University of Auckland

COMPSYS401A Project

Audio Watermarking

Final Report

Name:	Dackson LAM
ID:	2536685
Group:	61
Project Partner:	Andrew Sung
Supervisor:	Dr. Waleed H.Abdulla
Second Marker:	Prof. Zoran Salcic
Department:	Electronic and Electrical Engineering
Date:	15th September, 2003

Table of Contents

List of Illustrations

Dackson LAM
2536685

Abstract

An audio watermarking technique based on complementary filter band division is developed by Andrew Sung and me, Dackson Lam, as the final year project. This report presents the workings and results of the project in details. It has shown that audio watermarking is chosen for investigation due to its recentness and importance to the audio recording industry. Then, the report provides the background knowledge of audio watermarking. Then, it examines and compares four audio watermarking methods which are based on DC level shifting, quadrature mirror filter (QMF) bank band division, frequency masking and spread spectrum. The report justifies that the QMF bank scheme is chosen for implementation at first because it is a relatively robust blind watermarking scheme. This scheme is developed to be based on complementary filter bank later because perfect reconstruction of a signal in a QMF bank is not achieved in practice. The results from the evaluation of the developed scheme are presented. It shows that the modified scheme is a successful watermark scheme. Next, the graphical user interface (GUI) created for accessing the watermarking scheme is introduced. At last, the report proposes the possible future improvements such as robustness to different attacks to enhance this watermark scheme.

Dackson LAM
2536685

Declaration of originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Signed: ___________________________

1.0 Introduction

In recent years, the distribution of digital media has grown rapidly due to the wide spread of the Internet. The copyright of these digital media has become a lot more difficult to manage. As a result, a technique called digital watermarking is introduced to protect the ownership of these contents. Digital watermarking can be realized by many different methods. In common to all of those methods, digital watermarks are embedded into the media contents as secret copyright identification code. A watermark hidden in a file cannot be detected by general user because the watermark will not deteriorate the quality of the file (Furukawa & Konishi & Saito, 2002). The watermark can be extracted by special techniques if they are needed. Watermarks can be embedded into image, audio and video files (Jung & Lee, 2001). Audio watermarking is the case when watermarks are introduced to audio files. In comparison to watermarking of other media, audio watermarking is a relatively recent subject (SysCoP, 2000). It performs a key role in the music recording industries. Therefore, this research-based project is set up to investigate audio watermarking. Moreover, the project aims to implement one watermark scheme from research and to evaluate its performance.

In this project, Andrew Sung and I have examined the background of audio watermarking. Then, we selected four watermark schemes to investigate in detail. By using one of the schemes that employs QMF bank band division as a foundation, we have successfully developed a watermarking scheme that is based on complementary filter band division. After that, we evaluate the performance of our scheme. We have also created a GUI for easy access of our implemented scheme. We have divided the workload at both the research stage and the implementation stage evenly. The combined efforts and the results of our project are presented in this report in an orderly fashion.

2.0 Audio watermarking

Digital watermarking is techniques of embedding information into a signal. The host signal that carries the watermark is also called a cover signal. When the cover signal is an audio signal, the embedding technique is called audio watermarking. The purposes, types and requirements of audio watermarking are researched and presented in this section.

2.1 Purposes

There are various purposes for audio watermarking. The original intention of watermarking is for copyright protection. Therefore, the most obvious purposes are the needs for proof of ownership and the enforcement of usage policy. In addition, watermarking can also be used for fingerprinting and additional features to a media (Craver & Liu & Wu, 2001).

2.1.1 Proof of ownership

A watermark can represent a code of information to identify the owner of a digital signal. This application is similar to the function of international standard book number (ISBN) for book identification. The watermark must be correctly presented to proof an ownership in a court of law (Craver et al., 2001).

2.1.2 Enforcement of usage policy

Watermark can be used to provide copyright information to consumer devices. The usage of audio information will be limited or stopped by the devices if certain requirement is not fulfilled by the user. However, this function of watermark has posted a difficulty in actual application. This is because in order for a consumer device to recognize a watermark, the watermark or the secret key for watermark generation has to be kept by the device. Attackers can use reverse engineering to obtain the watermark or disable the watermark verifying function in a device (Craver et al., 2001).

2.1.3 Fingerprinting

The usage of an audio file can be recorded by a fingerprinting system. When a file is accessed by a user, a watermark, or called fingerprint in this case, is embedded into the file. The usage history can be traced by extracting all the watermarks that were embedded into the file (Craver et al., 2001).

2.1.4 Additional features

A watermark can also provide additional information to a file. For instance, the lyrics can be embedded into a song and extracted when it is played (Craver et al., 2001). Furthermore, the watermark can be a special label for convenient search function in databases (Petitcolas, 2000).

2.2 Requirements of watermark

For a scheme to fulfill the purposes of watermark, a number of requirements have to be satisfied. The most significant requirements are perceptibility, reliability, capacity and speed performance (Petitcolas, 2000).

2.2.1 Perceptibility

The most important requirement is that the quality of the original signal has to be retained after the introduction of watermark. A watermark cannot be detected by listeners (Petitcolas, 2000).

2.2.2 Reliability

Reliability involves the robustness and detection rate of the watermark. A watermark has to be robust against intentional and unintentional attacks (Petitcolas, 2000). The detection rate of watermark should be perfect whether the watermarked signal has been attack or not. Otherwise, the watermark extracted is not useful for proof of ownership. Secure digital music initiative (SDMI), an online forum for digital music copyright protection, has summarized a list of possible attacks to evaluate the robustness of watermarking schemes. These attacks include digital-to-analog, analog-to-digital conversions, noise addition, band-pass filtering, time-scale modification, addition echo and sample rate conversion (SDMI, 2000). If the quality of the watermarked signal after the attacks is not significantly distorted, the watermark should not be removed by these attacks.

2.2.3 Capacity

The amount of information that can be embedded into a signal is also an important issue. A user has to be able to change the amount embedded to suit different applications. An example can be seen in real-time application. If a watermark is spread across an audio signal, the complete signal has to be presented first. This is not possible in streaming over the Internet. Note that there is a trade-off situation in information capacity of a watermark and its quality. A signal will be degraded more if more information is embedded (Petitcolas, 2000).

2.2.4 Speed

Watermarking may be used in real-time applications, such as audio streaming mentioned before. The watermark embedding and extracting processes have to be fast enough to suit these applications (Petitcolas, 2000).

2.3 Classification by extraction requirements

Watermarking schemes can be classified by the information required in the watermark extraction process. In private watermarking, the original data and watermark are needed to verify the presence of a watermark. A secret key used in embedding may also be needed. In semi-private watermarking, the original secret key and watermark are needed in order to identify a watermark. The original signal is not required. Public watermarking, also called blind watermarking, only requires the secret key used in embedding for a watermark to be extracted (Jung & Lee, 2001).

3.0 Survey of techniques

Research for a suitable watermark technique for implementation is an essential part of the project. There are various watermark methods available. The decision on selecting a particular method has to be made carefully. The methods should be judged based on the requirements of audio watermarking.

3.1 Description of techniques

Four watermarking schemes were chosen to be investigated in detail. These schemes embed watermarks based on the idea of DC level shifting, QMF bank band division, frequency masking and spread spectrum.

3.1.1 DC Level Shifting

In the DC level shifting method, the watermark is embedded by shifting the DC level of the audio signal. Initially, an input signal is divided into frames of fixed length. Then, the DC level of each frame, calculated as the mean of a frame, is subtracted from the values in the frame. After that, a binary watermark sequence generated randomly is introduced to the shifted signal. This is achieved by introducing a DC offset level to each frame in the signal according to the watermark bits. If the watermark bit corresponding to a frame is 0, the signal is shifted downward. If the watermark bit is 1, the signal is shifted upward. The magnitude of the shifting is governed by a DC bias multiplier constant and the power of the frame.

$$level_0 = -DCBiasMultiplier \times FramePower \quad (1)$$
$$level_1 = +DCBiasMultiplier \times FramePower \quad (2)$$

where
$level_0$ is the DC level introduced when a watermark bit is 0 and
$level_1$ is the DC level introduced when a watermark bit is 1.

To extract a watermark, a watermarked signal is first divided into frames. Then, the mean of each frame is calculated. If the mean of a frame is positive, the corresponding watermark bit is 1. Else, the watermark bit is 0. The original signal and watermark is not needed in the extraction process. Therefore, DC level shifting is a blind watermarking scheme (Uludag, 2001).

3.1.2 Band Division based on QMF bank

For this scheme of watermark embedding using quadrature mirror filter (QMF) bank band division, the low frequency component of an input signal is extracted for watermark embedding. The QMF bank used in this scheme has a property of perfect reconstruction. A signal can be decomposed by the filter bank and recombined with no loss of data. A QMF bank contains levels of low pass and high pass filters. The filters used for dividing signals are called analysis filters. The filters used for combining signals are called synthesis filters. In each level, a signal is divided into a low frequency part and a high frequency part with equal bandwidths. The low frequency part in one level is passed to the next level for further decomposition. Therefore, the lowest frequency component of a signal is extracted at the last level of the QMF bank. This section is then used for watermark embedding. There are two reasons for embedding the watermark at low frequency. The first reason is that the distortion produced by the watermark is less perceptible at low frequency. The second reason is that audio signals often experienced compression such as MPEG1 Audio Layer 3 (mp3) encoding. The high frequency information is lost when a signal is compressed. Therefore, encoding at low frequency will decrease perceptibility and increase robustness of watermarks.

After the low frequency band of a signal is found, further processing is performed before the actual watermark embedding. First of all, the spectrum of the low frequency band is spread using a technique called direct sequence spread spectrum. The operation will increase the security of the watermarking scheme because a PN sequence is used to randomize the data. After that, the signal is transformed to the frequency domain using modified discrete cosine transform (MDCT). The MDCT will produce a set of frequency coefficients for the watermark to be embedded. Then, the watermark is embedded by rounding the coefficients to odd or even numbers depending on the corresponding watermark bit. After the embedding, the inverse MDCT (IMDCT) is performed to transform the coefficients back to time domain. The inverse spread spectrum technique is then used to reserve the effect of spread spectrum. Finally, the modified low frequency band is recombined with the high frequency bands for a watermarked signal to be constructed. A schematic of this watermarking method is presented in figure 1.

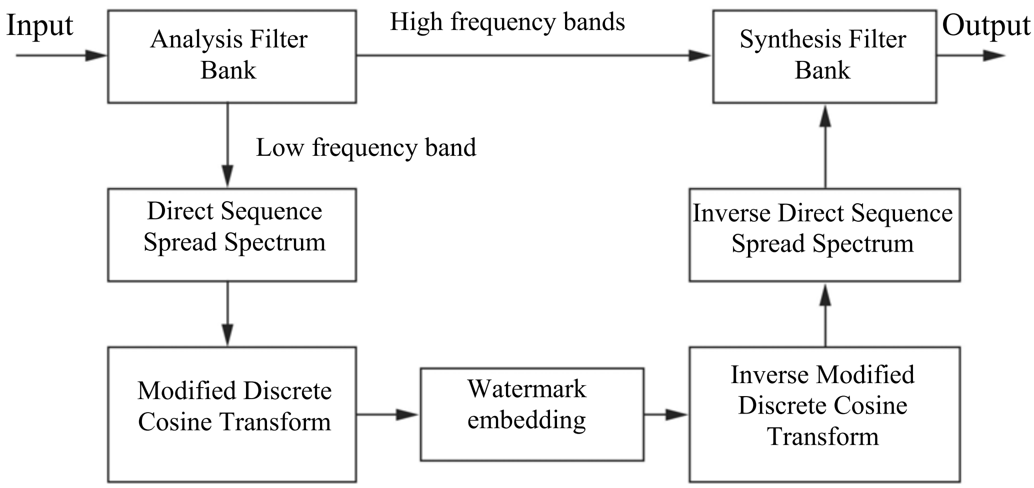

Figure 1. Watermark embedding scheme for QMF bank

The processes involve in the extraction of watermark are similar to the embedding processes. The low frequency band of a watermarked signal is first found by the same QMF bank used in embedding. Then, spread spectrum and MDCT is used. The set of frequency coefficients found MDCT is rounded. The watermark can be extracted by inspecting if the rounded coefficients are odd or even (Furukawa et al., 2002).

