Department of Computer Science

Submitted in part fulfilment for the degree of BSc.

Causal wavelets for analysis of sound

Daniel Kosky

Version 1.0, 2020-May-06

Supervisor: Samuel L. Braunstein

Acknowledgements

I would like to thank my supervisor Dr Samuel Braunstein for his guidance and support throughout the project.

Contents

Statement of Ethics viii

Executive Summary ix

1 Introduction 1
- 1.1 Wavelet Analysis . . . 1
- 1.2 Project Structure . . . 1

2 Literature Review 3
- 2.1 Progression of Approaches to Signal Processing . . . 3
- 2.2 The Wavelet Transform . . . 5
- 2.3 Properties of Wavelets . . . 6
  - 2.3.1 Common Wavelets . . . 7
- 2.4 Time Frequency Analysis of Music . . . 7
- 2.5 Causal Wavelets . . . 8
- 2.6 Conclusions . . . 9

3 Problem Analysis 10
- 3.1 Introduction . . . 10
- 3.2 Internal Representation . . . 10
- 3.3 Graphical Representation . . . 11
- 3.4 Metrics . . . 11

4 Design and Implementation 12
- 4.1 Selecting an Environment . . . 12
  - 4.1.1 Comparison of environments with regards to requirements . . . 13
- 4.2 Implementation of the Continuous Wavelet Transform . . . 14
  - 4.2.1 Implementation of signals . . . 14
  - 4.2.2 Implementation of wavelets . . . 14
  - 4.2.3 Implementation of continuous wavelet transform . . . 15
- 4.3 Implementation of the Causal Wavelet . . . 17
- 4.4 Implementing a Metric for Causality . . . 19
  - 4.4.1 Implementing a causal signal . . . 19
  - 4.4.2 Determining the pulse train . . . 20
  - 4.4.3 Evaluation of the causality metric . . . 21
- 4.5 Implementing a Metric for Distortion . . . 22
  - 4.5.1 How distortion might be interpreted . . . 22

4.5.2 Implementing a measurement based on angle 22
4.5.3 Evaluation of the distortion metric 24
4.6 Application to sound 24
4.7 Modifying the causal wavelet 25
4.8 Modifying the continuous wavelet transform 27

5 Conclusions 29

A Appendix:Large Figures 31

List of Figures

2.1 A spectrogram representation of, on the left, a drum beat and, on the right, an excerpt from a piano piece. 4
2.2 A scalogram representation of, on the left, a drum beat and, on the right, an excerpt from a piano piece (the same audio as the spectrograms in figure 2.1). 6

4.1 On the left, a plot of a $\sin(3x)$ and on the right a plot of the normalised signal of the piano excerpt (also used in figures 2.1 and 2.2) constructed using the `from_wav_file` constructor function. 15
4.2 Plots of Ricker wavelet (on the left) and Morlet wavelet (on the right) produced by `plot_wavelet` function. 16
4.3 Plots of Ricker wavelet (on the left) and Morlet wavelet (on the right) at different scales produced by plotting `get_scaled_wavelet` function. 16
4.4 Scalogram representation of the same piano excerpt from figures 2.1 and 2.2, created using the `plot_scalogram` function from the continuous wavelet transform class. 18
4.5 Representation of the real and imaginary parts of a causal wavelet as defined in equation 2.9. 18
4.6 Contour representation of the wavelet transforms of the signal $\sin(5x)$ between the values between 0 and 2π, note that the y-axis values represent j, where the scales are selected using 2^j and thus the axis is logarithmic (base 2). The transform on the left was performed using a symmetric Ricker wavelet, the transform in the middle uses the causal wavelet as represented in 4.5, but reflected in the y-axis (though just the real part of the transform is represented here), and the trasform on the right uses the causal wavelet as represented in figure 4.5 (again only the real part is represented here). . 19
4.7 From top to bottom, the causal signal, its continuous wavelet transform as represented by a contour scalogram, and the pulse train as derived from the continuous wavelet transform. The continuous wavelet transform on the left was performed using a Morlet wavelet, and the one on the right is performed using the causal wavelet as specified in 2.9. 21

4.8 A comparison of the wavelet scalograms between Morlet and Ricker wavelets and their adpated causal counterparts, where each scalogram corresponds to the wavelet depicted above it. Reading Morlet, Ricker, causal Morlet, causal Ricker from left to right. 22
4.9 Comparison of a wavelet transform performed with a Ricker wavelet and the causal adaption of the Ricker wavelet (as represented by the second and fourth wavelets in 4.8), and the lines between associated maxima of two specified scales that have been determined as per the `get_lines` function. 23
4.10 Above, real coefficients of wavelet transforms of a pulse like signal (the same as that used in figures 4.7, 4.8, and 4.9) using wavelets as defined by 4.5, with the calculated vectors between the local maxima of two scales represented by the red lines. Below, a representation of the average angle of the distortion (in degrees) against the constant used in defining the frequency of the causal wavelet used in the transform, as specified in 4.5). 26

A.1 Wavelet transform coefficients from the `peform_transform` function on a fraction of an audio signal at the dyadic scales 2^j. 31
A.2 Comparison of scalograms created by wavelet transform of a pulse using a symmetric Morlet wavelet, the causal adaption of the Morlet wavelet, and a modified causal wavelet transform of the modified causal wavelet (top to bottom). . . 32

List of Tables

3.1 Table of requirements regarding inputs and the representation of data within the system 10
3.2 Table of requirements regarding graphical representation of data by the system 11
3.3 Table of requirements regarding metrics calculated by the system 11

4.1 Delay as measured by the `get_delay` function described in section 4.4 using snippets of the same drum and piano excerpts displayed in figures 2.1 and 2.2. 24
4.2 Average angle of distortion as measured by the `get_angles` function in the `DistortionAnalysis` class. 25
4.3 Delay measured using the `CausalityAnalysis` and average angle of as measured by the `get_angles` function in the `DistortionAnalysis` class of the transforms seen in figure A.2. 27

Statement of Ethics

This project was completed with appropriate attention to ethical standards. All material used in the project is either the product of my work, or explicit reference is given to the source. No private information was used, no data was gathered from participants, and the project contains no information that might risk the safety of anyone.

Executive Summary

This project explores the application of the continuous wavelet transform, a signal processing technique used for time-frequency analysis; and the application of specialised 'wavelets' within the transform.

It is common to represent signals with amplitude as a function of time (i.e. in the time domain). It is also common, using a Fourier transform, to represent signals with amplitude as a function of frequency (i.e. in the frequency domain). Time-frequency representations of signals are three-dimensional represenations, providing the amplitude of a signal given the time and frequency. Common techniques for obtaining this representation include the short time Forier transform, and the the Wavelet transfrom. Unlike a Fourier transform, the wavelet transform does not represent the signal using infinite wave trains (sines and cosines), rather it employs oscilations known as wavelets, shifted in time and frequency. These wavelets are generated from a 'mother' wavelet, a function which given parameters for time and scale, generates a wavelet of that scale, compactly supported for a given time. The matrix representing the amplitude of these wavelets at time and scale is the wavelet decomposition of the signal. This allows for sparse representation of the signal, which is why the wavelet is a common tool for compression; and since it allows for the representation of transients within signals, it is also a useful tool for analysis of music, seismic data, and many other signals.

The admissibility condition defines the properties of acceptable wavelets. Within this constraint there are still an infinite number of possible wavelets. As such, different wavelets have been designed throughout different fields, selected for the properties that best suit their applications.

This project focuses on application of causal wavelets in the continuous wavelet transform. Since most wavelets are symmetric, the wavelet transform results in the component of a signal being represented in time before it begins. Causal wavelets are wavelets that have been designed for the express purpose of ensuring causality in the wavelet transform, such that components of a signal should only be represented as, or after, they occur. Causal wavelets are non-symmetric, and a result of their use is that the transform becomes distorted in time and frequency. The objective of this project is to produce causal wavelets that minimise this distortion. This would be useful for analysis of a signal in which causal representation of the components is necessary (e.g. an active sonar system).

It was decided that the best programming environment in which to explore this concept was python. Matlab and Wolfram Mathematica provided inbuilt support for the continuous wavelet transform, but it was decided that an implementing of the wavelet transform from scratch would better match the requirements of the project.

The design and implementation involved the construction of an object-ortientated system that could perform the continuous wavelet transform of any signal using any wavelet (within the limits of representation). Methods by which graphical representations of the transform are obtained had to be implemented.

A challenge of implementation was the implementation of metrics with regards to causality and distortion. The method for calculating a measurement of causality calulated the difference between a prediction of the start of the transform and the start of the signal. The method to calculate the distortion sought the local maxima of the transform over two scales and associated specific maxima in order to trace the angles of distortion within the transform, calculating an average to provide a rough measure of distortion. When tested against test signals and real audio, these methods seemed to perform as expected, with the use of the causal wavelet resulting in transforms with more causal behaviour and greater angles of distortion (as evaluated by the methods). Despite doubts regarding the objectivity of these metrics, they serve as useful comparative measurements.

In order examine the effects of different causal wavelets upon the distortion in the transform, a test was constructed wherein the distortion measured in the transform was measured against the a set of causal wavelets increasing in the internal frequency of their oscilations. The results showed that increasing frequency led to better frequency localisation in the transform, but also resulted in a greater degree of distortion.

Thus a method of cancelling the distortion was contrived using a modified wavelet that changed in internal frequency as it changed in scale. This method worked well, whithin certain scales, at cancelling the distortion whilst maintaining the causality present in the transform. However, this came at the cost of worsened frequecny localisation of in the transform and confused the relationship between wavelet scale and frequency (which is normally simply inversely proportional). This modification in many senses achieved the goal of cancelling distortion caused by the causal wavelet, but it likely has very little application in analysis unless it is better refined.

There are no legal, social, ethical, professional or commercial issues related with this topic of study, or my findings.

1 Introduction

1.1 Wavelet Analysis

The wavelet transform is a mathematical technique that can be used for the purpose of signal decomposition. The wavelet transform, unlike the Fourier transform, does not represent a given signal by a weighted sum of sinusoids, rather as coefficients over a set of functions known as wavelets.

