Quality scalability of a parametric audio coder

Schuijers, E.G.P.

Award date: 2000
Quality Scalability of a Parametric Audio Coder

Graduation Project

Author: Erik Schuijers
ID Nr.: 413692

Performed at:
- Sound Coding Group
- Advanced Systems and Applications Laboratory Eindhoven
- Philips Consumer Electronics

Research Chair: Signal Processing Systems
Section: Measurement and Control Systems
Faculty: Department of Electrical Engineering
University: Eindhoven University of Technology (EUT)

Supervisors ASALE: Ir. A.W.J. Oomen
January 2000 – December 2000

Supervisor EUT: Ir. J.H.F. Ritzerfeld

Professor: Prof.Dr.Ir. J.W.M. Bergmans

From: March 2000 – December 2000
TABLE OF CONTENTS

ABSTRACT ................................................. 4

INTRODUCTION ........................................ 5

1 AUDIO CODING ....................................... 6
  1.1 INTRODUCTION .................................. 6
  1.2 AUDIO CODING IN GENERAL .................. 6
  1.3 WAVEFORM CODING ............................ 7
  1.4 PARAMETRIC CODING ......................... 8

2 SINUSOIDAL CODING .................................. 9
  2.1 INTRODUCTION .................................. 9
  2.2 SSC PROJECT GOALS .......................... 9
  2.3 SSC DESCRIPTION ............................ 10
  2.4 TRANSIENTS .................................. 12
    2.4.1 TRANSIENT ANALYSIS .................. 13
    2.4.2 TRANSIENT SYNTHESIS ............... 16
    2.4.3 TRANSIENT CODING ................... 16
  2.5 SINUSOIDS .................................. 16
    2.5.1 SINUSOIDAL ANALYSIS ............... 18
    2.5.2 PSYCHO-AcouSTIC MODEL .......... 20
    2.5.3 TRACKING ................................ 22
    2.5.4 CODING OF SINUSOIDAL COMPONENTS .. 24
    2.5.5 SINUSOIDAL SYNTHESIS ............ 25
  2.6 NOISE ....................................... 27
    2.6.1 NOISE ANALYSIS ........................ 27
    2.6.2 NOISE SYNTHESIS ..................... 29
    2.6.3 NOISE QUANTISATION ................... 30
  2.7 QUANTISATION AND CODING ................. 30

3 QUALITY ASSESSMENT EXPERIMENTS .......... 33
  3.1 INTRODUCTION ................................ 33
  3.2 GENERAL QUALITY ASSESSMENT ............ 33
  3.3 FREQUENCY SELECTION BASED ON PSYCHO-AcouSTICS 34
  3.4 TWO-STEP EXTRACTION ....................... 37

4 IMPROVED EXTRACTION OF SINUSOIDS .... 38
  4.1 INTRODUCTION ................................ 38
  4.2 AMPLITUDE AND FREQUENCY VARIATION ... 38
  4.3 LINEAR REGRESSION .......................... 43
  4.4 MULTI-SCALE SINUSOIDAL EXTRACTION ...... 45
    4.4.1 FILTER COMPENSATION .............. 46
    4.4.2 SCALE TRANSITIONS .................. 47
  4.5 EXPERIMENTS ................................ 48
  4.6 ALTERNATIVE FREQUENCY ESTIMATION .... 49

5 PSYCHO-AcouSTIC MODEL ......................... 53
ABSTRACT

In this report the work performed on SinuSoidal Coding (SSC) during the graduation project is presented. SSC is a parametric audio coding technique, developed at Philips Research, aimed at a bit-rate of approximately 40 kbit/s for high quality stereo audio. Currently however, only monaural signals can be processed. The SSC coder divides audio into three objects: transients, sinusoids and noise. For each object relevant parameters are extracted and efficiently encoded into a bit-stream.

In the current implementation from a certain quality on, an increase in bit-rate will not lead to an improvement of quality. In other words, the quality saturates. In order to assess the cause of this problem, experiments have been performed. These experiments showed that especially the sinusoidal module causes the perceived quality loss with respect to the original signal. Within the sinusoidal module 3 main causes could be identified: 1. The extraction lacks the ability to model amplitude- and frequency-varying sinusoids. 2. In the tracking mechanism, a mechanism that links individual sinusoids in time, incorrect links are made. 3. During the removal of the non-tracks, sinusoids that have not been linked, too many sinusoids are discarded.

In order to improve the sinusoidal extraction a modulated 2nd order polynomial description of a sinusoid is proposed and has been added to the SSC reference encoder. It is shown that this description, to a certain extent, can model both amplitude and frequency variations.

The psycho-acoustic model in SSC, a model that describes the perceptual relevance of sinusoids, was based on a description of constant sinusoids. It is shown how this model is adapted to the 2nd order polynomial description.

One of the most important steps for the compression is the use of the tracking algorithm. The tracks formed by this algorithm can be encoded very efficiently using differential encoding. Furthermore, phaseless reconstruction can be applied to further improve coding gain. As this is not a lossless process, the course of the tracks actually determines the quality. It is shown how, using the information of the 2nd order polynomials, the linking of individual sinusoids is improved.

Furthermore, in the algorithm that removes the non-tracks the psycho-acoustic model is applied to further increase the audio quality. A formal listening test showed that indeed the perceived audio quality is increased, however at the cost of about 4 kbit/s extra.

Finally, it can be concluded that in order to bridge the gap to transparent audio quality the following issues should be considered: improvement of the transient representation, improvement of the phaseless reconstruction and improvement of quantisation.
INTRODUCTION

Audio coding is the process of digitally encoding/decoding audio signals. The goal of audio coding is to obtain a representation that is as compact as possible, while after decoding, quality should be as close as possible to the original signal.

Mainly thanks to the Internet, audio coding is also publicly known, in particular the MPEG-1 Layer III standard, better known as MP3. This compression scheme delivers high quality audio at compression gains of over a factor of 10. Since the standardisation of MPEG-1 many new and innovative ideas have been brought forward. In practice, this led to state-of-the-art MPEG4-AAC coders that provide compression-factors of around 15 while still maintaining the same high quality level as with MPEG-1 Layer III. The current opinion in the audio coding community is that no further improvement in compression gains is expected for waveform type of coders. There is a general belief that, in order to achieve even higher compression gains, audio should be coded parametrically.

In 1998 Philips Research started a feasibility study to parametric audio coding of general broadband audio. To parameterise the audio it was divided into three objects: transients, sinusoids and noise. This feasibility study indicated that, using these three objects, high quality audio coding should be possible at bit-rates of around 40kbit/s for stereo signals, i.e. a compression-gain of about $44,100 \cdot 2\cdot 16 / 40,000 \approx 35!$ This bit-rate is chosen such that an hour of music can be encoded into 16 MB of ram.

In the following few years an implementation has been developed, mainly for mono-signal at a goal of around 24 kbit/s. This implementation however doesn't deliver the desired high quality at the intended bit-rate. Furthermore, from a certain point on the spending of extra bits will not give an extra increase in quality.

The goal of the graduation project, described in this report, is first of all to assess what exactly causes this lack of quality. Secondly, for the problems that are recognised solutions within the current framework are investigated and evaluated.

In the first chapter a general introduction to audio coding is given. In the next chapter SSC, the parametric audio coder developed by Philips Research, is described. Next, the experiments are described that indicate where the current bottlenecks concerning the quality are. In the following chapters an extension to the sinusoidal model is given and it is shown how some other blocks of the sinusoidal module can be improved using the extended model. Thereafter performance results, both in quality and in bit-rate, are described. Finally we end with conclusions and recommendations.
1 AUDIO CODING

1.1 INTRODUCTION

Audio coding is the process of encoding/decoding digitised audio signals. This is done in such a manner that after encoding the data rate is kept as low as possible while maintaining as much as possible of the original quality after decoding. In this chapter a brief introduction to audio coding is given.

1.2 AUDIO CODING IN GENERAL

When referring to the term 'audio', this is meant in a very broad sense. Basically everything that can be perceived by the human auditory system can be described by the term 'audio'. So this includes music, speech but also non-coherent sounds. The only restriction to the term 'audio' in audio coding is the fact that this audio must be (made) available in a digitised form, preferably in Pulse Code Modulated (PCM) format. The most common form is the CD-standard 16-bits PCM sampled at 44.1kHz in stereo. A basic diagram of an audio encoder/decoder is given in Figure 1.1.

![Figure 1.1: General audio encoding/decoding process](image)

As can be seen in Figure 1.1, the digital PCM waveform is encoded into a so-called bit-stream. The amount of compression, also indicated as coding gain, compared to the original PCM waveform data can be derived from the bit-rate of the bit-stream. The bit-rate indicates the average amount of bits per time-segment (usually seconds) needed to decode one such time-segment of the signal.

The main goal of audio coding can be described in two ways: achieve the highest possible perceptual audio quality for a given bit-rate or achieve the lowest possible bit-rate for a given perceptual audio quality level. Although these descriptions seem straightforward it's hard to give a quantitative measure of perceptual audio quality because of subjectivity.

There are basically two steps in audio encoding. The first step consists of the removal of irrelevancy. From the original signal only the relevant parts, the parts that can be perceived, are kept for further encoding. This implies that removal of irrelevancy is not lossless. The second step consists of the removal of redundancy. This is, in contrary to the removal of irrelevancy, a lossless process. Coding schemes like e.g. Huffman or Tunstall coding exploit the redundant information in order to come to a more efficient representation.

There are many applications in which audio coding can be used advantageously. Some examples are mobile telephony, Internet radio, solid-state audio players, etc. Basically everywhere where digital audio has to be stored, in case of storage restrictions, or has to be transmitted, in case of channel restrictions, audio coding can be used to a certain advantage.

Of course audio coding also has some drawbacks. These drawbacks are mostly described in terms of some important properties:
• Audio quality; the perceived audio quality with respect to the original signal.
• Complexity; the amount of processing power/time needed for encoding or decoding,
• Delay; the amount of delay, usually expressed in samples, caused by the encoder/decoder system.
• Bitrate; the average amount of bits needed to decode a time-unit of the signal, usually expressed in kilobit per second.

Furthermore the cascading of audio coding systems will further degrade the audio quality.

Not every application has the same requirements, for e.g. mobile telephony it is very important that real-time encoding with very low encoding/decoding delay is possible.

1.3 WAVEFORM CODING

One of the more classic ways of audio coding is referred to as waveform or transform coding. A schematic description of waveform coding is given in Figure 1.2.

![Waveform Coding Diagram]

Figure 1.2: Waveform coding

A waveform coder is characterised by the transformation stage. First the PCM input signal gets transformed to the frequency domain by means of a filterbank like e.g. an FFT, a polyphase filterbank or a Modified Discrete Cosine Transform. The PCM input signal is also fed to a perceptual model. This model determines which frequency components can be perceived and which can not by means of a so-called masking model. This information, together with the transformed PCM data is fed to the quantisation module. In this module the frequency components are quantised, which inevitably results in quantisation noise. Inside this module the best possible quantisation for each frequency component is determined given the masking information from the perceptual model and the amount of bits that may be spent. Finally the quantised values are efficiently coded by removing the redundancy that’s still present in the quantised frequency components.

A lot of research on waveform coding has already been done. This makes the theory behind such coders well established. However progress in the amount of data-rate reduction achieved by waveform coders almost seems to be saturated. In order to come to an even lower bit-rate for the same high quality audio, it seems sensible to look for another description of audio.
1.4 PARAMETRIC CODING

When the input signals are restricted (e.g. to speech only) specific features of the input signal can be exploited to further improve the coding gain. One way of implementing these features in an audio coder is to use a parametric coding scheme. The main difference with a waveform coder is the explicit usage of a source model. For e.g. a speech model this is based on the human vocal tract. For a parametric model of a piano the hammers, the snares and the cabinet could be considered. Figure 1.3 depicts a schematic description of a parametric audio coder.

The main advantage of parametric audio coding is the exploitation of both the sender-end (source of sound) as well as the receiver-end (human auditory system), where the waveform coder exploits the receiver-end only. If the input signal however doesn't appropriately fit the source model this might lead to unpredictable results. This lack of robustness then again is the main disadvantage of parametric audio coding.

Another advantage of parametric audio coding is the perceptual model. This model can be adapted fully to the source model. If e.g. one object of the source model is defined as a set of harmonics, perceptual experiments using harmonics could be performed in order to come to a psycho-acoustic model for harmonics only.

Parametric models are also often called object-oriented models. Objects can be a bit abstract like for speech: 'tonal elements' and 'non-tonal elements' but also concrete like 'snare generated harmonics' of a guitar model. A description in terms of different objects automatically implies that within the encoder, decisions will have to be made about what part of the signal must be assigned to what object. The more such decisions have to be made the worse the robustness will probably be.

All in all parametric coding has more potential than waveform coding as far as bit-rate versus perceived audio quality is concerned. But, especially when designing a parametric model for a broader type of input signals, one has to consider robustness versus coding efficiency.
2 SINUSOIDAL CODING

2.1 INTRODUCTION

In 1998 Philips Research started the 'Sinusoidal Audio Coding' project in collaboration with IPO, Delft Technical University and the Royal Institute of Technology. This project aims at developing a low bit-rate audio codec that gives high quality output at bit-rates substantially lower than defined in the MPEG1 and MPEG2 standard. Within Philips, this project is also referred to as 'Sinusoidal Coding of Audio and Speech' (SiCAS) or 'Sinusoidal Coding' (SSC). In this chapter a description of SSC is given with the status at the beginning of the graduation project.

2.2 SSC PROJECT GOALS

In the first stage of the SSC project some qualitative goals, requirements and conditions have been determined [den Brinker and Oomen 1998]. Quantitative goals are hard to give because there exists no reliable method for measuring perceived audio quality. The main goals, requirements and conditions are:

- Input: bandwidths from 4kHz for narrow band speech to 24kHz for high quality audio.
- Quality: multiple quality levels, from medium quality to comparable to MPEG1 Layer III (mp3) at 128 kbit/s to transparent quality.
- Bit-rates: multiple bit-rates ranging from 2 kbit/s for intelligible speech up to 40 kbit/s for stereo CD-like quality audio.
- Bit-rate scalability: the bit-stream can be seen as a layered collection of streams each representing a quality improvement over the other.
- Low delay: useful for point-to-point streaming applications.
- Graceful degradation: quality must gradually decrease with decreasing bit-rate.

Later on in the project, after the first generation encoders and decoders were produced, one of the main problems with SSC proved to be the quality scalability. From a certain bit-rate on the quality doesn't increase anymore. This is shown graphically in Figure 2.1.

![Graph of Quality Scalability of the SSC Coder](https://example.com/graph.png)

Figure 2.1: Quality scalability of the SSC coder
The goal of the graduation project described in this report is to address this problem. More specifically the goal is to identify and analyse the current problems and investigate possible solutions.

At the start of the graduation project it was only possible to encode mono PCM signals sampled at 44.1kHz. The quality of the encoded material was however mediocre. These conditions have been taken as the point of departure for the graduation project. Furthermore all the encoder and decoder source code was written using Matlab.

2.3 SSC DESCRIPTION

The SSC coder is a parametric audio coder that aims at coding of audio in a broad sense, i.e. unrestricted to a classified source like e.g. speech. Therefore a source model must be developed that is both simple and effective. To do so, one first has to identify the different auditory occurrences that are present within audio signals. To illustrate these occurrences Figure 2.2 and Figure 2.3 show spectrograms of respectively a harpsichord signal and a castanets signal. A spectrogram is basically a time-frequency plot with on the x-axis the time and in the y-axis the frequency. In both Figures a lighter gray value indicates a higher signal power.

When looking at both figures one can easily recognise three phenomena that are quite common for audio signals in general.

1. Horizontal lines: these appear to be deterministic by nature. In Figure 2.2 these lines represent individual harmonic lines generated by the strings of a harpsichord. They are mostly characterised by their instantaneous frequency.

2. Vertical lines: these also appear to be deterministic by nature but are characterised mostly by their placement in time. Such lines indicate transient phenomena like attacks and steps in the audio signal. Figure 2.3 and to a lesser degree Figure 2.2 clearly show such phenomena.

3. Noise-like areas: there also seem to be areas that don't have a specific structure but are more stochastic by nature. Figure 2.3 clearly shows such areas. These areas are characterised as being noise-like.

Unknowningly, we have now classified audio into three objects, namely sinusoids, transients and noise. Such a description seems to be both simple as well as complete.

Another reason for choosing these objects as the base for a parametric audio model is the apparent relation to psycho-acoustics. When e.g. calculating the masking curve in most perceptual models of waveform coders a distinction is made between tonal elements (sinusoids) and non-tonal elements (noise). The masking effect of sinusoidal tones is different from the masking effect of noise.

Also, when looking at state-of-the-art audio codecs (encoder/decoder) like AAC or MPEG-1 Layer III (MP3) [Brandenburg and Stoll 1992, Pan 1995] a distinction between transients and non-transients is being made by means of window switching. Window switching can lead to a locally increased time-resolution. This is needed for describing transient-phenomena because of the characterisation in the time-domain.

Furthermore, most experiments that are performed in the field of psycho-acoustics are done using simple tones (sinusoids) and narrow-band or broad-band noise. Basically psycho-acoustic models can be fit quite well to a description using those objects.

A third and final reason for choosing the objects is the assumption that such a description of audio will be most effective.
The SSC codec is based on the three objects described above, namely: transients, sinusoids and noise. A block-diagram of the SSC-encoder is shown in Figure 2.4.
One of the most crucial steps in parametric audio coding is the subdivision of different parts of the signal to different objects. For both the analysis of sinusoidal components as well as for analysing noise, quasi-stationarity is a prerequisite. So if transient phenomena could be removed first the residual signal will be more stationary and thus easier to analyse. This is the main reason why the transients module (T) has been placed in front of the other two. The sinusoidal module (S) is placed before the noise module (N) because it is much harder to analyse and remove the noise from the residual signal of the transient module than it is the other way around.

2.4 TRANSIENTS

It’s hard to give a definition of transients when referring to audio signals. However, a transient mostly has one (or both) of the following properties:

- The signal characteristics just before and just after the transient are different. In this way a transient can be seen as a transition between auditory events.
- The transient itself consists of a short burst of signal energy.

A large class of signals falls under the above properties, e.g. attacks, steps, clicks, snaps, etc. This already indicates that a common model for transients is hard to give. However, within the SSC project an attempt has been made to provide such a model.

The block-diagram of the transient module is described in Figure 2.5. It consists of three blocks: transient analysis (TA), transient synthesis (TS) and transient quantisation (TQ).

In the transient analysis block, transients are detected and analysed into a set of parameters. In the transient synthesis block the parameters that have been found by the transient analysis block are used to synthesise the transients. The signal generated by the transient synthesis block is subtracted from the original signal to form a residual signal, which is fed to the sinusoidal module.

Finally the transient quantisation block performs both coding and quantisation of the parameters that have been extracted by the transient analysis block.
The transient analysis module also performs a first stage of segmentation of the input signal. In Figure 2.6 a PCM waveform is shown that clearly contains transients. The transient analysis module detects these transients and divides the original PCM waveform into smaller segments that are processed more or less individually at a later stage. This segmentation is in correspondence with the first transient property, namely that before and after the start of the transient characteristics of the signal can be different. It is therefore more efficient to divide the original signal into segments that are more or less quasi-stationary and code each of these segments individually.

![Figure 2.6: First stage of segmentation for PCM waveform of castanets](image)

### 2.4.1 Transient analysis

Because of the analysis-by-synthesis structure of the SSC encoder it is important that transient positions and parameters are extracted properly. The segmentation and stationarity of the residual signal are of great importance for the performance of the sinusoidal and noise module behind the transient module. A block diagram of the transient analysis block is given in Figure 2.7:

![Figure 2.7: Block diagram of transient analysis block](image)

First of all the position and type of the transients is determined from the PCM input signal. A choice is made between either a step-transient or a Meixner-transient [den Brinker and Oomen 1998], a transient of which the envelope is described by a Meixner function. A step-transient is described solely by its position. This type of transient corresponds to the first transient property of being a transition between two more or less quasi-stationary segments.
The Meixner-transient corresponds mostly to the second property of being a short signal-energy burst. This type of transient is therefore not only characterised by its position but also by a few parameters to describe the short burst.

