortions in the signal are inaudible to the human ear when in the presence of louder signals, an effect called masking, this means that some disturbances will not be a factor in a subjective test, therefore those distortions should not be a factor in the objective test either. To filter out disturbances that are masked a so called dead zone is applied to each time-

frequency cell in the signal. The original signals and the degraded signals minimum loudness density value per cell is computed and multiplied by 0.25. These arrays are called mask arrays. Then a set of rules are applied to each time-frequency cell as follows: When the cell has a positive raw disturbance density that is larger than the mask value, the mask value is subtracted from the raw disturbance density. When the raw disturbance density lies between the positive and negative mask value, it is set to zero. When the raw disturbance value is negative and more so than the negative mask value, the mask value is added to the raw disturbance value. This modification models the dead zone before a distortion is perceived. The result is called $D(f)_n$.

The next part of the algorithm models the asymmetry effect that was described earlier. It is based on the fact that a signal that has been distorted by a codec, will receive a new time-frequency signal that will not integrate well with the original and the distortion will therefore be very audible. A component that has been left out will not be as noticeable. PESQ models this effect by multiplying the disturbance density $D(f)_n$ with an asymmetry factor that is defined as the ratio of the original and the degraded signals to the power of 1.2 and then set to zero if that ratio is less than 3 and limited at 12 if the ratio exceeds that number. The result of this is that the cells that remain all have a degraded pitch power density that is larger than the original pitch power density. The result is saved in the variable $DA(f)_n$.

The two previous results $D(f)_n$ and $DA(f)_n$ are summed over the frequency axis with a use of two different L_p norms and a weighting on frames that have a low loudness. The formulas are:

$$D_n = M_n \sqrt[3]{\sum \left(|D(f)_n| W_f\right)^3}$$

$$DA_n = M_n \sum \left(|DA(f)_n| W_f\right)$$

Where f = 1,..., Number of Bark bands, M_n is a multiplication factor defined as $\left((power\ of\ original\ frame + 10^5)/10^7\right)^{-0.04}$ and W_f is a series of constants that is proportional to the width of the Bark bins. M_n is used to make distortions that occur during silences more important for the calculation. The values obtained is then limited to the maximum value of 45, and the results are called frame disturbances.

In the case that the distorted signal have a delay decrease larger than 16 ms, the strategy that was used earlier to repeat parts of the signal is applied, but the frame disturbances are set to zero instead since it was found that it was better to ignore them in this case.

When the time alignment part have miscalculated a delay, improbably large disturbances can be seen due to that miscalculation, so when some consecutive frames with a frame disturbance above a threshold is found, this is called a bad interval and the delay has to be recounted. The new delay value is estimated by finding the maximum cross correlation between the delay compensated absolute values of the degraded and the original signal. When this maximum is below a pre-determined threshold the bad interval is assumed to be matching the correct parts against each other, the new delay is saved and the interval is no longer bad. If that cannot be found the frame disturbance is recomputed and if the new disturbance is smaller than the old, it is chosen instead. The results of this process is called D''_n and DA''_n.

The frame disturbances D''_n and DA''_n are each aggregated over split second time intervals using an L_6 norm over 20 frames and an overlap of 50% that makes a window function unnecessary. The result is then aggregated over the entire time interval using an L_2 norm. The result is a disturbance value.

The final part of the model is to compute the PESQ score. This is done using the average disturbance value and the average asymmetric disturbance value, linearly combined in a way that was developed through experiments using subjective opinions. The final possible range of the PESQ score is -0.5 to 4.5 [8][10].

2.3.3.3 Limitations

The PESQ algorithm is created for testing the objective speech quality in telephone communication and it is intended for end-to-end use and real world conditions. The method is more varied and can handle more types of possible distortions in the signal to be evaluated than its predecessors and contemporary counterparts, but there are limitations to its uses. The documentation of the recommendation lists some test factors, codings and applications where the PESQ algorithm have been found to predict the score with acceptable accuracy. It also lists conditions where the PESQ algorithm have been found to predict the score inaccurately or where it is not intended to be used. The documentation lastly lists conditions that have not been sufficiently tested with PESQ to validate its accuracy [8].

The following tables lists the conditions where PESQ will give an accurate score:

Test Factors
Speech input levels to a codec
Transmission channel errors
Packet loss and packet loss concealment with CELP codecs
Different bit rates
Transcodings
Environmental noise (the original signal sent to PESQ should however be noise free)
Effects of varying delay
Short-term warping
Long-term warping

Codings
Waveform codecs, e.g. G.711, G.726, G.727
CELP and hybrid codecs at 4kbit/s and above, e.g. G.728, G.729, G.723.1
Mobile codecs and systems, e.g. GSM FR, EFR, HR

Applications
Codec evaluation
Codec selection
Live network testing with digital or analog connection
Testing of emulated and prototype networks

Table 1: Conditions where PESQ will give an accurate score

The following tables lists the conditions where PESQ is found to not give an accurate score:

Test factors
Listening levels, since PESQ assumes a listening level of 79 dB
Loudness loss
Effect of delay in conversational tests
Talker echo where subjects hear their own voice delayed
Sidetone where subjects hear their own voice distorted

Codings
Replacement of continuous sections of speech making up more than 25% of active speech by silence.

Applications
Non-intrusive measurements where the original signal is not available.
Two-way communications performance
Music

Table 2: Conditions where PESQ will not give an accurate score

The following table lists conditions that PESQ has not been validated for:

Test factors
Packet loss and concealment with PCM type codings
Temporal speech clippings
Amplitude speech clippings

Talker dependencies
Multiple simultaneous talkers
Mis-match in bit rates between encoder and decoder if codec have more than one bit rate
Network information signals as input
Artificial speech signals as input
Listener echo
Effects or artifacts from operation of echo cancellers
Effects or artifacts from noise reduction algorithms

Codings
CELP and hybrid codes with a low bit rate, below 4kbit/s
MPEG4 HVXC

Applications
Acoustic terminal or handset testing e.g. using HATS
Wideband speech

Table 3: Conditions where PESQ have not been validated

All tables are obtained from [10].

2.4 PEAQ

2.4.1 Introduction

PEAQ stands for Perceptual Evaluation of Audio Quality and is an objective intrusive measuring technique using psychoacoustic measures and neural networks to evaluate the quality difference between a test and a reference signal [11]. PEAQ differs from other audio quality methods by not being primarily intended for speech but all audio.

The method uses models to mimic the human auditory system in its evaluation, adding noise and filtering frequencies in coherence with accepted psycho-acoustic research.

ITU-R started the process in 1994 to identify a method for audio quality measurements. To be able to see if the method was good and precise enough extensive subjective tests were conducted with selected signal databases. The results were used as reference with the same databases used to test the suggested models. Seven models were evaluated and compared, but since none stood out as better than the others the designers collaborated to develop an improved model. The new model was created in two versions with one real-time implementation and one that required more computations but was more reliable; basic and advanced version [12].

2.4.2 Background

Earlier subjective assessments of audio quality have been the only reliable method, but it is both time consuming and very expensive. The subjective evaluation methods that are used today are standardized and efforts have continuously been made to refine them, making them more efficient. Since the demand for audio quality assessment is steadily rising with the development of bit-rate reduction schemes for digital broadcasting, the need and demand of a reliable objective assessment method for audio quality is increasing likewise. Earlier objective methods such as Signal-to-Noise Ratio and Total-Harmonic-Distortion have never been fully reliable and can not handle the modern non-linear and non-stationary codecs [13].

The seven methods that are the foundation of PEAQ are: Noise-to-Mask Ratio(NMR), Disturbance Index(DIX), Perceptual Audio Quality Measure(PAQM), Perceptual Evaluation(PERCEVAL),Perceptual Objective Measure(POM), Objective Audio Signal Evaluation(OASE) and The Toolbox approach [12].

NMR measures the difference between the masked threshold and the noise signal. To analyze the frequency content it uses DFT with a Hann window of 20ms. It does not take any psychoacoustics into consideration, which makes it hard to compare to the subjective test results, but it also means that its calculations are simplified making it suitable for low power real-time application [13].

PAQM on the other hand is a general method to measure subjective audio quality using an ear interpretation called the internal representation which is calculated for both reference and test signal [14]. It operates in four steps: DFT with Hann window of 40ms for frequency region representation, warping the frequencies into the Bark scale, a non-linear time frequency function and then compressing the signal. The quality output is based on the measured difference between the internal representation of the reference signal and the noise disturbed signal. The output is mapped to a subjective grading scale [13].

PERCEVAL uses a model of the inner and middle ear, applied to the reference and the test signal, and then evaluates the difference to give a quality value. The signal is first decomposed with a discrete Fourier transform with a Hann window of 40ms and 50% overlap. Then it is multiplied with a frequency dependant function that is the middle and inner ear model. The calculation is of the detection probability of the difference between the reference signal and the test signal [13].

POM calculates an inner representation called artificial ear which is an excitation pattern. This is calculated for both the reference and the test signal. The purpose of the method is to quantify the degradation between the signals. That is what the method states; if the difference is audible or not and if so in which way. Similar to PERCEVAL it uses a discrete Fourier transform with a 40ms Hann window and 50% overlap [13].

The Toolbox Approach is a three step method using other well-known and tested methods to get a measure on the perceived audio quality. It uses an FFT with Hann window that shifts 10ms at a time plus adding masking effects. Then it does a weighing of the signals that depends on the loudness difference and time variation. The outputs include mean, max, rms and deviation from the mean based statistically on the previous calculations [13].

DIX uses an auditory filter bank which is more precise than the FFT. The audible range is covered by 40 filters with around 0.6 Bark resolution. To separate linear and nonlinear distortions the filters dynamically adapts the levels and spectra between the signals [13].