3.1.3 Frequency Masking

The frequency masking technique exploits an effect called masking, where a faint sound is rendered inaudible by a louder sound that is played at the same time. In this case, the louder sound, also called the masker, is the host signal. The faint sound is the watermark. The masking operation is performed in the frequency domain. Thus, it is termed frequency masking. By taking the notion of frequency masking, a watermark is shaped according to the characteristics of the host signal. However, this is only the first stage of the watermarking scheme. As the QMF method mentioned, this watermarking scheme also focused on concentrating the watermark embedding in the low frequency part of the signal. The reason for such focus is also because low bit rate encoding such as mp3 will remove watermark embedded at high frequency. Hence, a second stage is set up to reinforce the robustness to compressions.

There are four key steps in the first stage of the watermark generation. First, a masking threshold of the input audio signal is calculated by MPEG audio psychoacoustic model. The main feature of the model is to weight the audio signal in the time domain using a Hanning window. Secondly, a pseudo noise sequence (PN sequence) is generated to be the basis of a watermark. This sequence is filtered to fall below the masking threshold to ensure inaudibility. Thirdly, the filtered PN sequence is amplified for a maximal extraction performance. After that, the amplified signal is scaled in the time domain to ensure its inaudibility. At last, the signal is quantized to form a first stage watermark signal. Figure 2 summarizes the basic operations for the first stage of this watermarking scheme.

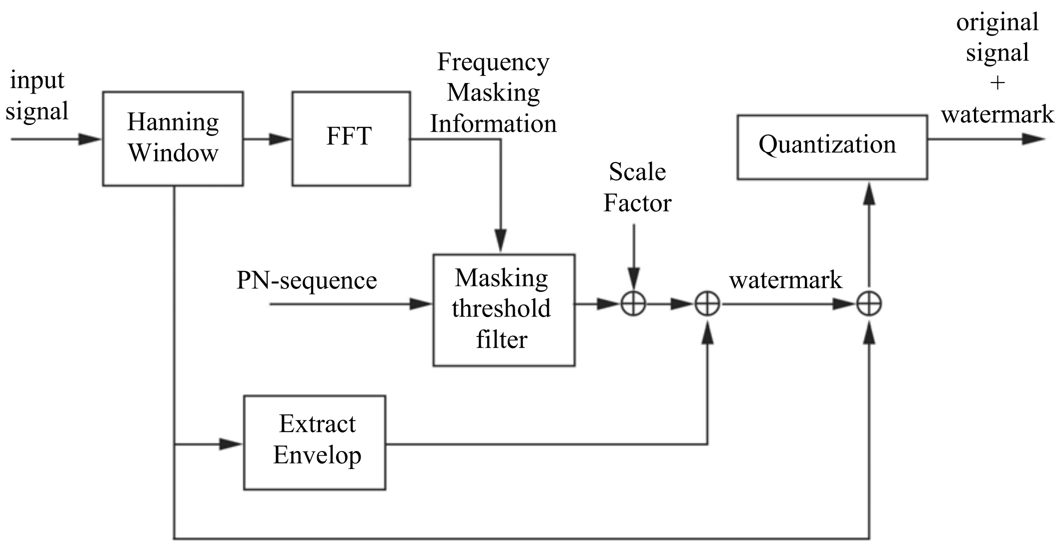

Figure 2. First stage watermark generation in frequency masking scheme

The watermark from the first stage is further process in the second stage. The watermark is first encoded at a lower bit rate than the sampling rate of the original cover signal. This bit rate is denoted as *br*. Then, a low frequency watermarking signal is calculated by subtracting the original signal encoded at the same low bit rate from the low bit rate watermark.

$$w_{br} = w'_{br} - x_{br} \quad (3)$$

where
x_{br} is the original signal at low bit rate,
w'_{br} is the watermark signal from the first stage at low bit rate, and
w_{br} is the watermark signal at the low bit rate.

After that, a coding error is calculated. The coding error is the difference of the original signal at original sampling rate and the original signal encoded at low bit rate.

$$\text{coding error} = x - x_{br} \tag{4}$$

where
x is the original signal encoded at original sample rate.

The next step is to generate a watermark for the coding error signal. It is calculated by subtracting the coding error from the watermark signal.

$$w_{err} = w'_{br} - \text{coding error} \tag{5}$$

where
w_{err} is the watermarked signal for the coding error.

Then, a final watermark is produced by the addition of the watermark at low bit rate and the watermark for the coding error.

$$w_f = w_{br} + w_{err} \tag{6}$$

where
w_f is the final watermark.

At last, the final watermarked signal is generated by adding the final watermark with the cover signal.

$$x_w = x + w_f \tag{7}$$

where
x_w is the final watermarked signal.

The second stage of the watermarking technique is illustrated in figure 3.

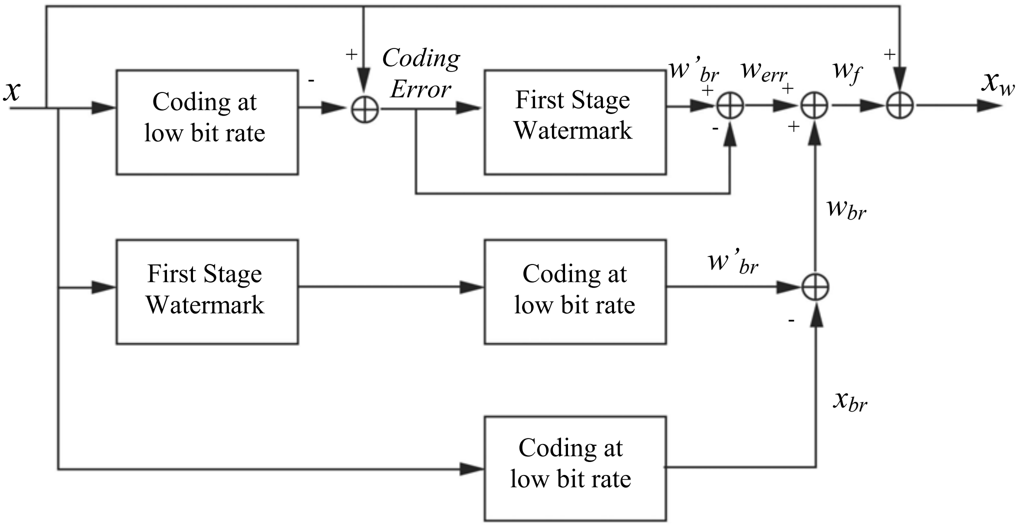

Figure 3. Second stage in the frequency masking watermarking scheme

In contrast to the DC level shifting scheme and the QMF scheme, the watermark is a semi-private watermarking scheme. A user has to possess the original watermark and the PN-sequence to detect if the watermark is presented in a given signal. To detect a watermark, the bit rate of the given signal has to be known first. A user needs to encode the original signal at this bit rate and subtract the encoded data from the given signal. The difference is then correlated with a modified watermark derived from the original PN sequence. Lastly, the result of the correlation is compared with a threshold to determine the presence of the watermark (Boney & Hamdy & Tewfik, 1996).

3.1.4 Spread Spectrum

The spread spectrum method researched is a variation of the basic spread spectrum watermarking scheme. It improves the basic scheme by enhancing the robustness against de-synchronization attacks. In a basic spread spectrum scheme, the watermark embedded in a signal cannot be detected correctly if the signal is scaled along time or frequency axis. This form of attack is known as de-synchronization. The researched spread spectrum method overcomes this weakness by introducing an embedding technique called redundant encoding.

A basic spread spectrum watermarking scheme involves a transformation to be performed to an audio signal at first. The transformation converts a signal from time domain to another such as frequency domain. The transformation should be reversible, i.e. a signal can be converted back to time domain. The watermark is a random binary sequence generated by a pseudo-random bit generator. It is spread using spread spectrum to reduce its distortion to the host signal. Then, the spread watermark is shaped according the human auditory system (HAS) characteristics for inaudibility. At the end, the watermark and the input audio signal is combined to form the watermarked signal.

In this spread spectrum method, a number of frames produced by the modulated complex lapped transform (MCLT) of the signal are used for watermark embedding. MCLT is a type of filter bank useful for noise suppression and echo cancellation (Malvar, 1999). The spread watermark will be embedded to the MCLT coefficients by redundant encoding. In this coding method, several blocks of MCLT coefficients are used to carry a watermark sequence. Each watermark bit is spread over a number of contiguous frequency coefficients in a MCLT block. Then, the MCLT block is replicated and repeated several times along the time axis consecutively. The redundant encoding scheme is illustrated in figure 4. After the encoding procedure, the coefficients values are merged with the phase of the cover signal. The result is combined with other filtered frames in the inverse MCLT to form the watermarked signal.

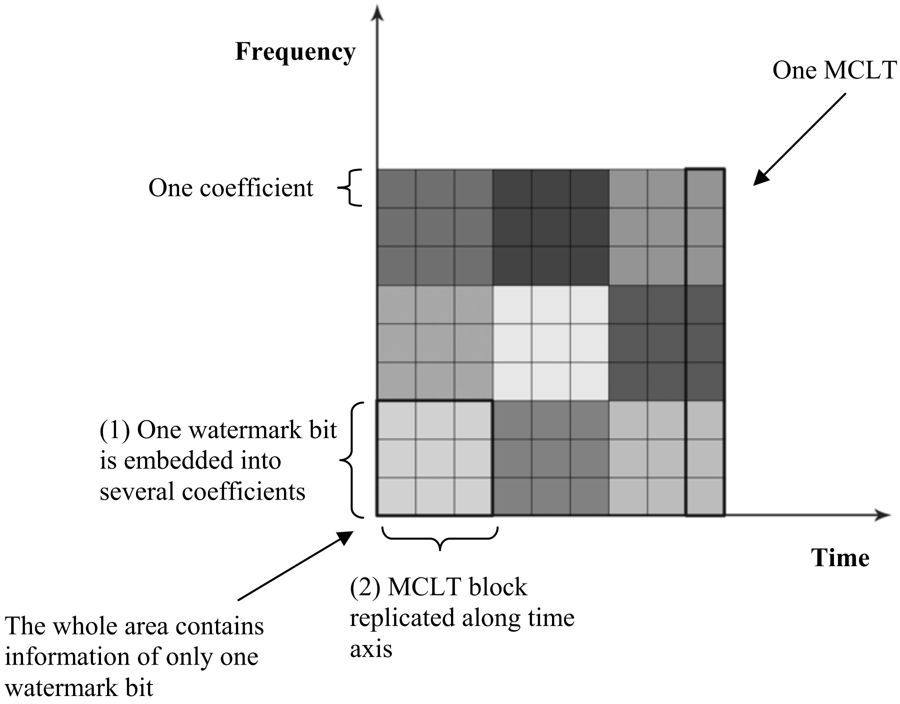

Figure 4. Redundant encoding in spread spectrum watermarking

This spread spectrum scheme is a private watermarking scheme. The presence of a watermark is tested by the correlation of a given signal with the original signal and watermark (Kirovski & Malvar, 2001).

3.2 The developed technique

After the algorithms of the four techniques are thoroughly examined, one technique has to be chosen for implementation. The decision is made by comparing the performance of the watermark schemes with respect to the watermark requirements.

The testing performed in DC level shifting method is limited. Experiments by the author of the method have shown that the audibility of the watermark is low and the detection performance of the watermark is high. However, no experiment is performed to test the robustness of the watermark (Uludag, 2001).

The technique based on QMF bank band division can provided high detection rate. Moreover, the signal-to-noise ratio is found to be 87dB. This proves that the watermark is inaudible. Although the aim of the watermarking scheme is to ensure robustness to compression, the watermark detection rate after mp3 compression is only 57% (Furukawa et al., 2002).

Extensive tests have been carried out to examine the performance of the frequency masking scheme. The detection rate is nearly perfect. The presence of watermark is imperceptible to listeners, even at the silent sections of signals. Furthermore, the watermark detection is resilient to attacks of additive noise, lossy coding/encoding, resampling and time-scaling. Multiple watermarks can also be detected accurately (Boney et al., 1996).