Developments in wavelet analysis in previous decades were made independently across fields such as mathematics, quantum physics, electrical engineering and seismic geology, and the technique has seen applications in fields such as astronomy, acoustics, nuclear engineering, music and magnetic resonance imaging to name a few [1].

Different wavelets exhibit different properties that make them well suited to uses in particular applications. This is to say that the wavelet selected for a wavelet transformation has an effect upon the output, and that some wavelet properties might be more desirable for certain types of analysis. This project will experiment with modification of wavelets and their symmetries in order to examine the effects that these modifications have on the wavelet analysis, with the ultimate goal of finding wavelets that exhibit 'causal' behaviour with minimal distortion to the transform.

1.2 Project Structure

Chapter 2 will examine the wavelet analysis within the context of existing literature. This will involve examining techinques in signal analysis that preceded wavelet analysis, how a wavelet transform is performed, what a wavelet is, the applications of the wavelet transform and the idea of causal wavelets. This section should provide context and the motivations for the objectives of the project.

Chapter 3 is used to describe the reasoning for the structure of this project, outlining how objectives have been evaluated to produce explicit, measurable requirements.

Chapter 4 will cover step-by-step the choices made in designing and implementing a system that fulfils the requirements elicited in 3.

Chapter 5 will discuss my evaluation of the project as per the phases of the implementation. With reference to my goals throughout the project, I will describe ways in which I think my project has been successful, and ways in which I think future work could improve upon mine.

2 Literature Review

This chapter will cover the relevant research and literature regarding wavelets in the context of this project. Section 2.1 will discuss some ideas in signal processing that can be seen as prerequisites for the wavelet transform. Section 2.2 discusses the wavelet transform, with reference to the contiuous and discrete definitions. Section 2.3 discusses different types of wavelets and the functions that define them. Section 2.4 examines the wavelet transform in the context of music. Section 2.5 discusses the design of the causal wavelet.

2.1 Progression of Approaches to Signal Processing

Time and frequency domain representation An important aspect of signal processing is signal representation. Signals are often represented in the time domain, where the signal is a function of time [2]. Though it is intuitive to consider a signal in the time domain, it is often desirable to observe the constituent frequencies of a signal in the frequency domain. For a long time Fourier transformations have been a useful tool for this sort of single domain analysis, allowing mathematicians to transform data from time domain representation to frequency domain representation and backwards by means of a couple of transforms. A definition for the forward Fourier transform is provided in [3] as:

$$F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi xs} dx \tag{2.1}$$

Gabor transform Despite the popularity of the Fourier transform, there are limitations of being able to observe a signal in one domain. Gabor in his 1945 paper [4] addresses the limitations of representing signals only within "sharply defined instants of time" or with "infinite wave trains of rigorously defined frequencies" with a method of representing signals in terms of both domains simultaneously. Gabor's solution is a type of Short time Fourier transform (STFT). The Gabor transform allows for the frequency content to be analysed at different time intervals. This is done by using Fourier transforms on the product of the signal and Gaussian window function for each interval, where the window function has compact support within the given time interval.

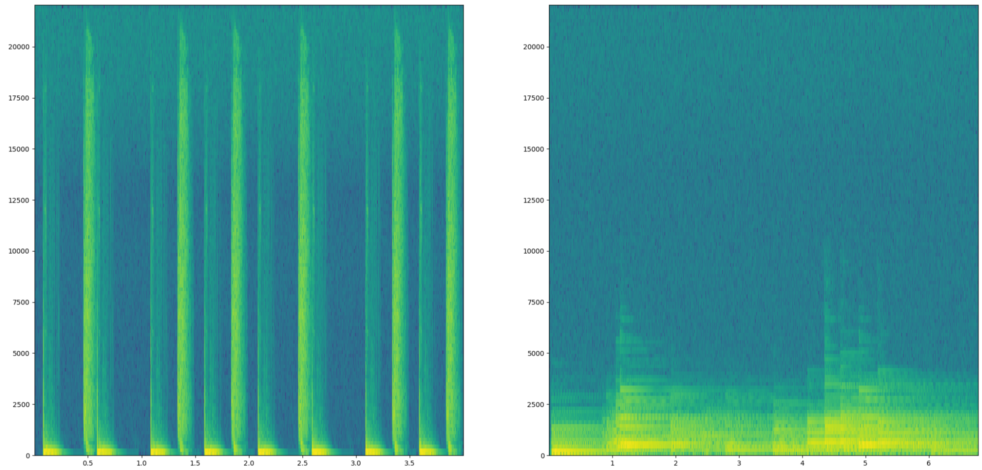

Figure 2.1: A spectrogram representation of, on the left, a drum beat and, on the right, an excerpt from a piano piece.

Spectrograms A signal that as been analysed in this way can be represented using a spectrogram. Spectrograms represent time and frequency simultaneously, and in this way can be interpreted as representing "the signal strength [...] of a signal over time at various frequencies present in a particular waveform" [5]. This allows for features such as transients in the signal to be represented at the time at which they occur.

Uncertainty principle It is worth noting that time and frequency are subject to the same rule of uncertainty that governs related pairs of variables, famously described by Heisenberg in the context of the position and momentum of a particle. In [4] Gabor relates the Heisenberg uncertainty principle to time and frequency, noting a mathematical identity:

$$\Delta t \Delta f \simeq 1 \tag{2.2}$$

where Δt and Δf are the uncertainties in the time and frequency of an oscillation. For this reason, lower frequency components of a signal can be more precisely located in frequency, but are generally less precisely located in time, and higher frequency components are more precisely located in time, but are less precisely located in frequency.

Though Short time Fourier transforms can be used to perform effective time-frequency analysis, there are some limitations inherent to the method. A fixed window of a short length will have poor representation of low frequencies, and a fixed window of long length will have poor localisation of high frequencies [6].

2.2 The Wavelet Transform

Continuous wavelet transform The wavelet transform can be understood as an alternative method to the Short time Fourier transform that addresses some of the aforementioned issues. In [7] Daubechies discusses Morlet's application of the wavelet transform as a progression from the Short time Fourier transform. She states that in order to attain good time resolution for high frequencies and good frequency resolution for low frequencies, Morlet was motivated to devise a different method of generating the transform functions. This method involves the use of a function known as a 'mother' wavelet that is shifted and dilated, which can be considered as taking the place of exponential $e^{-i2\pi xs}$ term in the Fourier transform. Daubechies provides a definition for the continuous wavelet transform in [8], in which it is represented as:

$$(T^{wav}f)(a,b) = |a|^{-1/2}\int dt f(t)\psi\left(\frac{t-b}{a}\right) \tag{2.3}$$

Where $(T^{wav}f)(a,b)$ is the wavelet transform of $f(x)$ over the continuous scaling and shifting parameters a and b and the mother wavelet is represented by ψ.

It should be noted that despite mention of Morlet above, wavelet techniques are considered to have developed independently across different fields [1].

Discrete wavelet transform The discrete wavelet transform is another implementation of the wavelet transform. A version of the discrete wavelet transform is defined in [8] as:

$$T^{wav}_{n,m}(f) = a_0^{-m/2}\int dt f(t)\psi(a_0^{-m}t - nb_0) \tag{2.4}$$

Upon inspection, this is not all too different from the continuous wavelet transform, both functions use the inner products of f and the generated wavelets. The variables m and n discretise the wavelet sampling. Computation of the discrete wavelet transform can be implemented using a filterbank, and is considered to be more efficient than the continuous wavelet transform [9].

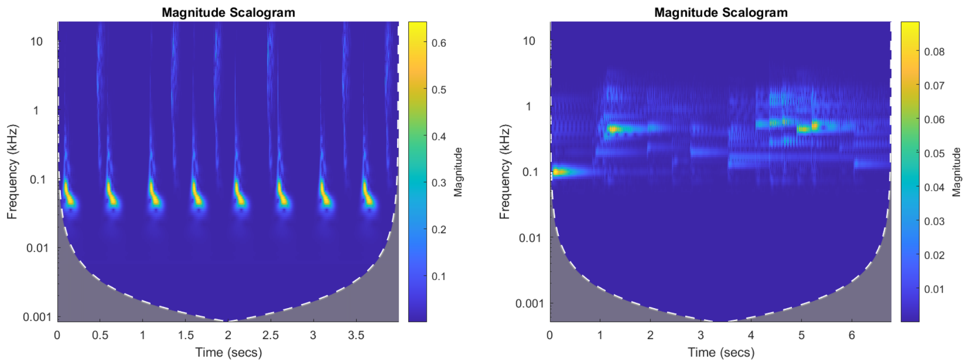

Figure 2.2: A scalogram representation of, on the left, a drum beat and, on the right, an excerpt from a piano piece (the same audio as the spectrograms in figure 2.1).

Wavelet scalograms The scalogram is a method of representing the wavelet transform, much as a spectrogram might represent a Short time Fourier transform, both are time-frequency representations of a signal. In the case of the wavelet scalogram, the absolute values of the wavelet coefficients are represented, and frequency is actually derived from the wavelet scale parameter (though sometimes wavelet scale is used instead of frequency). Due to the Heisenberg-Gabor uncertainty, these should not be thought of as direct representations of the energy at each point in time and scale, rather they should be thought of as representations of the energy density over each point in time and scale [10].

2.3 Properties of Wavelets

Admissibility condition There are an infinite number of possible wavelets [1]. For our purposes, the conditions of what shall constitute an acceptable mother wavelet are that it shall have support within a finite interval (compact support), and that it shall have an average value of zero in the time domain [11]. This allows the wavelet to maintain a constant energy when dilated, meaning that at all scales the wavelet will have the same energy. The admissibility condition as described in [11] can be written as:

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty \tag{2.5}$$

Where $\Psi(\omega)$ is the Fourier transform of the mother wavelet. This means that over all frequencies, the Fourier transform of the mother wavelet is finite. The admissibilty condition implies that the Fourier transform of the mother wavelet vanishes at a frequency of zero.