OASE also uses a filter bank of 241 filters. Here the center frequencies are at a 0.1 Bark distance meaning the filters partially overlap each other. The lower frequencies demand full sampling rate but the rate can be lowered at the higher frequencies, since lower frequencies are more precisely perceived by the human ear therefore demanding a higher resolution. A hearing model is calculated for both reference and test signal, also matching filters is applied to the signals to give a probability of audible detection value [13].

2.4.3 How PEAQ works

The basic PEAQ uses an FFT ear model only, while the advanced version also has a filter based model. Both of them combine the model output with a trainer neural network, comparing the reference and the test signal, to get a subjective grade according to the MOS scale used for subjective tests.

The first step in PEAQ is to adapt the levels of perceived loudness for the ear model output of the reference signal to match that of the test signal. This is important to be able to do a correct measurement and evaluation of the signal quality [12].

The model output and the test signal then has to be time aligned. Differing from PESQ there is no recommended time alignment function for PEAQ, each implementation has its own version, and the only demand is that the method reaches 24 samples accuracy which is the same as a 0.5ms resolution [13]. It can be assumed that the time alignment in PEAQ is not as essential and the function not as dependent on precise alignment as PESQ.

When these steps are handled the signals go through the model. The basic model only applies an FFT function comparing both the internal representations of the signals and the masked threshold concept, which is also known as noise signal evaluation [12]. These perceptual evaluations will be described later. Since it is only using FFT the basic version has a lower

real-time resolution but to compensate for this it produces more output variables and has higher spectral resolution than the advanced version. The output values given by the basic version are: the loudness of distortion, the amount of linear distortion, the relative frequency of audible distortions, the changes in the temporal envelope, the noise-to-mask ratio, the noise detection probability and the harmonic structure of the error signal.

The advanced version uses both an FFT function and a filter bank function to model the ear representation. Here the masked threshold concept is used with the FFT and the comparison of the internal representations is made with the filter bank. The output variables from the two functions are noise-to-mask ratio and harmonic structure of the error signal from the FFT and measure of the nonlinear distortions loudness, the amount of linear distortion and the disruption of the temporal envelope from the filter bank evaluation.

After the output variables have been calculated they are sent into the methods neural network where the representative subjective grade for the quality is calculated based on the model outputs. There are in total eleven model output variables used [15]:

MOV	**Explanation**
AvgBwRef	Average bandwidth of the reference signal
AvgBwTst	Average bandwidth of the signal under test.
NMRtotB	Total measure of the Noise-to-Mask ratio
ADB (ADF)	Average Distorted Block (or Frame). The logarithm of the ratio of the total distortion to the total number of distorted frames.
MFPD	Maximum Probability of Detection after low pass filtering.
EHS	Harmonic structure of error over time
RDF	Relative Fraction of frames where at least one frequency band contains an important noise component
WModDif1B	Windowed average difference in modulation between reference and test signal
AModDif1B	Average modulation difference
AModDif2B	Average modulation difference focused on introduced modulations and modulation changes where the reference signal contains little or no modulation.
NLoudB	RMS value of the average noise loudness with emphasis on introduced components.

Table 4: The output variables of PEAQ

The calculated PEAQ output is a grade between 0 and -4 that corresponds to the five step subjective impairment grade in the following way:

Impairment	Grade	ODG
imperceptible	5	0.00
perceptible, not annoying	4	-1.00
slightly annoying	3	-2.00
annoying	2	-3.00
very annoying	1	-4.00

Table 5: The PEAQ output grade

2.4.4 Perceptual models

As earlier described, PEAQ uses two different ear models depending on version. The FFT model is used by both versions while the filter bank model is only used by the advanced version.

The input signals, the test and the reference, sampled in 48 kHz when sent into the FFT model are cut into frames of 2048 samples each with 50% overlap. The signals are then mapped from time domain to frequency domain with a Hann window and a short term Fourier transform. Then the sound pressure scaling is calculated and the signals are adapted to have the same sound pressure level. After this the actual ear model algorithms are used starting with the outer and middle ear as a frequency dependant weighting function adding to the signal following the calculations of:

$$\frac{W(k)}{dB} = -0.6 \cdot 3.64 \cdot \left(\frac{f(k)}{kHz}\right)^{-0.8} + 6.5 \cdot e^{\left(-0.6 \cdot \frac{f(k)}{kHz} - 3.3\right)^2} - 10^{-3} \cdot \left(\frac{f(k)}{kHz}\right)^{3.6}$$

Given by the ITU-R recommendation for PEAQ where k is a time counter inside the frame.

The signals are then divided into frequency bands having converted the signal to the pitch unit Bark and calculated the auditory pitch scale. The Bark scale represents the 24 critical bands of hearing in the human auditory system. It ranges from 1 to 24 and is defined up to 15.5 kHz [16]. The width and spacing of the bands corresponds to a 0.25 Bark resolution for the basic version and a resolution of 0.5 Bark for the advanced version. This means the basic version has 109 bands and the advanced version has 55 bands. The frequency bands are mapped to the pitch unit depending on energy representation of the outer ear weighted output or the energy representation of the error signal [13].

After this mapping the internal noise, representing among other things the sound of the blood pumping in the ears, is added to the signals. It is a frequency dependant offset that is added to each frequency band, the output is referred to as pitch patterns.

The pitch patterns are then spread out over the frequencies in each frequency group with a spreading function. The function is a two-sided exponential function where the lower slope is always a fix value dB/Bark and the upper slope is frequency and energy dependant [13].

To be able to test forward masking the energies in each frequency group is spread over time with a first order low pass filter. The time constant for the filter depends on the center frequency for each group. The output patterns are called excitation patterns. The excitation patterns are then used to calculate the masking threshold, which is when a clearly audible signal becomes inaudible because a louder corresponding signal occurs. It is calculated with a weighing function weighting the patterns energies. They are also used to calculate the loudness patterns of the signal. Many of the output values are picked out along the way before the last step of the excitation patterns.

In the filter bank model the test and reference signals get adjusted to the same playback level, just as in the FFT model, but then they are passed through a fourth order Butterworth filter to remove any DC and subsonic components in the signal. Subsonic components refer to frequencies that is below what the human ear can hear. They therefore have no purpose when evaluating the audible sound quality and could disturb the filter evaluation.

Now comes the filter bank, with 40 filter pairs for each signal. The filter pairs are equal in frequency response but have a 90° phase shift. This means that the first filter give the real part of the signals and the second filter of the pair gives the Hilbert-transform, the imaginary part, of the signal. The filters are evenly spread and have a set absolute bandwidth related to the auditory pitch scale. The output from the filters are downsampled by a factor of 32 before entering the next step where a frequency dependant weighting function similar to the one in the FFT model is applied to the signals modeling the outer and middle ear.

The output from the filter bank is then smeared out over frequency with a two-sided exponential spreading function that is level dependant. The level is calculated for every filter and the spreading is done independently between the filters giving real-part and the filters giving imaginary-part of the signals.

All the following calculations in the model are based on the energies at the filter output. They are calculated by adding the squared values of the two filters in a filter pair.

In the filter model there is two time domain convolutions in order to model both backward and forward masking. The filter output energies are first spread over time by an FIR-filter with a $\cos^2$ shaped pulse response. The output signals are then downsampled again by a factor of 6. They are also multiplied with a calibration factor to keep the correct playback level. Then the inner noise representation, which is frequency dependant is applied to the signals. The output is now called the unsmeared excitation patterns. The same exponential spreading function as in the FFT model is then used to account for forward masking [13].

The output from this spreading function is now the filter bank models excitation patterns. They are used to calculate the specific loudness patterns and additionally, the unsmeared excitation pattern is used to calculate the modulation patterns.

In both the basic and the advanced version the outputs from the ear models are sent into a trainer neural network that evaluates and determines the objective quality grade, called objective difference grade ODG that is mapped to the subjective difference grade SDG, of the signals processed. The subjective impairment scale has 11 steps for the basic and 5 steps for the advanced version. The neural network performing the grade evaluation and mapping is a multilayer neural network that uses three units, for basic version, and five units, for advanced version per hidden layer. Outputs from the layers are generated by an asymmetric function.

The neural network is trained before proper use with all available data during development, but at the same time developers must make sure the network is not overfitted with information since it will start to make faulty evaluations if that happens. To guard against this cross-validation tests are used [12].

2.4.5 Limitations

One of the limitations that have been found concerning PEAQ is its inability to give a correct quality assessment when handling streaming audio. This mainly depends on two reasons. Firstly the fact that PEAQ was not designed or validated for the impaired audio that is the result of very high compression. Secondly the method has no implemented way to handle jitter while comparing the reference and test signal, meaning the comparisons will be incorrect. An extension of PEAQ for streaming audio has been suggested that is able to deal with these problems, PESAQ [17].

Another limitation is the fact that PEAQ was developed to handle typical audio coder distortions and therefore the neural network, which is part of the method, was only trained with data that complied to this. When using PEAQ in another environment it can still hold its ground but the results are much worse than normally [18].

PEAQ of course also has the requirement of the signals being time aligned to at least a set minimum but at the same time does not provide a recommended time alignment method. This means that the probability of a correct audio evaluation varies depending on the implementation.

2.5 Segmental SNR and Frequency Weighted Segmental SNR

The Signal-to-noise ratio, SNR for short, is one of the simplest ways of evaluating a channel there is. It is a measurement that compares the level of the desired signal to the level of background noise or the ratio of signal power over noise power measured in dB. Any measure over 0 dB means that the signal is stronger than the noise. But this plain measure is not of much use in determining the perceptual quality of sound, since it only measures the overall signal to noise ratio, and this have been shown to not have much correlation with subjective opinions of the sound quality.

