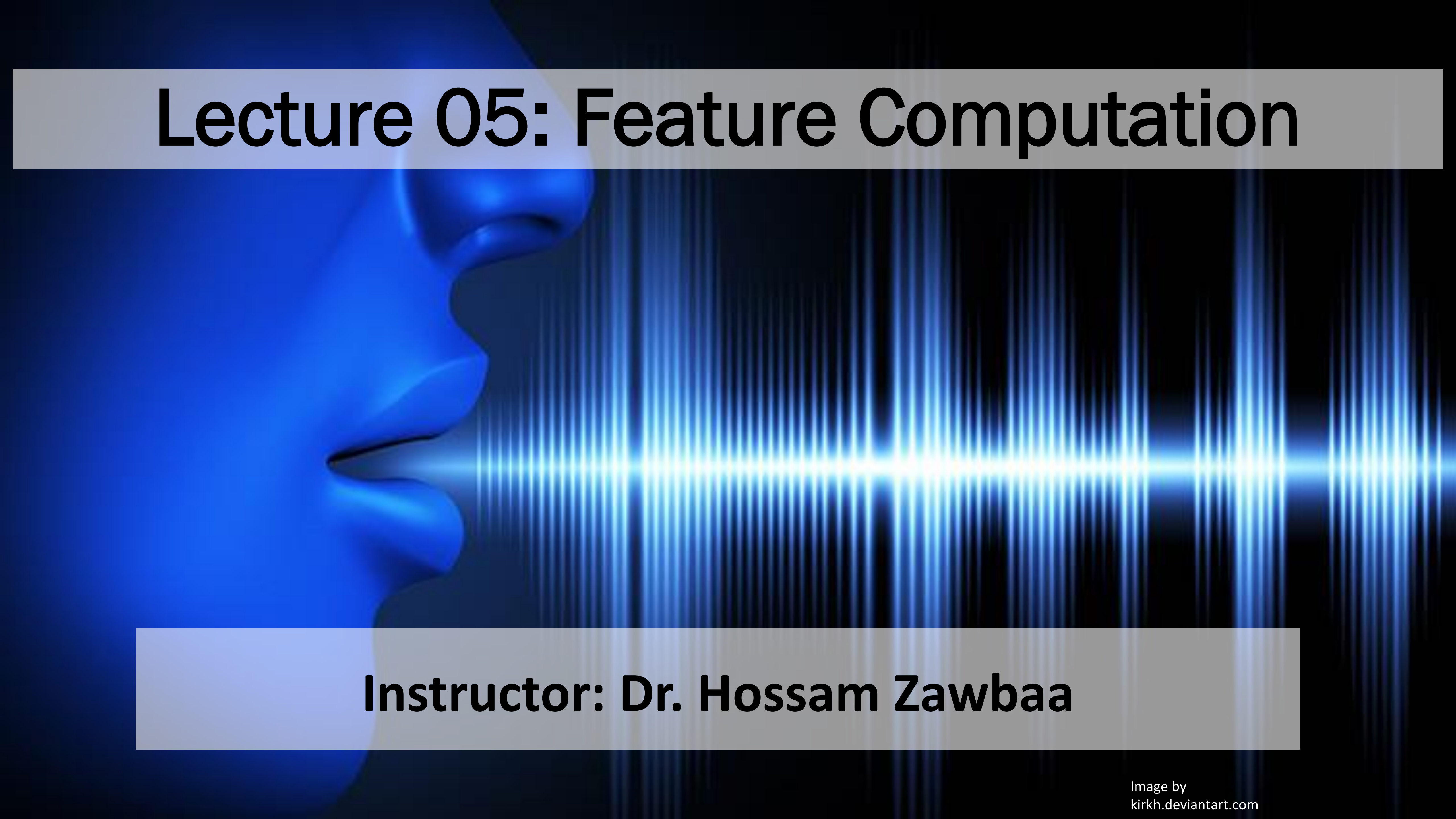