The spread spectrum method also has highly satisfactory results in the detection rate and perceptibility test. The attack by de-synchronization has been successfully overcome. In addition, this method is robust to editing of reverb, echo, denoising, filtering and combinations of these editing processes (Kirovski & Malvar, 2001).

From the analysis of performance, it can be seen that all of the schemes have high detection rate and low audibility of watermark. The frequency masking and the spread spectrum schemes have shown superior performance. However, both of these two schemes only provide watermark detection. They both require the original signal to be available. In some situations, the original signal may not be accessible. Therefore, a blind watermarking scheme is preferred and these two schemes are discarded. There is no proof of high security in the DC level shifting scheme. It is thought that the watermark can be easily removed by altering the DC levels of the audio contents. In comparison, the QMF method provides a higher security by having a PN sequence and the embedding is operated in the frequency domain. Moreover, the QMF method is introduced in 2002. It is more current than the other methods and therefore, more worth exploring. Hence, the QMF bank band division is chosen to be implemented in the project.

While the QMF bank band division scheme is being implemented, a number of results are found to be different to those stated by the creators of the scheme. Therefore, a modified scheme is developed. Although the scheme employs the idea of band division, spread spectrum, MDCT and the embedding method from the QMF bank scheme, several details of these stages are modified. The main difference is that instead of QMF bank, the developed scheme is based on the idea of complementary filter bank for band division.

4.0 Implementation

The implementation of the watermarking scheme based on QMF bank can be divided into four sections. These sections are filter banks, direct sequence spread spectrum, modified discrete cosine transform and watermark embedding. As the project progresses, it was found that a number of methods proposed by Furukawa, Konishi and Saito cannot be carried out to match the results described in their paper. Therefore, modifications are made to those parts in order to obtain the optimal results. As a result, a unique watermarking scheme is developed by using complementary filter bank as the basis.

All of the algorithms in this watermarking scheme are computed using MATLAB. This is because MATLAB is a standard software widely used in the audio industry and academic institutions. Moreover, it provides simple control in programming. An additional package of functions called WaveLab is also used for audio signal processing (Donoho & Duncan & Huo & Levi, 1999).

4.1 Filter bank

The objective of the first stage in the watermarking scheme is to extract a low frequency component of an input audio signal. Watermark will be embedded after this component is processed. There are two reasons for using low frequency component to carry watermark. The first reason is that human ears are more sensitive to high frequency than to low frequency. Introduction of watermark at a low frequency section can made the distortion less perceptible. The second reason is to increase the robustness of watermark from MPEG1 Audio Layer 3 (mp3) compressions. Mp3 is a widely used audio compression technique in recent years (Jung & Lee, 2001). This technique reduces the information size of an audio signal by discarding the high frequency components that are imperceptible to humans. Hence, if a watermark is embedded at high frequency portion of a signal, it can be removed by mp3 compression (Furukawa et al., 2002).

A low frequency component of a signal can be obtained by using QMF bank. However, the outcome found in actual implementation using QMF bank is not as ideal as in theory. As a result, complementary filter bank is used in the project.

4.1.1 QMF bank

Filter bank can be used to split signals into high and low frequency components and recombine them at a later stage. There is more than one level of filtering within a filter bank. In QMF bank, there are one high pass filter and one low pass filter in each level to divide the input. The bandwidths for the split high frequency and low frequency components are identical in the same level. This means that a signal is split in the middle in the frequency domain. In asymmetric QMF bank, the low frequency component extracted out, or called low frequency subband, is further decomposed at the next level. The high frequency subbands will not be processed until reconstruction. This notion of band division is illustrated in figure 5. QMF bank has a property of perfect reconstruction. This implies that a signal can be decomposed and then recombined without changes in data values. Thus, QMF bank is chosen to perform the splitting of signals (Furukawa et al., 2002).

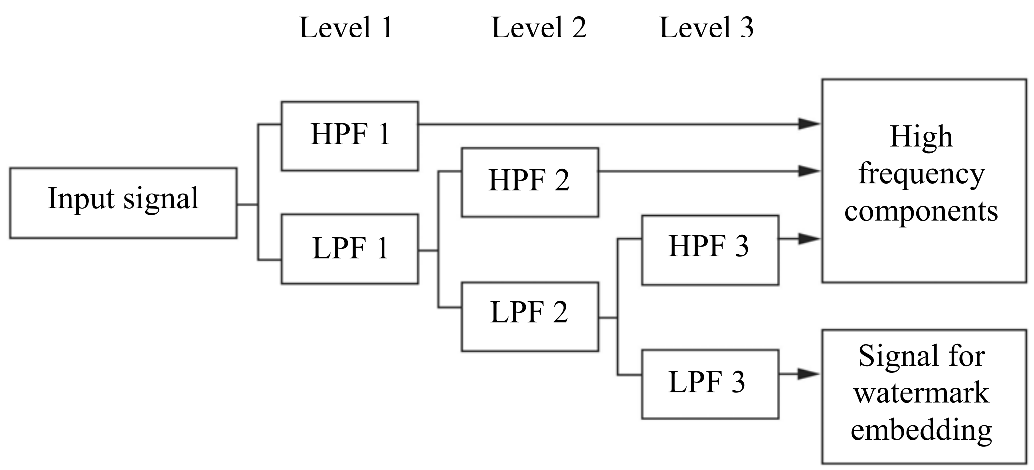

Figure 5. Structure of a 3-level QMF bank band division

The minimum sampling rate of a signal required to prevent alias distortion is called Nyquist frequency. The filters used in QMF bank, octave bank pass filters, have cut-off frequencies at the first quarter of the Nyquist frequency. For a Nyquist frequency of 2π, the cut-off frequency of a filter is at $\pi/2$. At each level, the low pass filter is a mirror image of the high pass filter at the cut-off frequency. Due to these properties, the filter bank is given the name quadrature mirror filter banks (Mitra, 1999, p.709).

The filters used for dividing an input signal are called analysis filters. The filters used for reconstructing are called synthesis filters. When the equation describing one filter is decided, the other filters can be derived. All of the filters are finite impulse response (FIR) filters. FIR filter is used because of its linear phase property. All signals that pass through the same FIR filter will have the same phase shift. This is an essential property since all of the subbands can only be recombined perfectly with the same phase shift. A related concept to bear in mind is that due to these phase shifts, the reconstructed signal after QMF bank will be a delayed version of its original signal.

After a signal is divided by analysis filters in one level, the two subbands have to be down-sampled. In QMF bank down-sampling, every second data of a subband in the time domain is discarded. Therefore, the size of a signal is halved after one down-sample process. The bandwidth of the signal is doubled with respect to the Nyquist frequency. Since the size of the signal is halved, the computation of the signal will be less. This is the reason such alteration of sampling rate is used in a QMF bank (Mitra, 1999, p.709). At the synthesis filters, an opposite action is performed. The combined data of two subbands has to be up-sampled. An up-sample process in QMF bank inserts a zero after every sample of a signal in the time domain. The bandwidth of a signal is halved after up-sampling (Mitra, 1999, p.660). The structure of up-sampling and down-sampling in one level of QMF bank is presented in figure 6.

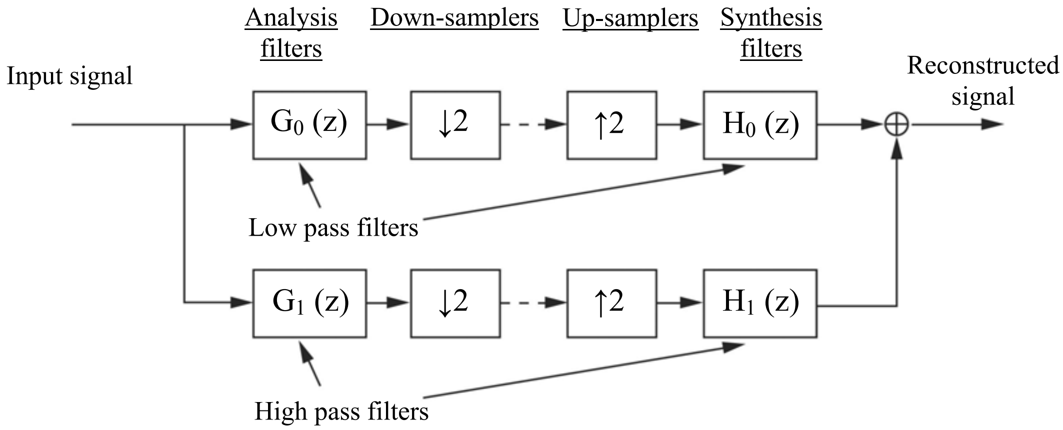

Figure 6. Up-sampling and down-sampling in one QMF bank level

After the theory of QMF bank is studied, the implementation in software begins. The parameters to design a QMF bank are the type of filters, their equations and the number of levels in the filter bank. First, the analysis low pass filter is decided to be generated as a base for the other three filters in the same level. The filter is created using remez filter function supplied by MATLAB. This function uses the desired frequency response of a filter given by a user to generate a set of coefficients to describe the filter. It is used in the project because MATLAB states that the error of the actual frequency response of the filter generated to the desired frequency response is minimal. The order of the filter is set to 120 for a good approximation of the desire filter. The responses of filters generated with higher orders do not show any significant improvement. Therefore, the order of 120 is kept for the ease of computation. The second step of implementation is to derive the equations for the analysis high pass filter and the synthesis low pass and high pass filters. For the filter bank to be alias-free, the equations of the filters in a level of QMF bank have to satisfy the following condition.

$$\frac{G_0(z)}{G_1(z)} = \frac{H_1(-z)}{H_0(-z)} \tag{8}$$

where

$G_0(z)$ represents the analysis low pass filter,
$G_1(z)$ represents the analysis high pass filter,
$H_0(z)$ represents the synthesis low pass filter, and
$H_1(z)$ represents the synthesis high pass filter.

One feasible solution to the constraint is presented here (Hang, 2000). The frequency responses of the four filters derived from this solution are shown in figure 7.

Analysis low pass filter: $G_0(z) = G(z)$ (9)

Analysis high pass filter: $G_1(z) = G(-z)$ (10)

Synthesis low pass filter: $H_0(z) = 2G(z)$ (11)

Synthesis high pass filter: $H_1(z) = -2G(-z)$ (12)

where

$G(z)$ is the filter generated by the MATLAB remez function.

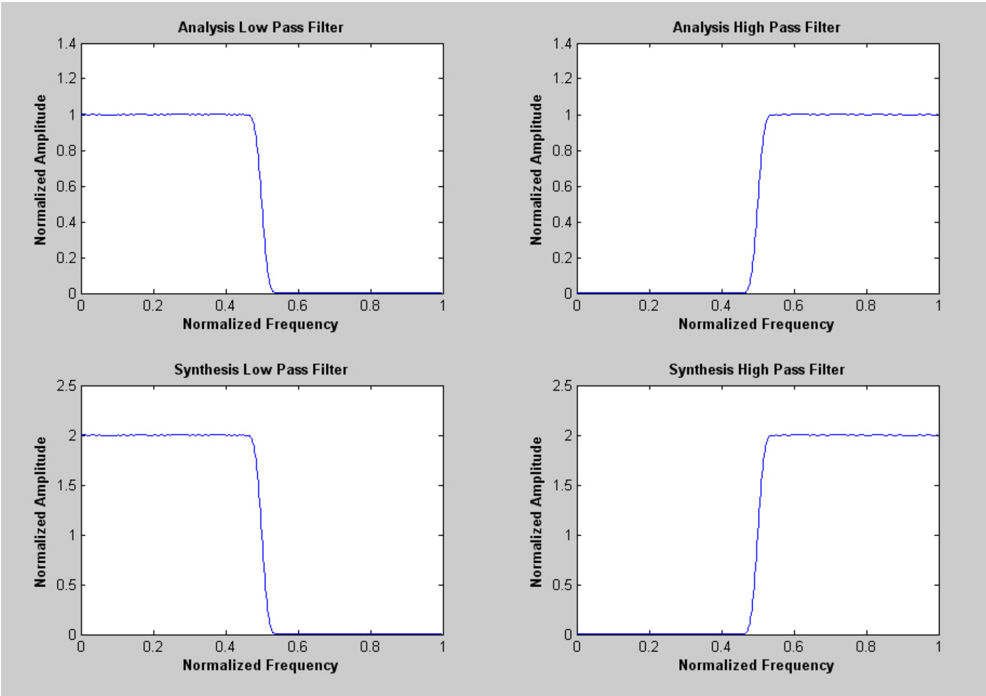

Figure 7. Frequency responses of the four filters in one level of QMF bank

The last parameter to decide for the QMF bank implementation is the number of levels of the filter bank. At first, the level is set to be 4 for a trial of QMF implementation. At a later stage, the number of levels is developed to 8 and 16. The distortions introduced by the embedded watermark are found to be reduced as the number of levels increases.