2.3.1 Common Wavelets

Haar Wavelets According to [1] the first reference to wavelets occurred in an appendix to a thesis by Alfréd Haar, though the term wavelet was not used until much later. The Haar mother wavelet as defined in [12] can be represented using the piecewise function:

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1/2, \\ -1, & \text{if } 1/2 \leq t < 1, \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

The Haar wavelet was devised as an orthonormal system that could be used as an alternative to the trigonometric system used in the Fourier transform for the approximation of functions [13]. Wavelets are considered to be "basis vectors in high-dimensional spaces" [14], which is to say that any point in the high dimensional space can be represented by only one combination of these vectors. The Haar wavelet basis function has the additional property of being orthonormal (i.e. the dot product of these basis functions is zero).

Morlet wavelets Morlet wavelets were first used by geologist Jean Morlet in his afformentioned adaption of the Gabor transform [8]. Morlet wavelets are defined in [15] as:

$$\psi_{a,b}(t) = exp\left[-\frac{\beta^2(t-b)^2}{a^2}\right]\cos\left[\frac{\pi(t-b)}{a}\right] \tag{2.7}$$

With scaling and shifting parameters a and b.

Ricker wavelets The Ricker wavelet is also known as the Mexican hat wavelet. It is defined in [16] as:

$$Ricker(t) = (1 - 2\pi^2 f^2 t^2)\exp(-\pi^2 f^2 t^2) \tag{2.8}$$

Where the parameter f defines the wavelet's peak frequency in the frequency domain representation of the wavelet.

2.4 Time Frequency Analysis of Music

The use of digital signal processing techniques to provide new insight into music is relatively common [17]. Time-frequency analysis is particularly suited to musical analysis since all music can be understood in terms of both time and frequency. The paper *Time-frequency Analysis of Musical Rhythm* [18] written by Cheng et al. discusses, in the context of music, the use of spectrograms such as those used displayed in figure 2.1 and wavelet scalograms such as those seen in figure 2.2 and their applications in analysing rhythm. When observing time-frequency representations of music it is common to see, corresponding to an event, a region of high amplitude (known as the root) succeeded along the axis in which the frequency is represented by higher frequencies of a lower amplitude known as harmonics. Though the root defines the prevalent note that we would hear when listening to the sound, the harmonics define many of the characteristics of the sound. The importance of good time-frequency representations for the analysis of sound is clear and the wavelet transform has properties that make it effective for this purpose, but also properties that make it less effective for this purpose. The symmetry of common wavelets as discussed later in section 2.5 represents an issue with the wavelet transform's application in musical analysis, since it can confuse the time at which a musical event (e.g. a drum beat) is perceived to have started, and might thus affect the determination of rhythm from a wavelet transform.

2.5 Causal Wavelets

Motivation for causal wavelets In [19] a causal system is described as a system that "cannot have output before input is applied", the authors provide the example of an active sonar system, where "an echo cannot be detected before a pulse is sent". The belief presented in [19] is that a method that enforces the feature of causality in the wavelet transform should be better in terms of the signal to noise ratio, since the "kernel is absolutely zero for times that are forbidden by the law of causality"[19]. This also makes sense in the context of analysis of other systems, for instance in the analysis of musical signals it does not make sense to detect a drum beat or a note on the piano before the drum is struck or the piano key is played. However, this causal representation of events is not inherent to the wavelet transform. The most common families of wavelets are symmetric or antisymmetric meaning that, for instance, transients will be represented within the wavelet transform before they actually occur.

The causal wavelet The causal mother wavelet as used in [19] is an adaption of the Morlet wavelet and is given by:

$$h(t) = \begin{cases} \cos(5t)\exp(-t/2) + j\sin(5t)\exp(-t/2), & \text{for } t \geq 0, \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

2.6 Conclusions

In this chapter I have discussed, in the context of existing literature, the wavelet transform in terms of some of the signal processing techniques that preceded it, the differences between the continuous and discrete wavelet transform, the properties of wavelets, the applications of the wavelet transform to the analysis of sound, and the qualities of the causal wavelet. I hope that the motivations for the investigation of causal wavelets have been made clear.

3 Problem Analysis

3.1 Introduction

The best method I can conceive of for representing the objectives is to construe them in terms of the requirements of a system that can perform tasks related to these objectives (e.g. perform a continuous wavelet transform). As such, the system that fulfils these objectives will be able to perform wavelet transforms using a variety of wavelets and provide the tools necessary to perform comparisons of these transforms. Minimising distortion will remain a general objective rather than a requirement.

3.2 Internal Representation

The prerequisites for performing a continuous wavelet transform are as such: a signal, a specified wavelet, and a range of scales over which the transform is performed. A system that performs a continuous wavelet transform using this data will need methods that appropriately handle the discrete nature of the components, and the output as well.

A_1	The system shall accept and utilise discrete time signals.
A_2	The system shall have a method of importing audio signals.
A_3	The system shall accept and utilise any possible wavelet.
A_4	The system shall accept and utilise any set of positive real number scale parameters.
A_5	The system shall provide the wavelet transform of the inputs in discrete dimensions that match those of the signals and those of the scales.

Table 3.1: Table of requirements regarding inputs and the representation of data within the system

3.3 Graphical Representation

For the purposes of analysis, it is important that there be a way to produce graphical representations of the signal, wavelet and the transform. Therefore, the system used should be able to represent plots of the signal and wavelet, and a scalogram (as discussed in the literature review section 2.2) of the wavelet transform.

B_1	The system shall produce time domain plots of signals.
B_2	The system shall produce time domain plots of wavelets.
B_3	The system shall produce scalogram representations of wavelet coefficients.

Table 3.2: Table of requirements regarding graphical representation of data by the system

3.4 Metrics

As this project is investigating causal wavelets, it is important that there is some measure of causality derived from the wavelet transform. This way we can compare how causal a wavelet might be in comparison to another. Thus, the system must have some method that calculates a metric for comparing how causal a particular transform is with reference to the signal.

The distortion present in wavelet transforms that utilise asymmetric wavelets is an artefact that is to be compared between different wavelets, since a goal of this project is to minimise this. Therefore the system should also have some method of calculating this distortion.

C_1	The system shall contain a method that calculates a metric describing how well a transform obeys the constraint of causality.
C_2	The system shall contain a method that calculates a metric describing the distortion present in a wavelet transform.

Table 3.3: Table of requirements regarding metrics calculated by the system

4 Design and Implementation

This chapter will discuss the approach to the design and implementation of a system that fulfils the requirements set out in chapter 3.

4.1 Selecting an Environment

There are numerous options for programming environments that can be used to implement the above requirements. Some have existing features and tools related to the wavelet transform. From these, I have elected to discuss Wolfram Mathematica, Matlab and Python as these are the three platforms that I have attempted to use.

Wolfram Mathematica Mathematica is a technical computing system with a vast repertoire of tools, including an implementation of the continuous wavelet transform [20]. It uses a proprietary language called the Wolfram Language, which blends elements of functional and rule based programming (amongst other paradigms), and has strong support of symbolic computation.

Matlab Matlab is a numerical computing environment. In many ways it differs little form Mathematica, it too has a wide range of inbuilt tools, and a proprietary programming lagnuage. The programming paradigm employed by Matlab is substantially less supportive of symbolic computing than Mathematica, and adopts more of a procedural, object-orientated paradigm. Matlab also has an implementation of the continuous wavelet transform [21].

Python The Python environment differs from the above two computing environments in that it is simply a combination of programming langauge and interpreter. The standard Python distribution does not provide a wide variety of tools, but it does have a significant repository of open-source libraries. One such package with support for the continuous wavelet transform is PyWavelets [22].

4.1.1 Comparison of environments with regards to requirements

Inputs and data representation All of the above environments are capable of satisfying these requirements. A_2, for instance, is a common feature with almost exactly comparable methods between these environments. However, when considering the compatability of some of these requirements in the context of both Matlab and Mathematica, the inbuilt functionality might match our requirements, but somewhat complicates its performance.

A_3 is possible within the provided frameworks of these environments. Mathematica requires a complex specification of the wavelet properties in order to use a custom defined wavelet within its framework, first requiring a specification of whether the wavelet is orthogonal and/or biorthogonal, then requiring a specification of a set of filter coefficients in order to define the wavelet.

Matlab requires that wavelets are defined within the framework of a tool named 'wavemngr', to which wavelets are saved before they can be used in Matlab. The 'wavemngr' tool does match our A_3 requirement, but obfuscates the parts of the process. Though it should be mentioned that the source code is accessible through Matlab, Matlab's source is very complicated to navigate.

The aforementioned PyWavelets Python package has a method for defining custom wavelets, but only supports the use of these custom wavelets in the discrete wavelet transform, thus failing requirement A_3.

It seems that in order to best fulfil the requirements regarding representation of data and inputs, I require an environment that has support for defining custom wavelets in terms of the wavelet function, can employ these custom wavelets in a continuous wavelet transform, and has a suitable degree of internal transparency.

Graphical representation of data All three of the proposed environments have support for all of the required graphical represenations. Mathematica and Matlab have built-in plot and scalogram functions, and there are many Python data visualisation libraries available that can perform all of the necessary plots.

Metrics Since the application of causal wavelets, in my experience of researching this project, has appeared to be atypical to much of wavelet analysis, it is not surprising that the metrics I have outlined in table 3.3 do not appear to be built in to any of the environments. The functions that I might require in order to calculate these metrics however are present in all environments.

Conclusion of environment selection After attempting to perform a continuous wavelet transform with a custom defined wavelet in a number of environments, I was frustrated to find that this relatively simple task was relatively difficult to perform within the existing frameworks. After conferring with my supervisor, it was decided that the best course of action would be to implement the continuous wavelet transform from scratch. I have elected to use Python for this implementation since it is the programming language I have the greatest level of experience with.

4.2 Implementation of the Continuous Wavelet Transform

In order to implement a system that performs the continuous wavelet transform I elected to use an object orientated approach. My motivations for this were that it allowed me to compare objects and their attributes/functions to the requirements that I had specified in chapter 3, and that I found it conceptually easier to develop with objects. Below is a discussion of objects I used to implement the continuous wavelet transform and some of their associated attributes and functions.