A way to increase the probability of the method giving an accurate measure is to make the SNR segmental, in either the time domain or the frequency domain. Making it segmental means taking a SNR measurement over short periods, or segments, of the signal and summing them over the total number of segments to get an overall value.

To make the SNR measure segmental in the time-domain, the following formula is used:

$$SNRseg = \frac{10}{M}\sum_{m=0}^{M-1} log_{10}\frac{\sum_{n=Nm}^{Nm+N-1} x^2(n)}{\sum_{n=Nm}^{Nm+n-1}(x(n)-\hat{x}(n))^2}$$

Where $x(n)$ is the original signal, $\hat{x}(n)$ is the distorted signal, *N* is the frame length, *M* is the number of frames in the signal and n is the current frame. The frames are created by windowing the signal with a window length of typically 15 to 20 ms.

This measure needs that the original and the distorted signal are time-aligned and that there are no phase errors, if they are not, the method could make an erroneous evaluation and give a result that is not meaningful. This is because the comparison is made with the corresponding segment of the original signal, and if there are time or phase shifts, the segments will not match.

The Segmental SNR have been shown to give a much better estimation of the quality of the connection than the plain SNR measurement when correlated with subjective tests, probably because the segmentation gives a more accurate estimation of the perceptual impact of the noise, but it has a few problems. One is that the noise will be very large in comparison to the signal when there are periods of silence in the sent file, since there is virtually no signal during those times, but the noise is still there. The calculated value will then be a large negative SNR value, even though the perception of the quality is not that the signal is worse at those times. This problem can be counteracted by excluding the silent frames from the calculation by comparing the signal to a threshold or by limiting the SNR values so that the large negative values are not considered.

To make the SNR segmental in the frequency domain, the following equation is used:

$$fwSNRseg = \frac{10}{M}\sum_{m=0}^{M-1}\frac{\sum_{j=1}^{K} W_j log_{10}\left[X^2(j,m)/(X(j,m)-\hat{X}(j,m))^2\right]}{\sum_{j=1}^{K} W_j}$$

Here W_j is the weight that is placed on the j:th frequency band, *K* is the number of bands, *M* is the number of frames in the signal, $X(j,m)$ is the filter-bank amplitude of the original signal

and $\hat{X}(j,m)$ is the filter-bank amplitude of the degraded signal, both in the j:th frequency band and at the m:th frame.

This version of the Segmental SNR has the advantage that each frequency band can get a different weight, adding to the flexibility of the method, and enabling it to use a frequency spacing that is more perceptually significant, trying to copy the critical bands of the ear. The weights for the frequency bands can be chosen in many ways, but one way is to calculate them using a regression analysis to give maximum correlation between subjective and objective measures. The weights used are [19]:

Band number	**Center freq (Hz)**	**Weight**	**Band number**	**Center freq (Hz)**	**Weight**
1	50	0.003	14	1148	0.032
2	120	0.003	15	1288	0.034
3	190	0.003	16	1442	0.035
4	260	0.007	17	1610	0.037
5	330	0.010	18	1794	0.036
6	400	0.016	19	1993	0.036
7	470	0.016	20	2221	0.033
8	540	0.017	21	2446	0.030
9	617	0.017	22	2701	0.029
10	703	0.022	23	2978	0.027
11	798	0.027	24	3276	0.026
12	904	0.028	25	3597	0.026
13	1020	0.030			

Table 6: The weights used for each frequency band

The method gives a measure of the difference between the signal and the noise in dB. If the output given from the method is the number 10, it means that the signal is 10 times stronger than the noise. This means that the theoretical maximum and minimum of the method is ± infinity, but the code used in this experiment limits the output to be between 0 dB and 35 dB, where 35 means that there is no difference between the signals, and therefore no noise [19][20].

2.6 Log Likelihood Ratio

The Log Likelihood Ratio (LLR) is based on the difference between all-pole models of the original and the degraded signal. All-pole model is a model for linear prediction where an estimation of a time-invariant system is created through the observation of inputs and outputs. When using an all-pole model the output estimate is made based on the previous samples recursively. The LLR was created primarily to give a quality measure of speech signals. The model works on the assumption that over short intervals of roughly 15ms the sound can be represented by a p:th order all-pole model of the form:

$$x(n) = \sum_{i=1}^{p} a_x(i)x(n-i) + G_x u(n)$$

Where $x(n)$ is the sampled sound signal, $a_x(i)$ are the coefficients of the all-pole filter determined using linear prediction techniques, G_x is the filter gain and $u(n)$ is a unit variance with noise excitation.

The Log Likelihood Ratio divides the signal into frames of 15 to 30ms in length and the method compares the clean frame x_x and the distorted frame x_d and it calculates a quality measure using the function:

$$d_{LLR}(\overline{a}_x, \overline{a}_d) = log \frac{\overline{a}_d^T R_x \overline{a}_d}{\overline{a}_x^T R_x \overline{a}_x}$$

Where $\overline{a}_d$ is the Linear Predictive Coding(LPC) coefficient vector of the distorted signal modeled by the all-pole filter (1, -a_d(1), -a_d(2), … , -a_d(p)), $\overline{a}_x$ is the LPC coefficient vector of the clean signal modeled by the all-pole filter (1, -a_x(1), -a_x(2), … , -a_x(p)), $\overline{a}^T$ is the transpose of a and R_x is the autocorrelation matrix of the clean signal. The elements of R_x is defined as:

$$r(|i-j|) = \sum_{n=1}^{N-|i-j|} x_x(n)x_x(n+|i-j|) \; for \; |i-j| = 0, 1, \ldots, p$$

Where N is the length of the frame used in the LPC analysis as mentioned above. LPC is a method for representing the spectral envelope of a digital signal in compressed form, using a linear predictive model. It was created for speech evaluation, and uses the known frequency contents of a few basic sounds used in speech to determine its quality.

If the LLR measure is interpreted in the frequency domain, it can be shown that the method weighs the differences between the studied windows more when measured in formant peak locations, which are resonance peaks that are important for speech recognition.

An important requirement for the measure to give a valid output is the assumption that both the clean signal and the distorted signal can be described using the all-pole model. If the distorted signal is the product of a speech coder or another system that significantly alters the statistics of the original speech, this might not be the case and the measure will not deliver a usable result.

To get a single value of the LLR measure, the mean is calculated and the result is limited in the range [0, 2] where a score of 0 indicates that the compared signal files are identical [19][20].

2.7 Cepstral Distance

The LPC coefficients as calculated by the method above can also be used to derive a quality measure using cepstral coefficients. This measure calculates an estimate of the log spectral distance between the signals to be tested. To obtain the cepstral coefficients, the following recursive equation is used:

$$c(m) = a_m + \sum_{k=1}^{m-1} \frac{k}{m} c(k) a_{m-k} \quad where\ 1 \leq m \leq p$$

Where p is the order of the LPC analysis and a is the vector of the LPC coefficients.
The signals are divided into frames just as in the LLR measure and the Cepstral Distance measure is then calculated as:

$$d_{cep}(c_x, c_d) = \frac{10}{\ln(10)} \sqrt{2 \sum_{k=1}^{p} [c_x(k) - c_d(k)]^2}$$

Where c_x and c_d are the cepstral coefficients of the clean and distorted signal.
Computing the measure using the LPC coefficients makes the estimate of the log spectral distance that is calculated become of a smoothed spectra instead of a regular degraded signal with noise.

To get a single value, the mean of the distance measures is calculated and limited to the range [0, 10] where a score of 0 indicates that the compared signals are identical [19][20].

2.8 Weighted Slope Spectral Distance

The Weighted Slope Spectral Distance (WSS) is a perceptually based measure that uses the fact that studies have shown that when subjects rate the phonetic distance between synthetic vowels that have been altered by spectral manipulations such as low-pass or high-pass filters, amplitude changes or formant frequency differences, the largest phonetic distance was attributed to the vowels that had been manipulated in formant frequencies. Formants are defined as the frequency contents of different vowel sounds, and are the characteristic that makes humans able to differentiate one vowel sound from another. Because of this, a measure was proposed that is based on weighted differences between the spectral slopes in each frequency band. The method prioritizes penalising differences in formant locations and ignoring differences between the compared spectra such as overall level, spectral tilt and differences in the spectral valleys, since those differences have been found to have little influence of the actual perception of the sound.

The measure is created using a filter bank that contains 25 overlapping filters where the bandwidth increases with each filter. These filters approximates the critical bands of the ear and each band is weighted according to its perceptual significance. The bands and weights are the same as used in the Frequency Weighted Segmental SNR measure.

The first step of calculating the distance is to find the spectral slopes of each band. If $C_x(k)$ and $C_d(k)$ are the clean and distorted critical band spectra in dB, the spectral slopes are calculated as:

$$S_x(k) = C_x(k+1) - C_x(k)$$

$$S_d(k) = C_d(k+1) - C_d(k)$$

Where k is the band index.

The differences between the slopes are then weighted according to if the band is near a peak or a valley and whether the peak is the largest in the spectrum or not. The weight for the k:th band is calculated as:

$$W(k) = \frac{K_{max}}{[K_{max} + C_{max} - C_x(k)]} \cdot \frac{K_{locmax}}{[K_{locmax} + C_{locmax} - C_x(k)]}$$

Where C_{max} is the largest logarithmic spectral magnitude among all bands, C_{locmax} is the value of the peak that is nearest to the current band and K_{max} and K_{locmax} are constants that can be adjusted for maximum correlation with subjective listening tests. The suggested values are $K_{max} = 20$ and $K_{locmax} = 1$, but they can be recalculated using regression techniques for each test if desired.