The determination of positions and types of transients is described in Figure 2.8.

![Figure 2.8: Block diagram of determination of position and type](image)

For both the original PCM signal as well as a high-pass filtered version of the original signal an envelope is calculated. This envelope is calculated as (Equation 2.1):

\[ e[n] = \max\{x[n], \tau e[n-1]\}, \]

where \( e[n] \) describes the envelope function and \( x[n] \) the input signal, both as a function of time \( n \), and \( \tau \) is a time-constant between zero and one. As an example the envelope for the castanets excerpt from Figure 2.6 is given (see Figure 2.9).

![Figure 2.9: Envelope for castanets excerpt (black line)](image)

In this envelope function jumps are sought that are greater than a certain predefined threshold value. The positions of these jumps are indicated as candidate transient positions.

This same process is also applied for the high-pass version of the original signal because of signal energy that may be present in the lower frequency areas. Low frequency components that are perceptually relevant in practice typically have high amplitude. Calculating
the envelope of signals that have much signal energy at the lower frequencies can easily mask transients present in the higher frequency area.

The candidate positions of both the original and the filtered signal are then used to determine the exact positions as well as a preliminary type-description of the transients. Transient positions that have only been found in the filtered version are always characterised as being a step-transient at this point. For all other positions the type has still to be determined.

Whether a transient can be classified as being either a Meixner-transient or a step-transient is determined first of all by the possibility to fit a Meixner-envelope on the input signal at the transient-position. If such a fit can be made to the input signal a transient is preliminary classified as being a Meixner-transient. A typical example of a Meixner-envelope is given in Figure 2.10.

A description of the Meixner-fitting process is given in Figure 2.11.

If the envelope can be fit to the data-segment this segment is amplified with the inverse of the envelope in order to make the transient as stationary as possible. By means of an FFT and interpolation techniques the frequencies of the highest peaks of the frequency spectrum are estimated. Finally for those frequencies found a sinusoidal fit is being made. If the amount of energy reduction achieved with such a fit however is below a certain threshold the type is still finally set to a step-transient.
2.4.2 Transient synthesis

The synthesis of the transients is the inverse process of the analysis section. For step-transients no signal has to be generated. For Meixner transients first of all the sinusoids are regenerated from the extracted parameters and summed together. Thereafter, the Meixner-envelope gets regenerated and multiplied with the summed sinusoids.

2.4.3 Transient coding

All in all the following data is extracted per transient:
1. Transient position
2. Transient type
3. Meixner envelope parameters (2 parameters, if applicable)
4. Sinusoidal parameters: frequency, amplitude and phase (maximally 8 sinusoids, if applicable)

The sinusoidal parameters are coded differentially where possible in order to decrease bit-rate. In order to do so first the frequencies, amplitudes and phases of the sinusoids are converted to representation levels. These representation levels represent quantised values of the input variables. For frequencies, those quantisation levels are related to the ERB-scale. This is a scale that closely matches the way the human auditory system perceives frequency. This is explained in more detail in paragraph 2.5.2. Amplitudes are quantised logarithmically which also corresponds to the sensitivity in the auditory system. Finally, the phase values are quantised uniformly to five bits, an experimentally determined value. When all representation levels are determined, the sinusoids are sorted from low to high frequency. For the first sinusoid the absolute values of the representation levels are coded in the bit-stream. For all following sinusoids the amplitude, frequency and phase representation levels are coded differentially with respect to the previous sinusoid in the bit-stream.

2.5 SINUSOIDS

The SSC project is based on the assumption that any digital audio signal can be described adequately by (Equation 2.2):

\[ x[n] = \sum \text{transients} + \sum \text{sinusoids} + \sum \text{noise}. \]  
\[ \text{2.2} \]

It is however not clearly defined what a sinusoid is. Within the SSC project elements that are classified as being a sinusoid can be described by (Equation 2.3):

\[ s[n] = \sum_{p} A_p(n) \cos(\omega_p(n)n + \varphi_p), \]  
\[ \text{2.3} \]

where \( A_p(n) \) is the slowly varying amplitude, \( \omega_p(n) \) is the slowly varying frequency and \( \varphi_p \) the phase of the \( p \)th sinusoid. This representation was first used by McAulay and Quatieri for describing speech signals [McAulay and Quatieri 1986].

It is neither efficient nor feasible because of complexity to extract \( A_p(n) \), \( \omega_p(n) \) and \( \varphi_p \) on a sample-by-sample basis. A more feasible method would be to extract these parameters on a frame-to-frame basis. So for a single frame the sinusoids could be described by:
\[ s_k[n] = \sum_p A_{p,k} \cos(\omega_{p,k} n + \varphi_{p,k}) \]

where \( k \) indexes the frame and \( p \) the \( p^{th} \) sinusoid.

In order to come to such a description another segmentation takes place on the PCM input signal. At this segmentation stage the space between two transient-positions is divided into overlapping frames of 720 samples. This number has been determined experimentally as a balance between stationarity and efficient coding of parameters. The segmentation is illustrated in Figure 2.11. The upper line describes the transient positions extracted by the transients module. The small blocks at the lower end show how the sinusoids are segmented between these transient positions. It is noted that the last frame of a segment is always placed in such a way that it ends just before another segment starts. This is also because of stationarity as described before by the first transient property.

![Segmentation of transients and sinusoidal module](image)

Figure 2.11: Segmentation of transients and sinusoidal module

The sinusoidal module can now be described as in Figure 2.12. It first of all consists of a sinusoidal analysis block (SA). In this block the sinusoidal parameters are estimated on a frame-to-frame basis. The sinusoidal synthesis block (SS) synthesises the sinusoidal signal from these parameters. Finally this signal is subtracted from the residual signal of the transient module to create a presumably noisy signal for the noise module.

![Block-diagram of sinusoidal module](image)

Figure 2.12: Block-diagram of sinusoidal module
Of course not all sinusoidal components that have been extracted are perceptually relevant. Inclusion of such components in the bit-stream is superfluous. Therefore a psycho-acoustic model (PA) is used to remove sinusoidal components that fall well below the masking threshold. The reason that the psycho-acoustic model is not used very tightly lies within the tracking block (TK).

To come to an efficient representation of all the individual sinusoidal components found over all the analysed frames a tracking algorithm (TK) is used. The main idea behind this algorithm is that sinusoids in general last longer than only a single frame and can thus form tracks. Differential encoding of e.g. amplitude and frequency can then prove to be efficient. This differential coding is the reason that the restraints of the psycho-acoustic model are set loosely. It is more efficient to code a long track differentially, even though it can't be perceived during the whole length of the track, than only encode the short relevant parts.

Even more coding gain can be achieved by applying phaseless reconstruction. This means that instead of updating the phase of a track at each frame only the phase of the birth of a track is encoded. For all following frames of the track the phase is calculated based on the presumption that the instantaneous frequency is a smooth and slowly-varying function in time.

Finally the sinusoidal quantisation (SQ) block codes the processed parameters to further increase coding gain. It does so by quantisation and coding of the parameters extracted for frequency, amplitude and phase.

2.5.1 Sinusoidal Analysis

The sinusoidal analysis block consists of an iterative algorithm for extracting the sinusoidal parameters [den Brinker and Oomen 1999]. The block diagram is depicted in Figure 2.13.

During the first iteration a segment of data (frame) is presented to the sinusoidal extraction block. The FFT is determined after which the maximum amplitude of the FFT is sought. A rectangular window is used because such a window has the smallest mainlobe width. However the main disadvantage of a rectangular window is the sidelobe attenuation which causes spectral smearing. It is because of this smearing that the extraction process has to be done in an iterative way.

For the maximum found in the FFT a fine search by means of interpolation is made in order to precisely extract the frequency of the underlying sinusoid. When the frequency has been determined the optimal amplitude and phase can be determined by use of linear regression. Finally the sinusoid is generated using the extracted parameters and subtracted from the original segment. This process is repeated so that finally fifty frequencies with accompanying amplitude and phase are extracted per frame. It is noted that this method of extraction doesn't consider whether a spectral peak is actually the result of a sinusoid.
The use of only a short segment length of 720 samples implies that the lower frequencies can't be estimated with high precision. Therefore a multi-scale sinusoidal extraction mechanism has been developed (see Fig. 2.15) [den Brinker and Oomen 1999]. The basic principle of this mechanism is as follows.

First the PCM input signal is fed to an anti-aliasing filter (AAF) after which the signal is downsamped by a factor three (DS3). Now the structure of Figure 2.13 is applied to 720 samples in the downsamped domain (SE). These samples correspond to three times 720 samples in the original domain. An appropriate segment of downsamped PCM samples that corresponds to the samples in the original domain will therefore have to be selected. This is done in the sinusoidal segmentation unit (SU). This segmentation is shown graphically for all three scales in Figure 2.14. Note that every segment on the third scale corresponds to multiple segments on the second scale. Likewise every segment on the second scale corresponds to multiple segments on the first scale. Also note that the segments on the second and third scale are always placed within two transient positions just like the segments (frames) on the first scale.

On the third scale three sinusoids are extracted, on the second scale seven sinusoids and on the first scale forty. When less than three sinusoids are found in the third scale, the second scale may extract seven sinusoids plus what has been left by the third scale. The same applies to the second scale and the first scale.

![Multi-scale segmentation](image-url)
Because of complexity reasons in both encoder and decoder a description on a single scale is preferable. To come to a single-scale description the estimated parameters must be converted back to the original non-downsampled domain. This is done in a few steps. First of all every scale is band-limited and thus only a limited amount of frequencies may be included. This selection is done in the frequency selection module (FS). For sinusoids that are kept, compensation in amplitude and phase is made for the gain and delay caused by the anti-aliasing filter (FC). Finally the sinusoidal components are converted to the non-downsampled domain by mapping the parameters to the segments (frames) of the previous scale (ST).

2.5.2 Psycho-acoustic model

The key element in removing irrelevancy in audio coding is the psycho-acoustic model. This model tries to describe, given a certain input-signal what parts of that signal can and cannot be perceived by the human auditory system. Psycho-acoustic models can be very comprehensive. They can describe time masking, the masking of parts of the signal over time, frequency masking, the masking of frequency components over each other, stereo masking/unmasking, the masking or unmasking caused by the use of stereo, etc.

The psycho-acoustic model currently used in the SSC encoder only includes a frequency masking model. Every sinusoid can be seen as a frequency component with its own masking ability determined by its power and frequency. All these components together form a so-called masking curve. This is a curve in the frequency domain that describes the total masking ability of all components present. As an example the masking curve of three sinusoids with frequencies of 1000, 2000 and 4000Hz has been calculated (see Fig. 2.17). Note that the individual masking curves get broader as the frequency gets higher. This effect, as many other laws in psycho-acoustics, are related to the critical band concept [Zwicker 1961]. This concept basically states that the human auditory system can be seen as a bank of bandpass filters with different bandwidths. This concept is described by the Equivalent Rectangular Bandwidth scale (ERB) defined as Equation 2.5 [Moore 1989]:

![Figure 2.15: Multi-scale sinusoidal analysis](image-url)
\[ e_f = 21.4 \log_{10} \left( \frac{4.37f}{1000} + 1 \right). \]

where the frequency \( f \) is in Hertz and \( e_f \) the frequency in erb. The ERB scale is depicted in Figure 2.16.

![Figure 2.16: ERB scale as a function of frequency](image)

**Figure 2.16: ERB scale as a function of frequency**

![Figure 2.17: Example of frequency masking of sinusoids](image)

**Figure 2.17: Example of frequency masking of sinusoids**
Apart from the individual masking curves the total masking curve is also partially determined by the hearing threshold in quiet. This is a threshold describing how much power a single component should contain in order to be perceived in an absolute sense. The hearing threshold in quiet is shown as an interrupted line, the total masking curve is shown as a solid line in Figure 2.17.

### 2.5.3 Tracking

The main function of the tracking algorithm is to improve coding efficiency. The tracking algorithm consists of three steps:

1. **Apply tracking**: linking of sinusoidal components in time.
2. **Phaseless reconstruction**: for tracks that have been found only the initial phase has to be transmitted.
3. **Removal of non-tracks**: tracks that are very short can't be perceived as being a sinusoidal component, they are therefore removed.

The first step of the algorithm tries to link the sinusoidal components that have been found on a frame-to-frame basis [Edler et al 1996, den Brinker and Oomen 1999]. This process is shown graphically in figure 2.18. A sinusoidal component will become one of the next four possibilities:

1. **Birth**: part of a track with only a successor.
2. **Continuation**: part of a track with predecessor and successor.
3. **Death**: part of a track with only a predecessor.
4. **Non-track**: a 'track' consisting of a single frame.

Now the problem arises how to link the sinusoidal components that have been extracted. It is assumed that a track is a slowly varying function of both amplitude and frequency (see Equation 2.3). Therefore separate cost-functions for both amplitude and frequency have been developed. For the frequency the cost-function is based on the ERB-scale. The reason to do so is that for complex stimuli the relevant events are more or less separated according to the ERB-scale. The cost-function for frequency then becomes (Equation 2.6):

$$Q_{p,q}^f = \begin{cases} 0 & \text{for } |e(f_{p,k}) - e(f_{q,k-1})| \geq e_{\text{max}} \\ \frac{|e(f_{p,k}) - e(f_{q,k-1})|}{e_{\text{max}}} & \text{for } |e(f_{p,k}) - e(f_{q,k-1})| < e_{\text{max}} \end{cases}$$  \hspace{1cm} (2.6)

where $e(f_{p,k})$ denotes the frequency in erb of the $p^{th}$ component in the $k^{th}$ frame and $e_{\text{max}}$ the maximally allowed deviation expressed in erb.

For the amplitudes a similar cost-function is used (Equation 2.7):

$$Q_{p,q}^a = \begin{cases} 0 & \text{for } |A_{p,k} - A_{q,k-1}| \geq A_{\text{max}} \\ \frac{|A_{p,k} - A_{q,k-1}|}{A_{\text{max}}} & \text{for } |A_{p,k} - A_{q,k-1}| < A_{\text{max}} \end{cases}$$  \hspace{1cm} (2.7)

where $A_{p,k}$ denotes the amplitude expressed in decibels of the $p^{th}$ sinusoidal component in the $k^{th}$ frame and $A_{\text{max}}$ the maximally allowed deviation. The total cost-function now becomes (Equation 2.8):

$$Q_{p,q} = Q_{p,q}^f Q_{p,q}^a$$  \hspace{1cm} (2.8)
When for a certain sinusoid $p$ there exists no $Q_{p,q}$ greater than zero it is marked as being the end of a track. If for a certain sinusoid $p$ there exist more than one $Q_{p,q}$ greater than zero sinusoid $q$ is chosen with the largest value of $Q_{p,q}$.

The second step of the algorithm consists of phaseless reconstruction. Phaseless reconstruction is based on the assumption that the instantaneous phase of a track is a smooth function of time. For two consecutive frames of a track the instantaneous phase is equated as shown below. Assume that the sinusoid in frame $k-1$ and frame $k$ are described as (Equation 2.9 and 2.10):

\[ s_{p,k-1}[n] = A_{p,k-1} \cos(\omega_{p,k-1} n + \phi_{p,k-1}) \]  
\[ s_{q,k}[n] = A_{q,k} \cos(\omega_{q,k} n + \phi_{q,k}) \]

The equation of the instantaneous phase can be best performed at the middle of the overlapping segments. This is because the synthesis windows are (symmetric) Hanning windows. The overlap is shown graphically in Figure 2.19 for a segment length of $N=40$. Note that $n$ is defined symmetrically around zero for both Equation 2.9 as well as 2.10; shifted time-axes are thus used.
Equating the instantaneous phase of Equation 2.9 and 2.10 at the middle of the overlap of a segment with length \( N \) then gives (Equation 2.11):

\[
\omega_{p,k-1} \left( \frac{N}{4} \right) + \varphi_{p,k-1} = \omega_{q,k} \left( \frac{-N}{4} \right) + \varphi_{q,k},
\]

which results in (Equation 2.12):

\[
\varphi_{q,k} = \left( \omega_{p,k-1} + \omega_{q,k} \right) \left( \frac{N}{4} \right) + \varphi_{p,k-1}.
\]

Equation 2.12 already indicates that the first step of the tracking algorithm, the linking procedure, has great influence on quality. Erroneous linking of tracks can seriously distort phase relations between tracks.

The final step of the tracking algorithm consists of the removal of short tracks. Tracks that are shorter than five sinusoidal periods cannot be perceived by the human auditory system as being a tonal component. Such tracks are therefore deleted.

### 2.5.4 Coding of sinusoidal components

The coding of the sinusoidal components consists of:

1. Quantisation of parameters.
2. Sort data in births and continuations.
3. Sort births in frequency.
4. Sort continuations in frequency (where deaths are also seen as continuations).
5. Apply absolute and differential coding.

The quantisation of the parameters is done according to the same rules as the sinusoids in the transient code (see paragraph 2.4.3).

In order to efficiently code the parameters and the tracking information the matrices containing the sinusoidal components' frequency, amplitude and phase are sorted. This is
shown graphically in Figure 2.20. The left matrix shows how the information is stored before sorting; the right matrix shows how the information is stored after sorting. Note that for both matrices the only information needed to pick the right index of the next set of parameters belonging to a track is whether or not a sinusoidal component is continued.

For the first birth in a frame the amplitude and frequency are coded absolutely. The amplitude and frequency for all other births within a frame are coded differentially to their predecessor in the frequency domain (vertically in Fig. 2.20). For continuations the amplitude and frequency is coded differentially in the time domain (horizontally in Fig. 2.20).

![Figure 2.20: Sorting of births and continuations](image)

### 2.5.5 Sinusoidal Synthesis

The synthesis of the sinusoidal components differs from the analysis only in the windowing that is used. In the analysis section overlapping frames were analysed as (Equation 2.13):

\[ s_k[n] = \sum_p A_{p,k} \cos(\omega_{p,k}n + \varphi_{p,k}) \]  

where \( p \) denotes sinusoid index and \( k \) the frame index.

In the synthesis section the overlapping frames are synthesised as (Equation 2.14):

\[ s_k[n] = h[n] * \sum_p A_{p,k} \cos(\omega_{p,k}n + \varphi_{p,k}) \]  

where \( h[n] \) denotes the window function and \( * \) denotes convolution. The windows that are used are amplitude complementary. Three types of windows are used during synthesis:

1. Normal window, defined as a Hanning window (Equation 2.15):

\[ h[n] = \begin{cases} \frac{1}{2} + \frac{1}{2} \cos \left( \frac{2\pi}{N} \left( \frac{n - (N-1)}{2} \right) \right) & \text{for } 0 \leq n \leq N-1, \end{cases} \]  

where \( N \) denotes the frame length.
2. Start window, defined as half rectangular and half a Hanning window (Eq. 2.16):

\[
h[n] = \begin{cases} 
1 & \text{for } 0 \leq n \leq \frac{N-1}{2} \\
\frac{1}{2} + \frac{1}{2} \cos\left(\frac{2\pi}{N}\left(n - \frac{N-1}{2}\right)\right) & \text{for } \frac{N-1}{2} < n \leq N-1.
\end{cases}
\]

3. Stop window, defined as half a Hanning window and for the rest a rectangular window (Equation 2.17):

\[
h[n] = \begin{cases} 
1 & \text{for } 0 \leq n \leq \frac{N-1}{2} \\
\frac{1}{2} + \frac{1}{2} \cos\left(\frac{2\pi}{N}\left(n - \frac{N-1}{2}\right)\right) & \text{for } \frac{N-1}{2} < n \leq T - N - 1,
\end{cases}
\]

where \( T \) denotes the length of the frame needed.

Only the frames at the edges of transients use start- and stop-windows (frames 1 and 41 in Figure 2.14), all others use normal windows (frames 2 until 40 in Figure 2.14). A typical windowing sequence is depicted in Figure 2.21.
2.6 NOISE

Finally the residual signal from the sinusoidal module gets fed to the noise module. An important precondition for this module is the amount of tonality of the PCM input signal. In order for this module to work properly the tonality of the PCM input signal must be kept to a minimum. In effect this means that the transient and synthesis modules must remove as much as possible of the tonal components present in the original signal. The noise module is depicted in Figure 2.22.

