Tsinghua University

Department of Electronic Engineering

Master Thesis

Audio Feature Extraction on Basketball Videos for Highlight Detection

Gonzalo Martínez Larumbe
2013

Resumen

Extracción de características de audio en vídeos de baloncesto para la detección automática de eventos destacados

Tutora: Mrs. Weibei DOU

Fecha de lectura: 19 de Julio de 2013

Universidad de Tsinghua
Departamento de Ingeniería Electrónica

Gonzalo Martínez Larumbe

Dada la amplia extensión de contenido multimedia hoy existente en Internet, la búsqueda manual de eventos destacados en un determinado vídeo supone una laboriosa tarea. Con el fin de evitar dicha búsqueda, este proyecto propone la detección automática de eventos destacados en partidos de baloncesto mediante la extracción de información relacionada con el audio y el vídeo.

Las ramas implicadas en este proyecto incluyen fundamentalmente las de inteligencia artificial y aprendizaje de máquinas para reconocimiento de patrones, procesado digital de imágenes y procesado digital de sonido. Es este último campo al cuál el proyecto desarrollado presta la mayor atención. Dado que la pista de sonido de los partidos de baloncesto contiene una gran cantidad de información, la obtención y procesado de características relacionadas con el audio supone un método razonable para conseguir una correcta deteción de eventos.

El procesado digital de audio y voz incluye una amplia variedad de técnicas para la recuperación de información. Algunas características han sido obtenidas en el dominio del tiempo, otras en el dominio de la frecuencia, mientras que algunas de ellas son consecuencia de técnicas más sofisticadas como el filtrado homomórfico o *cepstrum*, la predicción lineal, o el análisis mediante *wavelets*. Tras dicha obtención, una cuidadosa selección ha de ser llevada a cabo dando lugar a un *vector de características* para su posterior uso en la etapa del clasificador.

Varias consideraciones han sido tenidas en cuenta a la hora de extraer información de la pista de audio. El procesado digital de voz asume un modelo *fuente/sistema* por el cuál los fonemas sonoros son modelados como una excitación casi-periódica, consecuencia de la vibración de las cuerdas vocales, mientras que los fonemas sordos se modelan como ruido blanco gaussiano. Esta excitación es filtrada posteriormente por el tracto vocal,

proporcionando unas características espectrales adaptadas a la respuesta en frecuencia del mismo. Además, el sistema auditivo humano posee una percepción no lineal de las ondas sonoras tanto en amplitud como en frecuencia debido al funcionamiento de la cóclea en el oído interno. Este comportamiento es tenido en cuenta a la hora de extraer algunas características sonoras, llevando a cabo un mapeado de la escala de frecuencias de *Hercios* a *Mels* para su posterior post-procesado con el objetivo de incrementar la eficiencia global del sistema.

Para la obtención de las mencionadas características de audio es necesario llevar a cabo un análisis de tiempo corto de la señal de sonido dada la naturaleza no estacionaria de ésta. Para ello se recurre al uso de ventanas de 20 milisegundos de duración que segmentan la señal en cortas secuencias donde se puede asumir la estacionariedad, y donde por tanto pueden aplicarse las técnicas existentes en procesado digital de señales. Las características de audio obtenidas a partir de la pista de sonido pueden ser clasificadas en cinco diferentes categorías:

- **Dominio del tiempo:** son las más sencillas de procesar ya que no necesitan ser transformadas a diferentes dominios. De acuerdo a la naturaleza del sonido, el análisis en el dominio del tiempo proporciona información importante a un bajo coste computacional. Las características extraídas son:
  - *Energía de tiempo corto:* energía de la señal en cada ventana calculada a partir de la amplitud de las muestras.
  - *Ritmo de cruces por cero:* proporciona una medida aproximada del tono o *pitch* a partir de la relación existente entre los cruces por cero de una señal y la frecuencia de la misma.
  - *Función de autocorrelación de tiempo corto:* permite calcular el tono o *pitch* de la ventana bajo análisis a partir de la función de autocorrelación y las propiedades de ésta.
  - *Función de la diferencia en magnitud promediada:* característica similar a la función de autocorrelación que emplea diferencias en lugar de multiplicaciones y sumas con el fin de obtener el tono de una ventana.
  - *Máxima verosimilitud:* al igual que las anteriores, permite obtener el tono o *pitch* mediante la segmentación de la señal enventanada y la posterior comparación de dichos segmentos.
- **Dominio de la frecuencia:** suponen una valiosa fuente de información debido a la estructura del oído humano, dado que éste funciona como un analizador de frecuencias como consecuencia del funcionamiento de la cóclea. Sin embargo, son

más complejas de procesar que las características en el dominio del tiempo. Las características incluyen:

  - *Transformada de Fourier de tiempo corto:* análisis de Fourier de la señal enventanada que da lugar a la representación en dos dimensiones de una señal tridimensional: el espectrograma.
  - *Energía de subbanda:* representa la energía en determinadas sub-bandas del dominio de la frecuencia y permite la obtención de una aproximación del espectro con un decremento considerable de la dimensión del vector de salida.
  - *Producto de espectro armónico:* algoritmo para la detección del tono a partir del producto de versiones submuestreadas del espectro original.
  - *Máxima verosimilitud:* mediante el uso de posibles distribuciones espectrales, se llevan a cabo comparaciones con el espectro original y se obtiene el tono a partir de la distribución que más se asemeje a ella.
  - *Brillo:* medida que calcula el centroide del espectro.
  - *Ancho de banda:* estimación de la desviación energética respecto del brillo.
  - *Entropía espectral:* medida que estima la complejidad del espectro.
  - *Flujo:* variación espectral entre dos espectros adyacentes en el tiempo.
  - *"Rolloff":* frecuencia debajo de la cuál se encuentra el 85% de la distribución espectral.
  - *Factor de cresta y razón pico-promedio de potencia:* razón entre la amplitud de pico y la energía RMS de la distribución espectral.
  - *Rango dinámico:* razón entre la amplitud máxima y mínima del espectro.
  - *"Flatness":* medida que cuantifica cómo de tonal es un sonido.

- **Filtrado homomórfico:** derivado directamente del análisis en el dominio de Fourier, implica un mapeado no-lineal a un dominio distinto con el objeto de aplicar filtros lineales. En caso de que el dominio sea el logaritmo de la transformada de Fourier, entonces éste es denominado *cepstrum*. El cepstrum es ampliamente utilizado hoy en día para la detección del tono o *pitch*, y es una potente herramienta para la separación de fuente y sistema en el modelo asumido para la generación de la voz. Las características extraídas incluyen:

  - *Cepstrum:* se define como la transformada inversa de Fourier del espectro logarítmico y mide las variaciones existentes en el mismo. Hoy en día es considerado uno de los métodos más efectivos para el cálculo del tono o *pitch*.
  - *Coeficientes cepstrales en las frecuencias de Mel:* característica ampliamente utilizada en el campo del reconocimiento automático de voz que proporciona

una medida precisa del espectro con un reducido número de coeficientes y adaptada a la percepción humana del sonido.

  - *Coeficientes delta y delta-delta:* medidas que estiman la variación temporal de los coeficientes cepstrales en las frecuencias de Mel.

- **Modelos de predicción lineal:** la predicción lineal es una operación matemática en la cuál valores futuros de una secuencia temporal son estimados a partir de la combinación lineal de muestras pasadas. Actualmente es una potente herramienta para el procesado de voz, ya que permite suministrar estimaciones precisas de los parámetros vocales empleados en el modelo *fuente/sistema* con una gran simplicidad de computación. Las características extraídas incluyen:

  - *Coeficientes de predicción lineal:* permiten obtener una aproximación precisa de la forma del espectro de voz con un reducido número de coeficientes y un bajo coste computacional.
  - *Entropía de los coeficientes de predicción lineal:* miden la complejidad existente en los anteriores coeficientes.
  - *Coeficientes cepstrales de predicción lineal:* análogos a los coeficientes cepstrales en la escala de Mel, pero computados a partir de los coeficientes de predicción líneal, y no del espectro obtenido en el análisis de Fourier.
  - *Parejas espectrales lineales:* método alternativo y robusto para la codificación de los coeficientes de predicción lineal.

- **Análisis mediante wavelets:** la transformada Wavelet es un tipo especial de transformada de Fourier que representa una señal en términos de versiones trasladadas y dilatadas de una onda finita denominada wavelet madre. La transformada Wavelet se presenta como una alternativa multi-resolución al común análisis de Fourier. Las características obtenidas incluyen:

  - *Transformada Wavelet continua:* representación bidimensional de la señal unidimensional de audio a partir de variaciones en la escala y el desplazamiento.
  - *Transformada Wavelet discreta:* implementación discreta de la anterior transformada mediante el uso de escalas en potencias de 2.
  - *Transformada Wavelet estacionaria:* similar a la transformada discreta Wavelet que mantiene la propiedad de estacionariedad.
  - *Paquetes Wavelet:* generalización de la descomposición diádica utilizada en la transformada discreta Wavelet que permite una mayor flexibilidad en el análisis de la señal.

Contents

Chapter 1

Introduction

1.1 Motivation of this thesis

Analysis and classification of highlights in multimedia content has become an important field of study in recent years as a consequence of the growing number of multimedia databases. Research in this area has focused primarily on the use of audio and image information. In our scope, basketball matches are composed of audio and video content with a duration longer than one hour. Thus, manually seeking for highlights implies browsing through the whole video until finding the desired event, which may take long searching periods of time. In order to overcome such issue, highlights detection is proposed as a tool that automatically solves this manually based search [1] .

Several branches of knowledge are involved in this approach. Since the main field of study logically entails digital signal processing, many others complement it in an efficient manner, such as video and image processing, audio and speech processing, artificial intelligence, data mining or machine learning. However, this project focus only on audio and speech processing as results will be combined with an additional video processing approach of the same file. In a last step, a feature vector with audio and video information will be used as the input of a classifier to generate the proper keywords which will further decide when a highlight occur.

As an illustration, the clearest basketball highlights examples would be the ones most directly related to the match result, i.e., goal, half-time break, fouls or violations. At a lower level, keywords involve more basic procedures for a final highlight discrimination, i.e., excited commentator speech, excited audience and applause might indicate a goal has taken place, while whistling and excited audience might suggest some type of foul. Consequently, depending on which highlights are arranged to be detected, a different set of keywords must be specified.

Audio and speech processing include a wide variety of techniques for information retrieval. Some of the most well-known are implemented according to the aim of the project. Both time-domain and frequency-domain features are extracted from small windowed segments of the whole file, as well as features based on homomorphic filtering, linear prediction and wavelet analysis. Thereby, a careful selection of these features shall be made in order to improve the decision stage and the overall system efficiency.

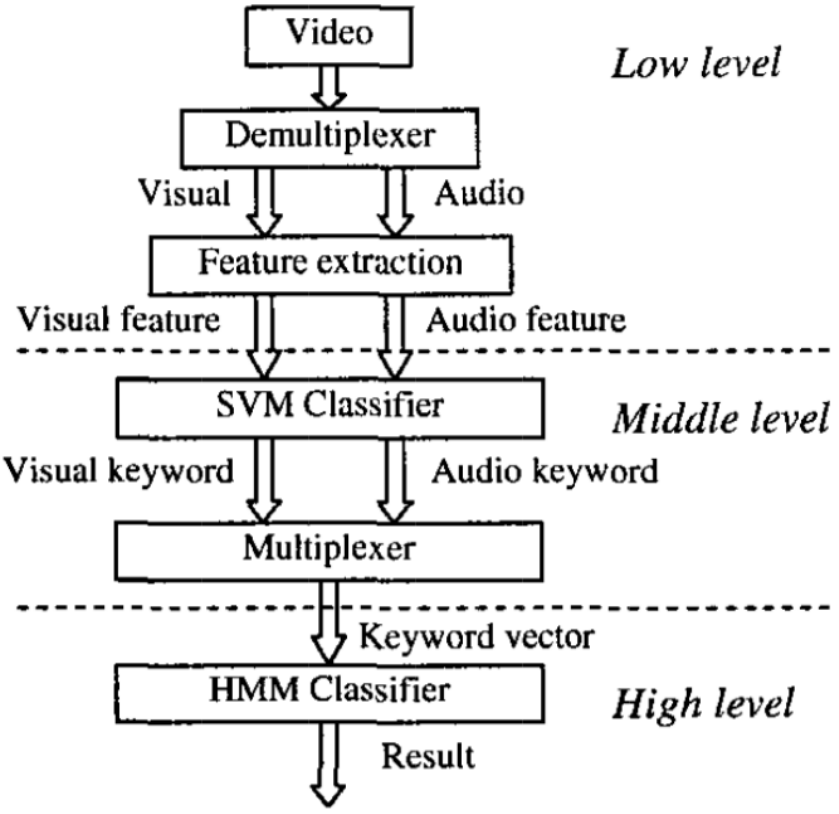


FIGURE 1.1: Proposed highlight detection method flowchart

As a summary, audio features are extracted from a basketball video file and used to generate keywords for further highlight detection. Since basketball matches contain plenty of audio information, previous approach seem to be a consistent method for successful highlight detection.

1.2 Summary

As it was previously mentioned, both time-domain and frequency-domain features are retrieved through a short-time windowed analysis. Homomorphic filtering and linear prediction are used as a basis to accurately estimate certain audio parameters, such as pitch, in a different manner as typical frequency analysis does. Finally, wavelets are

used as a flexible alternative to frequency analysis. Since this project focus mainly on audio features extraction, each of them will be explained deeply in subsequent chapters and a brief introduction is given below.

- Time-domain features are the most easily computable of the whole set. They do not need to be transformed into frequency domain with all the computation that process involves. Besides, according to sound nature this time-domain analysis gives us basic valuable information of the audio signal. Features include short-time energy (STE), zero-crossing rate (ZCR), short-time autocorrelation function (STACF), average magnitude difference function (AMDF) and maximum likelihood algorithm (ML) for pitch detection.

- Frequency-domain provides a set of very useful and intuitive features due to human hearing structure at the cost of computation. Frequency-domain features comprise short-time Fourier transform (STFT), sub-band energy (SBE), harmonic product spectrum (HPS), maximum likelihood (ML) and diverse spectral distribution statistics and derivatives. Nevertheless, the speech signal is considerably complex and sometimes more efficient techniques are required, such as homomorphic filtering or linear prediction.

- Homomorphic filtering derives directly from Fourier analysis and implicitly involves the concept of cepstrum in audio processing. Homomorphic filtering performs a non-linear mapping to a different domain in which linear filtering techniques are applied, followed by a mapping back to the original domain. Cepstrum results from a mapping to the logarithmic spectral domain in order to separate filter effects from excitation effects. Features extracted consist of power cepstrum, liftered spectrum, mel-frequency cepstral coefficients (MFCC) and its derivatives, delta and delta-delta coefficients.

- Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples. Nowadays, linear prediction is one of the most powerful and widely used speech analysis techniques. The importance of this method lies both in its ability to provide accurate estimations of the speech parameters and in its relative speed of computation. Features extracted include linear prediction coefficients (LPC), linear prediction entropy (LPCE), linear prediction cepstral coefficients (LPCC) and line spectral pairs (LSP).

- Wavelet analysis is presented as a multiresolution alternative to typical frequency analysis. Even though it is a relatively recent field, it has widely demonstrated its efficiency and applications due to its flexibility. Features extracted comprise

the continuous wavelet transform coefficients (CWT), discrete wavelet transform (DWT), stationary wavelet transform (SWT) and wavelet packets.

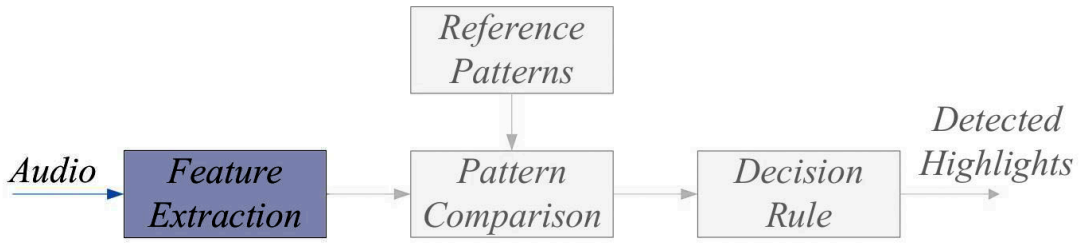


FIGURE 1.2: Audio feature extraction scope

Once the short-time analysis has been performed for the complete signal, an output text file is generated so as to combine it with additional video features in order to form a feature vector for further classification. Numeric outputs consist of a group of previously selected features, normalized and weighted depending on the feature dimension. This adjustment improves classifiers precision and recall as a consequence of taking into consideration the effect of larger dimension features that implicitly carry more information.

1.3 Audio characteristics and speech model

Sound is a vibration that propagates as a mechanical wave of pressure and displacement through some medium, commonly air. In an analog audio signal the instantaneous voltage varies continuously with the pressure of the sound waves, whereas in a digital audio signal this continuous pressure is represented by a discrete function which can only take on one of a finite number of values.

In the case of highlight detection, which is the last object of this project, audio is extracted directly from the video file and processed digitally afterwards in order to detect such highlights. MATLAB is the software environment used to digitally process the audio file and extract the required audio features. Some of them are processed in the time domain for simplicity, while others are processed in the frequency domain due to the nature of human hearing. Besides, the audio track of a basketball match has a length larger than an hour, and consequently it must be analyzed in small segments for retrieving appropriate results.

Digital signal processing of audio signals involves some important considerations, like the sampling frequency selection. In this project the CD standard of 44,1 KHz will be used for some reasons. Although speech requires only a sampling frequency of 8 KHz in order

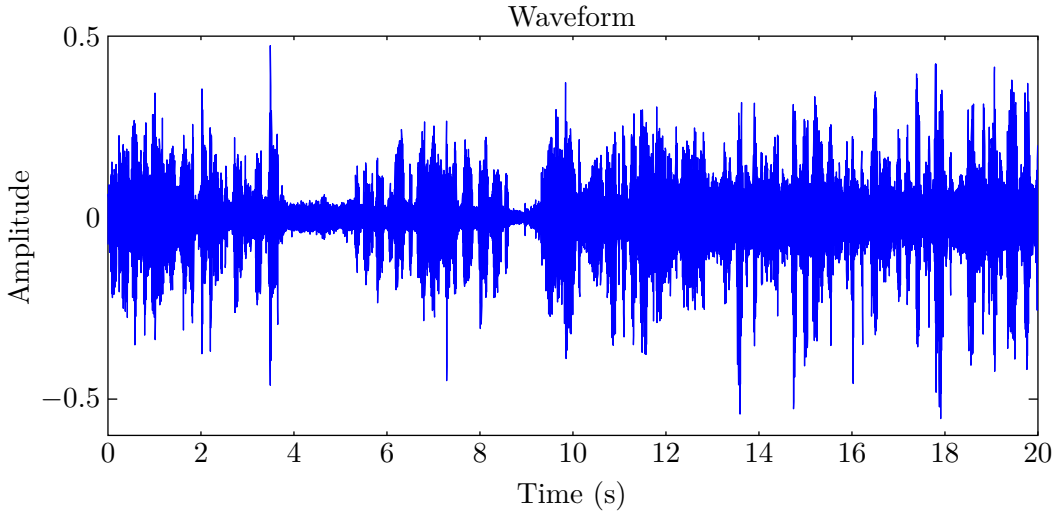


FIGURE 1.3: Waveform of a 20 sec basketball game extract

to properly reconstruct the most important frequency content, some other information is present in the audio signal for the human hearing range, which extends from 20 Hz to 20 KHz. Some particular high-frequency components might help us classify audio events as certain keywords in the decision stage, which entails a sampling frequency of at least double the human hearing range, according to *Nyquist–Shannon* sampling theorem. Thus, a minimum 40 KHz sampling frequency would be required. In order to avoid aliasing, signals must be low-pass filtered before sampling, and as a consequence sampling frequency raises up to 44.1 KHz.

Our approach for audio feature extraction will focus on interpreting some characteristics of the audio signal, such as energy, pitch or frequency response. Since speech plays an important role in this project (a commentator's voice is present during the whole audio track) some speech processing techniques are utilized as well. The importance of speech processing lies in the possibility of representing speech production as a *source/system* model [2].

Speech model

In the first place, it is necessary to point how speech is physically produced according to human anatomy. Figure 1.4 shows an approximate model. The vocal tract is modeled as a tube of nonuniform cross-sectional area that is bounded at one end by the vocal cords and at the other by the mouth opening. This tube serves as an acoustic transmission system for sounds generated inside the vocal tract and varies its shape with time due to motions of the lips, jaw, tongue, and velum. The speech waveform is created by the sound sources in the vocal tract, and the resonances of the vocal tract tube shape these

sound sources into phonemes. Hence, speech can be represented phonetically by a finite set of symbols called phonemes.

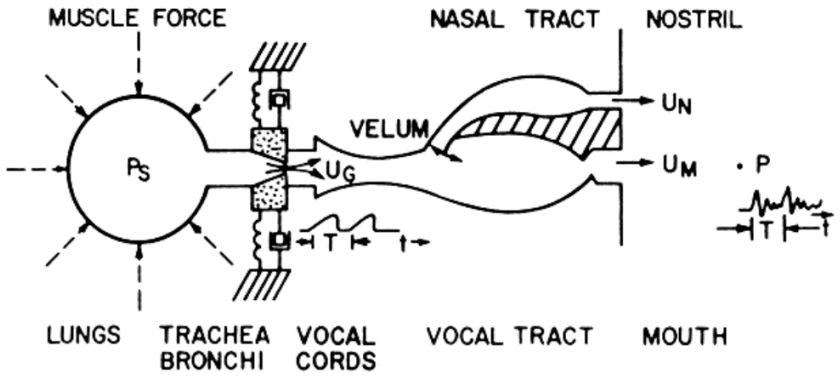


FIGURE 1.4: Schematic model of the vocal tract system

The sounds of speech are generated in several ways, although only two are considered in this model. Voiced sounds are produced when the vocal tract tube is excited by pulses of air pressure resulting from quasi-periodic vibration of the vocal cords, whereas unvoiced sounds are produced by creating a constriction somewhere in the vocal tract tube and forcing air through that constriction. Consequently, voiced sounds are modeled as periodic impulse train excitations and unvoiced sounds as random white noise excitations.

In general this model is called a *source/system* model of speech production. The time varying frequency response of the linear system simulates the frequency shaping of the vocal tract and is assumed to be slowly-time-varying compared to the speech signal, i.e., over the timescale of phonemes, the impulse response, frequency response, and system function remains relatively constant. Hence, the speech signal can be written in terms of the the overall model parameters as follows

$$s[n] = \sum_{m=0}^{\infty} h[m]e[n-m] \tag{1.1}$$

where $s[n]$ is the modeled speech signal, $h[n]$ the vocal tract impulse response, and $e[n]$ the input excitation to the filter, a periodic impulse train for voiced sounds or random white noise for unvoiced sounds. In either case, the linear system imposes its frequency response on the spectrum.

In some speech processing techniques, such as linear prediction, it is more appropriate to represent the vocal tract filter in terms of an all-pole frequency response as this simplifies the analysis required to estimate the parameters of the model from the speech signal.

Including the excitation effect of G in the transfer function for convenience leads to

$$H(z) = \frac{G}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{1.2}$$

where the filter coefficients a_k change at a rate on the order of 50–100 times/s. Some of the poles of the system function lie close to the unit circle and create resonances to model the formant frequencies. This all-pole model is a natural representation for non-nasal voiced speech although it also works reasonably well for nasals and unvoiced sounds.

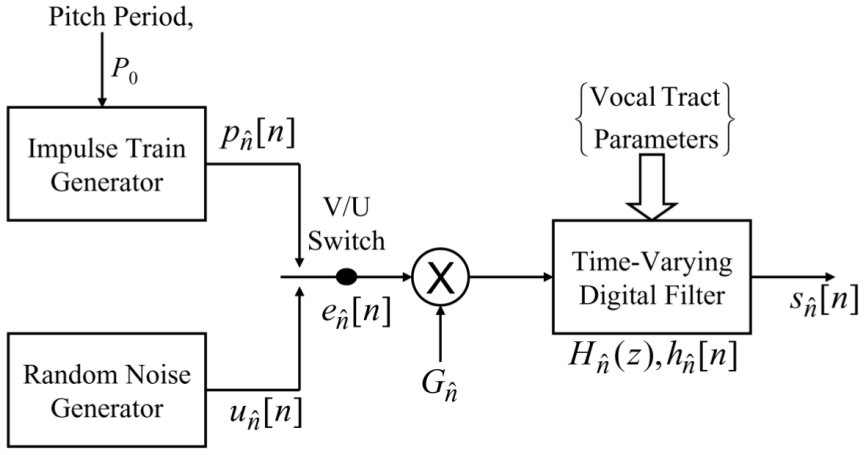


FIGURE 1.5: Speech model for voiced/unvoiced signals

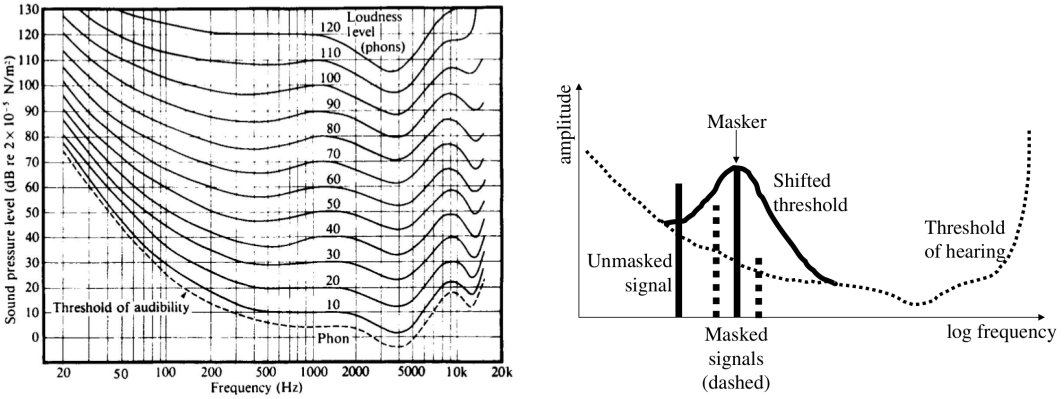
As a summary, speech can be represented as the output of a slowly time-varying digital filter with a voiced/unvoiced excitation, i.e., speech is represented by the parameters of the model instead of the sampled waveform. This is illustrated in figure 1.5. Such parameters can be estimated through short-time analysis assuming stationarity of the audio signal and the model for short time intervals.

Human hearing

The human ear is composed of three different sound processing sections: outer ear, middle ear and inner ear. The outer ear consists of the pinna, which gathers sound and conducts it through the external canal to the middle ear. Middle ear is comprised of the eardrum at the beginning, and three small bones (hammer, anvil and stirrup) which perform a transduction from acoustic waves to mechanical pressure waves. Lastly, inner ear consists of the cochlea and neural connections to the auditory nerve, which conveys neural signals to the brain.

Is in this part of the inner ear, the cochlea, where mechanical vibrations are transformed into neural activity interpreted by human brain. The cochlea is composed of a set of inner hair cells called basilar membrane that vibrates in a frequency-selective manner along its extent. Thereby a rough spectral analysis of the sound is performed.

The non-uniform frequency analysis carried out by the basilar membrane can be modeled as a set of band-pass filters whose frequency responses become increasingly broad. Besides, these frequency responses overlap significantly since points on the basilar membrane cannot vibrate independently of each other. This phenomenon yields some important consequences on some audio perception attributes, such as loudness, pitch or frequency masking.



(A) Loudness perception (B) Auditory masking

FIGURE 1.6: Human hearing non-linear perception

Loudness is a key element in the perception of audio. It is defined as an auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud. Therefore, it is closely related to audio amplitude and its perception is frequency-dependent as shown in figure 1.6. These curves illustrate that the auditory system is most sensitive for frequencies around 3 KHz to 4 KHz, and at the same time this is precisely the range of frequencies occupied by most of the sounds of speech. On the other hand, masking occurs when one sound makes a second superimposed sound inaudible. Loud tones causing strong vibrations at a point on the basilar membrane can mask other vibrations that occur nearby and pure tones can be masked by either other pure tones or noise. Finally, pitch is an attribute of sound that is related to its fundamental frequency, and like loudness its perception is subjective.

In conclusion, human hearing is based on a non-linear frequency analysis due to the basilar membrane functioning. This fact points frequency domain audio features as a key part of this project.

Pitch

As it was introduced in previous section, pitch is an auditory sensation in which a listener assigns musical tones to relative positions on a musical scale based primarily in the frequency of vibration. It is therefore a subjective attribute according to which sounds can be sorted on a scale from low to high depending on the wave frequency, i.e., it takes a human mind to map the internal quality of pitch.

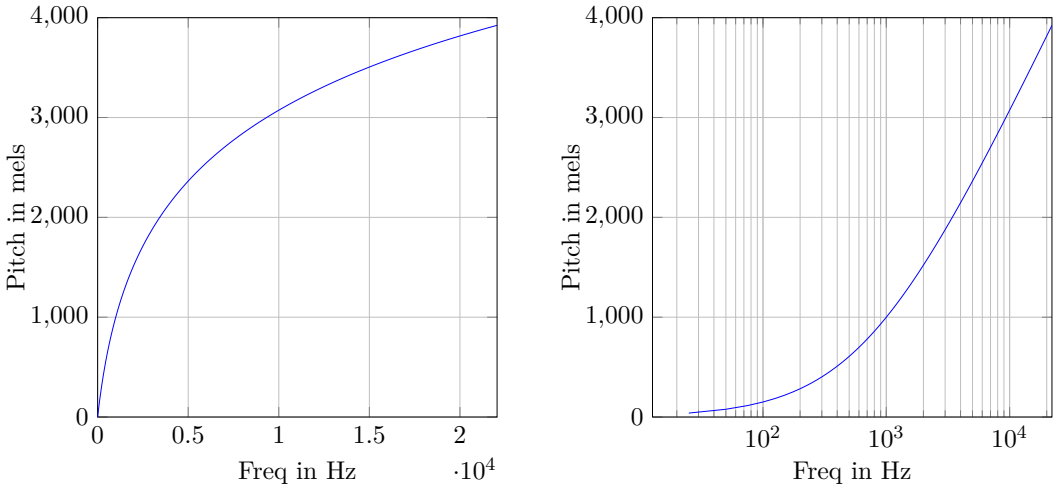
Most musical sounds have a periodic structure when viewed over short time intervals, and thereby an associated pitch. Human perception of musical intervals is approximately logarithmic with respect to fundamental frequency. The widely used MIDI standard maps this fundamental frequency f to a real number p and is expressed as

$$p = 69 + 12 \times \log_2 \left(\frac{f}{440 \text{ Hz}} \right) \tag{1.3}$$

The fundamental frequency of speech can vary from 40 Hz for low-pitched male voices to 600 Hz for children or high-pitched female voices. According to the previously mentioned model in 1.5, speech consists of voiced and unvoiced sounds. Voiced speech is quasi-periodic due to the vibration of the vocal cords, and although it contains many frequencies, many of the results obtained with pure tones are relevant to the perception of voice pitch as well. Moreover, the fundamental frequency of the glottal excitation determines the perceived pitch of the voice. On the other hand, absence of a perceptible pitch may indicate the presence of an unvoiced sound. Thus, an accurate pitch detection algorithm may lead to accurately distinguish voiced sounds from unvoiced sounds, or even male voices from female voices. The relationship between pitch (measured on a non-linear frequency scale called the mel-scale) and frequency of a pure tone is approximated by the equation

$$Pitch\ (mels) = 1127\ log_e \left(1 + \frac{f}{700} \right) \tag{1.4}$$

This expression is calibrated so that a frequency of 1000 Hz corresponds to a pitch of 1000 mels. Due to the basilar membrane nature, it was previously stated that human hearing spectral analysis can be modeled as a set of band-pass filters with an increasing bandwidth. Eventually, it turns out that more or less independently of the center frequency of the band, one critical bandwidth corresponds to about 100 mels on the pitch scale, and as a result mel-scale depicts a linear representation of subjective human pitch perception (the name mel comes from the word melody to indicate that the scale is based on pitch comparisons).



(A) Linear scale (B) Logarithmic scale

FIGURE 1.7: Pitch perception on the mel-scale

Many methods exist for pitch detection. In this project some audio features are implemented in order to obtain a clear estimation, such as short-time autocorrelation, harmonic product spectrum or cepstrum whereas other features give an insight, such as short-time Fourier transform, zero-crossing rate or brightness. All of them are thoroughly detailed in further sections.

1.4 Classifiers

In machine learning, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. Hence, a classifier is an algorithm that implements classification and maps input data to a category. Classifiers prediction is performed from a feature vector which describes measurable properties of the input data, and each of these properties is referred to as a feature.

In the scope of this project, features are extracted from the digital sampled audio signal existing in basketball video files, used as an input for different types of classifiers at various levels of the whole system. For instance, one type of classifier might be used for keyword generation from audio features, while other might be used for highlight detection from previous generated keywords. However, even though classifier implementation goes beyond this project, it implies an important step to understand how the overall system works.

A proper choice of the classifier leads to an enhancement of precision and robustness of the whole detection system. Different classifiers may be used depending on the needs of

the system. For instance, HMM classifier is widely used in speech recognition whereas GMM is preferred in speaker discrimination. The most commonly used classifiers are next presented.

Gaussian Mixture Model

A Gaussian mixture model (GMM) is a type of density model which comprise a number of component Gaussian functions. These functions are combined to provide a multimodal density function. Mixture models are a semi-parametric alternative to non-parametric histograms and provide greater flexibility and precision in modeling the underlying statistics of sample data.

During the training phase various Gaussian components can be used to learn the probability density functions of the audio signal. In the decision phase, the learned GMMs for individual audio types can be used to calculate the likelihood that an instance of a particular audio type presents.

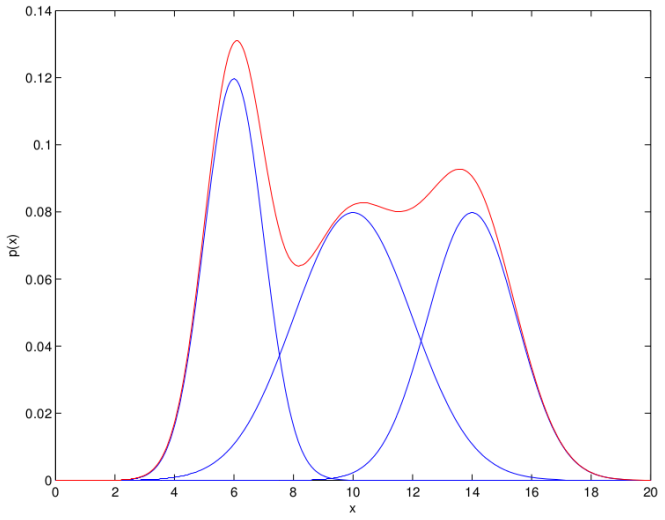


FIGURE 1.8: Example of a mixed Gaussian distribution

The advantages of using GMMs include computational efficiency and the flexibility of modeling arbitrary probability densities.

Hidden Markov Model

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved states. In a hidden Markov model the states are hidden and can not be observed explicitly. However, the

observation $O = \{o_1, o_2, ..., o_T\}$ has some stochastic relationship with the state sequence $S = \{s_1, s_2, ..., s_T\}$.

An HMM can be considered a generalization of a mixture model. The elements of an HMM are specification of two model parameters, N (number of states in the model) and M (number of different observation symbols per state), specification of the observations, and specification of the three sets of probability measures A (state-transition probability distribution), B (observation symbol probability distribution), and π (initial state distribution). For convenience, the compact notation is $\lambda = (A, B, \pi)$. The decision is made based on the maximum likelihood (ML) criterion.

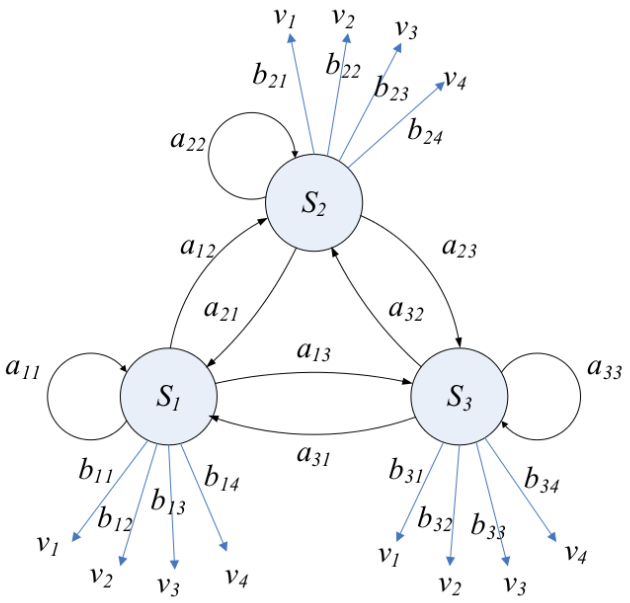


FIGURE 1.9: Example of a 3 states HMM

HMM is widely used in speech recognition where each word can be modeled by a distinct HMM, and in highlight detection where some complex highlights can be considered as the specific transition between certain keywords.

Support Vector Machine

A support vector machine (SVM) model represents the input data as points in space in order to classify them into different categories. Mapping of input data is done in order to divide categories by a clear gap being as wide as possible during the training stage. In the test phase, data is mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

An SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space used for classification. Best results are achieved by the hyperplane that has the largest distance to the nearest training data point of any class, since in general this approach minimizes the error of the classifier.

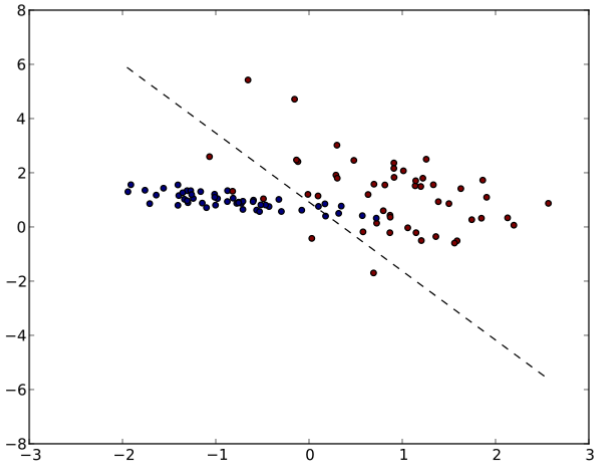


FIGURE 1.10: Example of an SVM

Although mapping may be done into a finite dimensional space, sometimes the sets to discriminate are not linearly separable. As a result, the original finite-dimensional space can be mapped into a much higher-dimensional one in order to make the separation easier. This is achieved by the use of a kernel function selected to suit the problem.

A special property of SVM is that they simultaneously minimize the empirical classification error and maximize the geometric margin, being also known as maximum margin classifiers. However, the SVM is only directly applicable for two-class discrimination. Multi-class SVM is proposed to overcome this limitation.

Neural Networks

A neural network (NN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. NN are usually presented as systems of interconnected *neurons* that can compute values from inputs by feeding information through the network.

In an NN a set of input neurons may be activated by a feature vector of an audio section. The activations of these neurons are then passed on, weighted and transformed by some function to other neurons, until finally an output neuron that performs decision is activated.

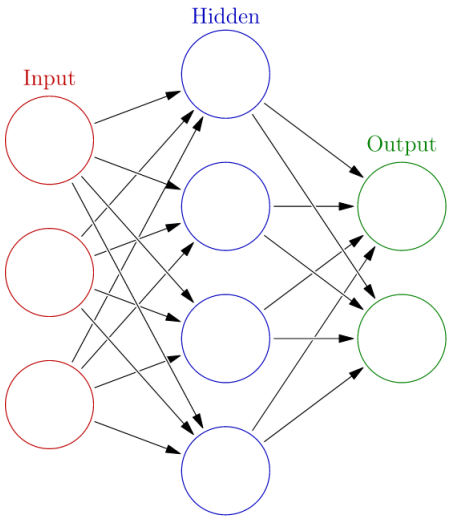


FIGURE 1.11: Example of an NN

The advantage of NN is the ability to learn to perform operations, not only for inputs exactly like the training data, but also for new data that may be incomplete or noisy. NN has also the benefit of easy modification by training with an updated data set.