The final distance measure is then calculated as:

$$d_{WSS}(C_x, C_d) = \sum_{k=1}^{25} W(k)(S_x(k) - S_d(k))^2$$

The result is the distance measure for each frame of speech. The final score is calculated as the mean score of all the frames. The LLR and Cepstral Distance methods are also sensitive to formant frequency differences, but they are sensitive to formant amplitude and spectral tilt changes which are not very important in speech quality assessment. An advantage of the WSS method is that to use it, it is not necessary to identify and align the formants or compensate for the differences in spectral tilt. It is one of the simplest perceptually motivated measures that exists. The WSS measure was created as a first step in determining how humans perceive speech, and particularly vowels, so its value in other areas of sound judgement might not be as big [19][20].

2.9 Subjective listening tests

In some instances, objective tests are not desired or possible to use, for example when validating or testing the objective methods, for test cases that have no recommended objective method or when it is really important that the sound quality is guaranteed to be good, in a way that the objective methods never can fully promise. In such cases, subjective testing methods are used.

In a subjective test, a group of human listeners are used to grade sound samples according to some predefined criteria, and since the people used are in most cases experts, the actual human perception of the quality can be measured with a high degree of certainty. The ITU-R have published recommendations [21][22] for how these listening test should be carried out to create a standard so that the tests can have a good accuracy and comparability.

The experiment should be designed to ensure the highest possible statistical certainty that the scoring is not affected by other factors than the actual impairments of the sound to be judged. For example, the sequence of the sound samples should not be in the same order for all subjects, because the judgements of one sample could be influenced by the one heard before or after. There should also be controls in the test, consisting of sound samples without distortions. If these are distributed randomly within the test, and without the subject knowing that they are undistorted, a conclusion can be drawn if the subjects are evaluating the impairments correctly. This consideration becomes more important the smaller the distortions are to be tested.

There are a couple of different methods to choose from when setting up the experiment, depending on what best suits the needs of the tester. If small impairments are to be tested, a method called “the double-blind triple-stimulus with hidden reference” [21] is recommended, since experiments have concluded it is especially sensitive. In the test, one subject at a time is presented with three stimuli, “A”, “B” and “C” to choose from. The sound “A” is always known to be a reference, and there is also a second reference available, randomly assigned to “B” or “C”. The last sound is the impairment to be tested. The task for the subject is to compare “B” with “A” and “C” with “A”, and to judge the difference according to a five-grade scale where 5 is an imperceptible impairment and 1 is a very annoying impairment. When one grading has been done, the subject can immediately move on to the next comparison in the test. Since one of “B” or “C” should be indistinguishable from “A”, quite small distortions can be detected this way.

If the distortions to be tested are not so small, tests can be carried out with a single sample being graded at a time, comparing two samples, where one might be the reference, or comparing many samples with each other, with or without reference. The subjects may be able to repeat sounds as required, or a fixed number of repetitions can be used, as desired by the test creator. A test session should not be longer than 15-20 min, since a human cannot concentrate in an effective manner for a very long time. If more time is needed, a pause of at least as long as one session should be inserted.

There are three different scales to use in the grading, depending on what quality attribute the test is focusing on. The following tables list the scales [22]:

Quality	Impairment
5 Excellent	5 Imperceptible
4 Good	4 Perceptible, but not annoying
3 Fair	3 Slightly annoying
2 Poor	2 Annoying
1 Bad	1 Very annoying

Table 7: The five-point scales used for grading sound quality

Comparison
3 Much better
2 Better
1 Slightly better
0 The same
-1 Slightly worse
-2 Worse
-3 Much worse

Table 8: The seven-point scale used for grading sound quality

For comparison, grading based on the five-point scales can also be used, but the tests using the seven-point scales and the five are not equivalent and cannot be expected to give the same results.

When giving a score, the resolution of the scales is recommended to be 1 decimal place.

The test subjects should preferably be made up of experts, even though it has been argued that non-experts may give a more accurate representation of the population, and that the experts would be more critical than necessary. However, experts will give a quicker and better result, especially if the tests are carried out over a longer time. Also, the smaller impairments that are to be tested, or the higher quality the product or service is desired to have, the more important it gets to use trained ears that can distinguish the small differences in quality.

If experts are used, the minimum number of participants in the test should be ten, where the minimum number for non-experts should be twenty, but with small impairments that require experts the group size should increase to at least twenty.

Before the test, the subjects should be given a training phase to get familiarized with the test method, what factors they are supposed to focus on and the material used.

The material chosen for the test should be able to deliver the desired test condition, be around 10-25s long and not be interesting or boring enough to distract the subject from the task at hand. The chosen material can vary a lot depending on the purpose it is to be used for, but the importance of the right sample decreases with the complexity of the test. The loudness of the sample should be adjusted to an acceptable level, according to a couple of defined criteria [21], before the test can start.

The headphones and/or speakers used for the tests should be as good as possible to not be a factor in the testing, but this also becomes more necessary the more sensitive the testing is. For a really high quality test, the size, shape, reverberation characteristics and proportion of the room should be taken into consideration, among other things. The equipment used should also meet some predefined properties [21].

When the test has been carried out, the results should be presented with a statistical analysis, describing the mean value, variance and confidence. A significance level of 0.05 is traditional. The presentation should also contain what subjects have been selected, some details about the room and equipment that was used so that their impact on the test can be evaluated, and the design of the experiments with instructions to the subjects, the sequences used, the test procedure and a review of the conclusions of the test [21][22].

CHAPTER 3

3 Research

3.1 Comparative tests

To create a model for sound quality assessment based on non-licensed methods, a large number of tests were carried out to determine which models to use and how to combine them for the best result in eight different distortion types respectively.

Five methods were chosen from the Matlab implementations made by Loizou [19], Frequency Weighted Segmental SNR, Log Likelihood ratio, Cepstral Distance, Weighted Slope Spectral distance and Segmental SNR. These methods were chosen because they had the highest correlation to subjective ratings in the tests carried out by Loizou, apart from the licensed method PESQ. When tested on audio files they also had a relatively good correlation to the scores set by PEAQ, making the methods appropriate candidates for a combined method for both the speech and audio case. The PEAQ implementation used for the project was made by Kabal [23].

3.2 Sound files

The speech files used in the tests were also taken from Loizou [19]. 30 clean speech files with both male and female speakers was available, and also distorted files, where eight types of background noise had been added to give a resulting SNR of 15 dB, 10 dB, 5 dB and 0 dB. Only the 15 dB, 10 dB and 5 dB levels were used in this research.

Background noise is the same as additive noise. It means that except from the speech or audio the receiver wants to hear, the transmitting microphone picks up and sends the noise in its immediate surroundings through the same channel. This means that the real signal will be added with the surrounding noise when coded and sent, disturbing it and decreasing the quality of the speech. The surrounding noise can for example be speech in the background, the sound of a car or wind. The three decibel levels indicates the sounding strength in regard to the noise and therefore also its degrading effect since the higher decibel value the noise has the more it will distort the real signal.

To give the combining of the methods as much background as possible the 30 original files were also distorted using the sound editing program Audacity, introducing a wider range of distortions than just added noise. The distortions introduced using Audacity were echo, clipping, reverberation, low-pass filter and a packet loss simulation. A bandstop filter was created and applied using Matlab.

Echo is an effect caused by imperfections in the transfer medium that reflects the sent signal back to the original sender. The transmitting device will then hear its own sent signal with delay. Here the echo is added with four different delay times: 1s, 0.5s, 0.1s and 0.05s. These delay times say how long it will take for the transmitting device to hear the signal it sent.

To add clipping to a signal it needs to be amplified to the point where the transfer codec, medium and receiver will cut some of the peaks in the signal since they are too big for the spectrum. The clipping of the signal means that the signal is now incorrect and this will of course affect the quality of the signal. The amplitude of a signal is measured in dB and the introduced peak amplitude values are 1 dB, 2 dB, 3 dB and 5 dB. The different amplitudes will mean that the amount of signal that is clipped increases when the amplitude does.

Reverberation, or reverb, when talking about signals means that the receiving device gets not only the direct signal that traveled the shortest way but reflections of the signal from surfaces such as walls. These reflections signal strength depends on the dampening qualities of the reflecting surfaces, the strength of the signal at the transmission point and also the frequency of the signal. The reflected signals will arrive at the receiver with delay but then they decay in time when they keep reflecting from different surfaces. This will create the impression of the transmission being made in a room. Dealing with telephones it is the microphone that will first detect the direct and reflected signals and then sent them to the receiving phone, where it will appear as if the transmitting phone is inside a room. There are four different levels of reverberation introduced as distortions to the test files: 10%, 30%, 50% and 80%. They are presented as percentages that states how much of the reflecting signals are absorbed by the simulated reflecting surfaces and how much is reflected. This means that the highest percentage has the highest amount of reverberation since 80% of the signals is reflected, thereby following that only 20% was absorbed, before it was recorded.

Introducing a low-pass filter as a disturbance means that the signal is filtered and only the frequencies in the signal that are equal to or lower than the so called cut-off frequency are allowed to be fully transmitted, while those that are higher gets damped and then weaker over time. How weak how fast depends on how sharp the low-pass filter is. The filtering is distorting the signal by removing part of the signal information, though there could exist information in the same time span since the file is still of the same length and the amount of samples have not changed. In this project five different cut-off frequencies were used: 2000 Hz, 1000 Hz, 700 Hz, 500 Hz and 300 Hz.

When packet loss occurs parts of the signal is completely lost. When information of any kind is transmitted it is divided into parts, encoded and then sent in a specific order, these encoded parts are called packets. Sometimes during transmission packets are lost or dropped because of errors in the coding or disrupting signals, for example. Depending on the protocols the receiver might ask for the transmitter to resend packets but that is not especially realistic or effective when dealing with real-time situations such as mobile phones for example. The test files simulate packet loss by having completely silent periods, where the signal was muted with the help of the previously mentioned audio modification program Audacity. Each test file has four muted segments of equal length. Five different muted segment lengths are used, each in its respective file: 0.1s, 0.5s, 0.3s, 0.01s and 0.005s.