A QMF bank with the parameters mentioned is created in MATLAB. The performance of the filter bank is tested by comparing the reconstructed signal with its original input. Three audio waves with CD quality are used because this suits the real application. This means these waves are all sampled at 44100 Hz with 16-bit resolution. The types of the audio signal are speech, instrumental music and a song with vocal. The result of the experiment shows that perfect reconstruction of a signal is not achieved. One result of the experiment is shown in figure 8. The spectrograms showing the relationships of the time and frequency of the signals are provided in figure 9.

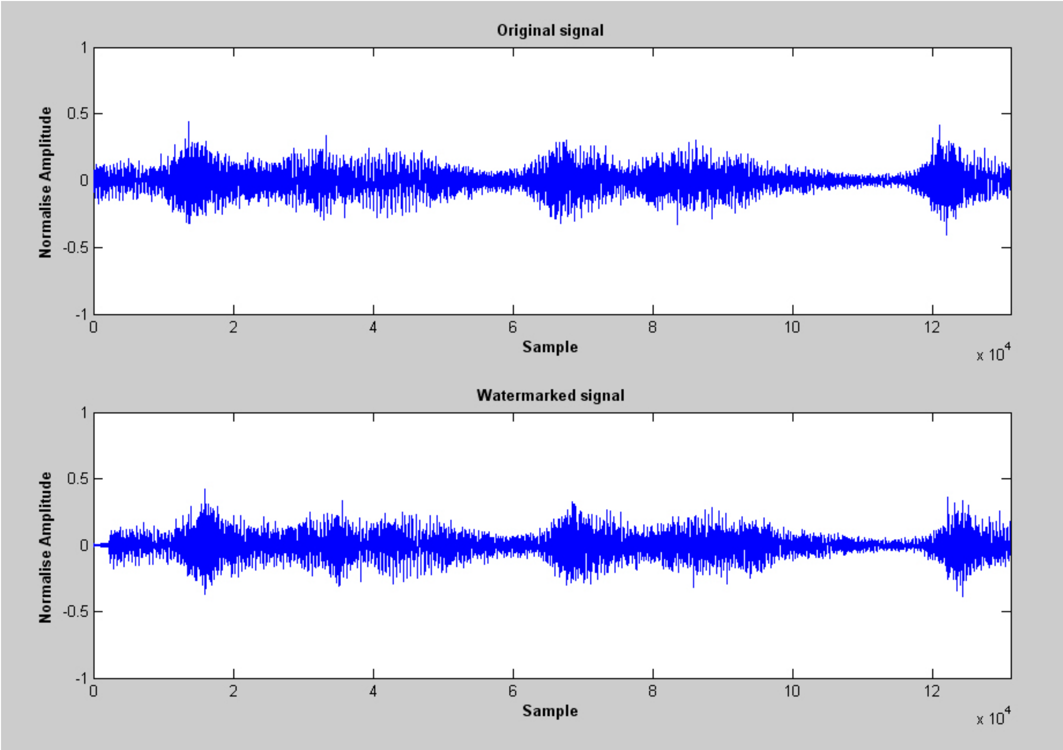

Figure 8. A signal before and after QMF

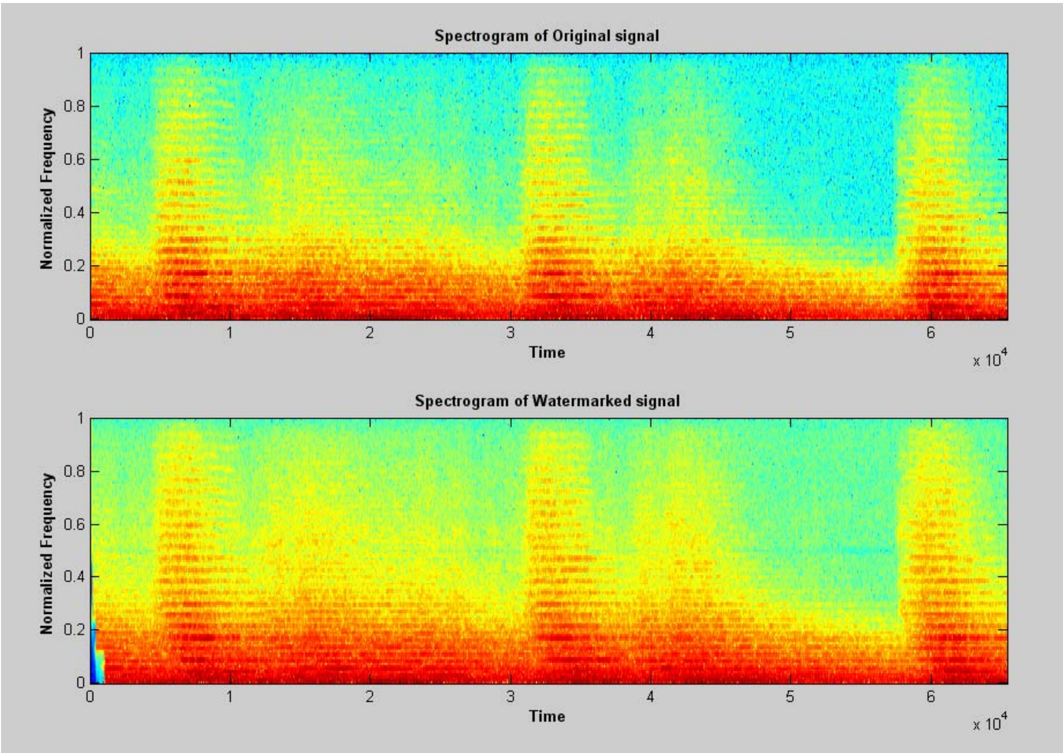

Figure 9. Spectrograms of a signal before and after QMF

Although the reconstructed signal possesses the characteristics of the original input, the sample values do not match perfectly. No difference has been detected by listening to the signals when there the QMF bank has 4 levels. However, when the number of levels is increased to 16, the effect of echo can be clearly identified from the speech signal. A signal and its delayed version can be heard simultaneously. The spectrograms of the input and output have revealed that a subband will experience more phase shift as it passes through more filters. Therefore, as the number of filter level is increased to 16, the delay of the lowest frequency subband becomes significant. The different degrees of delay can be identified at the lower left hand corner of the spectrograms shown in figure 9. It can be seen that the phase shift at lower frequency is more significant than those at higher frequency.

Several attempts have been made to address the problem of echo. The software code is carefully examined by dividing into several modules. It is found that both the down-sample and up-sample process and the divisions and combinations of subbands are correct. The problem should exist in the design of the filters. Thus, the analysis filter is first redesigned. The other functions in MATLAB for the design of FIR are tested. Moreover, other filter equations that satisfy the alias-free filter bank constraints are tried. In spite of this, no improvement is found. The incorrect phase shift of signals is still presented. Therefore, one more attempt is made by using different numbers of all pass filters in different levels to synchronize the phases. This is because an all pass filter can shift the phase of a signal without changing the amplitude of the signal. Nevertheless, although the phase shifts are adjusted, the signal still cannot be reconstructed perfectly.

From the effort spent on the redesign of the filters, it is concluded that there are three possible reasons the QMF bank does not provide perfect reconstruction in the software implementation. The first reason is that the low pass filter generated by MATLAB does not have its cut-off frequency at the first quarter of Nyquist frequency. The second reason is that the filter contains ripples in the frequency response, as shown in figure 7. The ripples may cause the values to be imperfect after several levels of filtering. The last reason is that the up-sampling and down-sampling operations may cause the values of the reconstructed signal to be imperfect. This is because non-zero values are discarded in down-sampling and zeros are inserted in up-sampling. At last, it is concluded that instead of concentrating on further improvement of the QMF bank, a different solution should be sought. It is because resource has stated that QMF bank will have imperfection in real practice due to sampling rate alterations and non ideal filter design (Mitra, 1999, p.706). As a result, complementary filter bank is used.

4.1.2 Complementary filter bank

The magnitudes of frequency responses of complementary filters can be added up to 1 (Heller, 2003). A signal can be decomposed by a pair of complementary low pass and high pass filters. The decomposed values can be combined by the same set of filters. Since the magnitudes of the frequency responses can be summed to 1, the signal can be perfectly reconstructed. The complementary filter bank developed can provide perfect reconstruction by employing ideal filters. Furthermore, there are no sampling rate alterations. Therefore, the difficulties encountered in QMF bank can be avoided.

The filters used in the QMF bank implementation are generated by MATLAB functions. There are imperfections such as ripples in the frequency responses of the filters. Hence, simple multiplications of data in the frequency domain are used for generating complementary filters. First, a signal is framed and transformed to the frequency domain by discrete cosine transform (DCT). DCT is used because it operates on real numbers only thus requires less computational complexity. The coefficients obtained from DCT represent the magnitudes of the signal at frequency of 0 to the Nyquist frequency. Therefore, low pass filtering can be achieved by multiplying the desired low frequency values with 1 and unwanted high frequency values with 0. In this case, an ideal filter with a sharp cut off frequency is realized. After the filtering by an ideal filter, no phase shift is introduced. Thus, there is no delay of subbands after the analysis and synthesis filter. This ensures that echoes will not be present. Since the sampling rate alterations may introduce errors in signal reconstruction, there is no down-sampling and up-sampling in this complementary filter banks.

By multiplying frequency components directly without sampling rate alterations, there is no need for separation of levels. The desired frequency components can be extracted directly by one multiplication. However, the notion of levels in QMF bank is still adopted. This is because the magnitude of low pass filtering can be easily altered by adjusting the number of levels in the filter bank. As in QMF bank, a level in complementary filter bank is defined as when the frequency components of a signal are divided in half. For example, if the lowest 1/8 of a signal is extracted, the filter bank has 3 levels of filtering.

In the synthesis part of the filter bank, the low passed and high passed signal are simply combined in the frequency domain. Then, the combined values are transformed back to the time domain by inverse DCT. A signal can be perfectly reconstructed since there are no sampling rate alterations and DCT is a reversible transform.

The complementary filter bank is implemented successfully. Input signals are perfectly reconstructed after the filter bank. Therefore, the filter bank process is concluded and the low frequency subband is passed to the second stage of the watermarking scheme for spectrum spreading.

4.2 Direct Sequence Spread Spectrum

In the second stage of the watermarking scheme, the low frequency output from the complementary filter bank is spread in the time domain by a technique called direct sequence spread spectrum. A type of this method called slow spread spectrum is used. The spread spectrum operation will increase the bandwidth of a signal, i.e. spread the frequency spectrum of a signal. The result can be achieved by multiplying a signal with a random binary sequence called pseudo noise sequence. The effect of spread spectrum can be reversed by multiplying the PN sequence with the spread signal. The reason for using spread spectrum is to increase the security of a signal (Furukawa et al., 2002).

4.2.1 PN sequence

Pseudo noise sequence (PN sequence) is used for spreading the spectrum of the low frequency signal in the time domain. This sequence is a binary periodic random signal. It can be generated by using a feedback shift register. The register is a feedback circuit that consists of a number of flip-flops to store one bit memory and a logic circuit to perform modulo-2 additions. The structure of the PN sequence generator is given in figure 10. The parameters for a PN sequence generation are the number of flip-flops, the initial stage of them and the feedback logic. At each clock period, the value of one flip flop is transferred to the next. At the same instance, the logic circuit will perform addition on the values of the flip-flops. Therefore, different PN sequences can be generated by changing these parameters. If there are m flip-flops in the shift register, the maximum number of states will be $2^m - 1$. The PN sequence generated by this number of states is called maximal-length-sequence or m-sequence. Any m-sequence is a random sequence. In the autocorrelation of an m-sequence, a high correlation value can only be obtained when one signal is phase shift by a multiple of 2π radians (Haykin, 1994, p.480). Hence, m-sequences are used in the spread spectrum watermarking techniques for detection of watermarks. The random nature of the sequences also implies that they are difficult to be decoded. A sequence with an arbitrary initial state and length is used in the software implementation, since the parameters do not interfere with the aim of the project. These parameters can be changed easily if desired. The feedback shift register used is provided by MATLAB toolbox.