1.5 Block Diagram

The input audio signal is extracted from the video file and converted into mono by summing both channels. By the use of different techniques various features are retrieved and grouped together into a feature vector for the classifier stage.

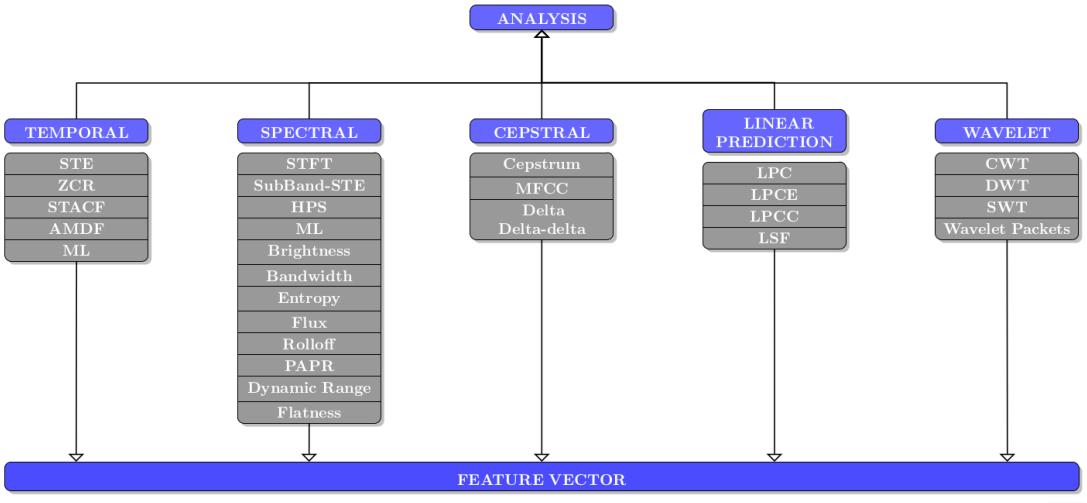


FIGURE 1.12: Extracted features

Finally, frame rate can be adjusted depending on the video file by simply modifying the window length in order to match audio features to video content.

Chapter 2

Time domain analysis

Since audio signals are naturally non-stationary, most of the signal processing tools would not be of practical use as these assume time invariance, i.e., a stationary signal. Hence these tools are not directly applicable for audio processing. In addition, computation of these tools for the whole signal lacks little practical significance due to the audio time varying characteristics. Therefore a short-time analysis is performed through the whole audio signal in order to determine how all these properties vary respect to time.

This is achieved by processing short audio segments where mentioned tools can assume the signal to be stationary. Commonly, blocks of 10-30 msec conform to the previous stationarity hypothesis for audio signals. This leads to the basic principle of short-time analysis, which is represented by a time-shifted window whose purpose is to select a segment of the sequence $x[m]$ in the neighborhood of sample $m = n$.

Many different types of windows can be used for short-time analysis. Each of them have advantages and drawbacks that make them suitable for different applications. The rectangular window is the simplest, equivalent to replacing all but N values of a data sequence by zeros

$$w[n] = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Since the rectangular window spectrum has a very narrow main lobe, it is useful to discern between closer frequencies with similar amplitude. However, side lobes leakage is considerably high, resulting in a poor amplitude dynamic range. Other windows are designed to moderate abrupt changes at the ends of the function as undesirable effects appear in the frequency domain due to these discontinuities. One alternative are the generalized Hamming windows which are of the form

$$w[n] = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right) \tag{2.2}$$

A Hanning window is obtained if values $\alpha = 0.5$ and $\beta = 0.5$ are selected. In this case, the ends of the cosine just touch zero, so the side-lobes roll off at about 18 dB per octave. In case we slightly vary previous parameters to $\alpha = 0.54$ and $\beta = 0.46$ a Hamming window is obtained instead. This is optimized to minimize the nearest side lobe giving it a height of about one-fifth that of the Hanning window.

As can be inferred from figure 2.1, windowed analysis involves a trade-off between resolving amplitude and frequency. For example, it can be shown that a $(2M+1)$ sample Hamming window has a frequency main lobe bandwidth of $4\pi/M$. Other windows will have similar properties, i.e., they will be concentrated in time and frequency, and frequency lobes width will be inversely proportional to the window length.

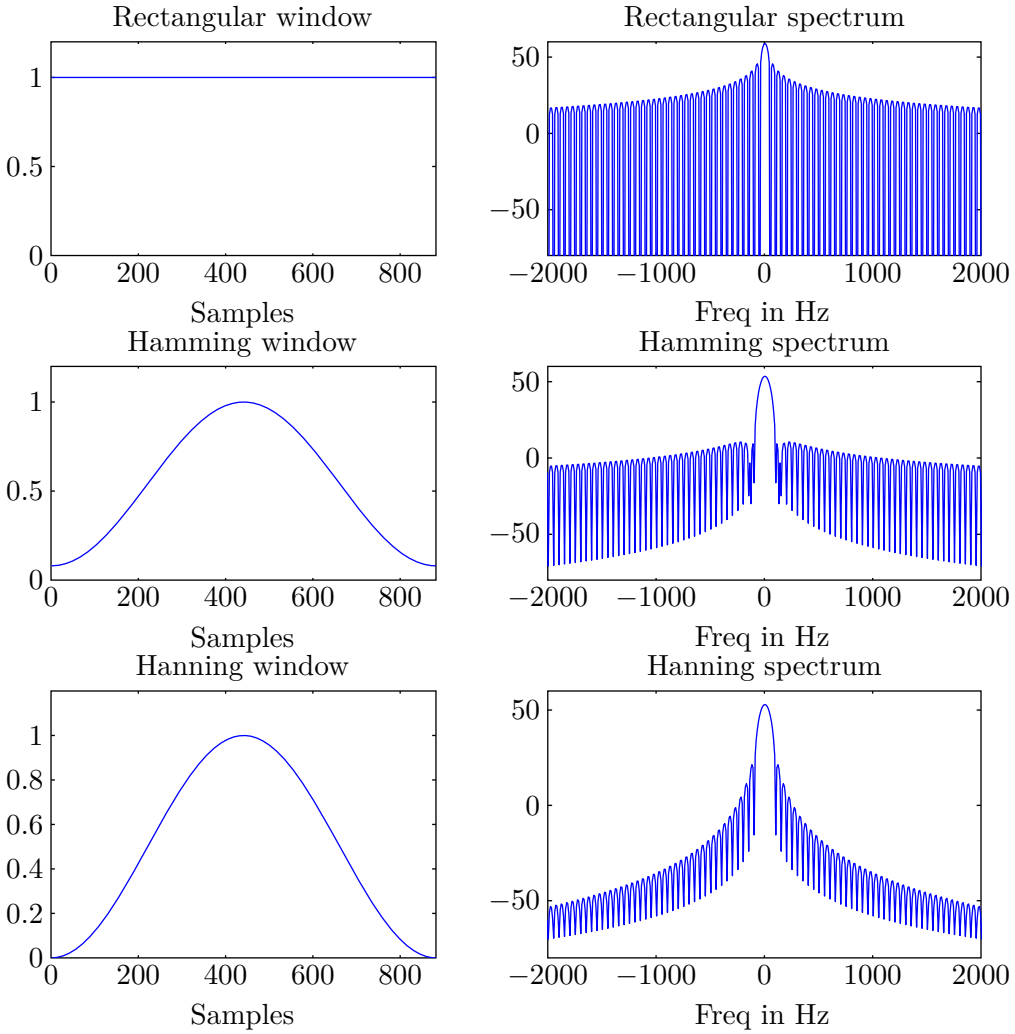


FIGURE 2.1: Time and spectral shape (dB) of three different windows

At this point, we have defined $w[n]$ as the particular type of window used for short-time analysis. Since windows will shift in time according to the segment of signal being treated, it will frequently be notated as $w[n-m]$. Initially, this involves a window shift every each sample, which is not a very efficient processing technique. As short-time analysis normally involves low-pass linear filtering due to windowing, audio features will vary slowly compared to the audio signal under analysis, and therefore larger increments can be used. In any case, this notation will be used in the following sections for simplicity, although a larger increment will be used, usually half the size of the window (50% overlap between frames).

As it is shown in figure 2.1, non-rectangular windows assign weights to the windowed signal samples. Normally, the more distant the sample is from the center of the window, the smaller the weight is. Therefore, overlapping is required to mitigate attenuation at the edges of the window while avoiding abrupt discontinuities. In addition, the FFT implementation assumes signal periodicity in a frame, and if the sequence is not periodic, a discontinuity between frames will exist, resulting in high frequencies in the transformed signal. Hence, windowing is used to smooth the signal between two consecutive frames. However, a higher overlap involves a smoother version of the time varying audio features, which yields a less accurate method for detecting event transitions.

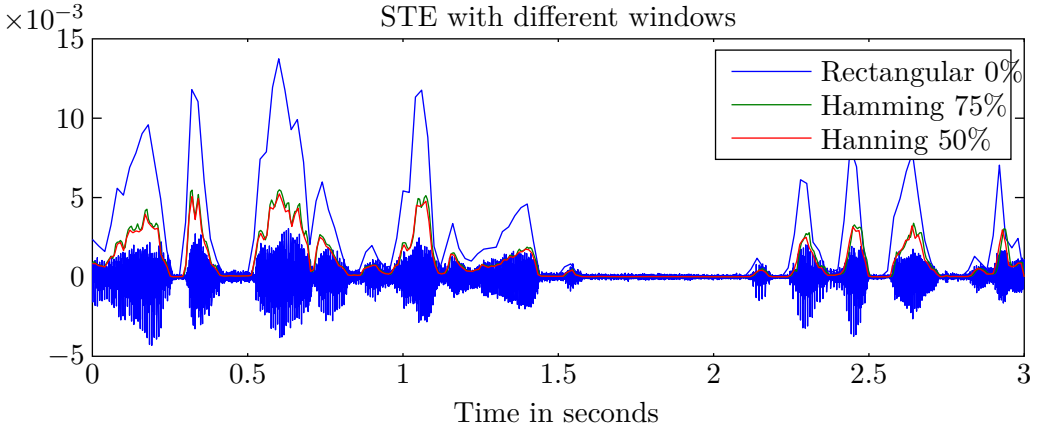


FIGURE 2.2: Short-time energy window and overlap effect

The main advantage of temporal features lies in their direct processing without a time-to-frequency conversion. Despite the ease at computation that time domain might imply, human hearing is generally based on the frequency properties of the signal. Therefore, time domain analysis is frequently insufficient for obtaining the required information of the audio signal. Nevertheless, some features are very consistent in some applications such as pitch estimation.

So far, the role of short-time analysis has been discussed for the audio signal in general terms. Speech signals are crucial for the current environment because of the almost

continuous presence of a commentator in the audio file. In the case of speech, the stationarity assumption is valid for 10 to 30 msec, making the previously chosen 20 msec a consistent decision. Furthermore, and according to 1.5 the impulse response of the linear system modeling the vocal tract is assumed to be slowly-time-varying, changing every 50–100 msec, i.e., over the timescale of phonemes, impulse response, frequency response, and system function of the vocal tract remains relatively constant. Hence a 20 msec window preserves stationarity.

As a summary it can be stated that short-time analysis entails ease at computation for the retrieval of audio and speech features through the use of the adequate window function. Most important time domain audio features are detailed below.

2.1 Short-time energy

The short-time energy function is an indicator of the signal amplitude within the interval under analysis. It is defined as follows:

$$E_n = \frac{1}{N} \sum_{m=-\infty}^{\infty} |x[m]w[n-m]|^2 \tag{2.3}$$

$$w[n] = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where $x[m]$ is the input signal, $w[n]$ the rectangular window function performing the short time analysis and N the window length in samples.

Thus, an averaged measure of the audio signal level is performed on each window. This feature provides a simple computable tool for determining loudness and volume, and as a result can be used to discriminate between high and low energy regions, such as cheering and silence. However, it does not perform well under noisy environments.

Furthermore, by the nature of speech production the speech signal consists of voiced, unvoiced and silence regions, and the energy associated with voiced regions is large compared to that of unvoiced and silences. Hence, short-time energy seems to be an appropriate feature to accurately differentiate each of them.

Figure 2.3 shows the STE obtained by a Hamming window with 50 % overlap time analysis. A noticeable difference exists between the voiced, unvoiced and silence regions energy. Voiced sounds are represented by the highest peaks, unvoiced sounds correspond to smaller peaks and silences to empty energy sections.

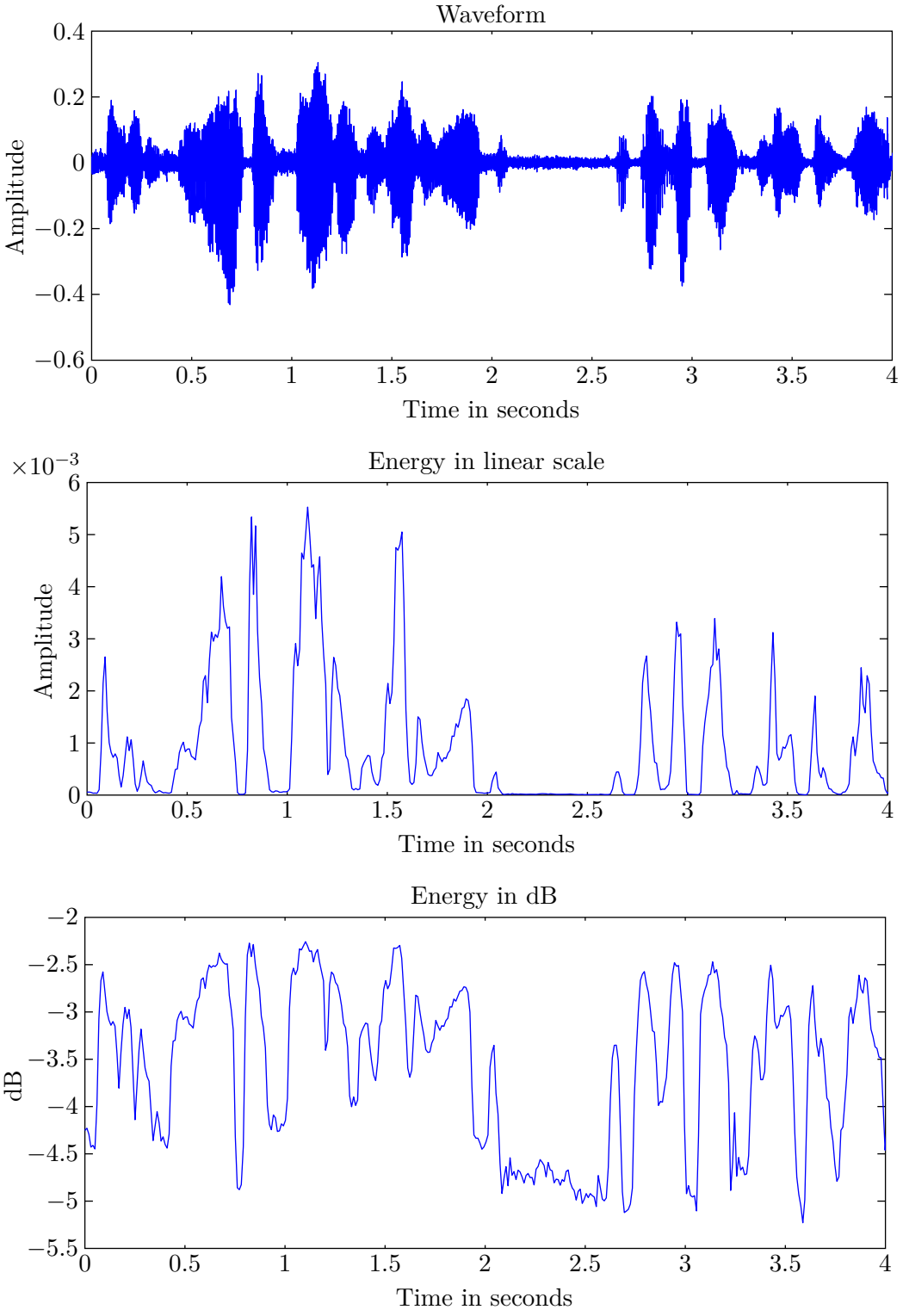


FIGURE 2.3: Short-time energy of a speech segment

Moreover, short-time energy is considered to be a useful feature for highlight detection due to the nature of sound. An excited audience or cheering audio section after a goal involve more energy than simple audience background noise, as it is shown in figure 2.4. Cheering takes place in the middle part of the waveform, where logarithmic energy

reaches a constant maximum between 1.7 secs and 2.2 secs, in contrast with voiced sounds, which show less than a 0.25 secs duration peak.

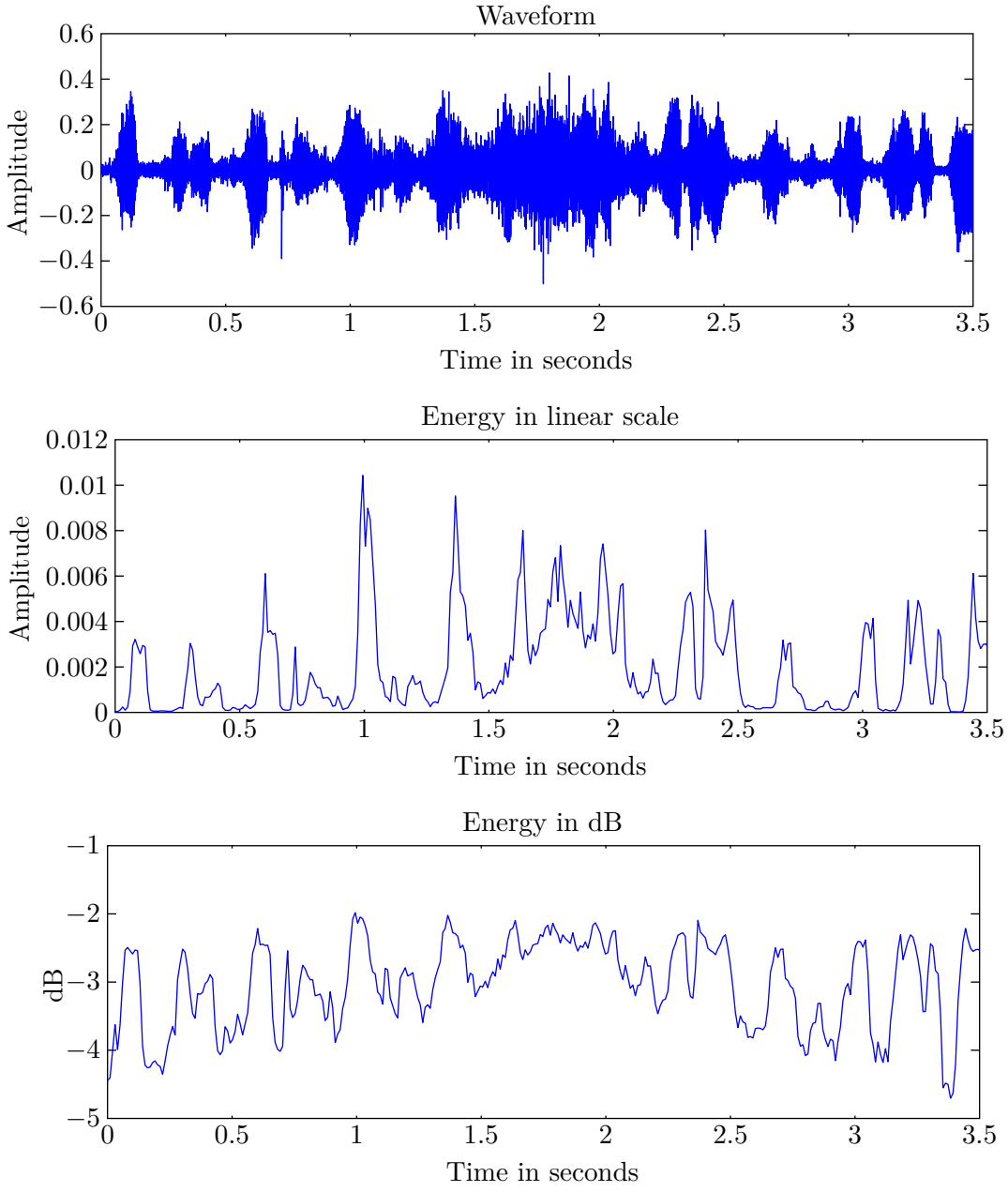


FIGURE 2.4: Cheering energy

It is often possible to express short-time analysis operators as a convolution or linear filtering operation.

$$E_n = \frac{1}{N}\sum_{m=-\infty}^{\infty}(x[m]w[n-m])^2 = \frac{1}{N}\sum_{m=-\infty}^{\infty}x^2[m]w^2[n-m] = x^2[n] * h_e[n] \qquad (2.5)$$

where $h_e[n] = w^2[n]$ is the impulse response of the low-pass linear filter.

Hence, short-time energy is the output of a low-pass filter whose frequency response is the Fourier transform of the window. Consequently, short-energy is slowly varying compared to the time variations of the speech signal, and therefore, it can be sampled at a much lower rate than that of the original speech signal. Usually it is about half the frame size, which means 50% overlap between frames.

In addition, further energy representations exist besides the one defined before. Root-mean-square energy (RMS) is computed by taking the root average of the square of the amplitude, i.e.

$$RMS_{energy} = \sqrt{\frac{1}{N}\sum_{m=-\infty}^{\infty} |x[m]w[n-m]|^2} = \sqrt{E_n} \tag{2.6}$$

RMS energy can be used to get an assessment of the temporal distribution of energy, in order to see if it remains constant throughout the signal or if some frames are more contrastive than others. This produces a measure called low energy rate, which estimates the percentage of frames showing less energy than the average.

As a summary, short-time energy provides a method to retrieve valuable information from the audio signal at a very low computational cost. Besides, some basic post-processing techniques of this feature yield new efficient measures, such as the previously mentioned RMS energy and low energy rate.

2.2 Zero-crossing rate

Short-time zero-crossing rate is defined as the weighted average of the number of times the signal changes sign within the time analysis window. Representing this operator in terms of linear filtering leads to

$$Z_n = \sum_{m=-\infty}^{\infty} \frac{1}{2}|sgn\{x[m]\} - sgn\{x[m-1]\}|w[n-m] \tag{2.7}$$

where $w[n-m]$ is the shifted window function and

$$sgn\{x[m]\} = \begin{cases} 1 & x[m] \geq 0 \\ 0 & x[m] < 0 \end{cases} \tag{2.8}$$

Zero-crossing takes place in the time domain when a signal changes its amplitude sign, i.e., when two adjacent samples have different sign. In 2.7, $|sgn\{x[m]\} - sgn\{x[m-1]\}|$

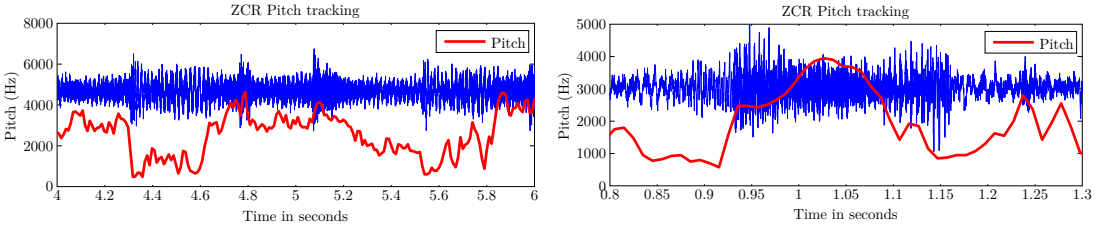
is equal to 2 if $x[m]$ and $x[m-1]$ have different sign, and 0 if they have the same. By multiplying 2.7 by 1/2, we obtain a measure of the zero-crossing amount in a particular windowed segment of the signal.

As zero-crossing rate gives information about the number of zero-crossings present in a given signal, it is intuitively possible to assert that if the number of zero crossings is high, then the signal is changing rapidly and accordingly it may contain high frequency information. On the other hand, if the number of zero crossing is low, the signal is changing slowly and accordingly it may contain low frequency information. Thus short-time zero-crossing rate is a crude frequency analyzer. Since this rate is measured in *crosses/window*, it is possible to calculate pitch easily from it, taking into account a 1/2 factor, as there will be two zero-crossings per cycle of a 1Hz signal.

$$Pitch(Hz) = Z_n \left[\frac{crosses}{window}\right] \times \frac{1}{N}\left[\frac{window}{samples}\right] \times Fs\left[\frac{samples}{sec}\right] \times \frac{1}{2}\left[\frac{Hz}{crosses/sec}\right] \quad (2.9)$$

where N is the window length and Fs the sampling frequency.

Therefore, zero-crossing rate provides a very easy to compute algorithm to estimate signal pitch with no further Fourier analysis involved. This approach might be useful in some particular situations due to the nature of certain audible sounds as it can help to classify keywords. In figure 2.5 pitch tracking of a 2 seconds music section and a whistle sound from the referee are shown.



(A) Music (B) Whistle

FIGURE 2.5: Pitch estimation of two different audio signals

Music shows high estimated pitch (between 2KHz and 4KHz) when there is an absence of low frequencies, with higher peaks corresponding to the beginning of musical sounds. On the right figure, whistle shows a high pitch estimation almost during its whole duration.

In speech, voiced signals happen to have a high frequency falloff due to the low-pass nature of the glottal pulses, whereas unvoiced sounds have much more high frequency energy since they are modeled as random noise. This feature can be an appropriate tool to classify speech segments into the proper category, as shown in figure 2.6.

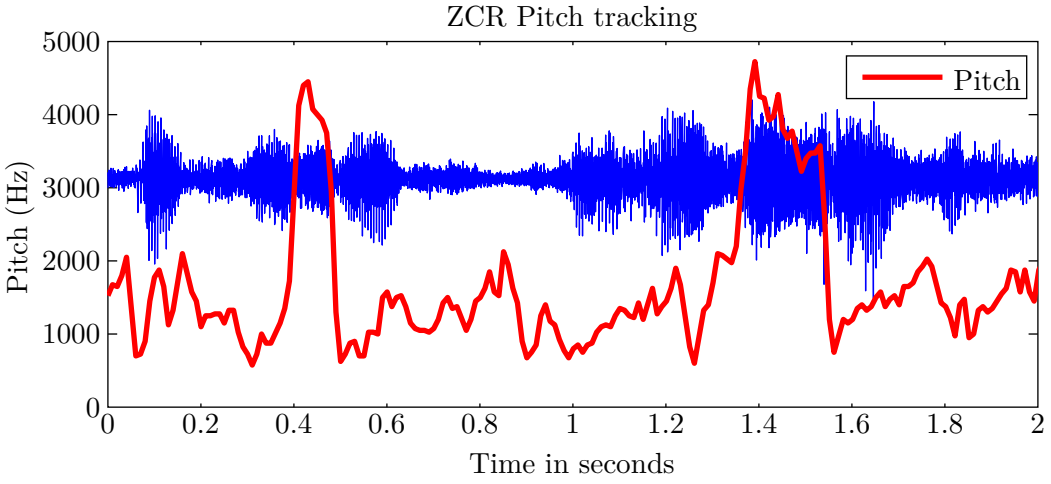


FIGURE 2.6: Pitch estimation of a speech section

The two high pitch peaks represent the occurrence of the unvoiced sounds 'S' and 'SH' respectively, while low pitches point the presence of quasi-periodic voiced sounds and silences.

As it was previously stated, as a consequence of the windowed analysis and the window low-pass nature, zero-crossing rate function is slowly varying compared to the speech signal as well.

In summary, zero-crossing rate provides an algorithm for pitch estimation at a low computational cost. Nevertheless, it loses precision when high frequency harmonics are present in the signal as shown in figure 2.7, causing the estimated pitch to shift in frequency depending on harmonics energy.

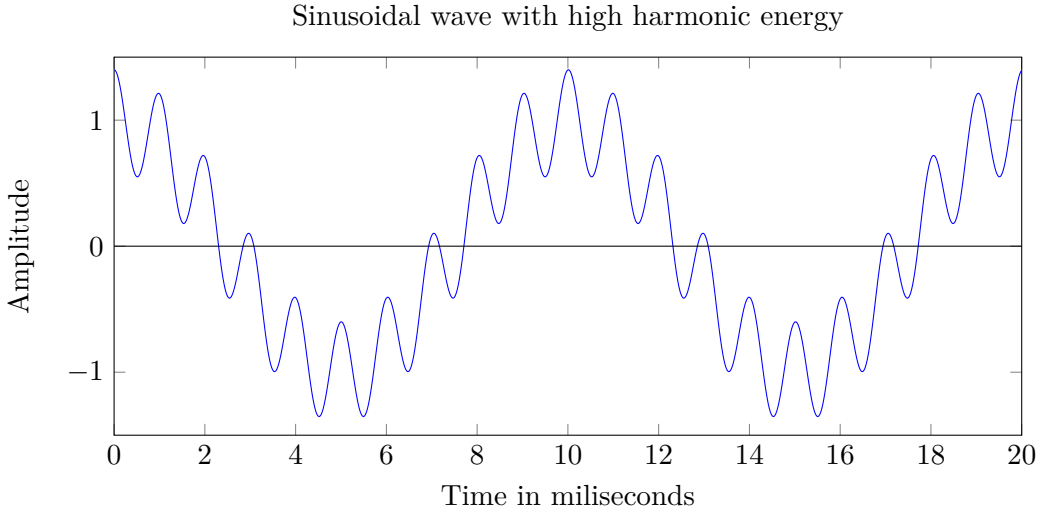


FIGURE 2.7: Sinusoidal signal with high harmonic energy

In this particular case, the pitch of a $100Hz$ sine wave is estimated as $300Hz$ since for each $100Hz$ cross high frequency components involve 2 additional crosses.

Hence, analysis of highly noisy or harmonic signals has poor results. However, there are many cases in which zero-crossing can be a useful measure, e.g. some speech applications where a single source is assumed.

2.3 Short-time autocorrelation

The autocorrelation of a stationary sequence $x[m]$ is the cross-correlation between such signal and itself and is expressed as

$$\phi_x[l] = \sum_{m=-\infty}^{\infty} x[m]x[m+l] \tag{2.10}$$

The interest of autocorrelation lies in observing how similar the signal characteristics are with respect to time. This is achieved by providing different values to the discrete-lag index l for sequence shifting, and comparing the signal to those shifted versions. It is a mathematical tool for finding repeating patterns, such as the existence of periodicity covered by noise, or for identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. Autocorrelation is also the basis for many spectrum analysis methods.

Periodic sequences have a peak in autocorrelation when the time lag l is equal to the fundamental period of the signal or an integer multiple. This fact yields a very accurate method for pitch estimation.

Short-time autocorrelation is defined as the deterministic autocorrelation function of the sequence $x_w[m] = x[m]w[n-m]$ that is selected by the shifted window, i.e.,

$$\phi_x[l] = \sum_{m=-\infty}^{\infty} x_w[m]x_w[m+l] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]x[m+l]w[n-(m+l)] \tag{2.11}$$

If we consider $w[n]$ to be a rectangular window in $0 \leq n \leq N-1$, and take into account the even-symmetric autocorrelation property $\phi_x[-l] = \phi_x[l]$ then 2.11 can be rewritten in terms of linear filtering as

$$\phi_x[l] = \sum_{m=-\infty}^{\infty} x[m]x[m-l]w_l[n-m] \tag{2.12}$$

where $w_l[n-m] = w[n-m]w[n-(m-l)]$. Equation 2.12 is a convolution sum within the region set by $w_l[n-m]$, which yields a tapered off version of $\phi_x[l]$ due to the window effect.

Since short-time autocorrelation depends on two parameters, w and l, pitch can be estimated on each window for selected lag values. The minimum frame size for computing short term autocorrelation should include at least two cycles of a periodic signal. For a 50 Hz signal this corresponds to a 40 msec window length, which leads to 20 msec minimum lag for detecting periodicity.

According to the *source/system* model, speech signals can be represented as $s[n] = e[n] * h[n]$, where $e[n]$ is the excitation to the linear system with impulse response $h[n]$. Considering autocorrelation properties, previous model will satisfy

$$\phi^s[l] = \phi^e[l] * \phi^h[l] \tag{2.13}$$

In the case of voiced speech, excitation is assumed to be a periodic impulse train with pitch period P_0. Therefore, autocorrelation is the periodic linear system autocorrelation function:

$$\phi^s[l] = \sum_{m=-\infty}^{\infty} \delta[l - mP_0] * \phi^h[l] = \sum_{m=-\infty}^{\infty} \phi^h[l - mP_0] \tag{2.14}$$

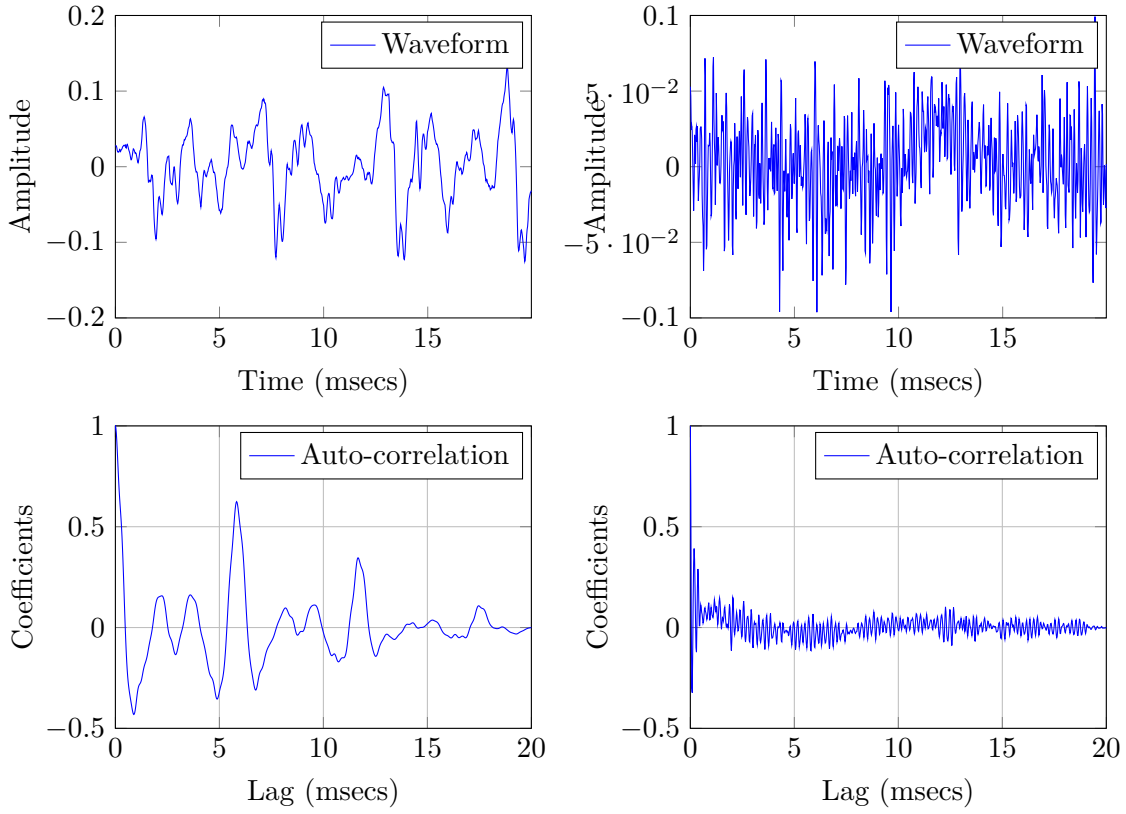
In the case of unvoiced speech, excitation is modeled as random white noise, whose stochastic autocorrelation function would be an impulse sequence at $l = 0$. Thus, the autocorrelation function of unvoiced speech computed using averaging would be

$$\phi^s[l] = \delta[l] * \phi^h[l] = \phi^h[l] \tag{2.15}$$

Hence, autocorrelation of voiced sounds shows peaks with period P_0, whereas autocorrelation of unvoiced sounds do not comply with this. This suggests that short-time autocorrelation could be the basis for an algorithm to discriminate voiced and unvoiced segments, as well as voiced pitch estimation.

Although this is considered for infinite periodic signals, the deterministic autocorrelation function of a finite-length segment of the speech waveform have similar properties. However, correlation values will taper off with l due to the tapering of the window and the effect of less and less data involved in the computation of the short-time autocorrelation for longer l values.

Autocorrelation methods need at least two pitch periods to detect pitch. For instance, a fundamental frequency of $100Hz$, needs 20 ms of the speech signal to be analyzed.



(A) Voiced sound (B) Unvoiced sound

FIGURE 2.8: STACF of a voiced/unvoiced sound

Maxlag determines the lowest detectable frequency since STACF range extends from *-maxlag* to *+maxlag*, and a peak in *+maxlag* corresponds to that minimum detectable frequency. Similarly, peaks corresponding to high frequencies will be close to the origin, and as a result a maximum detectable frequency must be specified as well. In our case this is $500Hz$ and it is performed by analyzing the STACF from a minimum lag value. For $500Hz$, it corresponds to $Fs/500$ samples.

Figure 2.9 shows a 4 seconds speech extract. Low pitches that are almost time-constant and clearly defined correspond to voiced speech while higher randomly varying pitches correspond to silence and unvoiced sounds.

Autocorrelation pitch detection algorithm is relatively robust to noise but it is highly sensitive to sampling rate. Since fundamental frequency (pitch) is calculated directly from a shift in samples it follows that if a lower sampling rate is used, lower resolution in pitch is obtained as well.

Furthermore, short-time autocorrelation implicitly contains the short-time energy:

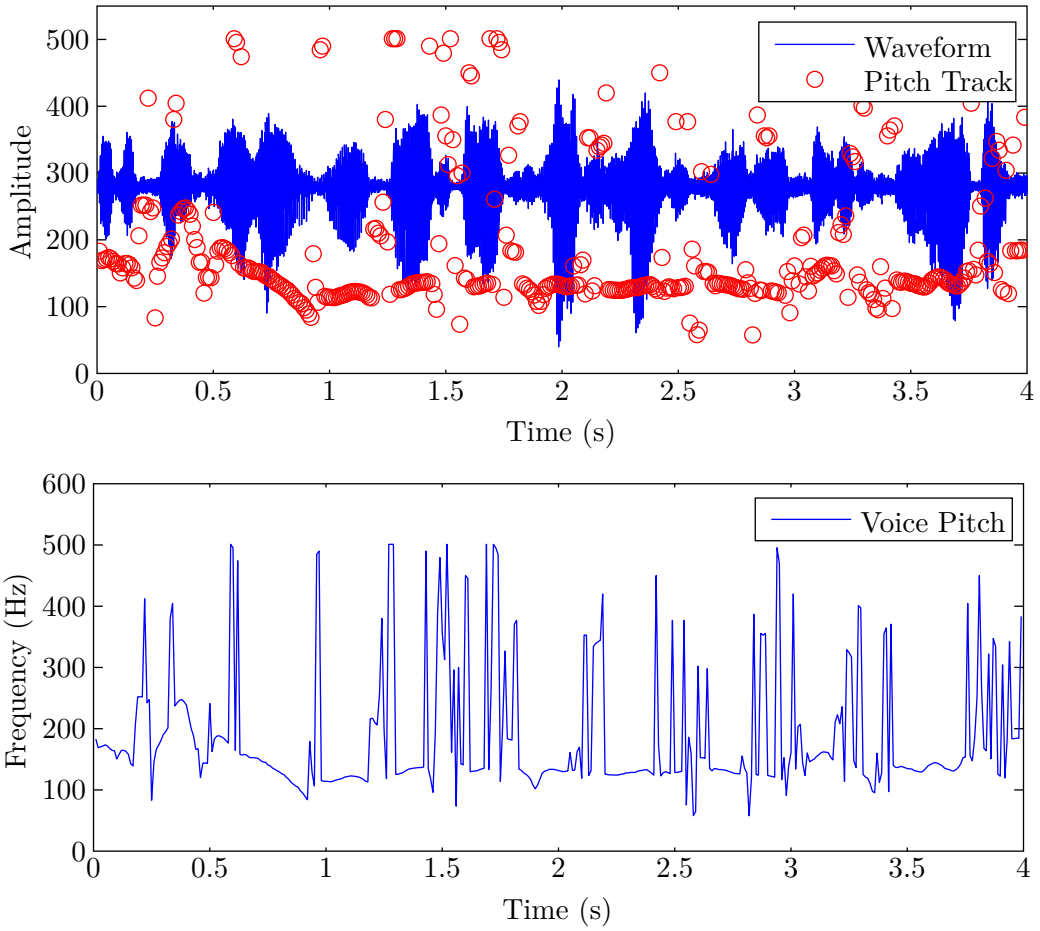


FIGURE 2.9: Pitch estimation by STACF

$$E_n = \sum_{m=-\infty}^{\infty} (x[m]w[n-m])^2 = \phi_n[0] \tag{2.16}$$

2.4 Average Magnitude Difference Function

The average magnitude difference function (AMDF) was proposed in [3] and is defined as

$$D(\tau) = \frac{1}{N-\tau-1} \sum_{n=0}^{N-\tau-1} |x(n) - x(n+\tau)| \tag{2.17}$$

where $x(n)$ is the audio sequence windowed by a window of length N, τ is the lag index and the constant term outside summation is for normalization. Analogously to the STACF, AMDF performs time shifting of the windowed signal. However, in AMDF a difference signal is formed by delaying the input τ samples, subtracting the delayed waveform from the original, and summing the magnitude of the differences between sample values. Thus, AMDF carries out the magnitude of the difference instead of

correlating the input at various delays with multiplications and summations. For a periodic or quasi-periodic signal, $D(\tau)$ should show minimum at the τ corresponding to the signal period, T_P, and minimum peaks with lower degree at integer multiples. In general, a rough estimation of pitch is derived by

$$T_p = argmin\{\tau\} \qquad \tau_{min} \leq \tau \leq \tau_{max} \tag{2.18}$$

where τ_{min} and τ_{max} correspond to possible minimum and maximum detectable pitch periods in samples. Similar to STAFC where lag values showed a peak at the signal fundamental period, on the AMDF the difference signal shows minimums at lags corresponding to the pitch period. Besides, AMDF is always zero at $\tau = 0$, since the signal is subtracted from itself, and therefore a reasonable τ_{min} must be chosen to properly estimate the minimum value corresponding to the pitch. Furthermore, the range of τ varies between 0 and $N - 1$, and the minimum detectable pitch depends on the window length N. On the calculation of $D(\tau)$ less data is involved at higher lags as a consequence of windowing and thereby, AMDF cannot show periodic nature at the later half of a frame. Therefore, a reasonable value of τ_{max} must be chosen as well.

