THE WAVELET TUTORIAL

SECOND EDITION

PART I

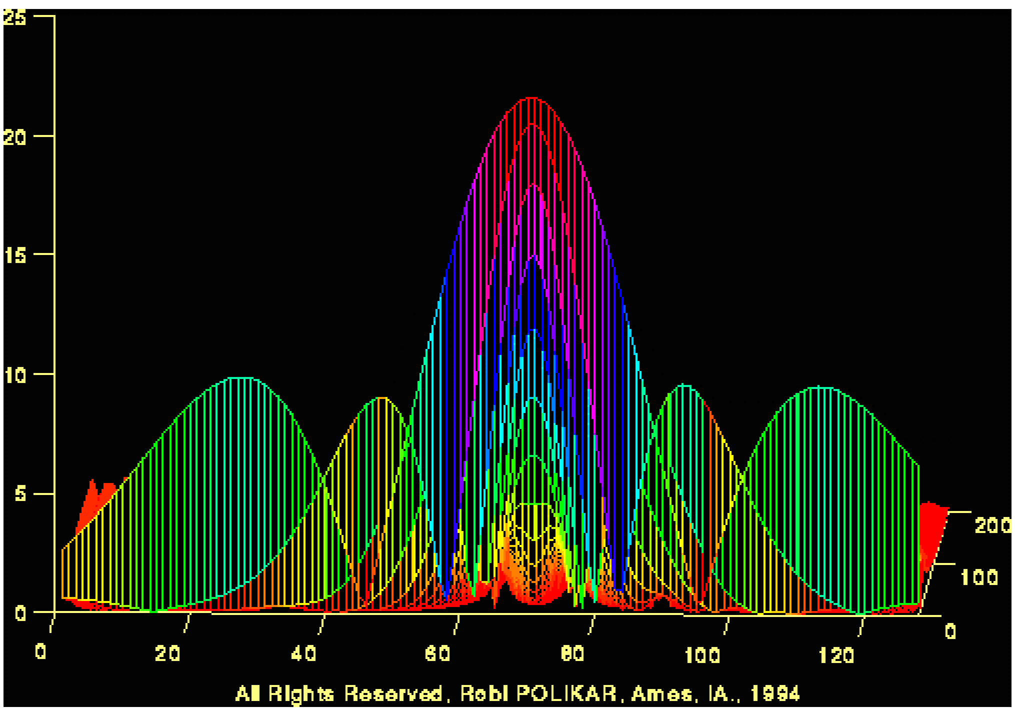
BY

ROBI POLIKAR

FUNDAMENTAL CONCEPTS

&

AN OVERVIEW OF THE WAVELET THEORY



Welcome to this introductory tutorial on wavelet transforms. The wavelet transform is a relatively new concept (about 10 years old), but yet there are quite a few articles and books written on them. However, most of these books and articles are written by math people, for the other math people; still most of the math people don't know what the other math people are talking about (a math professor of mine made this confession). In other words, majority of the literature available on wavelet transforms are of little help, if any, to those who are new to this subject (this is my personal opinion).

When I first started working on wavelet transforms I have struggled for many hours and days to figure out what was going on in this mysterious world of wavelet transforms, due to the lack of introductory level text(s) in this subject. Therefore, I have decided to write this tutorial for the ones who are new to the topic. I consider myself quite new to the subject too, and I have to confess that I have not figured out all the theoretical details yet. However, as far as the engineering applications are concerned, I think all the theoretical details are not necessarily necessary (!).

In this tutorial I will try to give basic principles underlying the wavelet theory. The proofs of the theorems and related equations will not be given in this tutorial due to the simple assumption that the intended readers of this tutorial do not need them at this time. However, interested readers will be directed to related references for further and in-depth information.

In this document I am assuming that you have no background knowledge, whatsoever. If you do have this background, please disregard the following information, since it may be trivial.

Should you find any inconsistent, or incorrect information in the following tutorial, please feel free to contact me. I will appreciate any comments on this page.

Robi POLIKAR

****

TRANS... WHAT?

First of all, why do we need a transform, or what is a transform anyway?

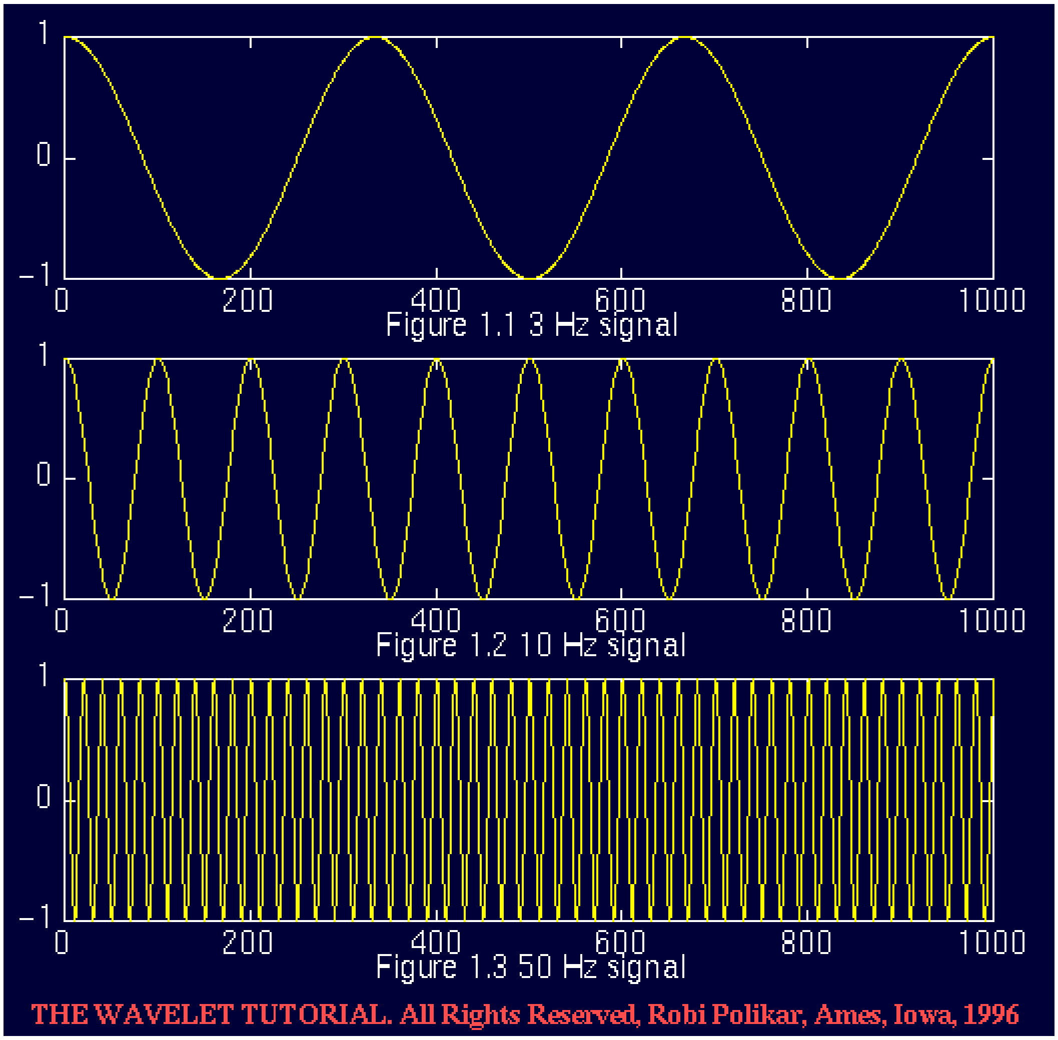
Mathematical transformations are applied to signals to obtain further information from that signal that is not readily available in the raw signal. In the following tutorial I will assume a time-domain signal as a **raw** signal, and a signal that has been "transformed" by any of the available mathematical transformations as a **processed** signal.

There are a number of transformations that can be applied, among which the Fourier transforms are probably by far the most popular.

Most of the signals in practice, are **TIME-DOMAIN** signals in their raw format. That is, whatever that signal is measuring, is a function of time. In other words, when we plot the signal one of the axes is time (independent variable), and the other (dependent variable) is usually the amplitude. When we plot time-domain signals, we obtain a **time-amplitude representation** of the signal. This representation is not always the best representation of the signal for most signal processing related applications. In many cases, the most distinguished information is hidden in the frequency content of the signal. The **frequency SPECTRUM** of a signal is basically the frequency components (spectral components) of that signal. The frequency spectrum of a signal shows what frequencies exist in the signal.

Intuitively, we all know that the frequency is something to do with the change in rate of something. If something (a mathematical or physical variable, would be the technically correct term) changes rapidly, we say that it is of high frequency, where as if this variable does not change rapidly, i.e., it changes smoothly, we say that it is of low frequency. If this variable does not change at all, then we say it has zero frequency, or no frequency. For example the publication frequency of a daily newspaper is higher than that of a monthly magazine (it is published more frequently).

The frequency is measured in cycles/second, or with a more common name, in "Hertz". For example the electric power we use in our daily life in the US is 60 Hz (50 Hz elsewhere in the world). This means that if you try to plot the electric current, it will be a sine wave passing through the same point 50 times in 1 second. Now, look at the following figures. The first one is a sine wave at 3 Hz, the second one at 10 Hz, and the third one at 50 Hz. Compare them.



So how do we measure frequency, or how do we find the frequency content of a signal? The answer is **FOURIER TRANSFORM (FT)**. If the FT of a signal in time domain is taken, the frequency-amplitude representation of that signal is obtained. In other words, we now have a plot with one axis being the frequency and the other being the amplitude. This plot tells us how much of each frequency exists in our signal.

The frequency axis starts from zero, and goes up to infinity. For every frequency, we have an amplitude value. For example, if we take the FT of the electric current that we use in our houses, we will have one spike at 50 Hz, and nothing elsewhere, since that signal has only 50 Hz frequency component. No other signal, however, has a FT which is this simple. For most practical purposes, signals contain more than one frequency component. The following shows the FT of the 50 Hz signal:

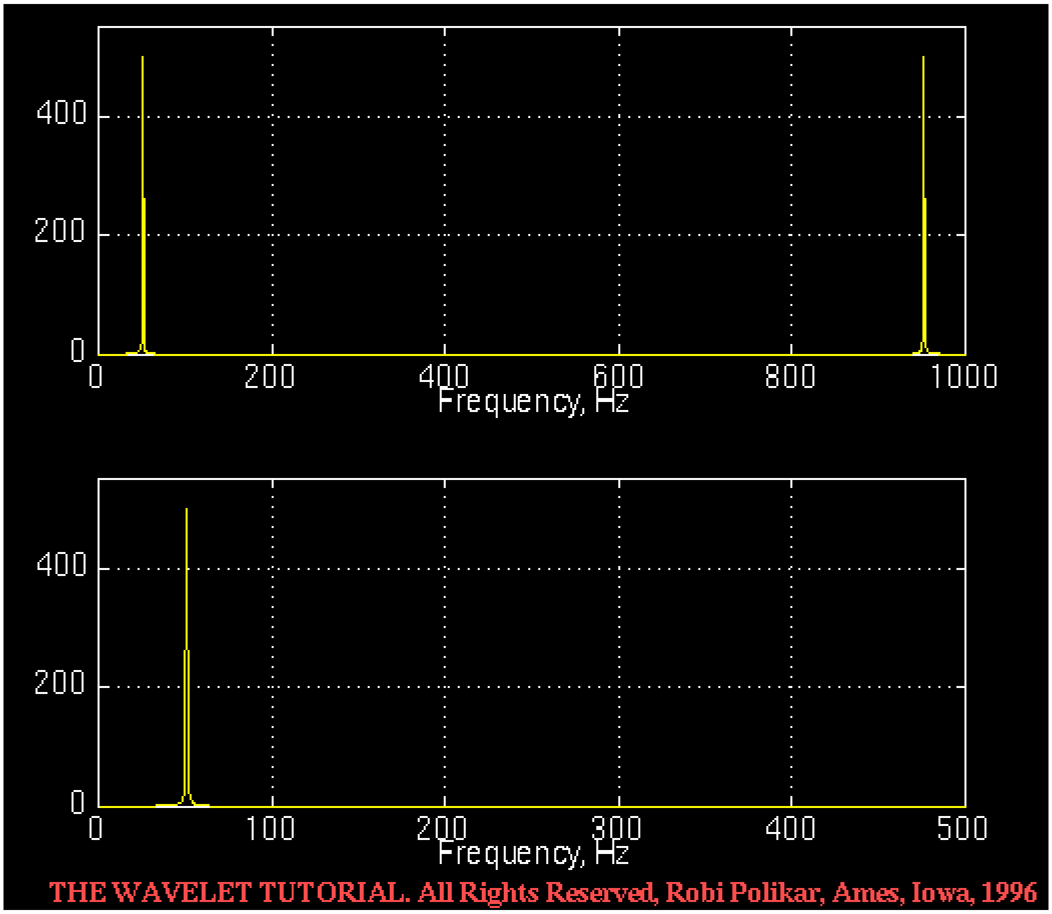


Figure 1.4 The FT of the 50 Hz signal given in Figure 1.3

One word of caution is in order at this point. Note that two plots are given in Figure 1.4. The bottom one plots only the first half of the top one. Due to reasons that are not crucial to know at this time, the frequency spectrum of a real valued signal is always symmetric. The top plot illustrates this point. However, since the symmetric part is exactly a mirror image of the first part, it provides no additional information. And therefore, this symmetric second part is usually not shown. In most of the following figures corresponding to FT, I will only show the first half of this symmetric spectrum.

WHY DO WE NEED THE FREQUENCY INFORMATION?

Often times, the information that cannot be readily seen in the time-domain can be seen in the frequency domain.

Let's give an example from biological signals. Suppose we are looking at an ECG signal (ElectroCardioGraphy, graphical recording of heart's electrical activity). The typical shape of a healthy ECG signal is well known to cardiologists. Any significant deviation from that shape is usually considered to be a symptom of a pathological condition.

This pathological condition, however, may not always be quite obvious in the original time-domain signal. Cardiologists usually use the time-domain ECG signals which are recorded on strip-charts to analyze ECG signals. Recently, the new computerized ECG recorders/analyzers also utilize the frequency information to decide whether a pathological condition exists. A pathological condition can sometimes be diagnosed more easily when the frequency content of the signal is analyzed.

This, of course, is only one simple example why frequency content might be useful. Today Fourier transforms are used in many different areas including all branches of engineering.

Although FT is probably the most popular transform being used (especially in electrical engineering), it is not the only one. There are many other transforms that are used quite often by engineers and mathematicians. Hilbert transform, short-time Fourier transform (more about this later), Wigner distributions, the Radon Transform, and of course our **featured transformation**, the wavelet transform, constitute only a small portion of a huge list of transforms that are available at engineer's and mathematician's disposal. Every transformation technique has its own area of application, with advantages and disadvantages, and the wavelet transform (WT) is no exception.

For a better understanding of the need for the WT, let's look at the FT more closely. FT (as well as WT) is a reversible transform, that is, it allows to go back and forward between the raw and processed (transformed) signals. However, only either of them is available at any given time. That is, no frequency information is available in the time-domain signal, and no time information is available in the Fourier transformed signal. The natural question that comes to mind is that is it necessary to have both the time and the frequency information at the same time?

As we will see soon, the answer depends on the particular application, and the nature of the signal in hand. Recall that the FT gives the frequency information of the signal, which means that it tells us how much of each frequency exists in the signal, but it does not tell us **when in time** these frequency components exist. This information is not required when the signal is so-called **stationary**.

Let's take a closer look at this **stationarity** concept more closely, since it is of paramount importance in signal analysis. Signals whose frequency content do not change in time are called **stationary signals**. In other words, the frequency content of stationary signals do not change in time. In this case, one does not need to know **at what times frequency components exist** , since **all frequency components exist at all times !!!** .

For example the following signal

x(t)=cos(2*pi*10*t)+cos(2*pi*25*t)+cos(2*pi*50*t)+cos(2*pi*100*t)

is a stationary signal, because it has frequencies of 10, 25, 50, and 100 Hz at any given time instant. This signal is plotted below:

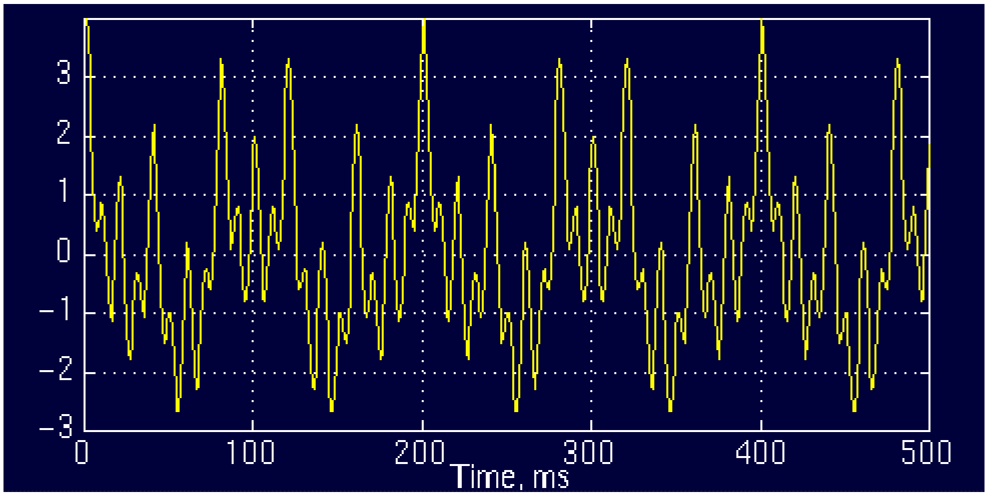


Figure 1.5

And the following is its FT:

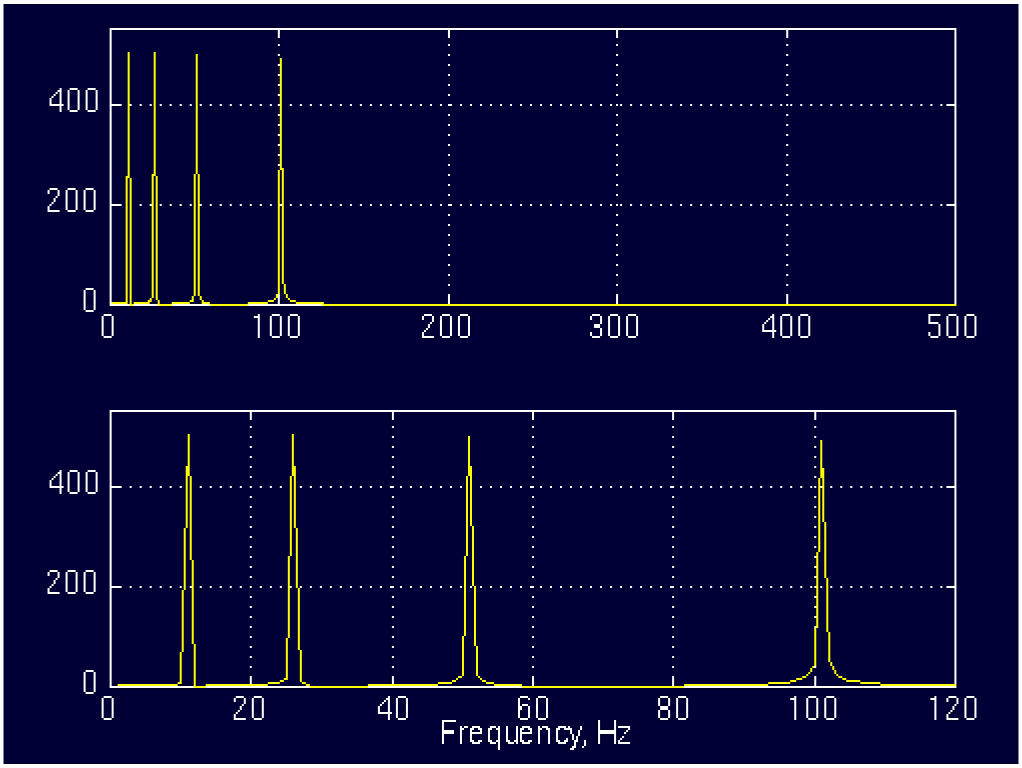


Figure 1.6

The top plot in Figure 1.6 is the (half of the symmetric) frequency spectrum of the signal in Figure 1.5. The bottom plot is the zoomed version of the top plot, showing only the range of frequencies that are of interest to us. Note the four spectral components corresponding to the frequencies 10, 25, 50 and 100 Hz.

Contrary to the signal in Figure 1.5, the following signal is not stationary. Figure 1.7 plots a signal whose frequency constantly changes in time. This signal is known as the "chirp" signal. This is a non-stationary signal.

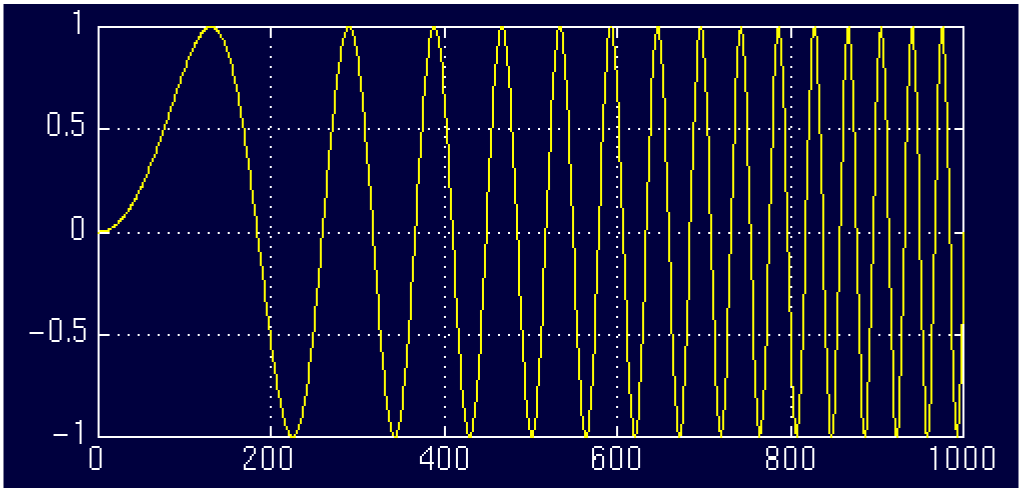


Figure 1.7

Let's look at another example. Figure 1.8 plots a signal with four different frequency components at four different time intervals, hence a non-stationary signal. The interval 0 to 300 ms has a 100 Hz sinusoid, the interval 300 to 600 ms has a 50 Hz sinusoid, the interval 600 to 800 ms has a 25 Hz sinusoid, and finally the interval 800 to 1000 ms has a 10 Hz sinusoid.