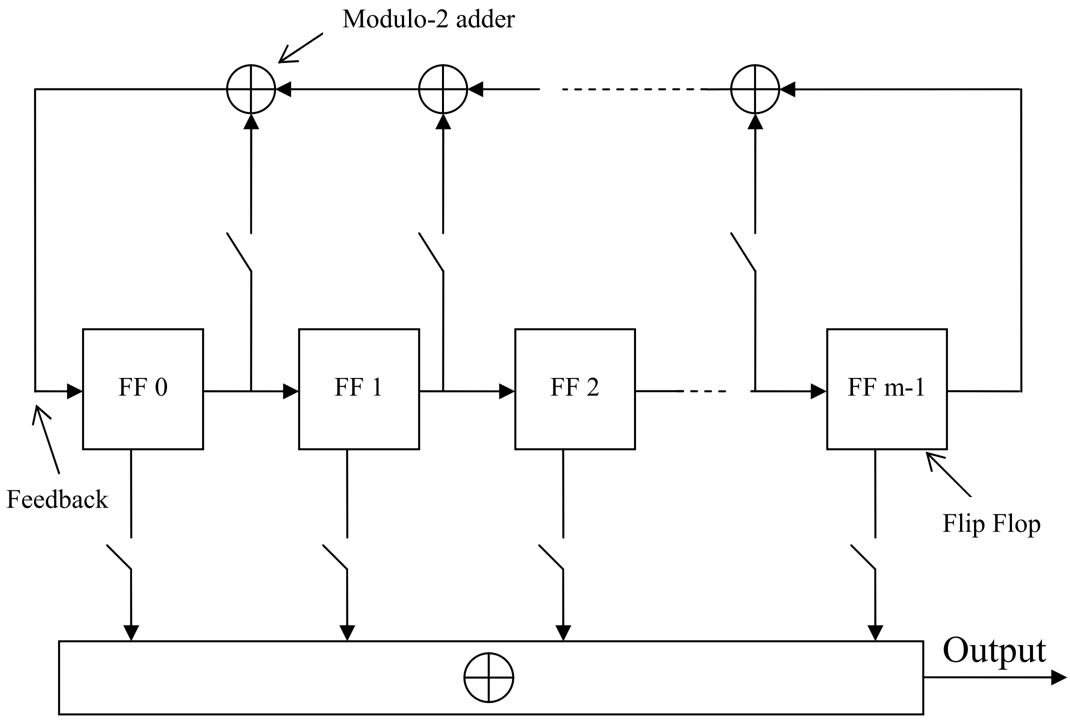

Figure 10. PN sequence generator

4.2.2 Slow spread spectrum

The first step to implement spread spectrum is obtained a PN sequence of 0s and 1s. Then, all 0s in the sequence is converted to -1s. The output with only 1 and -1 is used to multiply an input signal to spread the spectrum of the input. The original input signal can be retrieved by multiplying the spread signal with the PN sequence of 1 and -1. This is possible because 1 x 1 = 1 and -1 x -1 = 1, which is equivalent to multiplying the input signal by 1.

Spread spectrum can be categorized as fast or slow according to the relative rate of the input signal to the PN sequence. The type of spread spectrum is determined by the frequency of the signal and the frequency of the PN sequence. The frequency of the PN sequence can also be referred to as the chip rate, as each bit of the sequence is known as a chip. In fast spread spectrum, the chip rate is higher than the signal rate. Each input data is multiplied by a whole PN sequence. In slow spread spectrum, the chip rate is higher than the signal rate. Several input data is multiplying by a chip of the PN sequence (Fenwick, 2003). The following figure illustrates the difference between the two techniques.

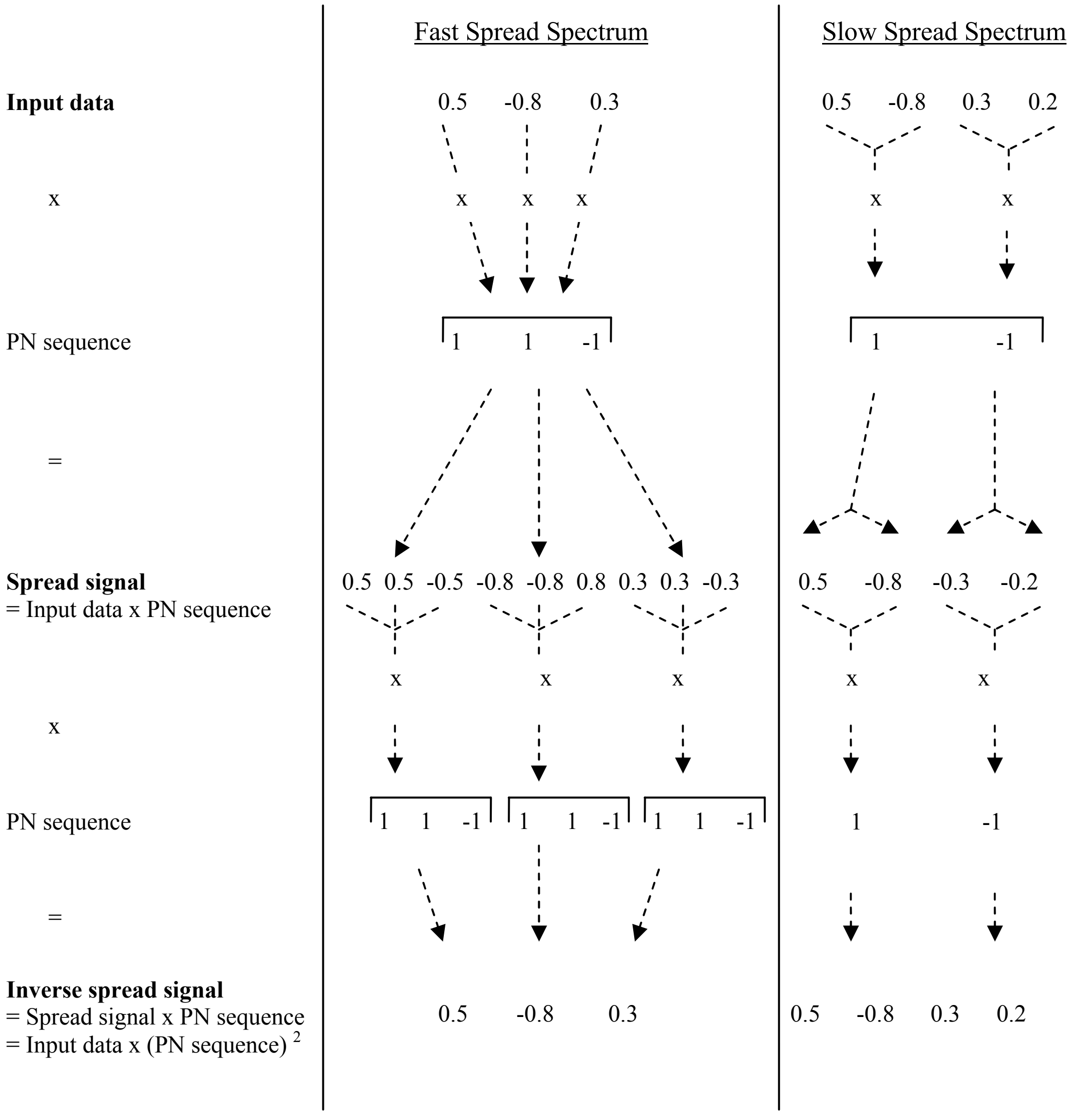

Figure 11. Fast spread spectrum vs. slow spread spectrum

As the spread signals in figure 11 shown, the number of data generated from fast spread spectrum is expanded by the length of the PN sequence after spreading, while the output of slow spread spectrum has the same length as the input signal. An increase in the data size will increase the duration of computation. Moreover, the signal in this stage will need to be combined with the filtered components using the same size. In fast spread spectrum, the signal will not be able to be changed back to the original size if it is altered. Although this output can be scaled back to the same size as the input, a scaling sequence will be needed in both the embedding process and the extraction process. It is not feasible to include one more sequence for a user to extract watermarks. Thus, slow spread spectrum is used in the project.

4.3 Modified Discrete Cosine Transform

A transform technique called modified discrete cosine transform (MDCT) is used in this stage of the watermark scheme. A spread signal in the time domain is produced in the spread spectrum stage. This signal is transformed to the frequency domain by MDCT. The outcome of the transform will be used to carry the watermark information (Furukawa et al., 2002).

The output of MDCT is a set of coefficients that represent a signal in frequency domain using cosine waves. MDCT is a derivation of DCT. A signal is first divided into frames before the transformation. MDCT coefficients are calculated by overlapping 50% of two consecutive frames. The overlapping and a sinusoidal window of the transform allow a signal to be converted from time domain to frequency domain and back with perfect reconstruction. Distortions due to framing can be eliminated (Vaananen & Vilermo & Wang & Yaroslavsky, 2000). Therefore, MDCT is used to produce frequency coefficients for watermark embedding.

The following equations are implemented in MATLAB to calculate MDCT coefficients.

$$X_i(k) = \frac{2}{M}\sum_{n=0}^{2M-1} w(n)c(k,n)x(n+iM) \qquad 0 \le k \le M-1,\ 0 \le n \le 2M-1 \tag{13}$$

where

$$w(n) = \sin\left(\frac{\pi(2n-1)}{4M}\right) \qquad 0 \le n \le 2M-1 \tag{14}$$

$$c(k,n) = \cos\left(\frac{\pi(2k+1)(2n+M+1)}{4M}\right) \qquad 0 \le k \le M-1,\ 0 \le n \le 2M-1 \tag{15}$$

The variable:
i is the frame index,
k is the coefficient index in each frame,
n is index of time domain sample in each frame, and
M is the total number of coefficients in each frame.

The coefficients of each frame are composed of three major elements. These are $w(n)$, $c(k,n)$ and $x(n + iM)$. $w(n)$ is a sinusoidal window function that can compensate the effect of distortion at frame boundaries and enable perfect reconstruction of a signal (Vilermo & Wang & Yaroslavsky, 2000). It is shown that the coefficients are related to cosine functions in $c(k,n)$. $x(n + iM)$ is a notation that represents a value of a time domain sample. The index $(n + iM)$ is the index of the sample before framing. From the constraint of k and n of a coefficient $X_i(k)$, it can be seen that in each frame $2M$ time domain samples are used to calculate M coefficients. However, from the index offset iM of $x(n + iM)$, it can be seen that the time domain sample is only shifted by M for each frame. This indicates that for $2M$ time samples, M of them are overlapped. This shows the property of 50% overlapping in MDCT.

In the software implementation of MDCT, the size of a frame is the sole parameter that has a specific fixed value. It is important to note that this size have to be integer product of 2 for the 50% overlapping to be exact.

A set of coefficients in the frequency domain is obtained from the MDCT calculations. These coefficients will be used for watermark embedding in the next stage.

4.4 Embedding

The step after MDCT is the fourth stage of the watermark embedding scheme. The actual watermark embedding operation takes place in this stage. The embedding is based on the notion that rounding to an even number represents watermark 1 and rounding to an odd number represents watermark 0. The implementation method of the project is an improvement to the method proposed by Furukawa, Konishi and Saito. The difference is in the scale of rounding.