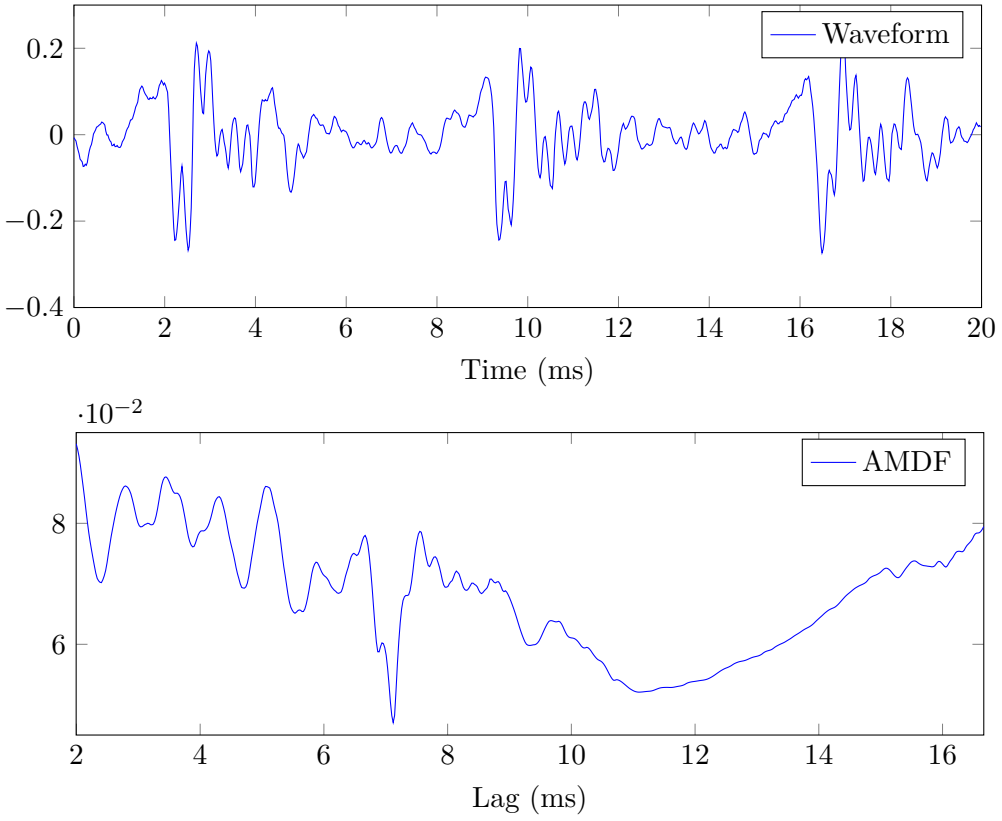


FIGURE 2.10: AMDF $D(\tau)$ estimation

Figure 2.10 illustrates pitch estimation in one window of a voiced quasi-periodic sound. Waveform frame shows peaks at $2.2ms$, $9.3ms$ and $16.4ms$ and $D(\tau)$ reflects this $7.1ms$ periodic structure with a function minimum. One important consideration to take into account is not to taper frame under analysis as it would result in tapering of $D(\tau)$ and thereby in an erroneous minimum estimation.

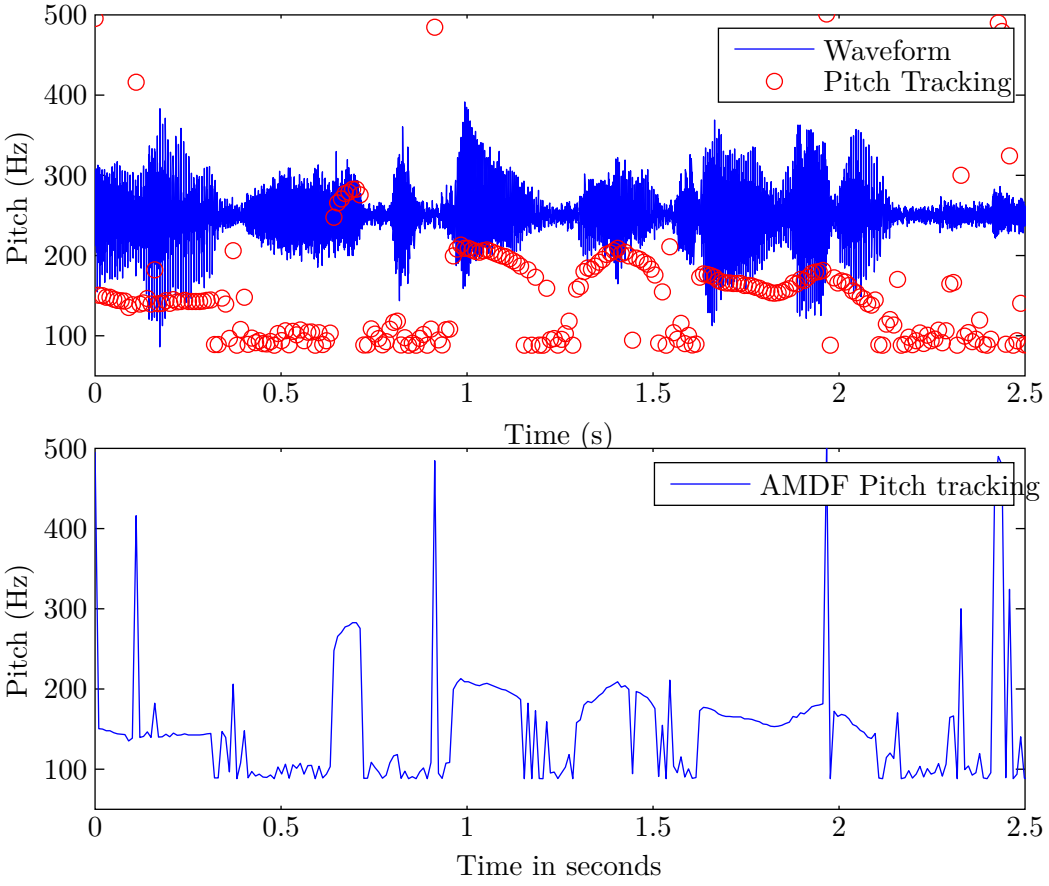


FIGURE 2.11: Pitch tracking with AMDF

As well as with other pitch tracking methods, AMDF shows almost constant pitch detection for voiced periodic sounds, approximately between 150Hz and 300Hz, whereas other unclear estimations correspond to unvoiced and silences.

As it occurs with other time-domain approaches AMDF has some shortcomings, especially in noisy environments where pitch estimation might be wrong, estimating half or double the pitch. Besides, AMDF does not deal well with polyphonic sounds since no spectral analysis is performed. In the case of speech, the limiting factor on pitch estimation accuracy is the inability to completely separate the fine structure of the excitation from the effects of the spectral envelope. This is achieved by other approaches, such as cepstral analysis and linear prediction, discussed in further chapters.

Nevertheless, AMDF has a clear advantage compared to STAFC as calculations do not require multiplications, which improves performance and makes it more suitable for

real-time applications. Another advantage of this method is that the relative sizes of the nulls tend to remain constant as a function of delay.

2.5 Maximum Likelihood

The Maximum Likelihood algorithm in the time-domain attempts to find the most likely value for pitch depending on the segment length [4]. In order to achieve this, the signal is broken into segments and these are coherently summed according to

$$r(t,\tau) = \begin{cases} \dfrac{1}{N+1}\displaystyle\sum_{n=0}^{N} r(t+n\tau) & 0 \le t \le b \\ \\ \dfrac{1}{N}\displaystyle\sum_{n=0}^{N} r(t+n\tau) & b \le t \le \tau \end{cases} \tag{2.19}$$

where N is the amount of segments, τ is the length of each segment and b is the length of the last one. Thus, the signal is split into several sections that are summed to each other. Finally, an average is carried out depending on the amount of segments computed for each sample.

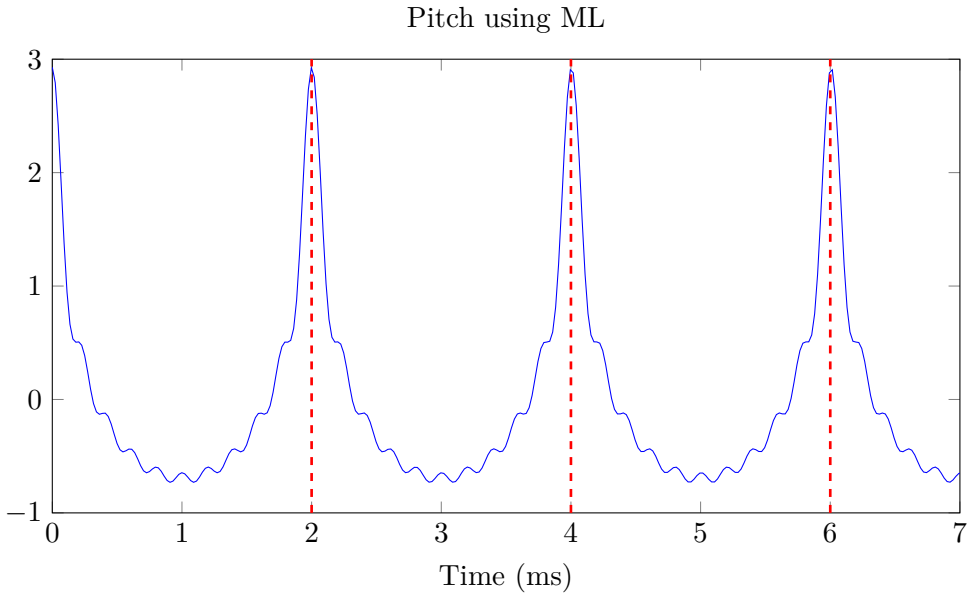


FIGURE 2.12: ML Pitch estimation process

The key idea behind this algorithm is that these segments will add coherently when τ reaches the fundamental period of the signal. Consequently, the sum with highest energy is most likely to be the one which determines what the pitch is. However, since

segments also add coherently with integer multiples of τ, pitch will be the one related to the smallest τ. We can define the objective function as

$$J(\tau) = (N+1)\sum_{t=0}^{b} r^2(t,\tau) + N\sum_{t=b+1}^{\tau} r^2(t,\tau) \tag{2.20}$$

which calculates the energy on the summed segment related to τ. The τ-segment with highest energy will be the one where summations where performed coherently, and therefore the one related to the pitch. On a last step, looking for the lowest local maximum will provide the pitch of the signal.

In figure 2.12, pitch of a 500 Hz periodic sine with high harmonic energy is to be estimated by the ML algorithm. Red dashed lines represent the separation between frames τ, which determines the value of $J(\tau)$ on figure 2.13. That is, for $\tau = 2ms$, $J(\tau)$ will show a maximum.

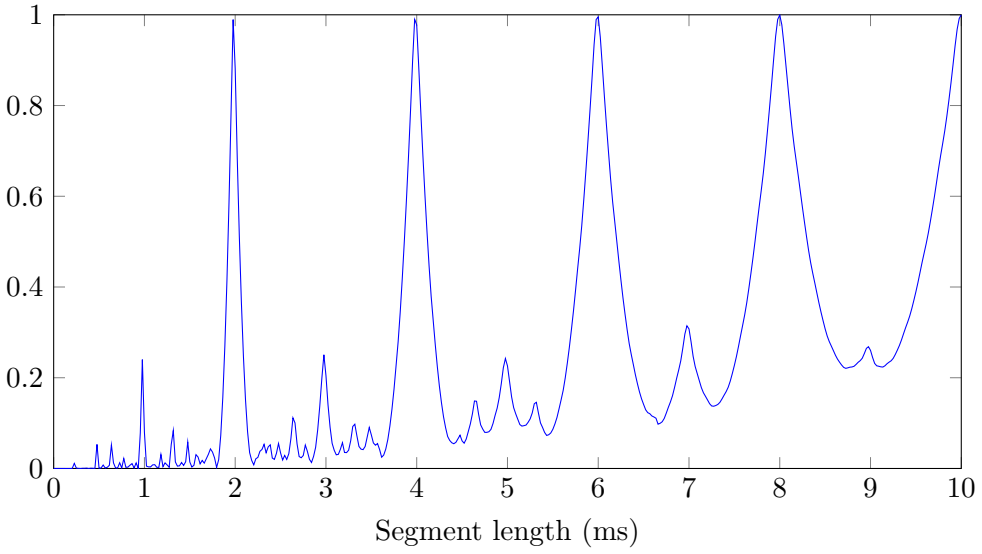


FIGURE 2.13: $J(\tau)$ represents likelihood of the summed signal depending on the signal length segmentation

The rest of local maxima below 2ms depict harmonics energy, e.g., $1ms$ peak relates to $1KHz$ harmonic. To conclude, it is remarkable to point the periodic structure of $J(\tau)$ due to the coherent sum at integer multiples of τ.

Chapter 3

Frequency domain analysis

As it was presented in the human hearing section earlier, models for auditory processing are based on a filterbank because of the basilar membrane functioning. Thereby it seems reasonable to frame much of our knowledge of audio perception in terms of frequency analysis. For instance, pitch is subjectively interpreted from periodic audio signals by humans and it is closely related to main frequency and harmonic content. This relationship between pitch and frequency along with Fourier representation generates some advantages with respect to time-domain algorithms, e.g., polyphonic detection is possible.

Efficient representations of the signal can be obtained through frequency analysis, providing valuable and simple interpretations. The following section details diverse frequency domain tools for feature extraction and introduces other processing methods based on it.

3.1 Short-time Fourier Transform

Short-time Fourier transform is a linear transformation used to determine frequency information and phase content of windowed segments on a signal as it changes over time. It is frequently used to analyze audio and music signals and is the basis for a wide range of speech analysis, coding and synthesis systems. In the case of digital processing, discrete-time STFT (DTFT) is defined as follows

$$X_n\left(e^{j\omega}\right) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m} \tag{3.1}$$

Each segment is transformed into the frequency domain and the complex result is added to a matrix containing magnitude and phase for each point in time. Therefore, STFT is a function of two variables, where n represents the window index and ω the angular frequency. Moreover, since it is based on short-time analysis it is possible to express it in terms of a linear filtering operation. Equation 3.1 can be expressed as

$$X_n\left(e^{j\omega}\right) = (x[m]e^{-j\omega n}) * w[n] = (x[m] * (w[n]e^{j\omega n}))e^{-j\omega n} \tag{3.2}$$

Considering that rectangular and Hamming windows have low-pass frequency responses, previous expression indicates that for a fixed value of ω, $X_n\left(e^{j\omega}\right)$ is slowly varying as n varies. In addition, these results yield two different interpretations of the STFT. If we consider $x[m]e^{-j\omega n}$ the input of a LTI system with impulse response $w[n]$, then $x[m]$ spectrum is shifted down by ω and the window low-pass filter selects the resulting band of frequencies. On the other hand, if the input of the LTI is $x[m]$ and the impulse response $w[n]e^{j\omega n}$, then the frequency response is a band-pass filter determined by the frequency shifted version of the initially low-pass $w[n]$, and finally $x[m]$ spectrum is filtered by this band-pass filter and shifted back to initial low-pass signal. Nonetheless, both results are equivalent.

STFT is a continuous function in the frequency domain and therefore needs a transformation so that it can be computed. Discrete Fourier Transform (DFT) is defined as

$$X[k] = \sum_{m=0}^{N-1} x[m]e^{-j\frac{2\pi k}{N}n} \qquad k = 0, 1, \ldots, N-1 \tag{3.3}$$

where k represents the k^{th} Fourier coefficient and N is the length of the periodic DFT in samples. In this project it is implemented by the widely known FFT algorithm. In the case of overlapping short-time analysis another parameter must be taken into account. If we name R to the overlapping step in samples, then it can be shown that

$$R \leq \frac{L}{2C} \tag{3.4}$$

where C is a constant dependent on the window frequency bandwidth. For a rectangular window C=1 and therefore a maximum 50 % overlapping is permitted. In the case of a Hamming window C=2, implying an overlap lower than 75 %. These results are related to sampling the STFT in time at a rate twice the window bandwidth in frequency with the purpose of avoiding frequency aliasing. Similarly, sampling in the frequency-domain must be performed at a rate of twice the equivalent time width of the window in order to avoid time-domain aliasing, setting the following constraint:

$$N \geq L \tag{3.5}$$

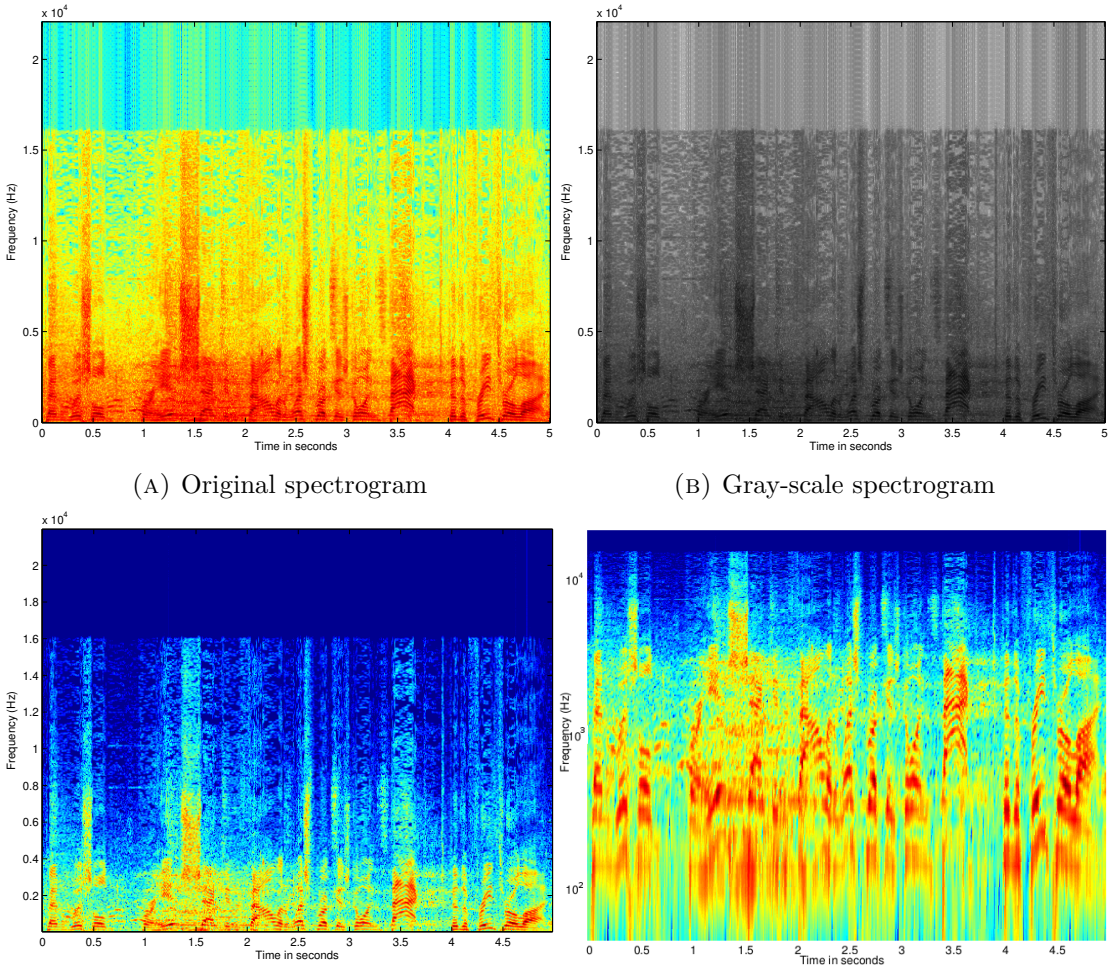
where N is the number of samples of the DFT and L the window length. Therefore larger window lengths will require more Fourier coefficients to correctly represent the signal, and as this previous length is fixed by the 10-30 msecs required for speech and audio stationarity, the amount of frequency coefficients is fixed as well. At least $L/2$ coefficients must be retrieved due to the spectrum symmetry. Sampling frequency has a direct impact on this result as well. If the sampling frequency is too high, the number of coefficients might reach a considerable dimension, producing poor results on the classifier (more features does not necessary involve more precision). Besides, basketball matches contain important high frequency information on the audible spectrum that might indicate the presence of important events, such as a whistle. This fact restricts the sampling frequency to values higher than 40 KHz, implying also a large number of Fourier coefficients. Hence, short-time Fourier Transform does not comply very well with the classifier requirements and is not directly used as an output for the feature vector. Nevertheless, it is the basis for many other frequency based methods and further approaches, such as cepstral analysis.

The magnitude squared of the STFT yields the spectrogram of the function:

$$Spectrogram(n,k) = |X(n,k)|^2 \tag{3.6}$$

Figure 3.1 shows 4 different spectrograms of a speech segment. The first one displays a raw unprocessed colored spectrogram on a linear scale with a Hamming window and 50% overlapping. The second one is a gray-scale pseudocolor spectrogram with interpolated shading. The third one has a threshold set under -35dB, so that a better discrimination in high energy levels can be made, specially useful for speech analysis. The last one is set in a logarithmic frequency scale so as to appreciate details better, since most energy is located on low/medium frequencies. As a different spectrogram example, figure 3.2 presents a musical section with high harmonic energy. A clear periodic harmonic structure can be observed during the entire segment

Another aspect must be taken into account when interpreting short-time spectrum: resolution. The length of the windowing function relates directly to how signal is represented, and consequently determines time and frequency resolution. Smaller window widths provide a better time resolution as shorter periods of time are processed. However, Fourier transforms use fewer samples for its calculation producing larger spectral leakage and a worse frequency resolution. Similarly, larger window widths provide more



(A) Original spectrogram

(B) Gray-scale spectrogram

(C) Clipped spectrogram

(D) Log scale of the clipped spectrogram

FIGURE 3.1: 4 different spectrograms of the same commentator's speech segment

samples computed on each transform, making time resolution worse and frequency resolution better. These approaches are called narrow-band and wide-band transforms respectively. Since audio signals commonly concentrate their energy at lower frequencies, and human hearing has a logarithmic amplitude perception, a more flexible analysis seems necessary. The discrete wavelet transform, discussed in the last chapter, solves this issue.

Furthermore, short-time autocorrelation can be expressed in terms of the STFT as follows

$$\phi_n[l] = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X_n\left(e^{j\omega}\right)|^2 e^{j\omega l} d\omega \tag{3.7}$$

Therefore, information provided by the STACF must be also given by the STFT, i.e., repetition patterns pointing to the presence of a periodic signal. If that is the case, the spectrum will have clearly defined harmonic components at integer multiples of the

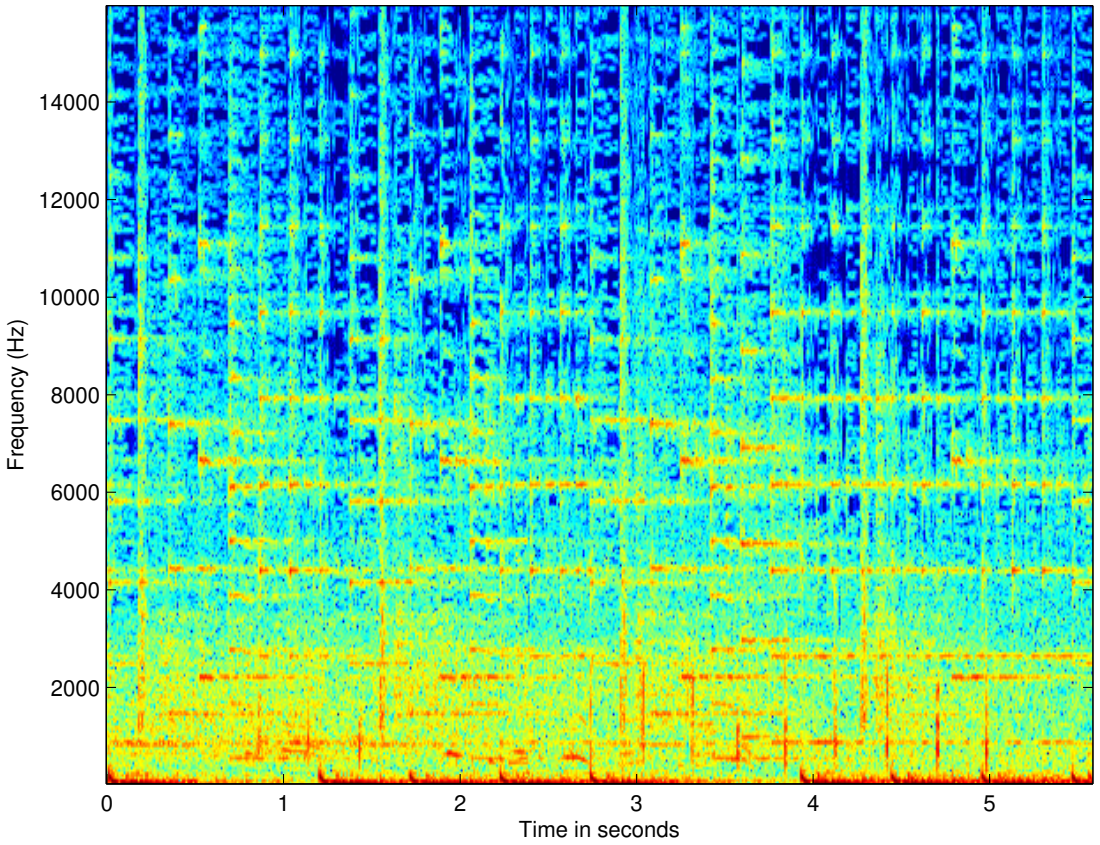
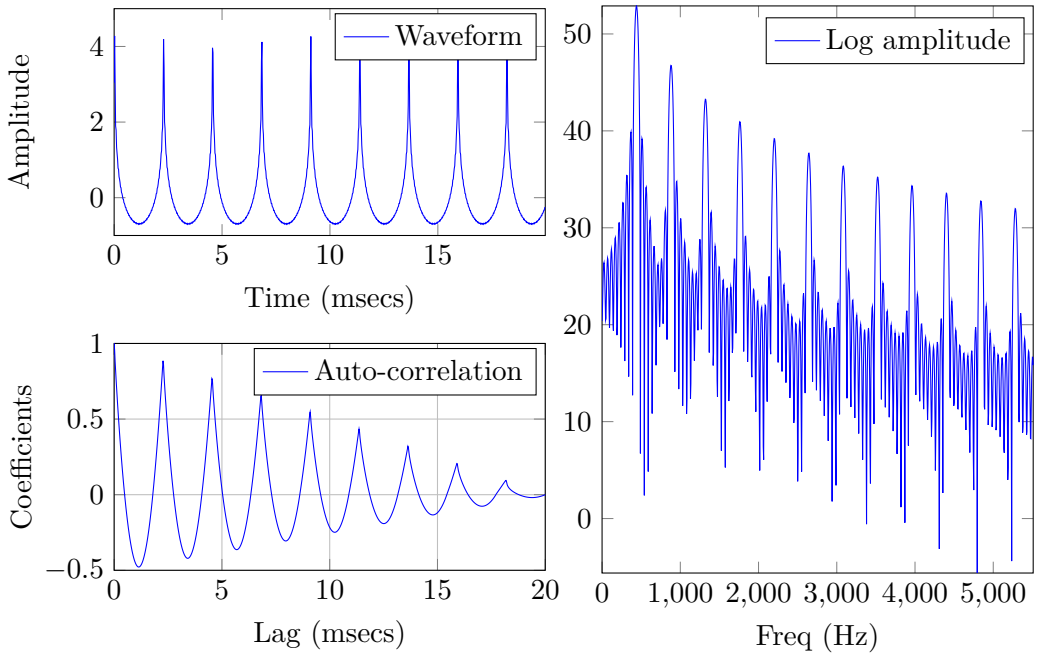


FIGURE 3.2: Spectrogram of a high harmonic energy musical section

fundamental frequency whereas autocorrelation has equally distanced peaks from the time reference, where both distances between peaks represent pitch.



(A) Waveform and STACF

(B) STFT of a window

FIGURE 3.3: Relationship between STFT and STACF

As an example, figure 3.3 shows a $440Hz$ musical note in both time and frequency domains. STFT window illustrates a clear periodic structure with a separation between peaks of $440Hz$. On the other hand, STACF shows maximum values with a period of $2.268ms$, that is $1/2.268ms = 440Hz$.

Spectrogram representation provides plenty of valuable information about a signal. However, due to the high amount of coefficients involved some statistics as the ones discussed next in this chapter constitute a better measure of the spectrum shape at a much lower computational cost.

3.2 Sub-band short-time energy

Sub-band short-time energy represents the average energy existing on each frequency sub-band of the particular window under analysis. It can be mathematically expressed as

$$E_n(m) = \sum_{k=0}^{N} |X_n(k)|^2 |H_m(k)|^2 \tag{3.8}$$

where $H_m(k)$ represents the $m-th$ filter of a filterbank and n the window index. Thus, a windowed segment is filtered by a filterbank of overlapping bandpass filters $H_m(k)$, and the energy on each bandpass filter is averaged.

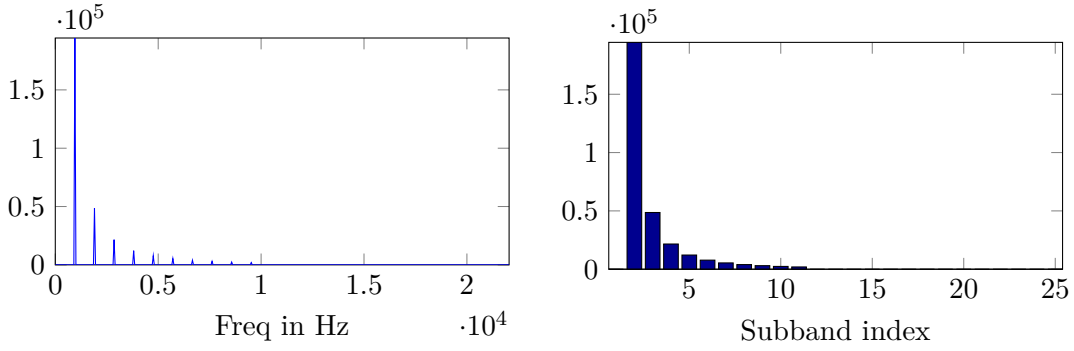
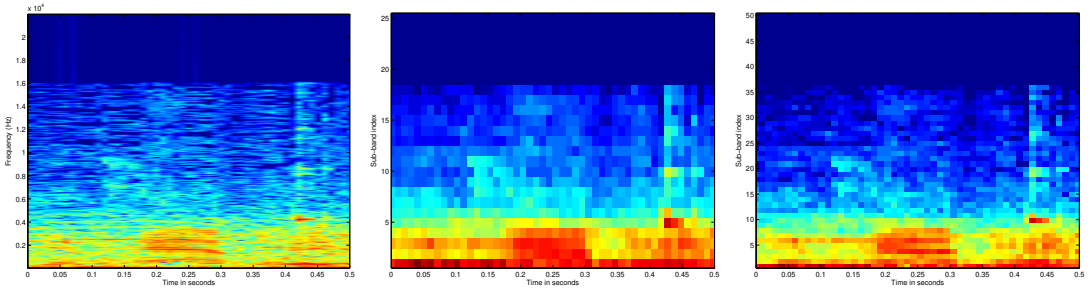


FIGURE 3.4: Subbband energy calculation of a 950Hz harmonic signal

Figure 3.4 shows the spectrum of a 950Hz periodic signal with high energy harmonic components and its corresponding sub-band STE, obtained from ideal band-pass filtering (simply summing energy on the range of frequencies of each sub-band). This approach seems computationally efficient as it only involves additions in the frequency domain.



(A) Spectrogram (B) n=25 bands (C) n=50 bands

FIGURE 3.5: Spectrogram vs Short Time Subband Energy

Nonetheless, a more accurate implementation with overlapping band-pass filters produces more precise results, although more complexity as well. In figure 3.6, $m = 10$ filters serve as a band-pass filterbank to estimate logarithmic SBE.

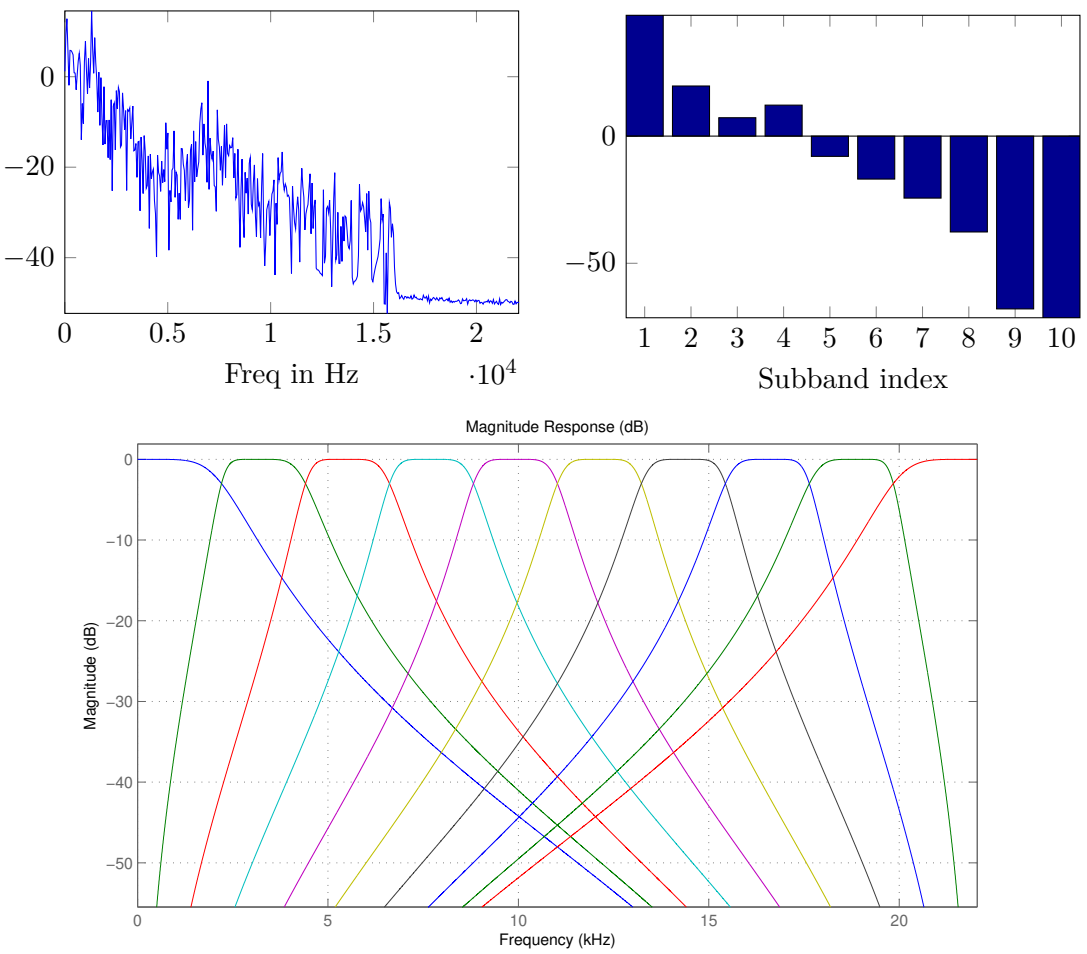


FIGURE 3.6: Logarithmic SBE Filtering with n=10 overlapping filters

Sub-band STE gives a rough estimation of the spectrum shape with an arbitrary low amount of coefficients. This result is particularly useful for the classifier stage. Furthermore, by using sub-band STE noise can be suppressed in specific frequency bands.

However, sub-band STE uses equally-distanced bandpass filters, and since commonly most energy is concentrated on low frequencies many details are missed in this range of frequencies due to the resolution tradeoff involved in Fourier analysis. To overcome this issue, wavelet analysis and dyadic filtering are used as well in this project, and discussed in last chapter.

3.3 Harmonic Product Spectrum

The harmonic product spectrum (HPS) is defined as the geometric mean of the amplitudes of the overtones associated with a particular frequency in the spectrum [5]:

$$HPS(k) = \left(\prod_{r=1}^{R} X(r,k) \right)^{\frac{1}{R}} \tag{3.9}$$

where k is the k^{th} indexed frequency of the overall multiplication, R is the number of harmonics to be considered and $X(r,k)$ is the magnitude spectrum of the signal compressed by a factor r. Assuming the input signal to have a periodic structure, then its spectrum should consist of a series of harmonic components equally distanced. This separation depends on the fundamental frequency and corresponds to integer multiples of it. Thereby, if the spectrum is downsampled to create compressed versions of itself and subsequently compared to the original spectrum, the strongest harmonic peaks will line up. This approach provides a simple computational method to estimate the main frequency (pitch) of a periodic signal.

$$Pitch = \max_{k_i} HPS(k_i) \tag{3.10}$$

Figure 3.7 illustrates the algorithm. The first peak in the original spectrum coincides with the second peak in the spectrum compressed by a factor of two. Similarly, it is aligned with the third peak in the spectrum compressed by a factor of three and so forth. Hence, when the R spectra are multiplied together, the result forms a clear peak at the fundamental frequency. In a last step, the maximum of the HPS is found. In essence, this method computes the greatest common divisor of the harmonic frequencies, which is in effect the main frequency or pitch.

Downsampling of the original spectrum is repeated $R-1$ times, and compared by multiplication to all the downsampled versions. It is required at least R perceptible harmonic peaks in the original spectrum. Otherwise, downsampled versions might lack lined-up peaks and as a result the multiplication on the main frequency might be close to 0, giving wrong pitch estimations.

Although HPS does not perform particularly well with non-musical signals where the structure is not clearly periodic, it has some helpful advantages. HPS is computationally inexpensive, it shows immunity to additive and multiplicative noise, and is adjustable to different kinds of input by simply changing the number of considered overtones (usually R=5) or replacing multiplication by sums at the comparison stage.