![Block-diagram of noise module](image)

Figure 2.22: Block-diagram of noise module

In the noise analysis module the noise gets analysed into a set of parameters per frame. The segmentation is equal to that of the sinusoidal module. In the noise quantisation block the noise parameters get quantised and coded for use in the bitstream. Unlike the transients and the sinusoidal module the noise module doesn’t contain a noise synthesis block because during encoding there is no need to do so. Some important notes on the synthesis block are however made.

2.6.1 Noise Analysis

The transients module and the sinusoidal module both tried to match the original waveform as closely as possible. For the noise module such an approach however will not lead to an efficient representation. From a perceptual point of view this is also not an efficient representation. From that viewpoint the amount of signal (noise) power per critical band is the only relevant information. Therefore the noise module tries to spectrally model the input signal per segment. It does so by means of an auto-regressive moving-average (ARMA) model [den Brinker and Oomen 2000]. This ARMA model tries to describe the power spectral density function (psdf) as well as possible by means of poles (auto-regressive model) and zeros (moving-average model). The transfer function of an ARMA model can be described by (Equation 2.18):

\[
H = \frac{H_n}{H_d} = G \frac{\prod_{k=1}^{K} (1 - z^{-1} p_k)}{\prod_{l=1}^{L} (1 - z^{-1} q_l)},
\]

2.18

where \( H_d \) is the denominator polynomial, \( H_n \) the numerator polynomial, \( G \) the gain factor, \( K \) the number of zeros and \( L \) the number of poles. Within SSC, \( K = 1 \) and \( L = 6 \).
The block-diagram of the noise analysis module is given in Figure 2.23.

![Block-diagram of noise analysis module](image)

**Figure 2.23: Block-diagram of noise-analysis module**

First of all from the input signal the right segment is selected. From this segment the psdf is calculated by means of a smoothed FFT. Finally from the psdf the ARMA coefficients are estimated. This is depicted in Figure 2.24.

![ARMA estimation from psdf](image)

**Figure 2.24: ARMA estimation from psdf**

The process of estimating the psdf by means of an ARMA model is iterative. Mostly ARMA models aren’t used because of their computational complexity. The method as described in Figure 2.24 however tries to split the psdf into a part that can be modelled adequately by the zeros, denoted $S_1$, and a part that can be modelled adequately by the poles, denoted $S_2$. When this split has been made new estimates of the denominator and numerator polynomials are made with the use of linear prediction. The best combination of the polynomials found in the previous iteration and the current iteration is picked based on a squared error criterion on a logarithmic scale. When no more improvement can be obtained or if the maximum number of iterations is exceeded, the ARMA coefficients are passed on for further processing.

As an example Figure 2.25 shows a psdf, shown as an interrupted line, estimated by the ARMA model as described above, shown as a solid line.
2.6.2 Noise synthesis

The synthesis of the noise modelled by the ARMA model is fairly simple. The block-diagram is shown in Figure 2.26.

First of all white noise is generated using a pseudo random generator. This noise is fed through the IIR filter. For every frame a segment of white noise is filtered by the ARMA coefficients belonging to that frame. For every such segment windowing is applied. Instead of an amplitude-complementary window in the sinusoidal module a power-complementary window is used. Such a window is defined as (Equation 2.19):

\[ h[n] = \sin \left( \pi \cdot \frac{n + 0.5}{N} \right) \quad \text{for} \quad 0 \leq n \leq N - 1, \]

where \( N \) is the segment length.

Finally an overlap-add method is used to create the PCM output signal.
2.6.3 Noise quantisation

In order to efficiently code and quantise the ARMA coefficients these coefficients are converted to the Log Area Ratio (LAR). In order to do so the coefficients of both denominator and numerator polynomial are first converted to reflection coefficients as used in lattice filters. Thereafter the reflection coefficients are converted to LAR coefficients by (Equation 2.20):

\[ g = \frac{1}{2} \log \left( \frac{1 + k}{1 - k} \right), \]

where \( k \) is the reflection coefficient and \( g \) the LAR coefficient. The advantage of the LAR representation is the fact that linear quantisation can be applied. This is currently applied with an accuracy of one decibel. The gain factor \( G \) (see Equation 2.18) is also quantised with an accuracy of one decibel.

So finally, per frame eight parameters are passed on to the bit-stream formatter: a gain factor, one LAR coefficient for the numerator polynomial and six LAR coefficients for the denominator polynomial.

2.7 QUANTISATION AND CODING

Until now the focus was mainly on the extraction and processing of the parameters towards an efficient representation. It was also shown how irrelevancy is exploited within the SSC coder. To further improve coding efficiency not only irrelevancy should be exploited but also the redundancy.

To efficiently encode all the extracted and processed parameters into a bit-stream three steps are required:
1. Pre-processing (sorting),
2. Quantisation,
3. Entropy coding.

First of all the parameters need to be pre-processed. This is mainly a sorting process in which the different parameters are grouped together to form sets that can each be represented efficiently. For example the sorting of births and continuations as described in paragraph 2.5.4 belongs to the pre-processing stage, but also the sorting of sinusoidal components by frequency in the transient module.

In the second stage, quantisation is applied to the sorted parameters. The parameters must be quantised in such a manner that after decoding just no differences can be perceived before and after quantisation. For e.g. the amplitudes of the sinusoids logarithmic quantisation can be applied.

Finally entropy coding is applied to the representation levels, which represent the quantised levels. The most common form of entropy coding is Huffman coding. This is a constant to variable wordlength entropy coding technique. The main advantage of Huffman coding over other entropy coders is the low complexity: both encoding and decoding can be performed by table look-up. Furthermore these tables are easily constructed.

A general diagram for coding audio parameters using Huffman coding is depicted in Figure 2.27.
As an example the following process $x$ is given (Table 2.1):

<table>
<thead>
<tr>
<th>Parameter value $x$:</th>
<th>Representation level:</th>
<th>Quantised representation $\hat{x}$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.00 \leq x &lt; 0.52$</td>
<td>0 (00)</td>
<td>0.32</td>
</tr>
<tr>
<td>$0.52 \leq x &lt; 0.82$</td>
<td>1 (01)</td>
<td>0.74</td>
</tr>
<tr>
<td>$0.82 \leq x &lt; 0.95$</td>
<td>2 (10)</td>
<td>0.89</td>
</tr>
<tr>
<td>$0.95 \leq x &lt; 1.00$</td>
<td>3 (11)</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Now state that these representation levels have the following distribution:

<table>
<thead>
<tr>
<th>Representation level:</th>
<th>Probability of occurrence:</th>
<th>Huffman code:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (00)</td>
<td>0.5</td>
<td>'0'</td>
</tr>
<tr>
<td>1 (01)</td>
<td>0.3</td>
<td>'01'</td>
</tr>
<tr>
<td>2 (10)</td>
<td>0.1</td>
<td>'001'</td>
</tr>
<tr>
<td>3 (11)</td>
<td>0.1</td>
<td>'000'</td>
</tr>
</tbody>
</table>

The Huffman table is built up in such a way that less probable representations get longer codewords. For this particular code the mean word-length (representing a single representation) is $0.5 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 3 = 1.7$ bits per representation, which is a coding gain of $2/1.7 = 1.17$.

One disadvantage of Huffman coding is that the look-up tables might become rather large. In case a large number of representations only occur with a small probability escape codes can be used advantageously. Table 2.3 shows an example of the usage of an escape code. Using escape-codes in Table 2.3 only four codes have to be stored, while in the case of no escape-code all sixteen codes have to be stored. The code with escape will on average generate 1.9 generate per symbol (representation) while the 'pure' code will generate 1.89 bits per symbol (representation).
Table 2.3: Huffman table with and without escape code

<table>
<thead>
<tr>
<th>Code</th>
<th>Probability:</th>
<th>Code with escape</th>
<th>Code without escape</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>0.3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0010</td>
<td>0.15</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>0011</td>
<td>0.0038</td>
<td>111 0011</td>
<td>111101</td>
</tr>
<tr>
<td>0100</td>
<td>0.0038</td>
<td>111 0100</td>
<td>111110</td>
</tr>
<tr>
<td>0101</td>
<td>0.0038</td>
<td>111 0101</td>
<td>111111</td>
</tr>
<tr>
<td>0110</td>
<td>0.0038</td>
<td>111 0110</td>
<td>1110000</td>
</tr>
<tr>
<td>0111</td>
<td>0.0038</td>
<td>111 0111</td>
<td>1110001</td>
</tr>
<tr>
<td>1000</td>
<td>0.0038</td>
<td>111 1000</td>
<td>1110010</td>
</tr>
<tr>
<td>1001</td>
<td>0.0038</td>
<td>111 1001</td>
<td>1110011</td>
</tr>
<tr>
<td>1010</td>
<td>0.0038</td>
<td>111 1010</td>
<td>1110100</td>
</tr>
<tr>
<td>1011</td>
<td>0.0038</td>
<td>111 1011</td>
<td>1110101</td>
</tr>
<tr>
<td>1100</td>
<td>0.0038</td>
<td>111 1100</td>
<td>1110110</td>
</tr>
<tr>
<td>1101</td>
<td>0.0038</td>
<td>111 1101</td>
<td>1110111</td>
</tr>
<tr>
<td>1110</td>
<td>0.0038</td>
<td>111 1110</td>
<td>1111000</td>
</tr>
<tr>
<td>1111</td>
<td>0.0038</td>
<td>111 1111</td>
<td>1111001</td>
</tr>
</tbody>
</table>
3 QUALITY ASSESSMENT EXPERIMENTS

3.1 INTRODUCTION

The first objective of the graduation project was to identify the problems that were within the SSC encoder. In order to do so numerous experiments have been performed. In this chapter the most important experiments are described and conclusions of these experiments are given.

3.2 GENERAL QUALITY ASSESSMENT

The first step in analysing the quality limitations of the SSC coder focused on listening and evaluating the decoded versions of extracted parameters and to residual signals at every step of the encoder. This is shown graphically in Figure 3.1. For this purpose a set of thirteen excerpts were used that are generally known to be critical for audio coding. Throughout the whole project this set has been used for evaluation purposes.

![Figure 3.1: Evaluation points for quality assessment](image)

First of all when comparing the original signal \((t1)\) with the quantised transients signal \((t3)\) plus the transients residual signal \((s1)\) no performance loss seems to occur. However, when listening to the extracted transients only \((t2)\), it seems that only a small part of what's perceived as a transient is actually removed. This is confirmed by listening to the residual signal of the transients module \((s1)\). Especially when listening to signals, which have clear transients like e.g. castanets, still some residue of the transients seem to be present.

What's first of all noticeable in the sinusoids extraction module is the fact that the residual signal of this module \((n1)\) still contains a lot of tonal elements. Ideally this signal should be perceived as being noisy. These tonal elements could be attributed to shortcomings in both the transients as well as the sinusoidal module. When listening to the sinusoids only \((s2)\) in comparison to the residual signal from the transients module \((s1)\) it is noticed that the sinusoids \((s2)\) have less sharpness or transient-like behaviour.
When comparing the sinusoids that have been extracted before \((s2)\) and after the psycho-acoustic model \((s3)\) no noticeable differences can be perceived. When however comparing the sinusoids after the psycho-acoustic model \((s3)\) with the sinusoids after tracking \((s4)\) a big quality loss is noticed. Further listening within this block showed two reasons:

1. The use of phaseless reconstruction seems to cause a certain echo-like effect. This effect is also described as 'metallic' sound.
2. The removal of non-tracks removes a lot of sharpness of the signal, especially on positions where transients have been detected.

Quantisation of the parameters \((s5)\) even further decreased the quality of the sinusoids. This mainly increases the 'metallic' sound of signals.

When comparing the noise signal generated from the extracted parameters \((n2)\) with the residual signal of the sinusoidal module \((n1)\) a big quality loss is observed. This is however not caused with certainty by the performance of the noise module. It could be caused by the condition of the residual signal. This signal still contains a lot of tonal elements that show up as peaks in the frequency domain. Because of the limited amount of ARMA coefficients the noise coder would profit from a smooth frequency spectrum.

Finally the noise signal synthesised from the quantised parameters \((n3)\) was compared with the noise signal before quantisation \((n2)\). This comparison showed that quantisation of the parameters caused no noticeable loss of quality.

All in all the following conclusions can be made:

- The transients module performs reasonably well. Quantisation of parameters leads to no perceptual loss. It is noted however that the residual signal from this module still contains transient-like information.
- The sinusoidal module isn't able to fully remove all tonal components from its input signal. Furthermore phaseless reconstruction, the removal of short tracks and quantisation further decrease the quality.
- It is unsure how the noise module performs because of the condition of its input signal. Quantisation however doesn't seem to lead to an additional decrease of quality.

### 3.3 FREQUENCY SELECTION BASED ON PSYCHO-ACOUSTICS

The biggest problems within the SSC coder seem to occur in the sinusoidal module. First of all, the extraction method on itself doesn't seem to perform as expected. What was primarily noted when looking at the extracted sinusoidal parameters is the fact that single peaks in the spectrum where often modelled by more than a single sinusoid. This is shown in more detail in Figure 3.2. The top-left of Figure 3.2 shows a segment of the input data. It consists of two sinusoids, one with high amplitude and one with low amplitude, that are both slightly amplitude and frequency modulated, with little noise added. A part of the absolute FFT spectrum is shown in the top-right. It clearly shows two peaks, belonging to the two sinusoids. After extraction of a constant sinusoid clearly the sinusoid with the high amplitude isn't removed completely. The spectrum of the residual signal clearly shows two other peaks arising that are still larger in amplitude than the peak of the second sinusoid. The next peak that will now be processed will be one of the peaks that have just arisen.
The process as described in Figure 3.2 in practice often causes a single spectral peak to be described by more than a single sinusoid. Extraction of sinusoids varying in amplitude and frequency doesn't seem to be necessary at first, because at the synthesis stage the overlap-add structure to a certain extent will re-create such variations. Apart from the fact that the extraction method isn't efficient the extra sinusoids disturb the tracking mechanism. All in all it seems more logical to assign at most a single sinusoid to every peak of the spectrum. In order to do so a frequency selection algorithm based on frequency masking is proposed. This is shown graphically in Figure 3.3. Instead of just picking the maximum peak of the spectrum the new criterion consists of taking the maximum peak of the difference between the masking curve and spectrum. This process is depicted in Figure 3.4.

The masking curve is initialised as the absolute threshold of hearing; this is however not shown in Figure 3.3. Then in the first step the maximum difference between the masking curve and the spectrum is selected. Before the sinusoid is actually extracted the masking curve of the single sinusoid is calculated and added to the total masking curve. Then the sinusoid is actually extracted and the process starts all over.
Figure 3.3: Frequency selection using frequency masking

The main goal of using frequency masking in the sinusoidal extraction routine is that a single spectral peak will now be described by a single sinusoid, unless there are only a small number of peaks present. Another advantage is that the frequency selection is done on a perceptual basis. Sinusoids are now selected on perceptual relevance.

Figure 3.4: Block-diagram of sinusoidal extraction using frequency masking

The adapted algorithm as described above (see Fig. 3.4) has been implemented and applied to all thirteen critical excerpts. A comparison has been made between the original extraction method and the new method. The main conclusions were:

- Quality of the sinusoids only (Figure 3.1, s2) didn’t increase at all.
- Quality of the sinusoids after tracking (Figure 3.1, s4) also didn’t increase. The same artefacts were still present.
- Tonality in the residual signal didn’t decrease.

All in all no quality gain was obtained using the method described above. It seems that for extraction the signal structure of a segment, being a sum of fifty constant sinusoids with noise, is insufficiently rich.
3.4 TWO-STEP EXTRACTION

The residual signal of the sinusoidal module still contains elements that could be described as being either transients or sinusoids. Therefore another experiment was proposed. In this experiment instead of feeding the residual signal from the sinusoidal module to the noise module, it is fed once again through the whole coder (see Figure 3.5).

![Block-diagram of two-step extraction of parameters](image)

In effect the structure of Figure 3.5 leads to a double amount of transient and sinusoidal parameters, which is of course unwanted. However some important conclusions could be drawn from this experiment:

- The residual signal of the second sinusoidal module was almost completely free of tonal elements. Comparing this residual signal with the noise from the noise module showed just a very little degradation in quality. This indicates that the noise module performs well when its input signal is also of a noise-like character.
- When listening to the signal synthesised as the sum of transient code 1, sinusoids code 1 and 2 and noise code, showed that the extraction now delivered almost transparent quality audio. However it must be noted that the sinusoids extracted in the second stage also seemed to code a part of the noise.
- When listening to the signal synthesised as the sum of transient code 1, sinusoids code 1 and noise code the quality is still close to transparency. Only at transient positions the signal seems to miss some transient-like information.

All in all, it can be concluded that if the sinusoids are adequately extracted both the transients as well as the noise module perform satisfactory. Furthermore the description using constant sinusoids is sufficient for the stationary parts of the signal.
4 IMPROVED EXTRACTION OF SINUSOIDS

4.1 INTRODUCTION

The experiments described in the previous chapter indicate that in order to achieve high quality audio the sinusoidal module will have to be significantly improved. The extraction of parameters, the tracking and the quantisation each show a significant decrease in quality. In this chapter the focus is placed on the extraction of the sinusoidal parameters.

The residual signal from the noise module still contains a lot of tonal elements. This indicates that the sinusoidal parameters aren't extracted adequately. In this chapter this problem is addressed. An extension for the sinusoidal model is described.

4.2 AMPLITUDE AND FREQUENCY VARIATION

The experiments that have been performed, as described in the previous chapter, seem to indicate that a description using sinusoids with constant parameters isn't sufficient. For a single segment a description using constant sinusoids is defined according to (Equation 4.1):

\[ s[n] = \sum_{p=1}^{P} A_p \cos(\omega_p n + \varphi_p) \]  

where \( p \) denotes the \( p \)th sinusoid, \( P \) the total number of sinusoids, \( A \) the amplitude, \( \omega \) the frequency, \( \varphi \) the phase and \( n \) is defined as being a uniformly spaced vector of segment length \( N \) (Equation 4.2):

\[ n = \left\{ \frac{-N-1}{2}, \frac{-N-3}{2}, \frac{-N-5}{2}, \ldots, \frac{3}{2}, \frac{1}{2}, \frac{3}{2}, \frac{N-3}{2}, \frac{N-1}{2} \right\} \]  

Vector \( n \) (see Equation 4.2) is chosen to be symmetric around zero. This has the advantage that some calculations can be performed more easily. It could however, without any change of results, also have been defined e.g. as a vector from 0 to \( N-1 \) or any other vector of length \( N \) with step-size one.

The experiments in the previous chapter indicated that broad spectral peaks in the FFT could not be modelled adequately by constant sinusoids. The question thus arises what causes this broadening of the peaks, so that this can be included into the model. It is known from signal theory that both amplitude variation and frequency variation cause broadening of spectral peaks. As an example consider the following signal (Equation 4.3):

\[ s[n] = (A + Bn)\cos(\omega n + \varphi) \]  

Amplitude \( A \), frequency \( \omega \) and phase \( \varphi \) are set to respectively 1000, 0.2\( \pi \) radians and 0 radians. \( n \) is defined as in Equation 4.2 and \( B \) is varied from 0 to 1 in steps from 0.2. \( s[n] \) is shown in Figure 4.1 for \( B = 0 \) and \( B = 1 \).
A part of the absolute FFT is shown in Figure 4.2. It is quite clear that the peak broadens for increasing linear amplitude modulation. For frequency variation similar results can be achieved.