4.2.1 Implementation of signals

Attributes of the signal class The signal class has attributes defining the x and y values of the signal, where the y values represent the amplitude over the sampling points represented by the x values. These attributes fulfil requirement A_1.

Functions of the signal class The signal has a function for plotting the signal `plot_signal`, which plots the signal (normalised) in the time domain, as specified by B_1. The function `from_wav_file` is a constructor that constructs a signal object from a .wav file.

4.2.2 Implementation of wavelets

Attributes of the wavelet class The wavelet class has attributes defining the wavelet psi, the x values over which the psi is sampled, and an attribute

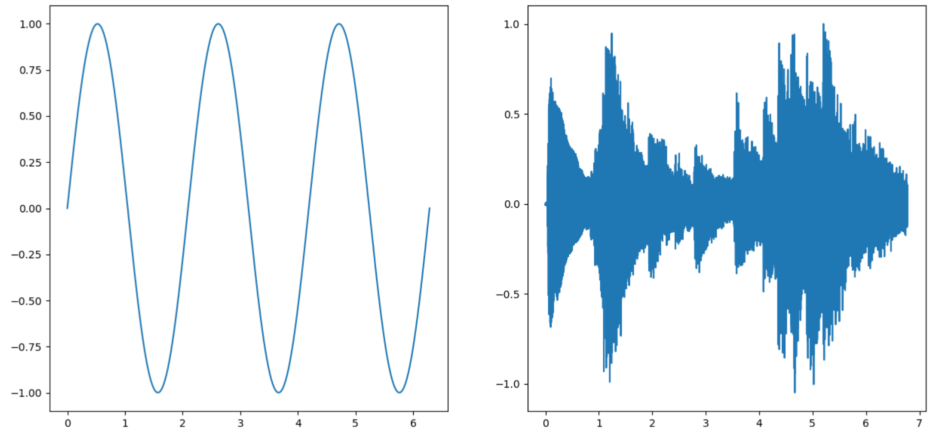

Figure 4.1: On the left, a plot of a $\sin(3x)$ and on the right a plot of the normalised signal of the piano excerpt (also used in figures 2.1 and 2.2) constructed using the `from_wav_file` constructor function.

defining the datatype of the psi values (either a 64 bit float, or a 128 bit complex conjugate represented by two 64 bit floats). As such I have chosen a numerical representation of the wavelet rather than one that is defined in each case by the function of the wavelet $\psi_{a,b}(t)$. This allows me to fulfil requirement A_3.

Functions of the wavelet class The wavelet class has a function that provides a graphical representation of the wavelet, this is titled `plot_wavelet`. This function allows me to fulfil requirement B_2.

The wavelet class has a function named `get_scaled_wavelet`. This function returns a wavelet of a given number of samples at a given scale. The sampling is done by linear interpolation of the wavelet's psi values, and the scaling is performed. One constraint is that the number of points sampled should be directly proportional to the scale.

4.2.3 Implementation of continuous wavelet transform

Attributes of the continuous wavelet transform class The continuous wavelet transform class is initialised using a signal object, a wavelet object and a set of scales to be used in the transform. The scales can be any set of positive real numbers (though naturally limited to the precision of the datatype they are presented as), which fulfils requirement A_4. The continuous wavelet transform class has an attribute named transform which

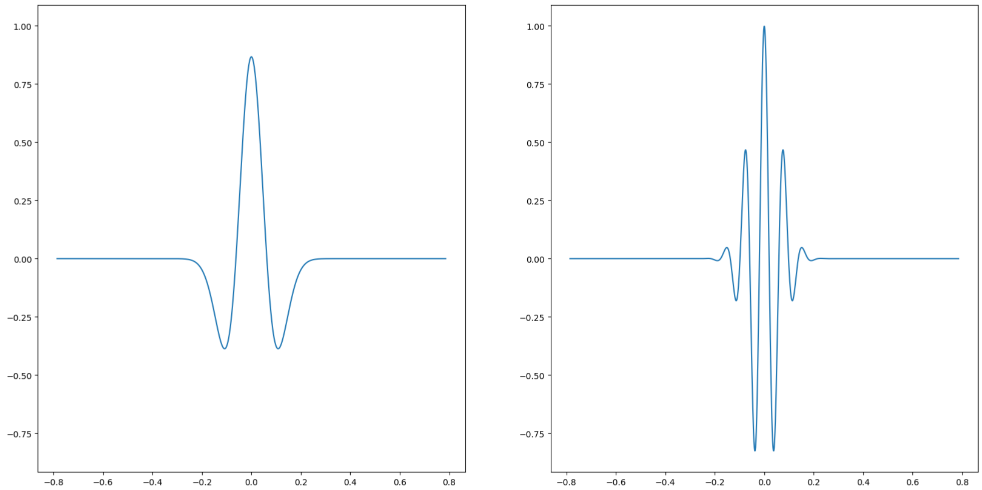

Figure 4.2: Plots of Ricker wavelet (on the left) and Morlet wavelet (on the right) produced by `plot_wavelet` function.

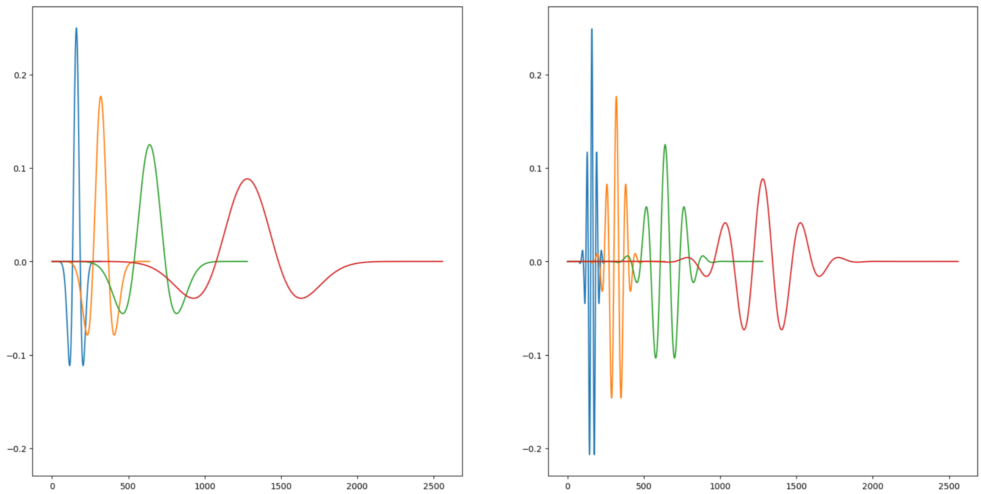

Figure 4.3: Plots of Ricker wavelet (on the left) and Morlet wavelet (on the right) at different scales produced by plotting `get_scaled_wavelet` function.

contains the coefficients of the wavelet transform.

Functions of the continuous wavelet transform class The continuous wavelet transform class has a function named `perform_transform` that performs the continuous wavelet transform with the given parmeters upon its instantiation. The `peform_transform` function is based on the time-invariant dyadic wavelet transform proposed in [23], which is defined by:

$$Wf(u,2^j) = \langle f, \psi_{u,2^j} \rangle = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{2^j}} \psi \left(\frac{t-u}{2^j} \right) dt = f \star \bar{\psi}_{2^j}(u) \quad (4.1)$$

where:

$$\bar{\psi}_{2^j}(t) = \psi_{2^j}(-t) = \frac{1}{2^j} \psi \left(\frac{-t}{2^j} \right) \quad (4.2)$$

The application of this is similar to that of the cwt function in the scipy.signal.wavelets module [24]. To apply the transform as specified in the above equations, the `perform_transform` function, at each scale (scales should be provided in the form 2^j), employs the `get_scaled_wavelet` function of the provided wavelet object. The returned array is reversed and convolved with the signal to provide the wavelet coefficients at that scale. If the scale of the wavelet is such that the number of samples exceeds the number of samples in the signal, the number of samples in the scaled wavelet is fixed at the length of the signal. This does not prevent the wavelet from scaling properly, since the `get_scaled_wavelet` function has separate parameters for the number of samples and the scale. Figure A.1 is a representation of coefficients of the transform at the dyadic scales 2^j.

The plotting functions of the wavelet function `plot_scalogram` achieves requirements B_3 by plotting a time-frequency representation of the wavelet transform using the absolute values of the wavelet coefficients. The `plot_scalogram_contour` performs a similar representation to `plot_scalogram` except with a contour diagram. Figure 4.4 displays the result of the `plot_scalogram` on the piano same piano excerpt from figures 2.1 and 2.2.

4.3 Implementation of the Causal Wavelet

Adopting a similar equation to 2.9 (except time reversed) as the first definition for a causal wavelet, as proposed by Szu et al. in [19], a wavelet object can be instantiated. This wavelet has real and imaginary parts as displayed in figure 4.5. Any subsequent definition of a causal wavelet (such as the adaption of the Ricker wavelet seen in section 4.5) is based on the same technique of using just one half of the symmetric mother wavelet. These causal wavelets pass the admissibility condition as specified in 2.3, with the Fourier transform of the wavelet integrating to a finite value, and vanishing at a frequency of zero.

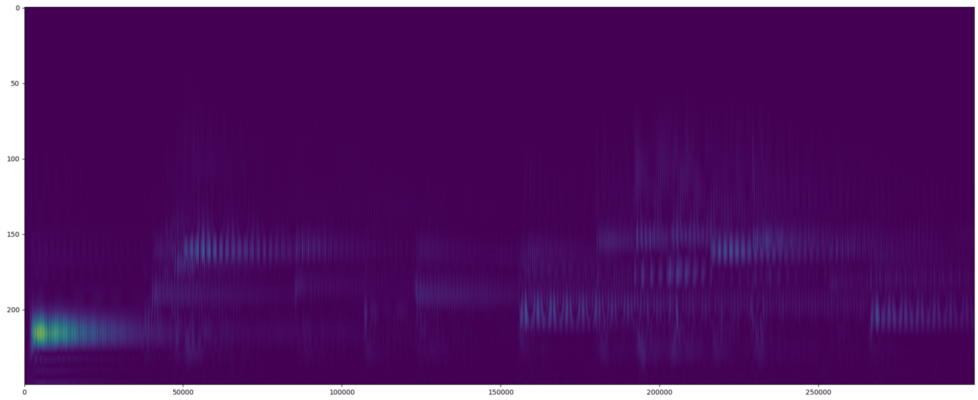

Figure 4.4: Scalogram representation of the same piano excerpt from figures 2.1 and 2.2, created using the `plot_scalogram` function from the continuous wavelet transform class.