4.1.1 Original method

The frequency coefficients from MDCT are used for embedding watermark. Each index of the coefficients is called a frequency key, k_f. All of the coefficients are first floored to integers, i.e. rounded down towards negative infinity.

$$e_i(k_f) \leftarrow \lfloor X_i(k_f) \rfloor \qquad \text{where } e_i(k_f) \text{ is a floored integer coefficient} \tag{16}$$

A watermark is a sequence containing bits 0 and 1. Each bit in the watermark will be related to one $e_i(k_f)$. Let the watermark bit for $e_i(k_f)$ be denoted by $b(k_f)$. If $e_i(k_f)$ is odd and $b(k_f)$ is 0, or if $e_i(k_f)$ is even and $b(k_f)$ is 1, then $e_i(k_f)$ will be changed to the ceiling value of the corresponding $X_i(k_f)$, i.e. $X_i(k_f)$ is rounded to integer towards positive infinity.

$$e_i(k_f) \leftarrow \lceil X_i(k_f) \rceil \qquad \text{if } \begin{cases} \lfloor X_i(k_f) \rfloor = odd \;\&\; b(k_f) = 0 \\ \lfloor X_i(k_f) \rfloor = even \;\&\; b(k_f) = 1 \end{cases} \tag{17}$$

There are maximum of four combinations for $X_i(k_f)$ and $b(k_f)$ since can only be $X_i(k_f)$ odd or even and $b(k_f)$ can only be 0 or 1. Therefore two combinations shown indicate that the other two combinations of the parameters must result in $e_i(k_f)$ to be the floored value of $X_i(k_f)$.

$$e_i(k_f) \leftarrow \lfloor X_i(k_f) \rfloor \qquad \text{if } \begin{cases} \lfloor X_i(k_f) \rfloor = odd \;\&\; b(k_f) = 1 \\ \lfloor X_i(k_f) \rfloor = even \;\&\; b(k_f) = 0 \end{cases} \tag{18}$$

It is important to notice that if $\lfloor X_i(k_f) \rfloor$ is odd, $\lceil X_i(k_f) \rceil$ must be even. Therefore, the significance of the rounding operation is that if the watermark bit is 0, $e_i(k_f)$ will always be even and if the watermark bit is 1, $e_i(k_f)$ will always be odd (Furukawa et al., 2002).

4.1.2 Modification

Modification of the proposed scheme was proven to be necessary when the algorithm is implemented into software. It is found that most of the coefficients calculated by MDCT have values ranging from $1\text{x}10^{-3}$ to $1\text{x}10^{-8}$. If these values are quantized to integers, the resulting sound wave after the embedding is altered significantly. The result of rounding coefficients to integers can be seen in figure 12. A possible reason for this situation is that all sound waves are normalized to a magnitude of 1 by MATLAB, whereas the sound waves used by Furukawa, Konishi and Saito may not be normalized. The solution to overcome the problem is to round the coefficients at a decimal level. Hence, experiments on the perceptibility of watermark introduction are conducted to find the threshold level. From the several trials performed with different audio signals, it is discovered that alterations made to the third decimal places of the coefficients are undetectable to human ears. Therefore, the rounding of coefficients must be at least at the third decimal place. However, it is also crucial to note that these coefficients will need to be converted in the inverse MDCT and reconstruction processes. Moreover, a watermarked signal may undergo other operations by intentional or unintentional attacks. It is doubtful that the coefficients will retain the identical values after these alterations. As the quantization is performed at levels of a lower decimal place, the accuracy of the values is more difficult to be preserved. Thus, the quantization is performed at the fourth decimal place as a balance of perceptibility and security.

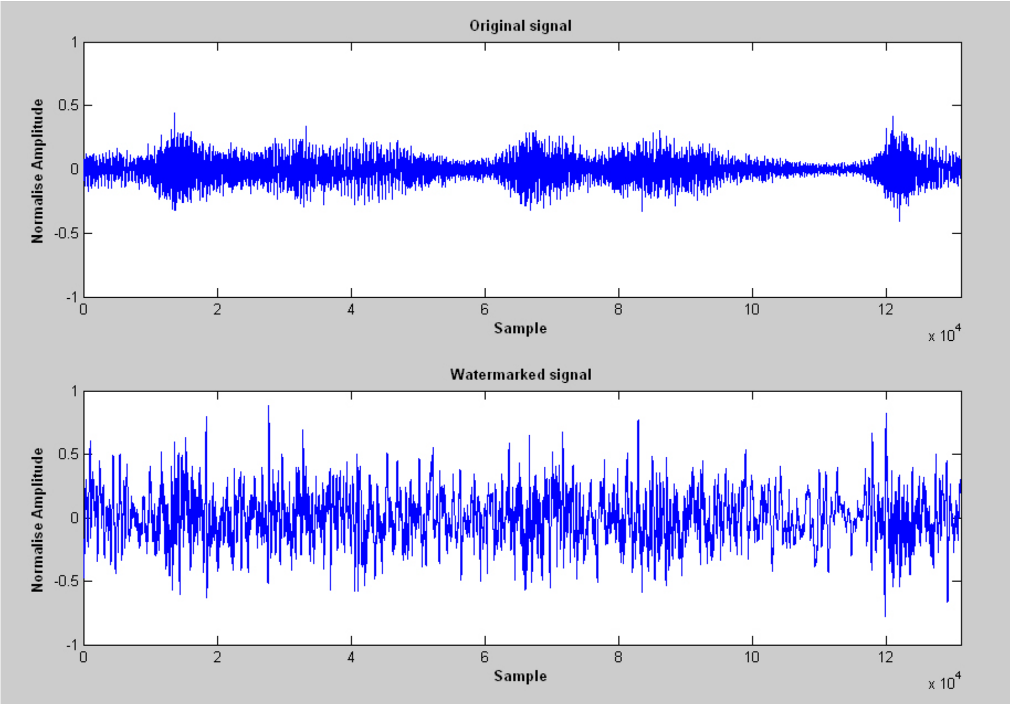

Figure 12. A signal before and after embedding (round to integer)

At a later stage of the project, the issue of enhancing watermark security is considered. By quantizing the values at different decimal places, the watermark will be more difficult to be extracted by unwanted parties. The PN sequence used in the spread spectrum procedure can be utilized here to randomize the quantization scale. In this way, the security of the watermark can be improved without any additional information to be kept for proper extraction. Therefore, the final rounding implementation for the project is set to 2 variable levels. The quantization will be at the fourth decimal place if the corresponding PN sequence bit is 1 and at the fifth decimal place if the PN sequence bit is 0.

The number of watermark bits must be less than or equal to the total number of frequency coefficients for the whole watermark to be embedded. In most cases, there are less watermark bits than the number of coefficients. In these circumstances, the watermark will be repeated periodically until all frequency components carry a watermark bit. In the extraction process, all watermark bits are extracted and averaged to calculate the watermark. Hence, the repetitions in the embedding allow tolerance against certain amount of error in individual watermark bit.

4.5 Reconstruction

After the watermark is embedded to the low frequency portion of the input signal, the portion can be recombined with the high pass filtered part to form the watermarked signal. Since the portion has gone through an analysis filter, spread spectrum and MDCT in the embedding process, it has to go through the inverse of those processes at reconstruction.

4.5.1 Inverse Modified Discrete Cosine Transform

The inverse process of MDCT is inverse modified discrete cosine transform (IMDCT). MDCT converts time domain sample to frequency domain. Therefore, IMDCT converts data in frequency domain to time domain.

A set of coefficients is generated from time domain data in the MDCT stage. These coefficients are transformed to carry the information of watermark after the embedding process. In the IMDCT stage, the coefficients with watermark are changed back to time domain according to the following formulas.

$$x_i'(n) = w(n)\sum_{k=0}^{M-1} c(k,n)X_i'(k) \tag{19}$$

$$x'(n+iM) = x_{i-1}'(n+M) + x_i'(n) \qquad 0 \le k \le M-1 \tag{20}$$

where
$x'_i(n)$ is the watermarked signal in the time domain in a frame and
$X'_i(k)$ is the watermarked frequency coefficients

The variables n, k and M in these formulas have the same meaning as in MDCT. The formulas for both $w(n)$ and $c(k,n)$ are identical as before as well. To perform IMDCT, time domain data in each frame, $x'_i(n)$, is first calculated. Then, the final result is the signal in time domain after framing is removed. It is denoted by $x'(n + iM)$. It is computed by adding a value at previous frame, $x'_{i-1}(n+M)$ and the current value, $x'_i(n)$. The addition is executed because each time domain data is calculated twice for overlapping frames in MDCT (Furukawa et al., 2002).

After the IMDCT on frequency coefficients is completed, a time domain data sequence will be obtained.

4.5.2 Inverse Direct Sequence Spread Spectrum

Inverse direct sequence spread spectrum (IDSSS) is used to inverse the effect of direct sequence spread spectrum. It is applied to the time domain output from IMDCT. IDSSS is carried out by multiplying the time domain data with a PN sequence of 1 and -1. Recall that in section 4.2.2, it was stated that the effect of spread spectrum can be cancelled out by the multiplication of spread signal with the original PN sequence used in spread spectrum. Thus, the PN sequence used before is applied to the time domain data in IDSSS. The chip rate of the PN sequence is also kept to be the same as in the DSSS process (Furukawa et al., 2002).

4.5.3 Synthesis Complementary filter

The last step of the watermark embedding scheme is to recombine the transformed low frequency subband with the filtered high frequency subbands. The transformed low frequency subband is the output from the IDSSS stage. It is represented in the time domain. In the implementation of the reconstruction, data is first converted to the frequency domain by DCT. The combination with high frequency subbands is carried out in the frequency domain. Lastly, the combined signal is changed back to the time domain by inverse discrete cosine transform (IDCT). This time domain signal is the final desired outcome of the watermarking scheme, a watermarked audio signal.

4.6 Extraction

Watermark can be extracted from a watermarked signal by procedures very similar to the embedding process. The watermarked input is first decomposed by the same complementary filter in the embedding process. The low frequency component of the input is retrieved. Then, this component will be spread by slow spread spectrum in the time domain. The PN sequence used in the spreading process is required to be the same PN sequence used in embedding. After spreading, MDCT is performed to obtain a set of frequency coefficients of the spread signal. If the watermarked signal has not been altered after its embedding process, these coefficients will be identical to the watermarked coefficients, $e_i(k_f)$, in the embedding operation (Furukawa et al., 2002). Recall that watermark is embedded at the fourth and fifth decimal places of the coefficients. Therefore, watermark is extracted by decoding the information at those locations of the coefficients.

A watermark bit can be extracted by rounding a coefficient at the fourth or fifth decimal place first according to the PN sequence. Checking is then performed to determine if each rounded coefficient is odd or even. As found in the embedding section, if $e_i(k_f)$ is even, the watermark bit is 0, and if $e_i(k_f)$ is odd, the watermark bit is 1. If the watermark is embedded repetitively into the signal, an averaging process will be performed to the watermark bits according the length of the watermark. The outcome of the averaging process is the extracted watermark.

5.0 Evaluations

After the complete watermark scheme is implemented, its performance is evaluated. The evaluation is carried out on the basis of the requirements of a watermarking scheme detailed in section 2.2. These requirements are the perceptibility, reliability, capacity and speed. The results from the evaluations provide a guideline for future improvements of the watermarking scheme.

5.1 Perceptibility

Two different tests are performed to evaluate the perceptibility of watermarked signals. One is a listening test and the other is the calculation of segmental signal-to-noise ratio (segmental SNR). Four different genres of audio signals are used in both tests to investigate the reaction of the watermarking scheme to audio signals with different characteristics. These signals are classical, pop, rock music and speech.

Six general students have participated in the listening test. They are presented with the original versions and the watermarked versions of the four signals. In every case, the audience cannot distinguish the differences in the signal. No distortion is perceived even at the silent part in the signal speech and classical music.

Segmental SNR is used for testing the power ratio of the original signal to the noise introduced after watermarked is embedded. The formula of the segmental SNR is given in equation 21 (Bruton & Gordy, 2000).

$$SNR = 10 \cdot \log_{10}\left\{\frac{\sum_{n=0}^{N-1} x^2(n)}{\sum_{n=0}^{N-1}\left[\tilde{x}(n) - x(n)\right]^2}\right\} \qquad (21)$$

where
$x(n)$ is a value in the original signal,
$\tilde{x}(n)$ is a value in the watermarked signal,
N is the index of a frame and
n is the index of a value in a frame.

To calculate the segmental SNR, the original signal and the watermarked signal are first divided into frames of length *n*. The signal power is the power of the original signal. The noise power is the power of the difference between the original signal and the watermarked signal. The segmental SNR can then be obtained by the total signal power in all frames divided by the total noise power in all frames. The result is converted to decibel scale by a logarithmic operation.