The first step in extending the sinusoidal model could be the inclusion of a linear amplitude term, just like above. Equation 4.1 turns into (Equation 4.4):

\[
 s[n] = \sum_{p=1}^{P} \left( A_p \cos(\omega_p n + \varphi_p) + B_p n \cos(\omega_p n + \varphi_p) \right) 
\]
With such a description of sinusoids, linearly varying amplitudes can be modelled. A further extension could be the inclusion of a quadratic term. The description would then look like (Equation 4.5):

$$s[n] = \sum_{p=1}^{P} \left( A_p \cos(\omega_p n + \varphi_p) + B_p n \cos(\omega_p n + \varphi_p) + C_p n^2 \cos(\omega_p n + \varphi_p) \right)$$

4.5

In Equations 4.4 and 4.5 the phase of each term is the same. When this constraint is loosened an even greater degree of freedom is achieved. In order to see the effect of phase independence in Equation 4.5, the focus is placed on a single sinusoid $s_p[n]$:

$$s[n] = \sum_{p=1}^{P} s_p[n],$$

4.6

where $s_p[n]$ now is defined as (Equation 4.7):

$$s_p[n] = A_p \cos(\omega_p n + \varphi_{p,a}) + B_p n \cos(\omega_p n + \varphi_{p,b}) + C_p n^2 \cos(\omega_p n + \varphi_{p,c})$$

4.7

Equation 4.7 can also be written in a complex form (Equation 4.8):

$$s_p[n] = \Re\left\{ (a + bn + cn^2) e^{j\omega_p n} \right\},$$

4.8

where $\Re\{\}$ describes the real part of a complex value, the subscript $p$ is left out for clarity and $a$, $b$ and $c$ are independent complex coefficients defined as (Equation 4.9):

$$a = A_p e^{j\varphi_{p,a}}, \quad b = B_p e^{j\varphi_{p,b}}, \quad c = C_p e^{j\varphi_{p,c}}.$$  

4.9

It has already been noted that with the inclusion of the extra terms, linear and quadratic amplitude variations can be modelled. The question now arises whether it is possible to model frequency variation. Equation 4.10 describes a sinusoid with constant amplitude and a linearly varying frequency.

$$s_p[n] = A \cos((\omega + \Delta \omega n)n + \varphi),$$

4.10

where $\Delta \omega$ describes the linear frequency variation. Equation 4.10 can also be written in its complex form. Using a Taylor approximation it can be shown that, to a certain extent, frequency variation can also be modelled (Equation 4.11):

$$s_p[n] = \Re\left\{ A e^{j\varphi} e^{j\omega n} e^{j\Delta \omega n^2} \right\}$$

$$= \Re\left\{ A e^{j\varphi} e^{j\omega n} (1 + j \Delta \omega n^2 + ...) \right\}$$

$$\approx \Re\left\{ (a + cn^2) e^{j\omega n} \right\},$$

4.11

where $a$ and $c$ are complex coefficients that are 90° out of phase. Note that the Taylor-approximation is only valid around the origin of the expansion. Equation 4.11 only indicates that frequency variation can be modelled to a certain extent. To get some insight to what extent, $s_p[n]$ can be written as:

$$s_p[n] = \Re\left\{ R(n) e^{j\varphi(n)} \right\}$$
where \( R(n) \) describes the envelope and \( p(n) \) describes the argument of the sinusoid. The *instantaneous* frequency as a function of time can then be described as:

\[
\Omega(n) = \frac{d}{dn} p(n)
\]

To come to a description as in Equation 4.12 \( s_p[n] \) is first written as (Equation 4.14):

\[
s_p[n] = \Re \left\{ (a + bn + cn^2) e^{in} \right\} = \Re \{q(n) + jr(n)\},
\]

with (Equation 4.15):

\[
q(n) = A \cos(\omega n + \varphi_a) + Bn \cos(\omega n + \varphi_b) + Cn^2 \cos(\omega n + \varphi_c),
\]

\[
r(n) = A \sin(\omega n + \varphi_a) + Bn \sin(\omega n + \varphi_b) + Cn^2 \sin(\omega n + \varphi_c).
\]

Now Equation 4.12 can be determined using the following relations:

\[
R(n) = \sqrt{(r(n))^2 + (q(n))^2},
\]

\[
p(n) = \arctan \left( \frac{r(n)}{q(n)} \right),
\]

Using Equation 4.13 and 4.16 the instantaneous frequency as a function of time can be calculated:

\[
\Omega(n) = \frac{d}{dn} \arctan \left( \frac{r(n)}{q(n)} \right) = \frac{q(n) \frac{dr(n)}{dn} - r(n) \frac{dq(n)}{dn}}{(r(n))^2 + (q(n))^2}.
\]

Finally this gives for the instantaneous frequency \( \Omega(n) \):

\[
\Omega(n) = \frac{A(\omega A + Bc_{ab}) + 2A(\omega Bc_{ab} + CS_{ca})n + (\omega B^2 + BCs_{cb} + 2\omega ACC_{ac})n^2 + 2\omega BCc_{bc}n^3 + \omega C^2 n^4}{A^2 + 2ABc_{ab}n + (B^2 + 2ACc_{ac})n^2 + 2BCc_{bc}n^3 + C^2 n^4},
\]

where \( s_{qr} \) and \( c_{qr} \) are defined as:

\[
s_{qr} = \sin(\varphi_q - \varphi_r),
\]

\[
c_{qr} = \cos(\varphi_q - \varphi_r).
\]

and are thus independent of \( n \). The instantaneous frequency \( \Omega(n) \) is thus a quotient of a fourth order numerator polynomial and a fourth order denominator polynomial. Looking at Equation 4.18 it's still hard to give quantitative measures in to what extent frequency variation can be modelled. However some important properties of the instantaneous frequency can be derived from Equation 4.18.
First of all for $n = 0$ the instantaneous frequency $\Omega$ can deviate from the polynomial modulation frequency $\omega$:

$$\Omega(0) = \frac{A(\omega A + B S_{ba})}{A^2} = \omega + \frac{B}{A} \sin(\varphi_b - \varphi_s)$$  \hspace{1cm} 4.20

For large values of $n$, both positive and negative, the instantaneous frequency $\Omega$ approaches the polynomial modulation frequency $\omega$:

$$\lim_{n \to \pm \infty} \Omega(n) = \lim_{n \to \pm \infty} \omega = \frac{\omega C^2 n^4}{C^2 n^4} = \omega$$  \hspace{1cm} 4.21

The terms $B$ and $C$ used in all the equations above aren't normalised. Therefore when $n$ is large, e.g. $(N - 1)/2$, the influence of the fourth order terms doesn’t necessarily dominate the other terms.

Finally, Figure 4.3 gives an example of frequency variation handled by the second order polynomial sinusoid model. The upper figure shows the waveform. The lower figure shows the instantaneous frequency (black line) and the polynomial modulation frequency (grey line) of $1/20\pi$ radians. The instantaneous frequency almost shows a linear course. This figure thus shows that independent phases of the complex polynomial of Equation 4.8 can indeed handle frequency variation to some degree.

![Figure 4.3: Example of frequency variation by second order polynomial](image.png)
4.3 LINEAR REGRESSION

A proposal for the extension of the sinusoidal model with linear and quadratic amplitude modulated polynomials has been given in the previous paragraph. However no mention has been made about how to extract the coefficients. An optimal solution, in a least-squares sense, for this problem can be found using linear regression. In order to come to such a solution, Equation 4.7 is written in its complex form (Equation 4.22):

\[ s_p[n] = ae^{j\omega n} + \bar{a}e^{-j\omega n} + bne^{j\omega n} + \bar{b}ne^{-j\omega n} + cn^2e^{j\omega n} + \bar{c}n^2e^{-j\omega n}, \]

where \( a, b \) and \( c \) are complex coefficients and \( \bar{a}, \bar{b} \) and \( \bar{c} \) are their complex conjugates. Note that \( a, b \) and \( c \) differ a factor two from the definition in equation 4.8 because all terms of Equation 4.8 are in fact included twice in Equation 4.22. \( s_p[n] \) now consists of six complex coefficients multiplied with six complex patterns.

Given a set of complex patterns and an input vector \( \mathbf{x} \) a general solution can be given, denoted as a vector \( \mathbf{a} \):

Equation \( P\mathbf{a} = \mathbf{x} \) where \( P \) is an \( m \times n \) matrix containing \( n \) complex patterns of length \( m \) and vectors \( \mathbf{a} \in \mathbb{C}^n \) and \( \mathbf{x} \in \mathbb{C}^m \) in general doesn't have a solution (\( m > n \)) or an infinite amount of solutions (\( m < n \)). When \( m \) observations, in our case \( N \) samples, of the signal vector \( \mathbf{x} \) have been made one can exploit these observations to form an \( n \)-variable linear model; where \( n \ll m \). This is known as linear regression. Because of the possible error in \( \mathbf{x} \), the coefficient vector \( \mathbf{a} \in \mathbb{C}^n \) is calculated that minimises the Euclidean length \( \|P\mathbf{a} - \mathbf{x}\| \). This is also known as the least-squares problem.

If \( \mathbf{a} \) is a solution it minimises (Equation 4.23):

\[
\|P\mathbf{a} - \mathbf{x}\|^2 = \langle P\mathbf{a} - \mathbf{x}, P\mathbf{a} - \mathbf{x} \rangle,
\]

\[
= \langle P\mathbf{a}, P\mathbf{a} \rangle + 2\langle -\mathbf{x}, P\mathbf{a} \rangle + \langle -\mathbf{x}, -\mathbf{x} \rangle,
\]

\[
= a^H P^H P\mathbf{a} - 2a^H P^H \mathbf{x} + \mathbf{x}^T \mathbf{x},
\]

where \( (.)^H \) denotes the Hermitian transpose, i.e. \( A^H = \overline{A}^T \). If \( f(\mathbf{a}) \) has a minimum, the gradient must equal zero: \( \nabla f(\mathbf{a}) = 0 \). Because \( \nabla f(\mathbf{a}) = P^H P\mathbf{a} - P^H \mathbf{x} \), the solution is: \( P^H (P\mathbf{a} - \mathbf{x}) = 0 \). Now suppose that indeed \( P^H (P\mathbf{a} - \mathbf{x}) = 0 \) and let \( \mathbf{c} \in \mathbb{C}^n \) and \( \mathbf{d} \in \mathbb{C}^n \). With \( \mathbf{c} = \mathbf{a} + \mathbf{d} \) the following can be derived (Equation 4.24):

\[
\|P\mathbf{c} - \mathbf{x}\|^2 = \langle P\mathbf{a} + P\mathbf{d} - \mathbf{x}, P\mathbf{a} + P\mathbf{d} - \mathbf{x} \rangle,
\]

\[
= \langle P\mathbf{a} - \mathbf{x}, P\mathbf{a} - \mathbf{x} \rangle + \langle P\mathbf{d}, P\mathbf{d} \rangle + 2\langle P\mathbf{a}, P\mathbf{d} \rangle + 2\langle -\mathbf{x}, P\mathbf{d} \rangle,
\]

\[
= \|P\mathbf{a} - \mathbf{x}\|^2 + \|P\mathbf{d}\|^2 + 2\mathbf{a}^H P^H P\mathbf{a} - 2\mathbf{d}^H P^H \mathbf{x},
\]

\[
= \|P\mathbf{a} - \mathbf{x}\|^2 + \|P\mathbf{d}\|^2 + 2\mathbf{a}^H (P^H (P\mathbf{a} - \mathbf{x})),
\]

\[
= \|P\mathbf{a} - \mathbf{x}\|^2 + \|P\mathbf{d}\|^2 \geq \|P\mathbf{a} - \mathbf{x}\|^2.
\]
So indeed \( P^H (Pa - x) = 0 \) is the optimal solution and thus:

\[ a = (P^H P)^{-1} P^H x. \]  \hspace{1cm} 4.25

The \( m \times m \) matrix \( P^H P \) is also known as the Gram matrix. It contains the energy of the pattern functions on the main diagonal and the cross-correlations between the pattern functions in the off-diagonal elements. If the pattern functions are linearly independent, the Gram matrix can be inverted and Equation 4.25 will have a unique solution.

Given equation 4.22 \( \bar{x} \) is defined as (Equation 4.26):

\[ x = x[n], \quad \text{where} \quad n = \{r, r + 1, r + 2, r + 3, r + 4, \ldots, r + N - 1\}, \]  \hspace{1cm} 4.26

and where \( r \) is an integer value denoting the start position of the segment that is to be analysed.

\( \bar{a} \) is defined as a vector containing the six complex coefficients:

\[ \bar{a} = \left[ \begin{array}{c} a \\ \bar{a} \\ b \\ \bar{b} \\ c \\ \bar{c} \end{array} \right]. \]  \hspace{1cm} 4.27

\( P \) is defined as a matrix containing the six complex patterns (Equation 4.28):

\[ P = \begin{bmatrix} e^{j\omega n} & e^{-j\omega n} & ne^{j\omega n} & ne^{-j\omega n} & n^2e^{j\omega n} & n^2e^{-j\omega n} \end{bmatrix} \]  \hspace{1cm} 4.28

where \( n \) is defined as Equation 4.2 in column-vector notation.

Using Equations 4.26, 4.27 and 4.28, given a segment of data and given a frequency \( \omega \), the optimal coefficients for that frequency and that segment will be determined by least-squares optimisation. The solution of Equation 4.25 will however only work optimally when the frequency has been estimated precisely. As described in Chapter 2, the frequency is first of all estimated on an FFT grid. After this coarse estimation a finer estimate is made using interpolation. Some small experiments using this technique showed that this interpolation was sufficiently accurate.

Finally, as an example the sinusoidal extraction using the higher order polynomials is compared to the extraction using constant (0\(^{th}\) order) sinusoids (see Figure 4.4). Where the residual signal using constant sinusoids still showed a clear residual in the FFT spectrum (gray line, bottom-right), the extraction using the higher order polynomials clearly gives less residual (black line, bottom-right).
4.4 MULTI-SCALE SINUSOIDAL EXTRACTION

The linear regression algorithm as described above will extract a set of three complex coefficients (and their complex conjugates) for each sinusoid. Where in the 0th order model only simple phase shifts and amplitude compensation had to be used to convert the parameters from one scale to the other (see Figure 2.15), now the higher order coefficients need to be scaled.

Both filter compensation and scale transitions consist of time shifts and time scaling. The effect of both transformations can be simply described using Equation 4.8.

First of all suppose \( s_p[n] \) is estimated on the interval \((-M, M)\). From this an approximation, \( s_{p,z}[n] \), must be made for the shifted interval \((\tau - M, \tau + M)\) (Equation 4.29):

\[
s_p[n] = \Re\left\{ (a + bn + cn^2)e^{j\omega n} \right\}
= \Re\left\{ (a + b(n - \tau + \tau) + c(n - \tau + \tau)^2)e^{j\omega(n-\tau+\tau)} \right\}
= \Re\left\{ (a_r + b_r m + c_r m^2)e^{j\omega m} \right\},
\]

Equation 4.29

with \( m = n - \tau \), \( a_r = (a + b\tau + c\tau^2)e^{j\omega} \), \( b_r = (b + 2c\tau)e^{j\omega} \) and \( c_r = ce^{j\omega} \). For the complex conjugate form of Equation 4.8 similar results can be achieved.
Now suppose $s_p[n]$ is estimated on the interval $(-M, M)$. An approximation, $s_{p, r}[n]$, must be made on the scaled interval $(-M/r, M/r)$ with the same amount of samples. This results in (Equation 4.30):

$$s_p[n] = \Re \left\{ (a + bn + cn^2) e^{j\omega n} \right\},$$

$$= \Re \left\{ (a + b(rn/r) + c(rn/r)^2/r^2) e^{j\omega (rn)/r} \right\},$$

$$= \Re \left\{ (a_r + b_r (rn/r) + c_r (rn/r)^2/r^2) e^{j\omega (rn)/r} \right\},$$

with $m = rn$, $a_r = a$, $b_r = b/r$, $c_r = c/r^2$ and $\omega_r = \omega/r$.

**4.4.1 Filter compensation**

For constant sinusoids ($0^{th}$ order descriptions), the filter compensation can be performed quite easily. For the given frequency first the amplitude attenuation and phase shift of the anti-aliasing filter are determined (see Figure 4.5 and 4.6). To compensate for the filter, the amplitude of the sinusoids is amplified by the inverse of the amplitude amplification and the phase is adjusted according to the phase shift. Such a parameter transformation is in essence only valid for stationary sinusoids, i.e. constant amplitude and constant frequency. The assumption is however made that for the first and second order polynomials stationary conditions also apply. Comparison the residual signals on the different scales (see Figure 2.15) indicates that, for the given anti-aliasing filter, this is indeed a reasonable assumption.

![Figure 4.5: Amplification (dB) of anti-aliasing filter as a function of frequency](image-url)
4.4.2 Scale transitions

The scale transitions consist of two separate parts: first of all the parameters are scaled to an upsampled domain, thereafter the time shifts according to the segmentation as described in Figure 2.14 is performed. This is illustrated once more in Figure 4.7.
4.5 EXPERIMENTS

In the previous paragraphs an extension for the multi-scale sinusoidal extraction with complex modulated second order polynomials has been proposed. In order to assess whether this alleviates the problems of the sinusoidal extraction the algorithm as described above has been implemented and some experiments have been performed. These experiments mainly consisted of informal listening tests on the set of critical excerpts.

In order to perform the listening tests the following signals have been generated (for every excerpt):
- \( t \): transient signal reconstructed from decoded quantised transient parameters,
- \( s0, s1, s2 \): respectively 0\(^{th} \), 1\(^{st} \) and 2\(^{nd} \) order polynomial signals reconstructed of unprocessed extracted parameters and
- \( n \): noise signal reconstructed of decoded quantised noise parameters obtained using \( s0, s1 \) and \( s2 \).

From these signals two other signals have been constructed; one consisting of the sum of \( t, s0, s1, s2 \) and \( n \) and one consisting of the sum of \( t, s0 \) and \( n \).
First of all, the signal consisting of \( t, s0, s1, s2 \) and \( n \) was compared to the original signal. The quality of this signal was close to transparency, i.e. for most excerpts it was impossible to find a distinction between the original signal and the parameterised version.

Secondly, the signal consisting of \( t, s0 \) and \( n \) was also compared to the original signal. The quality of this signal was slightly less than the quality of the signal consisting of \( t, s0, s1, s2 \) and \( n \). Especially around transients, differences were noticeable. The signal consisting of \( t, s0 \) and \( n \) clearly misses some transient-like, and thus fast amplitude-varying, information.

Finally, the noise signal \( n \) was compared to the residual signal of the sinusoidal module. The differences, although clearly perceivable, were much smaller than only using the extraction of the \( 0^{th} \) order polynomials.

From the three tests described above some important conclusions can be drawn:
- If the second order polynomial parameters are coded in a perceptually lossless manner, the quality of the total decoded signal, i.e. transients plus sinusoids plus noise, will be very close to transparent.
- When the extraction is based on 2\(^{nd} \) order polynomials while only the \( 0^{th} \) order parameters are used for synthesis, the quality is still close to transparent. However, especially around transients some information is lacking. This means that the information that is kept within the 1\(^{st} \) and the 2\(^{nd} \) order term of the polynomial is only perceptually relevant around transients.

From the last conclusion two other conclusions can be extracted: the noise-module seems to perform sufficiently well, the transient module however not. This is consistent with the quality assessment experiments described in Chapter 3. There it was stated that the transients-module didn't successfully remove all transient information. The shortcomings of the transient-module can be relaxed by implementing information of the 1\(^{st} \) and 2\(^{nd} \) order polynomial terms in the bit-stream at transient positions.

All in all, it can be concluded that the multi-scale extraction using the complex modulated second order polynomials drastically improves the quality, even when only the \( 0^{th} \) order information is actually encoded.

### 4.6 ALTERNATIVE FREQUENCY ESTIMATION

A major disadvantage of the current extraction method is the computational complexity. The iterative extraction method (see Figure 2.13) requires that for every segment fifty FFT's are calculated, one for each extracted sinusoid. For real-time applications this isn't acceptable. An approach as depicted in Figure 4.8 would be much more efficient.