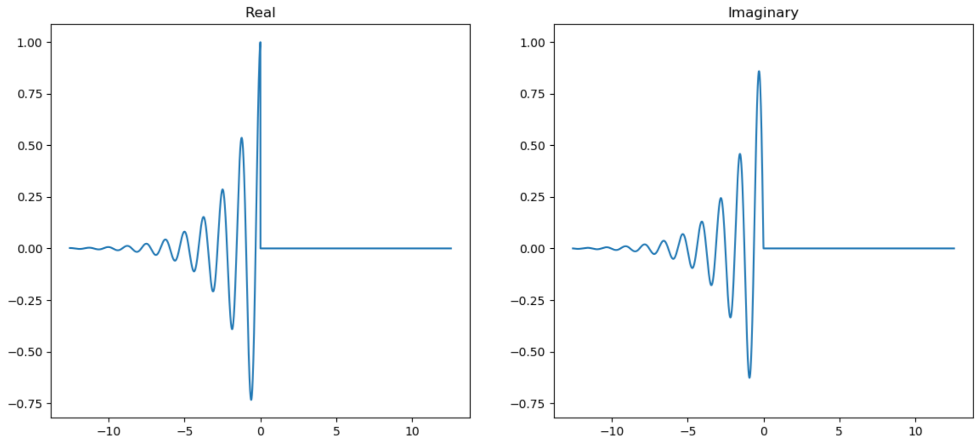

Figure 4.5: Representation of the real and imaginary parts of a causal wavelet as defined in equation 2.9.

When applying this wavelet in the case of a wavelet transform to a stationary signal, distortions of the coefficients in time seem to occur, such as those seen in figure 4.6, with higher and lower frequencies being represented earlier or later. The time distortion present in the wavelet transform is also reflected in the y axis when the causal wavelet is reflected in the y axis, this effect can be seen in the transforms in the middle and right of the figure.

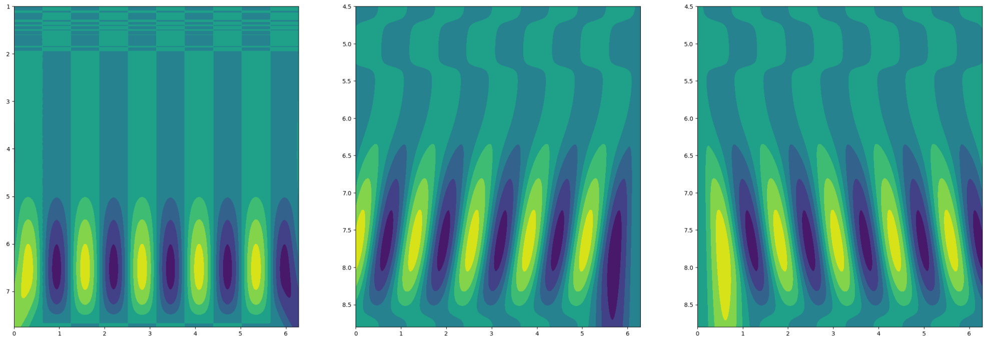

Figure 4.6: Contour representation of the wavelet transforms of the signal $\sin(5x)$ between the values between 0 and 2π, note that the y-axis values represent j, where the scales are selected using 2^j and thus the axis is logarithmic (base 2). The transform on the left was performed using a symmetric Ricker wavelet, the transform in the middle uses the causal wavelet as represented in 4.5, but reflected in the y-axis (though just the real part of the transform is represented here), and the trasform on the right uses the causal wavelet as represented in figure 4.5 (again only the real part is represented here).

4.4 Implementing a Metric for Causality

When considering the requirement C_1 from table 3.3 in the context of this system, I realised that it would be difficult to approach implementing a method of calculating a metric like this for signals in which more than one event occurs, due to the difficulty in associating parts of the transform to specific events. In order to simplify this, I have limited examination of causality to transforms that are performed on signals in which only one event occurs.

4.4.1 Implementing a causal signal

In examining the properties of the causal wavelet in the continuous wavelet transform, it is likely best that the transform is applied to a signal that conforms to the idea of a causal event. I will be refering to this signal as the causal signal. The chief properties of this causal signal are that it represents an oscilation, preceded by a signal with trivial amplitude, that starts with an initially large amplitude and decays over time. This imagining of a causal signal can be envisaged as a drum beat, a piano key being struck, the echo from an object detected by a sonar system [19], amongst many other types of signals. I have implemented a class that represents the causal signals named `DrumLikePulse` (these signals when listened to, remind me of percussive beats), that initialises with parameters describing the length of preceding silence, the length of the pulse in time, the features of the pulse (e.g. initial amplitude, rate of decay, etc), and the length of subsequent silence.

4.4.2 Determining the pulse train

The approach chosen for determining the time at which the event occurs from the wavelet transform is based on the 'pulse train' similar to that described by Cheng et al. in [18]. This involves the thresholding of the local energy function described by Smith in [17] as:

$$E(t) = \sqrt{\left[\sum_{n}^{N} \Re[W_s(t,n)]\right]^2 + \left[\sum_{n}^{N} \Im[W_s(t,n)]\right]^2} \tag{4.3}$$

Where N is the number of scales, and $\Re[x]$ and $\Im[x]$ are the real and imaginary outputs of the wavelet transform. The pulse train can thus be defined as:

$$P(t) = \begin{cases} 1, & \text{for } E(t) \geq \bar{g} \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

Where $\bar{g}$ is the average of $E(t)$.

With this approach, a metric for causality can be contrived by subtracting the time at which the signal starts from the time at which the first pulse in the pulse train occurs. A positive value will indicate that the pulse train started after the signal did, a value of zero will indicate that the pulse train and the start of the signal coincide, and a negative value will indicate that the pulse train started before the signal.

This has been implemented using a class named `CausalityAnalysis`, that intialises with a `ContinuousWaveletTransform` object as its single parameter. This calculates the energy function as specified in 4.3 using a funcion called `energy_function`. The pulse train is then calculated as per the function 4.4 by the function `get_pulse_train`. This allows for the metric as described above to be calculated using a function named get_delay. In figure 4.7 the delay for the continuous wavelet transform using the symmetric Morlet wavelet was calculated to be -4.144, and the delay for continuous wavelet transform performed using the asymmetric causal wavelet was calculated to be 0.3440, thus showing that the causal wavelet has the desired properties when contrasted with symmetric wavelets.

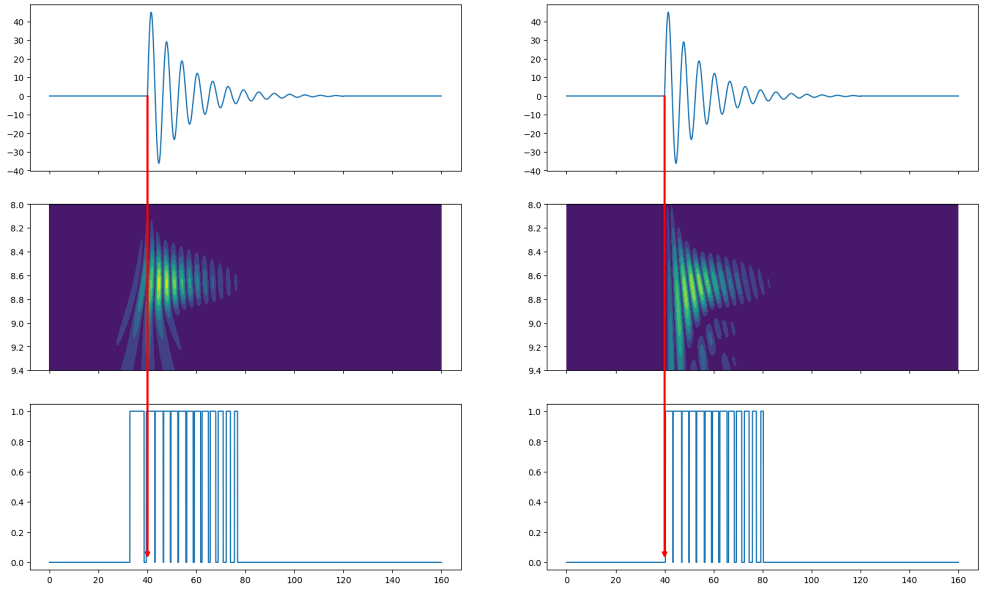

Figure 4.7: From top to bottom, the causal signal, its continuous wavelet transform as represented by a contour scalogram, and the pulse train as derived from the continuous wavelet transform. The continuous wavelet transform on the left was performed using a Morlet wavelet, and the one on the right is performed using the causal wavelet as specified in 2.9.

4.4.3 Evaluation of the causality metric

Though I have concerns about the effectiveness of this metric, it does serve as a rough metric that can be said to fulfil requirement C_1. My primary concern with this metric relate to the use of the pulse train, since its use in the method is essentially to represent a belief about when the wavelet transform *starts*. This is to say that a wavelet transform will not be considered to start before its amplitude reaches a certain threshold. This makes less sense in the pure scenario, where the signal is zero before an event occurs and thus the wavelet transform can be zero until it is considered to have started. However, in the context of real signals, noise in the signal will result in non-zero values in the transform, necessitating the use of some threshold. This thresholding introduces error, a threshold that is too low will represent the start of the transform as having occured earlier and a threshold that is too high, will set the start of the transform as having happened too late. There might be more intelligent ways of deciding upon a threshold than the simple use of the average that has been employed.

