Short Time Fourier Transform (STFT)

The Short Time Fourier Transform (STFT) is a special flavour of a Fourier transform where you can see how the frequencies in the signal change through time. It works by slicing up your signal into many small segments and taking the Fourier transform of each of these. The result is usually a colour plot which shows frequency against time.

In the STFT, we perform a series of windowing and FT operations: at each time shift t, we apply a window $\varpi(t' - t)$ centered around t, and then calculate the Fourier Transform of $f(t')\varpi(t' - t)$. For a function $f(t)$, the STFT is defined as follows:

$$F(t,u) = \int_{-\infty}^{\infty} f(t')\varpi(t' - t)e^{-iut'}dt'$$

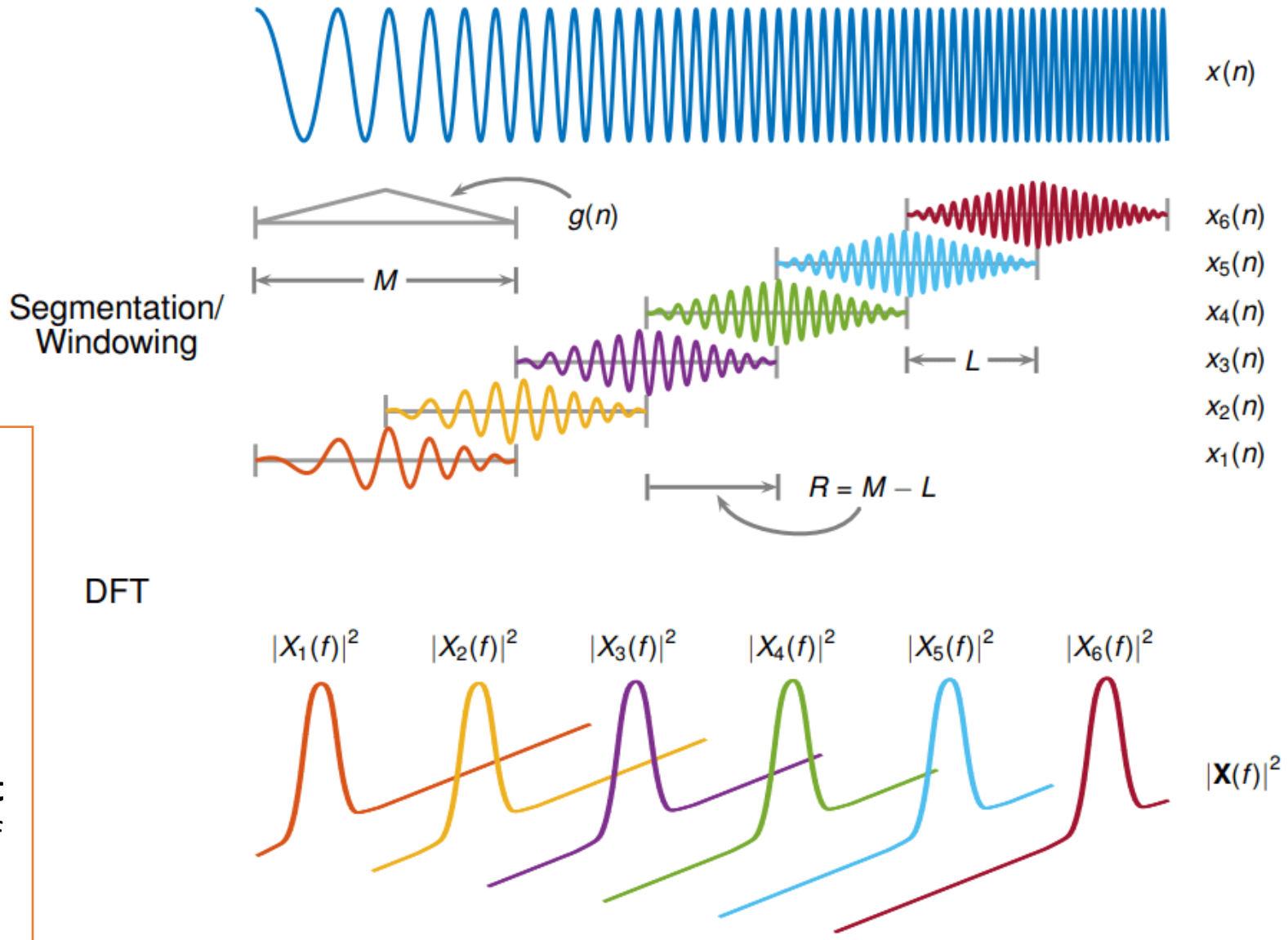
where u now represents the angular frequencies. In discretized form, the expression above can be written as

$$F(n,u) = \sum_{m=-\infty}^{\infty} f_m\, \varpi(n - m)e^{-ium}$$

But how to pick the STFT window?

If we expect a certain signal of interest to have rapidly fluctuating properties, we should use a narrow window $\varpi(t)$ (in order to enable fine temporal resolution). In contrast, if we expect our signal to have slowly fluctuating properties, we should use a wide window $\varpi(t)$ (in order to enable fine frequency resolution). However, there is generally no optimal way to pick the STFT window. Indeed, this trade-off between temporal and frequency resolution is a fundamental feature of the STFT.