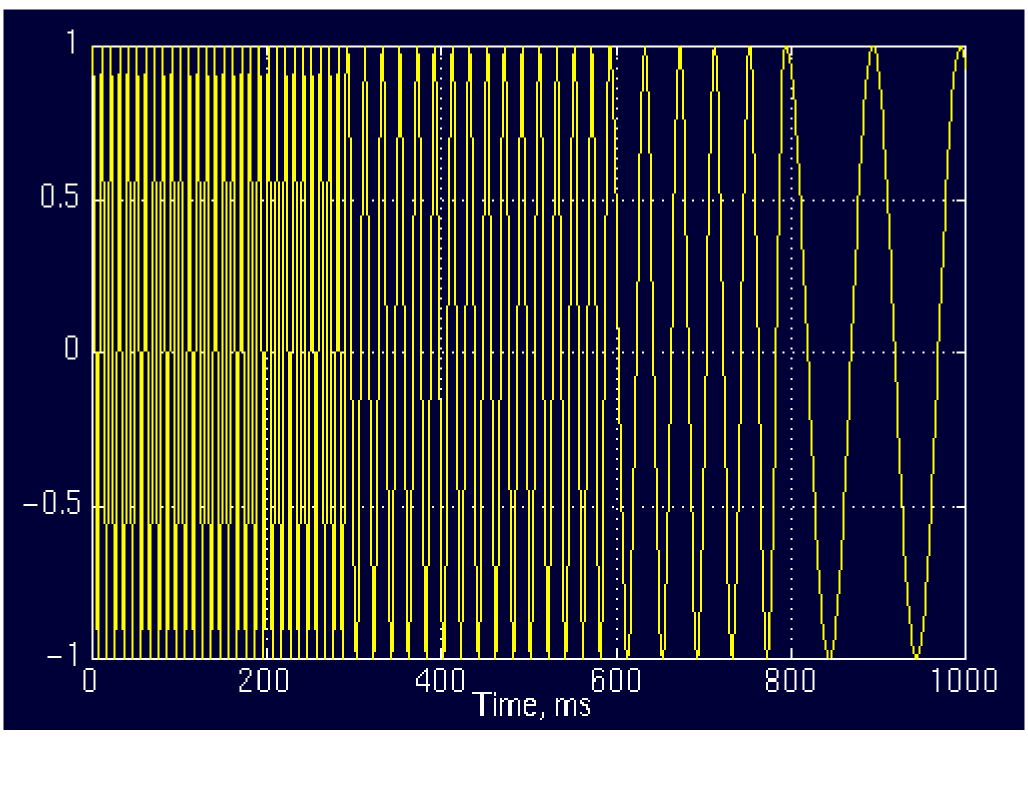


Figure 1.8

And the following is its FT:

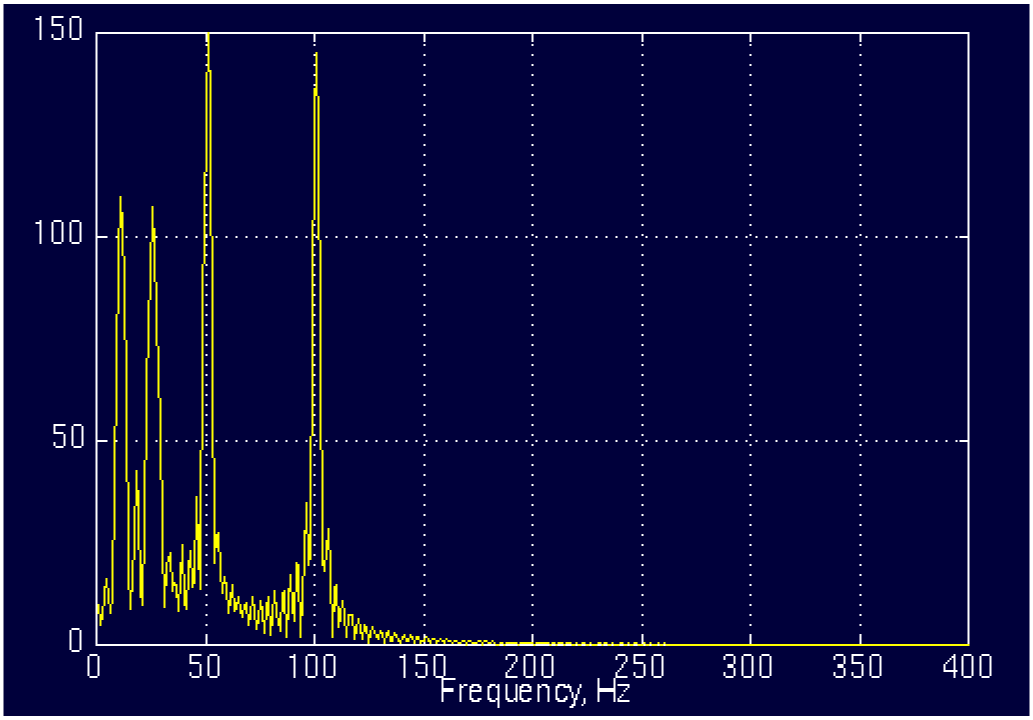


Figure 1.9

Do not worry about the little ripples at this time; they are due to sudden changes from one frequency component to another, which have no significance in this text. Note that the amplitudes of higher frequency components are higher than those of the lower frequency ones. This is due to fact that higher frequencies last longer (300 ms each) than the lower frequency components (200 ms each). (The exact values of the amplitudes are not important).

Other than those ripples, everything seems to be right. The FT has four peaks, corresponding to four frequencies with reasonable amplitudes... Right

WRONG (!)

Well, not exactly wrong, but not exactly right either... Here is why:

For the first signal, plotted in Figure 1.5, consider the following question:

At what times (or time intervals), do these frequency components occur?

Answer:

At all times! Remember that in stationary signals, all frequency components that exist in the signal, exist throughout the entire duration of the signal. There is 10 Hz at all times, there is 50 Hz at all times, and there is 100 Hz at all times.

Now, consider the same question for the non-stationary signal in Figure 1.7 or in Figure 1.8.

At what times these frequency components occur?

For the signal in Figure 1.8, we know that in the first interval we have the highest frequency component, and in the last interval we have the lowest frequency component. For the signal in Figure 1.7, the frequency components change continuously. Therefore, for these signals the frequency components **do not** appear at all times!

Now, compare the Figures 1.6 and 1.9. The similarity between these two spectrums should be apparent. Both of them show four spectral components at exactly the same frequencies, i.e., at 10, 25, 50, and 100 Hz. Other than the ripples, and the difference in amplitude (which can always be normalized), the two spectrums are almost identical, although the corresponding time-domain signals are not even close to each other. Both of the signals involve the same frequency components, but the first one has these frequencies at all times, the second one has these frequencies at different intervals. So, how come the spectrums of two entirely different signals look very much alike? Recall that the FT gives the spectral content of the signal, but it gives no information regarding **where in time those spectral components appear**. Therefore, FT is not a suitable technique for non-stationary signal, with one exception: FT can be used for non-stationary signals, if we are only interested in what spectral components exist in the signal, but not interested where these occur. However, if this information is needed, i.e., if we want to know, what spectral component occur at what time (interval), then Fourier transform is not the right transform to use.

For practical purposes it is difficult to make the separation, since there are a lot of practical stationary signals, as well as non-stationary ones. Almost all biological signals, for example, are non-stationary. Some of the most famous ones are ECG (electrical activity of the heart, electrocardiograph), EEG (electrical activity of the brain, electroencephalograph), and EMG (electrical activity of the muscles, electromyogram).

Once again please note that, the FT gives what frequency components (spectral components) exist in the signal. No more, no less.

When the **time localization** of the spectral components are needed, a transform giving the TIME-FREQUENCY REPRESENTATION of the signal is needed.

THE ULTIMATE SOLUTION: THE WAVELET TRANSFORM

The wavelet transform is a transform of this type. It provides the time-frequency representation. (There are other transforms which give this information too, such as short time Fourier transform, Wigner distributions, etc.)

Often times a particular spectral component occurring at any instant can be of particular interest. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. For example, in EEGs, the latency of an event-related potential is of particular interest (Event-related potential is the response of the brain to a specific stimulus like flash-light, the latency of this response is the amount of time elapsed between the onset of the stimulus and the response).

Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal.

How wavelet transform works is completely a different fun story, and should be explained after **short time Fourier Transform (STFT)**. The WT was developed as an alternative to the STFT. The STFT will be explained in great detail in the second part of this tutorial. It suffices at this time to say that the WT was developed to overcome some resolution related problems of the STFT, as explained in Part II.

To make a real long story short, we pass the time-domain signal from various highpass and low pass filters, which filters out either high frequency or low frequency portions of the signal. This procedure is repeated, every time some portion of the signal corresponding to some frequencies being removed from the signal.

Here is how this works: Suppose we have a signal which has frequencies up to 1000 Hz. In the first stage we split up the signal in to two parts by passing the signal from a highpass and a lowpass filter (filters should satisfy some certain conditions, so-called **admissibility condition**) which results in two different versions of the same signal: portion of the signal corresponding to 0-500 Hz (low pass portion), and 500-1000 Hz (high pass portion).

Then, we take either portion (usually low pass portion) or both, and do the same thing again. This operation is called **decomposition** .

Assuming that we have taken the lowpass portion, we now have 3 sets of data, each corresponding to the same signal at frequencies 0-250 Hz, 250-500 Hz, 500-1000 Hz.

Then we take the lowpass portion again and pass it through low and high pass filters; we now have 4 sets of signals corresponding to 0-125 Hz, 125-250 Hz,250-500 Hz, and 500-1000 Hz. We continue like this until we have decomposed the signal to a pre-defined certain level. Then we have a bunch of signals, which actually represent the same signal, but all corresponding to different frequency bands. We know which signal corresponds to which frequency band, and if we put all of them together and plot them on a 3-D graph, we will have time in one axis, frequency in the second and amplitude in the third axis. This will show us which frequencies exist at which time (there is an issue, called "uncertainty principle", which states that, we cannot exactly know **what frequency exists** at **what time instance**, but we can only know **what frequency bands** exist at **what time intervals**, more about this in the subsequent parts of this tutorial).

However, I still would like to explain it briefly:

The uncertainty principle, originally found and formulated by Heisenberg, states that, the momentum and the position of a moving particle cannot be known simultaneously. This applies to our subject as follows:

The frequency and time information of a signal at some certain point in the time-frequency plane cannot be known. In other words: We cannot know **what spectral component** exists at any given time **instant**. The best we can do is to investigate what **spectral components** exist at any given **interval** of time. This is a problem of resolution, and it is the main reason why researchers have switched to WT from STFT. STFT gives a fixed resolution at all times, whereas WT gives a variable resolution as follows:

Higher frequencies are better resolved in time, and lower frequencies are better resolved in frequency. This means that, a certain high frequency component can be located better in time (with less relative error) than a low frequency component. On the contrary, a low frequency component can be located better in frequency compared to high frequency component.

Take a look at the following grid:

```
 f ^
  |*******************************************          continuous
  |*  *  *  *  *  *  *  *  *  *  *  *  *  *  *          wavelet transform
  |*    *    *    *    *    *    *
  |*        *        *        *
  |*                *
   ------------------------------------------>        time
```

Interpret the above grid as follows: The top row shows that at higher frequencies we have more samples corresponding to smaller intervals of time. In other words, higher frequencies can be resolved better in time. The bottom row however, corresponds to low frequencies, and there are less number of points to characterize the signal, therefore, low frequencies are not resolved well in time.

```
 ^frequency
 |
 |
 |
 | ****************************
 |
 |
 |
 | *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *       discrete time
 |                                                              wavelet transform
 | *    *    *    *    *    *    *    *    *    *
 |
 | *        *        *        *        *
 | *              *              *
 |------------------------------------------------------->        time
```

In discrete time case, the time resolution of the signal works the same as above, but now, the frequency information has different resolutions at every stage too. Note that, lower frequencies are better resolved in frequency, where as higher frequencies are not. Note how the spacing between subsequent frequency components increase as frequency increases.

Below, are some examples of continuous wavelet transform: Let's take a sinusoidal signal, which has two different frequency components at two different times:

Note the low frequency portion first, and then the high frequency.

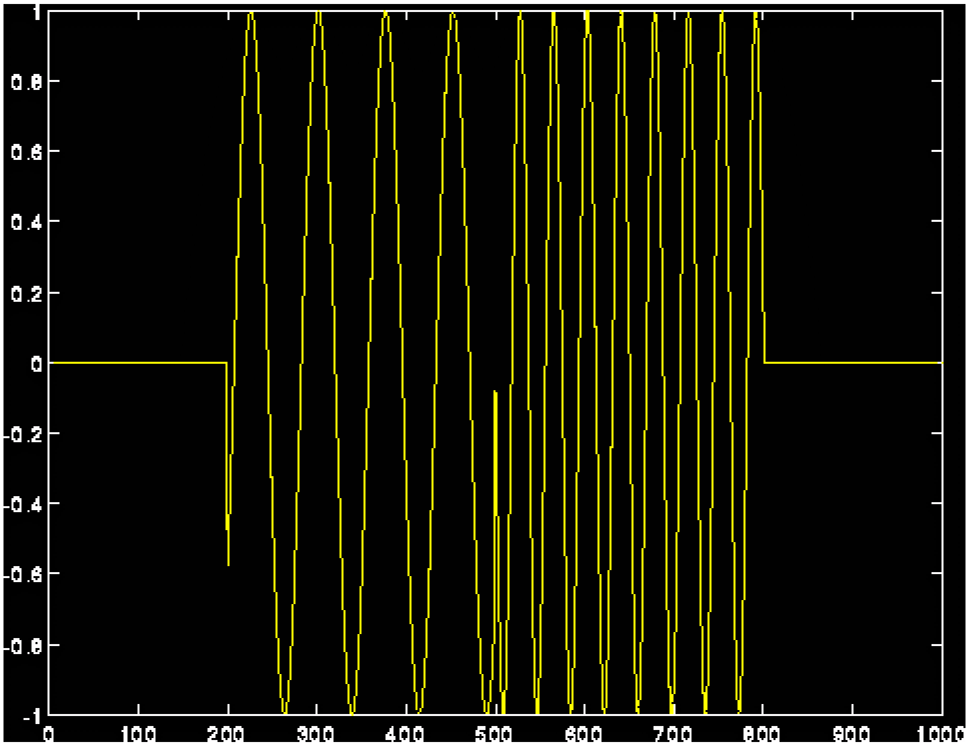


Figure 1.10

The continuous wavelet transform of the above signal:

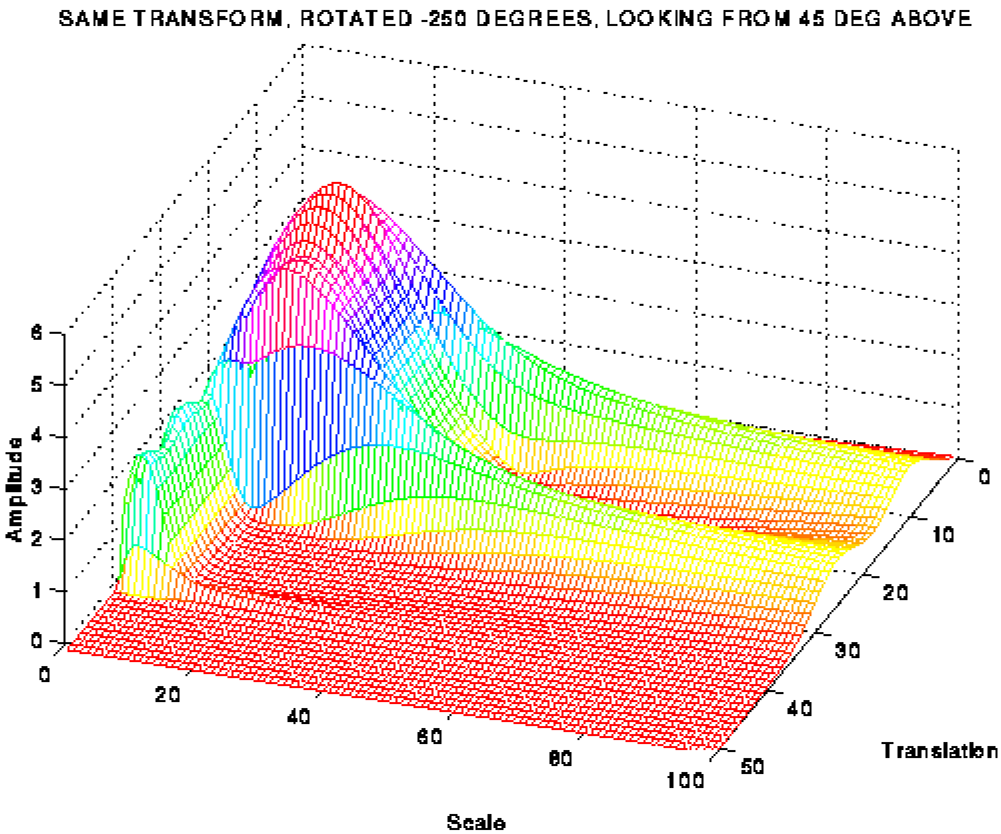


Figure 1.11

Note however, the frequency axis in these plots are labeled as **scale**. The concept of the scale will be made more clear in the subsequent sections, but it should be noted at this time that the scale is inverse of frequency. That is, high scales correspond to low frequencies, and low scales correspond to high frequencies. Consequently, the little peak in the plot corresponds to the high frequency components in the signal, and the large peak corresponds to low frequency components (which appear before the high frequency components in time) in the signal.

You might be puzzled from the frequency resolution shown in the plot, since it shows good frequency resolution at high frequencies. Note however that, it is the good **scale resolution** that looks good at high frequencies (low scales), and good scale resolution means poor frequency resolution and vice versa. More about this in Part II and III.

THE WAVELET TUTORIAL

PART 2

BY

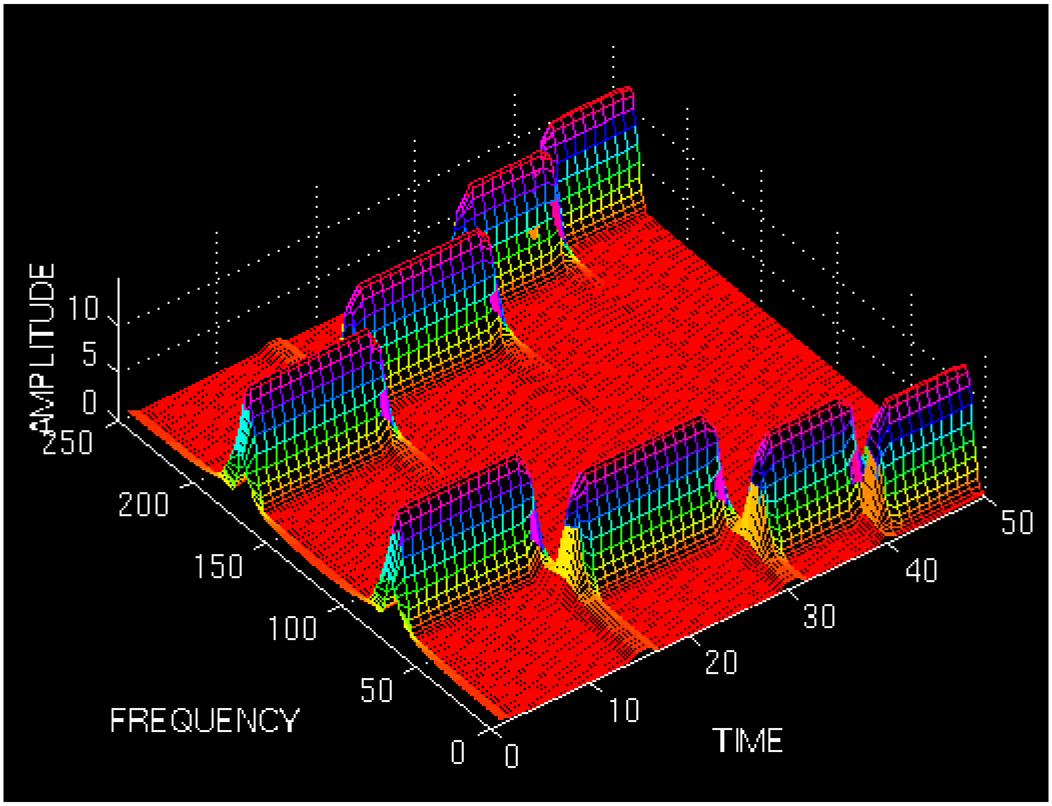
ROBI POLIKAR

FUNDAMENTALS:

THE FOURIER TRANSFORM

AND

THE SHORT TERM FOURIER TRANSFORM



FUNDAMENTALS

Let's have a short review of the first part.

We basically need Wavelet Transform (WT) to analyze non-stationary signals, i.e., whose frequency response varies in time. I have written that Fourier Transform (FT) is not suitable for non-stationary signals, and I have shown examples of it to make it more clear. For a quick recall, let me give the following example.

Suppose we have two different signals. Also suppose that they both have the same spectral components, with one major difference. Say one of the signals has four frequency components at all times, and the other has the same four frequency components at different times. The FT of both of the signals would be the same, as shown in the example in part 1 of this tutorial. Although the two signals are completely different, their (magnitude of) FT are the SAME**!** This obviously tells us that we cannot use the FT for non-stationary signals.

But why does this happen? In other words, how come both of the signals have the same FT? **HOW DOES FOURIER TRANSFORM WORK ANYWAY?**

An Important Milestone in Signal Processing:

THE FOURIER TRANSFORM

I will not go into the details of FT for two reasons:

1. It is too wide of a subject to discuss in this tutorial.
2. It is not our main concern anyway.

However, I would like to mention a couple important points again for two reasons:

1. It is a necessary background to understand how WT works.
2. It has been by far the most important signal processing tool for many (and I mean many many) years.

In 19th century (1822*, to be exact, but you do not need to know the exact time. Just trust me that it is far before than you can remember), the French mathematician J. Fourier, showed that any periodic function can be expressed as an infinite sum of periodic complex exponential functions. Many years after he had discovered this remarkable property of (periodic) functions, his ideas were generalized to first non-periodic functions, and then periodic or non-periodic discrete time signals. It is after this generalization that it became a very suitable tool for computer calculations. In 1965, a new algorithm called Fast Fourier Transform (FFT) was developed and FT became even more popular.

(* I thank Dr. Pedregal for the valuable information he has provided)

Now let us take a look at how Fourier Transform works:

FT decomposes a signal to complex exponential functions of different frequencies. The way it does this, is defined by the following two equations:

$$X(f) = \int_{-\infty}^{\infty} x(t) \bullet e^{-2j\pi ft} dt \ldots\ldots(1)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \bullet e^{2j\pi ft} df \ldots\ldots(2)$$

Robi Polikar, Ames IA. 1994

Figure 2.1

In the above equation, **t** stands for time, **f** stands for frequency, and **x** denotes the signal at hand. Note that **x** denotes the signal in time domain and the **X** denotes the signal in frequency domain. This convention is used to distinguish the two representations of the signal. Equation (1) is called the **Fourier transform of x(t)**, and equation (2) is called the **inverse Fourier transform of X(f)**, which is x(t).

For those of you who have been using the Fourier transform are already familiar with this. Unfortunately many people use these equations without knowing the underlying principle.

Please take a closer look at equation (1):

The signal x(t), is multiplied with an exponential term, **at some certain frequency "f"** , and then integrated over **ALL TIMES !!!** (The key words here are "all times", as will explained below).

Note that the exponential term in Eqn. (1) can also be written as:

Cos(2.pi.f.t)+j.Sin(2.pi.f.t).......(3)

The above expression has a real part of cosine of frequency **f**, and an imaginary part of sine of frequency **f**. So what we are actually doing is, multiplying the original signal with a complex expression which has sines and cosines of frequency **f**. Then we integrate this product. In other words, we add all the points in this product. If the result of this integration (which is nothing but some sort of infinite summation) is a large value, then we say that: **the signal x(t), has a dominant spectral component at frequency "f"**. This means that, a major portion of this signal is composed of frequency **f**. If the integration result is a small value, then this means that the signal does not have a major frequency component of **f** in it. If this integration result is zero, then the signal does not contain the frequency "f" at all.