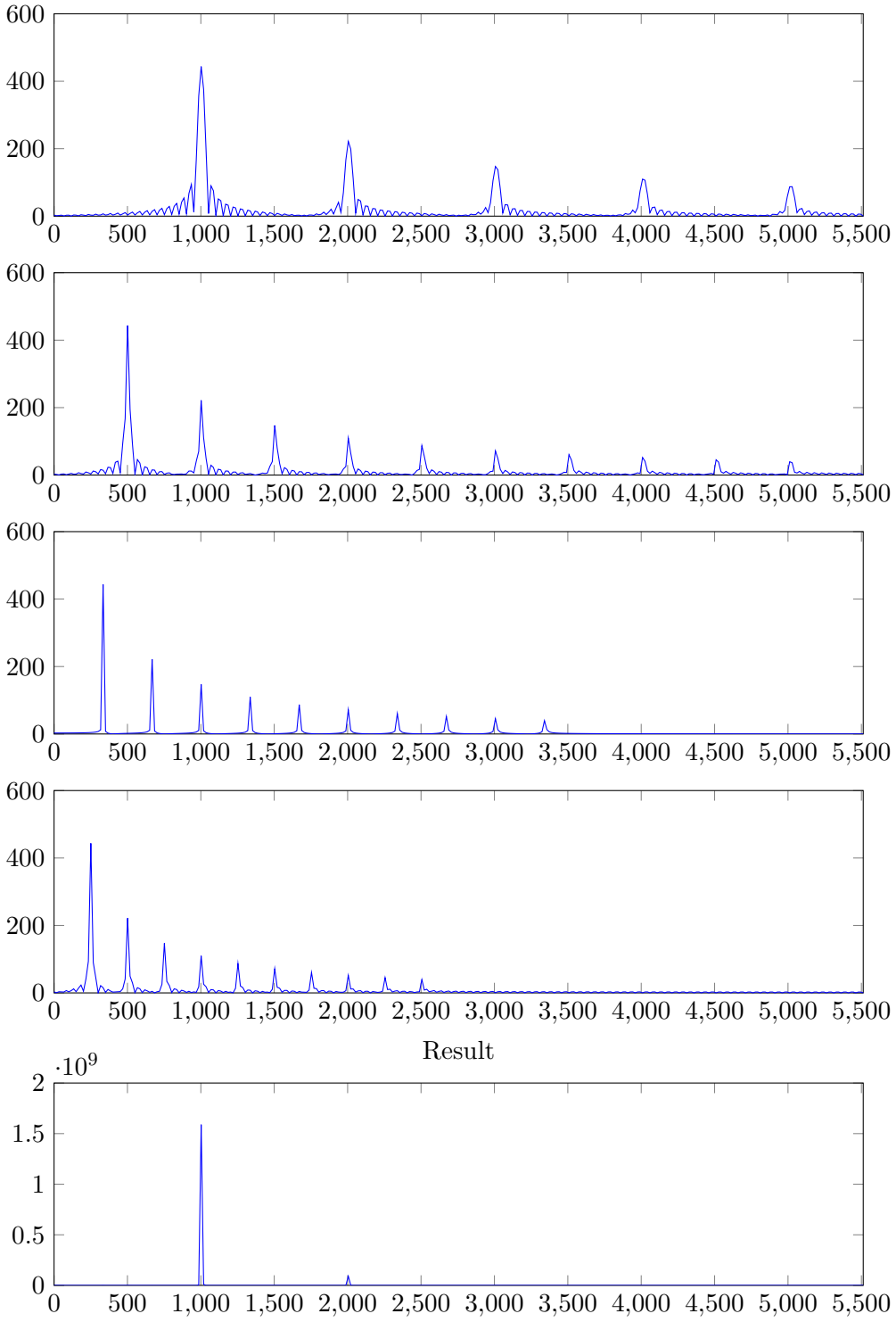
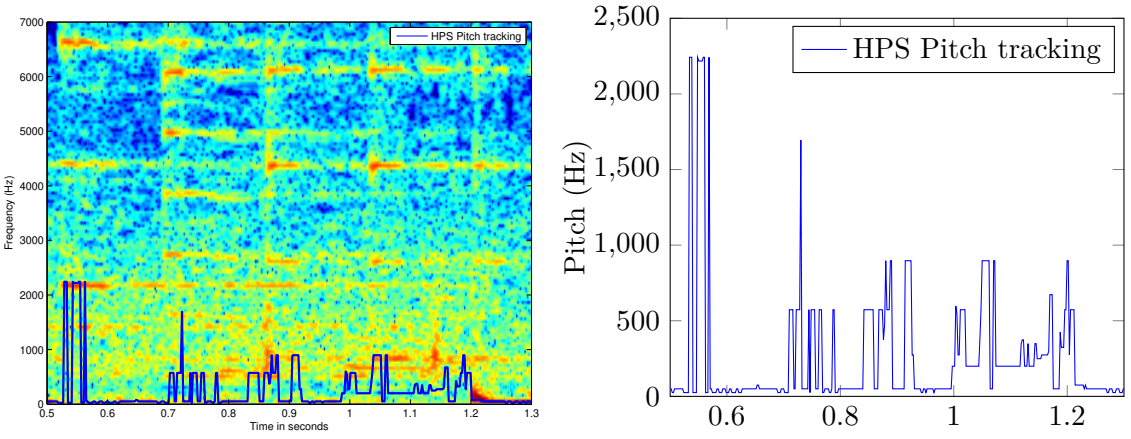


FIGURE 3.7: HPS Implementation

3.4 Maximum Likelihood

Similarly to the ML algorithm in the time-domain, ML in the frequency domain consists in searching the most likely value of the pitch within a certain distribution. Unlike the



(A) Spectrogram/HPS Pitch (B) Pitch tracking

FIGURE 3.8: Pitch tracking of a musical segment

time-domain approach, where estimation was based on the segmentation length, in the frequency domain approach several searches are carried out through a set of possible ideal spectra. The spectrum of this set which best matches the input spectrum is most likely to be the one which determines the pitch of the audio signal [5].

The frequency distribution of a periodic signal shows peaks at integer multiples of the pitch. Therefore, a set of harmonic shaped spectra are created and compared to the audio signal spectrum in the frequency domain for every computable pitch. This comparison is calculated through multiplications in the frequency domain, i.e., filtering of the audio signal by a band-pass filterbank, where the central frequency of these filters is located at integer multiples of the pitch under comparison.

Mathematically it can be expressed as

$$Pitch = \max_{w}\{Y_w\} = \max_{w}\{XH_w\} \tag{3.11}$$

where X denotes the audio input spectrum, H_w the filterbank associated to pitch w comparison, and Y_w is the likelihood function with every pitch comparison values.

In figure 3.9, the audio signal is compared to a harmonic structure of $650Hz$ and the result is summed and stored in a likelihood function. This sum is likely to be maximum when the input signal is a periodic signal of 650 Hz since most of its energy is concentrated inside the filterbank.

This process is repeated for each possible computable pitch and finally the likelihood function Y_w is obtained. It resembles the FFT of the audio signal and its shape is determined by the pitch resolution. Figure 3.10 shows a 650Hz periodic harmonic signal

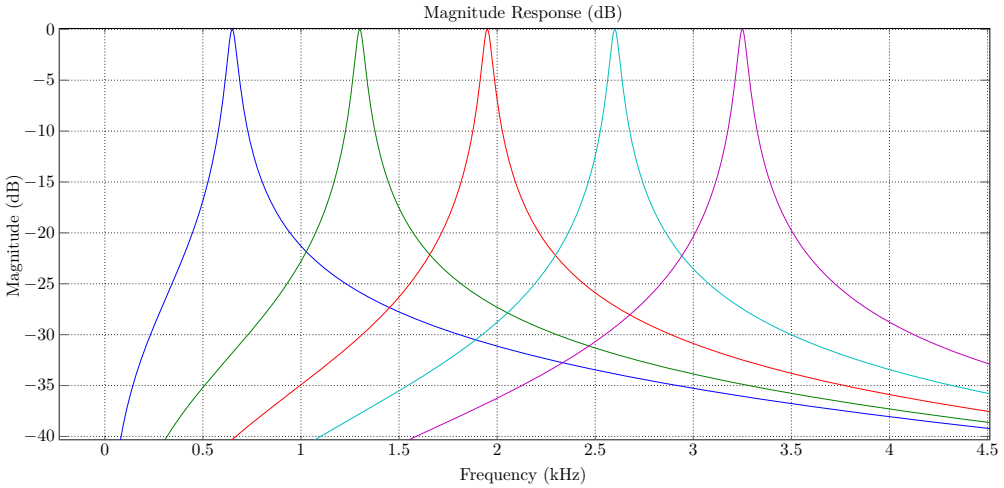


FIGURE 3.9: Filters corresponding to a 650 Hz pitch comparison

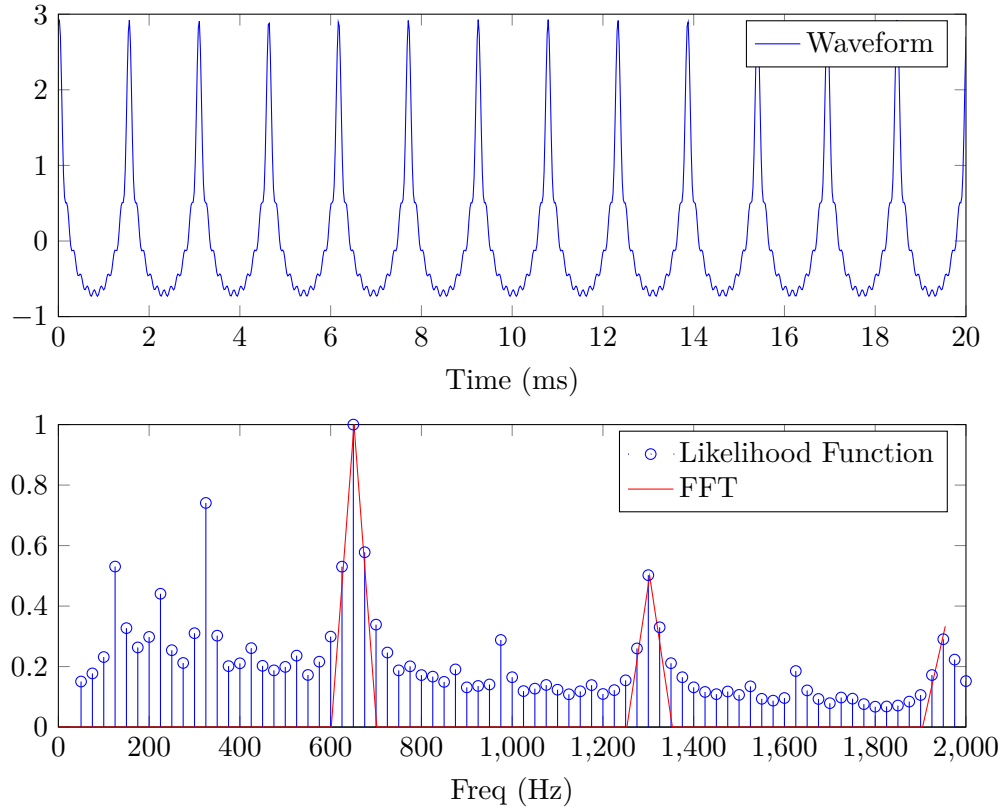


FIGURE 3.10: Periodic signal pitch estimation

analyzed with a pitch resolution of 25Hz, and in effect, a peak appears in the likelihood function.

Furthermore, for non-musical signals such as speech, a different range of detectable pitches must be properly chosen, as well as a different pitch resolution. On the following example, a quasi-periodic voiced sound is analyzed with a pitch resolution of 10Hz and

a pitch range of 50Hz-500Hz. As figure 3.11 presents, the peak at 190 Hz corresponds to the voiced segment period of approximately 5.15ms.

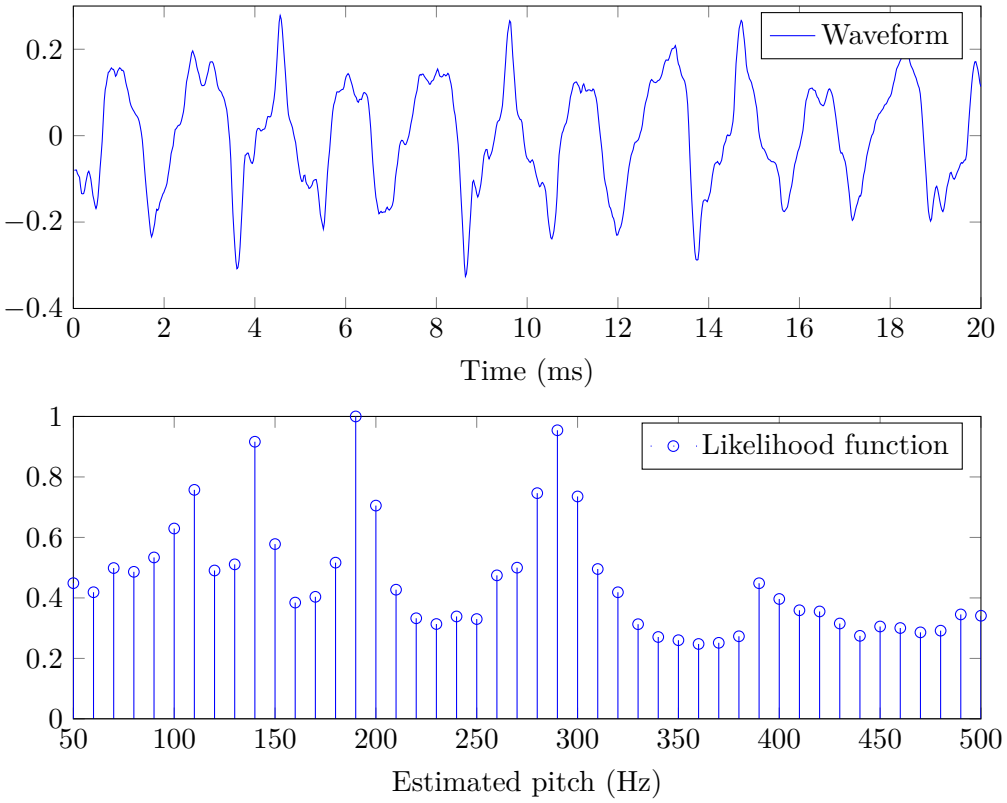


FIGURE 3.11: Voiced speech segment pitch estimation

Since the ML algorithm does not need to be spectrally interpolated like HPS, smaller transform sizes may be used and therefore a considerable improvement can be achieved on the algorithm speed.

Nevertheless, the main shortcoming of this approach lies on the pitch resolution estimation, as it may generate inaccurate results for non-fixed frequencies. As a consequence, the ML method is less tolerant of noise and weak signals than HPS.

3.5 Brightness

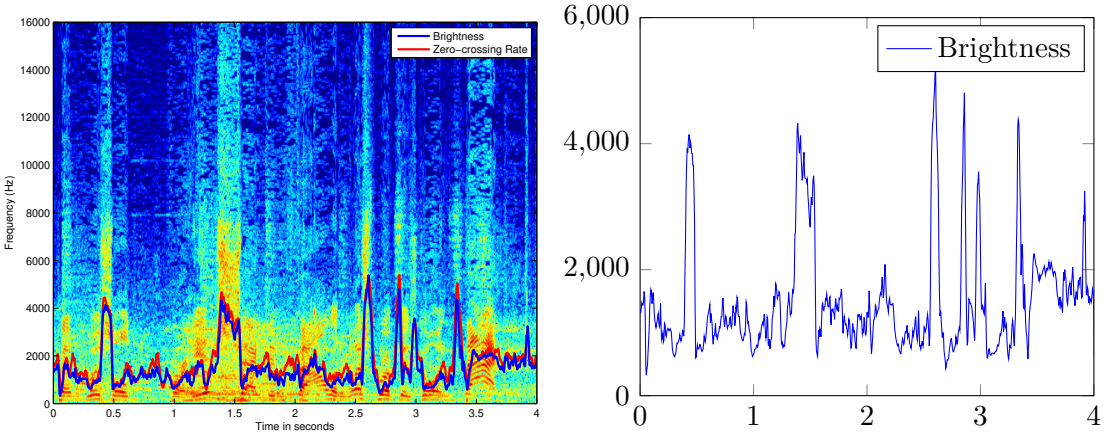
Brightness represents the frequency centroid (center of gravity) on a given frame of the magnitude STFT. It is defined as follows

$$BR_n = \frac{\sum_k k|X_n(k)|^2}{\sum_k |X_n(k)|^2} \qquad k = 0, 1, \ldots, N-1 \tag{3.12}$$

where $X_n(k)$ is the STFT of the audio signal on the $n-th$ window. The denominator represents the whole energy signal computed from the spectrum and the numerator represents a weighted sum of such energy. It can be explained as follows

- High values of $X_n(k)$ magnitude on small k will contribute less to the overall sum, resulting in a left-shifting of the centroid (low frequencies).
- High values of $X_n(k)$ magnitude on high k will contribute more to the overall sum, resulting in a right-shifting of the centroid (high frequencies).
- Dividing by the total energy will limit the result to the computed frequency range.

This centroid portrays a measure of spectral shape and gives an idea of where the energy of the signal is concentrated. Lower brightness points to the presence of high energy values on low frequencies, whereas higher brightness corresponds to "brighter" textures containing higher frequencies. Moreover, a middle-valued brightness might indicate either a plain spectrum or a predominance of middle frequencies.



(A) Brightness/ZCR pitch estimation (B) Brightness tracking

FIGURE 3.12: Brightness of a speech segment

Although brightness is not a direct pitch measure, it is closely related to it in the sense that main frequencies have higher energy than their harmonics, and therefore they compact most of it. Its relationship with the time domain feature ZCR is shown in figure 3.12, exposing its strong connection to pitch. Furthermore, brightness can be considered the first-order property of the spectrum distribution.

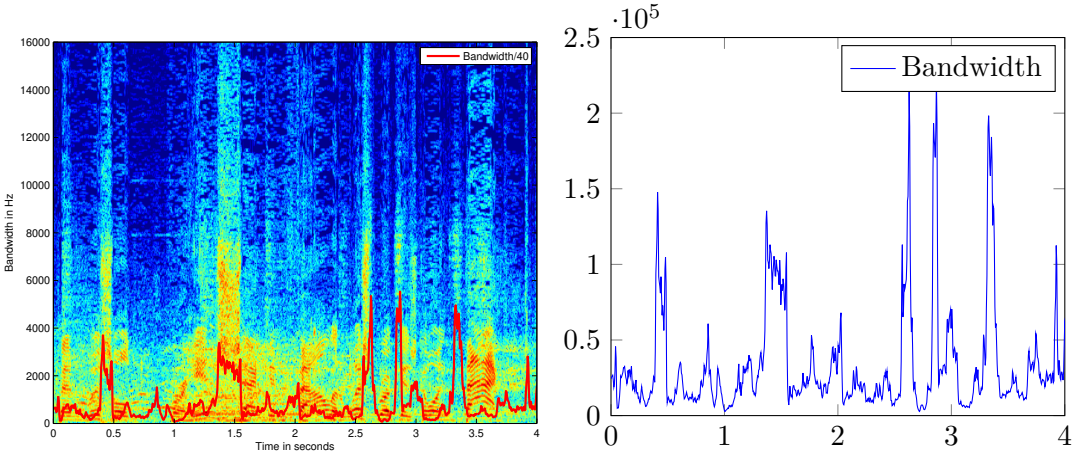
3.6 Bandwidth

Bandwidth describes the deviation of the spectrum with respect to brightness. It constitutes the second-order property of the spectrum distribution and is defined as

$$BW_n = \frac{\sum_k (k - BR_n)^2 |X_n(k)|^2}{\sum_k |X_n(k)|^2} \qquad k = 0, 1, \ldots, N-1 \tag{3.13}$$

where $X_n(k)$ is the STFT of the audio signal on the n^{th} window and BR is the previously explained brightness. The denominator represents, as well as in brightness, the whole signal energy computed from the spectrum and the numerator represents a weighted sum of such energy respect to the distance to brightness, i.e., deviation with respect to brightness. It can be explained as follows

- The term $(k - BR_n)^2$ assigns more weight to k's further from BR_n and less weight to k's closer to BR_n.
- Compact spectrum distributions have most of its energy near BR_n. Therefore, k values with most weight assigned will have a low energy and will barely contribute to the overall sum.
- On the other hand, spread spectrum distributions have energy dispersed far from BR_n. Hence, there will be k values far from BR_n with high energy that will contribute considerably to the overall sum, resulting in a higher bandwidth.
- Dividing by the total energy will normalize the result to a Hz frequency scale.



(A) Bandwidth/Spectrogram comparison (B) Bandwidth tracking

FIGURE 3.13: Bandwidth of a speech segment

In the case of basketball videos, and as it is shown in figure 3.13, bandwidth can provide valuable information of transitions between events due to changes in the spectrum distribution. For example, a low constant bandwidth might suggest the presence of audience noise in the low-middle frequency range, whereas the presence of peaks shows an energy increase in the spectrum components, due to coming up events in the game.

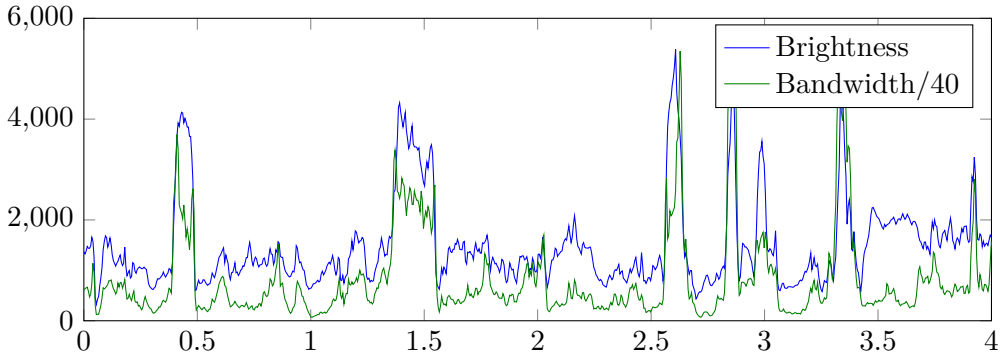


FIGURE 3.14: Brightness - Bandwidth/40 comparison

Finally, it is important to remark the relationship between bandwidth and brightness, as figure 3.14 does.

3.7 Entropy

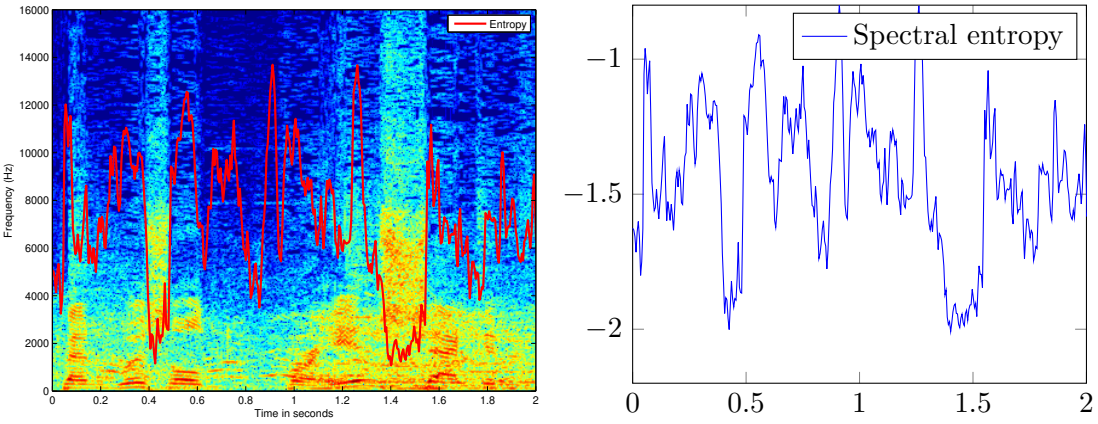
In the field of information theory, and particularly in signal processing, Shannon entropy describes the information complexity of a signal. Applied on a spectrum distribution it is expressed as

$$H_n = -\sum_{k=0}^{N-1} P_k \log P_k \tag{3.14}$$

where P_k depicts the percentage of total energy that lies on sample k of the spectrum in the n^{th} window, i.e.,

$$P_k = \frac{|X_n(k)|^2}{\sum_k |X_n(k)|^2} \qquad k = 0, 1, \ldots, N-1 \tag{3.15}$$

Plain spectrum distributions, such as random white noise, yield maximum entropy, whereas spectrum with peaks and concentrated energy result in low entropy values. Thus, it is a helpful measure along with other features to differentiate events with a strong presence of noise, such as excited or plain audience sound.



(A) Entropy/Spectrogram comparison (B) Entropy tracking

FIGURE 3.15: Entropy of a speech segment

3.8 Flux

Spectral flux is defined as the Euclidean distance between two normalized spectra, i.e., squared difference between the normalized magnitudes of successive spectral distributions. Flux represents an average variation value of the spectrum between adjacent frames. Hence, flux is a measure of the amount of local spectral change.

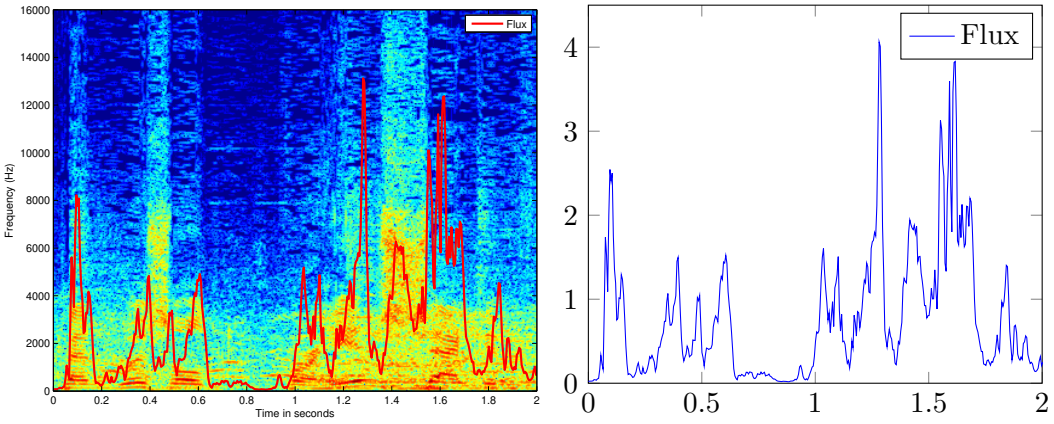
$$P_n = \sum_{k=0}^{N-1} (F_n(k) - F_{n-1}(k))^2 \tag{3.16}$$

where $F_n(k)$ and $F_{n-1}(k)$ are the normalized magnitude of the Fourier transform at the current time frame n and the previous time frame $n-1$ respectively. Calculated in this manner, the spectral flux is not dependent either on the whole energy as the spectra are normalized, nor on phase considerations since only magnitudes are compared. Unlike other statistics it depends on the time frame $n-1$ for its computation, i.e., it depends on previous frames. Thus it seems useful for purposes such as detecting transitions between events, where it reaches high values.

Logarithmic spectra can be used for calculations alternatively. Besides, if we extract the normalization factor from $F_n(k)$ it is possible to express equation 3.16 in terms of $X_n(k)$, which yields the following expression:

$$P_n = \frac{1}{N-1} \sum_{k=0}^{N-1} [\log(X_n(k)) - \log(X_{n-1}(k))]^2 \tag{3.17}$$

where $X_n(k)$ is the k^{th} component of the spectrum in the n^{th} frame.



(A) Flux/Spectrogram comparison (B) Flux tracking

FIGURE 3.16: Flux of a speech segment

As it can be inferred from figure 3.16, a low time varying spectrum has the lowest flux whereas a spectrogram with many temporal changes shows a high flux. This might be useful for event transition detection between keywords. Silences and plain audience would show a low flux, speech segments a medium flux due to silence transitions inbetween words, and abrupt changes in the spectrum distribution, such as the transition moment between two events, would show the maximum flux.

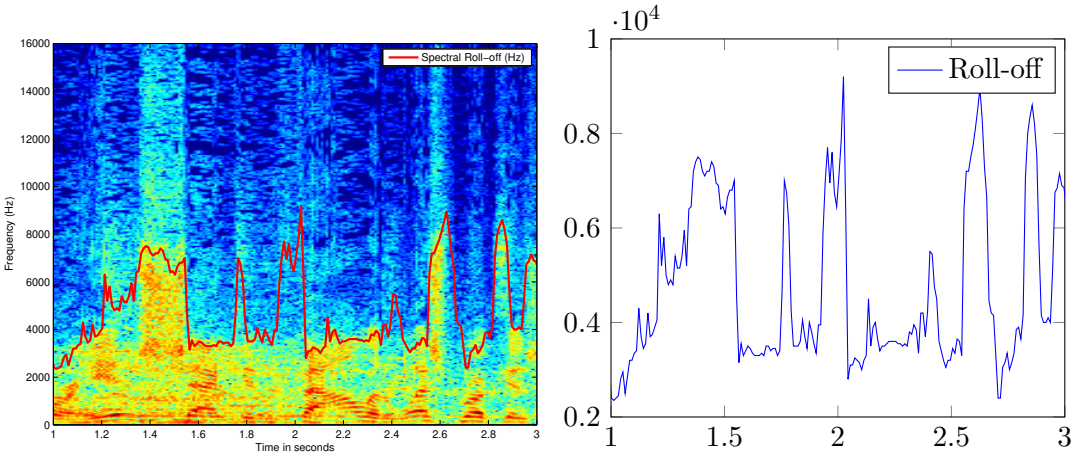
Spectral flux is also used in audio processing to determine the timbre of an audio signal, or in onset detection among other things. Onset refers to the beginning of a musical note or other sound, in which the amplitude rises from zero to an initial peak. In speech, onset refers to consonant sound or sounds at the beginning of a syllable occurring before the nucleus.

3.9 Rolloff

The spectral rolloff is defined as the frequency R below which 85% of the spectrum magnitude distribution is concentrated. It is mathematically expressed as

$$\sum_{k=0}^{R} |X_n(k)| = 0.85 \sum_{k=0}^{N-1} |X_n(k)| \tag{3.18}$$

Rolloff is a measure of spectral shape, and determines the range of frequencies containing most energy. Low values on rolloff indicate a strong presence of high energy low frequency content, such as those on speech segments, while high values point the absence of low frequencies or a strong presence of noise, that is, silence or audience noise respectively.



(A) Rolloff/Spectrogram comparison (B) Rolloff tracking

FIGURE 3.17: Spectrogram and rolloff of a speech segment

Furthermore, rolloff provides an accurate measure of the range of frequencies under interest, since they compact most energy. For instance, in figure 3.17 each speech segment has a value of approximately 4KHz, which corresponds to the human speech range, suggesting the presence of the commentator.

3.10 Crest Factor and PAPR

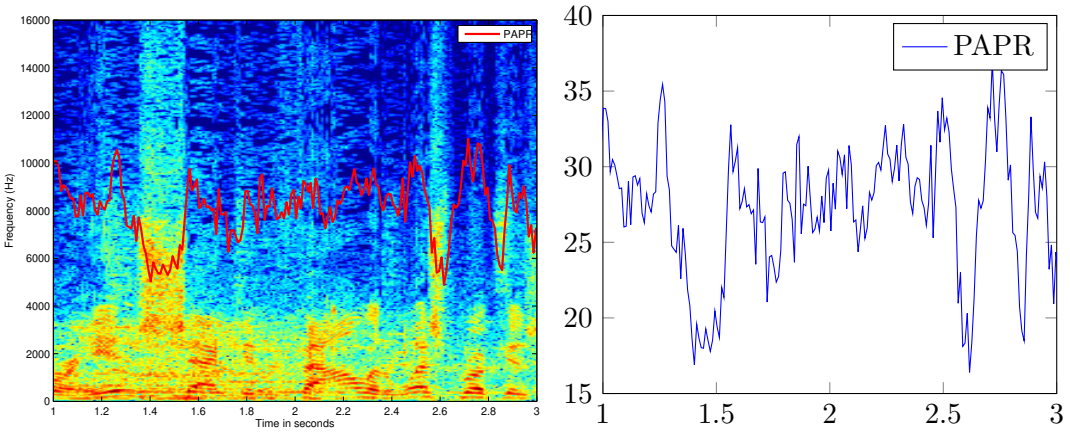
The spectral crest factor of a signal is defined as the ratio of the peak amplitude of a waveform spectrum to its RMS energy.

$$C = \frac{\max\{|X_n(k)|\}}{RMS_{energy}} = \frac{\max\{|X_n(k)|\}}{\frac{1}{N}\sum_{k=0}^{N-1}|X_n(k)|} \tag{3.19}$$

Although in principle the crest factor is a measure of a waveform, it can also be utilized to roughly determine the shape of a spectral distribution. Basically, crest factor indicates how extreme peaks are on the spectrum, varying from values of 1 for plain distributions such as noise, to higher values for peak-shaped spectra.

Peak-to-average power ratio (PAPR) is a related measure obtained directly from the crest factor, and provides the average power on a certain distribution. It is defined as

$$PAPR = \frac{\max\{|X_n(k)|^2\}}{(RMS_{energy})^2} = C^2 \tag{3.20}$$



(A) PAPR/Spectrogram comparison (B) PAPR Tracking (dB)

FIGURE 3.18: Crest factor of a segment

Since crest factor and PAPR are dimensionless quantities, they are usually expressed as a ratio. In our scope, PAPR gives more useful results as it is a power ratio on the spectrum distribution, measured in dB. Figure 3.18 shows the short-time PAPR of a speech segment, demonstrating the following results:

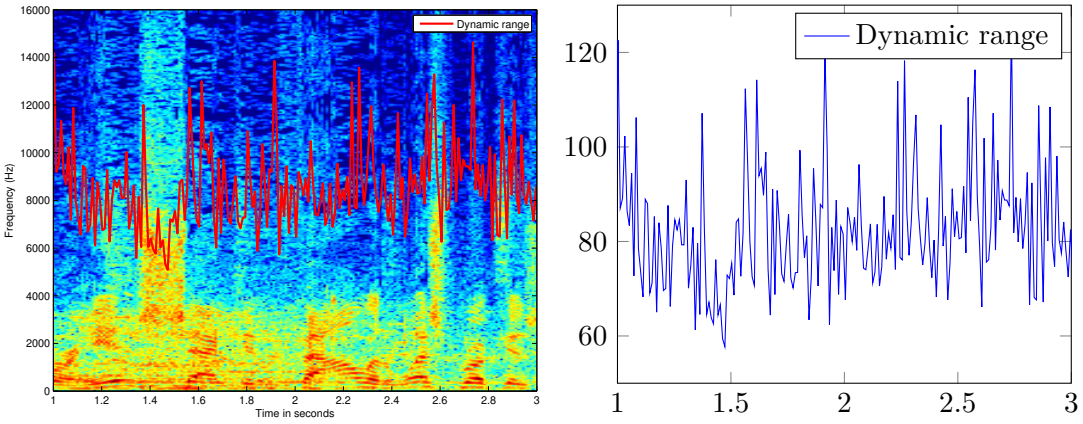
- Maximum values are reached when the signal is comprised of peaks and low RMS energy. That is the case of voiced speech or harmonic signals with no influence of noise.
- Average values show the presence of a peak-shape spectrum with a considerable RMS energy, that is, previous case with a significant amount of noise.
- Lowest values show lack of a peak-shaped spectrum. Therefore, lowest values point the absence of speech, silences and noise, since numerator term is considerably low compared to overall energy.

3.11 Dynamic range

The spectral dynamic range can be defined as the ratio of the largest to the smallest possible values of the spectrum power distribution. Mathematically can be expressed as

$$DR(dB) = 20\log_{10}\left(\frac{\max\{|X_n(k)|\}}{\min\{|X_n(k)|\}}\right) \tag{3.21}$$

Thereby, highest values suggest the existence of strong peaks and absence of weak peaks, whereas lowest values point the contrary. For computation, null values are ignored.

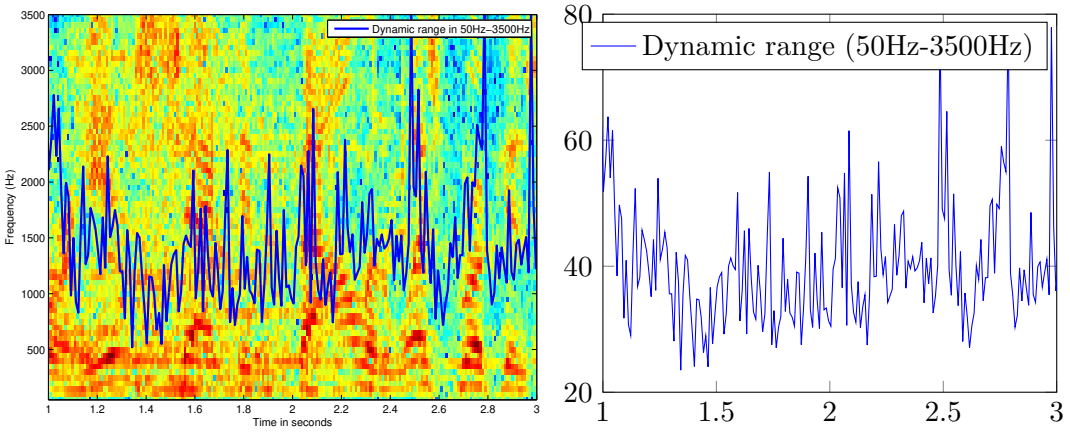


(A) Dynamic range/Spectrogram comparison (B) DR Tracking (dB)

FIGURE 3.19: Dynamic range

However, since high frequencies usually have very low energy components, dynamic range reaches extremely high values and it turns into a measure of low frequency peaks energy. This is shown in figure 3.19, where dynamic range varies approximately from 50dB to 100dB. In order to avoid this situation, a sub-band analysis might be performed instead.

Figure 3.20 shows a dynamic range analysis in the sub-band of 50Hz-3500Hz, displaying values between 30dB and 80dB. This method provides better results to understand and interpret the speech spectrum distribution.



(A) Dynamic range/Spectrogram comparison (B) DR Tracking (dB)

FIGURE 3.20: Dynamic range of 50Hz-3500Hz subband

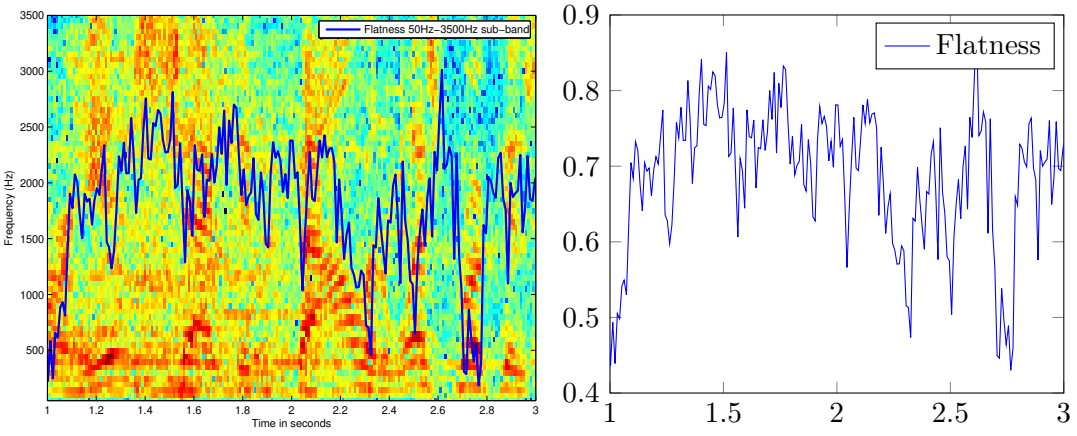
3.12 Flatness

The spectral flatness is calculated by dividing the geometric mean of the power spectrum by its arithmetic mean, i.e.,

$$Flatness = \frac{\sqrt[N]{\prod_{k=0}^{N-1} |X_n(k)|}}{\frac{1}{N}\sum_{k=0}^{N-1} |X_n(k)|} \tag{3.22}$$

This feature provides a method to quantify how tone-like a sound is. The meaning of tonal in this context is in the sense of the amount of peaks or resonant structure in a power spectrum. As a result, sometimes it is referred to as tonality coefficient.

A high spectral flatness indicates that the spectrum has a similar amount of power in every spectral band, i.e., white noise, and the graph of the spectrum would appear relatively flat and smooth. On the other hand, a low spectral flatness indicates that the spectral power is concentrated in a relatively small number of bands, resembling a mixture of sine waves, with a peak-shaped spectrum distribution. Thus, flatness reaches lowest values for harmonic tonal signals and highest values for a flat spectrum of white noise.