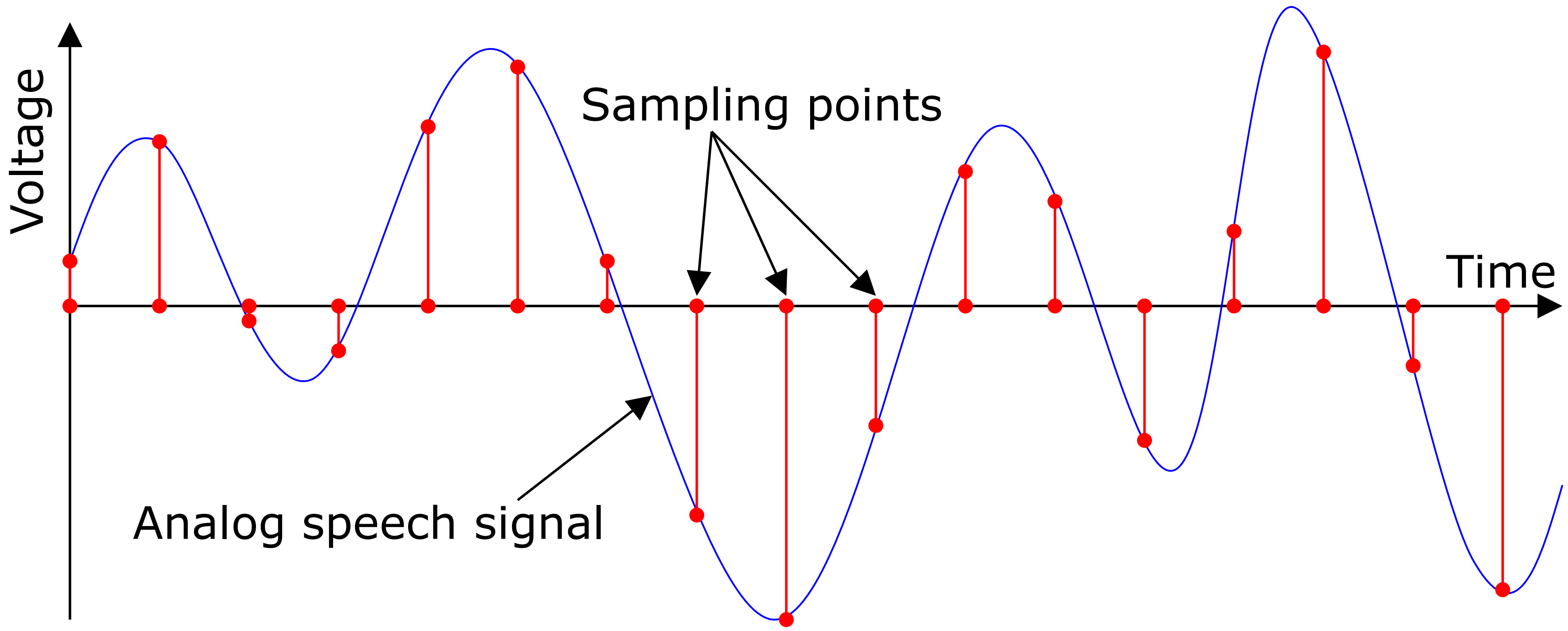
Lecture 05: Feature Computation