It is of particular interest here to see how this integration works: The signal is multiplied with the sinusoidal term of frequency "f". If the signal has a high amplitude component of frequency "f", then that component and the sinusoidal term will coincide, and the product of them will give **a (relatively) large value**. This shows that, the signal "x", has a major frequency component of "f".

However, if the signal does not have a frequency component of "f", the product will yield zero, which shows that, the signal does not have a frequency component of "f". If the frequency "f", is not a major component of the signal "x(t)", then the product will give **a (relatively) small value**. This shows that, the frequency component "f" in the signal "x", has a small amplitude, in other words, it is not a major component of "x".

Now, note that the integration in the transformation equation (Eqn. 1) is over time. The left hand side of (1), however, is a function of frequency. Therefore, the integral in (1), is calculated for every value of **f**.

IMPORTANT(!) The information provided by the integral, corresponds to all time instances, since the integration is from minus infinity to plus infinity over time. It follows that no matter where in time the component with frequency "f" appears, it will affect the result of the integration equally as well. In other words, whether the frequency component "f" appears at time t1 or t2, it will have the same effect on the integration. This is why **Fourier transform is not suitable if the signal has time varying frequency**, i.e., the signal is **non-stationary.** If only, the signal has the frequency component "f" at all times (for all "f" values), then the result obtained by the Fourier transform makes sense.

Note that **the Fourier transform tells whether a certain frequency component exists or not.** This information is independent of where in time this component appears. It is therefore very important to know whether a signal is stationary or not, prior to processing it with the FT.

The example given in part one should now be clear. I would like to give it here again:

Look at the following figure, which shows the signal:

x(t)=cos(2*pi*5*t)+cos(2*pi*10*t)+cos(2*pi*20*t)+cos(2*pi*50*t)

that is , it has four frequency components of 5, 10, 20, and 50 Hz., all occurring at all times.

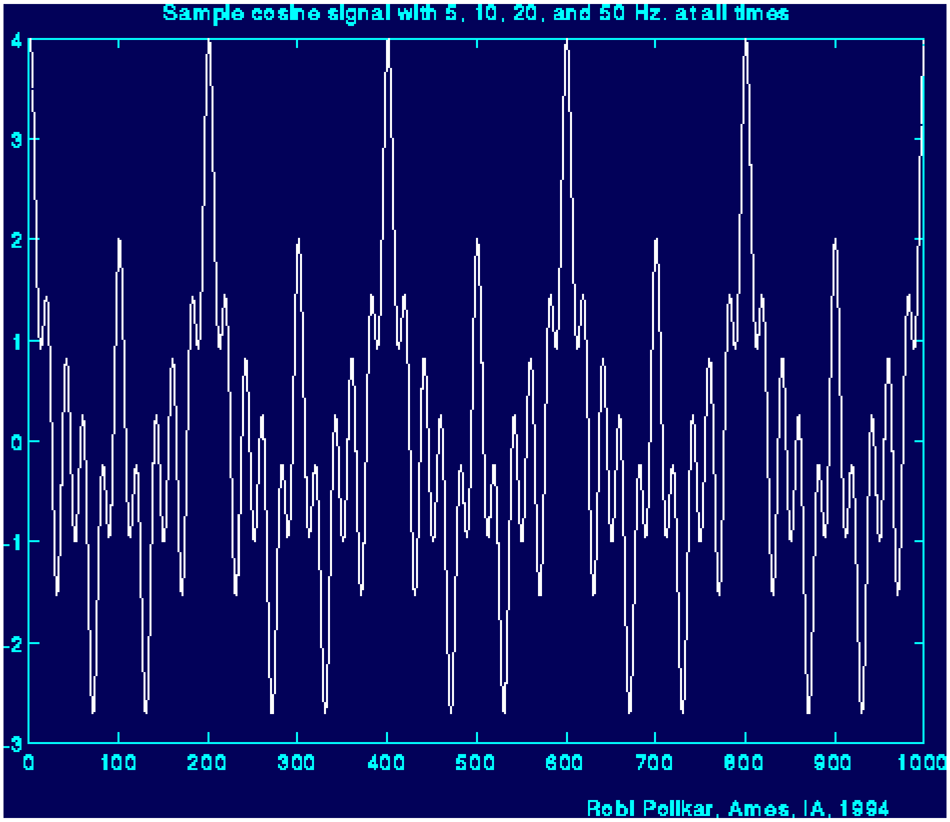


Figure 2.2

And here is the FT of it. The frequency axis has been cut here, but theoretically it extends to infinity (for continuous Fourier Transform (CFT). Actually, here we calculate the discrete Fourier transform (DFT), in which case the frequency axis goes up to (at least) twice the sampling frequency of the signal, and the transformed signal is symmetrical. However, this is not that important at this time.)

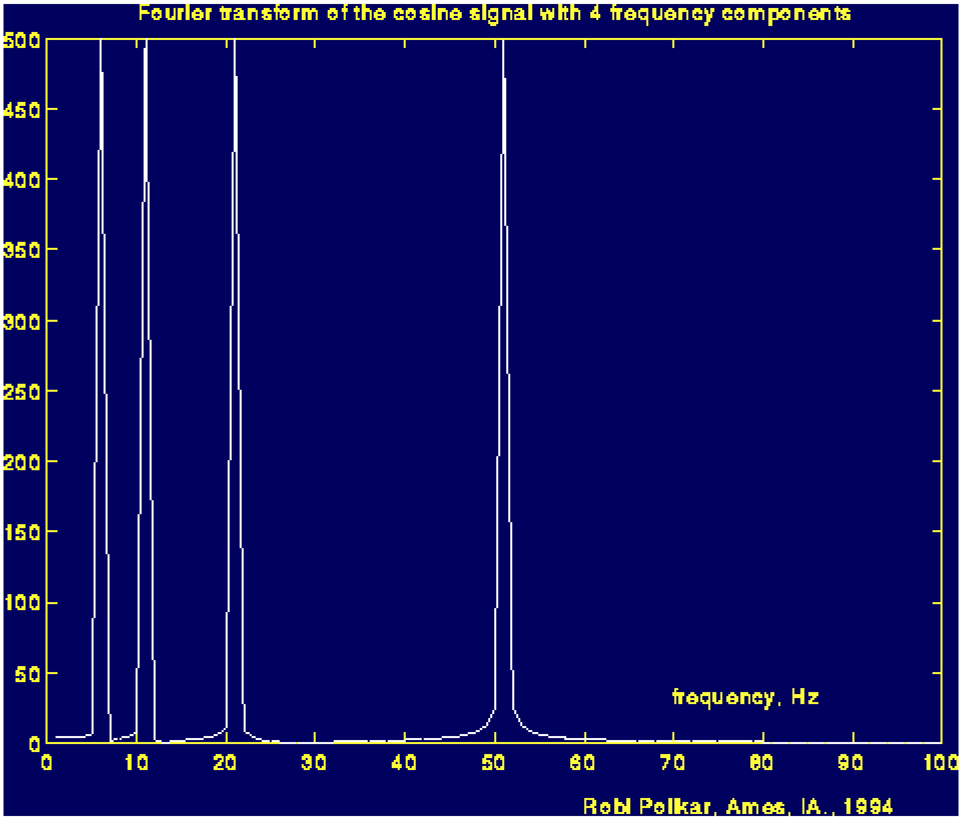


Figure 2.3

Note the four peaks in the above figure, which correspond to four different frequencies.

Now, look at the following figure: Here the signal is again the cosine signal, and it has the same four frequencies. However, these components occur at **different times**.

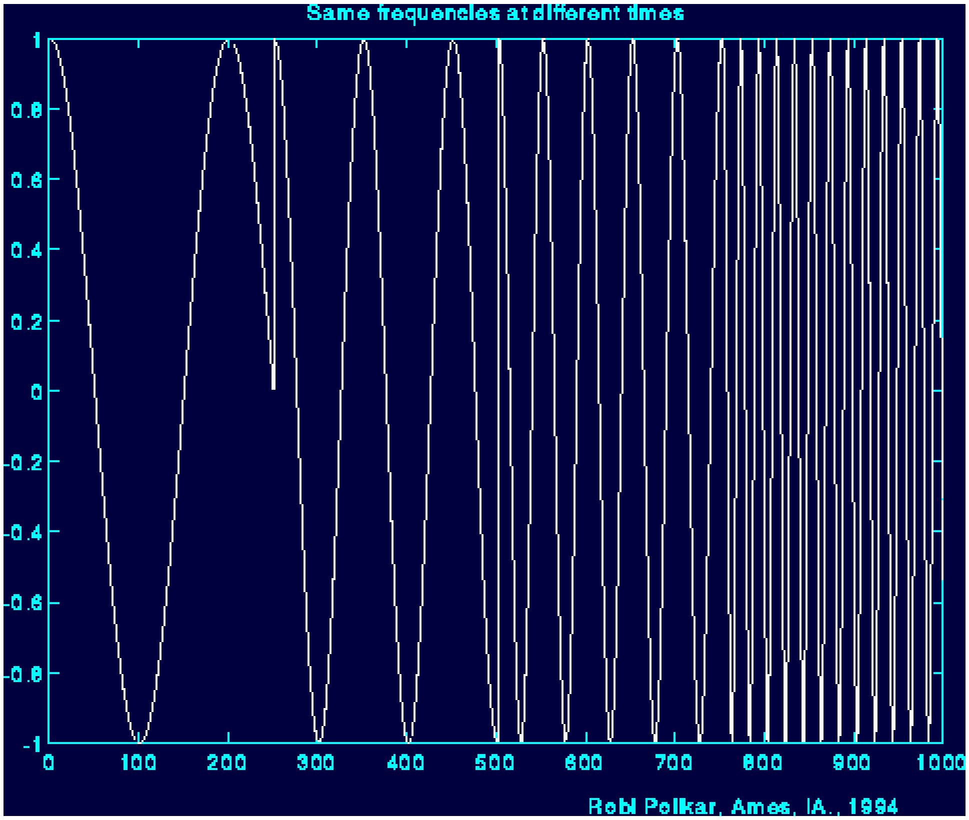


Figure 2.4

And here is the Fourier transform of this signal:

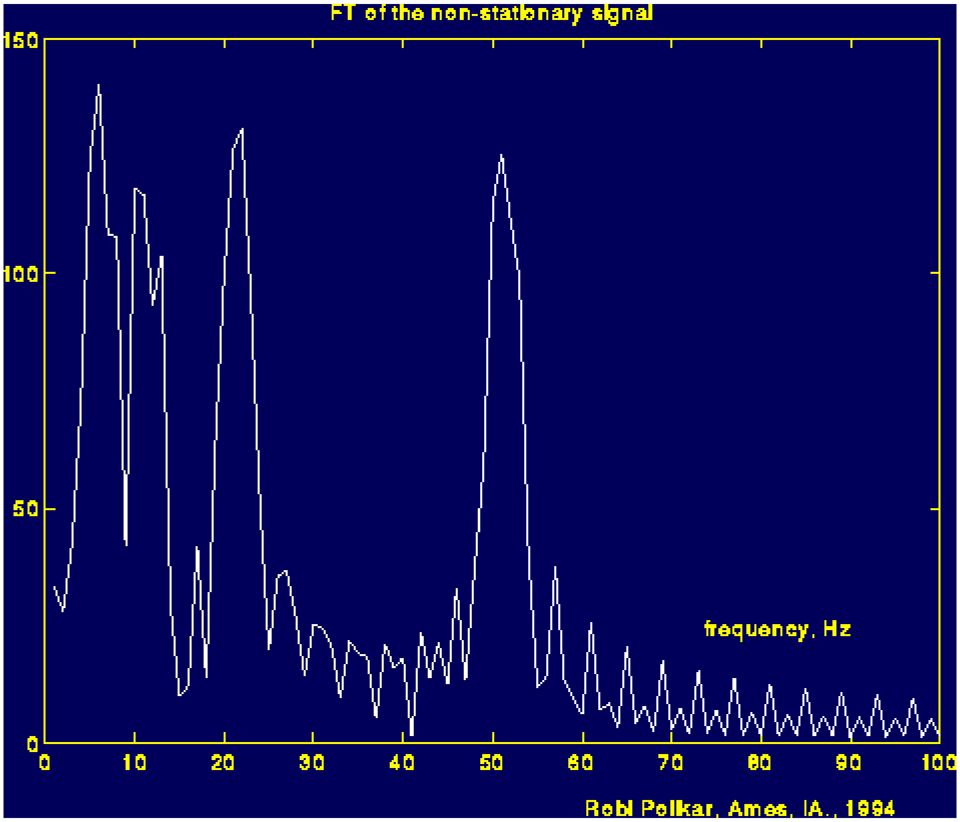


Figure 2.5

What you are supposed to see in the above figure, is it is (almost) same with the previous FT figure. Please look carefully and note the major four peaks corresponding to 5, 10, 20, and 50 Hz. I could have made this figure look very similar to the previous one, but I did not do that on purpose. The reason of the noise like thing in between peaks show that, those frequencies also exist in the signal. But the reason they have a small amplitude , is because, **they are not major spectral components of the given signal**, and the reason we see those, is because of the sudden change between the frequencies. Especially note how time domain signal changes at around time 250 (ms) (With some suitable filtering techniques, the noise like part of the frequency domain signal can be cleaned, but this has not nothing to do with our subject now. If you need further information please send me an e-mail).

By this time you should have understood the basic concepts of Fourier transform, when we can use it and we can not. As you can see from the above example, FT cannot distinguish the two signals very well. To FT, both signals are the same, because they constitute of the same frequency components. Therefore, FT is not a suitable tool for analyzing non-stationary signals, i.e., signals with time varying spectra.

Please keep this very important property in mind. Unfortunately, many people using the FT do not think of this. They assume that the signal they have is stationary where it is not in many practical cases. Of course if you are not interested in **at what times these frequency components occur**, but only interested in what frequency components exist, then FT can be a suitable tool to use.

So, now that we know that we can not use (well, we can, but we shouldn't) FT for non-stationary signals, what are we going to do?

Remember that, I have mentioned that wavelet transform is only (about) a decade old. You may wonder if researchers noticed this non-stationarity business only ten years ago or not.

Obviously not.

Apparently they must have done something about it before they figured out the wavelet transform....?

Well..., they sure did...

They have come up with ...

LINEAR TIME FREQUENCY REPRESENTATIONS

THE SHORT TERM FOURIER TRANSFORM

So, how are we going to insert this time business into our frequency plots? Let's look at the problem in hand little closer.

What was wrong with FT? It did not work for non-stationary signals. Let's think this: **Can we assume that, some portion of a non-stationary signal is stationary?**

The answer is yes.

Just look at the third figure above. The signal is stationary every 250 time unit intervals.

You may ask the following question?

What if the part that we can consider to be stationary is very small?

Well, if it is too small, it is too small. There is nothing we can do about that, and actually, there is nothing wrong with that either. We have to play this game with the physicists' rules.

If this region where the signal can be assumed to be stationary is too small, then we look at that signal from narrow windows, narrow enough that the portion of the signal seen from these windows are indeed stationary.

This approach of researchers ended up with a **revised** version of the Fourier transform, so-called: The Short Time Fourier Transform (STFT).

There is only a minor difference between STFT and FT. In STFT, the signal is divided into small enough segments, where these segments (portions) of the signal can be assumed to be stationary. For this purpose, a window function **"w"** is chosen. The width of this window must be equal to the segment of the signal where its stationarity is valid.

This window function is first located to the very beginning of the signal. That is, the window function is located at t=0. Let's suppose that the width of the window is **"T"** s. At this time instant (t=0), the window function will overlap with the first **T/2** seconds (I will assume that all time units are in seconds). The window function and the signal are then multiplied. By doing this, only the first T/2 seconds of the signal is being chosen, with the appropriate weighting of the window (if the window is a rectangle, with amplitude "1", then the product will be equal to the signal). Then this product is assumed to be just another signal, whose FT is to be taken. In other words, FT of this product is taken, just as taking the FT of any signal.

The result of this transformation is the FT of the first **T/2** seconds of the signal. If this portion of the signal is stationary, as it is assumed, then there will be no problem and the obtained result will be a true frequency representation of the first T/2 seconds of the signal.

The next step, would be shifting this window (for some t1 seconds) to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed, until the end of the signal is reached by shifting the window with "t1" seconds intervals.

The following definition of the STFT summarizes all the above explanations in one line:

$$STFT_X^{(\omega)}(t,f) = \int_t [x(t) \bullet \omega^*(t - t')] \bullet e^{-j2\pi ft} dt \ldots\ldots(3)$$

Figure 2.6

Please look at the above equation carefully. **x(t)** is the signal itself, **w(t)** is the window function, and ***** is the complex conjugate. As you can see from the equation, the STFT of the signal is nothing but the FT of the signal multiplied by a window function.

For every **t'** and **f** a new STFT coefficient is computed (Correction: The "t" in the parenthesis of STFT should be "t'". I will correct this soon. I have just noticed that I have mistyped it).

The following figure may help you to understand this a little better:

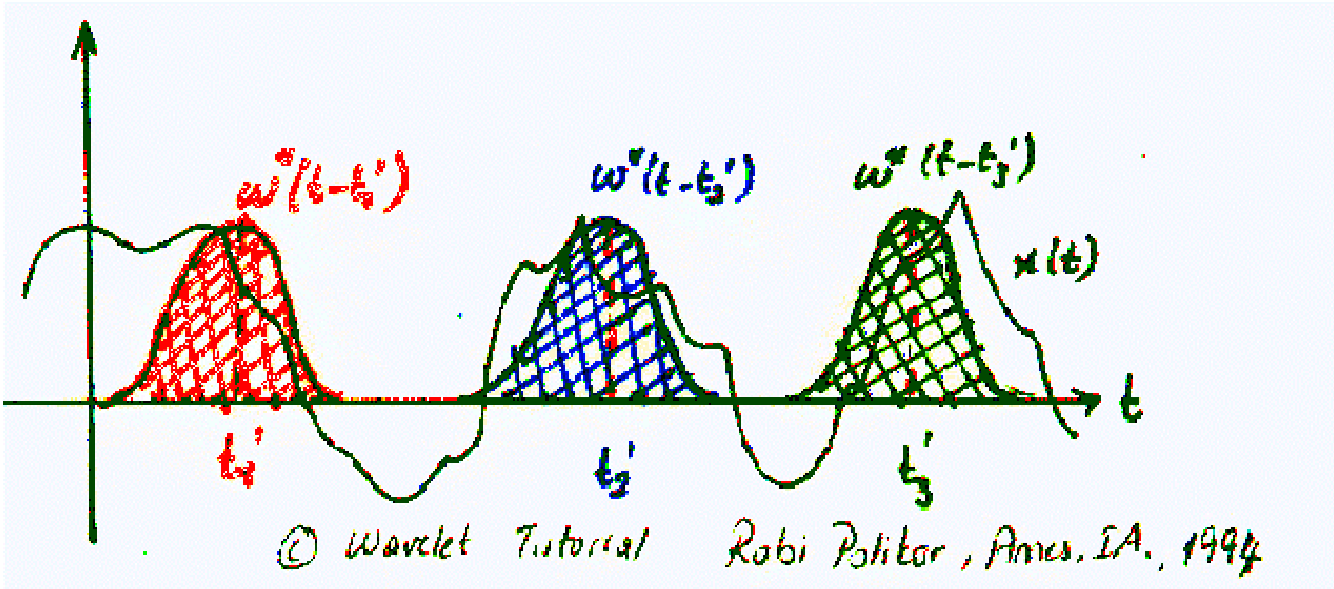


Figure 2.7

The Gaussian-like functions in color are the windowing functions. The red one shows the window located at t=t1', the blue shows t=t2', and the green one shows the window located at t=t3'. These will correspond to three different FTs at three different times. Therefore, we will obtain a true **time-frequency representation (TFR)** of the signal.

Probably the best way of understanding this would be looking at an example. First of all, since our transform is a function of both time and frequency (unlike FT, which is a function of frequency only), the transform would be two dimensional (three, if you count the amplitude too). Let's take a non-stationary signal, such as the following one:

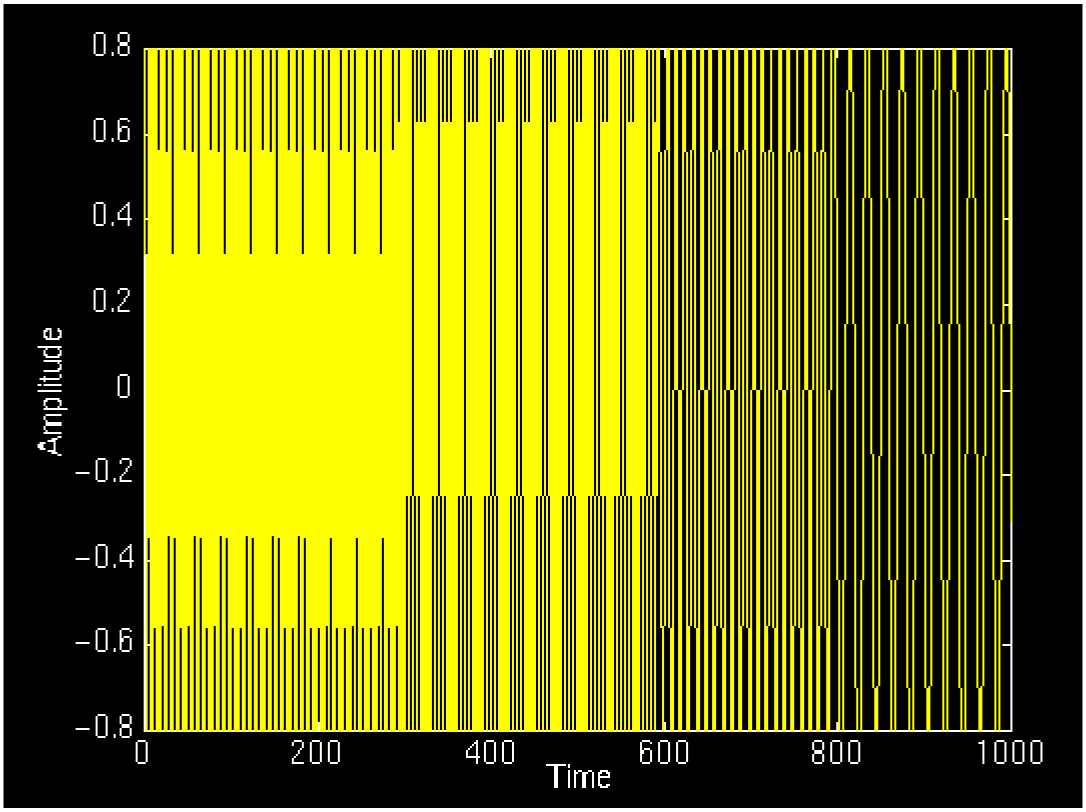


Figure 2.8

In this signal, there are four frequency components at different times. The interval 0 to 250 ms is a simple sinusoid of 300 Hz, and the other 250 ms intervals are sinusoids of 200 Hz, 100 Hz, and 50 Hz, respectively. Apparently, this is a non-stationary signal. Now, let's look at its STFT:

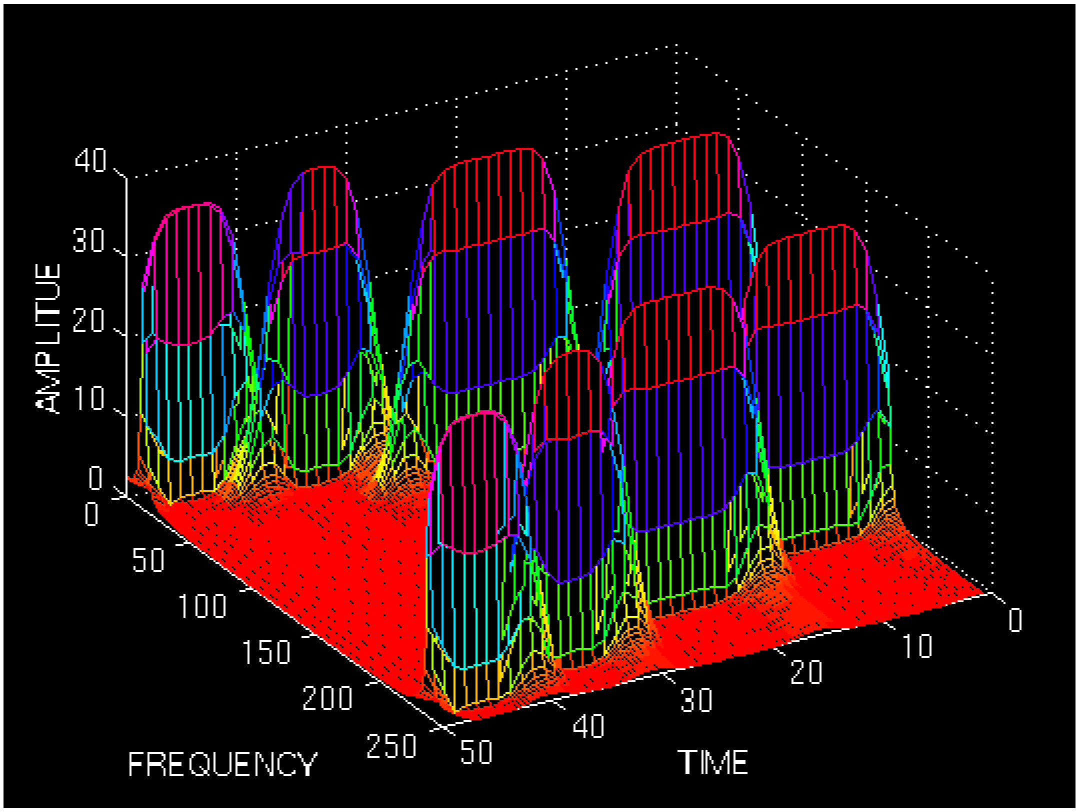


Figure 2.9

As expected, this is a two dimensional plot (3 dimensional, if you count the amplitude too). The "x" and "y" axes are time and frequency, respectively. Please, ignore the numbers on the axes, since they are normalized in some respect, which is not of any interest to us at this time. Just examine the shape of the time-frequency representation.