(A) Flatness/Spectrogram comparison (B) Flatness tracking

FIGURE 3.21: Flatness of 50Hz-3500Hz subband

It is important to consider that zero-valued frequencies yield a flatness of 0, as equation 3.22 numerator reaches values close to 0 due to the multiplications involved. That makes this measure most useful in sub-bands containing high-energy components. Hence, spectral flatness is more likely to be measured within a sub-band rather than across the whole spectrum.

In figure 3.21 a speech segment is analyzed on the 50Hz-3500Hz sub-band. Flatness presents low values for clear defined voiced speech and higher values for silences and ambient noise, as expected.

Since flatness is dimensionless, it can be measured as a ratio, varying from 0 to 1, being typically measured in decibels.

Chapter 4

Cepstral analysis

According to the *source/system* speech model, the speech signal is separated into an excitation sequence, quasi-periodic for voiced sounds and random noise for unvoiced sounds, and the impulse response corresponding to the vocal tract linear system. Cepstral analysis allows to transform both signals combined by convolution into an addition, providing methods to easily separate their contributions. Cepstral analysis requires transforming from one domain into another in order to linearly process the signal. This domain transformation process is called homomorphic filtering.

Cepstrum is useful for detecting echoes, estimating pitch and principally in the field of automatic speech recognition (ASR). Mel-Frequency Cepstral Coefficients (MFCC) are a cepstrum-based audio feature and are widely used nowadays due to their effectiveness on speech recognition. They are explained in detail at the end of this chapter.

To properly understand the concept of cepstrum an its applications, a brief introduction to homomorphic filtering must be provided first.

Homomorphic systems and homomorphic filtering

Homomorphic systems are non-linear systems based on a generalized principle of superposition. The corresponding representation depends on the operation of interest. For a linear system $\mathcal{L}(\cdot)$ the principle of superposition is satisfied for the addition operation:

$$\mathcal{L}(x_1[n] + x_2[n]) = \mathcal{L}(x_1[n]) + \mathcal{L}(x_2[n]) \tag{4.1}$$

$$\mathcal{L}(\alpha x[n]) = \alpha \mathcal{L}(x[n]) \tag{4.2}$$

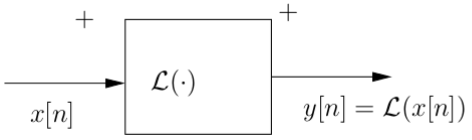


FIGURE 4.1: Linear system for addition

In the case of convolution operation the principle of superposition would result in

$$y[n] = \mathcal{H}(x_1[n] * x_2[n]) = \mathcal{H}(x_1[n]) * \mathcal{H}(x_2[n]) = y_1[n] * y_2[n] \tag{4.3}$$

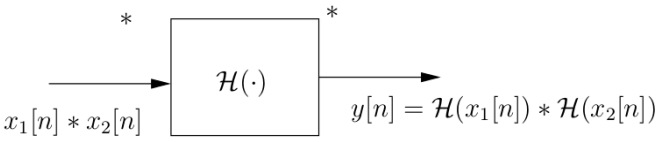


FIGURE 4.2: Homomorphic system for convolution

An important property of homomorphic systems for convolution is that they can be represented as a cascade of three homomorphic systems depicted as follows

- The first system takes inputs combined by convolution and transforms them into an additive combination of the corresponding outputs, that is
  $x[n] = x_1[n] * x_2[n] \longleftrightarrow \hat{x}[n] = \hat{x}_1[n] + \hat{x}_2[n]$
- The second system is a conventional linear system that obeys the principle of superposition
  $\hat{y}[n] = \mathcal{L}(\hat{x}_1[n] + \hat{x}_2[n]) = \hat{y}_1[n] + \hat{y}_2[n]$
- The third system is the inverse of the first system: it transforms signals combined by addition into signals combined by convolution
  $\hat{y}[n] = \hat{y}_1[n] + \hat{y}_2[n] \longleftrightarrow y[n] = y_1[n] * y_2[n]$

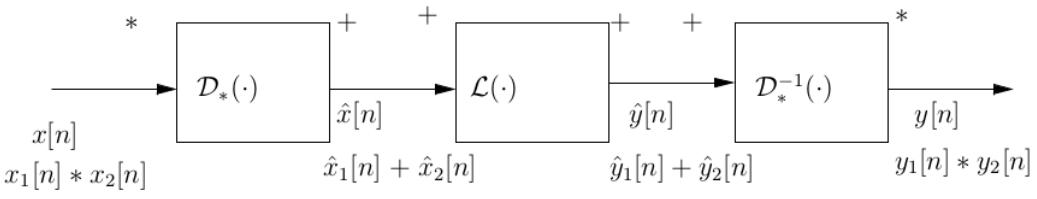


FIGURE 4.3: Canonical system for a homomorphic system for convolution

This is important because the design of such system reduces to the design of the linear system $\mathcal{L}(\cdot)$. Thus, homomorphic filtering is a generalized technique involving a nonlinear mapping to a different domain where linear filters are applied, followed by mapping back to the original domain.

Furthermore, the z-Transform of two convoluted signals results in the multiplication of their two respective z-Transforms. If a logarithm is applied to previous z-Transform, then we obtain a sum, as the following equations show:

$$x[n] = x_1[n] * x_2[n] \tag{4.4}$$

$$X(z) = X_1(z) \cdot X_2(z) \tag{4.5}$$

$$\log[X(z)] = \log[X_1(z)X_2(z)] = \log[X_1(z)] + \log[X_2(z)] \tag{4.6}$$

Thus, a logarithmic mapping on the z-Transform obeys the principle of superposition on homomorphic systems for convolution, and corresponds to the concepts of cepstrum and complex cepstrum.

4.1 Cepstrum

The complex cepstrum of a signal $x[n]$ is defined by

$$\hat{x}[n] = \mathcal{F}^{-1}\left[\log\{X(e^{j\omega})\}\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\{X(e^{j\omega})\}e^{j\omega n}d\omega \tag{4.7}$$

The real cepstrum, or just cepstrum, is defined as the even part of the complex cepstrum, and can be expressed as

$$c[n] = \mathcal{F}^{-1}\left[\log|X(e^{j\omega})|\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(e^{j\omega})|e^{j\omega n}d\omega \tag{4.8}$$

where $|X(e^{j\omega})|$ is the magnitude spectrum of $x[n]$. The independent variable of the cepstrum and complex cepstrum, called *quefrency*, is nominally time, although not in the regular sense of a signal in the time domain.

The original definition of cepstrum was motivated by the fact that the logarithm of the Fourier spectrum of a signal containing an echo has an additive periodic component depending only on the echo size and delay, and that further Fourier analysis of this logarithmic spectrum allows detecting the presence of such echo. Hence, logarithmic spectrum can be considered as a waveform for further Fourier analysis. This approach

creates some new vocabulary such as *cepstrum, liftering* or *quefrency* in place of spectrum, filtering and frequency respectively.

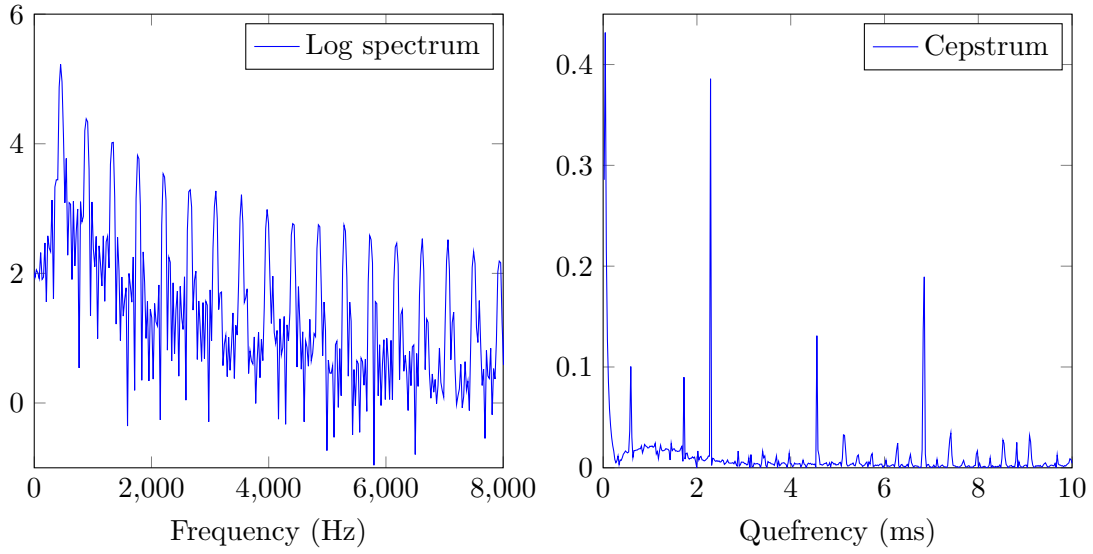


FIGURE 4.4: DFT and Cepstrum of a triangular signal with 25ms echo

In addition, cepstrum is related to the more general concept of homomorphic filtering of signals that are combined by convolution. In the theory of homomorphic systems $D_*\{\cdot\}$ is called the characteristic system for convolution. The connection between the cepstrum concept and homomorphic filtering of convolved signals is that the complex cepstrum operator transforms convolution into addition. That is, if $x[n]$ is a time-domain signal and $\hat{x}[n]$ the complex cepstrum of $x[n]$, then

$$x[n] = x_1[n] * x_2[n] \tag{4.9}$$

$$\hat{x}[n] = D_*\{x_1[n] * x_2[n]\} = \hat{x}_1[n] + \hat{x}_2[n] \tag{4.10}$$

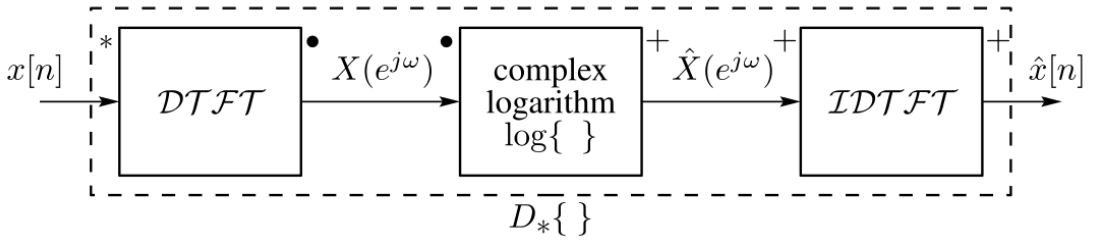


FIGURE 4.5: Computing the complex cepstrum using the DTFT

The main issue in the complex cepstrum is the computation of the complex logarithm, particularly the computation of the phase angle $arg\{X(e^{j\omega})\}$ which must be considered in order to preserve an additive combination of phases for two signals combined by

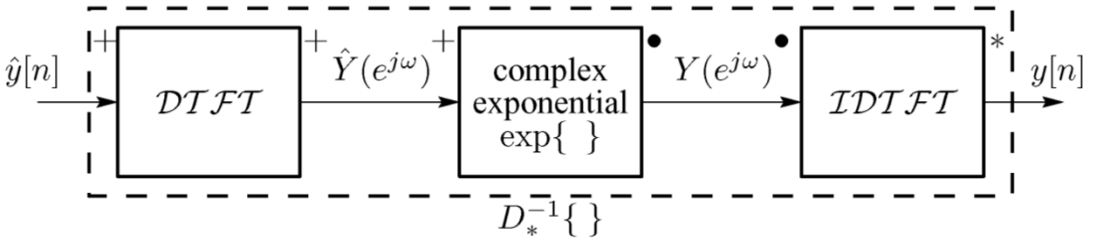


FIGURE 4.6: Inverse transform of the characteristic system for convolution (inverse complex cepstrum)

convolution. Since speech processing does not require the phase angle of the logarithmic spectrum, computation of the real cepstrum can be easily performed by the use of the DTFT.

Transforming convolutions into sums, which is satisfied for both the cepstrum and the complex cepstrum, is what turns this domain transform suitable for speech analysis. According to the *source/system* model, speech production involves convolution of the low-frequency periodic excitation from the vocal cords with the vocal tract impulse response. Since they convolve in the time domain and multiply in the frequency domain, they are also additive in different regions in the quefrency domain.

As a consequence, more robustness against convolutional noise and interference can be simply accomplished by performing a cepstrum analysis. If this interference is assumed to be constant, it can be discarded by a simple mean removal. Besides, enhancements performed in the logarithmic spectral domain can improve sound intelligibility in further stages of processing.

Short-time Cepstrum

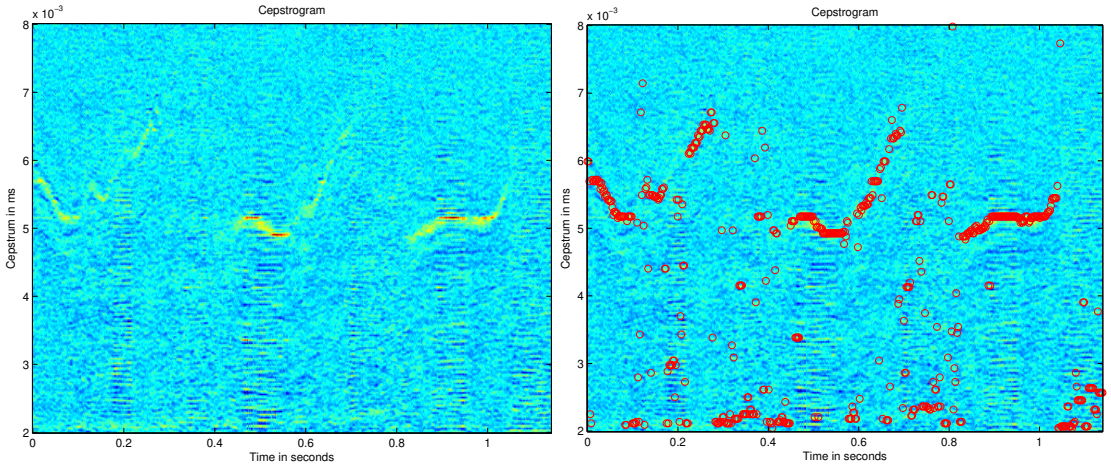
A precise analysis of the speech signal must be done in order to properly extract useful information. Short-time analysis is necessary in the cepstral domain as a result of the non-stationarity of the speech signal. Thus, short-time cepstrum is used as a representation of speech and as a basis for estimating the parameters of the speech generation model.

This representation involves replacing DTFT by STFT. Hence, the short-time cepstrum is defined as

$$c_n[m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X_n(e^{j\omega})| e^{j\omega m} d\omega \tag{4.11}$$

where $|X_n(e^{j\omega})|$ is the magnitude spectrum of $x[m]$ in the n^{th} window.

In similarity with the frequency domain, the short-time cepstrum is a sequence of cepstra of windowed segments of the audio waveform. By analogy, a *cepstrogram* is a 2D image obtained by plotting in color the magnitude of the short-time cepstrum as a function of *quefrency*, m and the time window n.



(A) Cepstrogram (B) Maxima on each window

FIGURE 4.7: Cepstrogram of the speech extract: 'into the starting five'

Since speech and audio processing does not need the argument of the logarithmic spectrum for processing, only the magnitude is considered and as a consequence cepstrum can be implemented by the use of the DFT. Otherwise other approaches using z-Transform are required, although they are not implemented in this project.

Using the DFT to implement the short-time cepstrum, the previous equations would be replaced by

$$c_n[m] = \frac{1}{N}\sum_{k=0}^{N-1} \log|X_n(k)|e^{j(2\pi k/N)m} \qquad k = 0, 1, \ldots, N-1 \tag{4.12}$$

where $|X_n(k)|$ represents the magnitude spectrum of $x[m]$ in the n^{th} frame. Due to the sampling of the logarithmic DTFT, time-domain cepstrum will be N-periodic, although this effect can be made negligible by using large N values, i.e., for a window length L, then $N \geq L$.

Liftering and smoothing

The term *liftering* refers to the fact of signal filtering in the cepstral domain. Liftering of convolved signals is achieved by windowing the resulting cepstrum of such convolution. For a signal $x[m] = x_1[m] * x_2[m]$ whose cepstrum is $\hat{x}[m] = \hat{x}_1[m] + \hat{x}_2[m]$ it follows

that

$$\hat{y}[m] = g[m]\hat{x}[m] = g[m]\left(\hat{x}_1[m] + \hat{x}_2[m]\right) = g[m]\hat{x}_1[m] + g[m]\hat{x}_2[m] \tag{4.13}$$

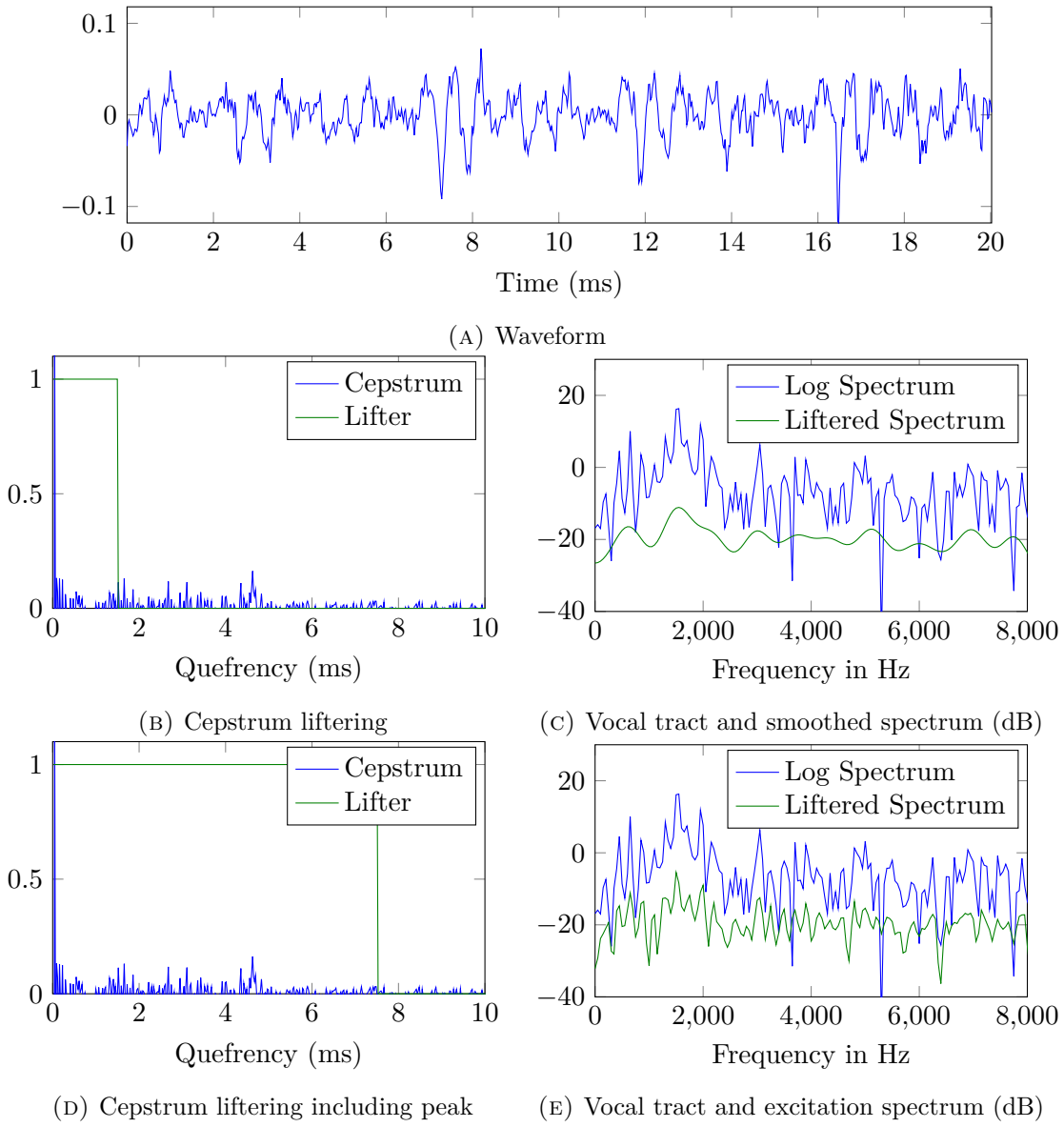
According to equation 4.13, if we consider $\hat{x}_1[m]$ to be the vocal tract cepstrum and $\hat{x}_2[m]$ the excitation cepstrum, an appropriate lifter $g[m]$ which excludes $\hat{x}_2[m]$ would only select the vocal tract cepstrum content. Thus, by liftering the cepstrum it is possible to separate information of the vocal tract from the short-time speech spectrum. Therefore, if on a voiced sound the excitation is discarded by liftering, the vocal tract spectral shape for that particular voiced sound can be obtained. Furthermore, it is only necessary to subtract that vocal tract from the cepstrum to obtain the excitation.

Analogously to the time-frequency relationship on the representation of a signal, the low quefrency part of the cepstrum depicts slow variations in the logarithmic spectrum, whereas high quefrency components correspond to more rapid fluctuations of the logarithmic spectrum, due primarily to the excitation. Consequently, low quefrencies retain the general spectral shape of the signal with peaks corresponding to the vocal tract formants for the segment of speech under analysis, while high quefrencies represent primarily the excitation. The effect of low-pass liftering results in a smoothed version of the spectrum as liftering resembles frequency filtering, although considering the spectrum a waveform.

In figure 4.8 an example of liftering is shown. After windowing a 20ms segment, the excerpt under analysis shows a 4.6ms quasi-periodic voiced sound. After cepstrum calculation, the peak corresponding to that sound appears at quefrency 4.6ms, as well as other peaks related to higher frequency components. That 4.6ms quefrency peak corresponds to $1/4.6ms = 217Hz$, matches to the main peak of the signal spectrum (pitch) and reflects a 217Hz periodic structure on it.

In the first case, cepstrum is liftered by a 1.5ms lifter, resulting in discarding the peak related to the vocal excitation. As a result, the liftered spectrum shows the formants of the vocal tract envelope. On the other hand, the second case illustrates a 7.5ms lowpass lifter, which includes the previous mentioned excitation, thereby yielding a spectrum reflecting that 217Hz periodic structure, and therefore including the voiced excitation. Furthermore, liftered spectrum shows a shape closer to the original one.

In the case of unvoiced excitations, since they are modeled as random white noise their spectra are assumed to be constant. Excitation is further filtered by the vocal tract, reshaping noise excitation spectrum and giving it the vocal tract formants shape. Hence, spectrum of unvoiced sounds is assumed to have the shape of the vocal tract influenced by rapid variations due to the noise excitation.



(A) Waveform

(B) Cepstrum liftering

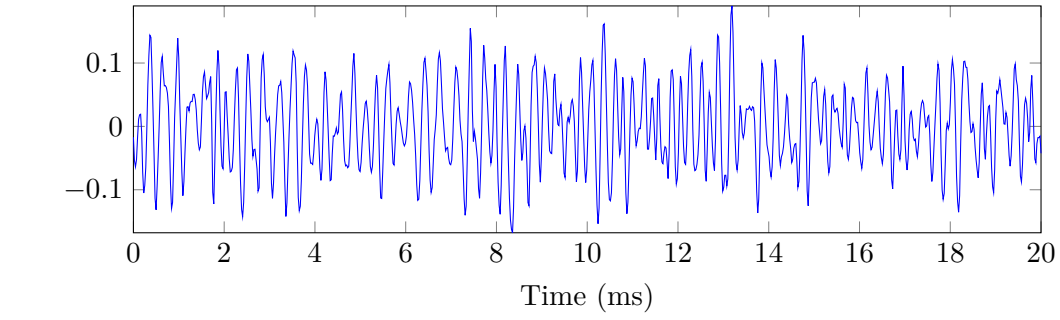
(C) Vocal tract and smoothed spectrum (dB)

(D) Cepstrum liftering including peak

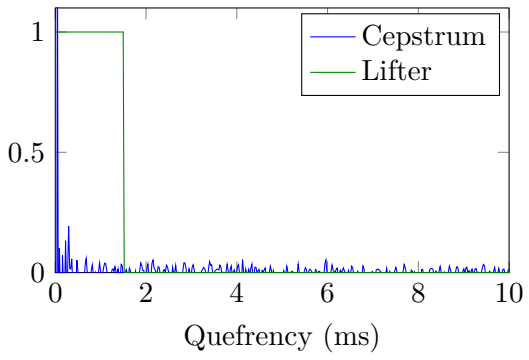
(E) Vocal tract and excitation spectrum (dB)

FIGURE 4.8: Liftering of a voiced speech segment and its smoothed spectrum

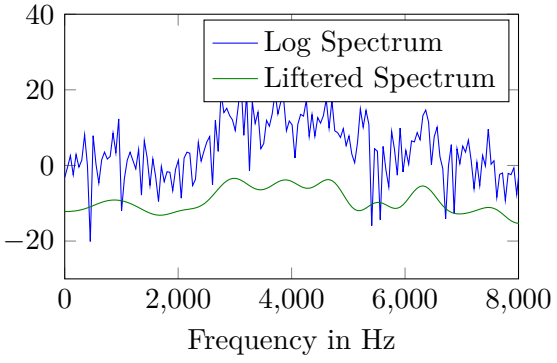
In figure 4.9 as an unvoiced sound is analyzed cepstrum shows no distinct peaks. Rapid variations in the spectrum can be observed as well, as a result of this unvoiced sound excitation. First liftering process involves a 1.5ms quefrency liftering capturing only slow spectral variations, and therefore obtaining the vocal tract formants. Nevertheless and opposed to voiced sounds, a higher liftering in the quefrency of 7.5ms does not show a more clear periodic structure in the spectral shape, concluding that there is no voiced periodic excitation involved in the segment under analysis.



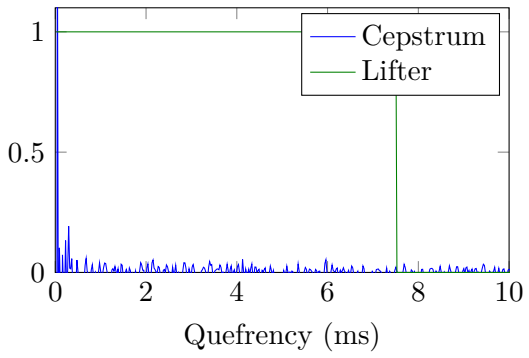
(A) Waveform



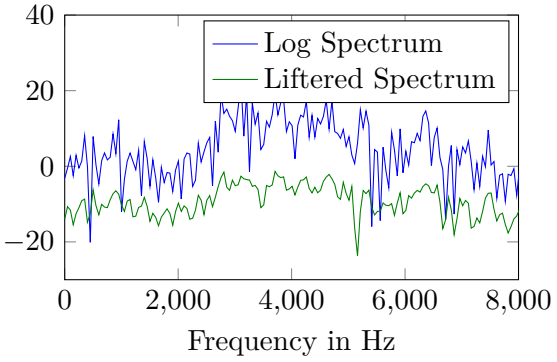
(B) Cepstrum liftering



(C) Vocal tract and smoothed spectrum (dB)



(D) Cepstrum liftering including peak



(E) Vocal tract and excitation spectrum (dB)

FIGURE 4.9: Liftering of an unvoiced speech segment and its smoothed spectrum

Pitch estimation

Since cepstrum depicts variations in the logarithmic spectrum, it constitutes a very effective technique to accurately determine pitch on speech signals. Nowadays, it still remains as one of the most effective indicators of voice pitch that have been devised. Pitch can be properly estimated in periodic and quasi-periodic signals, i.e., voiced sounds have a peak in a particular quefrency value depending on the vocal cords excitation period.

- For slow variations in the logarithmic spectrum, i.e., separated peaks indicating low frequencies, cepstrum shows a peak close to the origin pointing to the presence

of such low frequency.

- For rapid variations in the logarithmic spectrum, i.e., closer peaks indicating high frequencies, cepstrum shows a peak further from the origin pointing to the presence of such high frequency.
- For no variations in the logarithmic spectrum, i.e., random white noise, cepstrum does not have a clearly defined peak, pointing to the absence of harmonics and consequently pitch.

Accordingly, presence of a strong peak implies voiced speech and the quefrency location of the peak gives the estimation of the signal period. A formula to convert that peak position m to perceptible pitch in Hz would be

$$Pitch(Hz) = F_s \left[\frac{samples}{second}\right] \times \frac{1}{m} \left[\frac{1}{samples}\right] \tag{4.14}$$

This peak occurs in the cepstrum because harmonics in the spectrum are periodic, and this period corresponds to the signal pitch. Therefore, pitch of a pure sine wave with no harmonics could not be estimated by this method, as cepstrum pitch tracking is based on spectrum variations and not in spectrum peaks.

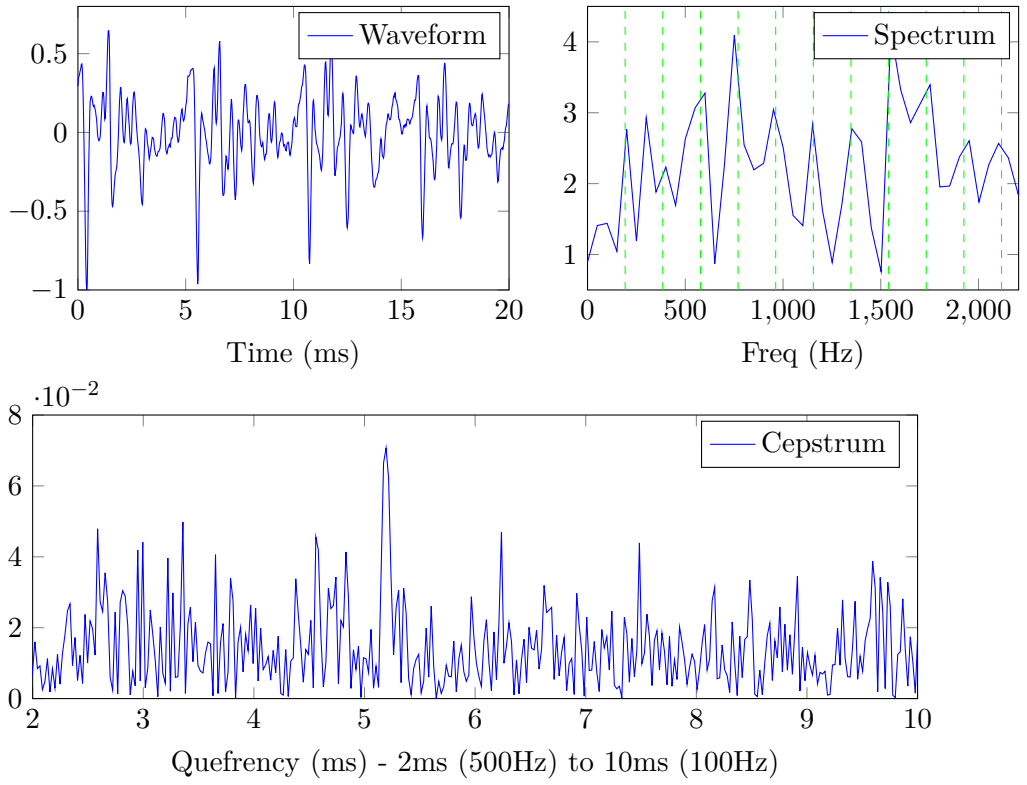
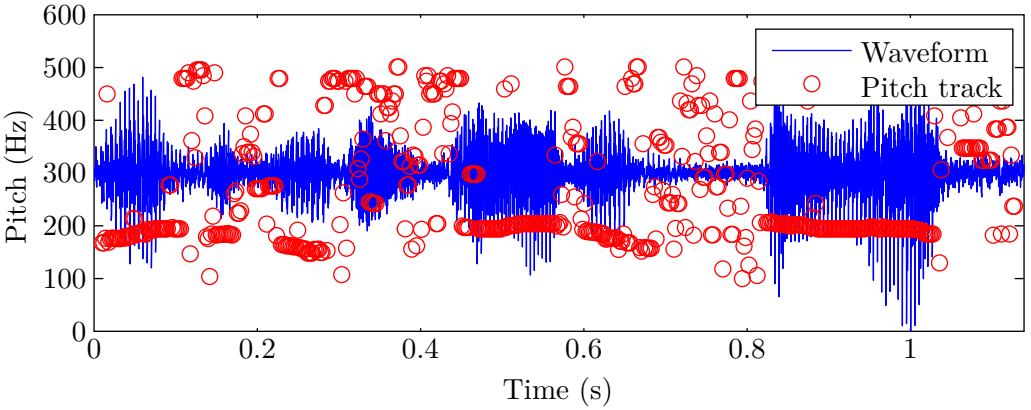


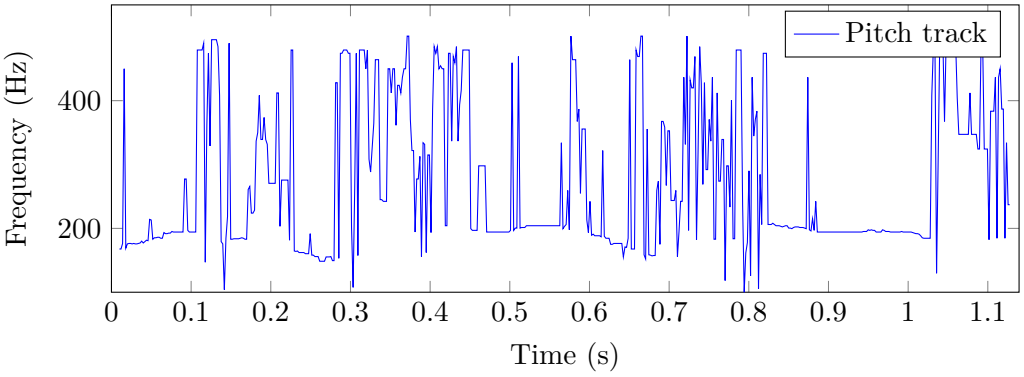
FIGURE 4.10: Waveform/Spectrum/Cepstrum relationship

However, voiced sounds contain harmonics and this technique can be used to determine speech pitch as it is shown in figure 4.10. This figure shows a 5,2ms periodic voiced sound. Thus, peaks on the spectrum ocurr every $1/5.2ms = 192Hz$ as it is shown in the spectrum of the segment. Finally, this 192Hz periodic structure is reflected by a 5,2ms peak in the quefrency domain.

An important consideration in cepstrum pitch estimation lies on the choice of which cepstrum range must be used. As it was mentioned before, DTFT sampling implies periodicity on the cepstrum. In this project, signal is segmented in 20ms windows, yielding a minimum detectable pitch of 100Hz as cepstrum is N-periodic, being N the length of the window and the amount of points on the DFT. Moreover, cepstrum maximum is always on the origin, as it occurs to the STACF, and a minimum value must be chosen, fixing a maximum detectable frequency. Therefore, cepstrum range extends from 2ms to 10ms, providing a detectable pitch range from 100Hz to 500Hz.



(A) Waveform of 'into the starting five'



(B) Pitch estimation between ms2 (500Hz) and ms10 (100Hz)

FIGURE 4.11: Pitch estimation of 'into the starting five'

In the figure above, the pitch of a speech segment containing the sentence 'into the starting five' is tracked. Figure 4.11 shows a clear constant pitch estimation of 180Hz-200Hz for voiced speech segments, while silences and unvoiced sections expose a blurry unclear pitch estimation.

Thereby, since words consist of both voiced and unvoiced sounds, the presence of a clear pitch suggests the existence of a commentator and aids in the discrimination of possible created keywords for highlight detection. Nevertheless, another feature is usually provided to classifiers instead of the whole cepstrum, as its involves too much data. Mel-Frequency Cepstral Coefficients are commonly used instead of cepstrum as they give an idea of its shape on a hearing adapted scale.

4.2 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCC) were introduced in the 1980's and still remain as one of the most important audio features in speech and speaker recognition systems. MFCCs are also increasingly finding uses in music information retrieval applications such as genre classification or audio similarity measures [6].

Each coefficient can be expressed as follows

$$c_n = \sqrt{\frac{2}{K}} \sum_{k=1}^{K} (\log S_k) \cos\left[\frac{\pi}{K}\left(k - \frac{1}{2}\right) n\right] \qquad n = 1, \ldots, L \tag{4.15}$$

where K represents the amount of filters used in the filter bank, S_k the logarithmic energy spanned by the k^{th} band and L is the amount of desired coefficients (normally $L < K$).

First, a pre-emphasis filter is applied to the audio signal in order to stress higher frequencies. Then a short-time Fourier analysis is performed, resulting in a DFT $X_n(k)$ for window n. Subsequently, $X_n(k)$ is filtered by an overlapping filter bank of K filters on the mel scale, and later the total energy included on each filter is calculated and normalized, generating K energy values corresponding to each filter. Finally, a discrete Cosine Transform is applied to those values, and the latest are discarded (only L coefficients are selected). In our case, one set of 23 MFCC coefficients is extracted for each frame.

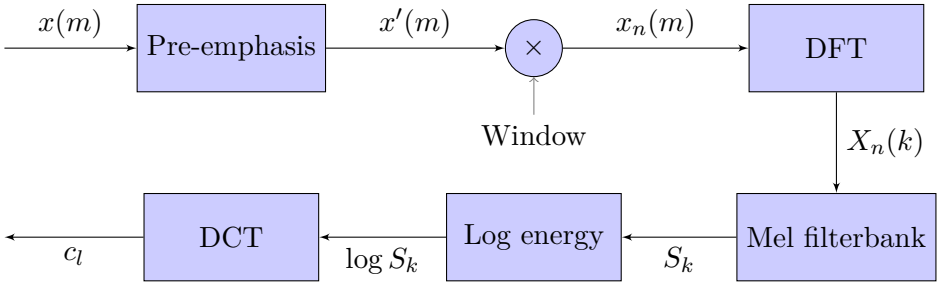


FIGURE 4.12: MFCC procedure scheme

All the implementation steps are detailed below.

Pre-emphasis filter

Pre-emphasis purpose is to emphasize the high energy part of the signal in order to model the perception of human hearing and speech. Since cepstrum measures the logarithmic spectrum periodic variations, an enhancement on high frequencies might yield a better distinction on cepstrum peaks, and thereby a more clear representation of coefficients.

Pre-emphasis filtering follows the next expression

$$x'(m) = x(m) - \alpha x(m-1) \tag{4.16}$$

where α represents the pre-emphasis coefficient. Higher values involve a stronger emphasis on higher frequencies. For our particular case a moderate $\alpha = 0.9$ is used.

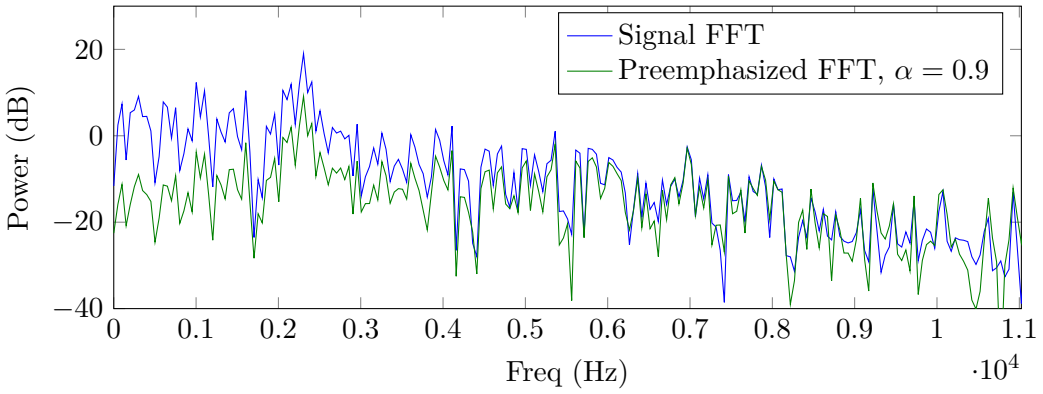


FIGURE 4.13: Preemphasis effect

Short-time analysis

Next step consists in signal windowing for a short-time analysis. In our case, a Hamming window with 50% overlapping is used in order to avoid abrupt changes between frames.

Then, the power spectrum of the window $X_n(k)$ is calculated by the FFT algorithm and used an in input for further steps.

Each of the following steps is then repeated for every window on the segment under analysis.

Mel-frequency filterbank

The mel-frequency filterbank is a filterbank of K filters following the next expression

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{4.17}$$

where m denotes the filter index, $f(m)$ is an array containing the sample index of computable frequencies for the filterbank, and k is the current frequency sample index. For example, first filter will extend from $f(0)$ to $f(2)$, reaching its maximum at $f(1)$.