Instructor: Dr. Hossam Zawbaa

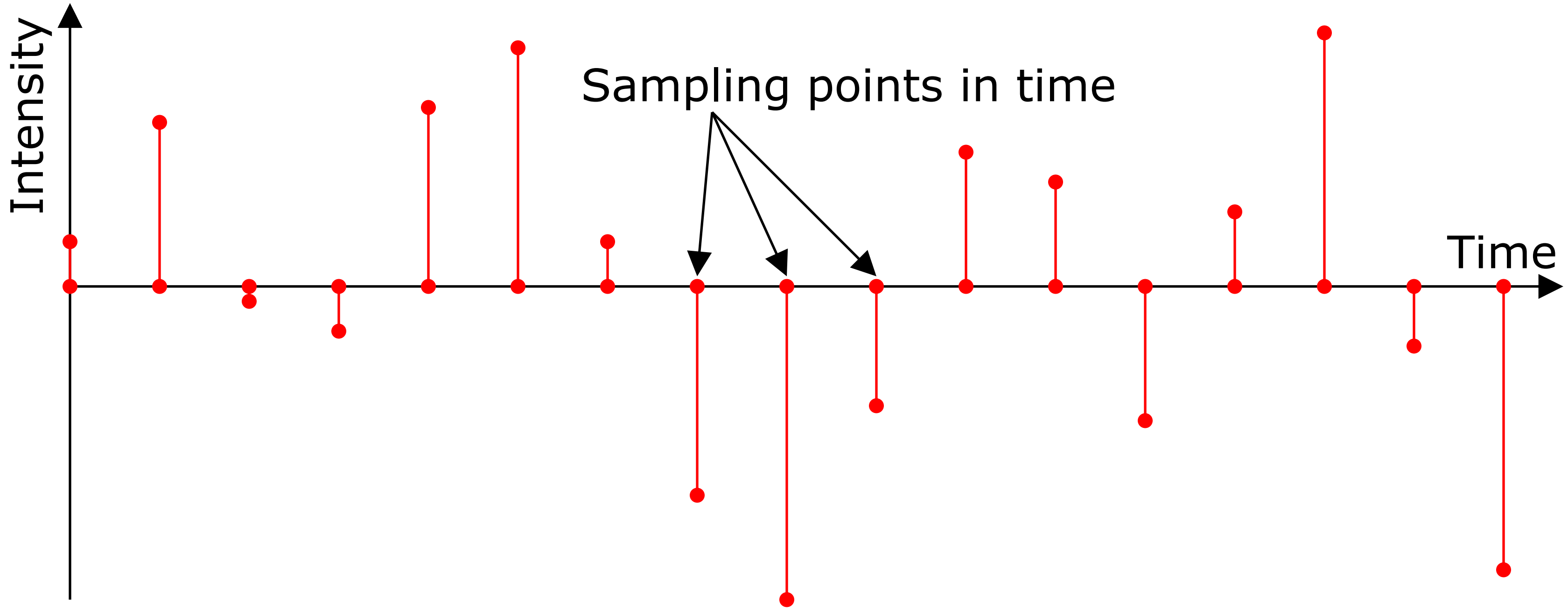
The Speech Signal: Sampling

- The analog speech signal captures pressure variations in air that are produced by the speaker
  - The same function as the ear
- The analog speech input signal from the microphone is *sampled* periodically at some fixed *sampling rate*



The Speech Signal: Sampling

- What remains after sampling is the value of the analog signal at *discrete time points*
- **This is the discrete-time signal**



The Speech Signal: Sampling

- The analog speech signal has many *frequencies*
  - The human ear can perceive frequencies in the range 50Hz-15kHz (more if you're young)
- The information about what was spoken is carried in all these frequencies
- But most of it is in the 150Hz-5kHz range

The Speech Signal: Sampling

- A signal that is digitized at *N* samples/sec can represent frequencies up to *N/*2 Hz only.
  - The Nyquist theorem

- A signal that is sampled at *N* samples per second must first be low-pass filtered at *N*/2 Hz to avoid distortions.

- Ideally, one would sample the speech signal at a sufficiently high rate to retain all perceivable components in the signal.
  - > 30kHz

- For practical reasons, lower sampling rates are often used, however
  - Save bandwidth / storage
  - Speed up computation

The Speech Signal: Sampling

- Audio hardware typically supports several standard rates
  - *E.g.*: 8, 16, 11.025, or 44.1 KHz (*n* Hz = *n* samples/sec)
  - **CD recording employs 44.1 KHz per channel – high enough to represent most signals accurate.**

- **Speech recognition typically uses 8KHz sampling rate for telephone speech and 16KHz for wideband speech**
  - Telephone data is *narrowband* and has frequencies only up to 4 KHz
  - Good microphones provide a *wideband* speech signal
    - 16KHz sampling can represent audio frequencies up to 8 KHz
    - This is considered sufficient for speech recognition