First of all, note that the graph is symmetric with respect to midline of the frequency axis. Remember that, although it was not shown, FT of a real signal is always symmetric, since STFT is nothing but a windowed version of the FT, it should come as no surprise that STFT is also symmetric in frequency. The symmetric part is said to be associated with negative frequencies, an odd concept which is difficult to comprehend, fortunately, it is not important; it suffices to know that STFT and FT are symmetric.

What is important, are the four peaks; note that there are four peaks corresponding to four different frequency components. Also note that, unlike FT, **these four peaks are located at different time intervals along the time axis**. Remember that the original signal had four spectral components located at different times.

Now we have a true time-frequency representation of the signal. We not only know what frequency components are present in the signal, but we also know where they are located in time.

It is grrrreeeaaatttttt!!!! Right?

Well, not really!

You may wonder, since STFT gives the TFR of the signal, why do we need the wavelet transform. The implicit problem of the STFT is not obvious in the above example. Of course, an example that would work nicely was chosen on purpose to demonstrate the concept.

The problem with STFT is the fact whose roots go back to what is known as the **Heisenberg Uncertainty Principle**. This principle originally applied to the momentum and location of moving particles, can be applied to time-frequency information of a signal. Simply, this principle states that one cannot know the exact time-frequency representation of a signal, i.e., one cannot know what spectral components exist at what instances of times. What one **can** know are the **time intervals** in which certain **band of frequencies** exist, which is a **resolution** problem.

The problem with the STFT has something to do with the **width** of the window function that is used. To be technically correct, this width of the window function is known as **the support** of the window. If the window function is narrow, than it is known as **compactly supported**. This terminology is more often used in the wavelet world, as we will see later.

Here is what happens:

Recall that in the FT there is no resolution problem in the frequency domain, i.e., we know exactly what frequencies exist; similarly we there is no time resolution problem in the time domain, since we know the value of the signal at every instant of time. Conversely, the time resolution in the FT, and the frequency resolution in the time domain are zero, since we have no information about them. What gives the perfect frequency resolution in the FT is the fact that the window used in the FT is its kernel, the **exp{jwt}** function, which lasts at all times from minus infinity to plus infinity. Now, in STFT, our window is of finite length, thus it covers only a portion of the signal, which causes the frequency resolution to get poorer. What I mean by getting poorer is that, we no longer know the exact frequency components that exist in the signal, but we only know a band of frequencies that exist:

In FT, the kernel function, allows us to obtain perfect frequency resolution, because the kernel itself is a window of infinite length. In STFT is window is of finite length, and we no longer have perfect frequency resolution. You may ask, why don't we make the length of the window in the STFT infinite, just like as it is in the FT, to get perfect frequency resolution? Well, than you loose all the time information, you basically end up with the FT instead of STFT. To make a long story real short, we are faced with the following dilemma:

If we use a window of infinite length, we get the FT, which gives perfect frequency resolution, but no time information. Furthermore, in order to obtain the stationarity, we have to have a short enough window, in which the signal is stationary. The narrower we make the window, the better the time resolution, and better the assumption of stationarity, but poorer the frequency resolution:

Narrow window ===>good time resolution, poor frequency resolution.

Wide window ===>good frequency resolution, poor time resolution.

In order to see these effects, let's look at a couple examples: I will show four windows of different length, and we will use these to compute the STFT, and see what happens:

The window function we use is simply a Gaussian function in the form:

w(t)=exp(-a*(t^2)/2);

where **a** determines the length of the window, and **t** is the time. The following figure shows four window functions of varying regions of support, determined by the value of **a**. Please disregard the numeric values of **a** since the time interval where this function is computed also determines the function. Just note the length of each window. The above example given was computed with the second value, **a=0.001** . I will now show the STFT of the same signal given above computed with the other windows.

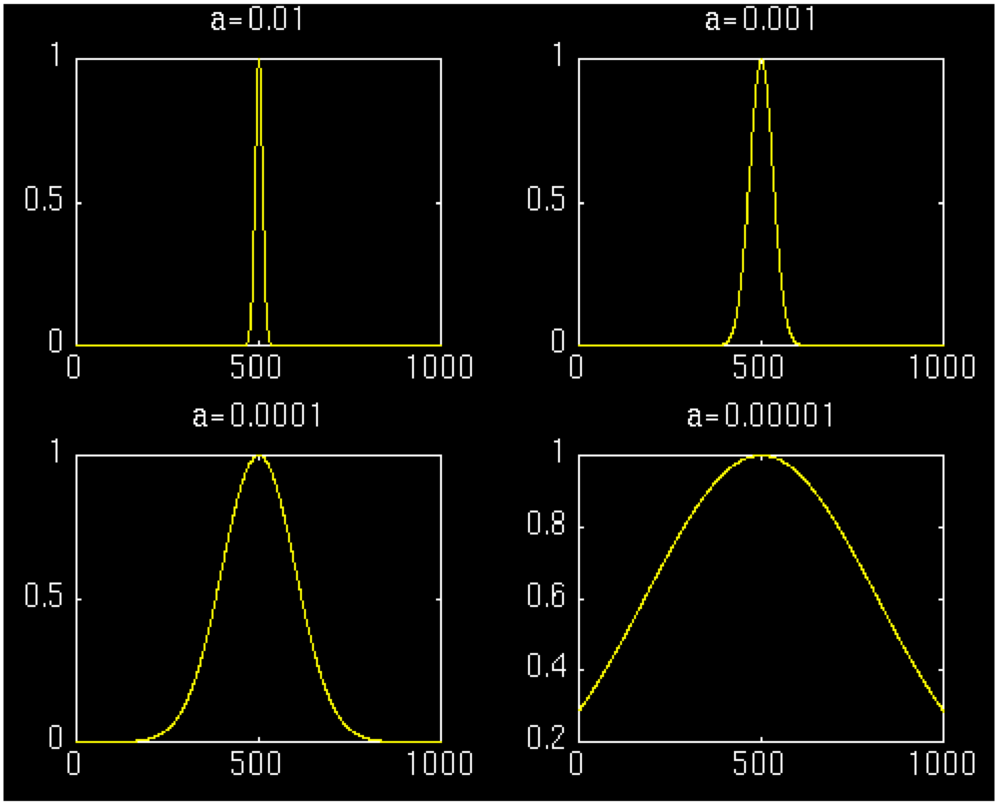


Figure 2.10

First let's look at the first most narrow window. We expect the STFT to have a very good time resolution, but relatively poor frequency resolution:

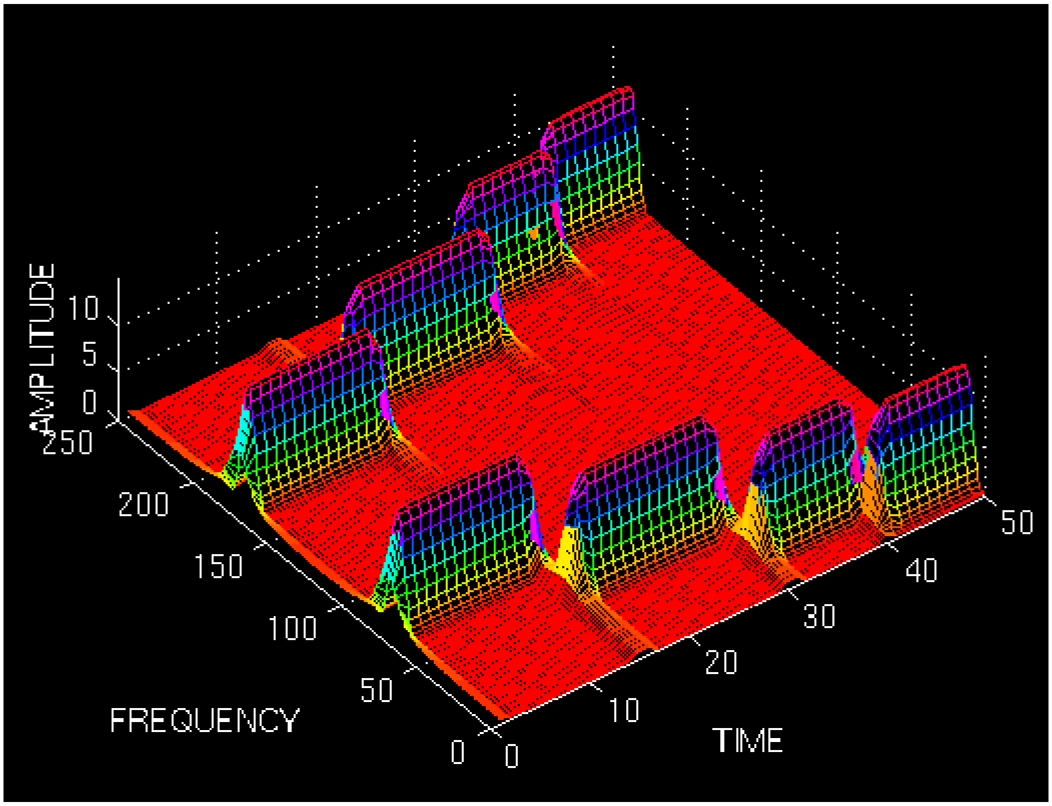


Figure 2.11

The above figure shows this STFT. The figure is shown from a top bird-eye view with an angle for better interpretation. Note that the four peaks are well separated from each other in time. Also note that, in frequency domain, every peak covers a range of frequencies, instead of a single frequency value. Now let's make the window wider, and look at the third window (the second one was already shown in the first example).

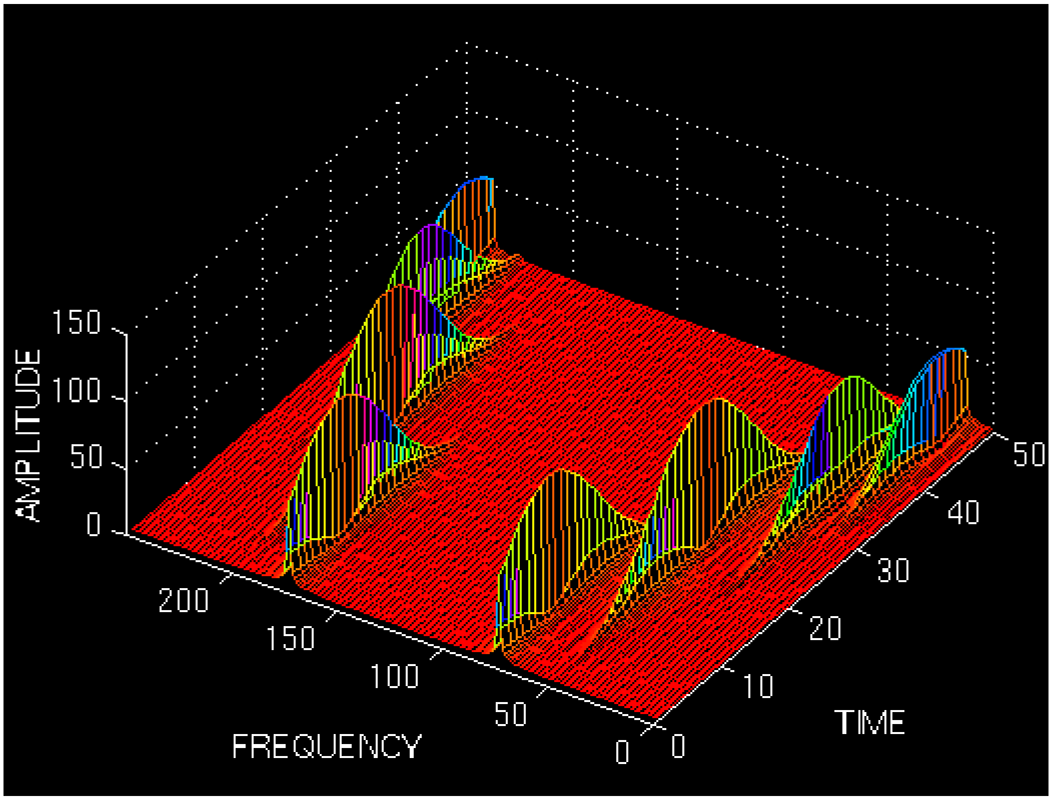


Figure 2.12

Note that the peaks are not well separated from each other in time, unlike the previous case, however, in frequency domain the resolution is much better. Now let's further increase the width of the window, and see what happens:

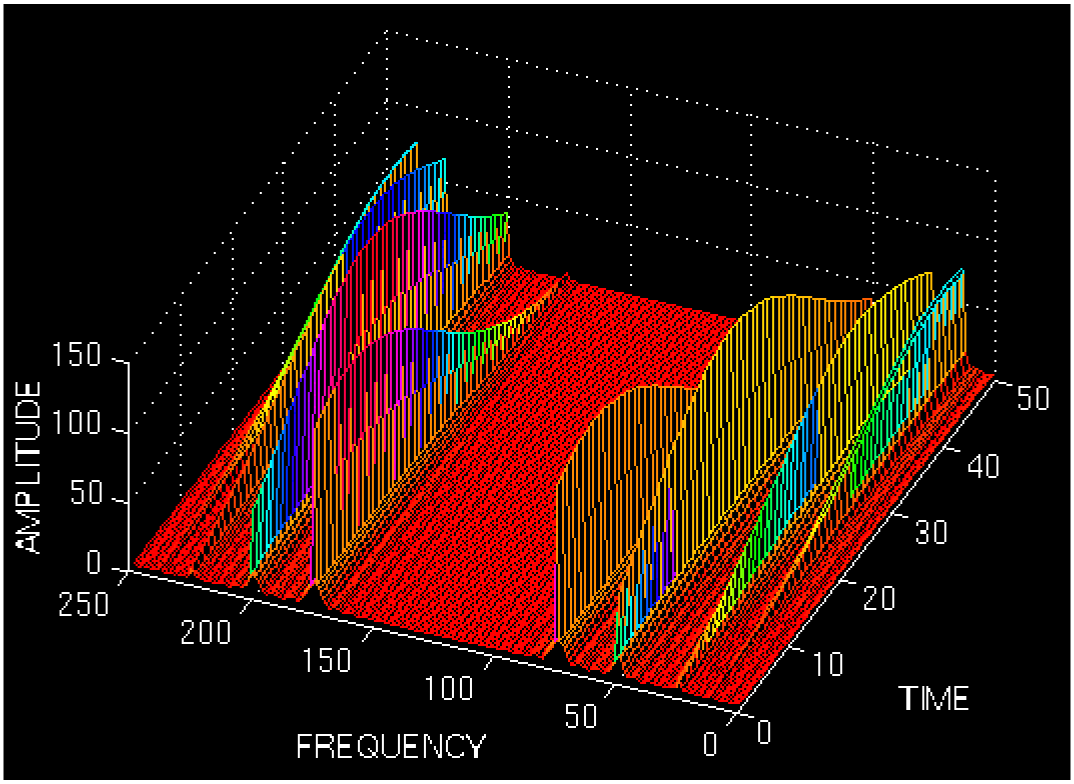


Figure 2.13

Well, this should be of no surprise to anyone now, since we would expect a terrible (and I mean absolutely terrible) time resolution.

These examples should have illustrated the implicit problem of resolution of the STFT. Anyone who would like to use STFT is faced with this problem of resolution. What kind of a window to use? Narrow windows give good time resolution, but poor frequency resolution. Wide windows give good frequency resolution, but poor time resolution; furthermore, wide windows may violate the condition of stationarity. The problem, of course, is a result of choosing a window function, once and for all, and use that window in the entire analysis. The answer, of course, is application dependent: If the frequency components are well separated from each other in the original signal, then we may sacrifice some frequency resolution and go for good time resolution, since the spectral components are already well separated from each other. However, if this is not the case, then a good window function, could be more difficult than finding a good stock to invest in.

By now, you should have realized how wavelet transform comes into play. The Wavelet transform (WT) solves the dilemma of resolution to a certain extent, as we will see in the next part.

This completes Part II of this tutorial. The continuous wavelet transform is the subject of the Part III of this tutorial. If you did not have much trouble in coming this far, and what have been written above make sense to you, you are now ready to take the ultimate challenge in understanding the basic concepts of the wavelet theory.

THE WAVELET TUTORIAL

PART III

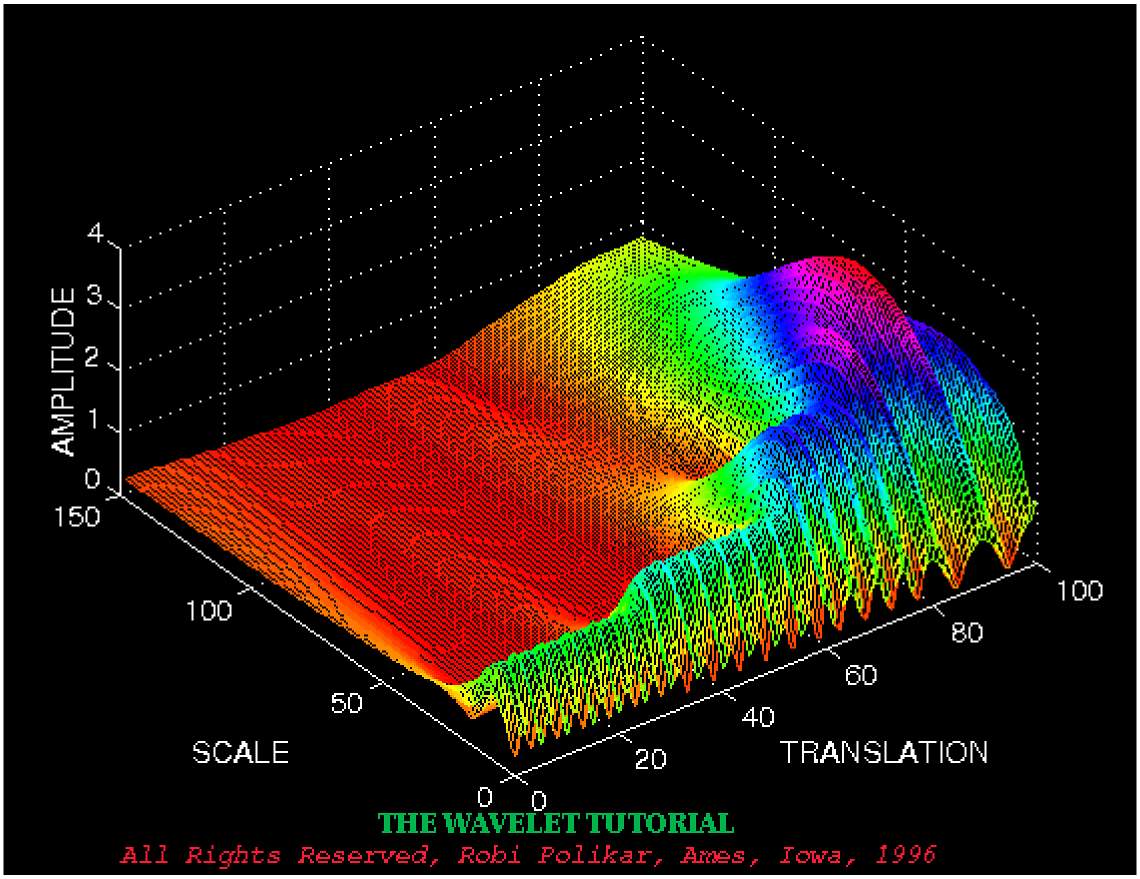
MULTIRESOLUTION ANALYSIS

&

THE CONTINUOUS WAVELET TRANSFORM

BY

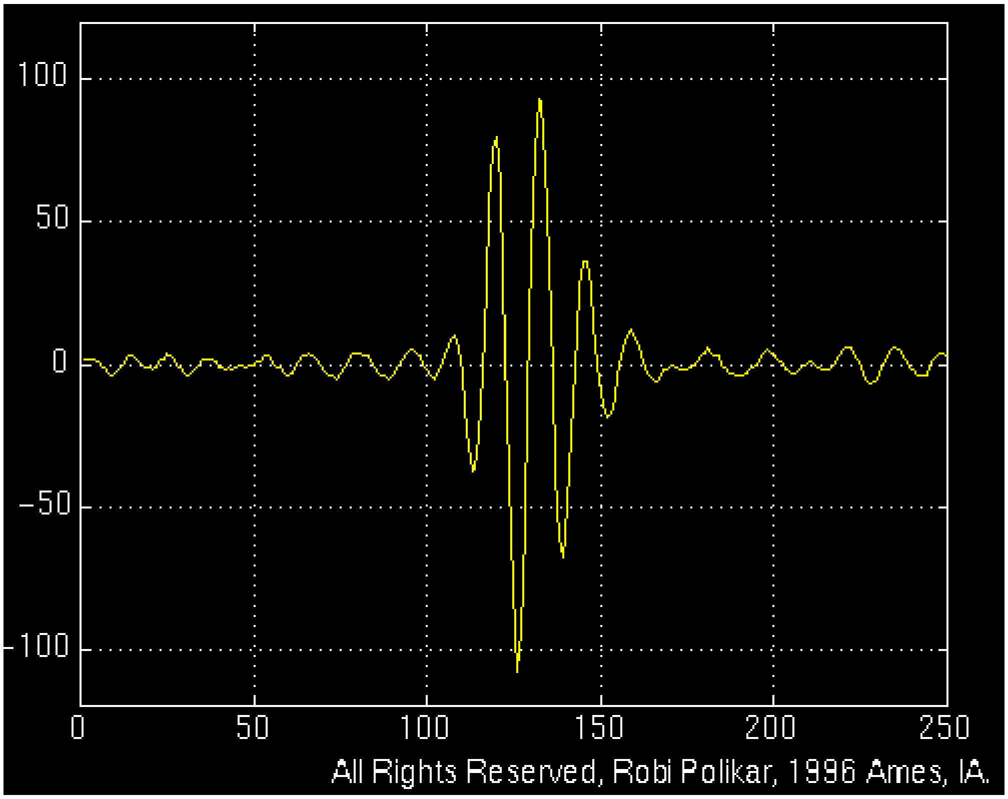
ROBI POLIKAR



MULTIRESOLUTION ANALYSIS

Although the time and frequency resolution problems are results of a physical phenomenon (the Heisenberg uncertainty principle) and exist regardless of the transform used, it is possible to analyze any signal by using an alternative approach called the **multiresolution analysis (MRA)** . MRA, as implied by its name, analyzes the signal at different frequencies with different resolutions. Every spectral component is not resolved equally as was the case in the STFT.

MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. This approach makes sense especially when the signal at hand has high frequency components for short durations and low frequency components for long durations. Fortunately, the signals that are encountered in practical applications are often of this type. For example, the following shows a signal of this type. It has a relatively low frequency component throughout the entire signal and relatively high frequency components for a short duration somewhere around the middle.



THE CONTINUOUS WAVELET TRANSFORM

The continuous wavelet transform was developed as an alternative approach to the short time Fourier transform to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is multiplied with a function, {\it the wavelet}, similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the CWT:

1. The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid, i.e., negative frequencies are not computed.

2. The width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The continuous wavelet transform is defined as follows

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^* \left(\frac{t-\tau}{s}\right) dt$$

Equation 3.1

As seen in the above equation, the transformed signal is a function of two variables, **tau** and **s**, the **translation** and **scale** parameters, respectively. **psi(t)** is the transforming function, and it is called **the mother wavelet** . The term **mother wavelet** gets its name due to two important properties of the wavelet analysis as explained below:

The term **wavelet** means a **small wave** . The smallness refers to the condition that this (window) function is of finite length (**compactly supported**). The wave refers to the condition that this function is oscillatory. The term **mother** implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a **prototype** for generating the other window functions.

The term **translation** is used in the same sense as it was used in the STFT; it is related to the location of the window, as the window is shifted through the signal. This term, obviously, corresponds to time information in the transform domain. However, we do not have a frequency parameter, as we had before for the STFT. Instead, we have scale parameter which is defined as $1/frequency$. The term frequency is reserved for the STFT. Scale is described in more detail in the next section.

THE SCALE

The parameter **scale** in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a non-detailed global view (of the signal), and low scales correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scales) correspond to a global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal (that usually lasts a relatively short time). Cosine signals corresponding to various scales are given as examples in the following figure.

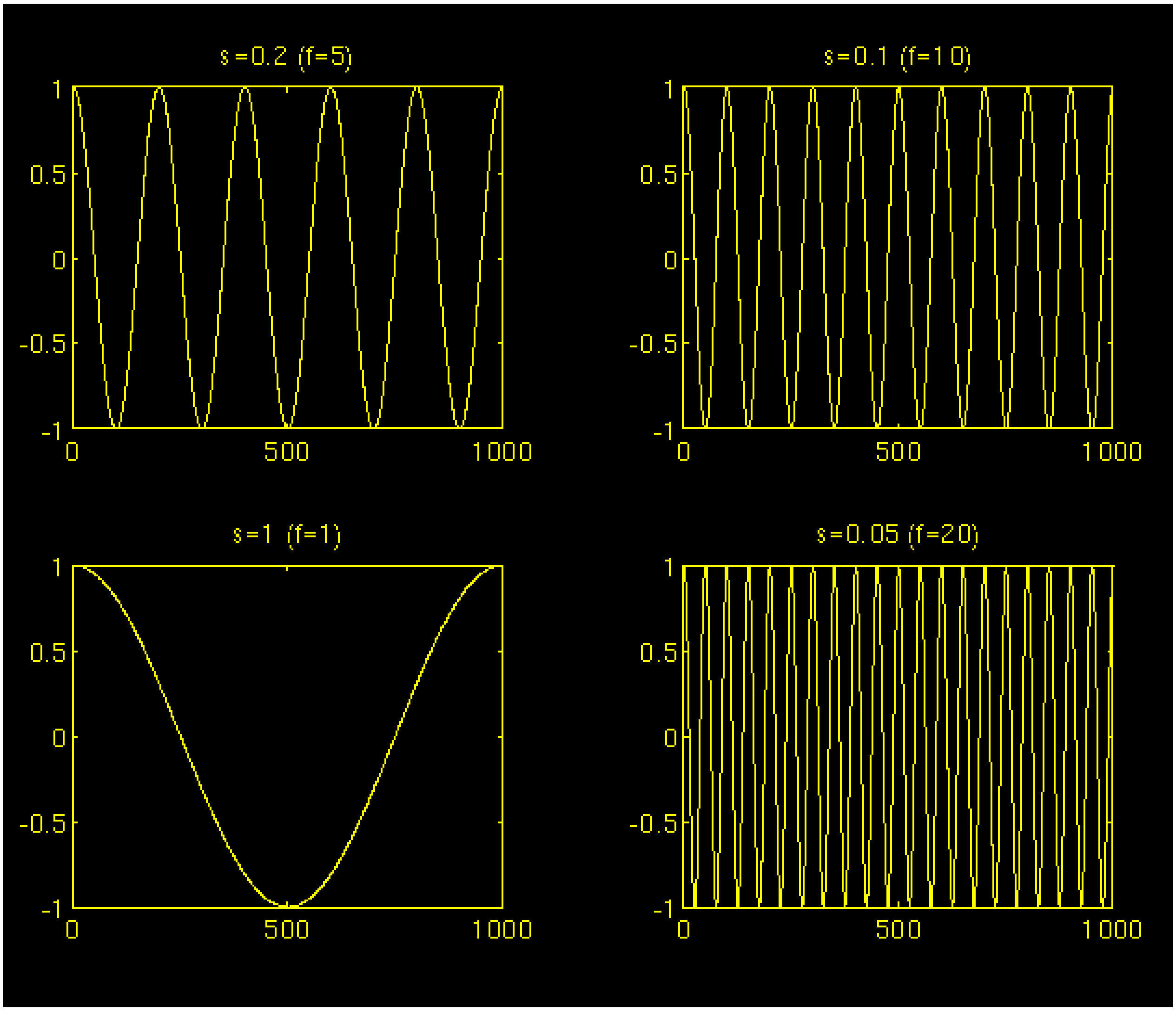


Figure 3.2

Fortunately in practical applications, low scales (high frequencies) do not last for the entire duration of the signal, unlike those shown in the figure, but they usually appear from time to time as short bursts, or spikes. High scales (low frequencies) usually last for the entire duration of the signal.

Scaling, as a mathematical operation, either dilates or compresses a signal. Larger scales correspond to dilated (or stretched out) signals and small scales correspond to compressed signals. All of the signals given in the figure are derived from the same cosine signal, i.e., they are dilated or compressed versions of the same function. In the above figure, **s=0.05** is the smallest scale, and **s=1** is the largest scale.

In terms of mathematical functions, if **f(t)** is a given function, **f(st)** corresponds to a contracted (compressed) version of **f(t)** if **s** > **1** and to an expanded (dilated) version of **f(t)** if **s** < **1** .

However, in the definition of the wavelet transform, the scaling term is used in the denominator, and therefore, the opposite of the above statements holds, i.e., scales **s > 1** dilates the signals whereas scales **s < 1**, compresses the signal. This interpretation of scale will be used throughout this text.

COMPUTATION OF THE CWT

Interpretation of the above equation will be explained in this section. Let **x(t)** is the signal to be analyzed. The mother wavelet is chosen to serve as a prototype for all windows in the process. All the windows that are used are the dilated (or compressed) and shifted versions of the mother wavelet. There are a number of functions that are used for this purpose. The Morlet wavelet and the Mexican hat function are two candidates, and they are used for the wavelet analysis of the examples which are presented later in this chapter.

Once the mother wavelet is chosen the computation starts with **s=1** and the continuous wavelet transform is computed for all values of **s**, smaller and larger than ``1". However, depending on the signal, a complete transform is usually not necessary. For all practical purposes, the signals are bandlimited, and therefore, computation of the transform for a limited interval of scales is usually adequate. In this study, some finite interval of values for **s** were used, as will be described later in this chapter.

For convenience, the procedure will be started from scale **s=1** and will continue for the increasing values of **s**, i.e., the analysis will start from high frequencies and proceed towards low frequencies. This first value of **s** will correspond to the most compressed wavelet. As the value of **s** is increased, the wavelet will dilate.

The wavelet is placed at the beginning of the signal at the point which corresponds to time=0. The wavelet function at scale ``1" is multiplied by the signal and then integrated **over all times**. The result of the integration is then multiplied by the constant number **1/sqrt{s}**. This multiplication is for energy normalization purposes so that the transformed signal will have the same energy at every scale. The final result is the value of the transformation, i.e., the value of the continuous wavelet transform **at time zero** and **scale s=1**. In other words, it is the value that corresponds to the point **tau =0 , s=1** in the time-scale plane.

The wavelet at scale **s=1** is then shifted towards the right by **tau** amount to the location **t=tau** , and the above equation is computed to get the transform value at **t=tau , s=1** in the time-frequency plane.

This procedure is repeated until the wavelet reaches the end of the signal. **One row of points on the time-scale plane** for the scale **s=1** is now completed.

Then, **s** is increased by a small value. Note that, this is a continuous transform, and therefore, both **tau** and **s** must be incremented continuously. However, if this transform needs to be computed by a computer, then both parameters are increased by **a sufficiently small step size**. This corresponds to sampling the time-scale plane.

The above procedure is repeated for every value of **s**. Every computation for a given value of **s** fills the corresponding single row of the time-scale plane. When the process is completed for all desired values of **s**, the CWT of the signal has been calculated.

The figures below illustrate the entire process step by step.

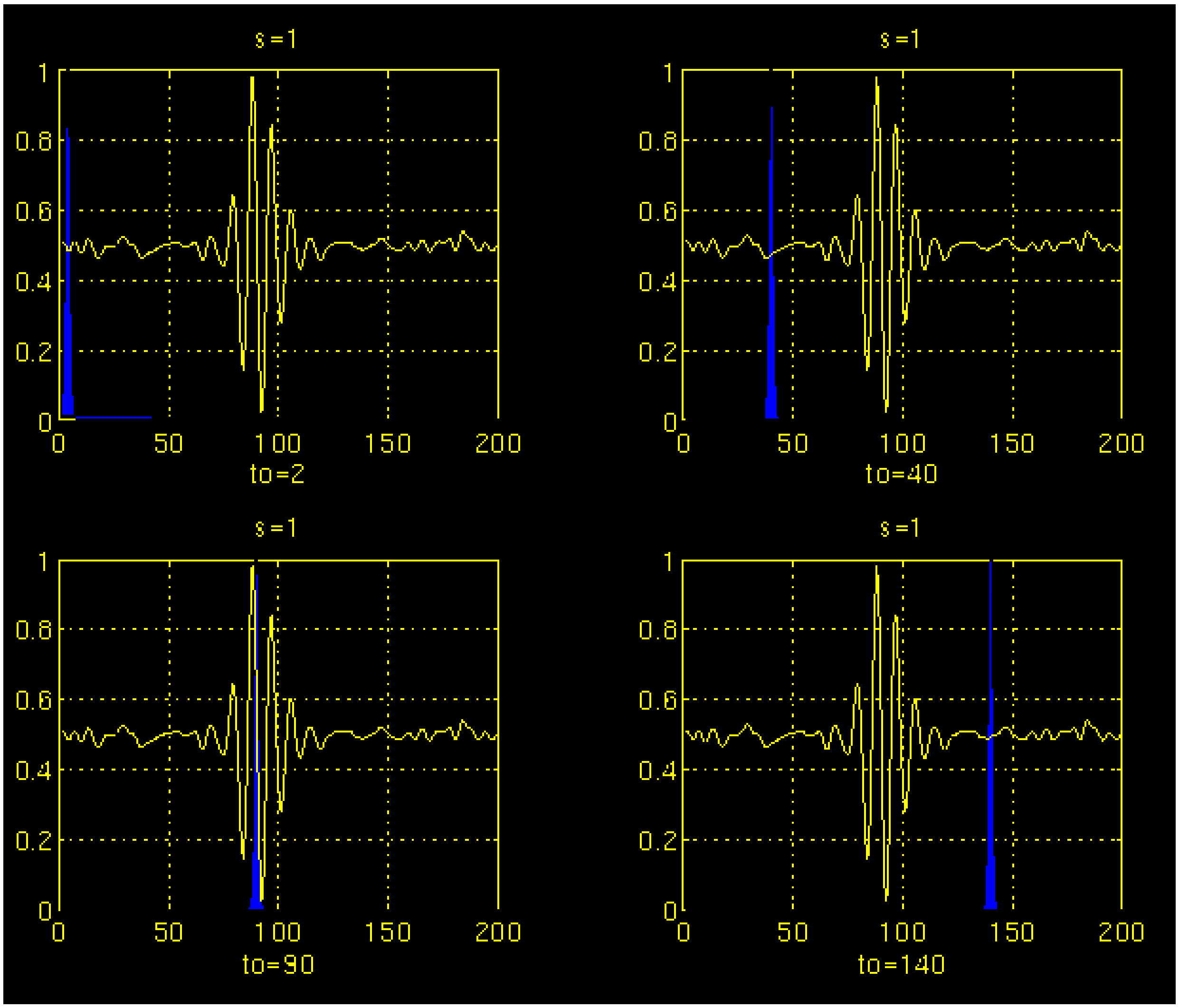


Figure 3.3

In Figure 3.3, the signal and the wavelet function are shown for four different values of **tau**. The signal is a truncated version of the signal shown in Figure 3.1. The scale value is **1**, corresponding to the lowest scale, or highest frequency. Note how compact it is (the blue window). It should be as narrow as the highest frequency component that exists in the signal. Four distinct locations of the wavelet function are shown in the figure at **to=2, to=40, to=90, and to=140**. At every location, it is multiplied by the signal. Obviously, the product is nonzero only where the signal falls in the region of support of the wavelet, and it is zero elsewhere. By shifting the wavelet in time, the signal is localized in time, and by changing the value of **s**, the signal is localized in scale (frequency).

If the signal has a spectral component that corresponds to the current value of **s** (which is 1 in this case), the product of the wavelet with the signal **at the location where this spectral component exists** gives a relatively large value. If the spectral component that corresponds to the current value of **s** is not present in the signal, the product value will be relatively small, or zero. The signal in Figure 3.3 has spectral components comparable to the window's width at **s=1** around t=100 ms.

The continuous wavelet transform of the signal in Figure 3.3 will yield large values for low scales around time 100 ms, and small values elsewhere. For high scales, on the other hand, the continuous wavelet transform will give large values for almost the entire duration of the signal, since low frequencies exist at all times.

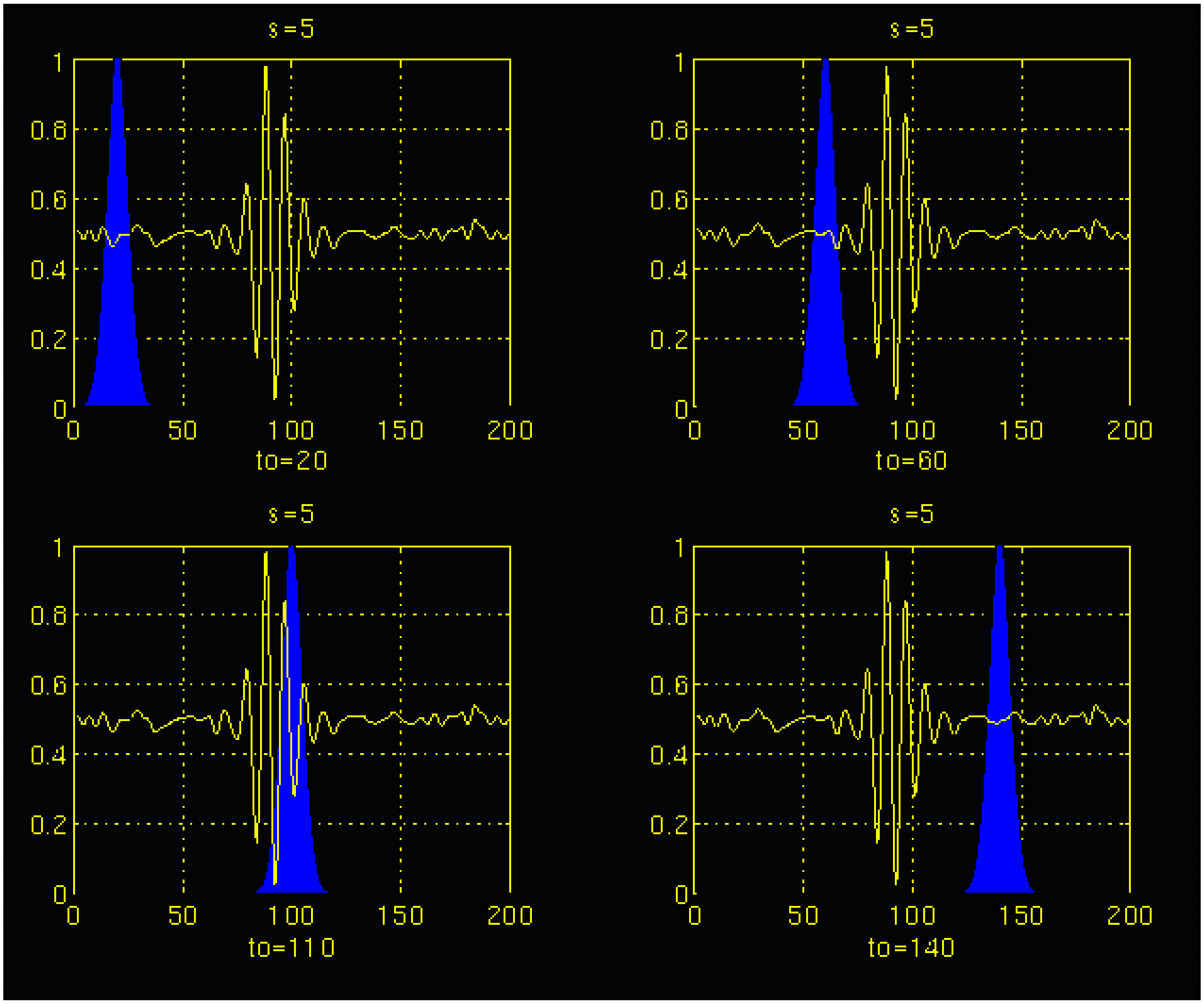


Figure 3.4

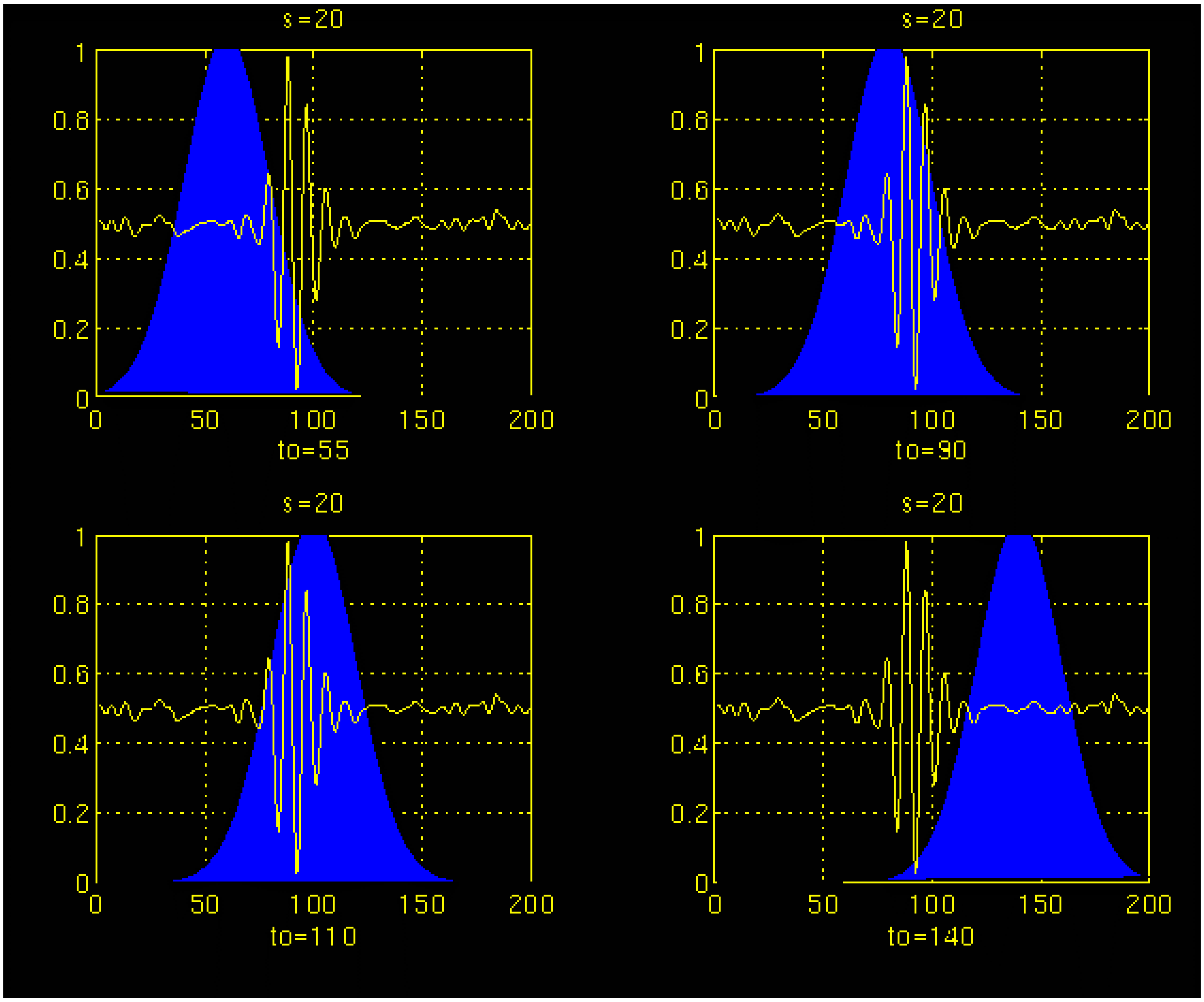


Figure 3.5

Figures 3.4 and 3.5 illustrate the same process for the scales s=5 and s=20, respectively. Note how the window width changes with increasing scale (decreasing frequency). As the window width increases, the transform starts picking up the lower frequency components.