This expression depicts a set of K triangular equally distanced in the mel-scale overlapping filters. Overlapping is done so that a filter will reach its maximum where the next one starts its pass band, and will come back to zero when next filter reaches its maximum. This is illustrated in figure 4.14.

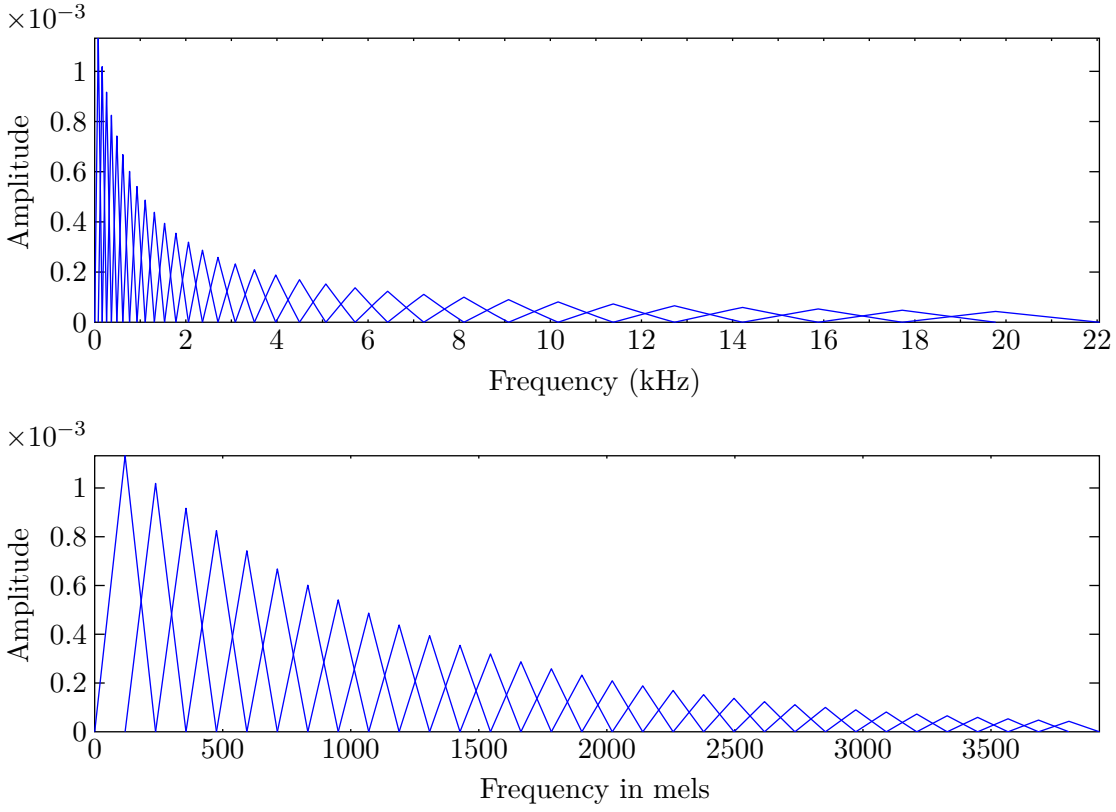


FIGURE 4.14: Mel-frequency triangular filterbank

First filters give an indication of how much energy exists at lower frequencies, and since human hearing is more sensitive to low frequencies they need more resolution and consequently are very narrow. As frequencies get higher filters become wider in view of the fact that high frequency variations are considered less important. Furthermore, filters of the filterbank are normalized. The reason why filters are scaled in frequency to sum to unity is that a perfectly flat input Fourier spectrum provides a flat mel-spectrum at the output of the filterbank.

The number of filters used in the filter bank is something important to consider. Commonly, a good estimation for the number of required filters depending on the sampling frequency Fs could be

$$N_{filters} = \lfloor 3\log(Fs) \rfloor \tag{4.18}$$

For speech purposes, $Fs = 8000Hz$ yields 26 filters and $Fs = 16000Hz$ gives 29 filters. For our particular purpose, where $Fs = 44100Hz$ a total amount of 32 filters is used.

Once the filterbank has been processed, energy spanned by each filter is estimated. To calculate filterbank energies each filterbank is multiplied with the power spectrum and subsequently coefficients on each band are added up, resulting in the vector S_k comprised of K filter energies.

Mel-scale adapts audible sound scale to human non-linear perception. Besides, the filterbank takes into account the effect that the cochlea can not discern between two closely spaced frequencies and includes it in the design of such filterbank, considering particular centers, bandwidths and amplitudes for each filter.

Logarithmic filterbank energies

After energies S_k on each filter have been calculated, the logarithm of each of them is performed, obtaining $\log(S_k)$. This is partly motivated by human hearing as loudness is not perceived on a linear scale. If the sound is loud enough, large variations in energy may not be accurately distinguished, thereby making features match more closely to human hearing by this compression operation.

In addition, the logarithmic spectrum allows the use of cepstral mean subtraction, which is a channel normalization technique. Moreover, it enables an easy removal of convolutional noise such as channel distortion and decreases the dynamic range of the output coefficient, in order to increase its robustness to noise.

Discrete Cosine Transform

The final step is to compute the discrete cosine transform (DCT) of the logarithmic filterbank energies so as to map them to the cepstral domain. There are 2 main reasons why DCT is performed instead of DFT.

First, since filters of the filterbank are all overlapping, filterbank energies are quite correlated with each other. The DCT decorrelates these energies, so that diagonal covariance matrices can be used to model features in a hiden Markov model (HMM) classifier. Secondly the DCT is used in many data compression applications because of a property that is referred to as "energy compaction". More specifically, the DCT-2 of a finite-length sequence often has its coefficients more highly concentrated at low indexes than the DFT does.

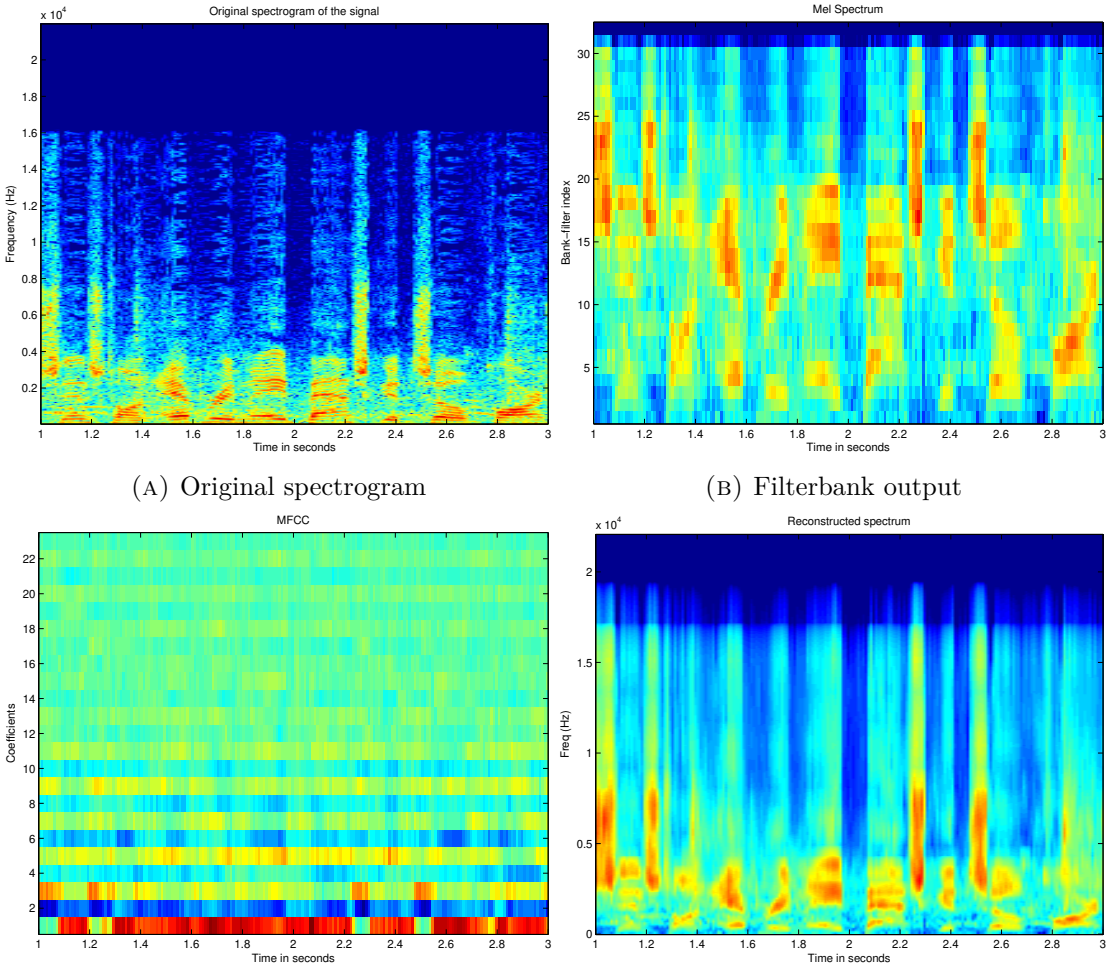
DCT Coefficients selection

That latter property of the DCT makes using all coefficients unnecessary as energy is compacted at low indexes. Therefore, only first L coefficients are required and are the resulting MFCC.

For automatic speech recognition (ASR) purposes, the power spectrum contains a lot of information not required and this effect becomes more pronounced as frequencies increase. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so a small improvement is achieved by discarding them. Commonly $L = 12$ is used for ASR.

According to equation 4.18, the chosen sampling frequency determines the amount of mel filters. Since only $L = 12$ are required in ASR, filters corresponding to high frequencies will be discarded. For example, for a $Fs = 16000Hz$, filter 12 frequency range extends from $1041mels$ to $1231mels$, which maps from $1063Hz$ to $1387Hz$. Similarly, if $Fs = 44100Hz$ is selected then filter 12 frequency range extends from $1308mels$ to $1546mels$, that is, from $1534Hz$ to $2059Hz$. Hence, rapid spectral variations are not represented on the MFCC.

Since MFCC in this project are not focused on ASR, a higher amount of coefficients is used. For $L = 23$ coefficients highest frequency corresponds to filter 23 at $2853mels$, which maps to $8100Hz$. This L provides a proper representation of the highest energy region of the spectrum under analysis on a human hearing adapted mel-scale cepstrum.



(A) Original spectrogram

(B) Filterbank output

(C) Mel-Frequency Cepstral Coefficients

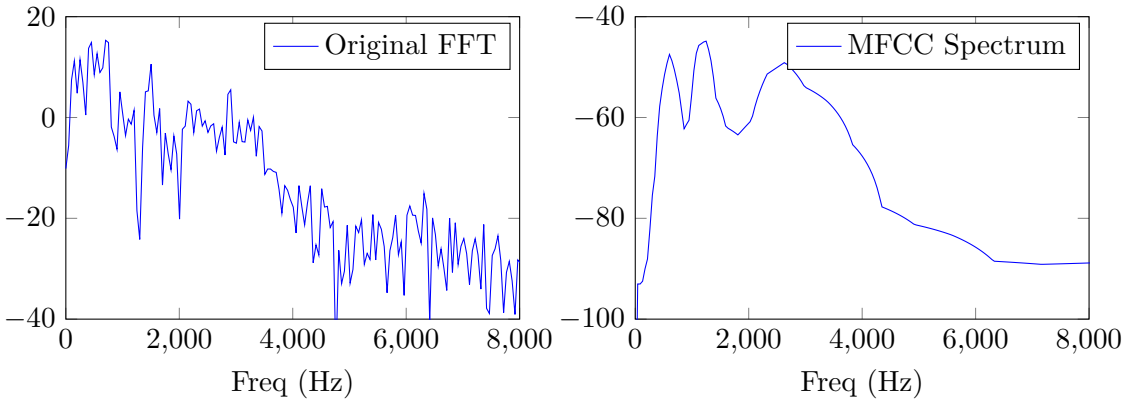
(D) Reconstructed spectrogram

FIGURE 4.15: MFCC Spectrum reconstruction with L=23 coefficients

Spectrum smoothing

Since MFCC are a cepstral representation, they depict variations in the filterbank logarithmic energies. Thus, lowest coefficients represent slow-varying energies in the spectrum whereas higher coefficients represent high-varying energies. The process of selecting only a particular number of coefficients after DCT is performed entails cepstral liftering of rapid spectral variations, and thereby the reconstructed spectrum will be a smoothed version of the original.

Considering the *source/system* speech model, and similarly to cepstrum liftering, this smoothed spectrum represents the vocal tract frequency response, while rapid liftered logarithmic variations represent the excitation. As a consequence, MFCC provides a good method to estimate parameters of the speech model with a low number of coefficients.



(A) Window FFT (dB) (B) Reconstructed spectrum (dB)

FIGURE 4.16: Smoothing effect of the reconstructed spectrum from MFCC

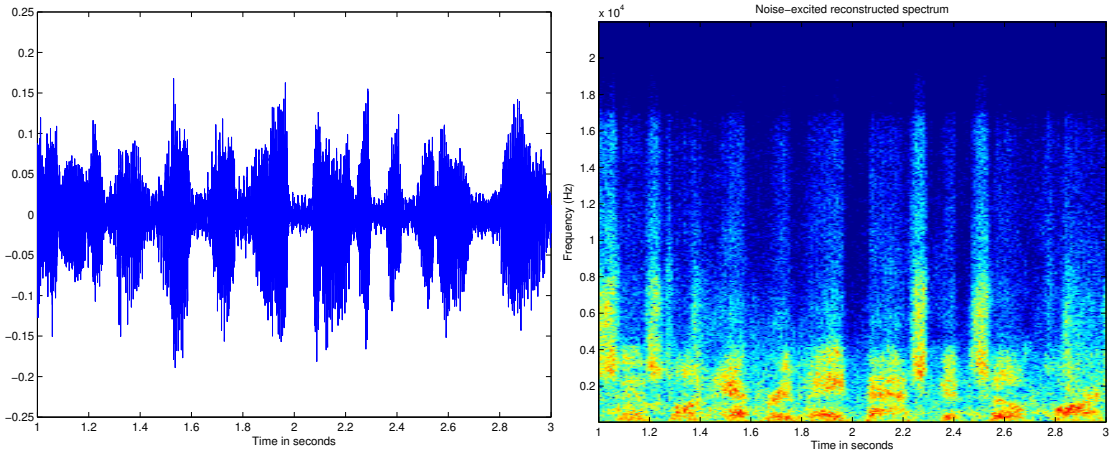
In figure 4.16, both spectra are different although they have in common peaks at the formant resonances of the vocal tract. Moreover, at higher frequencies the reconstructed mel-spectrum has more smoothing due to the structure of the filterbank.

To obtain the reconstructed spectrum the inverse procedure is carried out. An inverse DCT is performed to the current MFCC to obtain the logarithmic energy of each filter bank. Finally, a mapping from the mel-scale to the linear scale is made for both spectra comparison.

Furthermore, some methods of reconstructing a speech signal from a stream of MFCC vectors using a source/system model of speech production have been developed [7][8]. The MFCC vectors are used to provide an estimate of the vocal tract filter, and through noise excitation techniques, the original waveform and spectrum can be reconstructed.

Nevertheless, MFCC are not strictly speaking homomorphic filtering. That would be the case if the logical order of cepstral analysis was followed, i.e., taking the logarithm of the spectrum followed by filtering of energies and grouping into banks, instead of filtering into group energies and then calculate the logarithmic energy. In practice, however, the MFCC representation is approximately homomorphic for filters that have a smooth transfer function.

Besides, since MFCC is a cepstral representation it is robust against convolutional noise, which enables it to deal with a wide range of audio samples. Nonetheless, it is not very robust in the presence of additive noise, and so it is common to normalize their values in ASR to lessen the influence of noise. Some researchers propose modifications to the basic MFCC algorithm to improve robustness, such as by raising the logarithmic mel-amplitudes to a suitable power before taking the DCT, reducing the influence of low-energy components [9].



(A) Original Waveform (B) Noise excitation reconstructed spectrogram

FIGURE 4.17: MFCC Spectrum reconstruction with noise excitation

Seeing that the energy in a frame is also a valuable feature on speech analysis, logarithmic energy is commonly included along with the MFCC and its derivatives, delta and delta-delta coefficients, discussed in next section.

4.3 Delta coefficients, delta-delta coefficients

Delta cepstrum is simply the slope of a first-order polynomial fit to the cepstrum time evolution. Delta coefficients can be expressed as follows

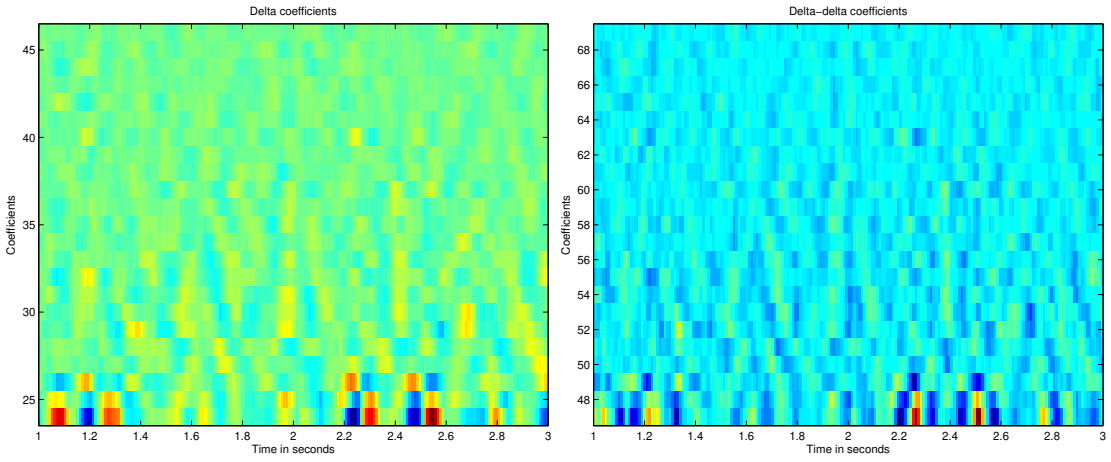
$$d_n = \frac{\sum_{i=1}^{N} i(c_{n+i} - c_{n-i})}{2\sum_{i=1}^{N} i^2} \tag{4.19}$$

where d_n is a delta coefficient computed in terms of the static coefficients c_n on frame n, i is the index and represents the time 'width' used to calculate deltas, and N corresponds to this maximum time 'width'. A typical value for maximum i is 2.

For $i = 1$, delta function simply calculates the difference between c_{t+1} and c_{t-1} and applies weight 1/2. On a next step it calculates the difference between c_{t+2} and c_{t-2} and applies weight 1/4. Therefore closest frames have the highest impact making $i = 2$ frames a reasonable choice for time analysis. Delta-delta coefficients follow the same scheme, although they are estimated from delta coefficients instead of from MFCC. These derivatives are also known as differential and acceleration coefficients respectively.

The MFCC feature vector describes only the power spectral envelope of a single frame and since speech also carries information in the dynamics, or trajectories of coefficients

over time, delta and delta-delta are usually appended to the feature vector in order to increase ASR performance.



(A) Delta coefficients (B) Delta-delta coefficients

FIGURE 4.18: Delta and delta-delta of a speech section

Furthermore, delta cepstrum can be applied to MFCC to remove the effects of linear filtering as long as the frequency response of the distorting linear filter does not vary much across each of the mel-frequency bands.

Chapter 5

Linear prediction

Linear predictive analysis is in essence a simple form of first-order extrapolation. If a signal has been varying at a certain rate it will probably continue to vary at approximately that same rate, at least in the short term. One important application of linear prediction is linear predictive coding (LPC), which can be used as a signal encoding compression technique.

Linear predictive coding is a tool used mostly in audio signal and speech processing for representing the spectral envelope of a digital signal of speech in a compressed form through the use of a linear predictive model. Nowadays it is one of the most powerful speech analysis techniques since it has been widely and effectively applied in almost every area of speech processing. Due to their efficiency, reliability and accuracy for speech, LPC methods are the most widely used in speech coding, speech synthesis, speech recognition, speaker recognition and verification and speech storage,

The importance of this method lies both in its ability to provide accurate estimates of the speech parameters and in its relative speed of computation. Linear prediction is based on the STACF and thus equivalently on the STFT, and even though separation techniques for the source/system speech model based on the cepstrum can be very effective, the linear predictive analysis methods discussed further have proven to be more efficient for a variety of reasons.

5.1 Linear Prediction Coefficients

According to the source/system speech model, and similarly to previously discussed cepstral speech processing, the primary object is to separate the excitation parameter from the vocal tract. In order to achieve this, the signal $s[n]$ is linearly predicted from

previous samples, obtaining $\tilde{s}[n]$ by using a prediction error filter with coefficients α_k. This idea is illustrated figure 5.1:

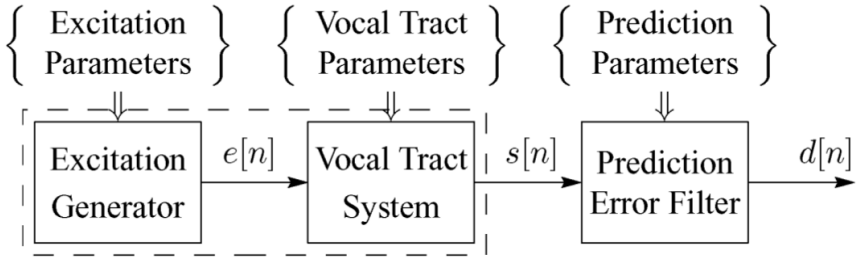


FIGURE 5.1: Model for linear predictive analysis of speech signals.

In linear predictive analysis the excitation is defined implicitly by the vocal tract system model, i.e., the excitation is whatever is needed to produce $s[n]$ at the output of the system. The major advantage of this model is that the gain parameter, G, and the filter coefficients α_k can be estimated in a simple and computationally efficient manner.

Some relationships must be satisfied between the error filter coefficients and the vocal tract transfer function. Finally, to obtain the values of those coefficients, the estimated error is minimized at the output of the error filter. Therefore, the source/system speech model is used again.

In summary, the target is to accurately estimate $s[n]$, that would be $s[n] = \tilde{s}[n]$. $H(z)$ is the z-Transform of the vocal tract and $A(z)$ the z-Transform of the error filter response.

The basic idea of linear prediction lies in considering that the current speech sample can be closely approximated as a linear combination of past samples. A predictor with p coefficients uses p previous samples to approximate the signal and is defined as

$$\tilde{s}[n] = \sum_{k=1}^{p} \alpha_k s[n-k] \longleftrightarrow \tilde{S}[z] = \sum_{k=1}^{p} \alpha_k z^{-k} \tag{5.1}$$

Besides, the output signal of the error filter is the prediction error $d[n] = s[n] - \tilde{s}[n]$, defined as the amount by which $s[n]$ fails to exactly predict current sample.

$$d[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \longleftrightarrow D(z) = S(z)\left[1 - \sum_{k=1}^{p} \alpha_k z^{-k}\right] \tag{5.2}$$

Considering previous equation 5.2, the z-Transform of the error filter is dependent on the prediction coefficients α_k.

$$A(z) = \frac{D(z)}{S(z)} = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{5.3}$$

If $H(z)$ is considered to be an all-pole filter with the G factor of the excitation source included, then it follows that

$$H(z)A(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \left(1 - \sum_{k=1}^{p} \alpha_k z^{-k}\right) \tag{5.4}$$

Finally, if the speech model exactly obeys the production model and $\alpha_k = a_k$, i.e., the vocal tract transfer function coefficients are equal to the prediction error coefficients, then $A(z)$ is an inverse filter for $H(z)$ and as a result the output of the error filter is the excitation source.

$$D(z) = E(z)H(z)A(z) = GE(z) \longleftrightarrow d[n] = Ge[n] \tag{5.5}$$

Hence, the parameters needed to obtain a useful estimation of the time-varying vocal tract system by separating the effects of excitation from the vocal tract filter are the excitation gain G and the set of predictor coefficients α_k. Following section provides a brief mathematical explanation to clarify two methods to properly estimate those coefficients.

5.1.1 Estimation of linear prediction coefficients

The basic approach is to find a set of predictor coefficients α_k that will minimize the mean-squared prediction error over a short segment of the speech waveform. The resulting parameters are then assumed to be the parameters of the system function $H(z)$ in the model for production of the given segment of the speech waveform. This process is repeated for short segments of speech periodically at a rate appropriate to track the phonetic variation (50-100 times per second).

First, the short-time average prediction error is defined as the difference between $s_n[m]$ and $\tilde{s}_n[m]$. Additionally, every subtraction must contribute to the overall sum by squaring them. Minimum prediction error assumes the estimated signal to be equal to the original.

$$E_n = \sum_m d_n^2[m] = \sum_m \left(s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k]\right)^2 \tag{5.6}$$

Values of α_k that minimize E_n can be found by setting $\partial E_n/\partial \alpha_i = 0$, obtaining the following equations:

$$\sum_{k=1}^{p} \alpha_k \left(\sum_m s_n[m-i] s_n[m-k] \right) = \sum_m s_n[m-i] s_n[m] \qquad 1 \leq i \leq p \tag{5.7}$$

Initially, only average of multiplications of delayed versions of the signal are necessary, yielding a linear system of equations for obtaining the coefficients. Moreover, if this averaging of delayed signals is rewritten as follows

$$\varphi_n[i,k] = \sum_m s_n[m-i] s_n[m-k] \tag{5.8}$$

Then equation 5.7 can be rewritten as

$$\sum_{k=1}^{p} \alpha_k \varphi_n[i,k] = \varphi_n[i,0] \qquad 1 \leq i \leq p \tag{5.9}$$

As it was mentioned before, $\varphi_n[i,k]$ consists in first delaying signal $s[m]$ i samples, then k samples, and multiplying both of them. Finally the sum of this multiplication is performed.

Therefore, a system of p equations in p unknowns is obtained. It can be solved for the α_k that minimize the squared error. This minimum mean-squared prediction error has the form

$$E_n = \sum_m s_n^2[m] - \sum_{k=1}^{p} \alpha_k \sum_m s_n[m] s_n[m-k] \tag{5.10}$$

According to the definition of φ_n, this equation can be rewritten as

$$E_n = \varphi_n[0,0] - \sum_{k=1}^{p} \alpha_k \varphi_n[0,k] \tag{5.11}$$

Computation of the quantities $\varphi_n[i,k]$ for $1 \leq i \leq p$ and $0 \leq k \leq p$ is first required to solve this linear system. Once this is done the matrix equation of p equations in p unknowns can be solved to obtain the coefficients α_k. In order to do so, it is necessary to specify the limits of summation m, to compute $\varphi_n[i,k]$, and to select the waveform segment $s_n[m]$.

Depending on previous considerations, two methods exist for solving this linear system: covariance and autocorrelation. A brief introduction to both and to their advantages and shortcomings are next given.

Autocorrelation

Autocorrelation method is the most widely used and is the one implemented in this project. In the autocorrelation method the signal is windowed by a tapering window in order to minimize discontinuities at the beginning and end of the interval. The beginning of such window predicts speech from zero-valued samples and the end predicts zero-valued samples from speech samples. This is illustrated by figure 5.2, where tapering window extends from $-M_1$ to M_2. The prediction error $d_n[m]$ is reduced due to that tapering.

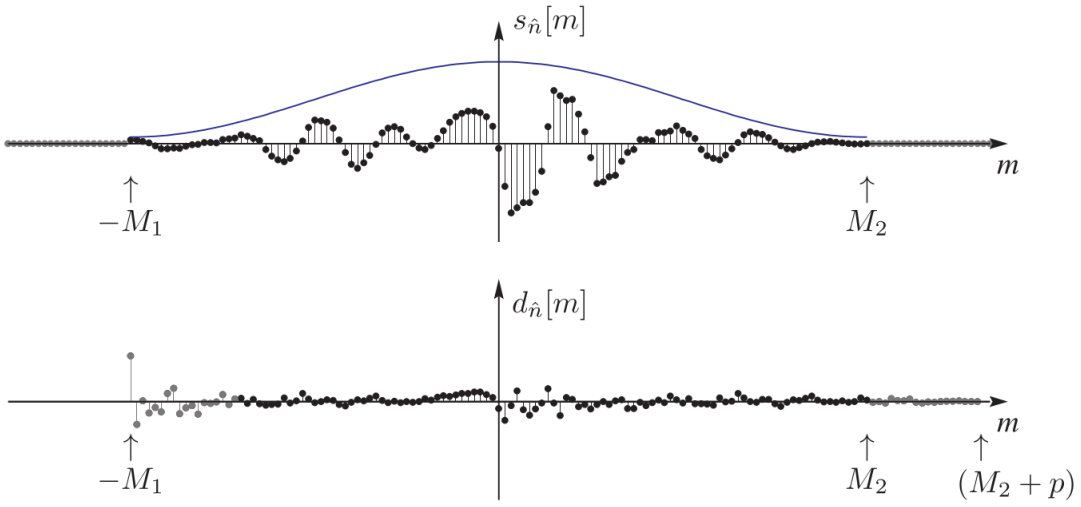


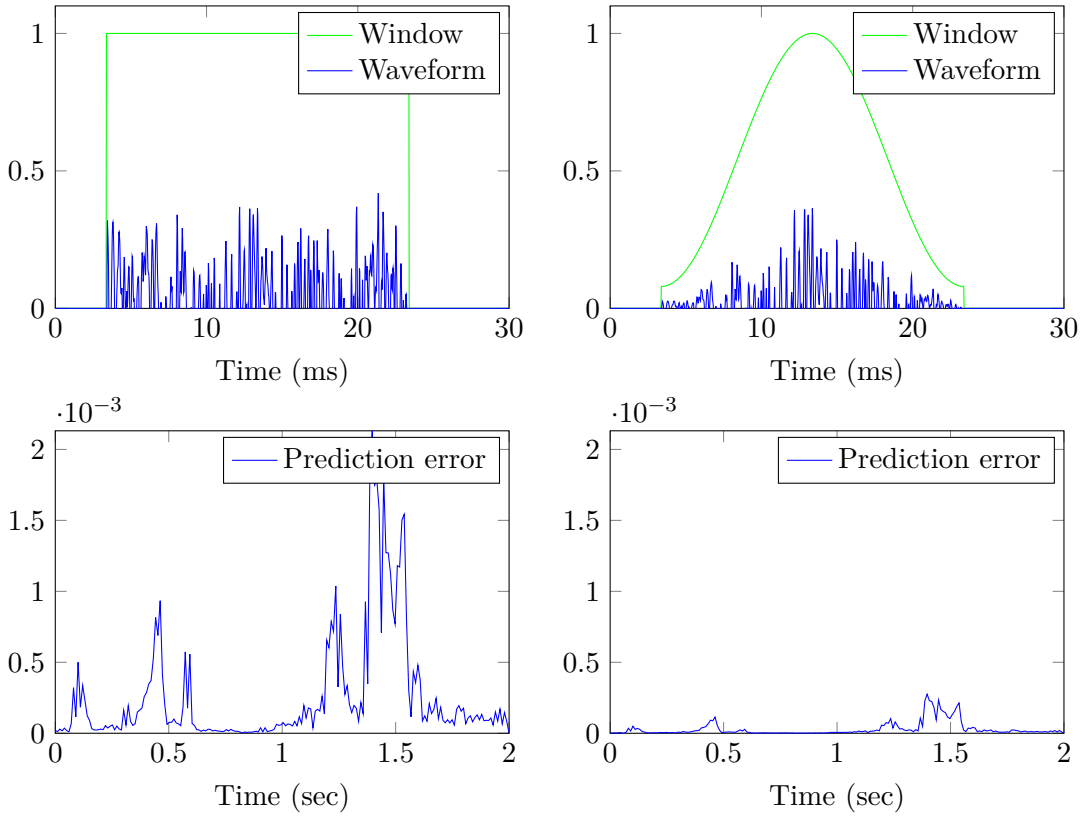
FIGURE 5.2: Autocorrelation method

An explicit comparison for error quantization between a tapering Hamming window and a rectangular window is carried out in figure 5.3. A 2 seconds excerpt is analyzed for both windows and the predicted error is proven to be much lower with a tapering window.

Moreover, the matrix $\varphi_n[i,k]$ is shown to be an autocorrelation function, and the resulting Toeplitz matrix can be easily solved. An algorithm called Levinson-Durvin recursion is implemented to solve this matrix equation efficiently. In addition, the roots of the prediction error filter $A(z)$ are guaranteed to lie within the unit circle of the z-plane and consequently the vocal tract model filter is stable.

The resulting set of equations for the optimum predictor coefficients is therefore

$$\sum_{k=1}^{p} \alpha_k \varphi_n[|i-k|] = \varphi_n[i] \qquad 1 \leq i \leq p \tag{5.12}$$



(A) Rectangular window

(B) Hamming window

FIGURE 5.3: Window tapering effect on the predicted error

However, with this method it is theoretically impossible for the error to be exactly zero since there will always be at least one sample at the beginning and one at the end of the prediction error sequence that will be nonzero.

Figure 5.3 shows the effect of window tapering on the resulting predicted error. As it was mentioned before, a Hamming window decreases considerably this predicted error while avoiding abrupt discontinuities in short-time analysis.

Covariance

In the covariance method the signal is extended by p samples outside the normal range of $0 \leq m \leq L-1$ to include p samples occurring prior to $m = 0$. This eliminates large errors in computing the signal from values prior to $m = 0$ as they are available, and eliminates the need for a tapering window. However, resulting matrix of correlations is symmetric but not Toeplitz, entailing a different solution method with a different set of optimal prediction coefficients α_k.

Unlike with autocorrelation method, with covariance it is theoretically possible for the average error to be exactly zero, and since the matrix is a symmetric positive-semidefinite

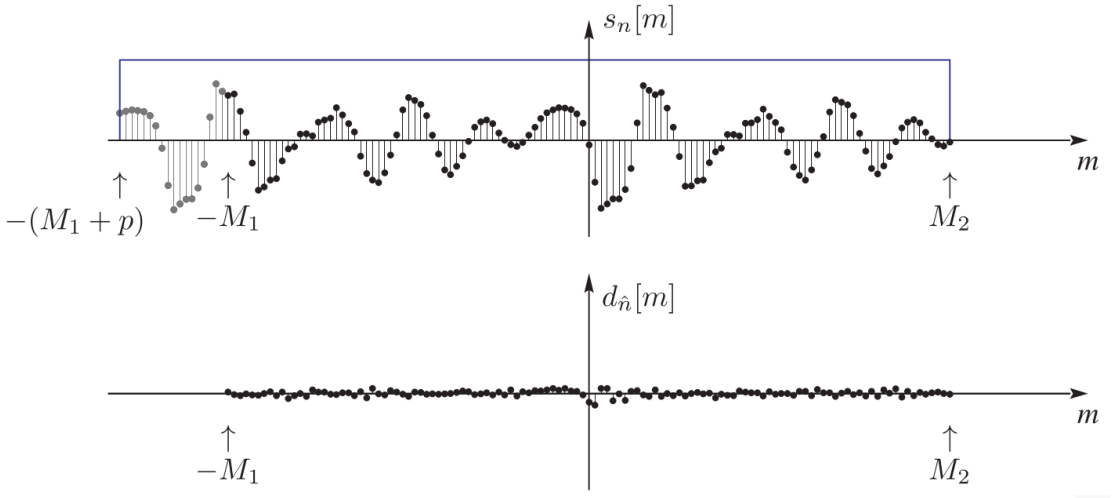


FIGURE 5.4: Covariance method

matrix it can be easily solved using the Cholesky decomposition. Nevertheless, the roots of the prediction error filter $A(z)$ are not guaranteed to lie within the unit circle of the z-plane, and thereby the vocal tract model filter is not guaranteed to be stable.

Gain parameter

Once linear prediction coefficients have been calculated, the gain parameter G is the remaining factor on the model to be estimated.

Since it is virtually impossible to guarantee that $\alpha_k = a_k$, it is not possible to match the signal energy to the energy of linearly predicted samples, which seems a reasonable method to determine the gain. Energy matching criterion is used instead, and it matches the energy in the error signal to the energy in the excitation.

$$G^2 \sum_{m=0}^{L-1+p} u^2[m] = \sum_{m=0}^{L-1+p} e^2[m] = E_n \tag{5.13}$$

Solving these equations for voiced sounds as quasi-periodic signals and unvoiced sounds as white random noise yields following equation

$$G^2 = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) = E_n \tag{5.14}$$

5.1.2 Linear Prediction Spectrum

After LPC analysis, we conclude that the coefficients α_k and the gain G determine how the structure of the vocal tract filter $H(z)$ is. Thereby, $H(z)$ would be

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{5.15}$$

And the magnitude of the frequency response of the vocal tract would be expressed as

$$|H(e^{j\omega})|^2 = \left| \frac{G}{1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k}} \right|^2 = \left| \frac{G}{A(e^{j\omega})} \right|^2 \tag{5.16}$$

LPC and the STFT are linked by the STAFC. LPC coefficients are calculated from the STACF, which is the IDFT of $|X_n(e^{j\omega})|$ of the windowed signal. Moreover, it is also related to cepstrum in the sense that rapid variations of the STFT, and therefore of the STACF, are due to the excitation while the overall shape is determined by the vocal tract transfer function.