4.5 Implementing a Metric for Distortion

4.5.1 How distortion might be interpreted

When setting the requirement C_2 in table 3.3, I was aware that it would be difficult to devise a term that adequately represents the distortion present in the wavelet transform. As mentioned in the discussion of figure 4.6, the distortion appears to be a shifting in time of the wavelet transform dependent on the scale. Looking at figure 4.8, it appears that the sort of distortion generated by the type of causal wavelet I have been using results in higher frequency coefficients being represented earlier and lower frequency components being represented later. Therefore, one possible measurement of distortion might be the average angle (from a straight vertical line) which occurs in the coefficients when represented in this way.

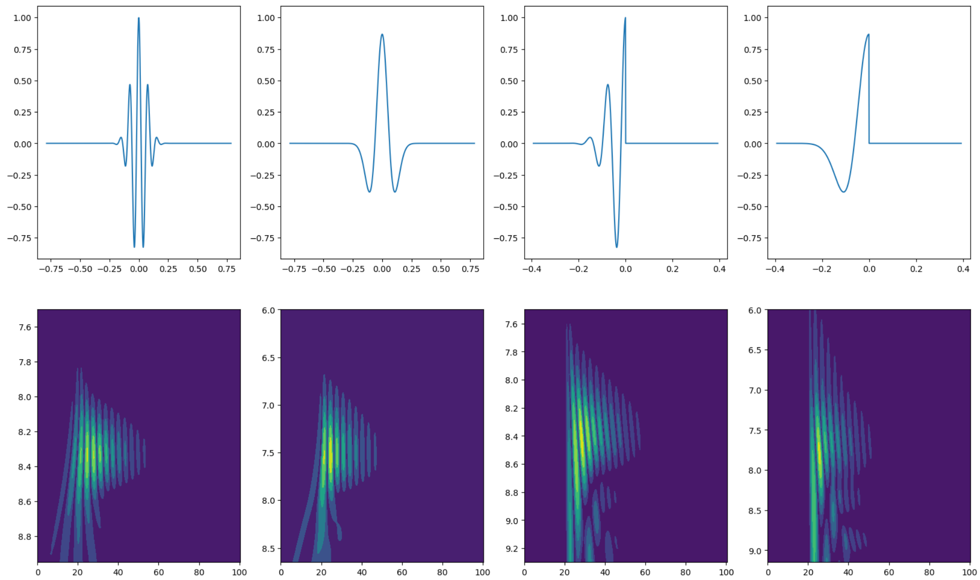

Figure 4.8: A comparison of the wavelet scalograms between Morlet and Ricker wavelets and their adpated causal counterparts, where each scalogram corresponds to the wavelet depicted above it. Reading Morlet, Ricker, causal Morlet, causal Ricker from left to right.

4.5.2 Implementing a measurement based on angle

Implementing this approach presented a number of difficulties, chief amongst which was finding a repeatable algorithm for finding the angle. The method I settled on is applied using a class called `DistortionAnalysis`, which has a function named `get_lines`, taking as its parameters two scales and a range. The function finds all the local maxima of the wavelet transforms at the two scales that it has been provided, and matches pairs of maxima between the scales within the given range, returning lines (as pairs of coordinates) that can be used to represent the distortion. Figure 4.9 shows the results of this function by plotting the results of this analysis over the contour scalogram.

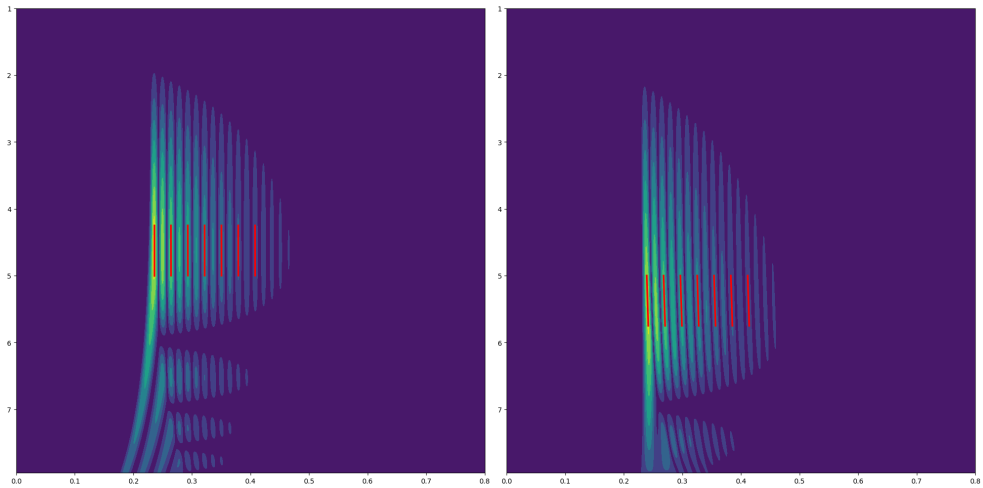

Figure 4.9: Comparison of a wavelet transform performed with a Ricker wavelet and the causal adaption of the Ricker wavelet (as represented by the second and fourth wavelets in 4.8), and the lines between associated maxima of two specified scales that have been determined as per the `get_lines` function.

The angle can be calculated by treating these lines as vectors and taking the angle between this line and the unit vector $[0, -1]$. It is important to note that the representation of scales here is logarithmic and also that the image representation is not a one-to-one scale representation of the axes. The only way this metric might be used is comparatively, since it is largely dependent on a host of factors, and even then, it should only be used to compare wavelet transforms that have been performed on the same signal, over the same set of scales, with the same x-dimensions. The angles determined for the analysis in 4.9 are $0.00490°$ for the Ricker wavelet and $0.00681°$ for the adapted causal Ricker wavelet, thus showing that the causal wavelet does create more distortion than the symmetric wavelet. Note that the angles are dependent on the scales employed in the x and y axis.

4.5.3 Evaluation of the distortion metric

Though, to some extent, this metric fulfils requirement C_2, I am not satsified with it. Given the implementation I have constructed, I do not believe that my code performs my specified method in the best way possible, or that my specified method (i.e. finding vectors that to some degree represent the distortion) is necessarily a good way of measuring distortion. The functions used rely heavily on user specified parameters, some of which, for the purposes of making this a more objective measurement, should be determined as expressions of the transform rather than arbitrarily. The method by which local maxima are associated is certainly not a reliable one, since it is just by proximity along the x-axis within a specified range that maxima are matched with each other. This requires that a user can verify that the resultant lines do indeed match the distortion visible on a scalogram (hence the function plot_lines_on_scalogram). Ideally, the method would instead use the maxima in the scales between the points to determine the most probable line between them, though this consideration alone casts doubt upon the usefulness of my angle based metric, especially given that the angle alone is not guaranteed to represent such a useful statistic when discussing the types of distortion present in the wavelet transform. Future work could ascertain better measurements of distortion and more reliable ways of measuring them. Despite this, I think that a nice feature of this method is that it produces a single measurement, taken from the average of the angles derived, which has at least comparative applications that make it suitable for my purposes.

4.6 Application to sound

Using the `from_wav_file` constructor function discussed in 4.2.1 the causal wavelet can be applied within the context of real audio signals.

Audio	Standard Morlet	Causal Morlet	Standard Ricker	Causal Ricker
Drum	0.00095s	0.00170s	0.00093s	0.00170s
Piano	0.01947s	0.02108s	0.01951s	0.02108s

Table 4.1: Delay as measured by the `get_delay` function described in section 4.4 using snippets of the same drum and piano excerpts displayed in figures 2.1 and 2.2.

Looking at the results in table 4.1, it appears that the causal wavelets perform as expected, with the delay between the start of the signal and

Audio	Standard Morlet	Causal Morlet	Standard Ricker	Causal Ricker
Drum	$0.00174°$	$0.0366°$	$0.00174°$	$0.00368°$
Piano	$0.00197°$	$0.00375°$	$0.00197°$	$0.00378°$

Table 4.2: Average angle of distortion as measured by the `get_angles` function in the `DistortionAnalysis` class.

the start of the pulse train being greater in all instances of causal wavelets being applied than in their symmetric counterparts.

Note that the positive values in all cases indicate that the pulse train begins after the signal does. This does not indicate that the wavelet transform begins after the signal, rather that the pulse train (as specified in 4.4) is calculated to begin after the start of the signal. With the application of a different threshold to determine the pulse train (as discussed in section 4.4.3) results for the specific values of the delay might have varied significantly. The trend of the transform performed with a causal wavelet beginning later should not vary.

Similarly, the values depicted in table 4.2 seem to confirm that there is a distortion of the transform present in the transforms performed with the causal wavelets relative to those performed using symmetric wavelets since the average angle is greater. However, as discussed in section 4.5.3 it is important to state that this measurement is untested and should not be taken as an absolute measurment of distortion, only a relative one.

4.7 Modifying the causal wavelet

Now that a system is in place that performs as the requirements specify, my aim is to apply this system to the comparsion of several causal wavelets. For this purpose I will be modifying the causal wavelet defined in equation 2.9, by changing the constant that defines the frequency of the sinusoid, which assumes a value of 5 in the case of equation 2.9, but still produces admissible wavelets when changed. By increasing the value of this constant, the wavelet tends towards a cosine function, which generally results in a transform with better frequency resolution. By decreasing the value, the wavelet tends towards a Dirac function, which generally results in a transform with better time resolution [15]. As such, the equation that will be used for the causal wavelet in this section is:

$$h(t) = \begin{cases} \cos(kt)\exp(-t/2) + j\sin(kt)\exp(-t/2), & \text{for } t \geq 0, \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Where k is the aforementioned constant.

Following my evaluation of the distortion metric in section 4.5.3, I decided that the comparison of these wavelets should use a more consistant method of selecting the scales by which to calculate angles. The method employed finds the coefficient with maximum value in the transform, selects the scale in which that coefficient occurs as a central scale, and selects equidistant scales above and below the central scale to be used for the calculation of distortion. Previously I had been thresholding the maxima that could be used in the calculation of distortion, such that only maxima above a certain amplitude would be used. This was done as a precaution to limit the effect of noise on the results. However, in setting this threshold to zero I found that the calculation of the angles improved.