The STFT of a signal is computed by sliding an analysis window of length M over the signal and calculating the discrete Fourier transform (DFT) of each segment of windowed data. The window hops over the original signal at intervals of R samples, equivalent to L = M – R samples of overlap between adjoining segments. **Most window functions taper off at the edges to avoid spectral ringing.** The DFT of each windowed segment is added to a complex-valued matrix that contains the magnitude and phase for each point in time and frequency.



https://www.mathworks.com/help/signal/ref/stft.html

Short Time Fourier Transform (STFT)

$$F(n,k) = \sum_{m=n-(N_{\varpi}-1)}^{n} f_m \, \varpi(n-m) e^{-i2\pi mk/N}$$

Note that $F(n,k)$ is a function of both time and frequency and now both the time and frequency variables are discrete. Note that $\omega_k = 2\pi k/N$ with N being the number of discrete frequency channels used in the STFT, and N_{ϖ} the length of the window function.

The variable n denotes the location of the analysis window along the time axis, and the segment of time delimited by the window is frequently referred to as the **analysis frame**. The variable k is a frequency index, and is sometimes referred to as a **frequency bin**.

Here the variable m is a "dummy" time argument and the variable n identifies the location of the short segment of the original time function as it is extracted using the window $\varpi(n-m)$, which moves along the m-axis according to the value of n.

Short Time Fourier Transform (STFT)

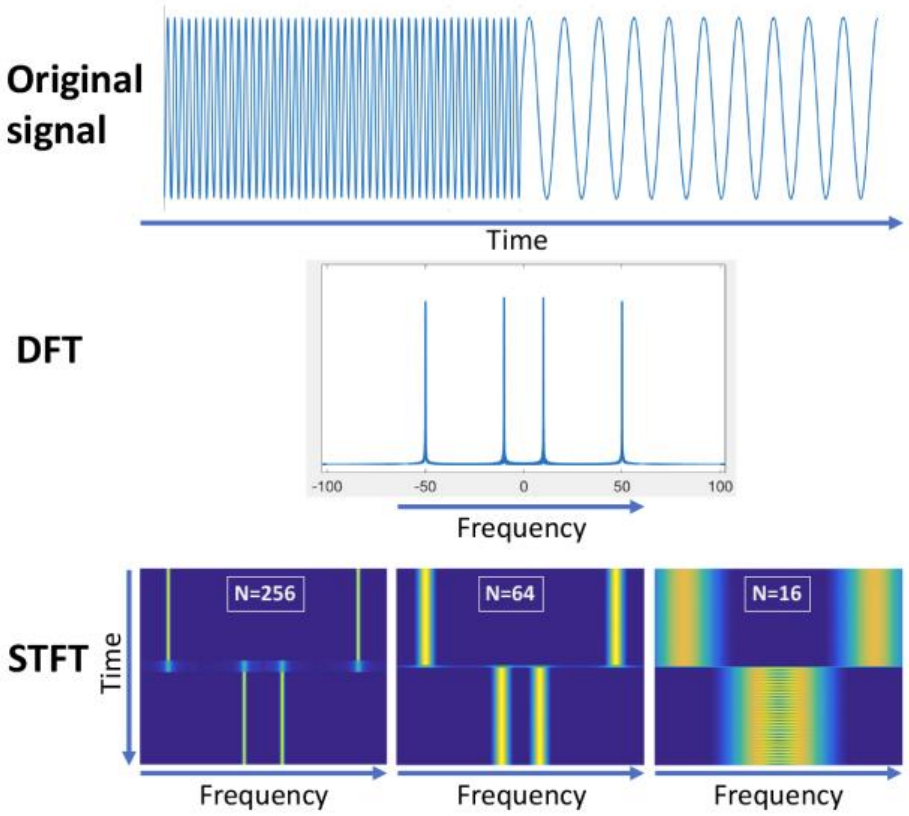
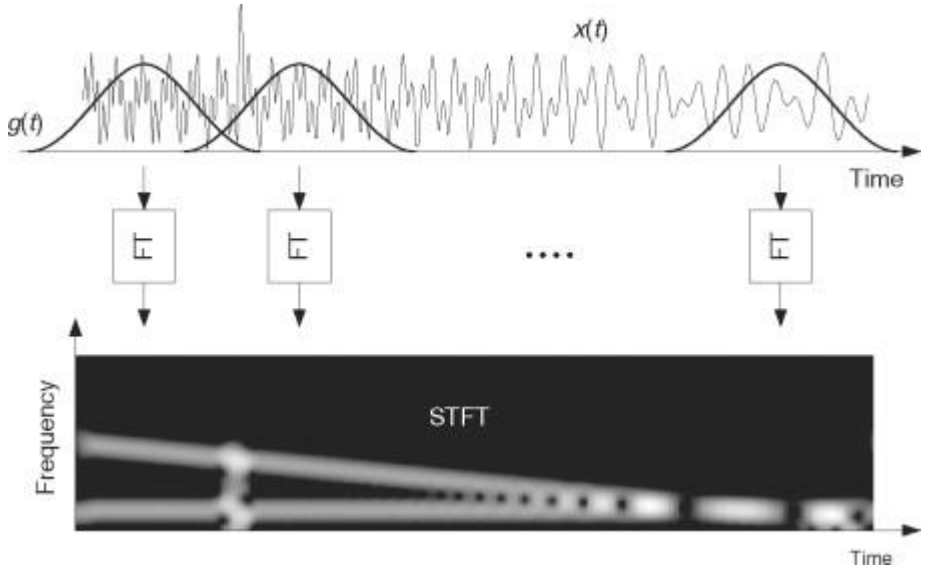


Figure 16.1: DFT vs STFT of a signal that has a high frequency for a while, then switches to a lower frequency. Note that the DFT has no temporal resolution (all of time is shown together in the frequency plot). In contrast, the STFT provides both temporal and frequency resolution: for a given time, we get a spectrum. This enables us to better represent signals with spectra that change over time. However, the STFT presents a fundamental trade-off between time resolution and frequency resolution. This trade-off is controlled by the choice of STFT window (note the improved time resolution but worsened frequency resolution as we make the STFT window narrower).