The results of the segmental SNR calculations are presented in figure 13. Furukawa et al. have found that the segmental SNR of the watermarked signals from QMF filter bank is about 87 dB (Furukawa et al., 2002). Figure 13 indicates that when there are 8 levels in the complementary filter banks, the resulting segmental SNR is similar. It can be seen that as the number of levels in the filter banks increases, the segmental SNR improves. However, the amount of information that can be carried by a cover signal will decrease since half the number of coefficients is discarded for every level. Therefore, an 8 level complementary filter bank is chosen as the optimal solution in this trade-off situation.

Segmental signal-to-noise ratio (dB)				
Level \ Genre	Classical	Pop	Rock	Speech
4	61.72	67.70	72.95	58.25
6	69.59	75.48	80.81	65.97
8	83.64	89.60	93.29	79.85
10	111.01	110.06	117.55	118.69

Figure 13. Table of segmental SNR

5.2 Reliability

The reliability of the watermarking scheme is tested by extracting the watermark directly after embedding and after compression. There are two methods to determine the reliability of the watermarking scheme. These are the bit error rate (BER) and the overall detection accuracy.

BER is the rate of the watermark bits correctly extracted compared to the watermark bits embedded. It is calculated by compared each watermark bit extracted with its corresponding embedded watermark bit. The result of BER when extraction is performed directly after embedding is presented in figure 14. In comparison to the 99.7% detection rate from the QMF method, the complementary filter bank scheme shows similar results (Furukawa et al., 2002).

Bit error rate (%)				
Level \ Genre	Classical	Pop	Rock	Speech
4	98.40	99.96	98.92	99.69
6	98.35	99.82	98.85	99.65
8	98.40	99.48	98.72	99.32
10	97.68	97.16	97.25	97.44

Figure 14. Table of bit error rate

The overall detection accuracy is found by comparing the extracted watermark with the original watermark. The extracted watermark is the watermark found after the averaging of all the extracted watermark bits. By testing in 4-, 6-, 8- and 10-level complementary filter bank, it is found that the overall detection accuracy is 100% for all four genres of audio signal.

The extraction tests have been performed to watermarked signals after mp3 compression at 128k bits per second. The results show that the watermark is lost after the compression. This is because the information embedded at the fourth and fifth decimal places is altered after compression. It is found that if the watermark is embedded at the first or second decimal places, the overall detection accuracy is 100%. This is an improvement in comparison to the 57% accuracy found in the QMF scheme (Furukawa et al., 2002). However, the perceptibility is not satisfactory if the watermark is embedded at these positions. Hence, it is concluded that the robustness to mp3 compression is yet to be enhanced.

5.3 Capacity

The capacity of a signal depends on the length of the original samples and the level of filtering. Since the level of complementary filter bank has been chosen to be 8, the capacity only depends on the length of input samples. For each level of filtering, the number of samples will be divided by 2. For an 8-level filter bank, a signal is divided by filters for 8 times. Therefore, the size of the output of the analysis filter bank can be calculated as the quotient of dividing the original sample size by 256 (2^8). For an input of 1 second, with sample rate of 44100Hz, approximately 170 watermark bits can be embedded. For 8-bit ASCII code, over 20 characters can be embedded in a 1-second segment. It is sufficient for song lyrics to be encoded.

5.4 Speed

The watermark embedding and extracting time are measured for filter bank of various levels. A 6-second sample is used in these timing tests since it contains 2^{18} samples, which is a power of 2. 50 tests are run on an Intel Pentium III 1.0 GHz computer. The results of these tests are summarized in figure 15. The time required for extracting watermark is approximately half the time for embedding. For a 4-level filter bank, the embedding time is over 11 minutes. For a 6-level filter bank, the embedding time is decreased to 10 seconds in average. For filter bank with more than 6 levels, both the embedding and extracting time are 2 seconds or less. Hence, this proves that an 8-level complementary filter bank is a suitable choice for the watermarking scheme.

Speed (s)		
Level \ Process	Embedding	Extracting
4	697	529
6	10	7
8	2	1
10	1	1

Figure 15. Table of watermark embedding and extracting speed

The 4-level filter bank has much longer process time because the information being processed is significantly larger. The computation complexity is therefore higher. Note that if the number of samples for embedding is increased, the computation complexity will also increase. Therefore, it is possible for the embedding process to consume a much longer time. This problem can be solved by dividing a long signal into several pieces of shorter signals for embedding. It is found that the transformation of the signal from a domain to another is the most time consuming operation. If the time for transformation can be decreased, the overall process time can be improved significantly.

6.0 Graphical User Interface

Although the watermarking scheme is implemented, the operations can only be controlled by command line input. Therefore, a graphical user interface (GUI) is written to enable access to the scheme for general users. The GUI allows users to carry out the full watermark scheme including embedding, extraction and analysis of the output data. The objects in the interface are laid out in a user friendly fashion for easy usage. A screenshot of the GUI is provided in figure 16.

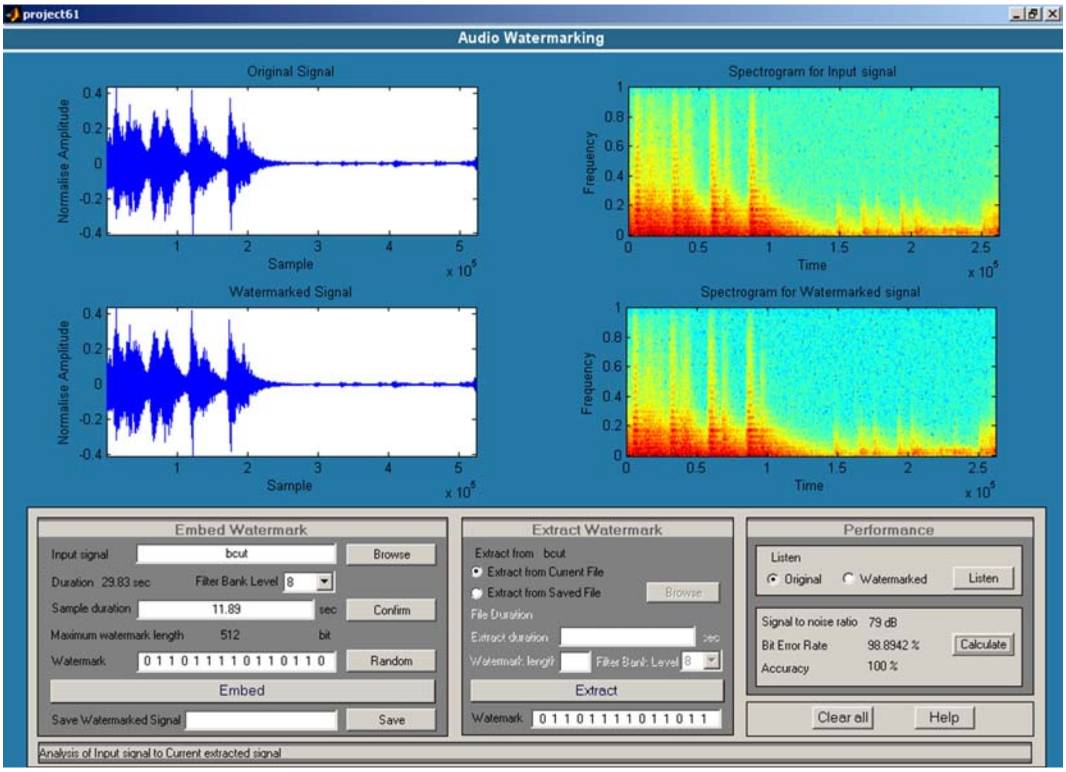

Figure 16. Screenshot of GUI

The GUI consists of mainly three frames of data controls and an area for graphical output. The three frames are for watermark embedding, watermark extraction and performance analysis. The area of graphical output is for displaying the graphs of an original cover signal and the watermarked signal in time domain and spectrograms.

In the watermark embedding frame, a user can first browse for an input file as the cover signal. The original signal and its spectrogram are plotted after the selection. The duration of the selected file is displayed. The user can listen to the signal by pressing the "Listen" button on the performance analysis frame. Then, the level of complementary filter bank can be chosen. Next, a period of the signal can be selected for watermark embedding. Since the data used for embedding can only be in a power of 2, the selected period will be adjusted if it does not satisfy this constraint. The maximum watermark length will be displayed to the user. After that, the user can enter a watermark or generate a random watermark. The random watermark generated by the "Random" button has a length equal to the maximum watermark length. When all the parameters are provided, the embedding can be performed by clicking the "Embed" button. The watermarked signal and its spectrogram are then displayed at the graphical area. The watermarked signal can be saved to hard disk for later use. The layout of the embedding frame is shown in figure 17.

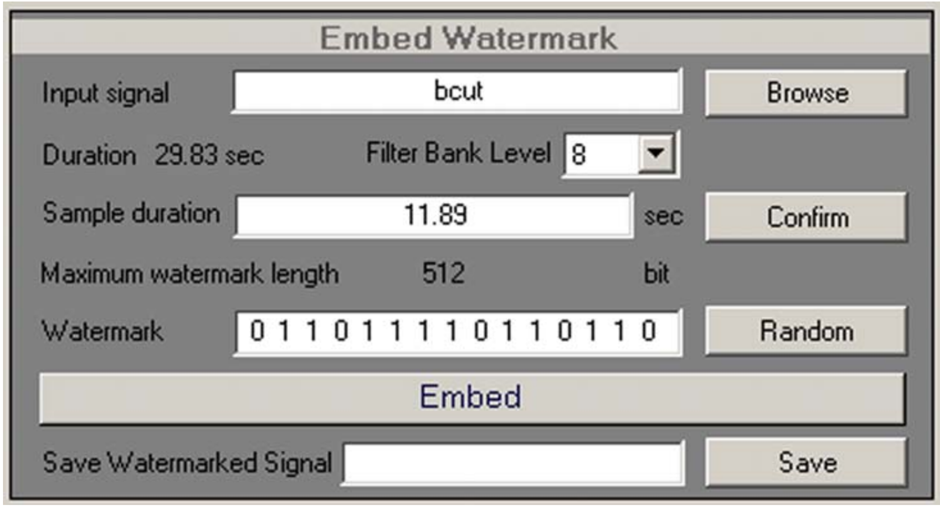

Figure 17. Screenshot of the embedding frame

The watermark extraction frame provides two options for selecting a watermarked signal for watermark extraction. If the watermarked signal is produced from the embedding frame, the watermark can be extracted by clicking the extract button. If the user chooses a signal from directory, the extraction duration, filter bank level and the length of the watermark and have to be specified for correct extraction. Since the watermark is not embedded to a whole signal in the embedding stage, extraction duration is required to indicate the watermarked duration of a signal. Figure 18 displays the layout of the extraction frame.

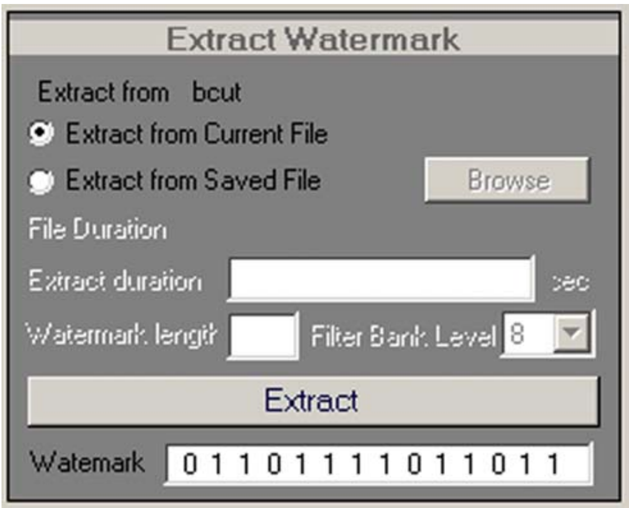

Figure 18. Screenshot of the extraction frame

The performance analysis frame is composed of two main functions. The first function enables user to listen to the original signal and the watermarked signal. This function allows the perceptibility of the watermarked signal to be tested. The second function provides the values of segmental SNR, BER and overall watermark detection accuracy to users. This allows the performance of the scheme to be tested as in section 5.0. The layout of the analysis frame can be seen in figure 19.