As a result, for every scale and for every time (interval), one point of the time-scale plane is computed. The computations at one scale construct the rows of the time-scale plane, and the computations at different scales construct the columns of the time-scale plane.

Now, let's take a look at an example, and see how the wavelet transform really looks like. Consider the **non-stationary** signal in Figure 3.6. This is similar to the example given for the STFT, except at different frequencies. As stated on the figure, the signal is composed of four frequency components at 30 Hz, 20 Hz, 10 Hz and 5 Hz.

Figure 3.6

Figure 3.7 is the continuous wavelet transform (CWT) of this signal. Note that the axes are translation and scale, not time and frequency. However, translation is strictly related to time, since it indicates where the mother wavelet is located. The translation of the mother wavelet can be thought of as the time elapsed since **t=0**. The scale, however, has a whole different story. Remember that the scale parameter **s** in equation 3.1 is actually the inverse of frequency. In other words, whatever we said about the properties of the wavelet transform regarding the frequency resolution, inverse of it will appear on the figures showing the WT of the time-domain signal.

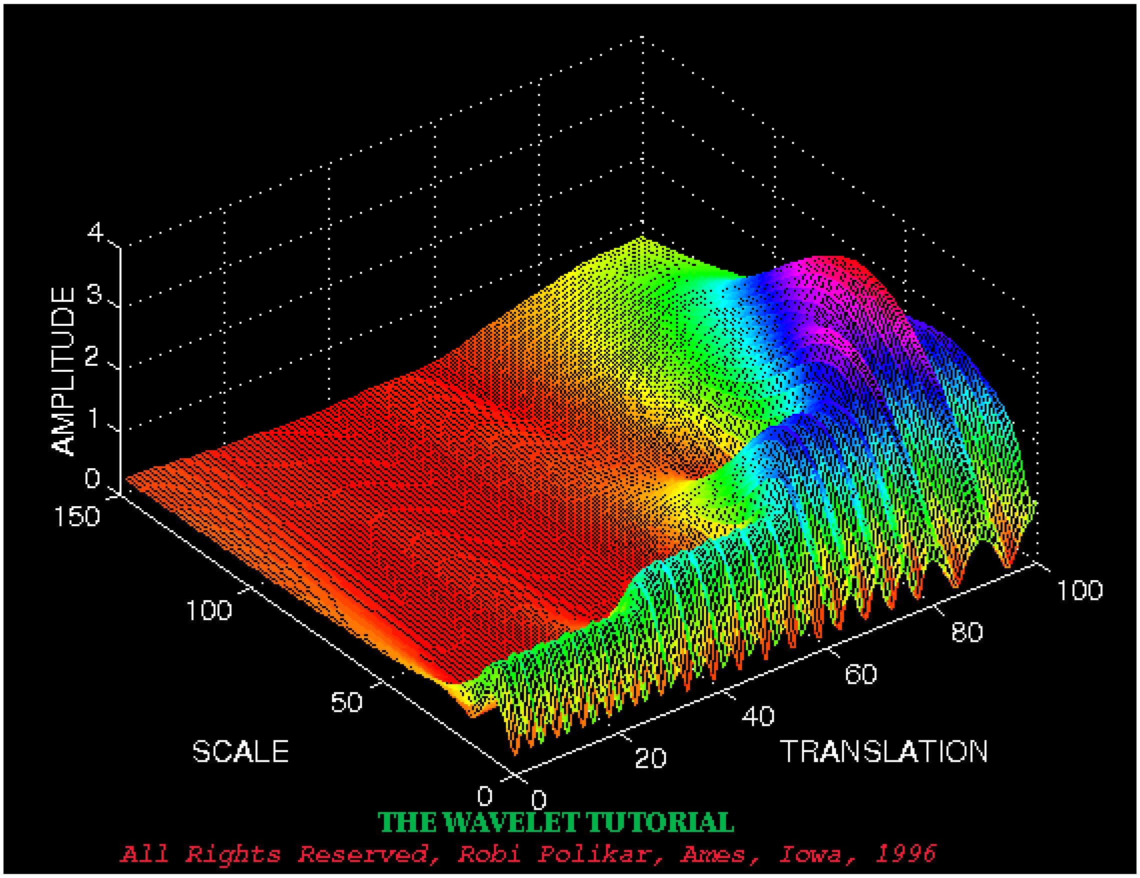


Figure 3.7

Note that in Figure 3.7 that smaller scales correspond to higher frequencies, i.e., frequency decreases as scale increases, therefore, that portion of the graph with scales around zero, actually correspond to highest frequencies in the analysis, and that with high scales correspond to lowest frequencies. Remember that the signal had 30 Hz (highest frequency) components first, and this appears at the lowest scale at a translations of 0 to 30. Then comes the 20 Hz component, second highest frequency, and so on. The 5 Hz component appears at the end of the translation axis (as expected), and at higher scales (lower frequencies) again as expected.

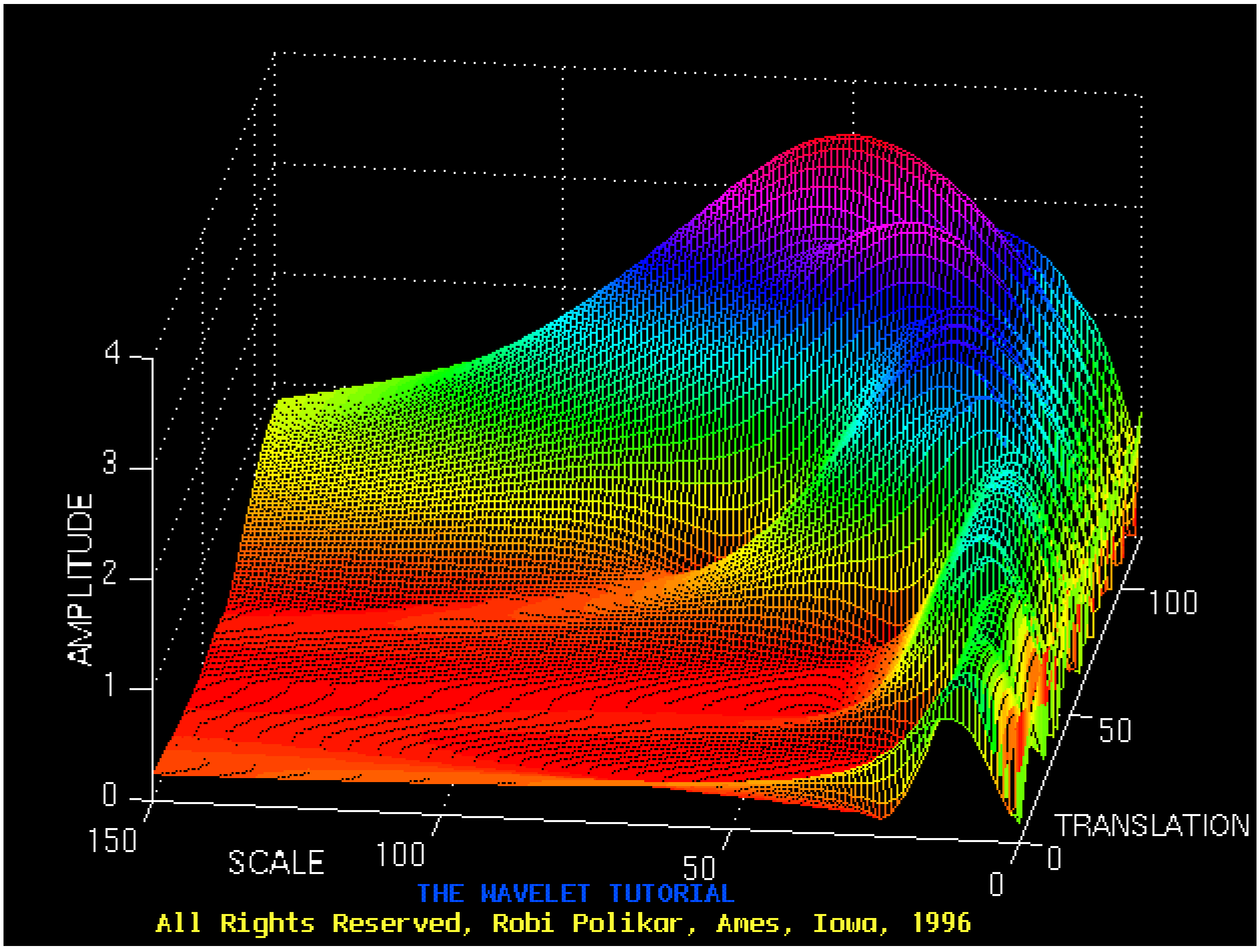


Figure 3.8

Now, recall these resolution properties: Unlike the STFT which has a constant resolution at all times and frequencies, the WT has a good time and poor frequency resolution at high frequencies, and good frequency and poor time resolution at low frequencies. Figure 3.8 shows the same WT in Figure 3.7 from another angle to better illustrate the resolution properties: In Figure 3.8, lower scales (higher frequencies) have **better scale resolution** (narrower in scale, which means that it is less ambiguous what the exact value of the scale) which correspond to **poorer frequency resolution**. Similarly, higher scales have scale frequency resolution (wider support in scale, which means it is more ambitious what the exact value of the scale is), which correspond to better frequency resolution of lower frequencies.

The axes in Figure 3.7 and 3.8 are normalized and should be evaluated accordingly. Roughly speaking the 100 points in the translation axis correspond to 1000 ms, and the 150 points on the scale axis correspond to a frequency band of 40 Hz (the numbers on the translation and scale axis **do not correspond to seconds and Hz, respectively**, they are just the number of samples in the computation).

TIME AND FREQUENCY RESOLUTIONS

In this section we will take a closer look at the resolution properties of the wavelet transform. Remember that the resolution problem was the main reason why we switched from STFT to WT.

The illustration in Figure 3.9 is commonly used to explain how time and frequency resolutions should be interpreted. Every box in Figure 3.9 corresponds to a value of the wavelet transform in the time-frequency plane. Note that boxes have a certain **non-zero** area, which implies that the value of a particular point in the time-frequency plane cannot be known. All the points in the time-frequency plane that falls into a box are represented by one value of the WT.

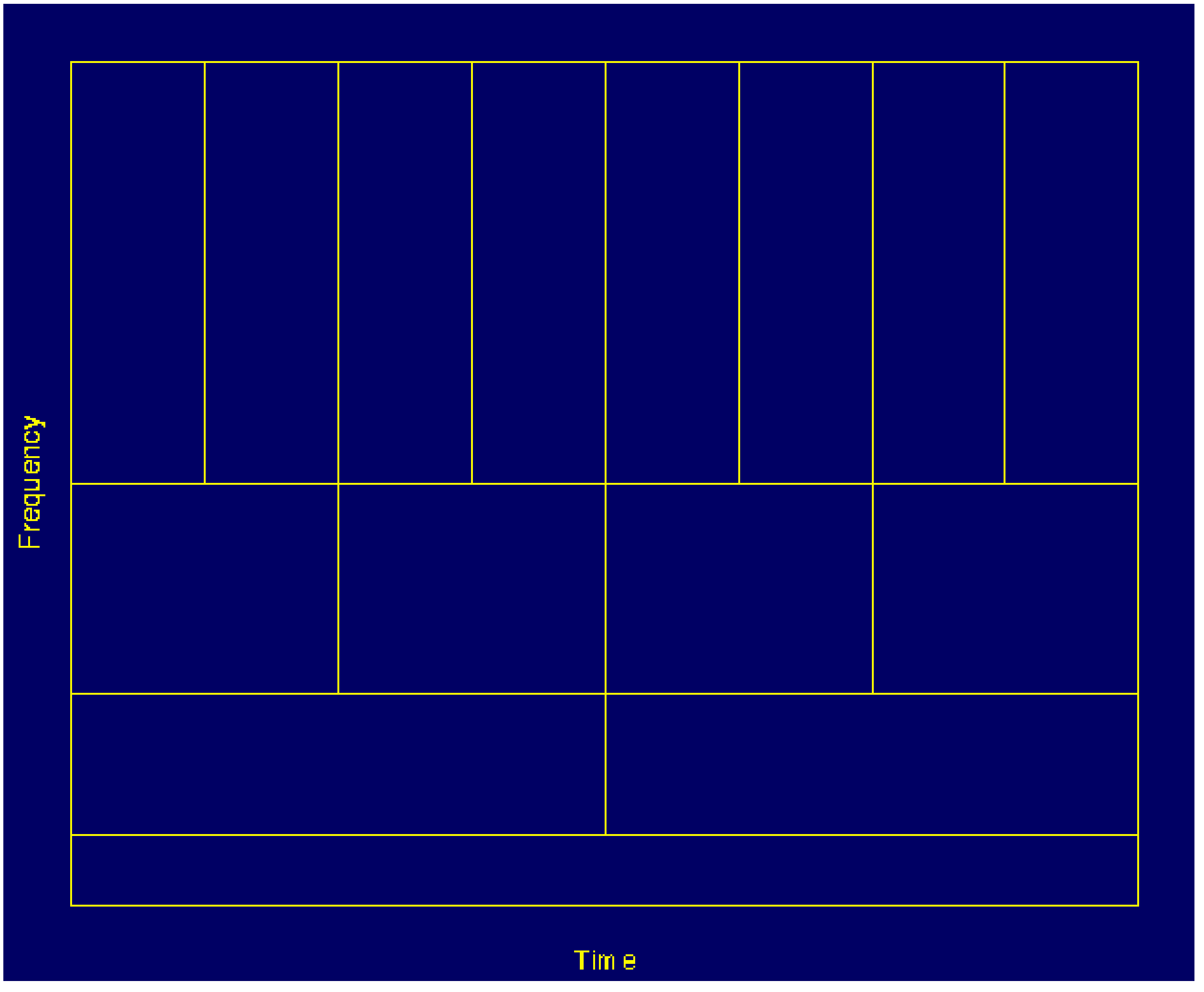


Figure 3.9

Let's take a closer look at Figure 3.9: First thing to notice is that although the widths and heights of the boxes change, the area is constant. That is each box represents an equal portion of the time-frequency plane, but giving different proportions to time and frequency. Note that at low frequencies, the height of the boxes are shorter (which corresponds to better frequency resolutions, since there is less ambiguity regarding the value of the exact frequency), but their widths are longer (which correspond to poor time resolution, since there is more ambiguity regarding the value of the exact time). At higher frequencies the width of the boxes decreases, i.e., the time resolution gets better, and the heights of the boxes increase, i.e., the frequency resolution gets poorer.

Before concluding this section, it is worthwhile to mention how the partition looks like in the case of STFT. Recall that in STFT the time and frequency resolutions are determined by the width of the analysis window, which is selected once for the entire analysis, i.e., both time and frequency resolutions are constant. Therefore the time-frequency plane consists of **squares** in the STFT case.

Regardless of the dimensions of the boxes, the areas of all boxes, both in STFT and WT, are the same and determined by **Heisenberg's inequality**. As a summary, the area of a box is fixed for each window function (STFT) or mother wavelet (CWT), whereas different windows or mother wavelets can result in different areas. However, **all areas are lower bounded by 1/4 \pi**. That is, we cannot reduce the areas of the boxes as much as we want due to the Heisenberg's uncertainty principle. On the other hand, for a given mother wavelet the dimensions of the boxes can be changed, while keeping the area the same. This is exactly what wavelet transform does.

THE WAVELET THEORY: A MATHEMATICAL APPROACH

This section describes the main idea of wavelet analysis theory, which can also be considered to be the underlying concept of most of the signal analysis techniques. The FT defined by Fourier use **basis functions** to analyze and reconstruct a function. **Every vector in a vector space can be written as a linear combination of the basis vectors in that vector space**, i.e., by multiplying the vectors by some constant numbers, and then by taking the summation of the products. The analysis of the signal involves the estimation of these constant numbers (transform coefficients, or Fourier coefficients, wavelet coefficients, etc). The synthesis, or the reconstruction, corresponds to computing the linear combination equation.

All the definitions and theorems related to this subject can be found in Keiser's book, **A Friendly Guide to Wavelets** but an introductory level knowledge of how basis functions work is necessary to understand the underlying principles of the wavelet theory. Therefore, this information will be presented in this section.

Basis Vectors

Note: Most of the equations include letters of the Greek alphabet. These letters are written out explicitly in the text with their names, such as **tau, psi, phi** etc. For capital letters, the first letter of the name has been capitalized, such as, **Tau, Psi, Phi** etc. Also, subscripts are shown by the underscore character **_**, and superscripts are shown by the ^ character. Also note that all letters or letter names written in bold type face represent vectors, Some important points are also written in bold face, but the meaning should be clear from the context.

A **basis** of a vector space **V** is a set of linearly independent vectors, such that any vector **v** in **V** can be written as a linear combination of these basis vectors. There may be more than one basis for a vector space. However, all of them have the same number of vectors, and this number is known as the **dimension** of the vector space. For example in two-dimensional space, the basis will have two vectors.

$$\mathbf{v} = \sum_k \nu^k \mathbf{b}_k$$

Equation 3.2

Equation 3.2 shows how any vector **v** can be written as a linear combination of the basis vectors **b_k** and the corresponding coefficients **nu^k** .

This concept, given in terms of vectors, can easily be generalized to functions, by replacing the basis vectors **b_k** with basis functions phi_k(t), and the vector **v** with a function f(t). Equation 3.2 then becomes

$$f(t) = \sum_k \mu_k \phi_k(t)$$

Equation 3.2a

The complex exponential (sines and cosines) functions are the basis functions for the FT. Furthermore, they are orthogonal functions, which provide some desirable properties for reconstruction.

Let f(t) and g(t) be two functions in L^2 [a,b]. (L^2 [a,b] denotes the set of square integrable functions in the interval [a,b]). The inner product of two functions is defined by Equation 3.3:

$$< f(t), g(t) >= \int_a^b f(t).g^*(t)\, dt$$

Equation 3.3

According to the above definition of the inner product, the CWT can be thought of as the inner product of the test signal with the basis functions psi_(tau ,s)(t):

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \int x(t).\psi^*_{\tau,s}(t)\, dt$$

Equation 3.4

where,

$$\psi_{\tau,s} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

Equation 3.5

This definition of the CWT shows that the wavelet analysis is a measure of similarity between the basis functions (wavelets) and the signal itself. Here the similarity is in the sense of similar frequency content. The calculated CWT coefficients refer to the closeness of the signal to the wavelet **at the current scale**.

This further clarifies the previous discussion on the correlation of the signal with the wavelet at a certain scale. If the signal has a major component of the frequency corresponding to the current scale, then the wavelet (the basis function) at the current scale will be **similar** or **close** to the signal at the particular location where this frequency component occurs. Therefore, the CWT coefficient computed at this point in the time-scale plane will be a relatively large number.

Inner Products, Orthogonality, and Orthonormality

Two vectors **v**, **w** are said to be **orthogonal** if their inner product equals zero:

$$< \mathbf{v}, \mathbf{w} >= \sum_n v_n w_n^* = 0$$

Equation 3.6

Similarly, two functions f and g are said to be orthogonal to each other if their inner product is zero:

$$< f(t), g(t) >= \int_a^b f(t).g^*(t)\, dt = 0$$

Equation 3.7

A set of vectors **{v_1, v_2, ...,v_n}** is said to be **orthonormal** , if they are pairwise orthogonal to each other, and all have length ``1''. This can be expressed as:

$$< v_m, v_n >= \delta_{mn}$$

Equation 3.8

Similarly, a set of functions {phi_k(t)}, k=1,2,3,..., is said to be orthonormal if

$$\int_a^b \phi_k(t)\phi_l^*(t)\,dt = 0 \;\; k \neq l \;\; \text{(orthogonality cond.)}$$

Equation 3.9

and

$$\int_a^b \{|\phi_k(t)|\}^2\,dx = 1$$

Equation 3.10

or equivalently

$$\int_a^b \phi_k(t)\phi_l^*(t)\,dt = \delta_{kl}$$

Equation 3.11

where, delta_{kl} is the **Kronecker delta** function, defined as:

$$\delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

Equation 3.12

As stated above, there may be more than one set of basis functions (or vectors). Among them, the orthonormal basis functions (or vectors) are of particular importance because of the nice properties they provide in finding these analysis coefficients. The orthonormal bases allow computation of these coefficients in a very simple and straightforward way using the orthonormality property.

For orthonormal bases, the coefficients, mu_k, can be calculated as

$$\mu_k = < f, \phi_k > = \int f(t).\phi_k^*(t)\, dt$$

Equation 3.13

and the function f(t) can then be reconstructed by Equation 3.2_a by substituting the mu_k coefficients. This yields

$$\begin{aligned} f(t) &= \sum_k \mu_k \phi_k(t) \\ &= \sum_k < f, \phi_k > \phi_k(t) \end{aligned}$$

Equation 3.14

Orthonormal bases may not be available for every type of application where a generalized version, **biorthogonal** bases can be used. The term ``biorthogonal" refers to two different bases which are orthogonal to each other, but each do not form an orthogonal set.

In some applications, however, biorthogonal bases also may not be available in which case **frames** can be used. Frames constitute an important part of wavelet theory, and interested readers are referred to Kaiser's book mentioned earlier.

Following the same order as in chapter 2 for the STFT, some examples of continuous wavelet transform are presented next. The figures given in the examples were generated by a program written to compute the CWT.

Before we close this section, I would like to include two mother wavelets commonly used in wavelet analysis. The Mexican Hat wavelet is defined as the second derivative of the Gaussian function:

$$w(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}}$$

Equation 3.15

which is

$$\psi(t) = \frac{1}{\sqrt{2\pi}\sigma^3} \left(e^{\frac{-t^2}{2\sigma^2}} . \left(\frac{t^2}{\sigma^2} - 1 \right) \right)$$

Equation 3.16

The Morlet wavelet is defined as

$$w(t) = e^{iat}.e^{-\frac{t^2}{2\sigma}}$$

Equation 3.16a

where **a** is a modulation parameter, and **sigma** is the scaling parameter that affects the width of the window.

EXAMPLES

All of the examples that are given below correspond to real-life non-stationary signals. These signals are drawn from a database signals that includes **event related potentials** of normal people, and patients with Alzheimer's disease. Since these are not test signals like simple sinusoids, it is not as easy to interpret them. They are shown here only to give an idea of how real-life CWTs look like.

The following signal shown in Figure 3.11 belongs to a normal person.