A bandstop filter dampens a range of frequencies, where the center frequency is completely muted. This distortion allows most of the signal to pass through undisturbed, while a range of frequencies is virtually omitted. To achieve this, the Matlab toolbox fdatool was used. This is a graphic interface that allows the user to design a filter and use it in Matlab. The response type bandstop was chosen, and the design FIR. Four filters was designed: The first filter stops the frequencies 280 - 300 Hz, with attenuation from 1 - 700 Hz. The second filter stops the frequencies 490 - 510 Hz with attenuation from 200 - 800 Hz. The third filter stops the frequencies 990 - 1010 Hz with attenuation from 700 - 1300 Hz. The last filter stops the frequencies 1990 - 2010 Hz with attenuation from 1500 - 2400 Hz. The FIR filter was chosen because the Butterworth IIR filter created an unwanted noise in the filtered signal. However, when using FIR, a wider span of attenuation was needed for convergence.

The music files were taken from a cd created for testing audio systems, available for download at the European Broadcasting Union. Ten files were chosen from this cd: solo bassoon, solo flute, solo violin, solo cello, solo organ, solo vibraphone, a singing quartet, “Der Hölle Rache” from the Magic flute by Mozart, a piece of Haydn’s trumpet concerto and the opening of “Also sprach Zarathustra” by Strauss. Only ten music files were chosen and not 30, as in the case of the speech files, because the music files are significantly longer and have a higher sampling rate and therefore take much more time to test and contain more signal information in each file.

The music files were distorted using the sound editing program Audacity and the same distortions as were used for the speech files and are described above: noise addition, echo, clipping, reverb, low-pass filter, packet loss simulation and bandstop using Matlab. But here the noise added to the files is pink noise compared to the white noise added before. The difference between white and pink noise is very small. Both kinds contain all for humans audible frequencies, 20 Hz to 20 kHz, but in white noise the power per hertz is the same for all frequencies as opposed to pink noise where the power per hertz is lower for higher frequencies than for low frequencies. Pink noise was chosen instead of white, because the PEAQ method is sensitive and much lower amplitude levels was required to get reasonable

results when using white noise. The program Audacity used to add the noise has a stated linear range with maximum and minimum level of 1 and -1. The added pink noise amplitudes were: 0.01, 0.005, 0.002, 0.0015 and 0.001 in an amplitude range of 0-1.

As for the rest of the added distortions the same distortion levels as in speech were used as with audio for each distortion category:

- Echo with the delay times 1s, 0.5s, 0.1s and 0.05s
- Amplification to the point where clipping is introduced with peak amplitude values 1 dB, 2 dB, 3 dB and 5 dB
- Reverberation with the reflection percentages 10%, 30%, 50% and 80%
- Low-pass filter with the different cutoff frequencies. In the speech files the frequency span was the same for all files but the music files has such varied frequency content that the cut-off frequencies had to be adapted for every file, with five different levels each. The frequencies are listed in the table below in Hz:

Music file	**Cut-off 1**	**Cut-off 2**	**Cut-off 3**	**Cut-off 4**	**Cut-off 5**
Bassoon	300	500	800	1000	1300
Cello	500	800	1000	1500	2000
Flute	1000	2000	3000	5000	10000
Opera	800	1000	3000	5000	7000
Organ	1000	3000	5000	7000	10000
Quartet	800	1000	3000	5000	7000
Trumpet	1000	2000	3000	5000	7000
Vibraphone	800	1000	2000	3000	5000
Violin	1000	3000	5000	7000	10000
Zarathustra	2000	5000	7000	10000	15000

Table 9: Cut-off frequencies for the music files

- Simulation of packet loss where eight segments of each file was muted with the segment sizes 0.1s, 0.5s, 0.3s, 0.01s and 0.005s
- Bandstop filter with the stop frequency ranges 280-300, 490-510, 990-1010 and 1990-2010 Hz, and attenuation ranges 1-700, 200 - 800, 700 - 1300 and 1500 - 2400 Hz.

3.3 Combination of methods

To be able to add the selected unlicensed methods together into one score, they had to be modified since the programming of them was not made for combining. A function was created that made the method addition possible with the help of linear regression, and also made the combination giving the new complete score.

In the case of the Segmental SNR and the Frequency Weighted Segmental SNR, the score received from the method is a measure of the overall SNR in dB, in the range of -10 - 35, where 35 indicates that there is no difference between the tested files. It was desired to make 0 the lowest value, so it was decided to add 10 to the received score, and make the range 0 - 45 instead. To make the score linear and usable, it was first converted to a linear scale by:

$$SNRscore = 10^{(45-(SNRscoredB+10))/10}$$

This was done for both the Segmental SNR score and the Frequency Weighted Segmental SNR score. The reason the score was subtracted from 45 is so that a score of 45 will give the linear score 0, to get a value of how much noise there is instead of how much signal. However, the equation above will give the result 1 if the score is 45, instead of the desired result 0, so a condition was added to make the result 0 if this should happen. For all other values, the correct answer will be given. The linear score was then divided by the theoretical maximum of the method ($10^{45/10}$) to get a percentage value. This value was then subtracted from 1 to get a value of how good the signal is, instead of how bad it is. The new percentage value was then multiplied by 5 to get a value in the range of 0 - 5.

The LLR delivers a value in the range 0 - 2 in a logarithmic scale with base e. A score of 0 indicates that there is no difference between the files. To make the score linear, $e^{llrscore}$ was calculated, if the score is not 0. The score was then inverted by subtracting the theoretical maximum (e^2) with the score, so that a score of 0 mean that the files are totally different, and a percentage value was calculated by dividing the score with the maximum value. The percentage value was then multiplied by 5 to get a value in the range of 0 - 5.

The Cepstrum Distance gives a value between 0 - 10, where 0 is no difference and 10 is completely different. Since the cepstrum value is already linear, all that needed to be done was to invert the score by subtracting it from 10, calculate the percentage and multiply the percentage by 5.

The WSS score does not have a defined maximum, and it also measures distances, so that a score of 0 is no difference. A test was made where the WSS was limited to 400, based on some quick observations of common outputs, and inverted, but the correlation in combination with the other methods was worse than just using the score as it was, so the WSS was not modified for the testing. For the finished software packet, however, a limitation that gave an acceptable correlation was found, and the WSS is limited by calculating:

$$WSS = x - x \cdot \tanh(\frac{wssscore}{100})$$

Where x is the maximum desired value for WSS since it gives the stable level of the tanh function. To be consistent x was set to five so that all functions have the same maximum score. Since the scoring from WSS was reversed, meaning low numbers were good and high

numbers were bad, the score was subtracted from the desired maximum score, making zero the lowest score for a bad result and five the highest for a good result.

To combine the methods a linear regression analysis was used. A vector was created containing the values of the chosen reference method (PESQ or PEAQ), and a matrix containing the scores of the methods to be combined, with the first column only being ones. A linear regression model has the following form:

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

Where y is the score of PESQ or PEAQ, x_{ij} is the value of the *i*:th score of the *j*:th method, and β_j is the weights to be calculated. The calculation is done in matrix form:

$$Y = B \cdot X$$

$$B = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$$

These weights were then used to combine the chosen methods into one score using the earlier mentioned function.

3.4 Testing the sound files

To test the sound files, the original and the degraded signals were sent through the chosen methods and weighted according to the linear regression analysis. The signals were also sent through PESQ if speech was tested, or PEAQ if music was tested, to provide a "correct" answer for what the score should be. PESQ and PEAQ was chosen as the baseline since they are commercial methods of much greater complexity than the measurements that make up the freely available methods, with a proven high correlation to subjective scores in a great number of disturbances. The names of the clean sound files and the degraded sound files were saved in two .m - files, deg and reference. Where each row of the reference file contained the name of the clean sound file corresponding to the distorted signal on the same row in the deg file. Each line of both files was then read one at a time using the Matlab command fgetl(), and the files sent into the methods. The results of each test was saved into a matrix of two columns, with the value of the reference method in one column and the combined method to be tested in the other column and then written into a .m - file called result using the fprintf() command. The correlation between the results of the reference method and the combined method was calculated using Pearson's correlation coefficient:

$$\rho_p = \frac{\sum_{j=1}^{n}(x_j - \overline{x})(y_j - \overline{y})}{n\sigma_x\sigma_y}$$

Where n is the number of observations, x_j is the j:th observation of PESQ or PEAQ, y_j is the j:th observation of the combined method, $\overline{x}$ is the mean of all PESQ or PEAQ observations, $\overline{y}$ is the mean of all observations of the combined method, σ_x is the standard deviation of PESQ or PEAQ and σ_y is the standard deviation of the combined method. The standard deviations are calculated as:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{n}\left(x_j - \overline{x}\right)^2}{n}}$$

Where n is the number of observations, x_j is the j:th observation and $\overline{x}$ is the mean of the observations. Pearsons coefficient is a common method for evaluating how well two sets of data compare to each other, and the result gives a value of the linear relationship between the variables $\{x_i\}_{i=1}^{n}$ and $\{y_i\}_{i=1}^{n}$. The coefficient is in the range of $-1 \leq \rho \leq 1$, where a value of 0 indicates no linear relationship between the data sets, and a value of 1 or -1 indicate a complete correlation where the points are lying exactly on a line where *y* increases or decreases as *x* does. Since a negative value only indicates that x_i and y_i tend to not be simultaneously greater or lesser than their respective means, that value is of little interest, and so the absolute value of the Pearson's coefficient was used and written into the last line of the result file.