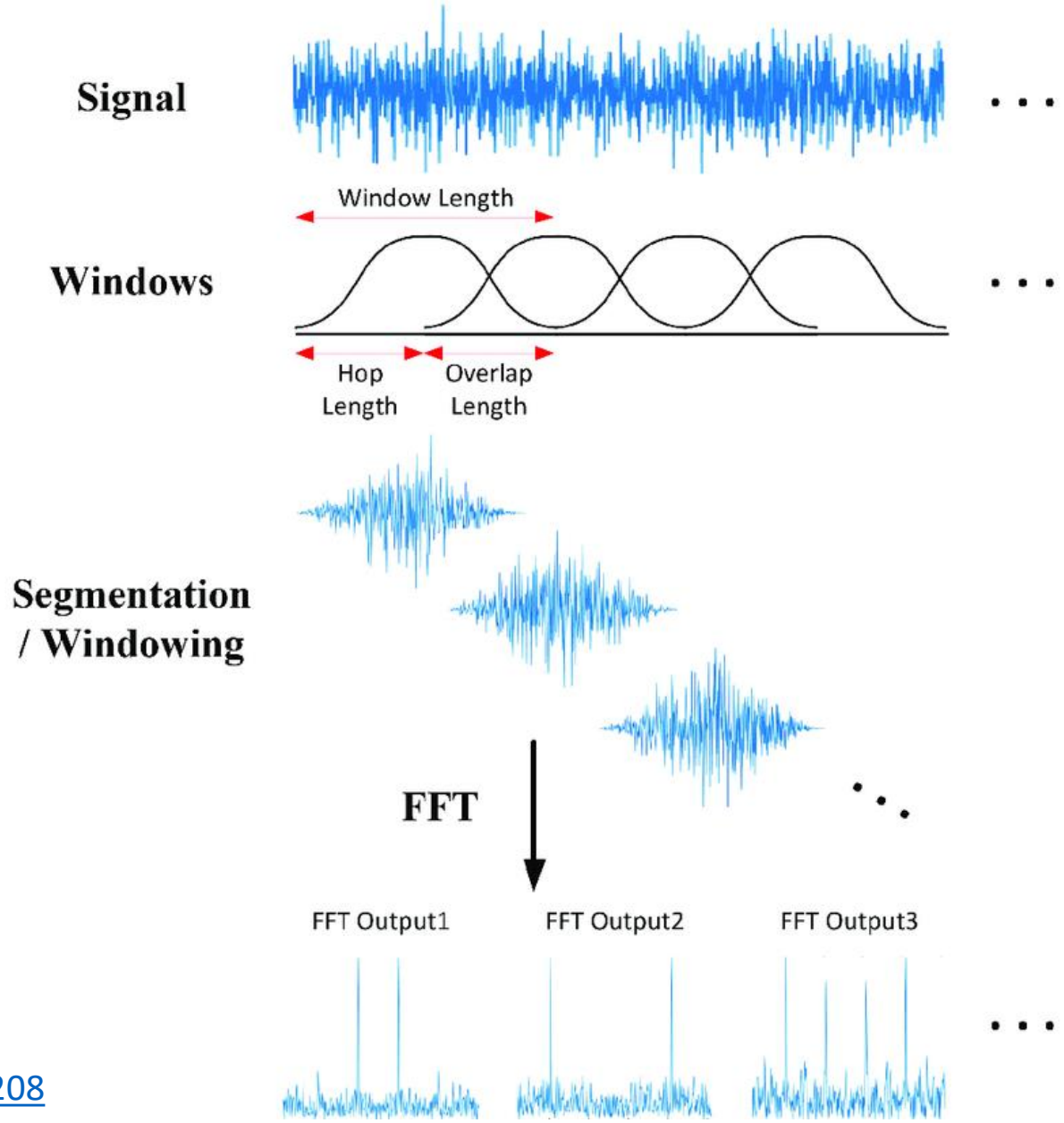
The width of the window is varied from 256 samples to 16 samples leading to improved temporal localization (vertical axis) but worsened frequency localization (horizontal axis).



Nasser Kehtarnavaz, in Digital Signal Processing System Design (Second Edition), 2008

Lecture notes: Limitations of the Fourier Transform: STFT, University of Wisconsin-Madison

A wide window gives better frequency resolution but poor time resolution. A narrower window gives good time resolution but poor frequency resolution. These are called narrowband and wideband transforms, respectively.