![Figure 4.8 Block-diagram of non-iterative sinusoidal extraction method](image)

The only change that is made compared to Figure 2.13 is the estimation of the frequencies. While in Figure 2.13 one frequency was estimated at a time, in Figure 4.8 all frequencies are estimated at once. In order to come to a more efficient sinusoidal extraction
scheme the problem thus becomes, given a segment of data, how to accurately estimate the frequencies of all relevant sinusoids.

Desainte-Catherine and Marchand [Desainte-Catherine and Marchand 2000] give a possible solution. The method they describe uses information of the signal derivative to estimate the frequency of a sinusoid. A sinusoid within a segment is written as (Equation 4.31):

\[ s_p[n] = A_p(n) \cos(\omega_p(n)n + \phi_p) \]  \hspace{1cm} 4.31

When \( A_p(n) \) and \( \omega_p(n) \) are slowly varying in time, the continuous derivative is given by (Equation 4.32):

\[
\frac{d s_p[n]}{dn} = \frac{d}{dn} \{ A_p(n) \cos(\omega_p(n)n + \phi_p) \},
\]

\[
\approx \frac{d}{dn} \{ A_p \cos(\omega_p n + \phi_p) \},
\]

\[
= -\omega_p A_p \sin(\omega_p n + \phi_p)
\]  \hspace{1cm} 4.32

Equation 4.32 shows that the amplitude of the Fourier transform of \( s_p[n] \), denoted as \( FT^0 \), compared to the Fourier transform of its continuous derivative, \( FT^1 \), differ by a factor \( \omega_p \) (Equation 4.33):

\[ \omega_p = \frac{FT^1(\omega_p)}{FT^0(\omega_p)} \]  \hspace{1cm} 4.33

This equation seems of no interest at first because of prior knowledge of \( \omega_p \). The discrete version however does seem of interest. In the discrete-time domain a continuous derivative can be approximated using first order discrete derivatives (Equation 4.34):

\[ s_p'[n] = (s_p[n] - s_p[n-1])F_s. \]  \hspace{1cm} 4.34

This has the form of a linear-phase high-pass filter with a gain function of (Equation 4.35):

\[ |H(e^{j\omega})| = 2F_s \sin \left( \frac{\omega}{2} \right). \]  \hspace{1cm} 4.35

The continuous derivative can also be seen as a filtering operation with a gain of \( F_s \omega \). This gain of the first order derivative differs from the continuous derivative by a scaling factor \( F \) defined as (Equation 4.36):

\[ F(\omega) = \frac{\omega}{2 \sin \left( \frac{\omega}{2} \right)}. \]  \hspace{1cm} 4.36
The discrete equivalent of Equation 4.33 is (Equation 4.37):

\[ \omega_p = F(\omega_p) \frac{DFT^1[m_p]}{DFT^0[m_p]}, \tag{4.37} \]

where \( DFT^0 \) denotes the Discrete Fourier Transform of the original discrete signal, \( DFT^1 \) the Discrete Fourier Transform of the first order derivative and \( m_p \) the index of the maximum in \( DFT^0 \) corresponding to frequency \( \omega_p \). Figure 4.9 shows the gain function of the first order derivative.

The main advantage of using Equation 4.37 over the iterative method is the usage of window functions. This means that a higher spectral resolution can be obtained than when using a rectangular window.

Using Equation 4.37, for every maximum of \( DFT^0 \) the frequency of the underlying sinusoid can be determined. Equation 4.36 and equation 4.37 can be combined into (Equation 4.38):

\[ \omega_p = 2 \arcsin \left( \frac{DFT^1[m_p]}{2DFT^0[m_p]} \right), \tag{4.38} \]

where \( DFT^0 \) represents the Discrete Fourier Transform of the original signal.

Towards higher frequencies the first order derivative more and more deviates from the continuous derivative. In effect this causes a lower precision towards these higher frequencies. A small deviation on the \( \arcsin \) function at a high frequency leads to a relatively large deviation in the frequency.

Instead of the first order derivative also the first order primitive can be used. This has the advantage that the scaling function is close to one for high frequencies, whereas the scaling
function for the first order derivative was close to one for low frequencies. Equation 4.39 describes the gain factor for the first order primitive:

\[ |H(e^{j\omega})| = 2F_s \cos \left( \frac{\omega}{2} \right) \]  

Equation 4.39 can also be combined with equation 4.37 to form a solution (Equation 4.40):

\[ \omega_p = 2 \arccos \left( \frac{DFT^{-1}[m_p]}{2DFT[0][m_p]} \right), \]  

where \( DFT^{-1} \) denotes the Discrete Fourier Transform of the first order primitive.

In order to assess whether the new frequency estimation technique performs sufficiently well two implementations of the algorithm above have been made:
1. Splitting the frequency axis into two, using Equation 4.38 for the frequencies \( \omega_p \leq \pi / 2 \) and Equation 4.40 for the frequencies \( \omega_p > \pi / 2 \).
2. Using Equation 4.41 for all frequencies.

Some small qualitative experiments showed that both frequency estimation methods didn't perform as well as the original iterative method. On forehand it was however assumed that the new non-iterative method could perhaps even improve the frequency estimation. Therefore, and because complexity reduction was no main goal in the graduation project, no further attention has been paid to the new extraction method. It is however noted that for further investigation the following issues could be desirable:
- Other filters can possibly improve the frequency estimation. Perhaps even a simple delay could be applied.
- The length of both the analysis window as well as the accompanying FFT have great impact on the resolution of the first stage peak search.
5 PSYCHO-ACOUSTIC MODEL

5.1 INTRODUCTION

The psycho-acoustic model is the key element in removing the irrelevancy in an audio coder. The more precise a psycho-acoustic model matches the human auditory system the better irrelevant information can be omitted and the higher the compression gain can be. Within SSC currently only frequency masking is exploited. The frequency-masking model in the SSC coder is based on a constant (0th order) sinusoids description [Oomen et al. 1999]. In this chapter the adaptation of the psycho-acoustic model to the higher order polynomials is described.

5.2 FREQUENCY MASKING

Frequency masking within the human auditory system is a very complex process. However a lot of research has been done on this subject. Numerous models and implementations have been proposed for frequency-masking models in waveform coders. They are however all more or less based on the concept depicted in Figure 5.1.

First of all, a segment of the PCM input signal is converted to power as a function of frequency. This is mostly done by means of a combination of windowing and an FFT. When the power has been calculated for each frequency component, it is converted to Sound Pressure Level in decibels (dB SPL).

In the perception of sound a distinction is made between tonal components and non-tonal components. When e.g. a tonal component has been detected by the human auditory system, distortions around that tonal component can be perceived more easily. Whether a component is tonal or not is not detected instantly, but rather gradually over time. A tonality measure is calculated in the frequency-masking model to take such effects into account.

The frequency-masking curve of a single spectral component is determined by the in-band masking, the upper slope and the lower slope on the Bark-scale (see Fig. 5.2) [Zwicker 1961]. The Bark-scale is quite similar to the ERB-scale. The main difference is that on the Bark-scale the critical bandwidths are of constant width below 500 Hz, while on the ERB-scale the critical bandwidths keep decreasing towards lower frequencies.

The in-band masking determines the maximum power masked by a spectral component (see Fig. 5.2). It is mainly determined by the amount of tonality, by frequency and, in case of a tonal component, by duration. For every spectral component the in-band masking is determined.
Before the masking curves are actually calculated, the Sound Pressure Levels are converted to sensation levels. A sensation level (in dB) is the amount of power that a spectral component is above the hearing threshold in quiet. The power thus obtained, better describes the power that actually excites the inner ear.

When the sensation levels have been determined the spreading in the frequency domain of these levels can be calculated resulting in an upper and a lower slope for each frequency component (see Fig. 5.2). Finally all these contributions are added together to form the total masking curve. This curve is compared to the threshold in quiet and the larger of the two is selected for each frequency (see Figure 2.17).

5.2.1 Frequency masking of sinusoids

The block-diagram as depicted in Figure 5.1 is typical for a waveform coder. Within SSC the frequency-masking model is only used for sinusoids, therefore a few changes can be made:

- The Sound Pressure Levels can now be calculated based on the power of individual sinusoids.
- There is no apparent need for a tonality index. It is assumed that all sinusoids that have been extracted are indeed tonal.

A block-diagram of frequency masking for SSC is given in Figure 5.3.

The power of a single sinusoid is defined as (Equation 5.1):

\[
P_p = \frac{1}{N} \sum_{n=-N/2}^{N-1} \left( s_p[n] \right)^2, \tag{5.1}
\]

where, for constant sinusoids, \( s_p[n] \) describes a single sinusoid as (Equation 5.2):
\[ s_p[n] = ae^{j\omega n} + ae^{-j\omega n}, \]
\[ = 2|a|\cos(\omega n + \varphi_a), \]
\[ = A\cos(\omega n + \varphi_a). \]

Using the square of the sinusoid (Equation 5.3):
\[ (s_p[n])^2 = \frac{1}{2} A^2 (1 + \cos(2\omega n + 2\varphi_a)), \]
and the sum for an arbitrary cosine function (Equation 5.4):
\[ \sum_{n=-N/2}^{N/2-1} \cos(2\omega n + \varphi) = \sum_{n=-N/2}^{N/2-1} \frac{1}{2} \left( e^{j(2\omega n + \varphi)} + e^{-j(2\omega n + \varphi)} \right), \]
\[ = \frac{1}{2} e^{j\varphi} \sum_{n=0}^{N-1} e^{j2\omega n} + \frac{1}{2} e^{-j\varphi} \sum_{n=0}^{N-1} e^{-j2\omega n}, \]
\[ = \frac{1}{2} e^{j\varphi} e^{-j\omega(N-1)} \sum_{n=0}^{N-1} e^{j2\omega n} + \frac{1}{2} e^{-j\varphi} e^{-j\omega(N-1)} \sum_{n=0}^{N-1} e^{-j2\omega n}, \]
\[ = \frac{\cos \varphi}{(1 - \cos(2\omega))} \left( \cos(\omega(N-1)) - \cos(\omega(N+1)) \right), \]
\[ = \frac{\sin(\omega N + \varphi) + \sin(\omega N - \varphi)}{2\sin \omega}. \]

The power of a 0th order sinusoid is equal to (Equation 5.5):
\[ P_p = \frac{1}{2} A^2 \left( 1 + \frac{\sin(\omega N + 2\varphi_a) + \sin(\omega N - 2\varphi_a)}{2\sin \omega} \right). \]

The right hand side of Equation 5.5 can be seen as compensation for the fact that the number of periods that is contained within a segment is not an integer.

Using Equation 5.5, and thus only 0th order information, the psycho-acoustic model has been tested on the second order polynomial description (Equation 4.7). For most excerpts there were no perceptual differences noticeable before and after the psycho-acoustic model. However especially for excerpts that contain clear transients, like e.g. the castanets excerpt, differences were clearly noticeable, i.e. too much information is discarded. It is noted here that for transients, in essence, a frequency-masking model isn't valid. This is because of the assumption of (quasi-) stationarity. This statement can however be alleviated by inclusion of the power of the 1st and 2nd order polynomials in Equation 5.1. So, \( s_p[n] \) becomes (Equation 5.6):
\[ s_p[n] = ae^{j\omega n} + ae^{-j\omega n} + bne^{j\omega n} + bne^{-j\omega n} + cn^2 e^{j\omega n} + cn^2 e^{-j\omega n}, \]
\[ = 2|a|\cos(\omega n + \varphi_a) + 2|b|n \cos(\omega n + \varphi_b) + 2|c|n^2 \cos(\omega n + \varphi_c), \]
\[ = A\cos(\Omega_a) + Bn \cos(\Omega_b) + Cn^2 \cos(\Omega_c), \]
where $\Omega_a, \Omega_b$ and $\Omega_c$ are defined as the instantaneous phases (Equation 5.7):

$$\Omega_a = \omega n + \varphi_a,$$

$$\Omega_b = \omega n + \varphi_b,$$

$$\Omega_c = \omega n + \varphi_c.$$  

The square of $s_p[n]$ then becomes (Equation 5.8):

$$\begin{align*}
\left(s_p[n]\right)^2 &= \left(A \cos(\Omega_a) + Bn \cos(\Omega_b) + Cn^2 \cos(\Omega_c)\right)^2, \\
&= A^2 \cos^2(\Omega_a) + B^2 n^2 \cos^2(\Omega_b) + C^2 n^4 \cos^2(\Omega_c) + 2ABn \cos(\Omega_a) \cos(\Omega_b), \\
&\quad + 2ACn^2 \cos(\Omega_a) \cos(\Omega_c) + 2BCn^3 \cos(\Omega_b) \cos(\Omega_c).
\end{align*}$$  

In Equation 5.8 clearly the individual power terms can be discerned from the cross-power terms. Using goniometric relations this equation can be rewritten to (Equation 5.9):

$$\begin{align*}
\left(s_p[n]\right)^2 &= \frac{1}{2} A^2 + \frac{1}{2} B^2 n^2 + \frac{1}{2} B^2 n^2 \cos(2\Omega_b), \\
&\quad + \frac{1}{2} C^2 n^4 + \frac{1}{2} C^2 n^4 \cos(2\Omega_c) + ABn \cos(\Omega_a + \Omega_b) + ABn \cos(\Omega_a - \Omega_b), \\
&\quad + ACn^2 \cos(\Omega_a + \Omega_c) + ACn^2 \cos(\Omega_a - \Omega_c), \\
&\quad + BCn^3 \cos(\Omega_b + \Omega_c) + BCn^3 \cos(\Omega_b - \Omega_c).
\end{align*}$$  

Equation 5.9 has the advantage that for every term the summation as in Equation 5.1 can be performed quite easily. The total solution is given in Appendix A.

After the power per sinusoid, $P_p$, has been calculated it is converted from linear power to dB SPL by means of (Equation 5.10):

$$P_{p,SPL} = 10 \log_{10} P_p + G,$$  

where $G$ is a correction factor that corresponds to the level of a signal with power one. In practice, $G$ depends on the sound reproduction level. Therefore no definite value for $G$ exists. Within SSC $G$ is chosen such that the power of a full-scale sinusoid yields a level of 96 dB SPL. This results in a correction factor of 8.73 dB.

In the next step, for every sinusoid the masking function is calculated using $P_{p,SPL}$. The masking function is represented on a Bark scale by means of the inband-masking and two linear slopes, a lower slope and an upper slope (see Figure 5.2). The lower slope is independent of frequency and power and equals $s_l = -22$ dB/Bark. The upper slope is given as:

$$s_u = 1.5 \log_{10} P_p - \frac{1}{0.3913b_p} - 18,$$  

where $s_u$ is expressed in dB/Bark and $b_p$ the frequency of the $p^{th}$ sinusoid expressed in Bark (Equation 5.12):

$$b_p = 0.6 \cdot 21.4 \cdot \log_{10} \left(\frac{4.37f_p}{1000} + 1\right).$$
where \( f_p \) is the frequency of the \( p \)th sinusoid expressed in Hertz.

The in-band masking for tonal components is a function of frequency and duration. However, currently within SSC these dependencies aren’t included. The in-band masking is therefore approximated with \( m_p = -10 \text{dB} \). The power spectral density function \( P_p \) can thus be corrected as (Equation 5.13):

\[
X_p = P_p + m_p.
\]

5.13

The hearing threshold in quiet can, to some extent, be seen as a prefiltering operation preceding the spreading in the inner ear. The spreading thus should be applied on Sensation Levels rather than absolute levels (Equation 5.14):

\[
X_p = X_p - h_p,
\]

5.14

where \( X_p \) is expressed as sensation level (dB) and \( h_p \) is the hearing threshold in quiet given by (Equation 5.15):

\[
h_p = 3.64 \left( \frac{f_p}{1000} \right)^0.8 - 6.5e^{-0.6 \left( \frac{f_p}{1000} - 3.3 \right)} + 0.007 \left( \frac{f_p}{1000} \right)^3,
\]

5.15

where \( h_p \) is expressed in dB and \( f_p \) is expressed in Hertz.

In the next step the spreading of the masking functions is performed. After the spreading is performed the absolute threshold of hearing is added again to neutralise Equation 5.14:

\[
X_p = X_p + h_p.
\]

5.16

Finally, the masking threshold \( X_p \) is compared to the hearing threshold in quiet and the larger of the two is selected for every sinusoid (Equation 5.17):

\[
M_p = \max(X_p, h_p)
\]

5.17

where \( M_p \) is the total masking power at the \( p \)th sinusoid.

When the total masking curve has been calculated for a frame, for every sinusoid it is checked whether it falls below the masking threshold. At first, sinusoids that fall below the masking threshold are removed. This removal will however alter the total masking threshold. Therefore, the psycho-acoustic model is performed once again to check whether the removed components still fall below the masking threshold after removal. If sinusoids that had been removed in the second stage don’t fall below the masking threshold, they are placed back.
5.3 EXPERIMENTS

The power of the second-order polynomials has been included in the psycho-acoustic model in order to assess whether this improves the behaviour around transients. The implementation was tested using an informal listening test. In this test the sinusoidal signals just before and just after the psycho-acoustic model were compared to each other. This experiment showed the following:

- There seems to be no loss of perceptual quality, unlike the case where only the 0th order information was included in the psycho-acoustic model.
- The number of sinusoids that are removed decreased comparing to the case of only 0th order information with only a few percent. This is consistent with the statement that at transient positions too much sinusoids were discarded.

It may be concluded that, if the sinusoids after the psycho-acoustic model are encoded perceptually lossless, near-transparent audio quality can still be achieved.
6 TRACKING ALGORITHM

6.1 INTRODUCTION

The tracking algorithm is both essential in terms of bit-rate reduction and perceived audio quality. In principle the linking of individual sinusoids as it is, only has influence on the bit-rate. However, since inextricably bound up with the phase-less reconstruction within a track, the quality is also affected. As indicated in Chapter 3, currently the phase-less reconstruction and the removal of non-tracks cause a significant perceptual loss of quality.

In this chapter a new approach to the tracking algorithm is presented, based on the information obtained from the second order polynomial description of sinusoids. This results in a decreased loss of quality. Also adjustments to the algorithm that removes non-tracks are described.

6.2 LINKING OF SINUSOIDAL COMPONENTS

The goal of the tracking algorithm is to link individual sinusoids between consecutive frames. This has two purposes. First of all, a track is an efficient representation of a sinusoid over time. The frequency and amplitude can e.g. be coded differentially. Secondly, if tracks represent the true course of the sinusoids over time, the phase information becomes redundant and can therefore be discarded. If however incorrect links have been made, this will cause audible discontinuities.

When observing spectrograms, like e.g. Figure 2.2 and 2.3, the process of linking seems pretty straightforward. In practice however some problems occur. To illustrate what problems can occur a synthetic signal is generated (Equation 6.1):

\[ s[n] = \sum_{p=1}^{P} \frac{A}{2p-1} \sin((2p-1)(\omega + \omega_d \sin(n \omega_m)))n, \]

where \( p \) indexes the \( p^{th} \) sinusoid, \( P \) describes the total number of sinusoids, \( A \) is the amplitude of the fundamental harmonic, \( \omega \) the frequency of the fundamental harmonic, \( \omega_d \) the modulation depth and \( \omega_m \) the modulation frequency. Figure 6.1 shows a part of the spectrogram of this signal for \( A = 1000, P = 30, \omega = 100F_s / 2\pi, \omega_d = 1.5F_s / 2\pi \) and \( \omega_m = F_s / 2\pi \).

The signal as described in Equation 6.1 basically describes an approximation of a frequency modulated square-wave by thirty sinusoids. As can be seen in Figure 6.1, the amount of frequency modulation increases over time. Visually the thirty spectral lines, each representing a single sinusoid, are easily followed. The tracking of this signal thus appears to be straightforward. Two types of problems however occur:
1. Wrong connections: two sinusoids of consecutive frames get connected while they should not have been.
2. Non-connection: two sinusoids of consecutive frames don’t get connected while they should have been.
Figure 6.2 shows how the current tracking mechanism, using Equations 2.6, 2.7 and 2.8, connects the extracted sinusoids for a small part of the signal in Figure 6.1. The solid lines describe tracks, where the crosses denote births (beginnings) and deaths (endings) of tracks.