All the sound files were then tested using all combinations of the chosen methods, which means that 31 tests were carried out for each type of distortion. Each type of distortion was tested separately to assess how well all of the methods, separately and in all possible combinations, work with that particular type of distortion.

When all tests had been run, a couple of tests were carried out to confirm that the distortions actually was captured by the methods, and that the correlation was not just a product of the

regression analysis. To do that, the four combinations of the music tests and the six combinations of the speech tests with the highest correlation scores was chosen and tested again using half the test files to calculate the regression and the other half to calculate the correlation, with both halves used for both things. If the scores are high enough, the regression is valid for all types of files with that particular distortion.

3.5 Real-world recordings

Real-world recordings were made to get an idea of how the sound can be affected by a mobile or Bluetooth connection in real life, and how well the proposed solution could handle that. The regression calculations obtained with the real life recordings became the foundation of the weights used in the software packet.

When the recordings were made, six different phones were used, one Bluetooth headset and two Bluetooth speakers. The phones used were: a Sony Xperia acro S LT26w, a HTC one S, Samsung/Google Galaxy Nexus, a Nokia 808, a LG Optimus True HD LTE P936 and a Blackberry Bold 9900. The headset was a Sennheiser MM450-X, and the speakers was one NDZ-03-GA and one Sandstrøm Juice SJUPBL14E. A Samsung Galaxy S3 GT-I9300 was used to connect to the Bluetooth headset.

The phones were combined in different ways so that all phones got to be the sender and the receiver in at least one setup.

The recordings were made using a head and torso simulator from Brüel & Kjær Type 4128C to record the received signals. It was connected to the laptop that made the recording via a Brüel & Kjær Nexus Conditioning Amplifier that controlled the microphones and provided the head and torso simulator with power. This setup was placed inside an echo-free room with very good sound insulation. The phone was placed in a holder by the ear that was calibrated to match the average position of a cell phone during use.

The phone that sent the signal was placed by a speaker of the make and model Fostex 6301B2. This in turn connected to a laptop that played the sounds to be sent. This setup was placed outside of the echo-free room.

Tests were made so that the phone was at a reasonable distance from the speaker, and so that the speaker had an appropriate volume. To make the recordings, the phone by the speaker made a call to the phone on the head and torso simulator and the sound files to be tested were played.

The headset was tested using both the headphones and the microphone. When testing the headphones, the headset was placed on the head and torso simulator and connected to a phone. Another phone was placed by the speaker and made a call to the phone inside the echo-free room. The headset then recorded the transmitted sound files.

When testing the microphone, the headset was connected to a phone and placed at a reasonable distance from the speaker. Another phone was placed on the head and torso simulator and a call was made to it to send the files via the microphone of the headset.

The Bluetooth speakers were tested using only the head and torso simulator. The speaker was placed in front of it, connected to a laptop nearby and the sound files were played.

The sound files received from these tests consisted of all the test files in one long segment. To get the files separated the long recorded file was manually divided using the sound editing program Audacity.

3.6 Testing the software packet

The software packet was tested using the same setup as the real-world recordings, but it was used to split the files and evaluate the results, just as if it was a real test. The only difference from a real test was that the program delivered both a calculated score and a PESQ or PEAQ score for comparing with. When a sufficient amount of tests had been run, the correlation between the result from the software packet and the PESQ or PEAQ score was calculated.

For music, two devices were tested. One Bluetooth headset: a Sennheiser MM450-X, and one Bluetooth speaker: Sandstrøm Juice SJUPBL14E. Two tests were made with each device, making it four tests total.

For speech, six phones was tested. The phones involved in the test were: a Google Nexus 5, a Samsung Galaxy S3, a HTC one S, a Samsung/Google Galaxy Nexus, a LG Optimus True HD LTE P936 and a Blackberry Bold 9900. These phones were combined in eight different ways, and two tests were made for each combination, making it sixteen tests in total.

CHAPTER 4

4 Software packet

The finished product has a main script running the recording and analysing of the sound files. To make the sound file recording as easy as possible, selected music and speech files were concatenated into one long file that the main script then separated with the help of pilots placed in the file. The pilots are a total of three seconds long: 1s silence, 1s of 3000 Hz and 1s silence. The pilot was made to be 3000 Hz because it is high enough for the speech files to not contain that frequency, and for the music files, frequencies that high is sufficiently uncommon to not affect the finding of the pilots. 3000 Hz is also high enough that it is unlikely that noise in the test environment will disturb the finding of the pilots. The pilots are, however, not above the range that a phone can transmit.

Once the test file is recorded it is sent to a script for separation, splitfile(). This script finds the pilots and separates the concatenated files so that they can be analyzed correctly. Once all the files from the test file have been found and accounted for the main function starts the analysis one file at a time following the procedure previously described in 3.4 with the help of two .m-files listing the files to go through. If speech is being tested then they are referencespeech.m and degspeech.m, if music is tested then they are referenceaudio.m and degaudio.m . Since the recording of a long file and the separation does not give any control over possible delay in the degraded files this is measured against the corresponding clean file before further analysis. If a delay exists, the concerned degraded and clean file is sent to a separate analysing script adapted for the delay. If there is no delay that separates the original from the received file than they are analysed without the delay script modification.

The earlier tests of the different combinations of methods handling different disturbances resulted in two separate weighting vectors, one to be used for speech and one for audio. The scripts for the methods used are made by Loizou [19]. During the analysis of the files the corresponding weights are used and the end result saved and displayed.

The received test files are resampled to match the original files, since different channels have different sampling frequencies.

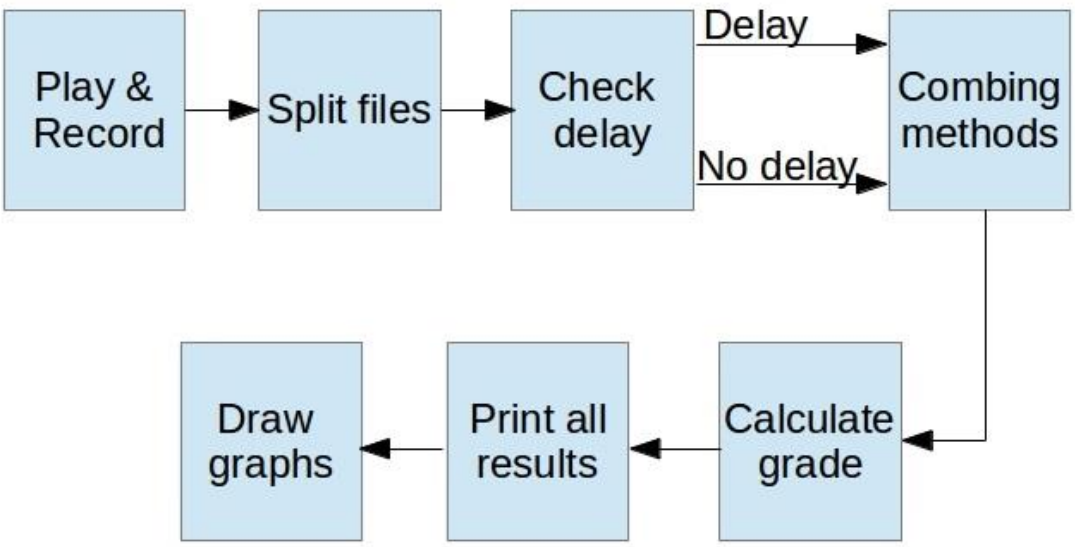

Figure 2: The flow of the software packet

The test setup has to include both the channel to be tested, phone-phone, phone-speaker for example, but also the analysing computer equipped with a mic and, if needed, a stereo setup to play the long test file into the channel. The sound file and the analysing program needs to be started at roughly the same time, therefore a warning has been placed in the product telling the operator when to start the sound file.

4.1 Recording

The program starts by recording the sound file to be tested. This is done by creating a recorder object with the Matlab function audiorecorder. This recorder is then started and made to record for a predetermined length of time. The time to record is 210 seconds if the file to be tested is speech, and 600 seconds if it is music. This is a little longer than the actual length of the sound files but the added seconds are there to give a safety zone in the recording for possible start-up delay. The recorded file is then written to a .wav file using the command wavwrite.

4.2 Splitfile

The splitfile() function takes in the whole recorded sound file. To make the separation possible, pilots consisting of 1 s silence, 1 s 3000 Hz and 1 s silence is placed in the original file that is sent and recorded, containing all the sound files to be tested. The pilots are placed before the first file, between every file and after the last file.

The script first reads the received file into Matlab using the function wavread(), obtaining a vector of the sound, the sampling frequency and the bitrate. The program then calculates how many samples there is in half a second by dividing the sampling rate in half, and how many half seconds there is in the sound file by dividing the amount of samples in the sound file by the amount of samples in a half second, and rounding down the answer. If the total number of calculated half seconds times the number of samples in a half second is equal to or larger than the length of the recorded sound file the index is reduced by one. A while-loop then runs as long as the index i is less than the number of half seconds the sound file has. In the while-loop the FFT of the current half second of the signal is calculated, by using the Matlab function fft() over the current i multiplied by the amount of samples in a half second and the current i + 1 multiplied by the amount of samples in a half second, and the frequency with the maximum power in that area is obtained. If that frequency matches a pilot, the current half second is added to a vector, to save its location, and 3 is added to the index of the while-loop so that it can be guaranteed that the next area where the program calculates the FFT will not contain the pilot it just found. If a pilot was not found in the current half second, 1 is added to the index.