Appl. Sci. 2020, 10(20), 7208; https://doi.org/10.3390/app10207208

Short Time Fourier Transform (STFT)

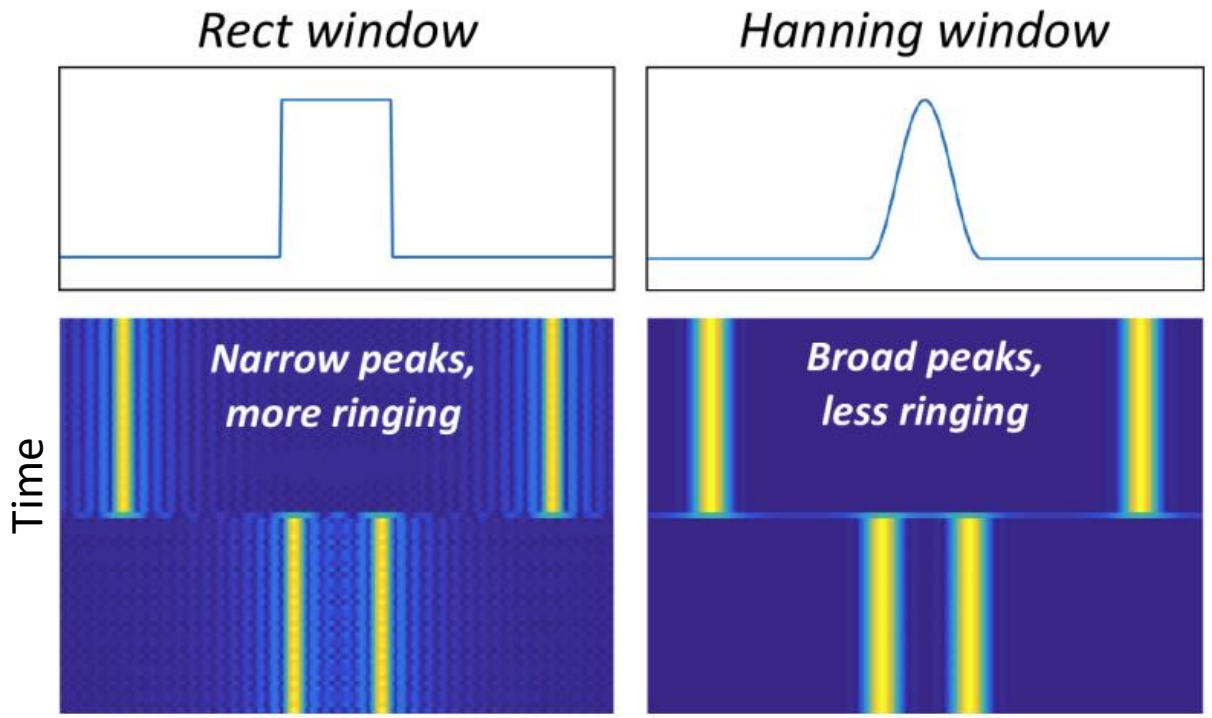


Figure 16.2: Besides the total width of the STFT window, the specific window shape determines the output of the STFT. This figure compares two choices of STFT windows with the same overall width, applied to the signal shown in Figure 16.1 above. A flat ('rect') window results in finer frequency resolution but more frequency ringing, whereas a more tapered ('hanning' in this case) window results in some loss of frequency resolution, but reduced ringing.