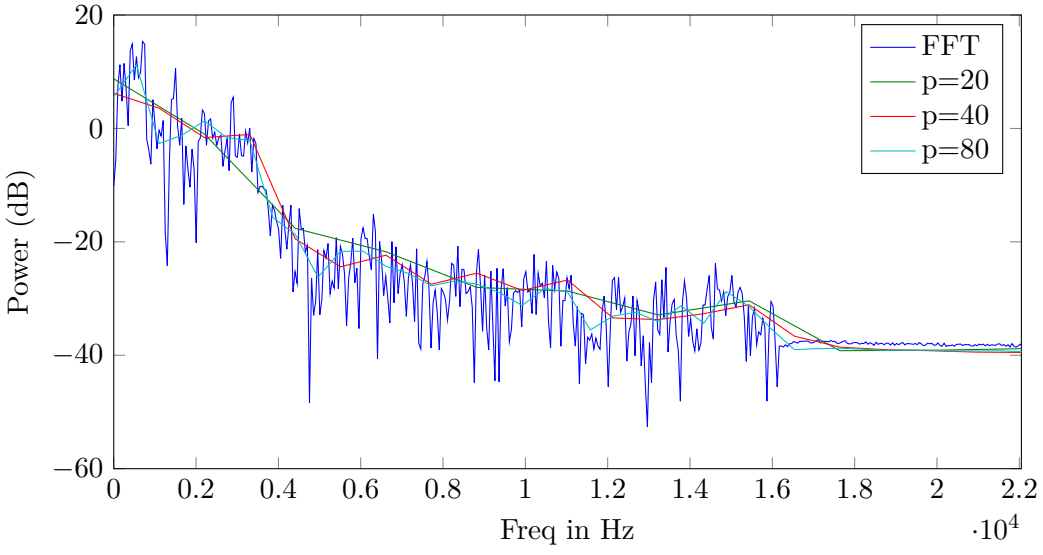
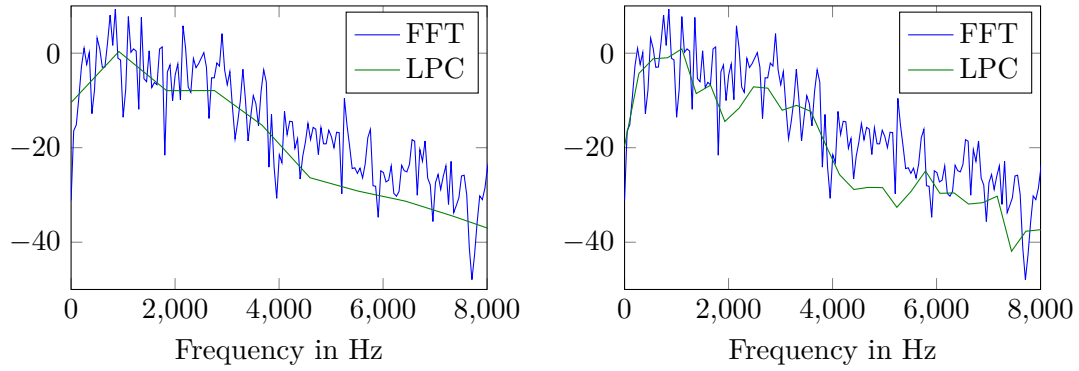


FIGURE 5.5: Influence of p on the vocal tract spectrum approximation

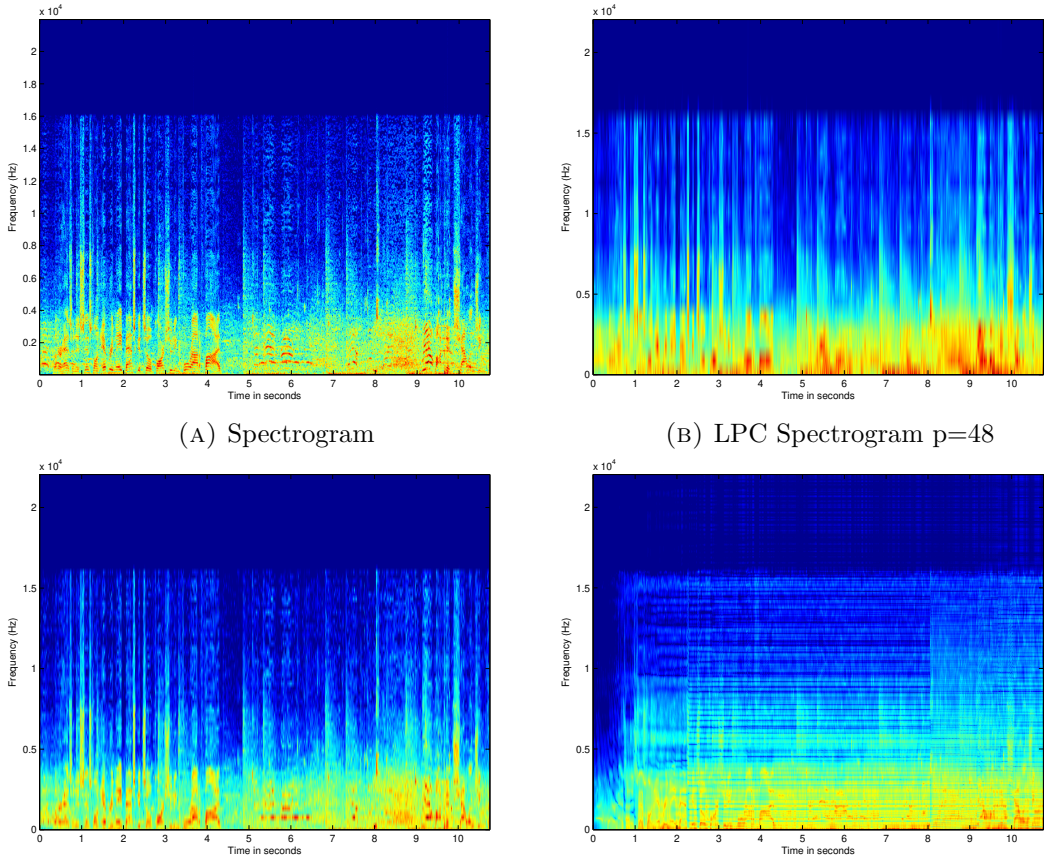
Analogously to cepstrum analysis, where the excitation effects can be removed by low-pass liftering, in linear predictive analysis the excitation effects can be removed by focusing on the low-time autocorrelation coefficients. The amount of smoothing of the spectrum is controlled by the choice of p, as it is shown in figure 5.5.



(A) p=48 coefficients (B) p=160 coefficients

FIGURE 5.6: FFT and LPC Spectrum comparison

If the STACF does not compute enough values to reach the peak corresponding to the excitation, the related spectrum will not depict those rapid variations. In a particular application, the prediction order is generally fixed at a value that captures the general spectral shape due to the vocal tract resonances. A good estimation for speech purposes is $p = 4 + F_s/1000$. For $F_s = 44100Hz$ we obtain $p = 48$ coefficients.



(A) Spectrogram (B) LPC Spectrogram p=48

(C) LPC Spectrogram p=160 (D) LPC smoothed Spectrogram p=48

FIGURE 5.7: Spectrogram and LPC Spectrogram of an audio segment

Noticeably, the linear predictive spectra tend to favor the peaks of the short-time Fourier transform. This is in contrast to homomorphic smoothing of the STFT, which tends toward an average of the peaks and valleys.

Linear prediction coefficients along with gain estimation are a technique to accurately represent speech spectrum, and therefore can be used instead of the FFT spectrum involving a much lower amount of data. This characteristic turns out to be very useful in the classifier stage, as its performance decreases with a large feature vector.

As a summary, it can be stated that linear prediction provides a robust, reliable and accurate method for estimating the parameters of the linear system for voiced speech. However, for other purposes different to speech, linear prediction has a much lower precision.

5.2 Linear Prediction Coefficients Entropy

Linear prediction coefficients are a polynomial approximation to the signal spectrum envelope, although they additionally eliminate the effect of noise. As a result, LPC spectrum representation is more stable than Fourier analysis representation. Hence, LPC coefficients are more suited to define the entropy measure. Linear prediction coefficients entropy (LPCE) is defined as follows

$$H = -\sum_{k=0}^{N-1} P_k \log P_k \tag{5.17}$$

where N represents the window length and P_k is defined as

$$P_k = \frac{|\alpha_n(k)|^2}{\sum_k |\alpha_n(k)|^2} \qquad k = 1, \ldots, p \tag{5.18}$$

where $\alpha_n(k)$ represents the coefficient k in the n-th window. Therefore, the procedure is the same as with spectral entropy although using linear prediction coefficients instead of Fourier coefficients.

In [10], LPC entropy is used as a temporal feature on larger windows in order to identify cheering audio sections. By looking at large windows of a sound track it is found that the spectrum of cheering sound is almost constant. Since this property is distinct for cheering sound and does not exist in speech sound, cheering detection might be better than speech detection for highlight classification.

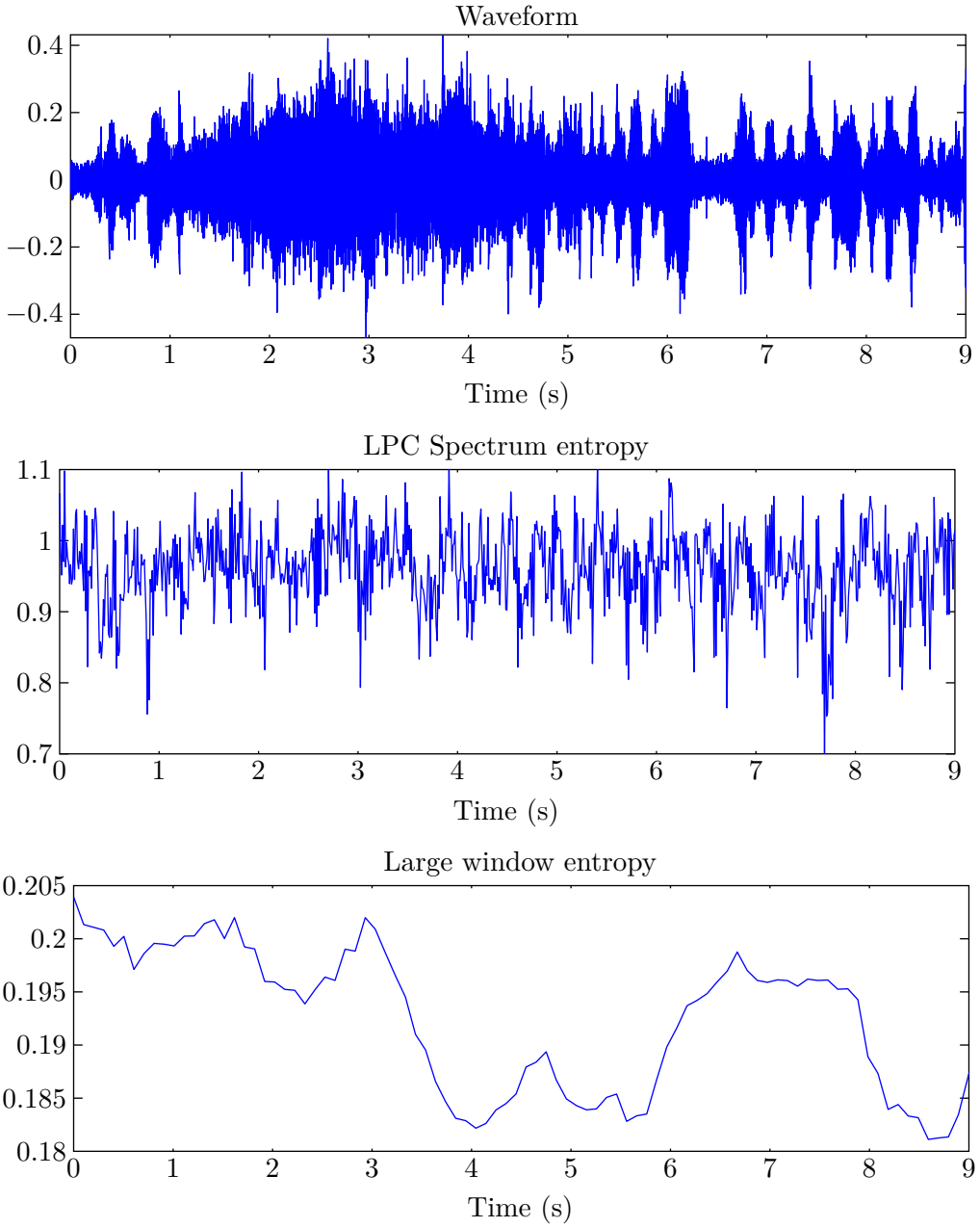


FIGURE 5.8: LPCE of a cheering/speech segment

The approach followed in [10] consists in estimating the LPC entropy of each coefficient in the temporal domain on a large window. Hence, here entropy does not depict the information coding cost. It is only a measure for signal stationarity, i.e., it measures spectral variations in the temporal domain.

In this case, the signal stationary measure in the time domain is defined as the average entropy of each LPC component:

$$H = \frac{1}{D} \sum_{d=1}^{D} H_d \tag{5.19}$$

where D represents the LPC order and H_d is the temporal LPCE of coefficient d on the large window n, i.e.

$$H_d = -\sum_{n=0}^{N-1} P_{dn} \log P_{dn} \tag{5.20}$$

and P_{dn} depicts the percentage of total energy that lies on coefficient d, i.e.

$$P_{dn} = \frac{|\alpha_n(d)|^2}{\sum_k |\alpha_n(d)|^2} \qquad k = 1, \ldots, D \tag{5.21}$$

With this approach LPC coefficients are estimated on smaller 20 ms windows, whereas their entropy is calculated in the time domain for larger windows n for each coefficient. Figure 5.8 shows results for a cheering/speech section.

5.3 Linear Prediction Cepstral Coefficients

Linear prediction coefficients are a very accurate method for speech parameters representation. However, LPC coefficients are too sensitive to quantization and since cepstrum has a number of advantages, such as source/filter separation, compactness and orthogonality, it is often desirable to transform these coefficients a_n into the cepstral domain, i.e., estimate cepstral coefficients c_n.

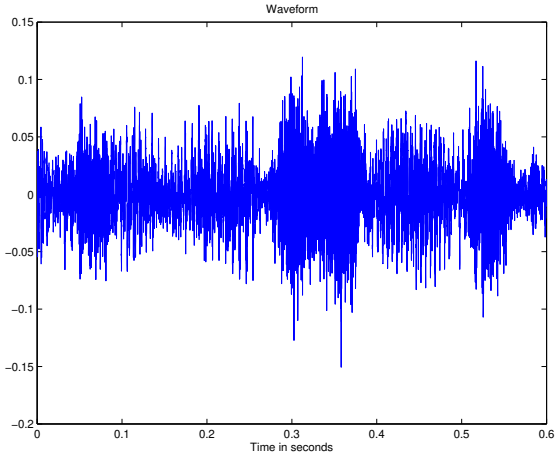
LPC-derived cepstral coefficient (LPCC) can be calculated by the following recursion [11]

$$c_n = \begin{cases} \ln(G) & n = 0 \\ -a_n + \frac{1}{n}\sum_{i=1}^{n-1} [-(n-i)a_i c_{n-i}] & n = 1, \ldots, p \end{cases} \tag{5.22}$$

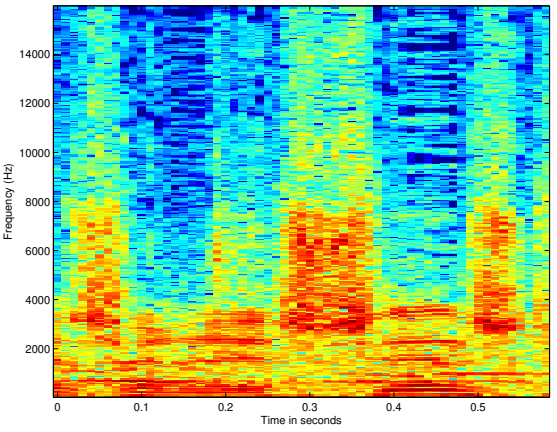
where a_i and a_n are the LPCs and p is the number of cepstral coefficients.

Figure 5.9 shows a speech signal containing voiced and unvoiced phonemes. Frequency content of voiced and unvoiced sounds is clearly reflected on the LPCC and compared to MFCC.

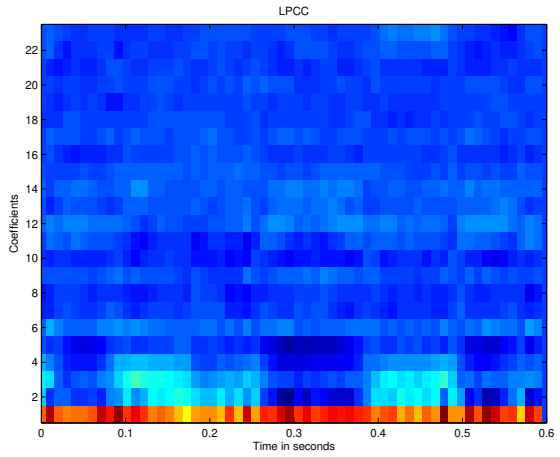
One of the main advantages of LPCC is that it is decorrelated so that a diagonal covariance could be used in an SVM classifier stage to model statistical properties of the signals.



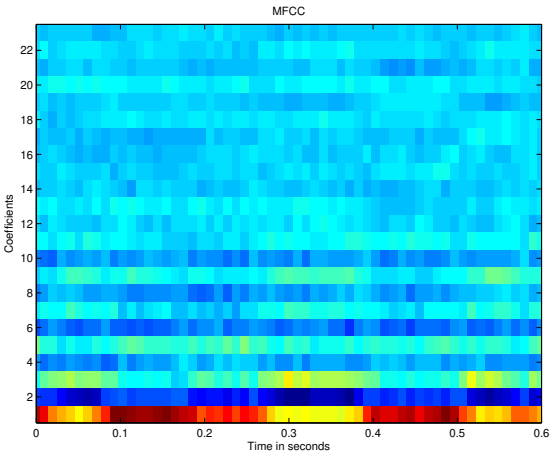
(A) Waveform



(B) Spectrogram



(C) LPCC



(D) MFCC

FIGURE 5.9: LPCC and other representations of an audio signal

5.4 Line Spectral Pairs

Since we often wish to quantize the model parameters for efficiency in storage or transmission of coded speech, a variety of different equivalent representations of LPC coefficients are used. These different representations are important, particularly when they are used in speech coding. An alternative is the line spectral pairs (LSP). LSP have several characteristics that make them more appropriate for direct quantization of LPCs, such as greater interpolation properties and robustness to quantization. For this reason, LSPs are very useful in speech coding.

The key idea of LSP is to decompose the p-th order linear predictor $A(z)$ into a symmetrical and antisymmetrical part denoted by the polynomials $P(z)$ and $Q(z)$ respectively. As a result, the LSP parameters are expressed as the zeroes (or roots) of $P(z)$ and $Q(z)$.

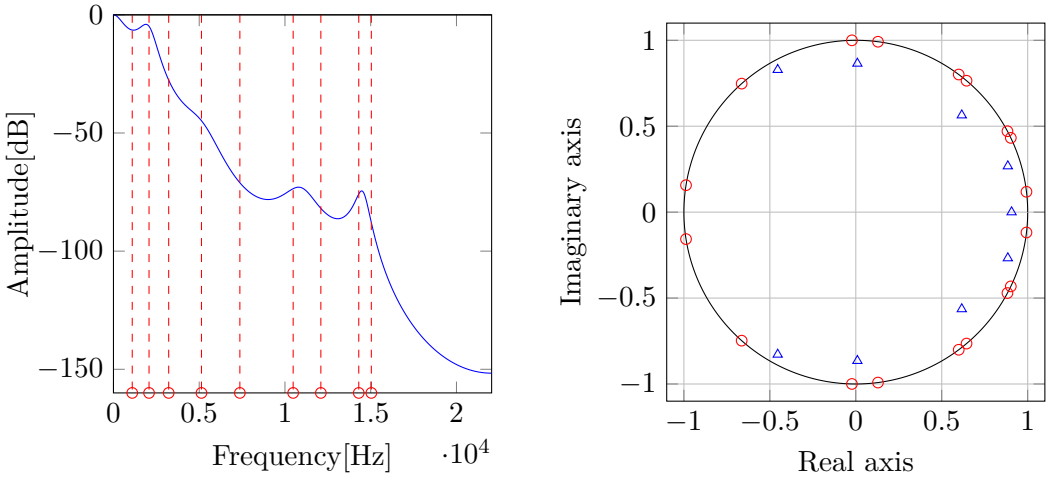
The LSP polynomials are defined as

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1})$$
$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1})$$

This polynomial representation of linear prediction coefficients has some particular properties:

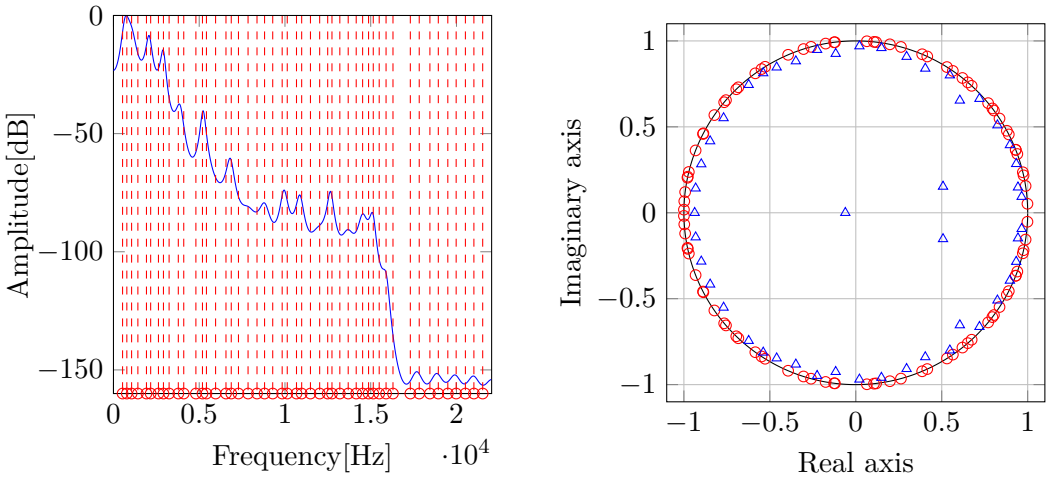
- The roots of $P(z)$ and $Q(z)$ lie on the unit circle in the complex plane.
- The roots of $P(z)$ and $Q(z)$ alternate on the unit circle if $A(z)$ is minimum phase, i.e. $A(z)$ has all its zeros within the unit circle.
- If p is an even integer, then $P(-1) = 0$ and $Q(1) = 0$.
- The LSP are close together when the roots of $A(z)$ are close to the unit circle. Therefore, the closer two roots are, the more resonant the filter is at the corresponding frequency.
- As the coefficients of $P(z)$ and $Q(z)$ are real, the roots occur in conjugate pairs.

All these properties make it possible to represent the linear predictor by quantized differences between the successive line spectral pairs. Since LSPs are not excessively sensitive to quantization noise and stability is easily ensured, LSP are widely used for quantizing LPC filters.



(A) Frequency response of vocal tract filter (B) Z-Diagram

FIGURE 5.10: LSF of a window with p=9



(A) Frequency response of vocal tract filter (B) Z-Diagram

FIGURE 5.11: LSF of a window with p=48

Due to the fact that all zeroes are located on the unit circle it is only necessary to specify the angle w in order to represent the LSP. If LSP is expressed in terms of the angular frequency the solutions are named line spectrum frequencies (LSF). The LSFs coefficients are commonly the prefered feature vectors used in vector quantization and are represented with a number between [0 π] or normalized as [0 1].

As a summary, line spectrum pairs is a robust alternative of uniquely representing the LPC coefficients, although its benefits are obtained at the cost of higher complexity of the overall system.

Chapter 6

Wavelet analysis

A wavelet $\psi(t)$ is a finite duration waveform with an average value of zero and nonzero norm. Mathematically wavelet functions belong to a subspace of the space $L^1(R) \cap L^2(R)$ of measurable functions that are absolutely and square integrable:

$$\int_{-\infty}^{\infty} |\psi(t)|\, dt < \infty$$
$$\int_{-\infty}^{\infty} |\psi(t)|^2\, dt < \infty$$

Being in this space ensures that the conditions of zero mean and square norm unity can be formulated.

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \qquad \text{is the condition for zero mean}$$
$$\int_{-\infty}^{\infty} |\psi(t)|^2\, dt = 1 \qquad \text{is the condition for square norm one}$$

Therefore, wavelets can typically be visualized as a brief oscillation. They have specific properties that make them useful for signal processing as they can be combined with segments of a known signal to extract information from an unknown signal.

In most situations it is useful to restrict $\psi(t)$ to be a continuous function with a higher number M of vanishing moments, i.e. for all integer $m < M$

$$\int_{-\infty}^{\infty} t^m\, \psi(t)\, dt = 0 \tag{6.1}$$

Similarly to Fourier analysis, the wavelet transform measures similarity between a signal and an analyzing function through a mathematical tool called inner product. However,

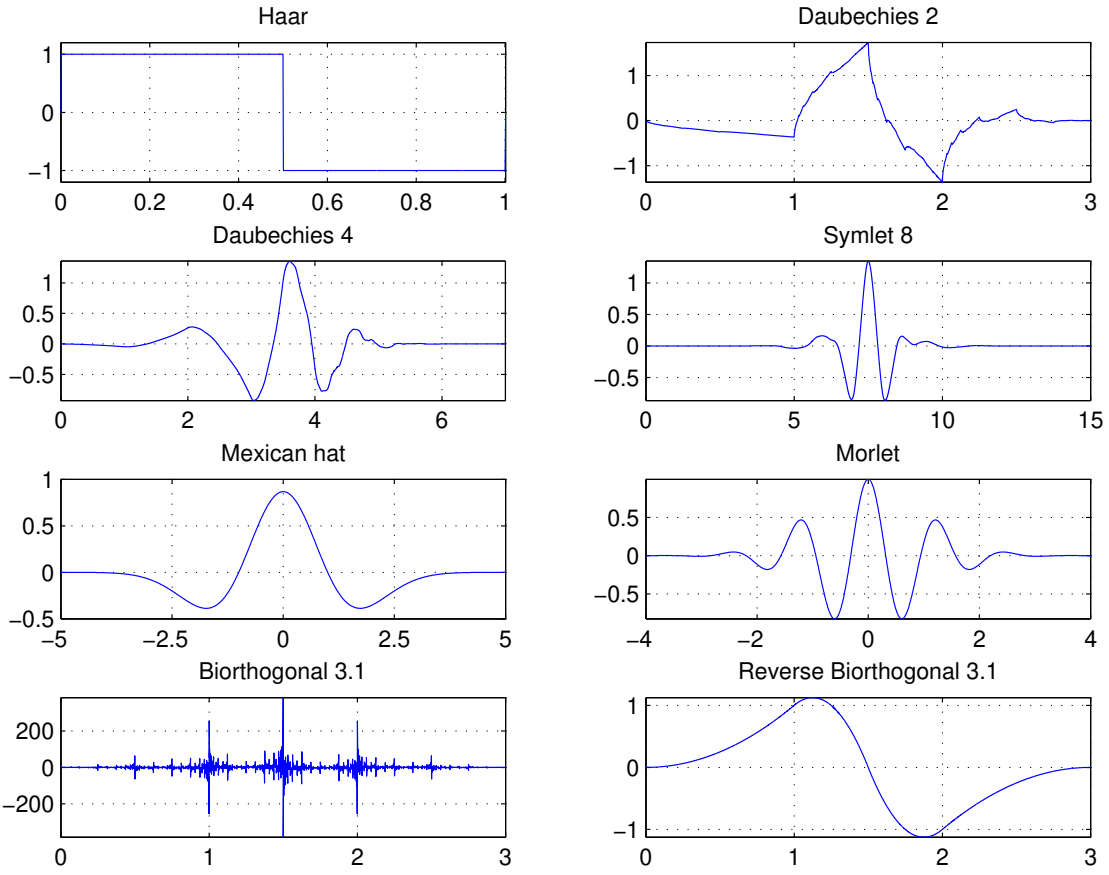


FIGURE 6.1: Various wavelet families

these two transforms differ in their choice of the analyzing function and consequently they provide different representations of the signal.

On Fourier analysis the signal is divided into sine waves of different frequencies while on wavelet analysis the signal is separated into shifted and scaled versions of a mother wavelet. This set of versions of the mother wavelet forms an orthonormal (typically, although not necessarily) family of basis functions.

This results in different information extracted from the signal. However, despite similarities in the procedure, sinusoids have infinite duration and are smooth and predictable whereas wavelets tend to be irregular and asymmetric. While sinusoids are useful in analyzing periodic and time-invariant phenomena, wavelets are well suited for the analysis of transient, time-varying signals.

$$X(\omega) = \langle x(t), e^{j\omega t} \rangle = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{6.2}$$

$$WT(a,b) = \langle x(t), \psi_{a,b}(t) \rangle = \int_{-\infty}^{\infty} x(t) \psi^*_{a,b}(t)\, dt. \tag{6.3}$$

where $\psi_{a,b}(t)$ is the analyzing function on wavelet analysis, a represents the scale and b the shifting.

The effectiveness of wavelet analysis lies in the use of many different families of admissible wavelets, depending on the signal features to be detected.

Moreover, the STFT has some important limitations concerning perception and computation of audio signals:

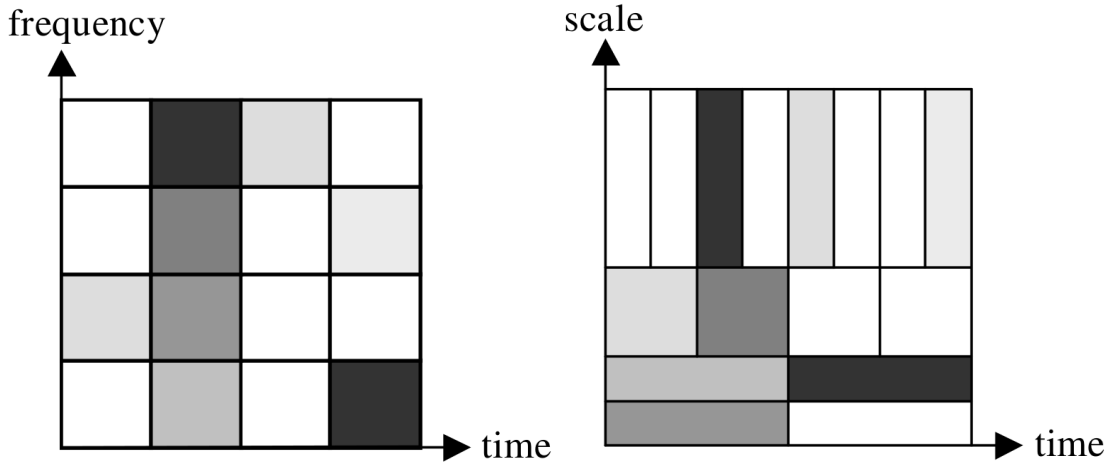
- The STFT produces equal-spaced frequency bands because of the chosen time window length, which is the same for every frequency. This does not correspond to human perception of frequencies. Therefore, an acceptable frequency resolution at low frequencies require an over-detailed high frequency resolution.
- Since digital audio data are not periodic or stationary, short-time and overlap techniques are used, resulting in less error at a computational cost increase.
- For a real time application of the STFT, a trade-off between time and frequency resolution must be accepted due to the uncertainty principle. It provides some information about both when and at what frequencies a signal event occurs. However, precision is determined by the size of the window.

Even though this last point of compromise between time and frequency information can be useful, many signals need a more flexible approach with a variable window size to determine more accurately either time or frequency. Audio signals are a good example due to human logarithmic perception of sound.

The Wavelet Transform was developed as an alternative to the STFT in order to overcome problems related to its frequency and time resolution properties. More specifically, unlike the STFT that provides uniform time resolution for all frequencies, the Wavelet Transform provides high time and low frequency resolution for high frequencies, and high frequency and low time resolution for low frequencies, making a fairly good compromise on the whole frequency scale. In addition, this approach resembles human ear time-frequency resolution characteristics [12].

Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, such as trends, breakdown points, discontinuities in higher derivatives and self-similarity. Furthermore, one major advantage of wavelets is the ability to perform local analysis, i.e., to analyze a localized area of a larger signal.

Some general applications of the Wavelet Transform include edge and corner detection on image processing, filter design, pattern recognition, music signal processing or economical data and temperature analysis. Moreover, since wavelet analysis employs a



(A) Time-frequency (B) DWT Time-scale

FIGURE 6.2: Spectrogram vs Scalogram

different representation of data than conventional techniques, it is often useful for signal de-noising or compression without appreciable degradation.

In this project, different implementations of the Wavelet Transform are carried out and detailed next.

6.1 Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) compares the signal to shifted and scaled versions of a mother wavelet $\psi(t)$. Scaling is achieved by stretching or compressing the analyzing function, and is sometimes referred to as dilation. Therefore, by comparing the input signal to the mother wavelet at various scales and positions a function of two variables is obtained. For a scale parameter, $a > 0$, and time shifting b the CWT is defined as

$$C(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t-b}{a} \right) dt \tag{6.4}$$

where $*$ denotes the complex conjugate, $x(t)$ is the input signal and $\psi\left(\frac{t-b}{a}\right)$ refers to the shifted and scaled versions of the mother wavelet.

By continuously varying the values of the scale parameter a and the position parameter b the CWT coefficients $C(a,b)$ are obtained. The scale parameter a is strictly positive and the shift parameter b can be any real number. For high values of a the mother wavelet has a lower frequency while for low values of a the mother wavelet has a higher frequency.

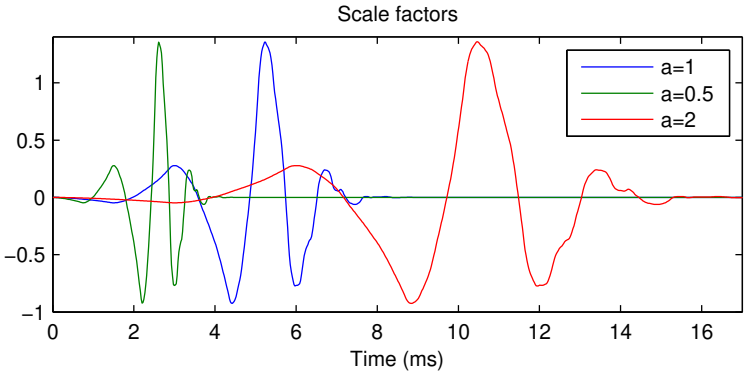


FIGURE 6.3: Scale effect on the mother wavelet

The scale factor is an inherently positive quantity $a > 0$. Its effect on wavelets is exactly the same as with sinusoids. The smaller the scale factor the more "compressed" the wavelet is and rapidly changing details are compared, i.e., high frequencies. On the other hand, the higher the scale the more "stretched" the wavelet is and the longer the portion of the signal with which it is being compared, i.e., the coarser the signal features measured by the wavelet coefficients. Hence, the scale factor a is inversely related to the angular frequency ω.

Even though there is a general relationship between scale and frequency, no precise relationship exists. A mapping between a wavelet at a given scale with a specified sampling period to a frequency in hertz can be done only in a general sense.

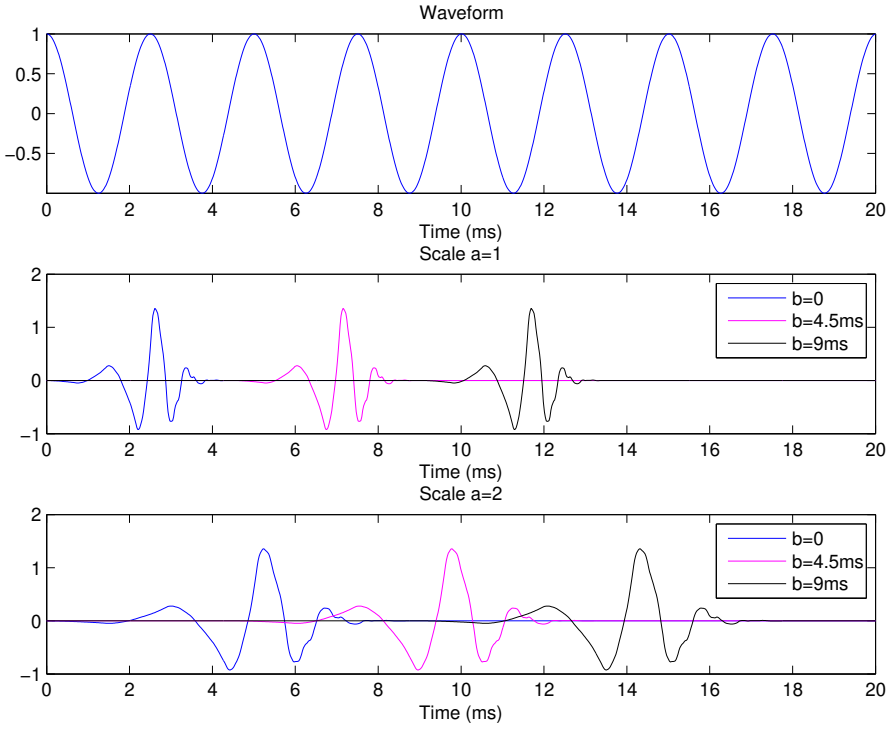
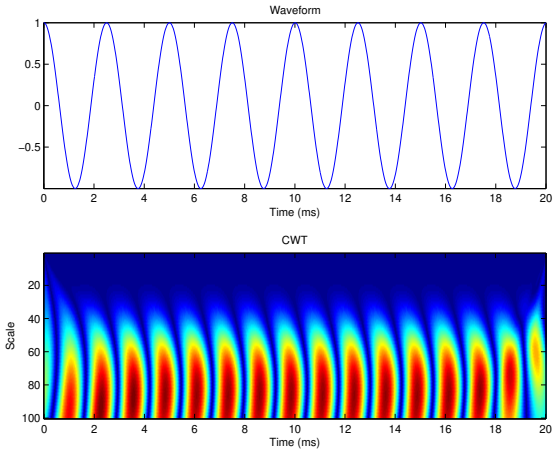
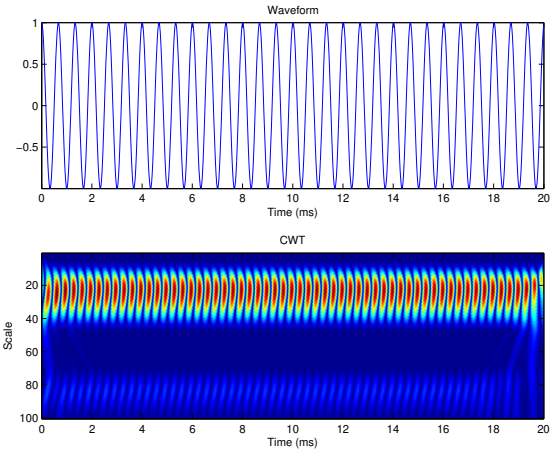


FIGURE 6.4: Scaling-shifting procedure

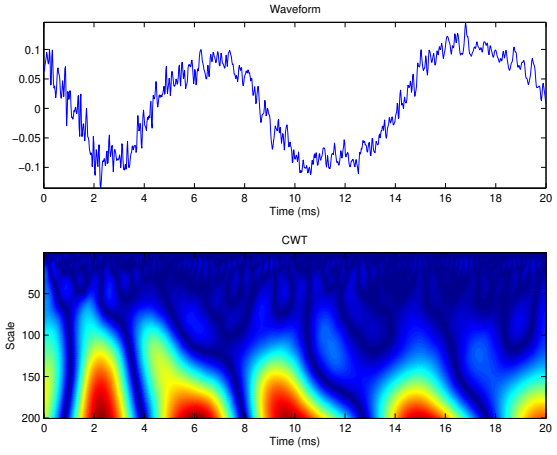
However, not only do the values of scale and position affect the CWT coefficients. They also depend on the analyzing signal employed for wavelet analysis. For a complex wavelet the CWT is complex-valued and for a real wavelet the CWT is real-valued. Compared to the STFT, the CWT substitutes the moving window function $w(n-m)$ for a scalable wavelet basis function $\psi(\frac{t-b}{a})$. Furthermore, these coefficients are a redundant two-dimensional representation of a one-dimensional signal.



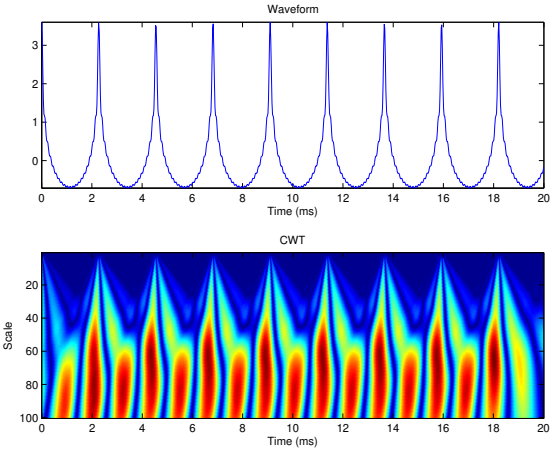
(A) 440 Hz sine wave



(B) 1500 Hz sine wave



(C) Low frequency with noise sine



(D) High harmonic energy sine

FIGURE 6.5: CWT Scalogram of various waveforms with 'db4' wavelet

Cooperated with a wavelet basis function, which can be scaled and shifted on the time scale, the CWT presents a non-uniform time and frequency resolution. Low time/high frequency resolution is shown for lower frequency signals and high time/low frequency resolution for high frequency signals. This phenomenon is illustrated in figure 6.5. Higher scales have a lower time resolution but represent better the frequency content of the signal. On the contrary, lower scales provide an accurate measure of temporal events with an inaccurate frequency representation. In addition, the CWT of a sine wave

shows an oscillatory pattern at scales where the oscillation in the wavelet approximates the period of the sine wave.

The CWT is a powerful tool for detecting discontinuities and abrupt changes in a signal as these produce large wavelet coefficients centered around the discontinuity at all scales. It is also useful for detecting smooth signal features, which produce large coefficients at scales where the oscillation in the wavelet correlates best with the signal feature. On the other hand, at scales where the oscillation in the wavelet occurs on either a much larger or smaller scale than the period of the sine wave the CWT coefficients are approximately zero.

Implementation

The CWT can be interpreted as a frequency-based filtering of the signal. If equation 6.4 is rewritten in terms of an inverse Fourier transform it can be expressed as

$$C(a,b) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\sqrt{a}\hat{\psi}^*(a\omega)e^{j\omega b}d\omega \tag{6.5}$$

where $X(\omega)$ and $\hat{\psi}(\omega)$ are the Fourier transforms of the signal and the mother wavelet respectively. As a result, the CWT can be efficiently computed through the use of the inverse Fourier transform.

Moreover, this previous expression depicts the CWT as a bandpass filtering of the input signal. If the term $\sqrt{a}\hat{\psi}^*(a\omega)$ is analyzed for different values of a, the following relationship is obtained:

- Low scale values increase the center frequency of the wavelet and its bandwidth while the amplitude decreases.
- High scale values decrease the center frequency of the wavelet and its bandwidth while the amplitude increases.

As a consequence, and as it was previously mentioned, CWT coefficients at lower scales represent the energy of the input signal at higher frequencies with a low frequency resolution, while CWT coefficients at higher scales represent energy of the input signal at lower frequencies with a high frequency resolution. This bandpass filtering effect is illustrated by figure 6.6.