The next step is to separate the files, so another while-loop then runs as long as another index i is less than the length of the vector where the pilot locations were saved. The i:th and the i:th + 1 entry in the location vector is saved as *a* and *b*. The pilot is then trimmed out, so that it will not be present in the separated file. This is done by moving *a* forward 3 half seconds and *b* backwards one half second. Everything between *a* and *b* is then saved into a new variable and converted into a .wav file using the Matlab function wavwrite(). The new file gets named after the current i. The index i is then updated by adding 1. When the loop is finished, the function prints out how many files were created.

4.3 Combo

The combo() function goes through the separated files in order and obtains the scores from each of the chosen methods for the file.

First combo() checks to see if the file from the channel and the reference file are delayed in respect to each other. If they are then they are both sent to a function called delayedCombo(). The only difference between the two functions is that delayedCombo() aligns the two signals before further processing. The alignment is done by the Matlab function alignsignals() and a temporary file is created for this script. When all method values have been calculated the temporary files are deleted. The degraded file is also resampled to the sampling frequency of the reference file in case they differ. This is done by using the Matlab function resample(). The combo() and delayedCombo() functions then both obtain the scores of the methods, which all have been recalculated to a common scale between zero and five in the same way as described in section 3.3, and sends them back to the QualityTest-script saving them in a matrix called methods.

4.4 Weighting

The next step in the program is to calculate the combined scores for each file and distortion type. This is done by sending the methods matrix and the weights for one of the distortions into the function weighting(). The distortion weights are previously saved in a file and loaded into the program by using the function load(). In the weighting() function the score for each file is calculated in the same way as described in section 3.3 and saved in an array which is returned from the function and saved as a column in a matrix called val.

4.5 Calcval

The final function call of the program is called calcval(), and this part calculates the final grade for the test. The function starts by creating a mean value for each type of distortion over all files. It then compares this value with values stored in a file for the closest match. These values are the calculated values for that type of distortion if all methods give out a score between 0-0.01, 0.01-0.02, 0.02-0.03, and so forth all the way up to 4.99-5. The region that is closest to the average score then becomes the score for that distortion. When this has been done for all distortions, the average score is the final score for the test.

The reason to use a table to find the score is that the weights for the distortions are so very different and therefore their respective score will also be very different. Unfortunately, the fact that there are five variables that can change independently when calculating the score means that the list of possible scores will not be linear. Therefore the table solution was chosen, because it gives a good idea of where the score should be while still preserving the importance the weights placed on each method.

The scores for each distortion and the average score of all distortions is then saved to a result file, and a graph is displayed along with the final score of the program. An example of the graph is seen below:

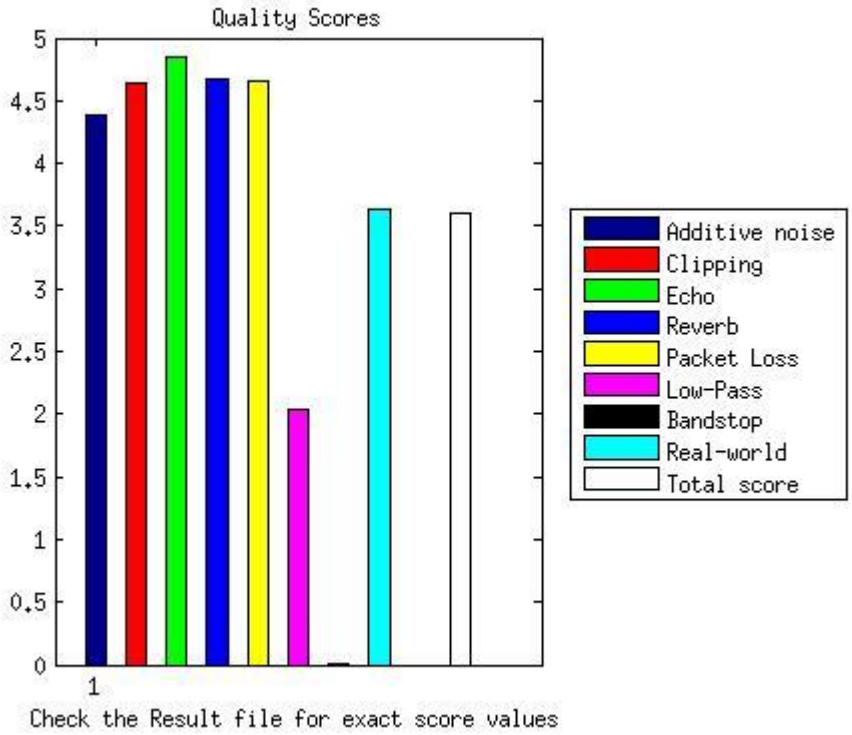

Figure 3: Example of the graph obtained from the program, each bar displaying the mean score given by that particular distortion weight

CHAPTER 5

5 Results

In this segment the results of the tests are presented. First the results of the tests for determining what weights are optimal for each type of distortion, and then the results when testing the software packet.

The results of the music tests where all files were used for calculating both regression and correlation:

Distortion type	**Methods combined**	**Highest score**
Linear	All	0.9102
Echo	llr, cep, wss, segsnr	0.6525
Reverb	All	0.8770
Clipping	All	0.8043
Low-pass filter	All	0.9691
Packet loss	All	0.8961
Bandstop filter	All	0.6101
Real-world recording	All	0.7350

Table 10: Music test results, all files used for regression

The results of the music tests where half the files were used for calculating regression and the other half for calculating correlation:

Distortion type	Methods combined	Highest score
Linear	fwseg, cep, wss, segsnr	0.9128
Echo	All	0.6227
Reverb	llr, cep, wss, segsnr	0.8225
Clipping	fwseg, llr, wss, segsnr	0.7114
Low-pass filter	cep, wss, segsnr	0.9535
Packet loss	fwseg, cep, wss, segsnr	0.8762
Bandstop filter	fwseg, cep, wss, segsnr	0.2545
Real-world recording	All	0.4862

Table 11: Music test result, half the files used for regression

The result of the music tests when the mean of the score with all files and the scores of the tests with half the files was calculated:

Distortion type	Methods combined	Highest mean score
Linear	All	0.8887
Echo	All	0.5250
Reverb	llr, cep, wss, segsnr	0.8364
Clipping	llr, cep, wss, segsnr	0.6895
Low-pass filter	cep, wss, segsnr	0.9451
Packet loss	fwseg, cep, wss, segsnr	0.7627
Bandstop filter	All	0.3240
Real-world recording	fwseg, cep, wss, segsnr	0.4583

Table 12: Music test results, highest mean correlation

The results of the speech tests where all files were used to calculate both regression and correlation:

Distortion type	Methods combined	Highest score
Linear	All	0.9083
Echo	All	0.8171

Reverb	All	0.8945
Clipping	All	0.7400
Low-pass filter	All	0.9220
Packet loss	All	0.7638
Bandstop filter	All	0.8480
Real-world recording	All	0.6534

Table 13: Speech test results, all files used for regression

The results of the speech tests where half the files was used to calculate regression and the other half was used to calculate correlation:

Distortion type	**Methods combined**	**Highest score**
Linear	All	0.9054
Echo	fwseg, wss, segsnr	0.7654
Reverb	fwseg, wss, segsnr	0.9097
Clipping	fwseg, llr, cep, wss	0.7680
Low-pass filter	All	0.9980
Packet loss	All	0.7845
Bandstop filter	All	0.8743
Real-world recording	cep, wss, segsnr	0.3914

Table 14: Speech test results, half the files used for regression

The result of the speech tests when the mean of the score with all files and the scores of the tests with half the files was calculated:

Distortion type	**Methods combined**	**Highest mean score**
Linear	All	0.9027
Echo	fwseg, wss, segsnr	0.6420
Reverb	fwseg, wss, segsnr	0.8966
Clipping	fwseg, cep, segsnr	0.7493
Low-pass filter	llr, cep, wss, segsnr	0.9400
Packet loss	fwseg, llr, cep, wss	0.7412

Bandstop filter	All	0.8203
Real-world recording	cep, wss segsnr	0.4222

Table 15: Speech test results, highest mean correlation

The result of testing the software packet where the scores obtained using the weights calculated for each distortion type and the mean of all the scores are correlated with PEAQ or PESQ.

Distortion type	**Mean score Speech**	**Mean score Music**	**Speech, correlation with PESQ**	**Music, correlation with PEAQ**
Linear	4.3044	4.645	0.5014	0.8691
Echo	4.7602	3.069	0.2931	0.4335
Reverb	4.6067	0.03375	0.4552	0.5222
Clipping	4.5882	0	0.0166	N/A
Low-pass filter	2.0535	1.1865	0.3700	0.8833
Packet loss	4.5994	5.0000	0.5623	N/A
Bandstop filter	0.001	3.343	N/A	0.9554
Real-world recording	3.576	4.371	0.5358	0.8692
Mean of all	3.5612	2.706	0.7179	0.9271

Table 16: Software packet test results

The scores for the respective distortion are the mean of all the scores for that particular distortion in all the tests. This is not a value of how much of this distortion is present in the test, simply a score of how the weightings optimized for that particular distortion performs. A high score does not mean that there is no distortion of that kind, and a low score does not mean that there is a lot of that distortion, it means that the particular weighting evaluates the channel to that score. The correlation score gives an idea of how well the scores for the different distortions correspond with the PEAQ or PESQ scores.

The score that is called “Mean of all” is the mean of the mean of all the distortion scores for every test, and this is the final grade of the tested channel. The correlation of that score gives an idea of how the mean score corresponds to PEAQ or PESQ.

Statistical values are presented below to help understand the results.