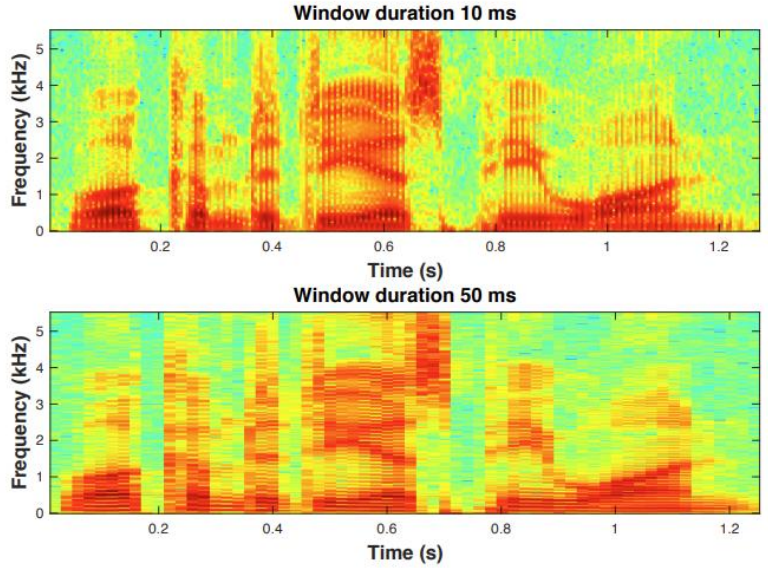


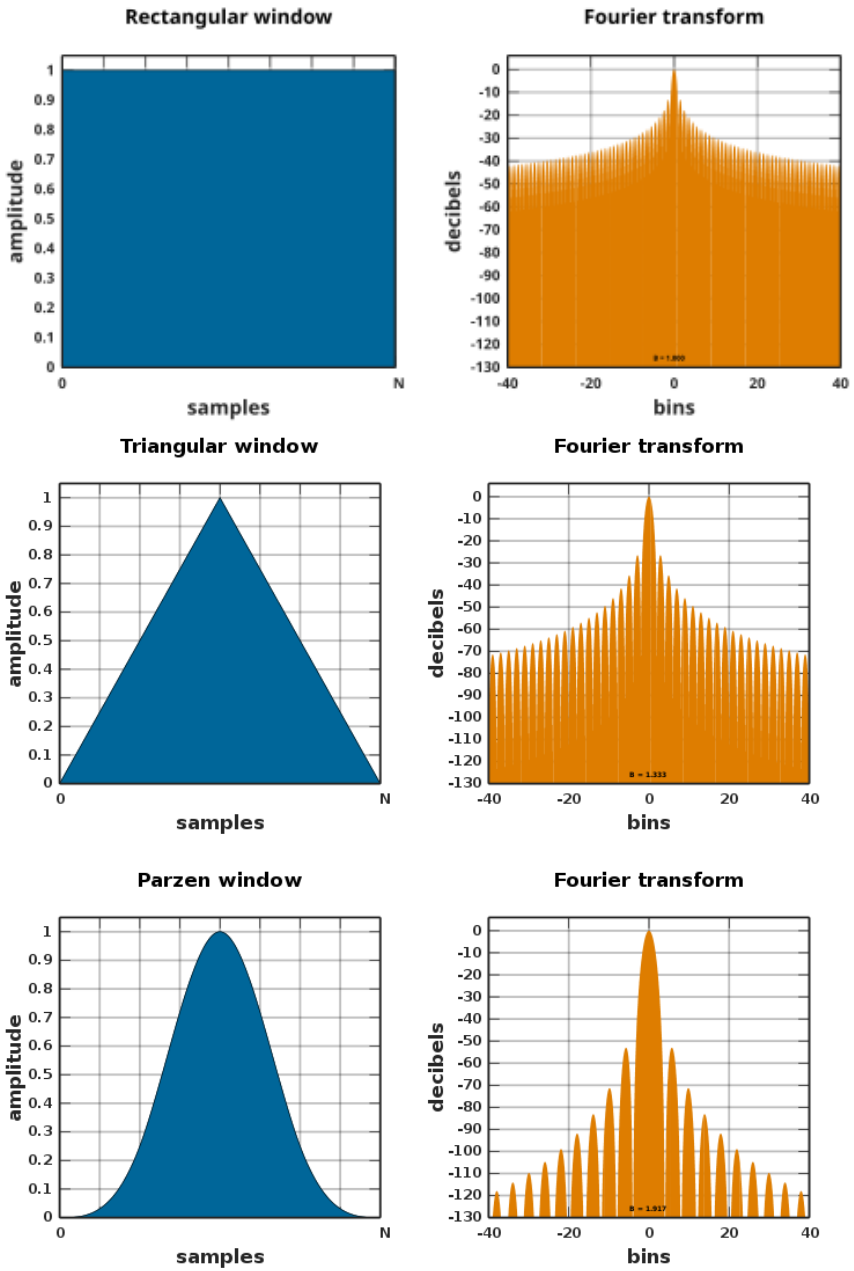
Figure 3.1: Impact of window duration on the STFT. A spectrogram of a brief utterance is shown, using Hamming windows of duration 10 ms (upper panel) and 50 ms (lower panel).

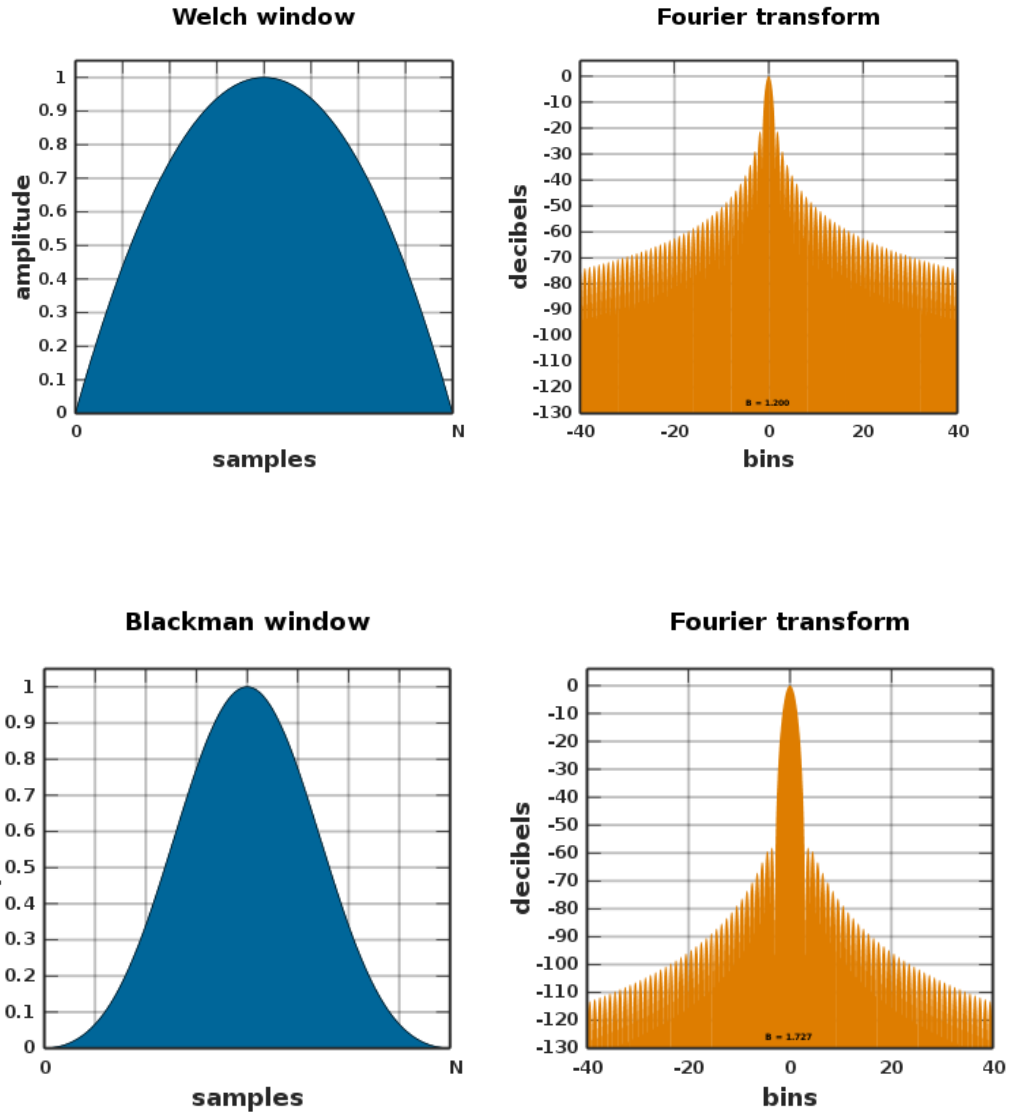
Spectrogram of a brief utterance ("Welcome to DSP-I") spoken by the author. The horizontal axis represents time while the vertical axis represents frequency.