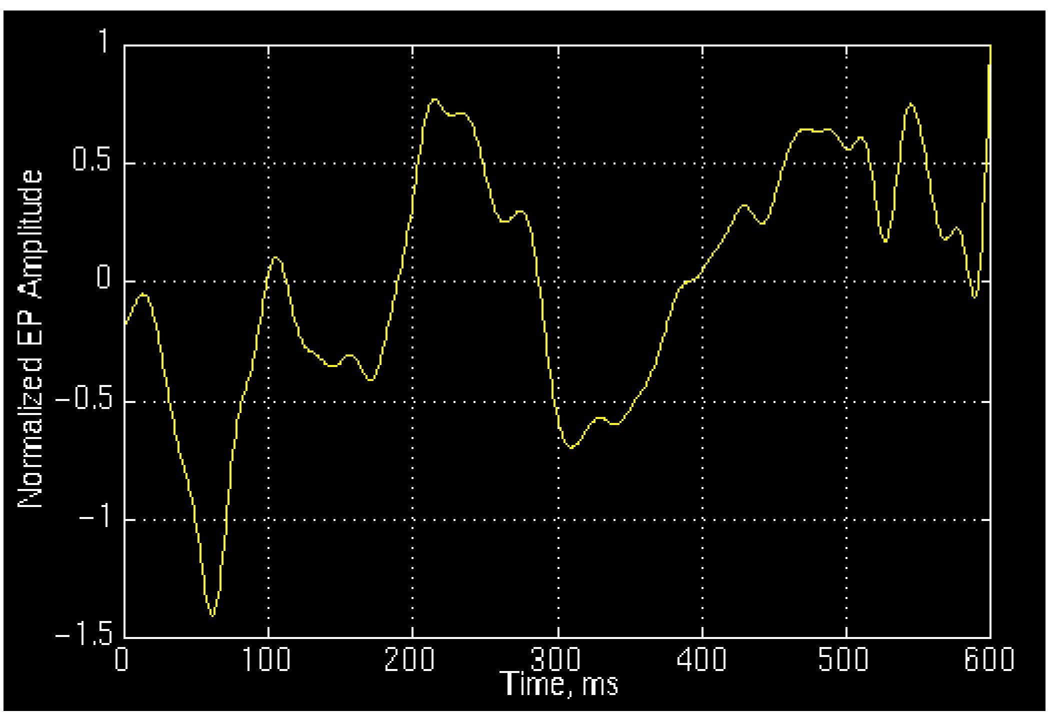


Figure 3.11

and the following is its CWT. The numbers on the axes are of no importance to us. Those numbers simply show that the CWT was computed at 350 translation and 60 scale locations on the translation-scale plane. The important point to note here is the fact that the computation is not a true **continuous** WT, as it is apparent from the computation at finite number of locations. This is only a discretized version of the CWT, which is explained later on this page. Note, however, that this is NOT discrete wavelet transform (DWT) which is the topic of Part IV of this tutorial.

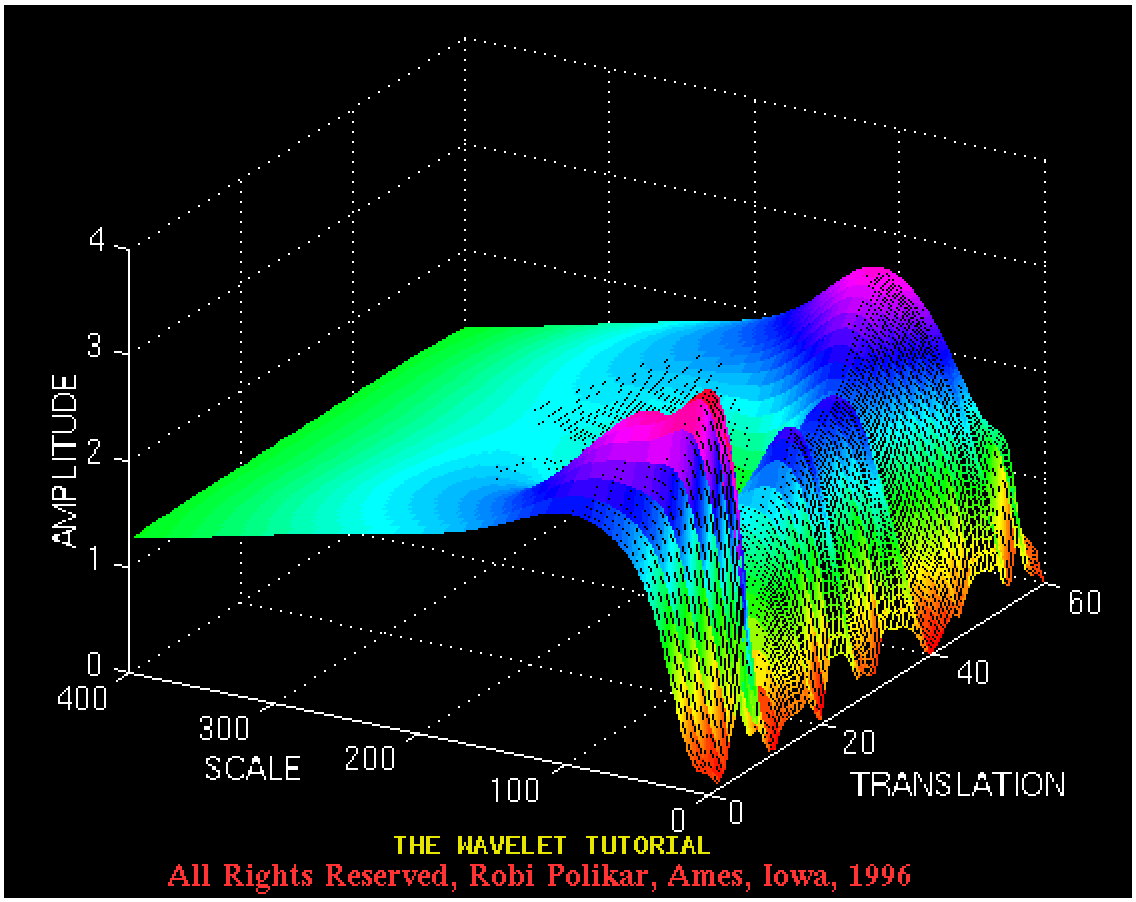


Figure 3.12

and the Figure 3.13 plots the same transform from a different angle for better visualization.

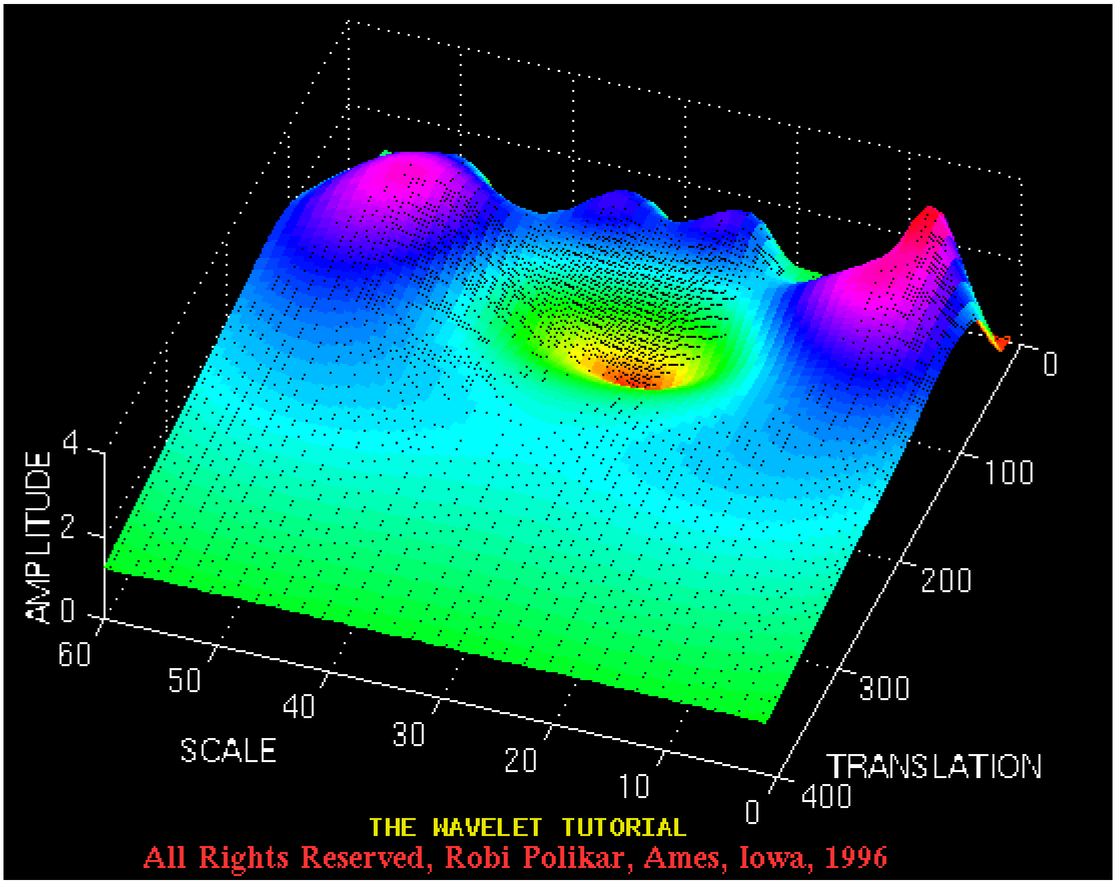


Figure 3.13

Figure 3.14 plots an event related potential of a patient diagnosed with Alzheimer's disease

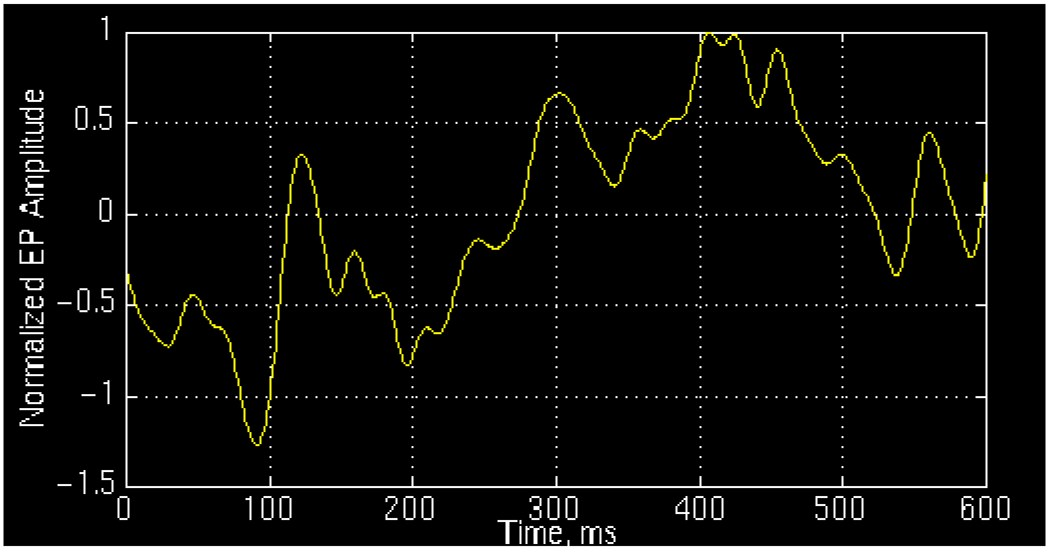


Figure 3.14

and Figure 3.15 illustrates its CWT:

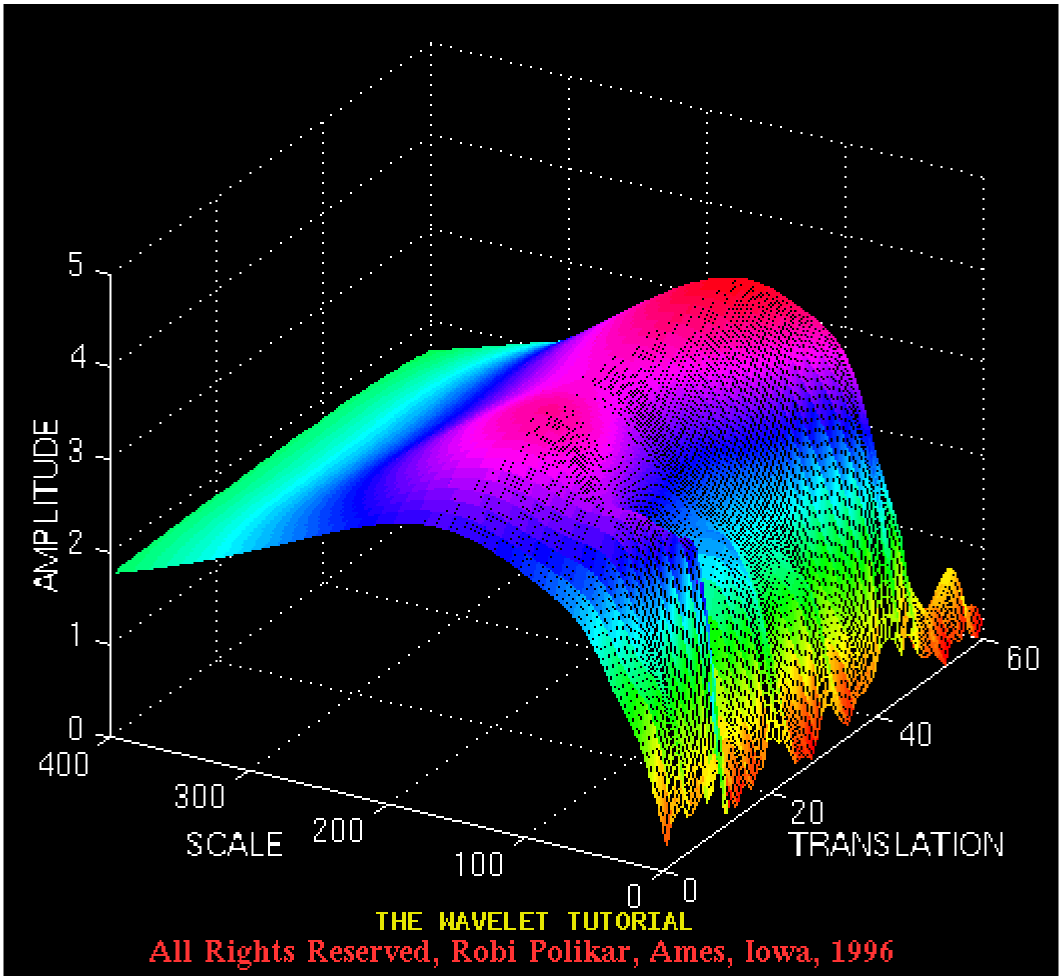


Figure 3.15

and here is another view from a different angle

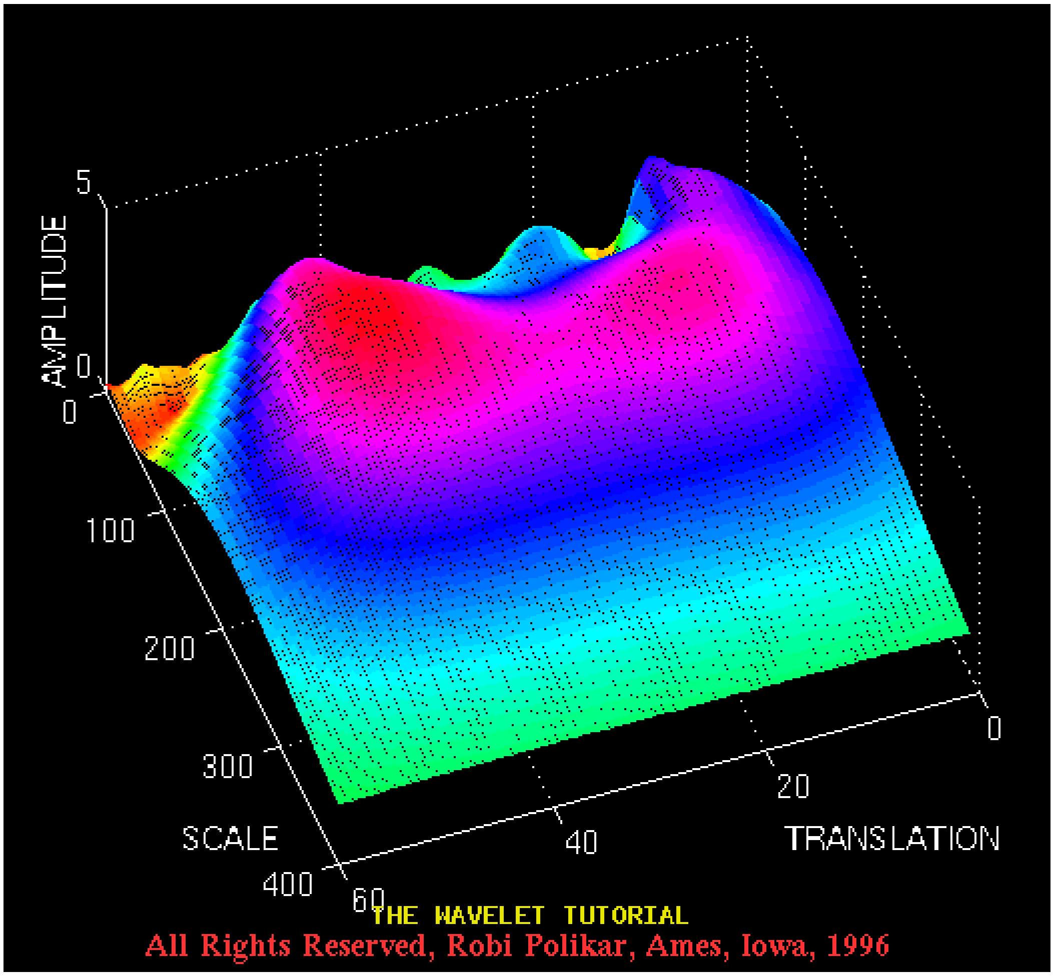


Figure 3.16

THE WAVELET SYNTHESIS

The continuous wavelet transform is a reversible transform, provided that Equation 3.18 is satisfied. Fortunately, this is a very non-restrictive requirement. The continuous wavelet transform is reversible if Equation 3.18 is satisfied, even though the basis functions are in general may not be orthonormal. The reconstruction is possible by using the following reconstruction formula:

$$x(t) = \frac{1}{c_\psi^2} \int_s \int_\tau \Psi_x^\psi(\tau, s) \frac{1}{s^2} \psi\left(\frac{t-\tau}{s}\right) d\tau\, ds$$

Equation 3.17 Inverse Wavelet Transform

where C_psi is a constant that depends on the wavelet used. The success of the reconstruction depends on this constant called, **the admissibility constant**, to satisfy the following **admissibility condition**:

$$c_{\psi} = \left\{ 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} \, d\xi \right\}^{1/2} < \infty$$

Equation 3.18 Admissibility Condition

where psi^hat(xi) is the FT of psi(t). Equation 3.18 implies that psi^hat(0) = 0, which is

$$\int \psi(t) \, dt = 0$$

Equation 3.19

As stated above, Equation 3.19 is not a very restrictive requirement since many wavelet functions can be found whose integral is zero. For Equation 3.19 to be satisfied, the wavelet must be oscillatory.

Discretization of the Continuous Wavelet Transform: The Wavelet Series

---

In today's world, computers are used to do most computations (well,...ok... almost all computations). It is apparent that neither the FT, nor the STFT, nor the CWT can be practically computed by using analytical equations, integrals, etc. It is therefore necessary to discretize the transforms. As in the FT and STFT, the most intuitive way of doing this is simply sampling the time-frequency (scale) plane. Again intuitively, sampling the plane with a uniform sampling rate sounds like the most natural choice. However, in the case of WT, the scale change can be used to reduce the sampling rate.

At higher scales (lower frequencies), the sampling rate can be decreased, according to Nyquist's rule. In other words, if the time-scale plane needs to be sampled with a sampling rate of **N_1** at scale **s_1**, the same plane can be sampled with a sampling rate of **N_2**, at scale **s_2**, where, **s_1 < s_2** (corresponding to frequencies **f1>f2**) and **N_2 < N_1**. The actual relationship between **N_1** and **N_2** is

$$N_2 = \frac{s_1}{s_2} N_1$$

Equation 3.20

or

$$N_2 = \frac{f_2}{f_1} N_1$$

Equation 3.21

In other words, at lower frequencies the sampling rate can be decreased which will save a considerable amount of computation time.

It should be noted at this time, however, that the discretization can be done in any way without any restriction as far as the analysis of the signal is concerned. If synthesis is not required, even the Nyquist criteria does not need to be satisfied. The restrictions on the discretization and the sampling rate become important if, and only if, the signal reconstruction is desired. Nyquist's sampling rate is the minimum sampling rate that allows the original **continuous time** signal to be reconstructed from its **discrete** samples. The basis vectors that are mentioned earlier are of particular importance for this reason.

As mentioned earlier, the wavelet **psi(tau,s)** satisfying Equation 3.18, allows reconstruction of the signal by Equation 3.17. However, this is true for the continuous transform. The question is: can we still reconstruct the signal if we discretize the time and scale parameters? The answer is ``yes", under certain conditions (as they always say in commercials: certain restrictions apply!!!).

The scale parameter **s** is discretized first on a logarithmic grid. The time parameter is then discretized **with respect to the scale parameter**, i.e., a different sampling rate is used for every scale. In other words, the sampling is done on the **dyadic** sampling grid shown in Figure 3.17:

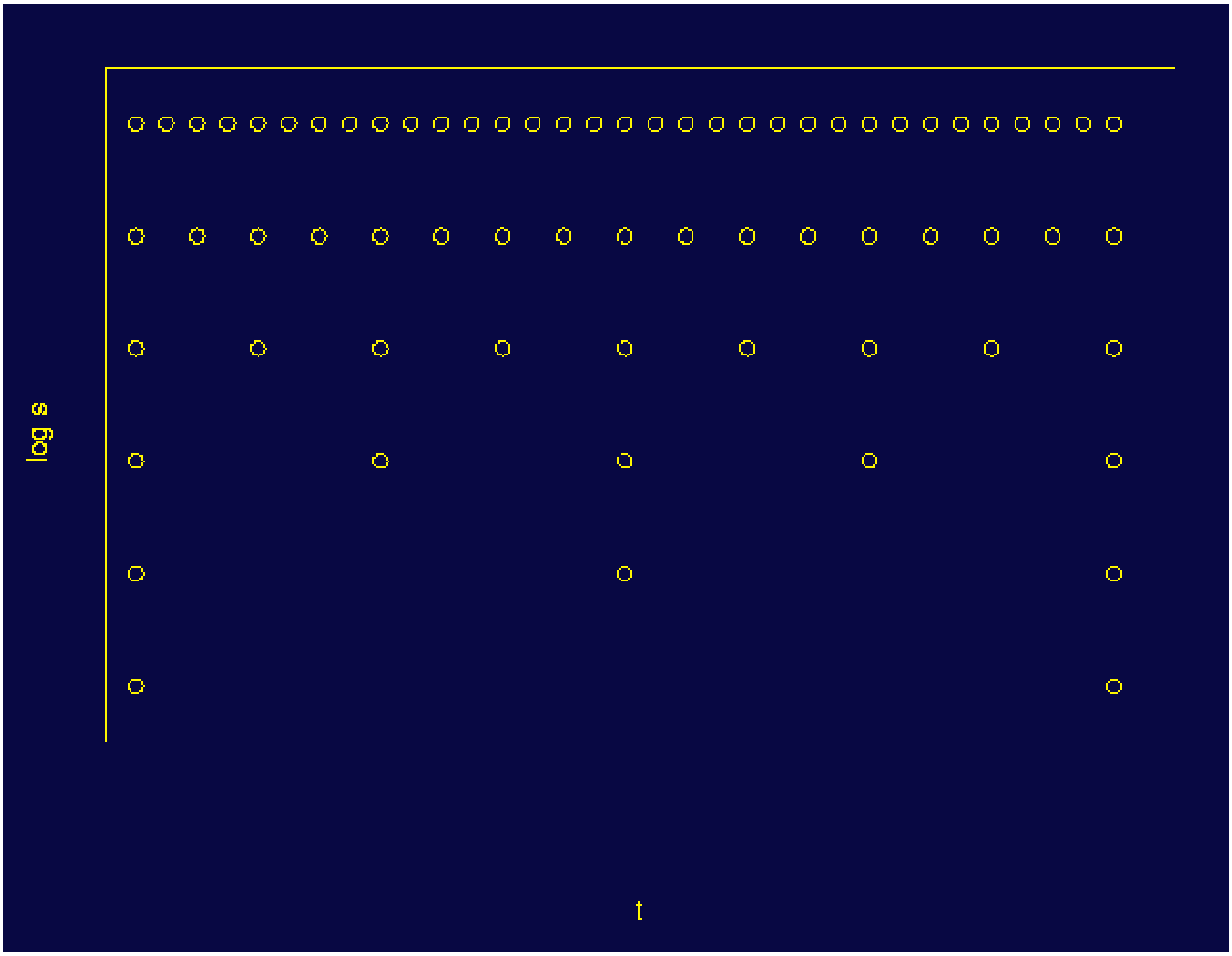


Figure 3.17

Think of the area covered by the axes as the entire time-scale plane. The CWT assigns a value to the continuum of points on this plane. Therefore, there are an infinite number of CWT coefficients. First consider the discretization of the scale axis. Among that infinite number of points, only a finite number are taken, using a logarithmic rule. The base of the logarithm depends on the user. The most common value is **2** because of its convenience. If 2 is chosen, only the scales 2, 4, 8, 16, 32, 64,...etc. are computed. If the value was 3, the scales 3, 9, 27, 81, 243,...etc. would have been computed. The time axis is then discretized according to the discretization of the scale axis. Since the discrete scale changes by factors of **2**, the sampling rate is reduced for the time axis by a factor of **2** at every scale.