Distortion type	Mean Speech	Mean Music	Standard deviation Speech	Standard deviation Music	Confidence interval Speech	Confidence interval Music
Linear	4.3044	4.645	0.0589	0.0724	± 0.0289	± 0.0710
Echo	4.7602	3.069	0.0754	0.0430	± 0.0370	± 0.0422
Reverb	4.6067	0.03375	0.0587	0.0412	± 0.0288	± 0.0405
Clipping	4.5882	0	0.0474	0	± 0.0232	± 0
Low-pass	2.0535	1.1865	0.3201	0.9838	± 0.1568	± 0.9641
Packet loss	4.5994	5.0000	0.0460	0	± 0.0226	± 0
Bandstop	0.001	3.343	0	0.0315	± 0	± 0.0309
Real-world	3.576	4.371	0.2529	0.2701	± 0.1235	± 0.2648
Mean of all	3.5612	2.706	0.0678	0.0757	± 0.0332	± 0.0742
PESQ/PEAQ	2.9052	-3.6450	0.2389	0.1077	± 0.1171	± 0.1056

Table 17: Statistical values based on the test scores

CHAPTER 6

6 Analysis

6.1 Results of distortion tests

The distortion tests resulted in the information of which of the tested unlicensed methods and which respective weights gave the best correlation with the licensed methods. This information was then used to create the weightings in the software packet. Which methods that gave the highest correlation in three different cases is presented in chapter 5, tables 10-15. The weights with highest mean score was the ones that was finally used in the software packet.

When doing the real-world recordings, all files, both music and speech, were transmitted in all cases, both when using phones and speakers. When listening to the results afterwards, it was clear that music is not valid to use for testing phones. Many brands have a noise cancelling function that recognizes that the music is not speech and as fast as possible attenuates the undesired elements. How this was done differed from phone to phone. Some cancelled everything that was not human, and some let the music through when there was nothing else deemed more important. In the software packet, music should therefore only be used when testing speakers or headsets without a phone call involved.

6.2 Results of testing the software

The correlation score, seen in table 16, from testing the software shows that the program, when compared to commercial methods, can be said to be accurate to 92.71% when testing music and 71.79% when testing speech. This shows that it can be used to evaluate the sound quality of different channels, but that it should not be the only basis for a quality assessment. The tester should listen to the recording the program made and the channel to determine if the score seems reasonable. It is worth to note that the relatively low number of tests made with the software packet makes the correlation scores less reliable than if it would have been more extensively tested. The time it took to carry out the tests, and the fact that the authors could not obtain more Bluetooth products made further testing not possible.

The software packet gives out a score between 0 and 5, where 5 means that there is no difference between the original and the degraded sound files, and where 0 means that they are completely different. The tests show that if a score below 3 is obtained for speech and below 2 is obtained for music, it indicates that the channel should be investigated further to find out why the score was low. However, the recording of the file is an important part of the testing, and if it is not done properly it could add a lot of distortions to the test, thus corrupting it.

The scores obtained from the weights calculated for clipping and packet loss for music and from bandstop for speech did not result in a correlation value. This is because the scores were identical for every test. They were still kept in the program, however, because of the chance that they might result in different values at some point, depending on the channel. Since every score was identical, they did not play a part in the calculation of the correlation, but still affected the mean score.

The scores obtained from the weights calculated for reverb when testing music was very low, almost zero, so a test was made to see if a mean score when reverb was excluded gave a higher correlation. This was not the case, so the reverb score was kept in the program.
The correlation score for clipping when testing speech was very low, so a test was made to see if the correlation of the mean score would be higher with the clipping score removed. It was not, so the clipping score was kept in the program.

As previously stated, some phones have noise cancelling algorithms. This could in some cases be a problem for the test. The authors found that one of the phones used during testing sometimes managed to cancel out the pilots. This caused the separation of the files to fail, and the test had to be redone.

Most phones used the first ten seconds or so of the recording to find a good volume for the call, and this volume turned out to be different for different phones, so each test with a new phone had to be calibrated before recording to make the levels good. For some phones, an amplitude of the pilots where they were about as loud as the speech was too low, so another file with louder pilots had to be used. For other phones, the pilots could not be louder than the speech, since they got clipped in the recording if they were, and the frequency turned out wrong.

6.3 Future work

To make a more accurate assessment of the sound quality, this software packet could be upgraded in a few ways. More types of possible distortions could be tested and integrated into the software. The tests for packetloss could also be implemented in a manner that more closely represent what really happens.

The distortions could also be weighted according to their importance to the final result, just as the scores of the methods are. In this packet the types of distortions are valued equally with the final score being a mean of the weighted scores. But the actual case is probably that some method and weight combinations give a more accurate result than others, since some of the distortions may not be present. To get the most correct quality grade answer the best would probably be to gather test results, final score from this solution and PESQ or PEAQ score respectively, and then calculate a new linear combination weighting from these. Though this would again give the problem of having to correspond the output to an actual grade. There is a solution presented above with vectors but it would also be a good improvement for the software packet if another way was found to limit the final scores to the interval zero to five.

The pilots for separating the sound files works in most cases, but it is not a perfect solution, since noise cancellers in phones can manage to attenuate them, unwanted noises from the environment can create false pilots, and they can become clipped. The authors can, however, not find any way of making a more reliable method for separating the files. A pilot containing a frequency spectrum was considered, where the program looks for several frequencies in each pilot, and cuts the file if at least one is found. The risk is, however, that noises from the environment will be a greater problem. Some more research could be made to find a solution that is better than the one currently used.

To make sure that the score given by the suggested solution is acceptable it would be a good idea to make subjective tests with a large group of people and then correlate their answers to the methods final score.

CHAPTER 7

7 Conclusion

A quality assessment of sound can be made with unlicensed methods, but they are not perceptually motivated, and therefore cannot be expected to determine how a human would rate the sound in question as well as the licensed methods. The software packet is modelled to follow the rating of the licensed methods PEAQ and PESQ and test show that they do so with a 92.71% correlation when testing music and 71.79% when testing speech. The program presented in this paper gives an idea of how well a channel performs, but factors such as the recording and environmental noise can corrupt the test and therefore a subjective assessment made by the tester is advised to verify that the score is reasonable.

References

[1] How does it sound? - Paul Denisowski, IEEE Spectrum, February 2001

[2] Concepts behind sound quality: Some basic considerations - Jens Blauert, Ute Jekosch, The 32nd International Congress and Exposition on Noise Control Engineering, Seogwipo, Korea, August 2003

[3] Comprehensive modeling of the formation process of sound quality - A Raake, J Blauert, 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX), IEEE

[4] Speech quality assessment - Philipos C. Loizou, University of Texas-Dallas, Department of Electrical Engineering, Richardson, TX, USA, Springer-Verlag Berlin Heidelberg 2011

[5] Wireless Communications, second edition, Andreas F. Molisch, John Wiley & Sons Ltd. 2011

[6] Quantifying the Suitability of Reference Signals for the PESQ Algorithm - Stefan Paulsen, Tadeus Uhl, 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service, IEEE

[7] Perceptual evaluation of speech quality(PESQ) - a new method for speech quality assessment of telephone networks and codecs - Antony W. Rix, John G. Beerends, Michael P. Hollier, Andries P. Hekstra, IEEE, 2001

[8] Perceptual Evaluation of Speech Quality (PESQ), the new ITU standard for end-to-end speech quality assessment. Part II – Psychoacoustic model - . G. Beerends, A. P. Hekstra, Ae. W. Rix, and M. P. Hollier, JAES Volume 50 Issue 10 pp. 765-778; October 2002

[9] Perceptual Evaluation of Speech Quality (PESQ), the new ITU standard for end-to-end speech quality assessment. Part I – Time alignment - Antony W. Rix, Michael P. Hollier, Andries P. Hekstra, and John G. Beerends, JAES Volume 50 Issue 10 pp. 755-764; October 2002

[10] ITU-T recommendation P.862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs (02, 2001)

[11] An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality - P.Kabal, McGill University May, 2002

[12] PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality - Thilo Thiede, William C. Treurniet, Roland Bitto, Christian Schmider, Thomas Sporer, John G. Beerends, Catherine Colomes, Michael Keyhl, Gerhard Stoll, Karlheinz Brandenburg, Bernhard Feiten, J. AudioEng. Soc., Vol. 48, No. 1/2, 2000 January/February

[13] ITU-R recommendation BS.1387-1, Method for objective measurements of perceived audio quality, 2001

[14] A perceptual audio quality measure based on a psychoacoustic sound representation - Beerends J. G., Stemerdink J. A. : J. Audio Eng. Soc., Vol. 40, No. 12, pp. 963-987, 1992

[15] Estimating Perceptual Audio System Quality Using PEAQ Algorithm - Marija Šalovarda, Ivan Bolkovac, Hrvoje Domitrovic, Applied Electromagnetics and Communications, 2005. ICECom 2005. 18th International Conference on.

[16] Bark and ERB Bilinear Transforms - Julius O. Smith III, Jonathan S. Abel, IEEE Transactions on Speech and Audio Processing, November, 1999

[17] A novel objective method for evaluating the quality of streaming audio - Yang Yue, Xie Xiang, Wei Yaodu, IEEE 2009

[18] PEMO-Q—A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception - Rainer Huber, Birger Kollmeier, IEEE Transactions on Audio, Speech, and Language Processing, VOL. 14, NO. 6, November 2006

[19] Speech Enhancement, Theory and Practice, second edition - Philipos C. Loizou, CRC Press Taylor & Francis Group, 2013

[20] Objective Measures of Speech Quality – Shuyler R. Quackenbush, Thomas p. Barnwell III, Mark A. Clements, Prentice-Hall Inc. 1988

[21] Recommendation ITU-R BS.1116-2, Methods for the subjective assessment of small impairments in audio systems, June 2014

[22] Recommendation ITU-R BS.1284-1, General methods for the subjective assessment of sound quality, 2003

[23] PEAQ software package, Kabal, McGill university, Telecommunications & Signal Processing Laboratory, 2004

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2015-449

http://www.eit.lth.se