Unlike Fourier bandpass filtering, the width of the bandpass filter in the CWT is inversely proportional to scale. This relationship follows from the uncertainty principle between

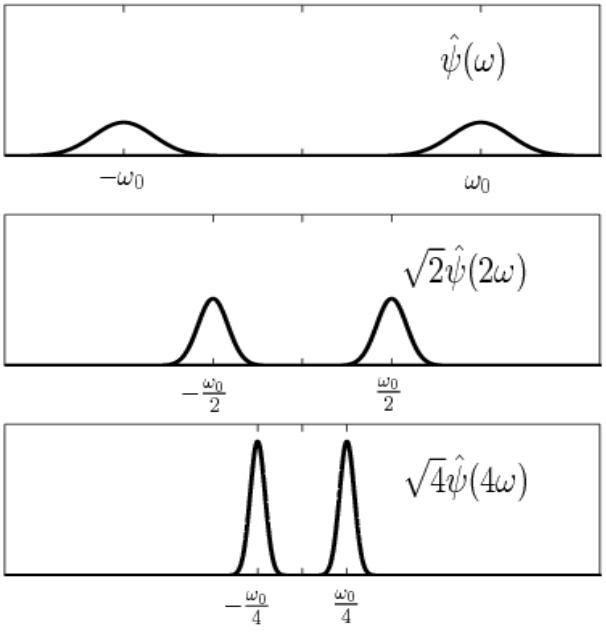


FIGURE 6.6: CWT as a filtering technique

time and frequency resolution of a signal. Furthermore, the Q factor of a filter can be seen as the ratio of its center frequency to its bandwidth. Therefore, since wavelets are referred to as constant Q filters, the analysis bands are thin for low frequencies and wide for high frequencies. This resembles more accurately human hearing perception of sound, since human ear has a similar frequency response as a constant Q filterbank, especially above 500Hz. Fourier transform, however, could be classified as a constant bandwidth transform.

Inverse Continuous Wavelet Transform

For wavelets satisfying the admissibility condition, the inverse CWT of a finite-energy function $x(t)$ can be defined as

$$x(t) = \frac{1}{C_\psi} \int_a \int_b \frac{1}{a^2} \langle x(t), \psi_{a,b}(t) \rangle \psi_{a,b}(t) da db \tag{6.6}$$

where C_ψ represents the admissibility condition, i.e.

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{6.7}$$

and $\hat{\psi}(\omega)$ is the Fourier transform of the mother wavelet used for CWT analysis. It is possible to show that the admissibility condition implies that $\hat{\psi}(0) = 0$, so that a wavelet must integrate to zero.

Since the CWT is a redundant transform there is not a unique way to determine its inverse. For analyzing wavelets and functions satisfying the following conditions a single integral formula for the inverse CWT exists:

- The analyzed function $x(t)$ is real-valued and the analyzing wavelet has a real-valued Fourier transform.
- The analyzed function $x(t)$ is real-valued and the Fourier transform of the analyzing wavelet has support only on the set of non-negative frequencies. This is referred to as an analytic wavelet. A function whose Fourier transform only has support on the set of non-negative frequencies must be complex-valued.

The CWT operates over every possible scale from that of the original signal up to some maximum scale determined by trading off a detailed analysis with computational complexity. In terms of shifting, the analyzing wavelet is shifted smoothly over the full domain of the analyzed function. Hence, calculating wavelet coefficients at every possible scale and translation is a considerable amount of work and generates an extensive quantity of data.

As it was mentioned before, the CWT implies a redundant two-dimensional representation of a one-dimensional signal. Therefore, a specific subset of scale and translation values exists, yielding the Discrete Wavelet Transform (DWT). DWT approximation is commonly used in engineering and computer science whereas CWT is mostly used in scientific research.

6.2 Discrete Wavelet Transform

Wavelet analysis can be much more efficient and just as accurate as continuous analysis by simply choosing scales and positions based on powers of two, usually referred to as dyadic analysis. It yields the Discrete Wavelet Transform (DWT), defined by the following equation [13]

$$W(j,k) = \sum_j \sum_k x(k) 2^{-j/2} \psi(2^{-j} n - k) \tag{6.8}$$

where $\psi(n)$ represents the discrete-time mother wavelet, j is the scale parameter and k the translation.

The DWT is a special case of the CWT that provides a compact representation of a signal in time and frequency. It is a relatively recent and computationally efficient technique for extracting temporal and spectral information about non-stationary signals such as audio.

The DWT has a huge number of applications in science, engineering, mathematics and computer science. Most notably, it is used on signal coding for representing a discrete signal in a more redundant form, often as a preconditioning for data compression.

Implementation

The DWT can be implemented using a hierarchical algorithm proposed by Mallat [14] and yields a fast wavelet transform. In wavelet analysis, the terms "approximations" and "details" are often used. Approximations are the high-scale and low-frequency components of the signal, whereas details are the low-scale and high-frequency components.

This previous hierarchical implementation consists of two different stages: analysis for approximation and detail coefficients extraction, and synthesis for signal reconstruction from these previous coefficients. Both are next detailed.

Analysis

DWT analysis or decomposition can be performed using a fast, pyramidal algorithm related to multirate filterbanks. As a multirate filterbank the DWT can be viewed as a constant Q filterbank with octave spacing between the centers of the filters. Each subband contains half the samples of the neighboring higher frequency subband due to downsampling.

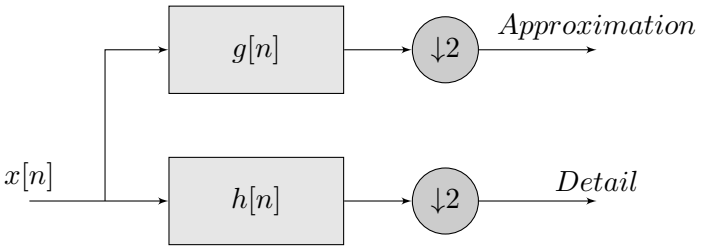


FIGURE 6.7: One-stage decomposition

In the pyramidal algorithm the signal is analyzed at different frequency bands with different resolution by decomposing the signal into a coarse approximation and detail

information. The coarse approximation is then further decomposed using the same wavelet decomposition step so that one signal is separated into many lower resolution components. This is called the wavelet decomposition tree and is achieved by successive highpass and lowpass filtering of the time domain signal. It is defined by the following equations

$$y_{high}[k] = \sum_n x[n]h[2k-n] \tag{6.9}$$

$$y_{low}[k] = \sum_n x[n]g[2k-n] \tag{6.10}$$

where $y_{high}[k]$, $y_{low}[k]$ are the outputs of the highpass ($h[n]$) and lowpass ($g[n]$) filters, respectively after downsampling by 2. Because of the downsampling the number of resulting wavelet coefficients is exactly the same as the number of input points.

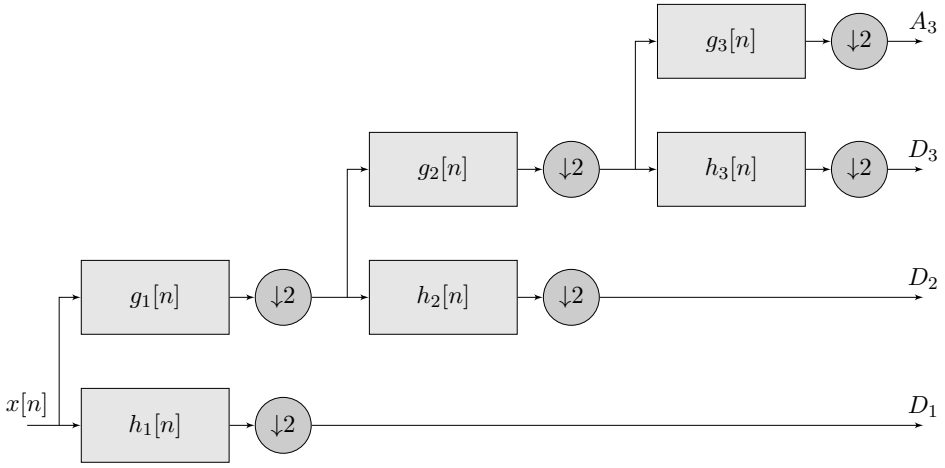


FIGURE 6.8: Wavelet decomposition tree

The detail coefficients are small compared to the approximation coefficients and consist mainly of high-frequency noise. On the other hand the approximation coefficients contain much less noise than the original signal due to the low-pass filtering effect.

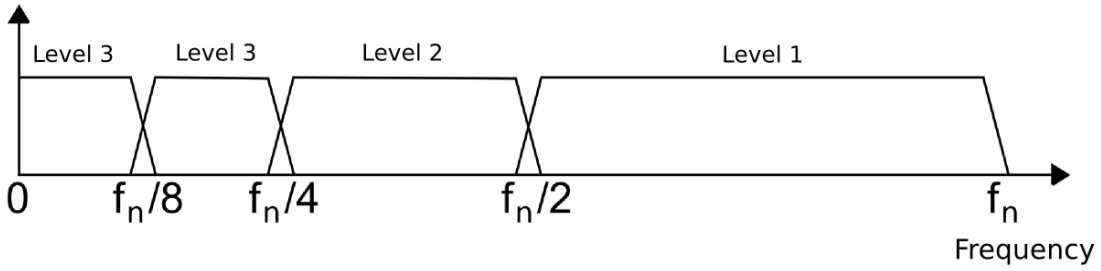
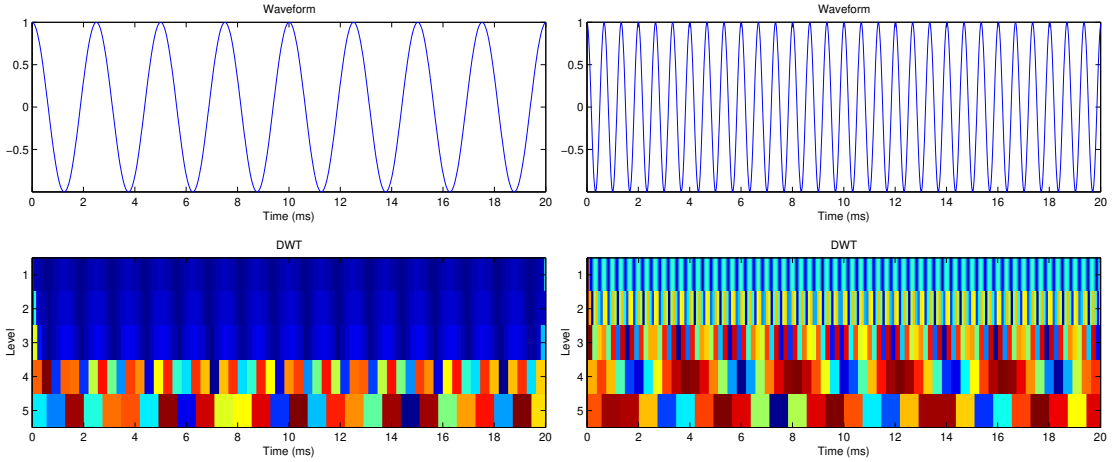
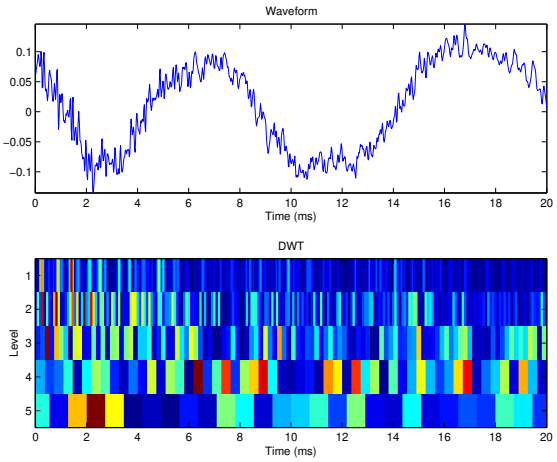


FIGURE 6.9: 3 Levels decomposition filter structure

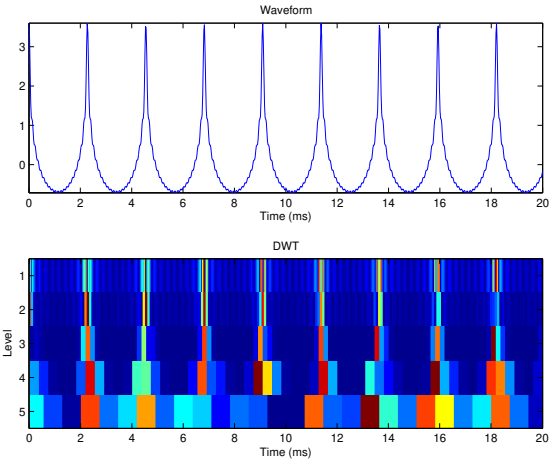


(A) 440 Hz sine wave

(B) 1500 Hz sine wave



(C) Low frequency with noise sine



(D) High harmonic energy sine

FIGURE 6.10: 4 Levels DWT Scalogram of various waveforms with 'db4' wavelet

Figure 6.10 shows a 4 level DWT, performed on 4 different audio signals. As it can be perceived, high levels represent low frequency components with an insufficient temporal resolution, as it happened to the CWT.

Since the analysis process is iterative, in theory it can be continued indefinitely. However, the decomposition can proceed only until the individual details consist only of a single sample.

Synthesis

The process of obtaining the original signal from the approximation and detail coefficients without loss of information is called reconstruction or synthesis. Mathematically, synthesis corresponds to the inverse discrete wavelet transform (IDWT).

Synthesis procedure is similar to analysis in the sense that it is implemented by a set of hierarchical highpass and lowpass multirate dyadic filters from the approximation and detail coefficients. For one-stage synthesis the scheme is detailed below

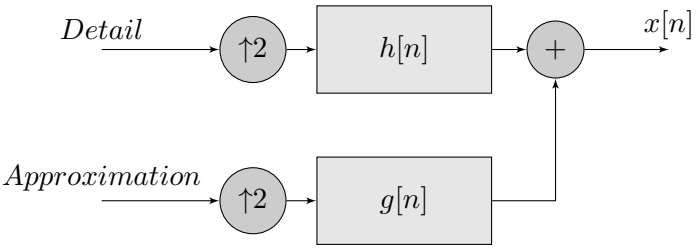


FIGURE 6.11: One-stage synthesis

Upsampling must be performed in order to undo decomposition decimation. As a result, the number of points of the reconstructed signal is the same as the number of wavelet coefficients. In addition, two synthesis filters are needed to smooth the zeros corresponding to the upsampling process.

Moreover, the downsampling performed during the analysis process introduces aliasing to the signal. An appropriate choice of the decomposition and reconstruction filters allows to avoid it. However, there are many aspects to consider so as to achieve the perfect reconstruction property for a filterbank. In fact, the choice of filters not only determines whether perfect reconstruction is possible, it also defines the shape of the wavelet used to perform the analysis.

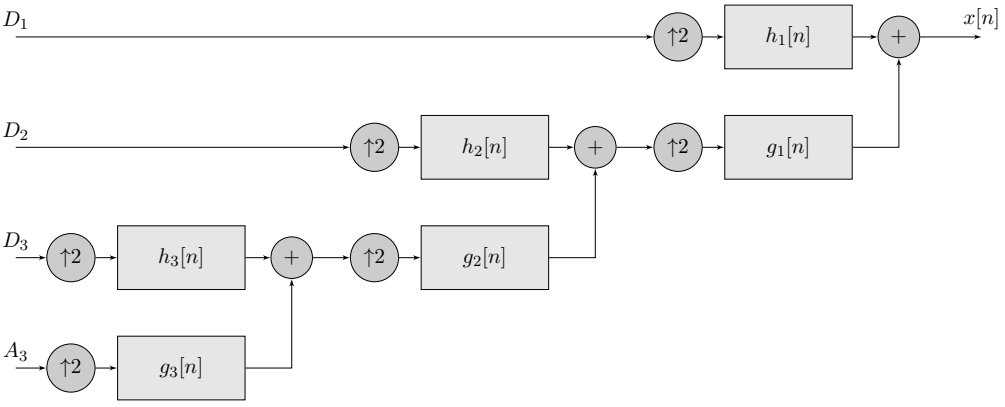


FIGURE 6.12: 3 Levels synthesis

The wavelet function ψ is determined by the high-pass filter, which also produces the details of the wavelet decomposition. There is an additional function associated with some, but not all, wavelets. It is called scaling function, ϕ. It is very similar to the wavelet function and is determined by the low-pass filters. The scaling function is associated with the approximations of the wavelet decomposition. Hence, while upsampling and convolving the high-pass filter produces a shape approximating the wavelet function

ψ, iteratively upsampling and convolving the low-pass filter produces a shape approximating the scaling function ϕ.

However, the DWT suffers an important drawback due to the non-linear processes involved in its implementation. DWT is not a time-invariant transform because of downsampling and upsampling, i.e., the DWT of a translated version of a signal is not, in general, the translated version of the DWT of such signal. In order to overcome this issue, the stationary wavelet transform is presented next.

6.3 Stationary Wavelet Transform

The Stationary wavelet transform (SWT) is a wavelet transform algorithm designed for overcoming the lack of translation-invariance of the DWT. It may be used as a signal-processing tool for visualization and analysis of non-stationary, time-sampled waveforms [15]. The highly desirable property of shift invariance can be obtained at the cost of a moderate increase in computational complexity, and accepting a least-squares inverse "pseudoinverse" in place of a true inverse.

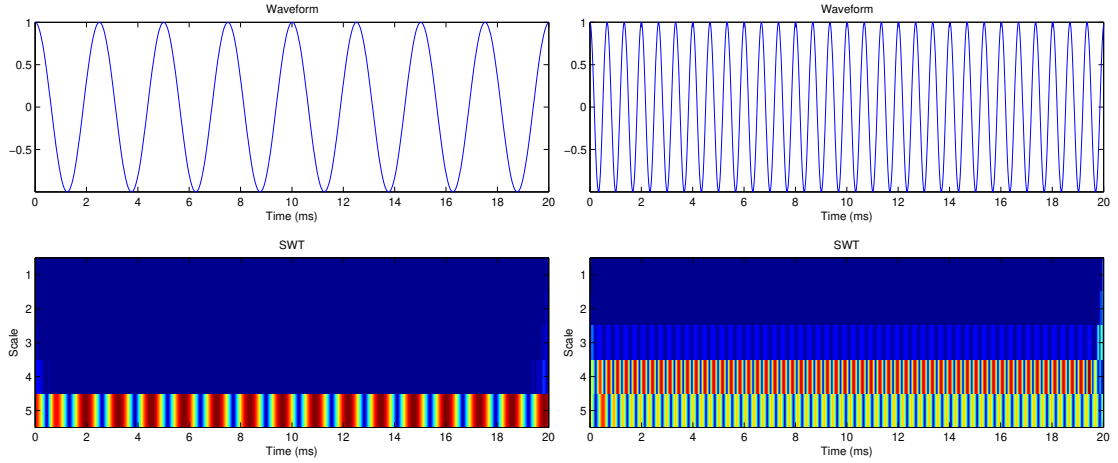
The basic idea for restoring the translation-invariance is to average different DWT implementations called ϵ-decimated DWT. The DWT basic computational step is a convolution followed by a decimation which only retains even indexed elements. That decimation could be carried out by choosing odd indexed elements instead of even indexed elements and this choice concerns every step of the decomposition process. As a result, 2^j possible decompositions, or ϵ-decimated DWT, exist for a given maximum level j.

Let us denote $\epsilon_j = 0$ for even indexed elements and $\epsilon_j = 1$ for odd indexed elements at step j. Each decomposition is labeled by a sequence of 0s and 1s where $\epsilon = \epsilon_1, \ldots, \epsilon_j$ for a j level decomposition. This transform is the previously mentioned ϵ-decimated DWT. Hence, depending on the choice of indexing even or odd elements at step j a different ϵ-decimated DWT is obtained.

The main application of the SWT is de-noising by the average of several de-noised signals, each of them obtained by using the usual de-noising scheme but applied to the coefficients of an ϵ-decimated DWT. Another important application of the SWT, such as the one concerning this project, is pattern recognition.

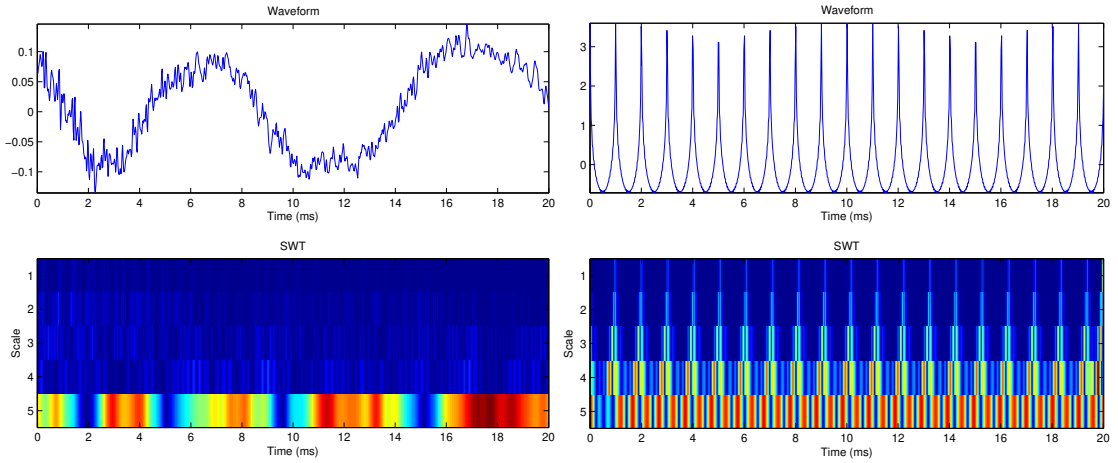
Implementation

The general step j convolves the approximation coefficients at level $j-1$ with upsampled versions of the appropriate original filters in order to produce the approximation



(A) 440 Hz sine wave

(B) 1500 Hz sine wave



(C) Low frequency with noise sine

(D) 1 KHz High harmonic energy sine

FIGURE 6.13: 4 levels SWT Scalogram of various waveforms with 'db4' wavelet

and detail coefficients at level j. Thus, SWT implementation consists in removing the downsamplers and upsamplers in the DWT, and upsampling the filter coefficients by a factor of $2^{(j-1)}$ in the j-th level of the algorithm. This is illustrated on figure 6.14 for a 3 levels implementation.

As a result, the SWT has an inherently redundant scheme since the output of each level of the SWT contains the same number of samples as the input, so for a decomposition of j levels there is a redundancy of j on the wavelet coefficients.

Analogously, the idea of the inverse discrete stationary wavelet transform is to average the inverses obtained for every ϵ-decimated DWT. This can be done recursively, starting from level j up to level 1.

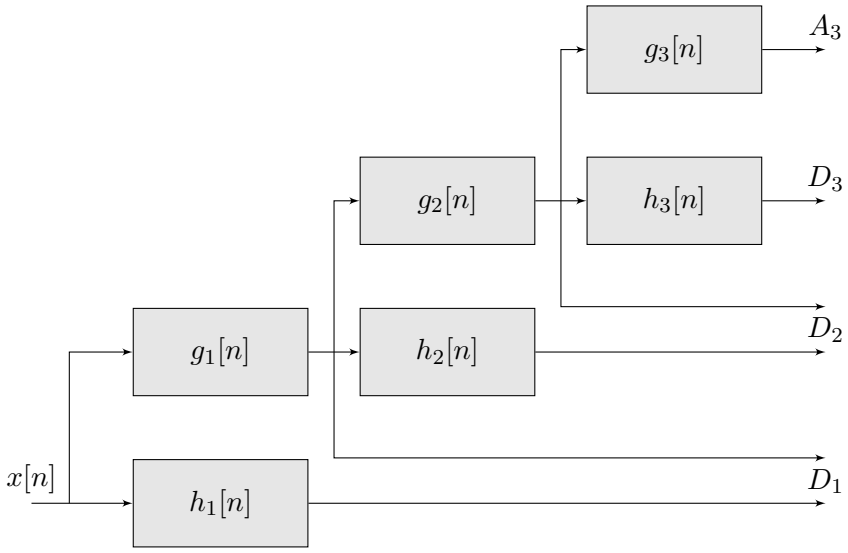


FIGURE 6.14: 3 Level SWT decomposition

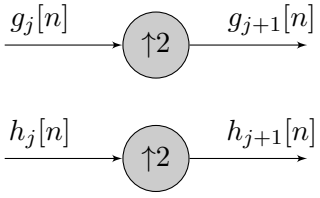


FIGURE 6.15: Filter computation

6.4 Wavelet Packets

The wavelet packets method is a generalization of wavelet dyadic decomposition that offers a richer range of possibilities for signal analysis.

In the DWT dyadic decomposition a signal is split into an approximation and a detail, and the approximation is further split into a second-level approximation and detail. This process is repeated for an $n-$level decomposition, yielding $n+1$ possible choices to decompose or encode the signal. In contrast to the DWT, on wavelet packets analysis the details, as well as the approximations, can be split. This provides $2^{2^{n-1}}$ choices to encode the signal. A wavelet packets decomposition tree is illustrated on figure 6.16.

The scheme of the wavelet packets decomposition provides an equal-distance filterbank with 4 approximations and 4 detail coefficients. This representation is not possible with an ordinary wavelet analysis. However, different hierarchical representations are possible depending on the splitting of approximations and details. In this case an entropy-based criterion is used, although different ones can be used depending on the application.

The procedure is similar to that one in the DWT although splitting of details is also allowed. Figure 6.17 shows some scalogram examples of previously analyzed waveforms

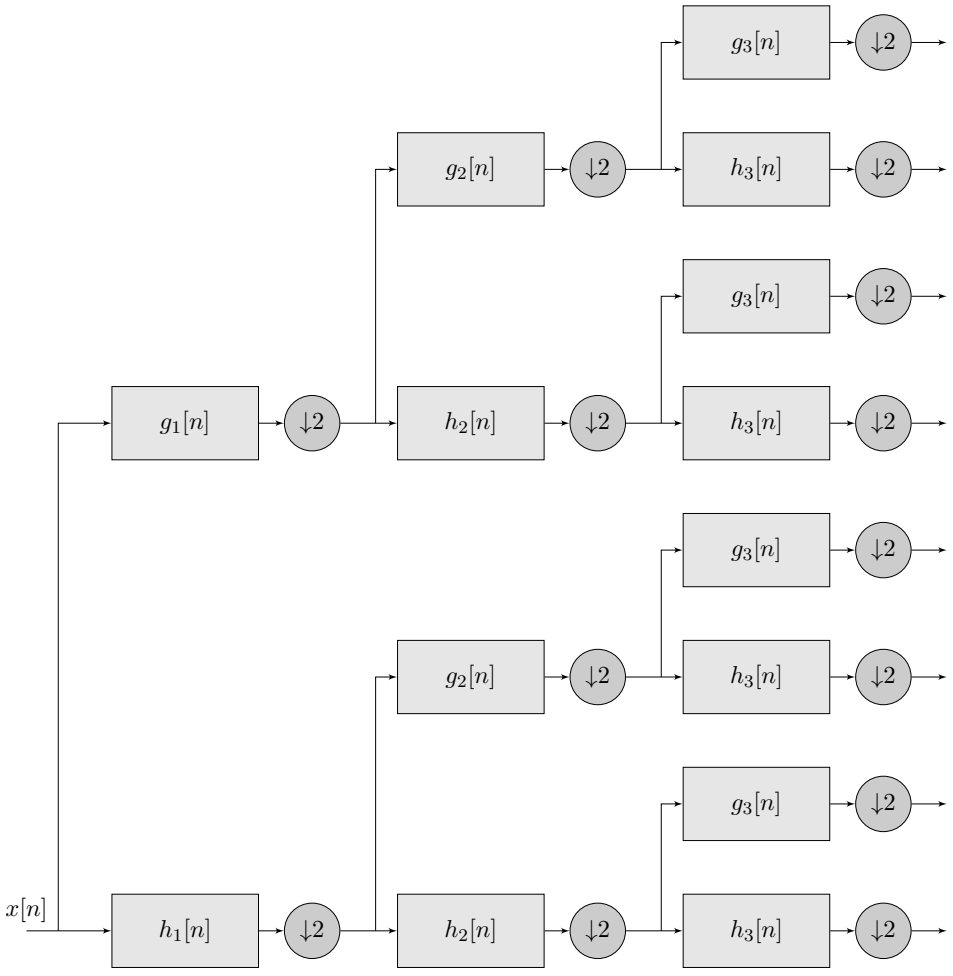
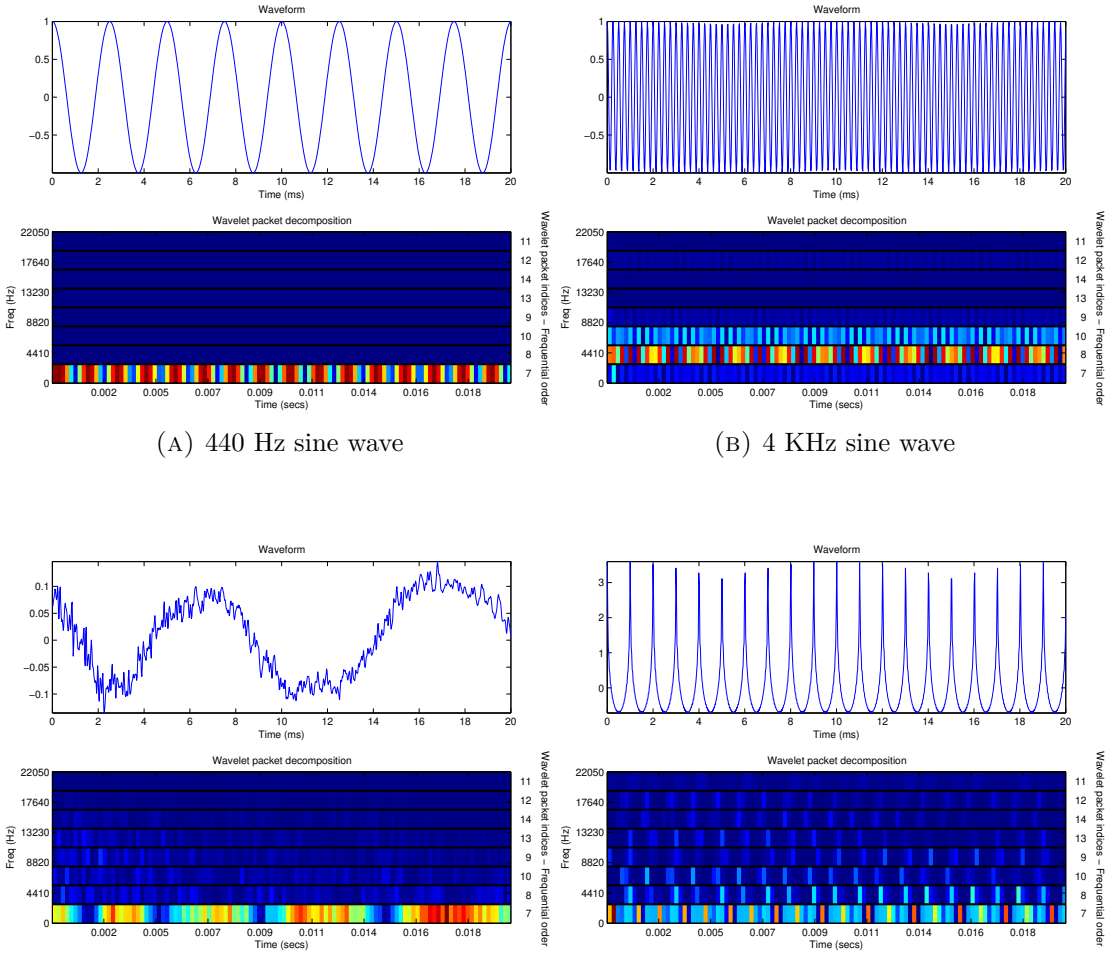


FIGURE 6.16: Wavelet packets 3 Level synthesis

through other wavelet methods and their corresponding wavelet packets representations with the scheme presented before.

Analogously to DWT synthesis, wavelet packets employ upsamplers and a similar hierarchical filterbank to reconstruct the signal from the approximations and detail coefficients depending on the chosen decomposition tree structure used to represent the signal.



(A) 440 Hz sine wave

(B) 4 KHz sine wave

(C) Low frequency with noise sine

(D) 1 KHz High harmonic energy sine

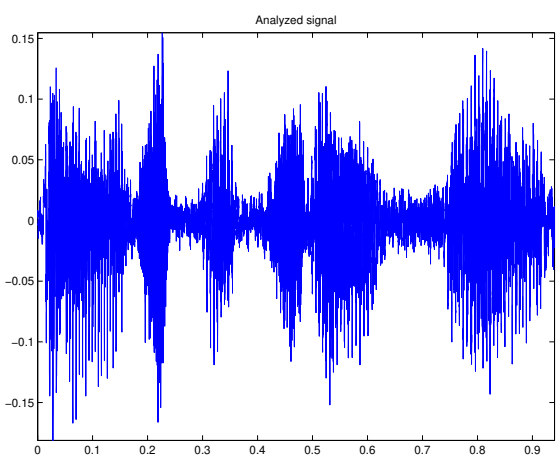
FIGURE 6.17: 3 levels wavelet packets scalogram of various waveforms with 'db4'

6.5 Uses of wavelet

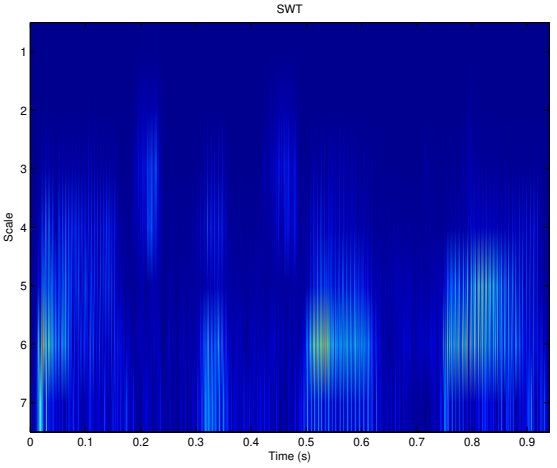
Due to its nature, wavelet analysis has been successfully employed for music processing in several applications such as as pitch tracking, time and pitch scaling, beat tracking, onset detection and feature extraction for pattern recognition. The extracted wavelet coefficients provide a compact representation that shows the energy distribution of the signal in time and frequency. In order to reduce the size of the feature vector some statistical properties, such as mean, deviation and ratios, can be used instead of the coefficients themselves.

However, even though compelling results have been achieved, time-frequency analysis generally results in a more accurate analysis than wavelet analysis at a higher computational cost.

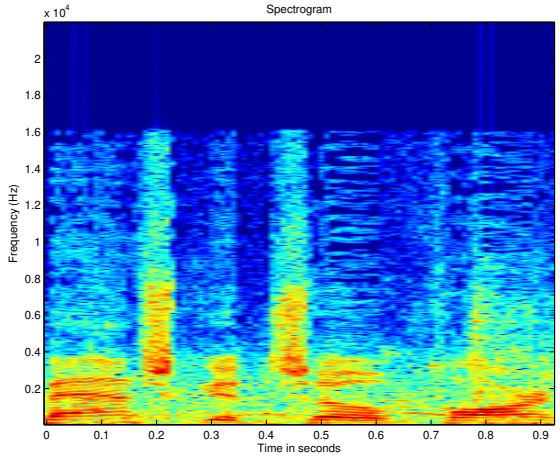
In the field of speech recognition, the wavelet transform has been applied with some success in areas such as pitch detection, formant tracking and phoneme classification



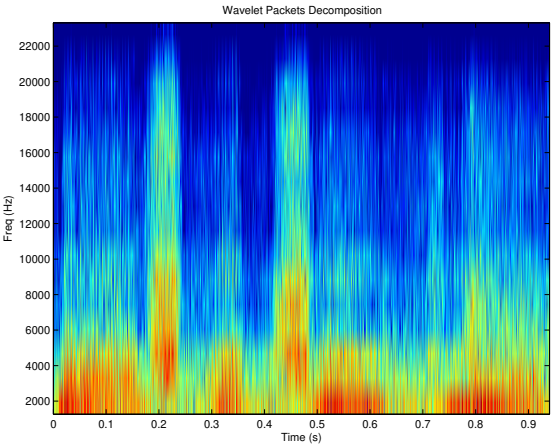
(A) Speech waveform



(B) SWT 7 Levels



(C) Spectrogram



(D) Logarithmic-amplitude wavelet packets

FIGURE 6.18: Various wavelet transforms of a speech signal

[16][17]. Wavelet transform and its variants are useful in speech recognition due to their good feature localization but furthermore because more accurate (non-linear) speech production models can be assumed.

Bibliography

[1] C. Zeng and W. Dou. Audio keywords detection in basketball video. In *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, pages 1765 – 1770. IEEE, 2010.

[2] L. R. Rabiner and R. W. Schafer. Introduction to digital speech processing. *Foundations and trends in signal processing*, 1(1):1 – 194, 2007.

[3] L. Tan and M. Karnjanadecha. Pitch detection algorithm: Autocorrelation method and AMDF. In *Proceedings of the 3rd International Symposium on Communications and Information Technology*, volume 2, pages 551 – 556, 2003.

[4] A Michael Noll. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Proceedings of the symposium on computer processing communications*, volume 779, 1969.

[5] P. De La Cuadra, A. Master, and C. Sapp. Efficient pitch detection techniques for interactive music. In *Proceedings of the 2001 International Computer Music Conference*, pages 403 – 406, 2001.

[6] Meinard Müller. *Information retrieval for music and motion*, volume 6. Springer, 2007.

[7] B. Milner and X. Shao. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. In *Interspeech*, 2002.

[8] B. Milner and X. Shao. Clean speech reconstruction from MFCC vectors and fundamental frequency using an integrated front-end. *Speech Communication*, 48 (6):697 – 715, 2006.

[9] Vivek Tyagi and Christian Wellekens. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In *Proc. ICASSP*, volume 5, pages 529–532, 2005.

[10] D. Zhang and D. Ellis. Detecting sound events in basketball video archive. *Dept. Electronic Eng., Columbia Univ., New York*, 2001.

[11] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian. Creating audio keywords for event detection in soccer video. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II – 281. IEEE, 2003.

[12] F. Bömers. *Wavelets in real time digital audio processing: Analysis and sample implementations*. PhD thesis, Master Thesis. University of Mannheim, 2000.

[13] G. Tzanetakis, G. Essl, and P. Cook. Audio analysis using the discrete wavelet transform. In *Proc. Conf. In Acoustics and Music Theory Applications*, 2001.

[14] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[15] J. Enders, W. Geng, P. Li, M. W. Frazier, and D. J. Scholl. The shift-invariant discrete wavelet transform and application to speech waveform analysis. *The Journal of the Acoustical Society of America*, 117(4):2122 – 2133, 2005.

[16] C. J. Long and S. Datta. Wavelet based feature extraction for phoneme recognition. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 264 – 267. IEEE, 1996.

[17] A. N. Akansu, W. A. Serdijn, and I. W. Selesnick. Emerging applications of wavelets: A review. *Physical communication*, 3(1):1 – 18, 2010.

[18] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al. *Discrete-time signal processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.

[19] W. Holmes. *Speech synthesis and recognition*. CRC Press, 2001.

[20] U. Zölzer. *Digital audio signal processing*. John Wiley & Sons, 2008.

[21] Z. Liu, J. Huang, Y. Wang, and T. Chen. Audio feature extraction and analysis for scene classification. In *Multimedia Signal Processing, 1997., IEEE First Workshop on*, pages 343 – 348. IEEE, 1997.

[22] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V – 632. IEEE, 2003.

[23] J. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2 – 10, 1999.

[24] S. W. Smith et al. The scientist and engineer's guide to digital signal processing. 1997.

[25] G. Muhammad. Extended average magnitude difference function based pitch detection. *The International Arab Journal of Information Technology*, 8(2):197 – 203, 2011.

[26] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674 – 693, 1989.

[27] A. F. Haque. FFT and Wavelet-Based Feature Extraction For Acoustic Audio Classification. *International Journal of Advance Innovations, Thoughts & Ideas*, 1 (1), 2012.

[28] A. Kulin, A. Ghorawat, and A. Sankaran. EE678 Wavelets Application Assignment Applications of Wavelet Theory to Digital Audio Signal Processing.