The Speech Signal: Digitization

- **Each sampled value is *digitized* (or *quantized* or *encoded*) into one of a set of fixed discrete levels**
  - Each analog voltage value is *mapped* to the nearest discrete level
  - Since there are a fixed number of discrete levels, the mapped values can be represented by a number; *e.g.* 8-bit, 12-bit or 16-bit
- **Digitization can be *linear* (uniform) or *non-linear* (non-uniform)**

The Speech Signal: Linear Coding

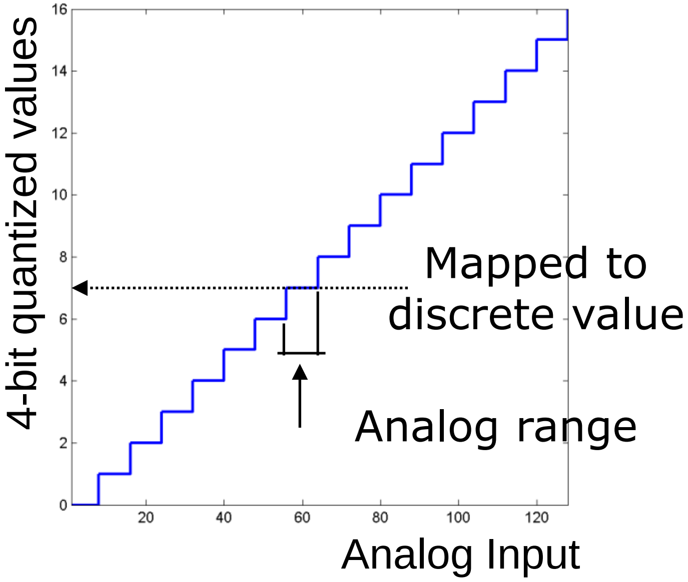
- Linear coding (also known as *pulse-code modulation* or PCM) **splits the input analog range into some number of uniformly spaced levels.**

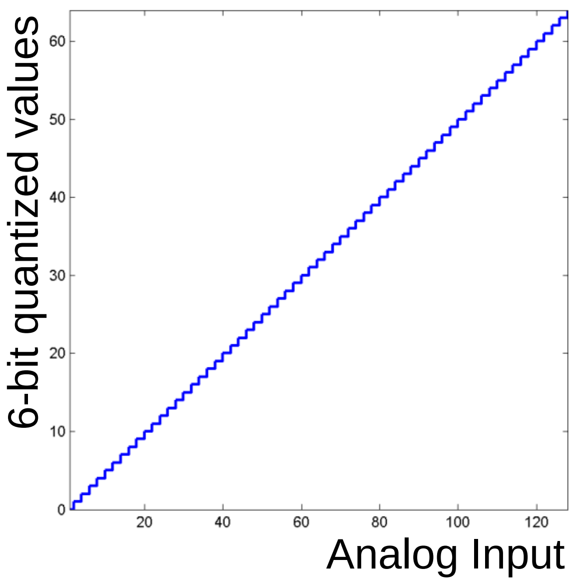
- **The no. of discrete levels determines no. of bits needed to represent a quantized signal value**; *e.g.*:
  - 4096 levels require 12-bit representation
  - 65536 levels require 16-bit representation