The window functions used in the figure are Hamming windows of duration 10 ms (upper panel) and 50 ms (lower panel). The windows are overlapped by 50% in this case.

Lecture notes: Limitations of the Fourier Transform: STFT, University of Wisconsin-Madison

https://en.wikipedia.org/wiki/Window_function



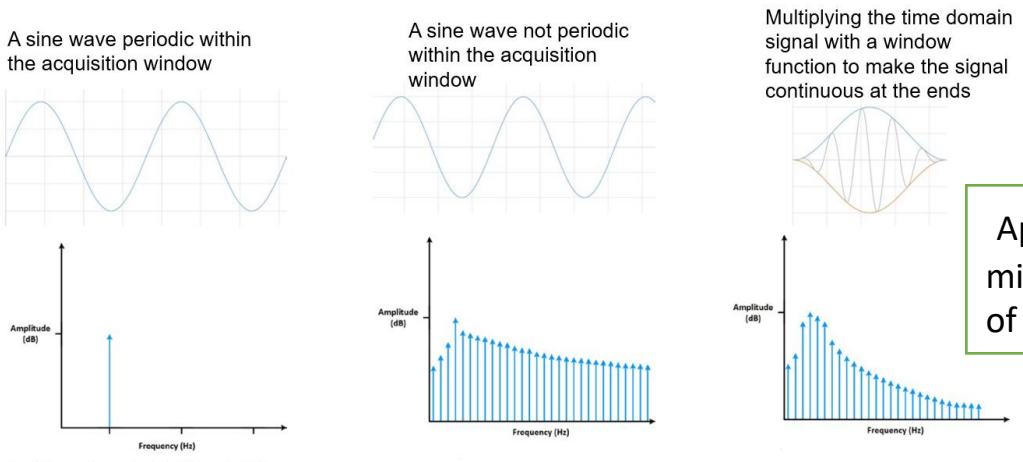


and many others…

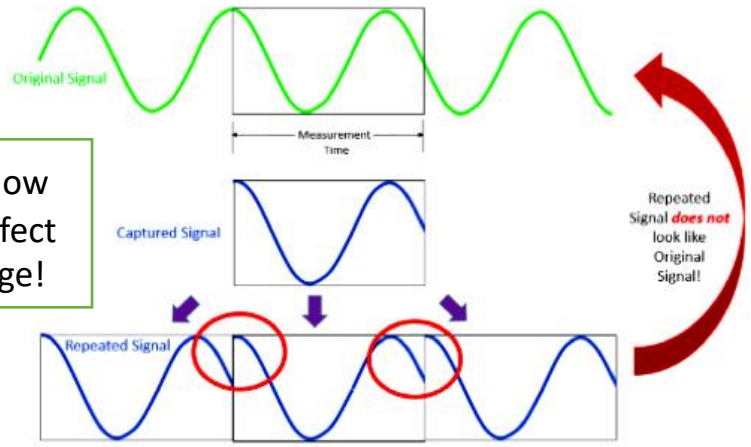
The Fourier transform of a function of time, s(t), is a complex-valued function of frequency, S(ω), often referred to as a frequency spectrum. Any linear time-invariant operation on s(t) produces a new spectrum of the form H(ω) x S(ω), which changes the relative magnitudes and/or angles (phase) of the non-zero values of S(ω). Any other type of operation creates new frequency components that may be referred to as **spectral leakage** in the broadest sense. But the term "leakage" usually refers to the effect of *windowing*, which is the product of s(t) with a different kind of function, the **window function**. Window functions happen to have finite duration, but that is not necessary to create leakage. Multiplication by a time-variant function is sufficient.

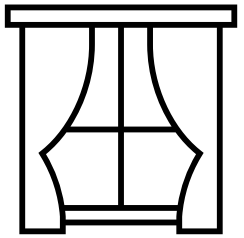
If the component frequencies are dissimilar and one component is weaker, then leakage from the stronger component can obscure the weaker one's presence. Windows that are effective against this type of interference, namely where components have dissimilar frequencies and amplitudes, are called **high dynamic range**.

If the frequencies are too similar, leakage can render them unresolvable even if their amplitudes are of similar strengths. Windows that can distinguish components with similar frequencies and amplitudes are called **high resolution**.



Applying a window minimizes the effect of spectral leakage!