In Figure 6.2 the problems described above are clearly visible. Especially high frequency variations cause erroneous linking of sinusoids. This can be explained by the cost-function (Equation 2.8). One of the partial cost-functions consists of a criterion, based on the difference in frequency of sinusoids in subsequent frames. This causes the linking mechanism to have a tendency to go straight on. This is depicted in Figure 6.3: two subsequent frames are shown as planes in a three-dimensional environment. The x-axis describes time, the y-axis the
amplitude and the z-axis (going into the paper) describes frequency. The dots on these planes
denote sinusoidal components. For the \( p^{th} \) sinusoid in frame \( k \) the cost-function is visualised as
a circle. Sinusoids that fall outside of the circle will not be linked to the sinusoid. For all
sinusoids that fall within the circle the sinusoid that is closest to the centre of the circle will be
chosen as link. In Figure 6.3 this is depicted using grey values, where a darker grey gives
preference over a lighter grey.

![Figure 6.3: 'Straight ahead'-property of current linking mechanism](image)

Besides the 'straight ahead'-property, Figure 6.3 also indicates another weakness of the
current linking mechanism: time information isn't taken into account during the linking process.
The analysis of sinusoidal components took place using 50% overlap. This means that on an
average basis the matching of sinusoids is at best at half the overlap. Ideally the linking should
thus be performed there. The only time-information that is available, the phases of the
sinusoids, is currently not exploited.

The linking mechanism can be drastically improved using the information obtained from
the second order polynomial sinusoid description. For this purpose a sinusoid is written once
again as the real part of three complex vectors (Equation 6.2):

\[
s_p[n] = \Re \left\{ (a + bn + cn^2) e^{jo \omega} \right\},
\]

6.2

This description of a sinusoid is depicted in Figure 6.4. In this figure the three complex vectors
\( a e^{jo \omega}, b e^{jo \omega} \) and \( c n^2 e^{jo \omega} \) of Equation 6.2 are shown on the complex plane for two
consecutive time instances. The mapping on the real axis of the total vector, given as the sum
of the three complex vectors, describes \( s_p[n] \). Note that even though each complex vector
rotates with frequency \( \omega \), the total vector, given as the sum of the three complex vectors, does
not necessarily rotate with this same frequency \( \omega \). This figure thus also proves that frequency
variation can be handled using the three complex vectors.
The three complex vectors of Equation 6.2 can also be seen as a single complex vector consisting of the sum of these three vectors. Such a description was already used in Chapter 4:

\[
\text{s}_p[n] = \Re\{C(n)\},
\]
\[
= \Re\{R(n)e^{j\phi(n)}\},
\]

where \(C(n)\) denotes the complex vector representing the whole signal, \(R(n)\) the envelope function and \(\phi(n)\) the instantaneous phase function. As an example a sinusoid is shown graphically as a single complex vector in Figure 6.5.

The upper figure of Figure 6.5 shows the waveform as a function of time. The lower figure shows complex samples of this same waveform denoted using arrows. Each arrow describes a sample of the waveform in its complex form (see Equation 6.3). Figure 6.5 tries to indicate that the complex description using the parameters: instantaneous phase,
Quality Scalability of a Parametric Audio Coder
Eindhoven University of Technology

instantaneous frequency and instantaneous amplitude (envelope) is no different from the waveform description. The shape of the waveform of a single sinusoid is thus fully described by Equation 6.3. The instantaneous amplitude \( R(n) \) is the time-dependent equivalent of the amplitude \( A \), the instantaneous phase \( p(n) \) is the time-dependent equivalent of the phase \( \varphi \) and the instantaneous frequency \( dp(n)/dn \) is the time-dependent equivalent of the frequency \( \omega \). The second-order polynomial thus gives extra information on how the amplitude, phase and frequency vary over time, within a segment. This extra information can be used to improve the linking mechanism.

In the case a sinusoid is part of a track, there should be a close match of waveforms of sinusoids in consecutive frames. Figure 6.6 shows two waveforms of overlapping frames with such a close match at the overlap region. A possible linking criterion account for the amount of matching found between sinusoids at the overlap region. Equation 6.3 is ideal for such a purpose because it describes the three independent complex vectors of a single sinusoid as one. The question now remains how to define a matching criterion on \( C(n) \).

![Figure 6.6: Waveform matching in overlapping frames](image)

Now, fully analogous to the current linking mechanism, the following cost-function has been defined (Equation 6.4):

\[
Q_{p,q} = Q_{p,q}^R Q_{p,q}^P Q_{p,q}^\Omega.
\]

6.4

where the subscript \( p \) denotes the \( p \textsuperscript{th} \) sinusoid of frame \( k \) and subscript \( q \) denotes the \( q \textsuperscript{th} \) sinusoid of frame \( k - 1 \). When normal windows are applied, the cost-function is applied at the middle of two overlapping segments. When start- or stop-windows are used, the cost-function is applied at the edges. This is shown graphically in Figure 6.7. So, instead of the current cost-function that makes use of global values, now instantaneous values are used.
The cost-function for the instantaneous amplitude has been defined as (Equation 6.5):

\[
Q^R_{p,q} = \begin{cases} 
0 & \text{for } |R_{p,k} - R_{q,k-1}| \geq R_{\text{max}} \\
1 - \frac{|R_{p,k} - R_{q,k-1}|}{R_{\text{max}}} & \text{for } |R_{p,k} - R_{q,k-1}| < R_{\text{max}},
\end{cases}
\]  

\[6.5\]

where \( R \) denotes the instantaneous amplitude expressed in decibels and \( R_{\text{max}} \) denotes the maximally allowed deviation in decibels. Both are expressed in decibels to match the human auditory system.

The cost-function for the instantaneous frequency is defined as (Equation 6.6):

\[
Q^\Omega_{p,q} = \begin{cases} 
0 & \text{for } |e(\Omega_{p,k}) - e(\Omega_{q,k-1})| \geq e(\Omega_{\text{max}}) \\
1 - \frac{|e(\Omega_{p,k}) - e(\Omega_{q,k-1})|}{e(\Omega_{\text{max}})} & \text{for } |e(\Omega_{p,k}) - e(\Omega_{q,k-1})| < e(\Omega_{\text{max}}),
\end{cases}
\]  

\[6.6\]

where \( e(\cdot) \) denotes the frequency expressed in ERB. The instantaneous frequency is approximated using a first order difference (Equation 6.7):

\[
\begin{align*}
\Omega_{p,k} &= p_{p,k}(n_{\text{overlap},k} + 1) - p_{p,k}(n_{\text{overlap},k}) \\
\Omega_{q,k-1} &= p_{q,k-1}(n_{\text{overlap},k-1} + 1) - p_{q,k-1}(n_{\text{overlap},k-1}),
\end{align*}
\]  

\[6.7\]

where \( n_{\text{overlap},k} \) denotes the overlap position of a frame \( k \) at the left-hand side, \( n_{\text{overlap},k-1} \) denotes the overlap position of a frame \( k - 1 \) at the right-hand side as shown in Figure 6.7, and \( p_{p,k}(n) \) the instantaneous phase of the \( p^{\text{th}} \) sinusoid in frame \( k \) at position \( n \).

The cost-function for the instantaneous phase is defined as (Equation 6.8):

\[
Q^p_{p,q} = \begin{cases} 
0 & \text{for } |p_{p,k} - p_{q,k-1}| \geq p_{\text{max}} \\
1 - \frac{|p_{p,k} - p_{q,k-1}|}{p_{\text{max}}} & \text{for } |p_{p,k} - p_{q,k-1}| < p_{\text{max}},
\end{cases}
\]  

\[6.8\]

where \( p_{p,k} \) and \( p_{q,k-1} \) are defined as (Equation 6.9):
$P_{p,k} = P_{p,k}(n_{\text{overlap},k})$

$P_{q,k-1} = P_{q,k-1}(n_{\text{overlap},k-1})$

where $n_{\text{overlap},k}$ and $n_{\text{overlap},k-1}$ are defined as described above.

Experiments using the partial cost-functions as described above showed that the partial cost-function for the instantaneous frequency (Equation 6.6) sometimes behaved unpredictably. Some further research showed that this happened especially at areas where the envelope-function is close to zero. Figure 6.8 shows an example of such behaviour. The upper figure shows an amplitude-modulated sinusoid as a function of time. The lower figure shows the instantaneous frequency that belongs to the waveform of the upper figure. As can be seen quite clearly the instantaneous frequency around position zero becomes negative. In itself this isn’t a problem; both sinusoids that are to be linked will show such behaviour. Some experiments however showed that the matching of waveforms at such critical areas in general isn’t very good. This especially affects the instantaneous frequency. Therefore the cost-function of the instantaneous frequency was replaced with the cost-function for the (modulation) frequency $\omega$.

![Figure 6.8: Amplitude modulated sinusoids and its instantaneous frequency](image)

The cost-function for the frequency then becomes identical to the current cost-function for the frequency (Equation 6.10):

$$Q_{p,q}^\omega = \begin{cases} 0 & \text{for } |e(\omega_{p,k}) - e(\omega_{q,k-1})| \geq e(\omega_{\text{max}}) \\ 1 - \frac{|e(\omega_{p,k}) - e(\omega_{q,k-1})|}{e(\omega_{\text{max}})} & \text{for } |e(\omega_{p,k}) - e(\omega_{q,k-1})| < e(\omega_{\text{max}}). \end{cases}$$

6.10

The partial cost-functions of Equation 6.5, 6.8 and 6.10 contain respectively the thresholds $R_{\text{max}}$, $P_{\text{max}}$, and $e(\omega_{\text{max}})$, which are still to be determined. At first this has been done
in such a way, that for almost every track that is clearly visible in a spectrogram, links are indeed made.

Secondly an optimisation towards quality has been made. For this purpose the values of \( R_{\text{max}} \), \( p_{\text{max}} \) and \( e(\omega_{\text{max}}) \) have been slightly adjusted for an optimum between quality and amount of links. A smaller amount of links means that more births will occur. Every birth is coded including its original phase. This in comparison to a continuation costs more bits. However, the quality in general will increase, but so will the bit-rate. A balance has been sought between the amount of links and the perceived audio quality. This led to the following values:

\[
e(\omega_{\text{max}}) = 0.5 \text{ erb}, \quad p_{\text{max}} = 1/3\pi \text{ radians} \quad \text{and} \quad R_{\text{max}} = 12 \text{ dB}.
\]

As an example, Figure 6.2 is depicted once again, but now with improved linking as described above (see Figure 6.9). The solid lines once again denote tracks that have been formed; the crosses denote births and deaths of tracks. When compared to Figure 6.2 the improvement is quite obvious. However still some erroneous links are made and sometimes no connection is made at all.

![Figure 6.9: Linking on synthetic signal of Equation 6.1](image-url)
6.3 CONTINUATION OF PHASE

In paragraph 2.5.3 the following equation was derived for the continuation of the phase (Equation 6.11):

\[ \varphi_{q,k} = \left( \omega_{p,k-1} + \omega_{q,k} \right) \frac{N}{4} + \varphi_{p,k-1}. \]

This equation was derived from the concept of constant sinusoids. As seen in Chapter 4 to a certain extent frequency variation can also be modelled. As an approximation the \( p^{th} \) sinusoid in frame \( k \) then is described as (Equation 6.12):

\[ s_{p,k}(n) = A_{p,k} \cos \left( (\omega_{p,k} + \Delta \omega_{p,k} n) + \varphi_{p,k} \right) \]

where \( \Delta \omega_{p,k} \) describes linear frequency variation.

The \( q^{th} \) sinusoid in frame \( k-1 \) can then be described on the same time-axis as Equation 6.12 as:

\[ s_{q,k-1}(n) = A_{q,k-1} \cos \left( (\omega_{q,k-1} + \Delta \omega_{q,k-1} (n + \frac{N}{2}) + \frac{N}{2} + \varphi_{q,k-1} \right). \]

Suppose that both sinusoids of Equation 6.12 and Equation 6.13 have been linked together. To obtain a smooth transition between the two sinusoids, the instantaneous phases must be equated (Equation 6.14):

\[
(\omega_{p,k} + \Delta \omega_{p,k} n) + \varphi_{p,k} = (\omega_{q,k-1} + \Delta \omega_{q,k-1} \left( n + \frac{N}{2} \right) + \frac{N}{2} + \varphi_{q,k-1}.
\]

Now suppose the two segments will be linked in the middle of the overlap \( n=-N/4 \), as is the general case (Equation 6.15):

\[
-\left( \omega_{p,k} - \Delta \omega_{p,k} \frac{N}{4} \right) + \varphi_{p,k} = (\omega_{q,k-1} + \Delta \omega_{q,k-1} \left( -\frac{N}{4} + \frac{N}{2} \right) - \frac{N}{4} + \frac{N}{2} + \varphi_{q,k-1}.
\]

Using Equation 6.15 the following can be derived (Equation 6.16):

\[
\varphi_{p,k} = \frac{N}{4} (\omega_{p,k} + \omega_{q,k-1}) + \frac{N^2}{16} (\Delta \omega_{q,k-1} - \Delta \omega_{p,k}) + \varphi_{q,k-1}.
\]

For the two terms \( \Delta \omega_{q,k-1} \) and \( \Delta \omega_{p,k} \) different approximations can be used:

- Forward and backward differences (Equation 6.17):

\[
\Delta \omega_{p,k}^F = 2 \frac{\omega_{p,k+1} - \omega_{p,k}}{N}, \quad \Delta \omega_{p,k}^B = 2 \frac{\omega_{p,k} - \omega_{q,k-1}}{N},
\]

\[
\Delta \omega_{q,k-1}^F = 2 \frac{\omega_{p,k} - \omega_{q,k-1}}{N}, \quad \Delta \omega_{q,k-1}^B = 2 \frac{\omega_{q,k-1} - \omega_{q,k-2}}{N}.
\]
• Approximations based on second order polynomials
  The approximations based on the second order polynomials are of no use for the decoder because the second order polynomials will not be included in the bit-stream.
  Different combinations of the forward and backward differences however might give better results. Therefore they have been evaluated using informal listening tests. For $\Delta \omega_{q,k}$ and $\Delta \omega_{p,k}$ respectively the following combinations have been evaluated: forward-forward, forward-backward, backward-forward and backward-backward. Note that the combination forward-backward leads to Equation 6.12. Also note that the forward difference of $\Delta \omega_{p,k}$ needs future information of the track while the backward difference of $\Delta \omega_{q,k}$ needs history of the track. Both differences can thus only be applied on a limited area of a track.
  In informal listening tests signals where generated using the four combinations as described above. None of the combinations proved superior to another. Therefore equation 6.12 was preferred over any other combination.

Without further discussion, in this chapter the assumption has been made that the continuation of the phase should always be performed from the birth of a track. There are however reasons to choose another initialisation point within a track:
• Most sinusoids will not exactly start at the segmentation boundaries. For such a sinusoid, the parameters can be estimated incorrectly. The same applies to the end of a sinusoid. This, in contrast to the current continuation of phase, indicates that a distance is to be kept from the beginning and the end of a track.
• The accumulation of errors will be smaller when the continuation is started from the middle.
• The encoder delay becomes larger when the continuation of the phase is performed from a point away from the start of a track. So, a point close to the beginning of a track is preferred. Note that for the decoder no such problems occur. The original phase can be calculated all the way back to the birth of a track. This is shown graphically in Figure 6.10. Just like Equation 6.11, the continuation of the phase one frame forward in time, the continuation of the phase one frame backwards in time can be calculated. So when a whole track has been determined starting with the original phase at frame $k$, the continued phase at frame $k-1$ can be calculated. From this frame, the continued phase at frame $k-2$ can be calculated, etc.

![Figure 6.10: Calculation of phase at birth from original phase at arbitrary position](image)

Another set of informal listening tests has been performed in order to assess the optimal place within a track of length $L$ for the original phase. The following configurations have been tested:
1. Original phase at birth of a track: at $L = 1$, just like the current continuation.
3. For tracks with $L > 10$: original phase placed at $L = 5$, for shorter tracks at $L = 1$.
4. For tracks with $L > 5$: original phase placed at $L = 3$, for shorter tracks at $L = 1$. 

Page 68 of 91
5. For all tracks with $L > 1$: original phase placed at $L = 2$, else at $L = 1$.

Although the perceptual differences where quite small, configuration 2,3,4 and 5 all outperformed configuration 1, as expected. For the others no preference could be given, therefore configuration 5 is preferred because of its short delay.

### 6.4 EVALUATION OF TRACKING MECHANISM

In order to evaluate the tracking algorithm described above, another set of informal listening tests has been performed. The tracking algorithm that is currently in the SSC coder, as described in Chapter 2, has been compared with respect to the algorithm described above. Both tracking algorithms have been used on the same set of extracted parameters for the second order polynomial extraction to give a fair comparison. For this purpose, once again the thirteen critical excerpts have been used. The following conclusions could be drawn:

- Although the new tracking algorithm drastically improves quality it surely doesn’t give transparent quality. Especially speech fragments still suffer from a large quality loss.
- The metallic or echo-like effect that was perceived using the tracking algorithm of the current SSC coder has been reduced drastically, however for very high frequency variations, as can be often found in speech, such artefacts still occur.
- The tracking algorithm still causes some loss of detail in the sound.
- Excerpts containing clear harmonics (e.g. speech and harpsichord) still suffer quality loss from minor discontinuities like shortly interrupted tracks.

All in all the quality can be drastically improved using the current tracking mechanism. However (near-)transparency doesn’t seem feasible using constraints of the same order as in the improved linking mechanism described above. Stricter linking-constraints will however increase the bit-rate which is unwanted.

### 6.5 REMOVAL OF NON-TRACKS

Non-tracks are sinusoidal components that have not been linked, i.e. their birth and death takes place in the same frame. Such components are expensive in terms of bit-rate, while perceptually they are often irrelevant. In order to achieve low bit-rates, removal therefore is necessary.

From the field of psycho-acoustics there is not much knowledge on perceptual relevance of non-tracks. It is however known that sinusoids consisting of less than five periods aren’t perceived as being sinusoidal. Therefore in the current removal-algorithm all sinusoids with a track-length of one and all tracks consisting of less than five sinusoidal periods are discarded.

In Chapter 3 it was already noted that the removal of non-tracks caused a perceptual loss of quality, especially around transient positions. This in essence means that too much sinusoidal elements are discarded. Two main reasons can be given:

- Transients typically contain of a lot of short sinusoids. This can be seen in the spectrograms (see Chapter 1) where transients typically show up as vertical lines.
- As described above, sinusoids consisting of less than five periods aren’t perceived as being sinusoidal. This however doesn’t mean they may be left out without consequences.

In order to improve the quality of the removal-algorithm a concession towards the bit-rate will have to be made. This is done by means of the inclusion of a psycho-acoustic model. First of all on a frame-to-frame basis the masking curve is calculated with a low in-band masking. This is shown in Figure 6.11 and 6.12.
In Figure 6.11 the solid lines denote tracks; the crosses denote births and deaths of frames. Crosses that aren't at the beginning or end of a line denote the non-tracks. Figure 6.12 shows the masking curve of the sinusoidal components of the selected frame of Figure 6.11. Two sinusoidal components fall under the masking curve. However only the sinusoidal component that is a non-track is removed.

Informal listening tests showed that this technique could be applied with an in-band masking of up to 0 dB without any loss of quality. This however means that especially around transients a lot more information (sinusoids) will have to be included which will lead to a higher bit-rate. This is described, among other subjects, in the next chapter.

Furthermore, the tracks that contain less than five periods are still removed. Informal listening tests showed that this did not further degrade the quality.

All in all the removal of the non-tracks had to be restricted in order to preserve the audio quality. In other words less sinusoids are removed, and thus more sinusoids have to be encoded. This will always lead to a higher bit-rate.
7 PERFORMANCE RESULTS

7.1 INTRODUCTION

During this graduation project a lot of changes to the encoder have been made. These changes affect both audio quality and bit-rate. In this chapter this is illustrated quantitatively. Bit-rates before and after the applied changes are compared and discussed. Furthermore the subjective audio quality has been compared by means of a listening test. The results of this test are also discussed.