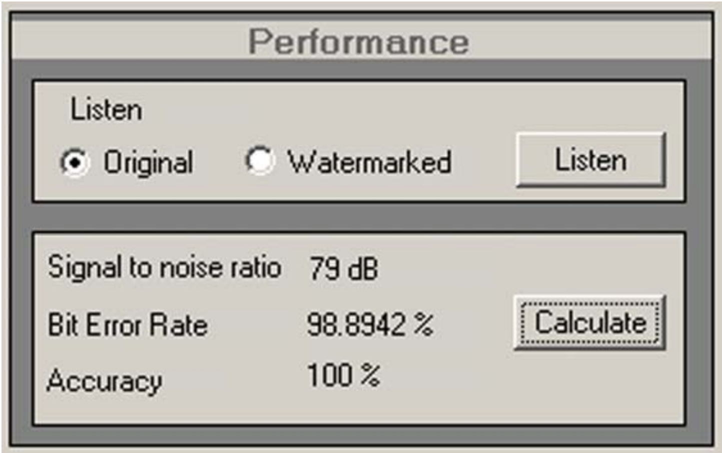

Figure 19. Screenshot of the performance analysis frame

Aside from the three main functionalities, the GUI also provides help files and a clear function. These are two types of help functions. The first one is a detail help file that guides users to use the software in a step-by-step manner. It is a pop up window triggered by the help button. The second one is a "quick help" bar located at the bottom of the interface. A user can right-click any object on screen and the functionality of that object will be introduced at the "quick help" bar. This bar also provides current status of the software to the user by messages such as "watermark is being embedded" and "watermark extracted successfully". The clear function is used for clearing all of the inputs and data on screen.

Error checking has been implemented to prevent errors from careless use of the software. Warnings are displayed as pop up dialog boxes to inform users in the case of any errors such as invalid inputs. The program has been debugged thoroughly.

From the functionalities described, it can be seen that the GUI is a user friendly interface that enables general users to perform the complementary filter based watermark scheme. A CD-ROM is included with the hard copy of this report to demonstrate the use of the GUI.

7.0 Future improvements

A wide range of improvements can be made to enhance the watermarking scheme in the project. From the results found in the evaluation, it can be seen that the speed of the watermarking scheme needs to be optimized by reducing the time for domain transformation process. This can be achieved by implementing the processes by a low-level language such as C or assembly language. The robustness of the scheme to mp3 compression has to be improved. Moreover, further experiments based on the requirements of SDMI can be performed to test the robustness of the scheme. The reliability of the scheme can then be enhanced by modifying the implementation techniques according to those test results. Although the capacity is not yet adjustable by users, this can be an objective for further improvements. The GUI can also be improved by incorporating more detail functions. These include alterations of frame sizes, generation of PN sequence and scale of quantization in the embedding stage. Moreover, since a watermark can only be embedded to a signal with a sample size of power of 2, the scheme can be improved by embedding the watermark into the whole signal. A possible method to implement this is to divide an input signal into sections with sample sizes of power of 2. For example, a signal with 640 samples can be divided into two sections with sizes of 512 and 128 samples respectively.

8.0 Conclusions

The report has shown that audio watermarking is chosen as the topic of this research-based project due to its recentness and importance to the music recording industries. Besides the functions as copyright protection, it is found that audio watermarking can also be used for fingerprinting and additional features to audio contents. The report has also found that in order to achieve these functionalities, a watermarking scheme has to meet the requirements of perceptibility, reliability, capacity and speed. It is shown that a watermarking scheme can be classified as private, semi-private or blind depending on the resources required in watermark extraction. Four watermarking schemes that based on the techniques of DC level shifting, QMF bank band division, frequency masking and spread spectrum are examined in this report. It has justified the decision of choosing the method based QMF bank band division for implementation because it is a blind watermark scheme and it is a more recent method. The report has stated the technical difficulties in perfect reconstruction in the implementation of the QMF scheme and the modifications made to address the problems. The successful implementation of a watermark scheme developed by using complementary filter bank band division is demonstrated in details by separating the scheme into four stages. These four stages are filter bank, spread spectrum, MDCT and watermark embedding. Then, the performance of the scheme is evaluated. The results from the evaluations are presented to show that the scheme in this project has satisfied the requirements of perceptibility perfectly. The reliability of the scheme is also remarkable under normal circumstances. However, the robustness to mp3 compression has to be improved. The capacity of the scheme is sufficient for song lyrics encoding. Furthermore, it can be made adjustable in the future. It is found that the computational speed can be improved by reducing the time for domain transformations. Next, the GUI written for easy access of the implemented scheme is presented. Lastly, the report has shown that further robustness tests are needed to understand the true reliability of the scheme under signal processing attacks.

9.0 References

Boney, L., Hamdy, K.N., Tewfik, A.H. (1996) Digital watermarks for audio signals. *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on 06/17/1996 -06/23/1996* [Online], p.473-480. Available: IEEExplore / 5357988 [2003, March 25]

Bruton, L.T., Gordy, J.D. (2000). Performance evaluation of digital audio watermarking algorithms. *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000* [Online], *Volume 1*, p.456-459. Available: IEEExplore / 7138482 [2003, July 18]

Craver, S.A., Liu, B., Wu, M. (2001). What can we reasonably expect from watermarks? *Applications of Signal Processing to Audio and Acoustics, 2001. IEEE Workshop on 10/21/2001 -10/24/2001.* [Online], p. 223-226. Available: IEEExplore / 7268211 [2003, March 22]

Donoho, D., Duncan, M.R., Huo, X., Levi, O. (1999). WaveLab802 for Matlab5.x. [Online]. Available: http://www-stat.stanford.edu/~wavelab/ [2003, April 10]

Fenwick, P. (2003). Spread Spectrum Radio. [Online]. Available: http://www.cs.auckland.ac.nz/compsci314s1c/lectures/97SpreadSpectrum.pdf [2003, May 30]

Hang, H.M. (2000). Subband coding. [Online]. Available: http://cwww.ee.nctu.edu.tw/course/source/srccdng_11.pdf [2003, March 16]

Heller, M. (2003). Complementary (Matched) Filters. [Online]. Available: http://gaia.ecs.csus.edu/~hellerm/EEE242/chapter%204/cmf.htm [2003, August 18]

Haykin, S. (1994). *Communication Systems, 3rd edition*. New York: John Wiley and Sons.

Jung, S.H., Lee, S.J. (2001). A survey of watermarking techniques applied to multimedia. *IEEE International Symposium on Industrial Electronics, 2001. Proceedings. 06/12/2001 - 06/16/2001.* [Online], *Volume 1*, p.272-277. Available: IEEExplore / 7091759 [2003, March 23]

Kirovski, D., Malvar, H. (2001). Spread-spectrum audio watermarking: requirements, applications, and limitations. *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on 10/03/2001 -10/05/2001* [Online], p.219-224. Available: IEEExplore / 7211241 [2003, March 22]

Malvar, H. (1999). A Modulated Complex Lapped Transform and its Applications to Audio Processing. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. [Online], *Volume 3*, p. 1421-1424. Available: http://www.telecom.tuc.gr/paperdb/icassp99/PDF/AUTHOR/IC992457.PDF [2003, March 25]

Mitra, S.K. (1999). *Digital signal processing laboratory using MATLAB*. Boston: McGraw-Hill.

Petitcolas, F.A.P. (2000). Watermarking schemes evaluation. *IEEE Signal Processing Magazine* [Online], *Volume 17*, Issue 5, p.58-64. Available: IEEExplore / 6746724 [2003, March 20]

SDMI, (2000). Call for Proposals for Phase II Screening Technology Version 1.0. [Online]. Available: http://www.sdmi.org/download/FRWG00033102-AMD1.htm [2003, August 19]

Saito, S., Furukawa, T., Konishi, K. (2002). A digital watermarking for audio data using band division based on QMF bank. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002* [Online], *Volume 4*, p.3473-3476. Available: IEEExplore / 7473940 [2003, March 22]

SysCoP (2000). Audio (C)opyright Watermarking. [Online]. Available: http://syscop.igd.fhg.de/audio.html [2003, March 16]

Uludag, U. (2001). *Audio Watermarking using DC Level Shifting.* Turkey: Umut Uludag

Vaananen, M., Vilermo, M., Wang, Y., Yaroslavsky, L. (2000). Some peculiar properties of the MDCT. *5th International Conference on Signal Processing Proceedings* [Online], *Volume 1*, p.61-64. Available: IEEExplore / 6918769 [2003, July 18]

Vilermo, M., Wang, Y., Yaroslavsky, L. (2000). On the relationship between MDCT, SDPT and DFT. *5th International Conference on Signal Processing Proceedings* [Online], *Volume 1*, p.44-47. Available: IEEExplore / 6918765 [2003, July 18]

10.0 Bibliography

Amaratunga, K., Strang, G. (2003). Wavelets, Filter Banks and Applications. [Online]. Available: http://web.mit.edu/18.327/ [2003, March 25]

Arnold, M. (2000). Audio Watermarking: Features, Applications and Algorithms. IEEE *International Conference on Multimedia and Expo 2000 Proceedings* [Online], *Volume 2*, p.1013-1016. Available: http://syscop.igd.fhg.de/Publications/Arnold02a.pdf [2003, February 20]

Berry, K., Byrnes, J.S., Byrnes, J.L., Hargreaves, K.A. (1994). *Wavelets and Their Applications.* Dordrecht: Kluwer Academic Publishers.

Bopardikar, A.S., Raghuveer, R.M. (1998). *Wavelet Transforms: Introduction to Theory and Applications.* Canada: Computer and Engineering Publishing Group, Addison Wesley Longman, Inc.

Brigham, E.O. (1988). *The Fast Fourier Transform and its Applications.* New Jersey: Prentice-Hall, Inc.

Chui, C.K. (1997). *Wavelets: A Mathematical Tool for Signal Analysis.* USA: Society for Industrial and Applied Mathematics.

Cobby, M. (No date). Fourier and Frequency Domain. [Online]. Available: http://www.spd.eee.strath.ac.uk/~interact/fourier/fourier.html [2003, March 19]

David, F.W. (2002). *An Introduction to Wavelet Analysis.* Boston: Edwards Brothers, Inc.

Dittmann, J., Steinebach, M. (2001). Joint watermarking of audio-visual data. Multimedia *Signal Processing, 2001 IEEE Fourth Workshop on 10/03/2001 -10/05/2001* [Online], p.601-606. Available: IEEExplore / 7196675 [2003, February 20]

Etter, D.M. (1993). Engineering Problem Solving with MATLAB. USA: Prentice-Hall, Inc.

Ingemar, J.C. Matthew, L.M., Jeffrey, A.B. (2002). *Digital Watermarking.* San Francisco: Morgan Kaufmann Publishers.

Johnson, S. (2003). Modified discrete cosine transform - From Wikipedia, the free encyclopedia. [Online]. Available: http://www.wikipedia.org/wiki/Modified_discrete_cosine_transform [2003, August 6]

Kaiser, G. (1994). *A Friendly Guide to Wavelets.* USA: Quinn-Woodbine

Kirovski, D., Malvar, H. (2001). Robust Covert Communication over a Public Audio Channel Using Spread Spectrum. [Online]. Available: http://www.cs.ucla.edu/~darko/papers/SSW2.pdf [2003, March 20]

Metois, E. (1999). Audio Watermarking and Applications. [Online]. Available: http://xenia.media.mit.edu/~metois/Projects/Waterm/waterm.htm [2003, March 16]

Polikar, R. (1999). The Engineer's Ultimate Guide to Wavelet Analysis. [Online]. Available: http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html [2003, March 30]

Valens, C. (1999). A Really Friendly Guide to Wavelets. [Online]. Available: http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html [2003, March 19]

Vanhoucke, V. (2001). Block artifact cancellation in DCT based image compression. [Online]. Available: http://www.stanford.edu/~nouk/mdct/mdct/ [2003, August 6]

11.0 Appendix

A CD-ROM is included in the hard copy of this report. This CD-ROM contains a video file demonstrating the use of the GUI.