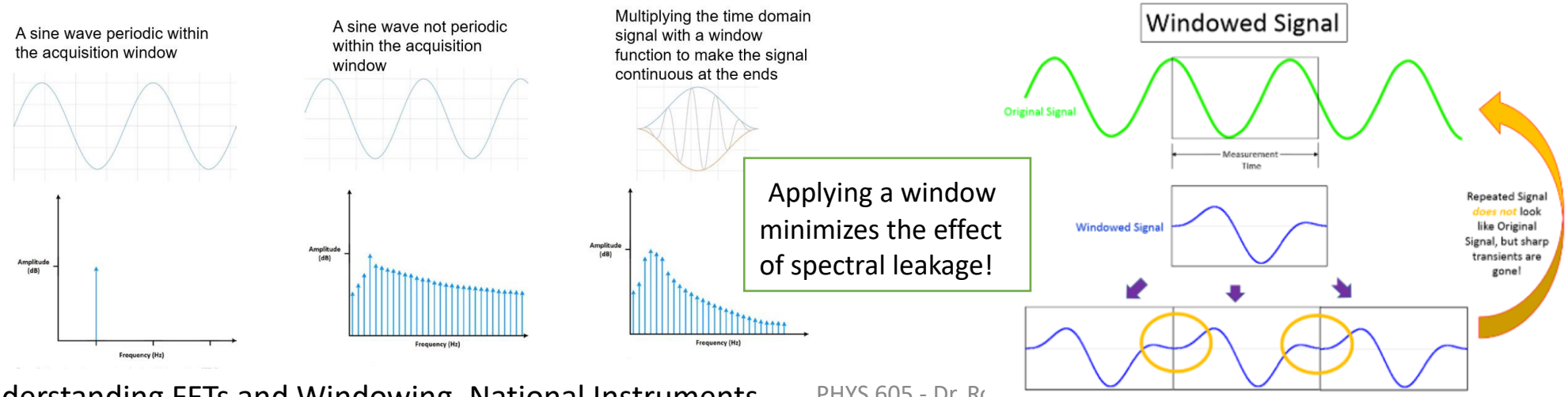


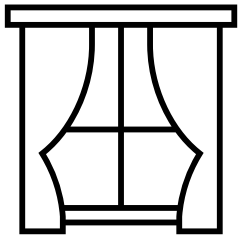
Understanding FFTs and Windowing, National Instruments

The Fourier transform of a function of time, s(t), is a complex-valued function of frequency, S(ω), often referred to as a frequency spectrum. Any linear time-invariant operation on s(t) produces a new spectrum of the form H(ω) x S(ω), which changes the relative magnitudes and/or angles (phase) of the non-zero values of S(ω). Any other type of operation creates new frequency components that may be referred to as **spectral leakage** in the broadest sense. But the term "leakage" usually refers to the effect of *windowing*, which is the product of s(t) with a different kind of function, the **window function**. Window functions happen to have finite duration, but that is not necessary to create leakage. Multiplication by a time-variant function is sufficient.

If the component frequencies are dissimilar and one component is weaker, then leakage from the stronger component can obscure the weaker one's presence. Windows that are effective against this type of interference, namely where components have dissimilar frequencies and amplitudes, are called **high dynamic range**.

If the frequencies are too similar, leakage can render them unresolvable even if their amplitudes are of similar strengths. Windows that can distinguish components with similar frequencies and amplitudes are called **high resolution**.





Understanding FFTs and Windowing, National Instruments

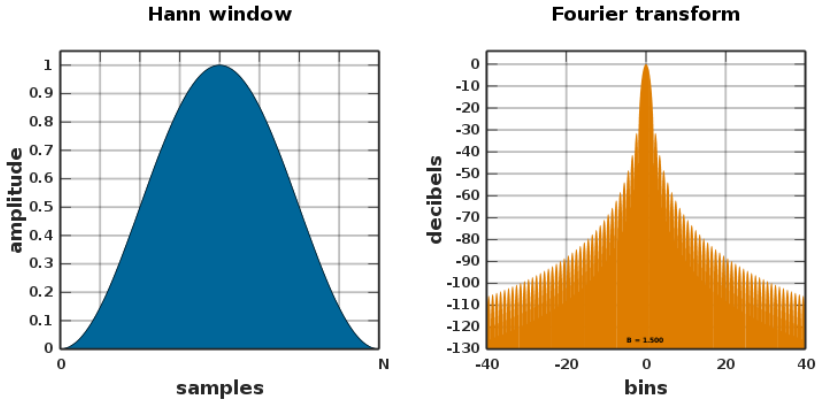
Example: the **rectangular window is an example of a window that is high resolution but low dynamic range**, meaning it is good for distinguishing components of similar amplitude even when the frequencies are also close, but poor at distinguishing components of different amplitude even when the frequencies are far away. High-resolution, low-dynamic-range windows such as the rectangular window also have the property of **high sensitivity**, which is the ability to reveal relatively weak sinusoids in the presence of additive random noise.

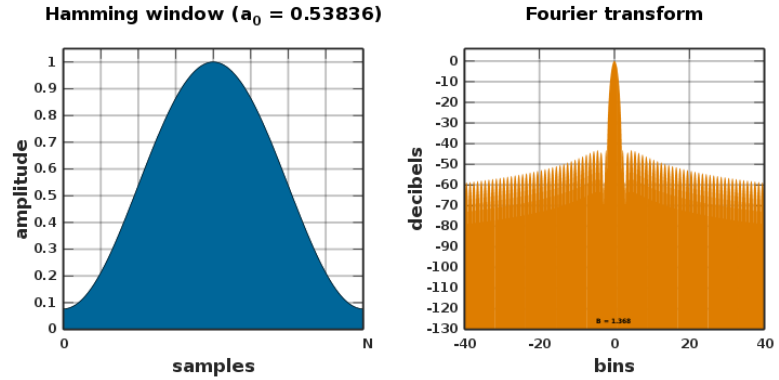
At the other extreme of the range of window types are windows with **high dynamic range but low resolution and sensitivity**. High-dynamic-range windows are most often justified in *wideband applications*, where the spectrum being analyzed is expected to contain many different components of various amplitudes.