7.2 BIT-RATES

In audio coding bit-rates are always very dependent on the statistics of the input signal. The set of excerpts that has been used for experimenting during this graduation project provides a broad representation of typical audio. Furthermore, these excerpts are known to be critical, i.e. in audio coding these excerpts are likely to cause audible artefacts. Table 7.1 shows a list of the excerpts used, including a brief description.

Table 7.1: List of all excerpts used during listening test

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Description</th>
<th>Fs (kHz)</th>
<th>Duration (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Castanets</td>
<td>Series of castanets snaps</td>
<td>44.1</td>
<td>7.28</td>
</tr>
<tr>
<td>2</td>
<td>Kraftwerk</td>
<td>Bassdrum</td>
<td>44.1</td>
<td>10.93</td>
</tr>
<tr>
<td>3</td>
<td>Abba</td>
<td>Pop</td>
<td>44.1</td>
<td>10.02</td>
</tr>
<tr>
<td>4</td>
<td>Trumpet</td>
<td>Trumpet solo, low BW: 8kHz</td>
<td>44.1</td>
<td>10.45</td>
</tr>
<tr>
<td>5</td>
<td>Celine Dion</td>
<td>Introduction to song</td>
<td>44.1</td>
<td>13.00</td>
</tr>
<tr>
<td>6</td>
<td>English fem. voice</td>
<td>Spoken sentence</td>
<td>44.1</td>
<td>6.84</td>
</tr>
<tr>
<td>7</td>
<td>German male voice</td>
<td>Spoken sentence</td>
<td>44.1</td>
<td>7.30</td>
</tr>
<tr>
<td>8</td>
<td>Metallica</td>
<td>Heavy metal</td>
<td>44.1</td>
<td>10.14</td>
</tr>
<tr>
<td>9</td>
<td>Harpsichord</td>
<td>Harpsichord</td>
<td>44.1</td>
<td>11.92</td>
</tr>
<tr>
<td>10</td>
<td>Contemporary</td>
<td>Pop</td>
<td>44.1</td>
<td>10.05</td>
</tr>
<tr>
<td>11</td>
<td>Susan Vega</td>
<td>Spoken (sung) vocal</td>
<td>44.1</td>
<td>10.27</td>
</tr>
<tr>
<td>12</td>
<td>Stravinsky</td>
<td>Orchestral piece</td>
<td>44.1</td>
<td>11.93</td>
</tr>
<tr>
<td>13</td>
<td>Orff</td>
<td>Choir</td>
<td>44.1</td>
<td>10.90</td>
</tr>
</tbody>
</table>

Within the SSC project the bit-stream currently consists of frames with a variable length each preceded by a header. Each such frame holds the information of nine sub-frames. These sub-frames correspond to the segments that are referred to in the previous chapters of this report. An example of the bit-stream is depicted in Figure 7.1. The fact that nine sub-frames are embedded within a single frame comes from the multi-scale analysis. During analysis of the lowest frequency scale one segment corresponds to 9 segments on the highest frequency scale.

![Variable framewidth](image-url)

Figure 7.1: Schematic description of bit-stream
During the graduation project basically only the sinusoidal module has been altered. The transient description however is expanded using the information of the 1st and 2nd order polynomials from the sinusoidal module. Experiments showed that inclusion of the 5 pairs of 1st and 2nd order terms (complex parameters b and c of Equation 4.8), with the most power at each transient improved the perceived quality of transients. The question thus remains how to incorporate this data into the bit-stream.

The following solution is proposed: for every pair of 1st and 2nd order terms that is to be included, already the frequency is included within the bit-stream for the sinusoids. Therefore, instead of actually encoding the frequency, an index to the frequency can be included. As there is a maximum of fifty sinusoids within a sub-frame, the frequency can always be indexed using 6 bits. Furthermore, just like the actual sinusoidal parameters, the complex 1st and 2nd order terms are split into amplitude and phase (see Equation 4.9). From psycho-acoustics there is however no specific knowledge available about how to encode these parameters. Therefore, as an experiment, the amplitude and the phase are quantised on the same grid as the amplitude and the phase of the sinusoidal parameters. This proved to be sufficiently accurate. Figure 7.2 shows how the extra transient information is included.

![Figure 7.2: Inclusion of 1st and 2nd order terms into bit-stream](image)

So for every transient an amount of 130 bits is added to the bit-stream. Table 7.2 shows for every excerpt the increase of bit-rate.
As stated before, the only changes that have been made to the encoder are within the sinusoidal module. Therefore the bit-rates corresponding to the extracted parameters for the transients and the noise module will not be influenced much. When estimating the total bit-rate, for both versions of the encoder the bit-rates of the original version are used for the transients and noise module. Table 7.3 shows estimates of the bit-rates common to the old and the new version of the encoder.

Table 7.3: Increase in bit-rate for all excerpts

<table>
<thead>
<tr>
<th>Excerpt index</th>
<th>Number of transients</th>
<th>Number of frames</th>
<th>Percentage (%)</th>
<th>Extra bit-rate (bits/second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>869</td>
<td>5.06</td>
<td>806</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>1301</td>
<td>4.69</td>
<td>747</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>1217</td>
<td>1.73</td>
<td>275</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1279</td>
<td>0.08</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>1577</td>
<td>2.16</td>
<td>343</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>830</td>
<td>2.05</td>
<td>326</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>889</td>
<td>1.24</td>
<td>197</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>1237</td>
<td>0.73</td>
<td>115</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>1455</td>
<td>0.69</td>
<td>109</td>
</tr>
<tr>
<td>10</td>
<td>33</td>
<td>1214</td>
<td>2.72</td>
<td>433</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>1249</td>
<td>1.76</td>
<td>281</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1459</td>
<td>0.41</td>
<td>65</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1335</td>
<td>0.07</td>
<td>12</td>
</tr>
<tr>
<td>Mean:</td>
<td></td>
<td></td>
<td></td>
<td>268</td>
</tr>
</tbody>
</table>

In order to incorporate the sinusoidal parameters into the bit-stream they are split into 15 different parameters:
1. Number of Continuations (NC): the number of continuations of the first sub-frame of a frame, this is needed for random access of encoded streams.
2. Number of Births (NB): the number of births within a sub-frame.
3. Continuation Length (CL): the length (within the current frame) of a track belonging to a birth.
4. Frequency Birth Absolute (FA): the absolute frequency of the first birth of a sub-frame, expressed in representation level.
5. Amplitude Birth Absolute (AA): the absolute amplitude of the first birth of a sub-frame, expressed in representation level.
6. Phase Birth Absolute (PA): the phase of the first birth of a sub-frame, expressed in representation level.
7. Frequency Birth Relative (FBR): the relative frequency of a birth, expressed in difference in representation level to the previous birth in the same sub-frame.
8. Amplitude Birth Relative (ABR): the relative amplitude of a birth, expressed in difference in representation level to the previous birth in the same sub-frame.
9. Phase Birth Relative (PBR): the relative phase of a birth, expressed in difference in representation level to the previous birth in the same sub-frame.
10. Frequency Continuation Absolute (FCA): the absolute frequency of a continuation, expressed in representation level, only used in case of random access playability.
11. Amplitude Continuation Absolute (ACA): the absolute amplitude of a continuation, expressed in representation level, only used in case of random access playability.
12. Phase Continuation Absolute (PCA): the absolute phase of a continuation, expressed in representation level, only used in case of random access playability.
13. Frequency Continuation Relative (FCR): the relative frequency of a continuation, expressed in difference in representation level to its predecessor in time.
14. Amplitude Continuation Relative (ACR): the relative amplitude of a continuation, expressed in difference in representation level to its predecessor in time.
15. Phase Continuation Relative (PCR): the relative phase of a continuation, expressed in difference in representation level to its predecessor in time, not used when continuous phase is applied.

A graphical interpretation of these 15 parameters is shown in Figure 7.3. In this figure tracks are denoted as curves. The crosses at the end of curves denote births and deaths of tracks. Crosses that aren't connected to a curve are the 'non-tracks'. As can be seen from Figure 7.3 the Birth Relative parameters are encoded differentially in the frequency domain while the Continuation Relative parameters are encoded differentially in the time domain. The Continuation Absolute parameters are only used at frame boundaries.

![Figure 7.3: Graphical representation of bit-stream parameters](image)

During the graduation project parameters 10, 11 and 12 aren't used. This in effect means that random access playing is not possible. Furthermore, parameter 15 isn't used because continuous phase has been applied. So, all in all eleven parameters are included in the bit-stream.

Figure 7.4 shows a simplified graphical representation of the bit-stream. At sub-frame \( k \), the first sub-frame in this example, there are three births (NB=3). A decoder would thus be provided with information where the continuations for sub-frame \( k \) would start in the bit-stream. For the first birth, the amplitude, phase and frequency are coded absolutely (AA, FA and PA). The amplitude, frequency and phase of the second birth get coded differentially to the first birth (FBR, ABR and PBR). The third birth on its turn gets coded differentially with respect to the
second birth. For every birth also the length of the track within the frame is included in the bit-stream. In this way no information about the number of continuations within a sub-frame is needed.

As there are no continuations within sub-frame \( k \) this concludes the bit-stream for the first sub-frame. The stream continues with the number of births within sub-frame \( k + 1 \). Once again the first birth is coded absolutely and the second is coded differentially. Finally the continuation is coded differentially to the first birth of frame \( k \).

In sub-frame \( k + 2 \) no births are present, therefore the bit-stream is resumed immediately with the continuations. Finally for sub-frame \( k + 3 \) the first birth is once again coded absolutely followed by the continuation (death) of the track started in sub-frame \( k + 1 \).

![Figure 7.4: Graphical representation of bit-stream](image)

For an efficient representation of the eleven parameters describing the sinusoids, Huffman tables will have to be constructed for each parameter. These can be easily constructed using histograms. Histograms describe how many times every possible value of a certain process occurs. Simply said, in order to generate the optimal Huffman table for a process, shorter codewords must be assigned to more probable values.

For three parameters of the list, 1, 6 and 9, no Huffman table has been generated. The first parameter, the number of continuations (NC), is always encoded with a fixed number of bits. Parameters 6 and 9, describing the phases of the sinusoids, both show an equally spread histogram such that Huffman coding gives no significant compression gain.

Figure 7.5 shows histograms for the eight parameters that are left summed over all thirteen critical excerpts for the original encoder. The x-axis of every figure denotes the representation level corresponding to the process, the y-axis denotes the number of occurrences for each representation level. As can be seen quite clearly, the figures on the right of Figure 7.5 show a much more peaked course than the ones on the left. The compression gain on these variables will thus be much larger than the more equally spread variables.
Figure 7.5: Histograms of the eight parameters for old encoder, *left* (top-down): FA, AA, NB, CL, *right* (top-down): FBR, ABR, FCR, ACR

Figure 7.6 shows the same eight figures as Figure 7.5 but now for the new encoder. Some clear differences can be observed when comparing both figures. First of all, comparing the Continuation Length histograms (lower-left) shows that the number of continuations of length one has increased dramatically. This is caused by the deletion of the non-tracks. The inclusion of a psycho-acoustic model causes that a lot more tracks of length one, the short 'non-tracks' are preserved. This leads to an overall increase of births. This is also illustrated by the Frequency Birth Relative (upper-right) and the Amplitude Birth Relative (second from above on the right). Both show an increased surface area. Furthermore the Amplitude Continuation Relative after the changes becomes less sharp (lower-right).
For both the histograms in Figure 7.5 and 7.6 the Huffman tables have been calculated. Using these tables, the bit-rate for each parameter averaged over all 13 excerpts has been calculated for both the old and the new encoder. The bit-rates and the difference in bit-rate for each parameter are shown in Table 7.4.

Table 7.4: Bit-rate for sinusoidal parameters averaged over critical excerpts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bitrate old (kbps)</th>
<th>Bitrate new (kbps)</th>
<th>Difference (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>0.0817</td>
<td>0.0817</td>
<td>+0.0000</td>
</tr>
<tr>
<td>NB</td>
<td>0.4406</td>
<td>0.4797</td>
<td>+0.0391</td>
</tr>
<tr>
<td>CL</td>
<td>2.4280</td>
<td>2.7926</td>
<td>+0.3646</td>
</tr>
<tr>
<td>FA</td>
<td>0.8754</td>
<td>0.9792</td>
<td>+0.1038</td>
</tr>
<tr>
<td>AA</td>
<td>0.4427</td>
<td>0.4750</td>
<td>+0.0323</td>
</tr>
<tr>
<td>PA</td>
<td>0.5205</td>
<td>0.5566</td>
<td>+0.0361</td>
</tr>
<tr>
<td>FBR</td>
<td>2.6641</td>
<td>3.6168</td>
<td>+0.9527</td>
</tr>
<tr>
<td>ABR</td>
<td>1.2312</td>
<td>1.7831</td>
<td>+0.5519</td>
</tr>
<tr>
<td>PBR</td>
<td>2.0111</td>
<td>2.6905</td>
<td>+0.6795</td>
</tr>
<tr>
<td>FCR</td>
<td>7.0584</td>
<td>6.9025</td>
<td>-0.1559</td>
</tr>
<tr>
<td>ACR</td>
<td>5.6562</td>
<td>7.2749</td>
<td>+1.6187</td>
</tr>
<tr>
<td>Total</td>
<td>23.41</td>
<td>27.84</td>
<td>+4.23</td>
</tr>
</tbody>
</table>
Table 7.4 clearly shows that the main increase in bit-rate is caused by all the relative birth variables, Frequency Birth Relative, Amplitude Birth Relative and Phase Birth Relative, and the Amplitude Continuation Relative variable. The increase in bit-rate of the relative birth variables can be mainly explained by the increase of births caused by the changes in the removal of the non-tracks. The increase in bit-rate of the Amplitude Continuation Relative is slightly more complex to explain. The improved tracking algorithm is the main cause for this. The old linking mechanism was partly based on the difference in amplitude of sinusoids in consecutive frames. Therefore sinusoids that were linked were sure to have amplitudes that are close to each other. Differences would thus always be small which automatically leads to a sharp peak in the histogram. The new linking mechanism uses instantaneous amplitudes for linking. This in effect means that the actual amplitude difference that is encoded might be much larger than the difference of the instantaneous values during linking. This causes that the peak spreads. This effect can be seen when comparing the lower-left figures of Figure 7.6 and Figure 7.6.

Finally for all excerpts in Table 7.1 the mean bit-rate has been calculated (see Table 7.5). Note that the mean values for the total bit-rate for sinusoids slightly differ from the values obtained above. This is mainly due to overhead that hasn’t been taken into account in Table 7.4.

Table 7.5: Mean bit-rate for every excerpt for old and new encoder

<table>
<thead>
<tr>
<th>Excerpt</th>
<th>Bit-rate sinusoids old (kbps)</th>
<th>Bit-rate sinusoids new (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27.1</td>
<td>32.7</td>
</tr>
<tr>
<td>2</td>
<td>24.0</td>
<td>27.2</td>
</tr>
<tr>
<td>3</td>
<td>25.6</td>
<td>31.1</td>
</tr>
<tr>
<td>4</td>
<td>14.2</td>
<td>16.0</td>
</tr>
<tr>
<td>5</td>
<td>21.3</td>
<td>25.2</td>
</tr>
<tr>
<td>6</td>
<td>23.0</td>
<td>26.1</td>
</tr>
<tr>
<td>7</td>
<td>23.0</td>
<td>27.1</td>
</tr>
<tr>
<td>8</td>
<td>31.9</td>
<td>38.8</td>
</tr>
<tr>
<td>9</td>
<td>13.1</td>
<td>17.0</td>
</tr>
<tr>
<td>10</td>
<td>30.7</td>
<td>36.6</td>
</tr>
<tr>
<td>11</td>
<td>24.8</td>
<td>28.2</td>
</tr>
<tr>
<td>12</td>
<td>26.1</td>
<td>30.4</td>
</tr>
<tr>
<td>13</td>
<td>24.7</td>
<td>27.9</td>
</tr>
<tr>
<td>Mean</td>
<td>23.6</td>
<td>27.7</td>
</tr>
<tr>
<td>Transients + Noise</td>
<td>+3.4</td>
<td>+3.4 + 0.3</td>
</tr>
<tr>
<td>Total</td>
<td>27.0</td>
<td>31.0</td>
</tr>
</tbody>
</table>

Table 7.5 shows that the changes that have been made to the encoder give an overall increase in bit-rate of about 4 kbit/s.
7.3 SUBJECTIVE LISTENING TEST RESULTS

In this report it has already been noted that an objective measurement of audio quality is hard to give. However, for comparison of audio codecs some standards have been developed. In order to evaluate the differences between the original and the adjusted codec, the seven-grade comparison scale of Recommendation BS.562 of the International Telecommunication Union (ITU-R BS.562) is used [ITU-R 1994]. This scale is depicted in Figure 7.7.

![7-grade comparison scale of ITU-R BS.562](image)

Figure 7.7: 7-grade comparison scale of ITU-R BS.562

In the listening test that has been performed subjects were presented a so-called ref-a-b test. In such a test for every excerpt the subject is presented with the reference (original) signal and two coded versions. The two coded versions are hidden, i.e. the subject does not know beforehand which of the two coded versions is codec a and which is codec b. After listening, the subject must grade the coded versions using the 7-grade scale of above. So, for every excerpt the subject has to give a comparative grade ranging from -3 until 3, where non-integer numbers are also allowed.

In order to improve reliability of a test, every subject must grade every pair of codecs twice for every excerpt, one time in the order ref-a-b and one time in the order ref-b-a. To further improve reliability of the results a number of ten subjects have actually performed the test.

The whole test has been performed using headphones rather than loudspeakers. Experience shows that listening-tests performed at the ASALab using headphones more clearly distinct small impairments between codecs.

In the test that has been performed, the SSC codec before and after the graduation project have been compared. The set of thirteen critical excerpts of Table 7.1 has been used for this purpose.

Figure 7.8 shows the results obtained from the ten subjects for every excerpt. For every excerpt a star, denoting the average grade, and a line-piece, denoting the 95% confidence interval, is shown. This interval shows with a probability of 95% where the mean is located. The small circle on the right denotes the average grade over all subjects and excerpts.
The following can be concluded from Figure 7.8:
- Excerpts 2 until 7 and excerpts 11 and 12 all clearly show an increase in quality.
- Excerpts 1, 8, 10 and 13 show no increase in quality.
- Excerpt 9 shows a slight decrease in quality.

After the test has been performed some people were asked for their opinion on the excerpts. The following was observed:
- For excerpt 1, the castanets, no quality improvement was observed as a cause of the extended transient representation using the $1^{st}$ and $2^{nd}$ order polynomials. Especially for this excerpt, consisting of a lot of transients, this was expected. The attacks are still not represented completely. Compared to the original, excerpt 1 still sounds noisy and of a different timbre.
- Excerpt 2, the Kraftwerk excerpt, clearly shows an improvement in both the attacks of the bass-drum as well as the attacks of the snare-drum. However, compared to the original, still some detail is lacking, especially in the snare-drums.
- Excerpt 3, the Abba excerpt, was perceived as less noisy. Also the voice was much clearer. Compared to the original once again some detail is lacking.
- The trumpets of excerpt 4 showed a small increase in quality. It was mainly observed as being less noisy than the old codec. This excerpt sounds very close to the original.
- Excerpt 5, the excerpt of Celine Dion, was perceived as less noisy. The singing also was much clearer. Compared to the original, the introduction of the excerpt, the drums, shows a lack in detail. Furthermore the voice still misses some presence.
- The female speech of excerpt 6 showed a large increase in quality. The metallic effect (also perceived as echo-like) was greatly reduced. However, compared to the original, still this effect could be observed.
For excerpt 7, the male speech, the same effect could be observed. The echo-like effect is however still more clearly present.

Excerpt 8, the Metallica excerpt, showed no demonstrable increase in quality. It was however still, in comparison to the original, perceived as being 'under water'.

The harpsichord excerpt, excerpt 9, showed a decrease in quality. This is caused by low-frequency 'stumbling' that became apparent in the new version. This, on its turn, is caused by the incorrect handling of very low, and in theory inaudible, frequencies. Both codecs model frequencies that fall below 15 Hz with a linear function. The new codec however removes these frequencies based on frequency masking laws. This excerpt indicates that this is incorrect.