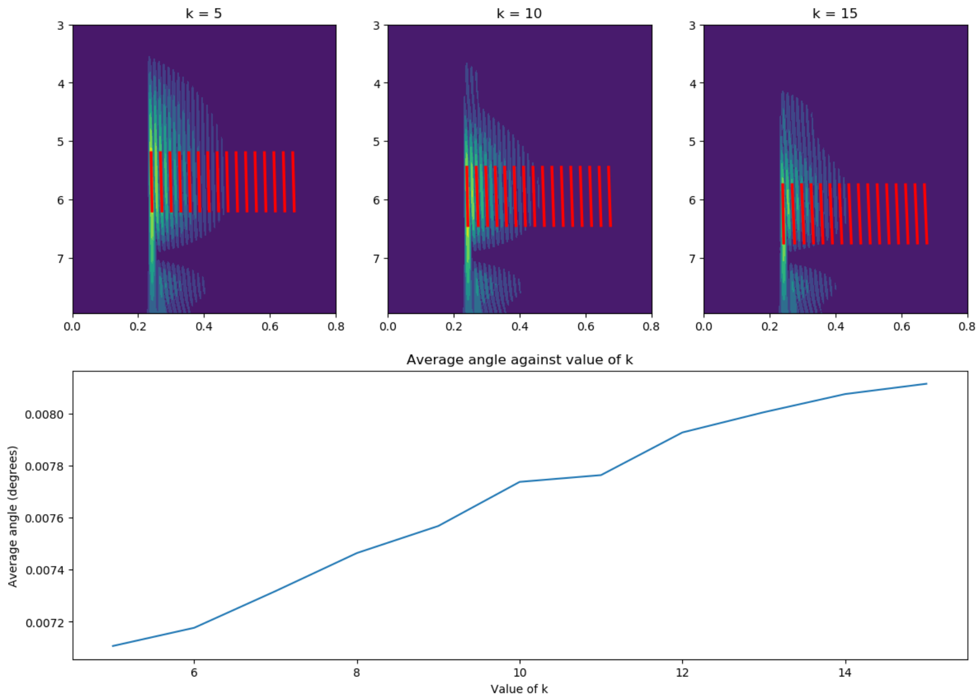

Figure 4.10: Above, real coefficients of wavelet transforms of a pulse like signal (the same as that used in figures 4.7, 4.8, and 4.9) using wavelets as defined by 4.5, with the calculated vectors between the local maxima of two scales represented by the red lines. Below, a representation of the average angle of the distortion (in degrees) against the constant used in defining the frequency of the causal wavelet used in the transform, as specified in 4.5).

When looking at figure 4.10, the obvious result of increasing the frequency of the wavelet is better localisation in the transform in frequency. However it seems that the average angle of the distortion present in the transform increases linearly as the frequency of the wavelet increases. It therefore appears that, added to the time-frequency tradeoff of the frequency of the wavelet, better frequency localisation with the causal wavelet comes with greater distortion.

4.8 Modifying the continuous wavelet transform

From the results of section 4.7, it seemed that a modification to the wavelet transform that applies a wavelet modulated in frequency as it is modulated scale could perhaps be used to rectify the distortion present in the wavelet transform performed with a causal wavelet.

Implementation In order to implement this modified wavelet transform, a modified version of the `ContinuousWaveletTransform` object named `ModifiedContinuousWaveletTransform` was used, which employs a modified `Wavelet` object named `CausalMorletModifiedWavelet`. The transform applied by this implementation differs from the previous in that the causal adaption Morlet wavelet used is modulated in frequency as well as scale, as the frequency of the wavelet is defined by the scale divided by some constant. This tampers with the continuous wavelet transform in many ways that significantly limit its applications. In the standard wavelet transform, scale and frequency are inversely proportional, but since the internal frequency of the wavelet is being modulated, the relationship between scale and frequency is no longer clear.

Analysis	Standard Morlet	Causal Morlet	Modified Causal Morlet
Delay	-0.00622s	0.00155s	0.00114s
Average angle	$0.00084°$	$0.00697°$	$0.00281°$

Table 4.3: Delay measured using the `CausalityAnalysis` and average angle of as measured by the `get_angles` function in the `DistortionAnalysis` class of the transforms seen in figure A.2.

This modification to the transform, when performed within similar scales, seems to have the effect of minimising the distortion, while maintaining the causal behaviour, as can be seen in figure A.2 and table 4.3. However, this improvement comes at the cost of confusing the relationship between scale and frequency, which significantly impacts the analytical applications of this method. Also, when increasing the scale past this point, it becomes apparent that there is some other, perhaps more significant form of distortion present.

5 Conclusions

Throughout this report I have discussed my progress in terms of the requirements I set out in chapter 3. This has been a useful method for dividing up the task into achievable parts. The purpose of the requirements was to be restrictive and therefore measurable, and for the most part I feel that all of them have been met, that the methods by which they have been met have been described, and that these methods have been evaluated in terms of their abilities and their possible deficiencies in the fulfilment of the requirements. I feel that all requirements listed performed well as requirements, though if better requirements might have been in order, then this should have been explicitly stated. New requirements can be inferred for future work from the evaluation in these subsequent paragraphs. For this reason, in this chapter I will evaluate the system in more general terms than the requirements.

Implementation of the continuous wavelet transform I am mostly satisfied with the implementation of the continuous wavelet transform. Multiple environments and algorithms for the transform were tested before arriving at this implementation. The results achieved and the plots produced using the implementation matched my expectations, and the object orientated structure allowed the flexibility to prototype and experiment with different wavelets in an expedient fashion. However, this flexibility is in some senses a potential problem. The lack of testing within the implementation allowed for fast application of irregular wavelets, but provided no assurances about the viabilty of the output. Equally, the lack of useful error messages ensures that any other user is going to have little idea of why the system might break. Were development continued, even for just for personal use, at least a few tests and meaningful error messages would need to be implemented.

Implementation of metrics In some sense I am satisified with the implementation of the metrics describing the causality of the transform with relation to the signal and the distortion within the transform, provided that they are used in the context of comparision for transforms of the same signals. I have largely discussed this in sections 4.4.3 and 4.5.3, but I do not believe these methods necessarily make objective representations of what they represent. The use of the method whereby the scales used in the calculation of the distortion metric were selected, as in sections 4.7 and

4.8, as scales equidistant from the scale containing the greatest coefficient, proved a decent method of making the calculation of the metric a little more reliable. Nonetheless, future work could focus on finding more objective methods for these types of measurements. It would also be useful to have a method for calculating the inherent uncertainties in the results found.

Application to sound This chapter can be imagined as a proof of the implementation. It is the application of the theory explored in the previous sections and the system developed throughout chapter 4 to the real instances of audio signals. This chapter does, to some extent, confirm that the use of causal wavelets in a wavelet transform creates the predicted effects of causality and distortion in the transform. In all cases, the wavelet transform of the audio as performed with a causal wavelet can be measured to begin later than the transform as performed with a symmetric wavelet. In all cases, the wavelet tranform performed with a causal wavelet appeared to have a greater average angle of distortion than that of the transform as performed with a symmetric wavelet. A more extensive analysis and comparison of the transforms of these audio files would have been interesting, but the application here does serve the purpose of demonstrating the viabilty of these sorts of causal wavelet transforms for the analysis of real signals.

Modifying the causal wavelet and continuous wavelet transform The approach in sections 4.7 and 4.8 was perhaps not the optimal approach that could have been taken to solving the problem. There are rather significant issues with the solution that was arrived at, despite its seeming to perform well by the casuality and distortion metrics. Most notable is its seeming failure to properly localise the signal in the frequency domain. Although imperfect, the method employed does seem to fulfil its goal of correcting the distortion within a certain set of scales, though I am inclined to doubt its suitabilty for the purpose of analysis. Future work could verify if this method is reliable and, if so, what adpations of it produce the best results.

A Appendix:Large Figures

This chapter is for large figures that I have elected not to represent in the body of the report.

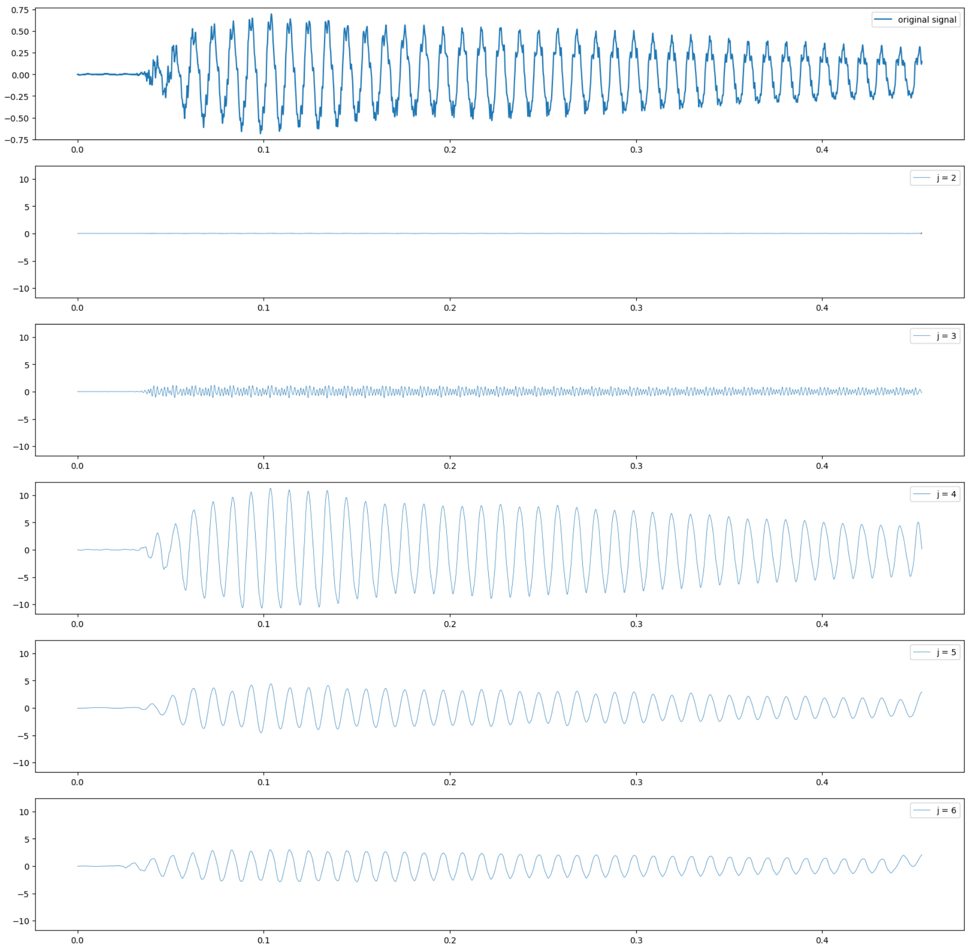

Figure A.1: Wavelet transform coefficients from the `peform_transform` function on a fraction of an audio signal at the dyadic scales 2^j.