Note that at the lowest scale (s=2), only 32 points of the time axis are sampled (for the particular case given in Figure 3.17). At the next scale value, s=4, the sampling rate of time axis is reduced by a factor of 2 since the scale is increased by a factor of 2, and therefore, only 16 samples are taken. At the next step, s=8 and 8 samples are taken in time, and so on.

Although it is called the time-scale plane, it is more accurate to call it the **translation-scale** plane, because ``time" in the transform domain actually corresponds to the shifting of the wavelet in time. For the wavelet series, the actual time is still continuous.

Similar to the relationship between continuous Fourier transform, Fourier series and the discrete Fourier transform, there is a continuous wavelet transform, a semi-discrete wavelet transform (also known as wavelet series) and a discrete wavelet transform.

Expressing the above discretization procedure in mathematical terms, the scale discretization is **s = s_0^j**, and translation discretization is **tau = k.s_0^j.tau_0** where **s_0>1** and **tau_0>0**. Note, how the translation discretization is dependent on scale discretization with **s_0**.

The continuous wavelet function

$$\psi_{\tau,s} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

Equation 3.22

$$\psi_{j,k}(t) = s_0^{-j/2}\psi(s_0^{-j}t - k\tau_0)$$

Equation 3.23

by inserting **s = s_0^j** , and **tau = k.s_0^j.tau_0** .

If **{psi_(j,k)}** constitutes an orthonormal basis, the wavelet series transform becomes

$$\Psi_x^{\psi_{j,k}} = \int x(t)\psi_{j,k}^*(t)\,dt$$

Equation 3.24

or

$$x(t) = c_\psi \sum_j \sum_k \Psi_x^{\psi_{j,k}}\psi_{j,k}(t)$$

Equation 3.25

A wavelet series requires that **{psi_(j,k)}** are either orthonormal, biorthogonal, or frame. If **{psi_(j,k)}** are not orthonormal, Equation 3.24 becomes

$$\Psi_x^{\psi_{j,k}} = \int x(t)\hat{\psi_{j,k}^*}(t)\,dt$$

Equation 3.26

where **hat{ psi_{j,k}^*(t)}** , is either the **dual biorthogonal basis** or **dual frame** (Note that * denotes the conjugate).

If **{psi_(j,k) }** are orthonormal or biorthogonal, the transform will be non-redundant, where as if they form a frame, the transform will be redundant. On the other hand, it is much easier to find frames than it is to find orthonormal or biorthogonal bases.

The following analogy may clear this concept. Consider the whole process as looking at a particular object. The human eyes first determine the coarse view which depends on the distance of the eyes to the object. This corresponds to adjusting the scale parameter **s_0^(-j)**. When looking at a very close object, with great detail, **j** is negative and large (low scale, high frequency, analyses the detail in the signal). Moving the head (or eyes) very slowly and with very small increments (of angle, of distance, depending on the object that is being viewed), corresponds to small values of **tau = k.s_0^j.tau_0**. Note that when **j** is negative and large, it corresponds to small changes in time, **tau**, (high sampling rate) and large changes in **s_0^-j** (low scale, high frequencies, where the sampling rate is high). The scale parameter can be thought of as magnification too.

How low can the sampling rate be and still allow reconstruction of the signal? This is the main question to be answered to optimize the procedure. The most convenient value (in terms of programming) is found to be ``2" for s_0 and "1" for tau. Obviously, when the sampling rate is forced to be as low as possible, the number of available orthonormal wavelets is also reduced.

The continuous wavelet transform examples that were given in this chapter were actually the wavelet series of the given signals. The parameters were chosen depending on the signal. Since the reconstruction was not needed, the sampling rates were sometimes far below the critical value where s_0 varied from 2 to 10, and tau_0 varied from 2 to 8, for different examples.

This concludes Part III of this tutorial. I hope you now have a basic understanding of what the wavelet transform is all about. There is one thing left to be discussed however. Even though the discretized wavelet transform can be computed on a computer, this computation may take anywhere from a couple seconds to couple hours depending on your signal size and the resolution you want. An amazingly fast algorithm is actually available to compute the wavelet transform of a signal. The discrete wavelet transform (DWT) is introduced in the final chapter of this tutorial, in Part IV.

Let's meet at the grand finale, shall we?

THE WAVELET TUTORIAL

PART IV

BY

ROBI POLIKAR

MULTIRESOLUTION ANALYSIS:

THE DISCRETE WAVELET TRANSFORM

WHY IS THE DISCRETE WAVELET TRANSFORM NEEDED?

Although the discretized continuous wavelet transform enables the computation of the continuous wavelet transform by computers, it is not a true discrete transform. As a matter of fact, the wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned. This redundancy, on the other hand, requires a significant amount of computation time and resources. The discrete wavelet transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time.

The DWT is considerably easier to implement when compared to the CWT. The basic concepts of the DWT will be introduced in this section along with its properties and the algorithms used to compute it. As in the previous chapters, examples are provided to aid in the interpretation of the DWT.

THE DISCRETE WAVELET TRANSFORM (DWT)

The foundations of the DWT go back to 1976 when Croiser, Esteban, and Galand devised a technique to decompose discrete time signals. Crochiere, Weber, and Flanagan did a similar work on coding of speech signals in the same year. They named their analysis scheme as **subband coding**. In 1983, Burt defined a technique very similar to subband coding and named it **pyramidal coding** which is also known as multiresolution analysis. Later in 1989, Vetterli and Le Gall made some improvements to the subband coding scheme, removing the existing redundancy in the pyramidal coding scheme. Subband coding is explained below. A detailed coverage of the discrete wavelet transform and theory of multiresolution analysis can be found in a number of articles and books that are available on this topic, and it is beyond the scope of this tutorial.

THE SUBBAND CODING AND THE MULTIRESOLUTION ANALYSIS

The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. Recall that the CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity. The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor *n* reduces the number of samples in the signal *n* times.

Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of *n* increases the number of samples in the signal by a factor of *n*.

Although it is not the only possible choice, DWT coefficients are usually sampled from the CWT on a dyadic grid, i.e., $s_0 = 2$ and $t_0 = 1$, yielding $s=2^j$ and $t = k*2^j$, as described in Part 3. Since the signal is a discrete time function, the terms function and sequence will be used interchangeably in the following discussion. This sequence will be denoted by x[n], where n is an integer.

The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz.

The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to 2p radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be p radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to p rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p/2 radians instead of p radians. Simply discarding every other sample will **subsample** the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. Note, however, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale.

This procedure can mathematically be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n - k]$$

Having said that, we now look how the DWT is actually computed: The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p/2 radians instead of p. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n] \cdot g[2k - n]$$

$$y_{low}[k] = \sum_{n} x[n] \cdot h[2k - n]$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence double the frequency resolution). Figure 4.1 illustrates this procedure, where x[n] is the original signal to be decomposed, and h[n] and g[n] are lowpass and highpass filters, respectively. The bandwidth of the signal at every level is marked on the figure as "f".

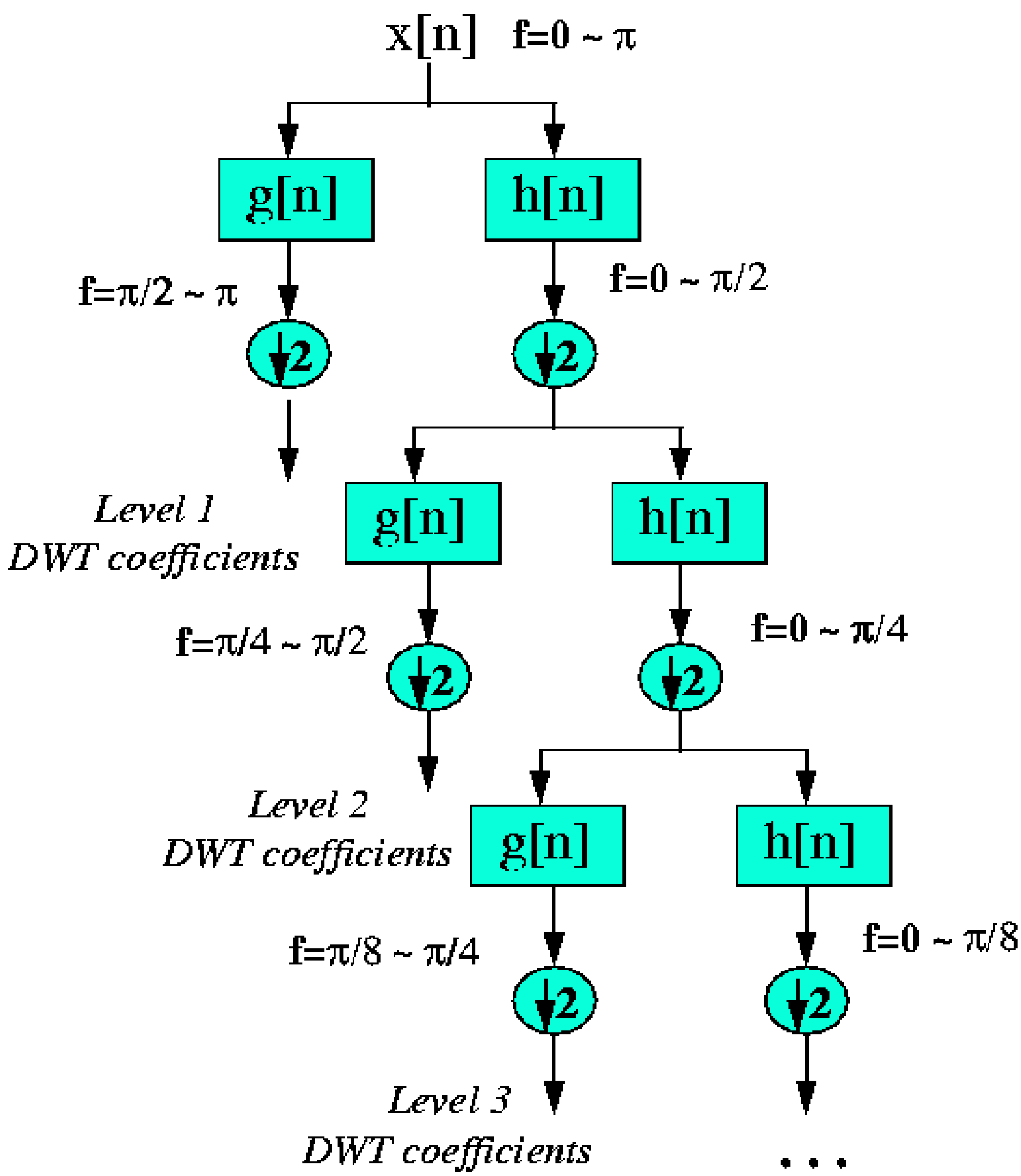


Figure 4.1. The Subband Coding Algorithm

As an example, suppose that the original signal x[n] has 512 sample points, spanning a frequency band of zero to p rad/s. At the first decomposition level, the signal is passed through the highpass and lowpass filters, followed by subsampling by 2. The output of the highpass filter has 256 points (hence half the time resolution), but it only spans the frequencies p/2 to p rad/s (hence double the frequency resolution). These 256 samples constitute the first level of DWT coefficients. The output of the lowpass filter also has 256 samples, but it spans the other half of the frequency band, frequencies from 0 to p/2 rad/s. This signal is then passed through the same lowpass and highpass filters for further decomposition. The output of the second lowpass filter followed by subsampling has 128 samples spanning a frequency band of 0 to p/4 rad/s, and the output of the second highpass filter followed by subsampling has 128 samples spanning a frequency band of p/4 to p/2 rad/s. The second highpass filtered signal constitutes the second level of DWT coefficients. This signal has half the time resolution, but twice the frequency resolution of the first level signal. In other words, time resolution has decreased by a factor of 4, and frequency resolution has increased by a factor of 4 compared to the original signal. The lowpass filter output is then filtered once again for further decomposition. This process continues until two samples are left. For this specific example there would be 8 levels of decomposition, each having half the number of samples of the previous level. The DWT of the original signal is then obtained by concatenating all coefficients starting from the last level of decomposition (remaining two samples, in this case). The DWT will then have the same number of coefficients as the original signal.

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, as happens most often, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure in effect offers a good time resolution at high frequencies, and good frequency resolution at low frequencies. Most practical signals encountered are of this type.

The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction. Figure 4.2 illustrates an example of how DWT signals look like and how data reduction is provided. Figure 4.2a shows a typical 512-sample signal that is normalized to unit amplitude. The horizontal axis is the number of samples, whereas the vertical axis is the normalized amplitude. Figure 4.2b shows the 8 level DWT of the signal in Figure 4.2a. The last 256 samples in this signal correspond to the highest frequency band in the signal, the previous 128 samples correspond to the second highest frequency band and so on. It should be noted that only the first 64 samples, which correspond to lower frequencies of the analysis, carry relevant information and the rest of this signal has virtually no information. Therefore, all but the first 64 samples can be discarded without any loss of information. This is how DWT provides a very effective data reduction scheme.

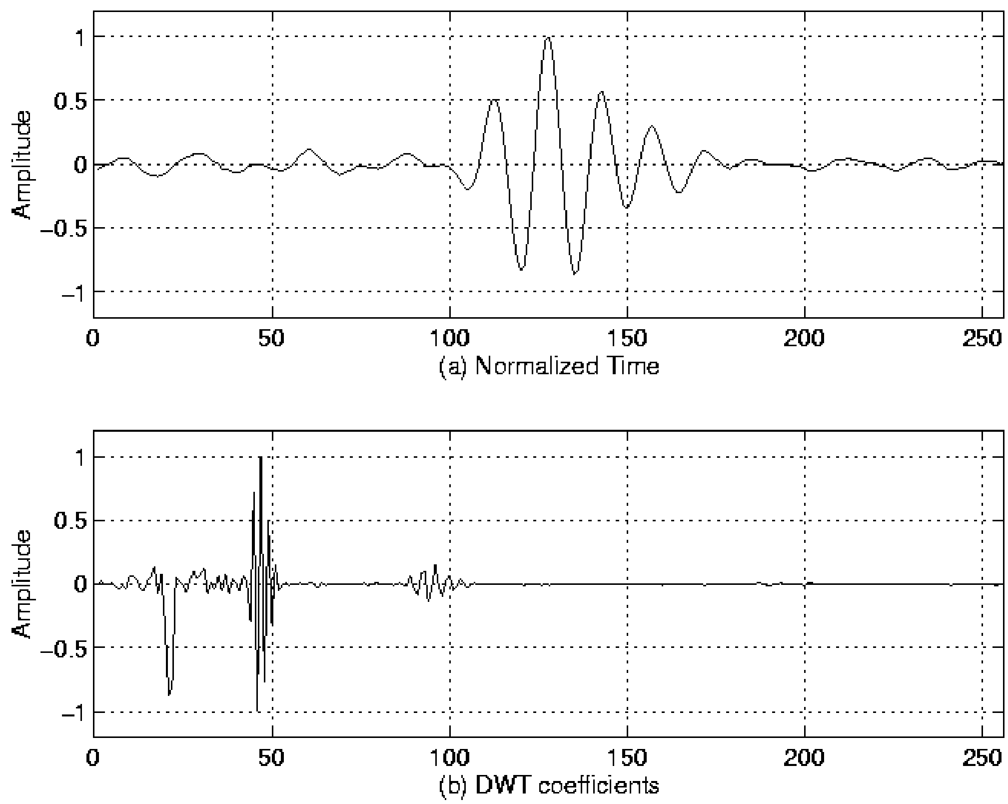


Figure 4.2 Example of a DWT

We will revisit this example, since it provides important insight to how DWT should be interpreted. Before that, however, we need to conclude our mathematical analysis of the DWT.

One important property of the discrete wavelet transform is the relationship between the impulse responses of the highpass and lowpass filters. The highpass and lowpass filters are not independent of each other, and they are related by

$$g[L-1-n] = (-1)^n \cdot h[n]$$

where g[n] is the highpass, h[n] is the lowpass filter, and L is the filter length (in number of points). Note that the two filters are odd index alternated reversed versions of each other. Lowpass to highpass conversion is provided by the $(-1)^n$ term. Filters satisfying this condition are commonly used in signal processing, and they are known as the Quadrature Mirror Filters (QMF). The two filtering and subsampling operations can be expressed by

$$y_{high}[k] = \sum_n x[n] \cdot g[-n+2k]$$
$$y_{low}[k] = \sum_n x[n] \cdot h[-n+2k]$$

The reconstruction in this case is very easy since halfband filters form orthonormal bases. The above procedure is followed in reverse order for the reconstruction. The signals at every level are upsampled by two, passed through the synthesis filters g'[n], and h'[n] (highpass and lowpass, respectively), and then added. The interesting point here is that the analysis and synthesis filters are identical to each other, except for a time reversal. Therefore, the reconstruction formula becomes (for each layer)

$$x[n] = \sum_{k=-\infty}^{\infty} \left(y_{high}[k] \cdot g[-n+2k]\right) + \left(y_{low}[k] \cdot h[-n+2k]\right)$$

However, if the filters are not ideal halfband, then perfect reconstruction cannot be achieved. Although it is not possible to realize ideal filters, under certain conditions it is possible to find filters that provide perfect reconstruction. The most famous ones are the ones developed by Ingrid Daubechies, and they are known as Daubechies' wavelets.

Note that due to successive subsampling by 2, the signal length must be a power of 2, or at least a multiple of power of 2, in order this scheme to be efficient. The length of the signal determines the number of levels that the signal can be decomposed to. For example, if the signal length is 1024, ten levels of decomposition are possible.

Interpreting the DWT coefficients can sometimes be rather difficult because the way DWT coefficients are presented is rather peculiar. To make a real long story real short, DWT coefficients of each level are concatenated, starting with the last level. An example is in order to make this concept clear:

Suppose we have a 256-sample long signal sampled at 10 MHZ and we wish to obtain its DWT coefficients. Since the signal is sampled at 10 MHz, the highest frequency component that exists in the signal is 5 MHz. At the first level, the signal is passed through the lowpass filter h[n], and the highpass filter g[n], the outputs of which are subsampled by two. The highpass filter output is the first level DWT coefficients. There are 128 of them, and they represent the signal in the [2.5 5] MHz range. These 128 samples are the last 128 samples plotted. The lowpass filter output, which also has 128 samples, but spanning the frequency band of [0 2.5] MHz, are further decomposed by passing them through the same h[n] and g[n]. The output of the second highpass filter is the level 2 DWT coefficients and these 64 samples precede the 128 level 1 coefficients in the plot. The output of the second lowpass filter is further decomposed, once again by passing it through the filters h[n] and g[n]. The output of the third highpass filter is the level 3 DWT coefficiets. These 32 samples precede the level 2 DWT coefficients in the plot.

The procedure continues until only 1 DWT coefficient can be computed at level 9. This one coefficient is the first to be plotted in the DWT plot. This is followed by 2 level 8 coefficients, 4 level 7 coefficients, 8 level 6 coefficients, 16 level 5 coefficients, 32 level 4 coefficients, 64 level 3 coefficients, 128 level 2 coefficients and finally 256 level 1 coefficients. Note that less and less number of samples is used at lower frequencies, therefore, the time resolution decreases as frequency decreases, but since the frequency interval also decreases at low frequencies, the frequency resolution increases. Obviously, the first few coefficients would not carry a whole lot of information, simply due to greatly reduced time resolution. To illustrate this richly bizarre DWT representation let us take a look at a real world signal. Our original signal is a 256-sample long ultrasonic signal, which was sampled at 25 MHz. This signal was originally generated by using a 2.25 MHz transducer, therefore the main spectral component of the signal is at 2.25 MHz. The last 128 samples correspond to [6.25 12.5] MHz range. As seen from the plot, no information is available here, hence these samples can be discarded without any loss of information. The preceding 64 samples represent the signal in the [3.12 6.25] MHz range, which also does not carry any significant information. The little glitches probably correspond to the high frequency noise in the signal. The preceding 32 samples represent the signal in the [1.5 3.1] MHz range. As you can see, the majority of the signal's energy is focused in these 32 samples, as we expected to see. The previous 16 samples correspond to [0.75 1.5] MHz and the peaks that are seen at this level probably represent the lower frequency envelope of the signal. The previous samples probably do not carry any other significant information. It is safe to say that we can get by with the 3rd and 4th level coefficients, that is we can represent this 256 sample long signal with 16+32=48 samples, a significant data reduction which would make your computer quite happy.

One area that has benefited the most from this particular property of the wavelet transforms is image processing. As you may well know, images, particularly high-resolution images, claim a lot of disk space. As a matter of fact, if this tutorial is taking a long time to download, that is mostly because of the images. DWT can be used to reduce the image size without losing much of the resolution. Here is how:

For a given image, you can compute the DWT of, say each row, and discard all values in the DWT that are less then a certain threshold. We then save only those DWT coefficients that are above the threshold for each row, and when we need to reconstruct the original image, we simply pad each row with as many zeros as the number of discarded coefficients, and use the inverse DWT to reconstruct each row of the original image. We can also analyze the image at different frequency bands, and reconstruct the original image by using only the coefficients that are of a particular band. I will try to put sample images hopefully soon, to illustrate this point.

Another issue that is receiving more and more attention is carrying out the decomposition (subband coding) not only on the lowpass side but on both sides. In other words, zooming into both low and high frequency bands of the signal separately. This can be visualized as having both sides of the tree structure of Figure 4.1. What result is what is known as the **wavelet packages**. We will not discuss wavelet packages in this here, since it is beyond the scope of this tutorial. Anyone who is interested in wavelet packages, or more information on DWT can find this information in any of the numerous texts available in the market.

And this concludes our mini series of wavelet tutorial. If I could be of any assistance to anyone struggling to understand the wavelets, I would consider the time and the effort that went into this tutorial well spent. I would like to remind that this tutorial is neither a complete nor a through coverage of the wavelet transforms. It is merely an overview of the concept of wavelets and it was intended to serve as a first reference for those who find the available texts on wavelets rather complicated. There might be many structural and/or technical mistakes, and I would appreciate if you could point those out to me. Your feedback is of utmost importance for the success of this tutorial.

Thank you very much for your interest in The Wavelet Tutorial.