Excerpt 10, the contemporary pop excerpt, showed no significant increase (or decrease) in quality. In comparison to the original a lot of detail is lacking.

The Susan Vega excerpt, excerpt 11, as all of the other speech fragments shows a clear improvement in quality. The metallic effect is once again decreased but still apparent when compared to the original signal.

Excerpt 12, Stravinsky, showed a slightly improved quality. It was mainly perceived as being less noisy. This quality of this excerpt is also very close to the original.

Excerpt 13, the Orff excerpt, showed no demonstrable increase in quality. Compared to the original it is still perceived as being a bit band-limited.

All in all it can be stated that for every excerpt the quality has increased or has been kept equal. The reason for the artefacts that show up in excerpt 9 are known, and the solution is already at hand. It is however noted that the following artefacts are still present within the improved codec:

1. The metallic or echo-like effect; caused by incorrect linking and amplified by the quantisation of the sinusoidal parameters.
2. Lack of detail; mostly caused by the linking and the removal of non-tracks.
3. Band-limiting; caused by the removal of non-tracks and/or a shortage of sinusoids during analysis.
8 CONCLUSIONS AND RECOMMENDATIONS

8.1 CONCLUSIONS

The listening-test of the previous chapter shows that the average quality over all excerpts has significantly improved. Excerpt number 9 is the only excerpt for which the 95% confidence interval falls below zero. The reason of this is described in the previous chapter and can be corrected quite easily.

There are many indications that the overall quality improvement is mainly caused by the inclusion of the 2nd order polynomials. Although not directly subjectively proven, because this inclusion influences many elements of the encoder, the following indications of improvement by the 2nd order polynomials are identified:

- During extraction, every spectral peak can now be modelled with a single sinusoid, i.e. one 2nd order polynomial, instead of multiple (constant) sinusoids as in the old situation. This has two positive side effects. First of all, because of the fixed number of sinusoids that are extracted, now the whole frequency spectrum gets treated more equally. Secondly, the residual signal will now contain less spectral peaks and will thus be noisier. The condition of the input-signal of the noise module will thus be favourable.

- As indicated in Chapter 4, the signal consisting of reconstructed quantised transient and noise parameters, and decoded unprocessed sinusoidal parameters gives near transparent quality. So, first of all, if the sinusoidal parameters could be encoded perceptually lossless, near-transparent quality can actually be achieved. This was not the case when constant (0th order) sinusoids were extracted. Secondly, this also means that the condition of the input-signal of the noise module has improved, as already indicated in the previous point.

- Visual observation of tracks of synthetic signals, as in Chapter 6, indicates that the operation of the linking algorithm seems to improve drastically when using the information obtained from the 2nd order polynomial information. This is confirmed by the experiments in Chapter 6. These experiments also showed that the use of continuous phase, with the current constraints in the linking mechanism, still does not deliver near-transparent quality. Furthermore, as shown in the previous chapter, the improvement of the tracking algorithm costs about 2 kbit/s extra bit-rate. Correct tracking thus costs extra bit-rate.

- The 2nd order polynomial description can handle amplitude variations within a segment. Therefore it is able to model that part of a transient that could not be modelled by the transients module. For high quality encoding this information could then be included, as is shown in Chapter 7. This however indicates that currently the transient representation is not sufficient.

The inclusion of the 2nd order polynomial description in the sinusoidal analysis and tracking was currently incorporated into the SSC reference coder.

Apart from the changes induced by the 2nd order polynomials, also the changes to the algorithm that removes the non-tracks, influences the quality:

- The inclusion of the psycho-acoustic model in this algorithm only seems to deliver a marginal improvement in quality. However, especially around transients some quality improvement is observed. This once again shows that the current transient representation is not sufficient. Furthermore, the inclusion of the psycho-acoustic model, in the algorithm that removes non-tracks, costs about 2 kbit/s extra bit-rate.
Experiments on the improved encoder showed that the quantisation of the sinusoidal parameters becomes more and more a bottleneck for the quality. Before the improvements, the quantisation only seemed to cause a very small loss of quality; all other artefacts that occurred masked the artefacts caused by the quantisation. As the quality of the other modules improved, it became clear that the current quantisation is too coarse, especially for harmonic content that is present in e.g. speech.

Summarising, the remaining gap towards transparent quality is caused by the following 3 issues:

1. **The transient representation still is not sufficient.** The fact that transients in waveform coders cost a lot of bits while transients within SSC cost only a very small amount of bits seems to be, at least partially, a contradiction. Non-optimal segmentation of waveform coders causes pre-echoes. The suppression of these pre-echoes is an important reason that transients cost a lot of bits.

2. **The continuation of the phase in the tracking mechanism still causes quality loss.** A possible reason is that during linking the constraints of the cost-functions for frequency, amplitude and phase are set too loosely. This causes that too many sinusoids are linked together. This in effect causes that incorrectly linked sinusoids form a smooth track, while they actually were not. The true course of sinusoidal tracks within typical audio should be used as guidelines for the cost-functions. The question however remains when a spectral peak may be considered a sinusoid and when not. Also, the assumption that continuation of phase may actually be applied on correct tracks has never been formally verified.

3. **The quantisation of the sinusoidal parameters is too coarse.** However for some excerpts, especially with harmonic content, this causes more audible artefacts than for others.

### 8.2 RECOMMENDATIONS

At the end of the graduation project still a lot of open ends remain. Both on the subject of complexity reduction and quality improvement work remains. In the previous paragraph already 3 remaining issues were mentioned. For these 3 issues the following recommendations are given:

1. **Insufficiently accurate transient representation**
   - The transient object should probably be completely revised. When comparing the transient to a sinusoidal track in a spectrogram it seems logical that a transient should be coded in the time-domain (vertical line) rather than in the frequency domain (horizontal line), as is the case now.

2. **Quality loss because of continuation of phase**
   - The linking of sinusoidal components can probably be improved even more by implementation of frequency dependent linking criteria. The instantaneous phase of a low-frequency sinusoid e.g. will rotate much slower than that of a high-frequency sinusoid. Over time the estimate of this phase will thus be more precise. Linking can thus probably be stricter than in the case of high frequencies.
   - Currently, the frequency and the phase together describe the instantaneous phase function of a sinusoid. It is questionable whether such a representation is optimal for the purpose of audio coding. Other representations could be investigated.
   - The implementation of harmonic grids as an object could further increase both quality and coding efficiency. First of all, harmonic grids can be coded efficiently as a fundamental (track) plus a set of harmonics (tracks). Such a description could also prove to be superior in quality to the current description consisting of individual sinusoids.
3. Quality loss caused by quantisation of sinusoidal parameters

- The rules for the quantisation of sinusoidal parameters should be improved. Non-tracks e.g. do probably not need to be coded as precisely as a long track. Harmonic grids however need to be encoded extra precisely because relations between harmonics must be preserved.

Apart from the recommendations mentioned above some more general recommendations are given that might be considered to further improve the codec:

**Quality improvement:**

- To further improve the quality of the encoder, a better distinction between noise and sinusoids should be made. Often sinusoids get coded as noise and noise gets coded as sinusoids. This is however a fundamental problem and will probably not be solved very easily.
- In this report no mention has been made about bit-rate control. In order to use SSC in practical applications bit-rate control is necessary. This is a further constraint to the bit-rate/quality ratio. It should be investigated how bit-rate control can be incorporated within the codec.

**Complexity reduction:**

- An efficient extraction method for sinusoids is absolutely necessary to achieve real-time encoding. The non-iterative extraction of sinusoids as described in Chapter 6 should be further investigated.
- The linear regression process also demands a lot of calculations per sinusoid. Basically for every sinusoid, every time-sample gets multiplied six times, once for every pattern function, with a complex value. These multiplications probably do not need to be performed on every sample. It could be investigated whether sub-sampling could be used to decrease the number of multiplications.
APPENDIX A  POWER OF SECOND ORDER POLYNOMIALS

The power of a single sinusoid is defined as the sum of the squared signals:

\[ P_p = \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} (s_p[n])^2, \]  

where a sinusoids is described as (see Chapter 4):

\[ s_p[n] = ae^{j\omega n} + ae^{-j\omega n} + bne^{j\omega n} + bne^{-j\omega n} + cn^2e^{j\omega n} + cn^2e^{-j\omega n}, \]

\[ = 2|a|\cos(\omega n + \varphi_a) + 2|b|n \cos(\omega n + \varphi_b) + 2|c|n^2 \cos(\omega n + \varphi_c), \]

where:

\[ \Omega_a = \omega n + \varphi_a, \]
\[ \Omega_b = \omega n + \varphi_b, \]
\[ \Omega_c = \omega n + \varphi_c. \]

Using Equation A.2 the squared signal can be written as:

\[ (s_p[n])^2 = \frac{1}{2} A^2 + \frac{1}{2} A^2 \cos(2\Omega_a) + \frac{1}{2} B^2 n^2 + \frac{1}{2} B^2 n^2 \cos(2\Omega_b), \]

\[ + \frac{1}{2} C^2 n^4 + \frac{1}{2} C^2 n^4 \cos(2\Omega_c) + ABn \cos(\Omega_a + \Omega_b) + ABn \cos(\Omega_a - \Omega_b), \]

\[ + ACn^2 \cos(\Omega_a + \Omega_c) + ACn^2 \cos(\Omega_a - \Omega_c), \]

\[ + BCn^3 \cos(\Omega_b + \Omega_c) + BCn^3 \cos(\Omega_b - \Omega_c). \]

The final solution, the power of a single sinusoid, is given as the sum of the following terms (Equation A.5 - A.16):

\[ \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{1}{2} A^2 = \frac{1}{2} A^2 \]

\[ \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{1}{2} A^2 \cos(2\theta n + 2\varphi_a) = \frac{1}{2} A^2 \left(\frac{\sin(\theta N + 2\varphi_a) + \sin(\theta N - 2\varphi_a)}{2N \sin \theta}\right) \]

\[ \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{1}{2} B^2 n^2 = \frac{1}{24} B^2 (N-1)(N+1) \]
\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} \frac{1}{2} B^2 n^2 \cos(2\theta n + 2\varphi_b) \\
= \frac{1}{2} B^2 \cos(2\varphi_b) \left( \frac{2N \sin \theta \cos \theta \cos \theta N - \sin \theta N \left(2 \cos^2 \theta + \left(1 - N^2\right) \sin^2 \theta \right)}{4N \sin^3 \theta} \right)
\]
A.8

\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} \frac{1}{2} C^2 n^4 = \frac{1}{2} C^2 \left( N - 1 \right) \left( \frac{1}{80} N^3 + \frac{1}{80} N^2 - \frac{7}{240} N - \frac{7}{240} \right)
\]
A.9

\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} \frac{1}{2} C^2 n^4 \cos(2\theta n + 2\varphi_c) \\
= \frac{1}{2} C^2 \frac{\cos(2\varphi_c)}{16N \sin^5 \theta} \sin(N \theta) \left(24 \cos^4 \theta + 4 \left(7 - 3N^2\right) \sin^2 \theta \cos^2 \theta + \left(N^4 - 6N^2 + 5\right) \sin^4 \theta \right)
\]
A.10

\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} ABn \cos(2\theta n + \varphi_a + \varphi_b) = AB \frac{\sin(\varphi_a + \varphi_b) \left(N \sin \theta \cos \theta N - \cos \theta \sin \theta N\right)}{2N \sin^2 \theta}
\]
A.11

\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} ABn \cos(\varphi_a - \varphi_b) = 0
\]
A.12

\[
\frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} ACn^2 \cos(2\theta n + \varphi_a + \varphi_c) =
\]
A.13

\[
AC \cos(\varphi_a + \varphi_c) \left( \frac{2N \sin \theta \cos \theta \cos \theta N - \sin \theta N \left(2 \cos^2 \theta + \left(1 - N^2\right) \sin^2 \theta \right)}{4N \sin^3 \theta} \right)
\]
\[ \frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} ACn^2 \cos(\varphi_a - \varphi_c) = \frac{AC}{12} \cos(\varphi_a - \varphi_c)(N-1)(N+1) \]  \hspace{1cm} A.14

\[ \frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} BCn^3 \cos(2\theta n + \varphi_b + \varphi_c) = \\
BC \sin(\varphi_b + \varphi_c) \left( \frac{\sin(\theta N)(6 \cos^3 \theta + (5 - 3N^2) \cos \theta \sin^2 \theta)}{8N \sin^4 \theta} \right) - \\
BC \sin(\varphi_b + \varphi_c) \left( \frac{\cos(\theta N)(6N \cos^2 \theta \sin \theta + N(3 - N^2) \sin^3 \theta)}{8N \sin^4 \theta} \right) \]  \hspace{1cm} A.15

\[ \frac{1}{N} \sum_{n=\frac{N-1}{2}}^{N-1} BCn^3 \cos(\varphi_b - \varphi_c) = 0 \]  \hspace{1cm} A.16
APPENDIX B  DESCRIPTION OF SOFTWARE

During the graduation project all software has been developed using Matlab. To encode the set of critical excerpts into quantised parameters and accompanying output waveforms a set of modules has to be run in the correct order. The table below (Table B.1) describes the order in which the software is to be used. It also describes the function of each module and a description of all the output files.

Table B.1: Software and descriptions

<table>
<thead>
<tr>
<th>Function-name (.m)</th>
<th>Function:</th>
<th>Parameter-files (coefsX*.mat)</th>
<th>Output-files (sicasX*.wav):</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod1</td>
<td>Extract parameters</td>
<td>mod1</td>
<td>t1: reconstructed transients</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s1: reconstructed sinusoids</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n1: reconstructed noise</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y1: t1+s1+n1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>r1: residual from sinusoids</td>
</tr>
<tr>
<td>mod2</td>
<td>Quantisation of transient and noise parameters</td>
<td>mod2</td>
<td>t8: quantised transients</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n8: quantised noise</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y1q: s1+t8+n8</td>
</tr>
<tr>
<td>mod3</td>
<td>Psycho-acoustic model</td>
<td>mod3</td>
<td>s2: psycho’d sinusoids</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y2: s2+t8+n8</td>
</tr>
<tr>
<td>mod5</td>
<td>Calculate tracking parameters</td>
<td>mod5</td>
<td>s6bcp: continued phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y6bcp: s6bcp+t8+n8</td>
</tr>
<tr>
<td>mod6</td>
<td>Linking of sinusoids</td>
<td>mod6</td>
<td>s7cp: short tracks removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y7cp: s7cp+t8+n8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s8cp: s7cp quantised</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y8cp: s8cp+t8+n8</td>
</tr>
</tbody>
</table>

All the functions described above are called without parameters as, for example:

```matlab
> mod1.m
```

The function that is called will then perform the function of table B.1, for this example the extraction of the parameters for the transients, sinusoids and noise, for all thirteen critical excerpts.

Furthermore the source code is path-dependent. It is therefore important that the directory structure of Figure B.1 is preserved.
Figure B.1: Directory structure for the Matlab code

Apart from the internal functions in each module the following (sub-)functions are used:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bark.m</td>
<td>converts frequency in Hertz to frequency in bark</td>
</tr>
<tr>
<td>Code.m</td>
<td>extracts the parameters for a given input signal</td>
</tr>
<tr>
<td>Codearma.m</td>
<td>extracts the noise parameters (ARMA coefficients)</td>
</tr>
<tr>
<td>Codeattack.m</td>
<td>extracts the transient parameters</td>
</tr>
<tr>
<td>Cos_gen.dll</td>
<td>function for fast generation of cosine function</td>
</tr>
<tr>
<td>Csinplus3.m</td>
<td>multi-scale extraction of sinusoidal parameters</td>
</tr>
<tr>
<td>Dbp.m</td>
<td>returns decibel value of a power</td>
</tr>
<tr>
<td>Dbv.m</td>
<td>returns decibel value of an amplitude</td>
</tr>
<tr>
<td>Decarma.m</td>
<td>decode noise signal from ARMA coefficients</td>
</tr>
<tr>
<td>Decparts.m</td>
<td>decode signal</td>
</tr>
<tr>
<td>Decpartsso.m</td>
<td>decode sinusoids only</td>
</tr>
<tr>
<td>Decpartssoplus.m</td>
<td>decode sinusoids only, include higher order information</td>
</tr>
<tr>
<td>Delnt2.m</td>
<td>removal of non-tracks</td>
</tr>
<tr>
<td>Erb.m</td>
<td>convert frequency in hertz to frequency in ERB</td>
</tr>
<tr>
<td>Fastcos.m</td>
<td>fast generation of cosine function</td>
</tr>
<tr>
<td>Inverb.m</td>
<td>convert frequency in ERB to frequency in hertz</td>
</tr>
<tr>
<td>Lar2rc.m</td>
<td>convert Log Area Ratio coefficients to reflection coefficients</td>
</tr>
<tr>
<td>Lin.m</td>
<td>conversion from power in decibel to linear power</td>
</tr>
<tr>
<td>Psycho.m</td>
<td>frequency masking model</td>
</tr>
<tr>
<td>Quantarma.m</td>
<td>quantise noise (ARMA) coefficients</td>
</tr>
<tr>
<td>Quantatt.m</td>
<td>quantise attack parameters</td>
</tr>
<tr>
<td>Rc2lar.m</td>
<td>convert reflection coefficients to log area ratio parameters</td>
</tr>
<tr>
<td>Ssc_f2erl.m</td>
<td>convert frequency to representation level</td>
</tr>
<tr>
<td>Trackpav3.m</td>
<td>tracking mechanism</td>
</tr>
<tr>
<td>Tracks2parold.m</td>
<td>convert tracks to matrices</td>
</tr>
<tr>
<td>Tracksanalod.m</td>
<td>convert matrices to tracks</td>
</tr>
<tr>
<td>Ximax.m</td>
<td>sub-function of transient module</td>
</tr>
</tbody>
</table>
APPENDIX C REFERENCES

[1] Zwicker, E.
SUBDIVISION OF THE AUDIBLE FREQUENCY RANGE INTO CRITICAL BANDS
(FREQUENZGRUPPEN)

AN INTRODUCTION TO THE PSYCHOLOGY OF HEARING

A PSYCHO-ACOUSTIC MASKING MODEL FOR A SUB-BAND CODER - STATUS QUO

[4] Pan, D.
A TUTORIAL ON MPEG/AUDIO COMPRESSION

THE ISO/MPEG-AUDIO CODEC: A GENERIC STANDARD FOR CODING OF HIGH
QUALITY DIGITAL AUDIO
92nd AES-convention, 1992, preprint 3336, Vienna, Austria.

SPEECH ANALYSIS/SYNTHESIS BASED ON SINUSOIDAL REPRESENTATION.

FAST ARMA MODELLING OF POWER SPECTRAL DENSITY FUNCTIONS

[8] den Brinker, A.C. and A.W.J. Oomen
SINUSOIDAL CODING OF AUDIO AND SPEECH (SiCAS) - A LOW BITRATE
UNIVERSAL AUDIO AND SPEECH CODER

A FEASIBILITY STUDY ON SINUSOIDAL CODING FOR AUDIO AND SPEECH - A LOW
BITRATE UNIVERSAL AUDIO AND SPEECH CODER

[10] den Brinker, A.C. and A.W.J. Oomen
DESCRIPTION OF PARAMETERIZATION FOR AUDIO AND SPEECH SIGNALS
BASED ON A SINUSOIDAL APPROACH (SiCAS)
TECHNICAL DESCRIPTION OF THE MPEG-4 AUDIO-CODING PROPOSAL FROM UNIVERSITY OF HANNOVER AND DEUTSCHE BUNDESPOST TELEKOM AG (REVISED)

HIGH-PRECISION FOURIER ANALYSIS OF SOUNDS USING SIGNAL DERIVATIVES

SUBJECTIVE ASSESSMENT OF SOUND QUALITY

[14] Schuijers, E.G.P.
METHODICAL LITERATURE SEARCH PROJECT – QUALITY SCALABILITY OF A PARAMETRIC AUDIO CODER