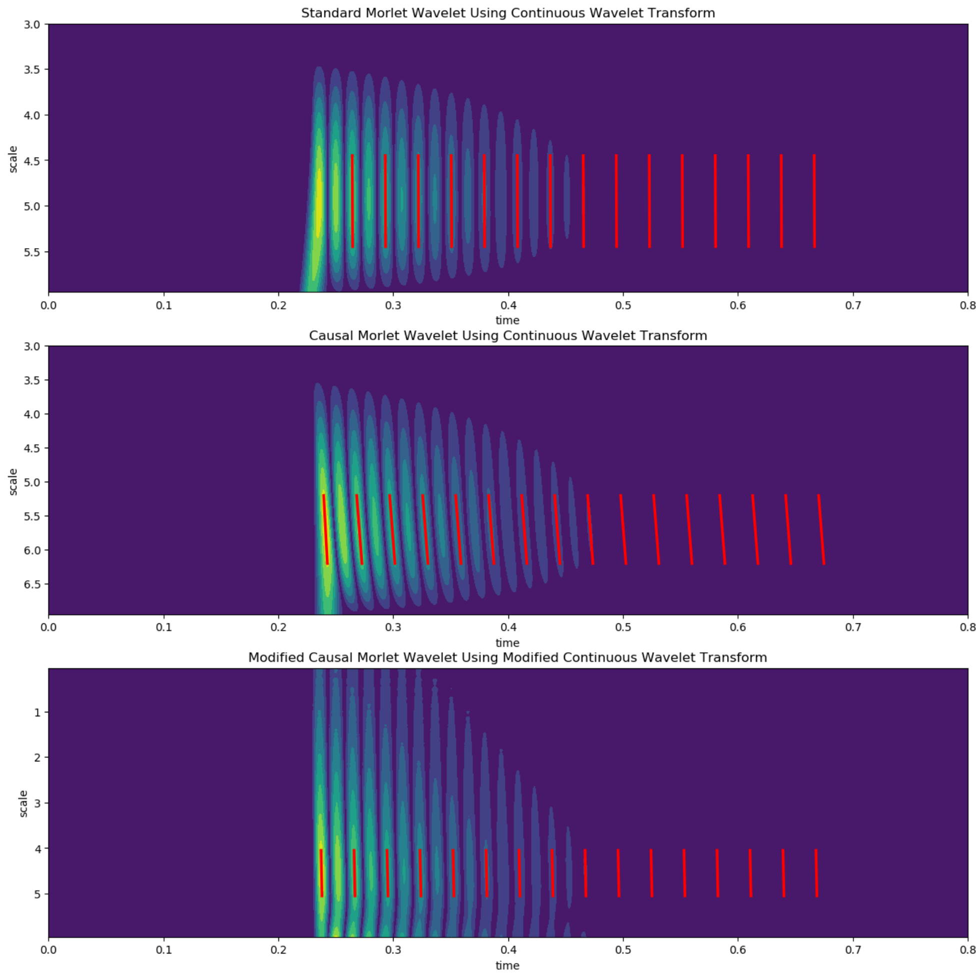

Figure A.2: Comparison of scalograms created by wavelet transform of a pulse using a symmetric Morlet wavelet, the causal adaption of the Morlet wavelet, and a modified causal wavelet transform of the modified causal wavelet (top to bottom).

Bibliography

[1] A. Graps, 'An introduction to wavelets', *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1995. DOI: 10.1109/99.388960. [Online]. Available: https://ieeexplore.ieee.org/document/388960 (visited on 17/03/2020).

[2] B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*. Jan. 2003, p. 5. [Online]. Available: https://www.researchgate.net/publication/43517721_Time-Frequency_Signal_Analysis_and_Processing_A_Comprehensive_Reference (visited on 17/03/2020).

[3] R. N. Bracewell, *The Fourier Transform and its applications*, Third, ser. Electrical Engineering Series. McGraw-Hill, 1986.

[4] D. Gabor, 'Theory of communication. part 1: The analysis of information', *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946. DOI: 10.1049/ji-3-2.1946.0074. [Online]. Available: https://ieeexplore.ieee.org/document/5298517 (visited on 15/03/2020).

[5] P. N. S. Network, *What is a spectrogram?* [Online]. Available: https://pnsn.org/spectrograms/what-is-a-spectrogram (visited on 31/03/2020).

[6] T. Strohmer, *Local time-frequency analysis and short time fourier transform*. [Online]. Available: https://www.math.ucdavis.edu/~strohmer/research/gabor/gaborintro/node3.html (visited on 01/04/2020).

[7] I. Daubechies, 'Where do wavelets come from? a personal point of view', *Proceedings of the IEEE*, vol. 84, no. 4, pp. 510–513, 1996. DOI: 10.1109/5.488696. [Online]. Available: https://ieeexplore.ieee.org/document/488696 (visited on 20/03/2020).

[8] I. Daubechies, *Ten lectures on wavelets*. Siam, 1992, vol. 61, p. 3. DOI: 10.1137/1.9781611970104.

[9] O. Rioul and P. Duhamel, 'Fast algorithms for discrete and continuous wavelet transforms', *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 569–586, 1992. DOI: 10.1109/18.119724. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=119724 (visited on 10/04/2020).

[10] Z. Peng, F. Chu and Y. He, 'Vibration signal analysis and feature extraction based on reassigned wavelet scalogram', *Journal of Sound and Vibration*, vol. 253, no. 5, pp. 1087–1100, 2002.

[11] C. Valens, 'A really friendly guide to wavelets', 1999. [Online]. Available: https://www.researchgate.net/publication/2467463_A_Really_Friendly_Guide_to_Wavelets (visited on 30/04/2020).

[12] H. Karimi, B. Moshiri, B. Lohmann and P. J. Maralani, 'Haar wavelet-based approach for optimal control of second-order linear systems in time domain', *Journal of Dynamical and Control Systems*, vol. 11, no. 2, pp. 237–252, 2005. DOI: 10.1007/s10883-005-4172-z. [Online]. Available: https://link.springer.com/article/10.1007/s10883-005-4172-z (visited on 04/04/2020).

[13] Y. Meyer, 'Wavelets-algorithms and applications', *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p.*, pp. 14–15, 1993.

[14] G. Strang, 'Wavelets', *American Scientist*, vol. 82, no. 3, pp. 250–255, 1994. [Online]. Available: https://www.jstor.org/stable/29775194.

[15] J. LIN and L. QU, 'Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis', *Journal of Sound and Vibration*, vol. 234, no. 1, pp. 135–148, 2000. DOI: https://doi.org/10.1006/jsvi.2000.2864. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022460X00928645 (visited on 07/04/2020).

[16] H. Ryan, 'Ricker, ormsby; klander, bntterwo-a choice of wavelets', 1994. [Online]. Available: https://csegrecorder.com/articles/view/ricker-ormsby-klander-butterworth-a-choice-of-wavelets (visited on 30/04/2020).

[17] L. M. Smith, 'Modelling rhythm perception by continuous time-frequency analysis', in *ICMC*, 1996. [Online]. Available: https://www.semanticscholar.org/paper/Modelling-Rhythm-Perception-by-Continuous-Analysis-Smith/1bf126eae72dfadf85eaa5f94bf08e0d971e02b4 (visited on 10/04/2020).

[18] X. Cheng, J. V. Hart and J. S. Walker, 'Time-frequency analysis of musical rhythm', *Notices of the AMS*, vol. 56, no. 3, pp. 256–372, 2009. [Online]. Available: https://www.researchgate.net/publication/241146028_Time-frequency_Analysis_of_Musical_Rhythm (visited on 05/04/2020).

[19] H. H. Szu, B. A. Telfer and A. W. Lohmann, 'Causal analytical wavelet transform', *Optical Engineering*, vol. 31, no. 9, pp. 1825–1829, 1992. DOI: 10.1117/12.59911. [Online]. Available: https://doi.org/10.1117/12.59911 (visited on 30/03/2020).

[20] W. R. Inc., *Wavelet analysis*, Champaign, IL, 2020. [Online]. Available: http://reference.wolfram.com/language/guide/Wavelets.html (visited on 25/03/2020).

[21] MATLAB, *Wavelet toolbox*, Natick, Massachusetts, 2020. [Online]. Available: https://www.mathworks.com/products/wavelet.html (visited on 25/03/2020).

[22] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt and A. O'Leary, 'Pywavelets: A python package for wavelet analysis', *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019. DOI: 10.21105/joss.01237. [Online]. Available: https://doi.org/10.21105/joss.01237 (visited on 01/04/2020).

[23] M. Stéphane, *A Wavelet Tour of Signal Processing (Third Edition)*, Third Edition. Boston: Academic Press, 2009, pp. 170–172, ISBN: 978-0-12-374370-1. DOI: https://doi.org/10.1016/B978-0-12-374370-1.00009-4.

[24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python', *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: https://doi.org/10.1038/s41592-019-0686-2. [Online]. Available: https://rdcu.be/b08Wh (visited on 12/04/2020).