In between the extremes are moderate windows, such as *Hann and Hamming*. They are commonly used in narrowband applications, such as the spectrum of a telephone channel.

https://en.wikipedia.org/wiki/Spectral_leakage

In summary, *spectral analysis* involves a trade-off between resolving comparable strength components with similar frequencies (high resolution/sensitivity) and resolving disparate strength components with dissimilar frequencies (high dynamic range). That trade-off occurs when the window function is chosen.





https://vru.vibrationresearch.com/lesson/window-function-characteristics/

https://vru.vibrationresearch.com/lesson/tables-of-window-function-details/

The features of window functions to look at:

- **Main lobe**

The main lobe is centered at each frequency component of the time-domain signal whereas the side lobes (see below) approach zero. This affects the ability to distinguish between two frequency components that are close together. A window function with a small main lobe width indicates high-frequency resolution.

- **Side lobe roll-off rate**

The asymptotic decay rate in decibels per decade of the frequency of the side lobes' peaks. By increasing the side lobe roll-off rate, one can reduce spectral leakage.

- **Side lobe level**

The height of the side lobes indicates the affect the windowing function has on frequencies around main lobes. An ideal window function has a high side lobe level, which indicates good noise suppression and high detection ability.

The best window function for a given application depends on the characteristics of the signal and the objectives of the analysis. For example, if the signal contains strong interfering frequency components that are far from the frequency of interest, a smoothing window with a high side lobe roll-off rate is best.

https://www.ni.com/en/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/understanding-ffts-and-windowing.html

- If the signal contains strong interfering frequency components distant from the frequency of interest, choose a smoothing window with a high side lobe roll-off rate.
- If the signal contains strong interfering signals near the frequency of interest, choose a window function with a low maximum side lobe level.
- If the frequency of interest contains two or more signals very near to each other, spectral resolution is important. In this case, it is best to choose a smoothing window with a very narrow main lobe.
- If the amplitude accuracy of a single frequency component is more important than the exact location of the component in a given frequency bin, choose a window with a wide main lobe.
- If the signal spectrum is rather flat or broadband in frequency content, use the uniform window, or no window.

In general, the Hanning (Hann) window is satisfactory in 95% of cases. It has good frequency resolution and reduced spectral leakage. If you do not know the nature of the signal but you want to apply a smoothing window, start with the Hann window.

Search the docs ...

scipy
scipy.cluster
scipy.constants
scipy.datasets
scipy.fft
scipy.fftpack
scipy.integrate
scipy.interpolate
scipy.io
scipy.linalg
scipy.misc
scipy.ndimage
scipy.odr
scipy.optimize
scipy.signal
scipy.sparse
scipy.spatial
scipy.special
scipy.stats

scipy.signal.stft

scipy.signal.stft(*x, fs=1.0, window='hann', nperseg=256, noverlap=None, nfft=None, detrend=False, return_onesided=True, boundary='zeros', padded=True, axis=-1, scaling='spectrum'*) [source]

Compute the Short Time Fourier Transform (STFT).

STFTs can be used as a way of quantifying the change of a nonstationary signal's frequency and phase content over time.

Parameters: **x** : *array_like*

Time series of measurement values

fs : *float, optional*

Sampling frequency of the *x* time series. Defaults to 1.0.

window : *str or tuple or array_like, optional*

Desired window to use. If *window* is a string or tuple, it is passed to `get_window` to generate the window values, which are DFT-even by default. See `get_window` for a list of windows and required parameters. If *window* is array_like it will be used directly as the window and its length must be nperseg. Defaults to a Hann window.

nperseg : *int, optional*

Length of each segment. Defaults to 256.

noverlap : *int, optional*

Number of points to overlap between segments. If *None*, `noverlap = nperseg // 2`. Defaults to *None*. When specified, the COLA constraint must be met (see Notes below).

nfft : *int, optional*

Length of the FFT used, if a zero padded FFT is desired. If *None*, the FFT length is *nperseg*. Defaults to *None*.

detrend : *str or function or False, optional*

Specifies how to detrend each segment. If `detrend` is a string, it is passed as the *type* argument to the `detrend` function. If it is a function, it takes a segment and returns a detrended segment. If `detrend` is *False*, no detrending is done. Defaults to *False*.

On this page

stft

Window types:

- `boxcar`
- `triang`
- `blackman`
- `hamming`
- `hann`
- `bartlett`
- `flattop`
- `parzen`
- `bohman`
- `blackmanharris`
- `nuttall`
- `barthann`
- `cosine`
- `exponential`
- `tukey`
- `taylor`
- `lanczos`
- `kaiser` (needs beta)
- `kaiser_bessel_derived` (needs beta)
- `gaussian` (needs standard deviation)
- `general_cosine` (needs weighting coefficients)
- `general_gaussian` (needs power, width)
- `general_hamming` (needs window coefficient)
- `dpss` (needs normalized half-bandwidth)
- `chebwin` (needs attenuation)

Complete Jupyter Notebooks to follow!

Guide to GW (gravitational waves) detections and noise using the LIGO data!

https://gwosc.org/tutorials/

https://gwosc.org/s/events/GW150914/LOSC_Event_tutorial_GW150914.html

https://gwosc.org/s/events/GW150914/GW150914_tutorial.html (no longer maintained)