- In speech recognition, PCM data is typically represented using 16 bits

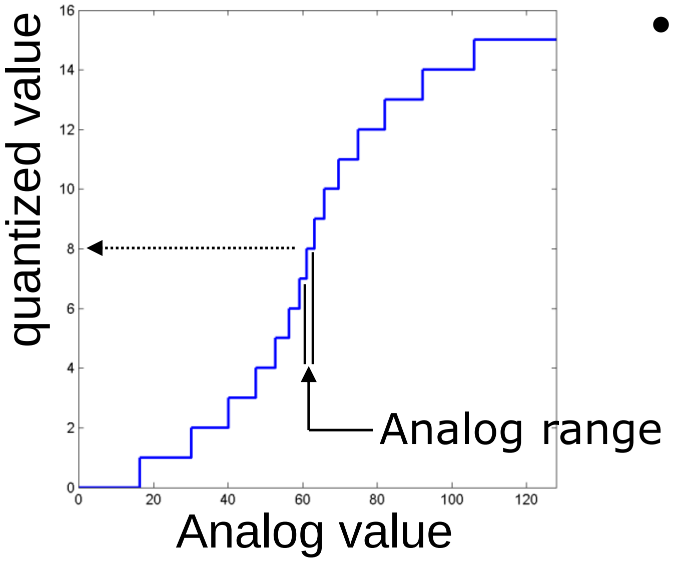
The Speech Signal: Linear Coding

- Example PCM quantizations into 16 and 64 levels:
- Since an entire analog range is mapped to a single value, quantization leads to *quantization error*
  - Average error can be reduced by increasing the number of discrete levels

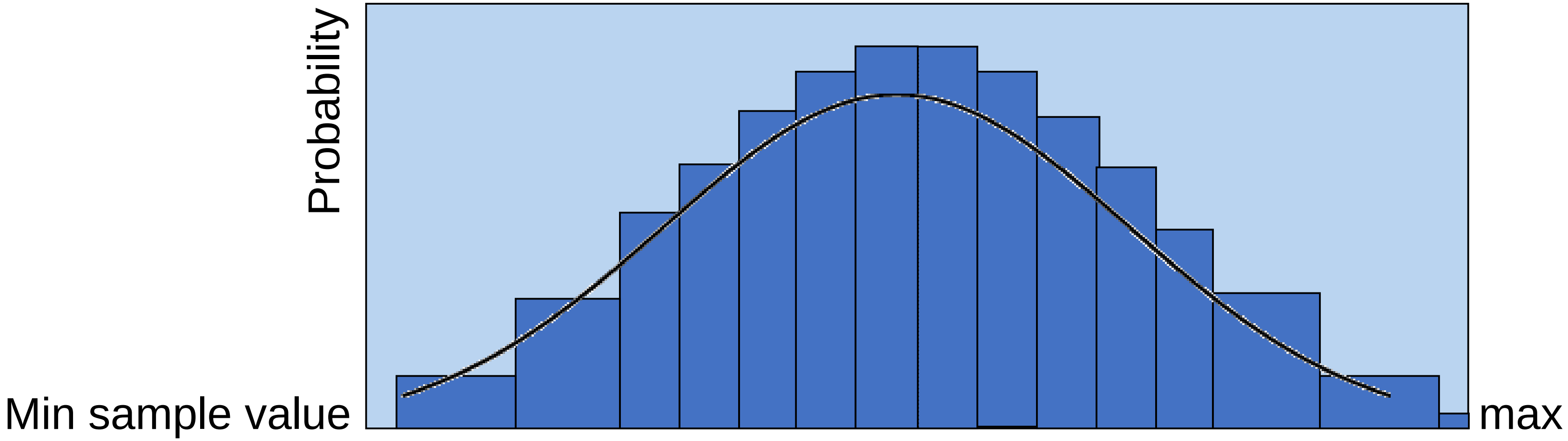




The Speech Signal: Non-Linear Coding



- Converts non-uniform segments of the analog axis to uniform segments of the quantized axis
  - Spacing between adjacent segments on the analog axis is chosen based on the relative frequencies of sample values in that region
  - Sample regions of high frequency are more finely quantized



The Speech Signal: Non-Linear Coding

- Thus, fewer discrete levels can be used, without significantly worsening *average* quantization error
  - High resolution coding around the most probable analog levels
    - Thus, most frequently encountered analog levels have lower quantization error
  - Lower resolution coding around low probability analog levels
    - Encodings with higher quantization error occur less frequently

- *A-law* and *μ-law* encoding schemes use only 256 levels (8-bit encodings)
  - Widely used in telephony
  - Can be converted to linear PCM values via standard tables

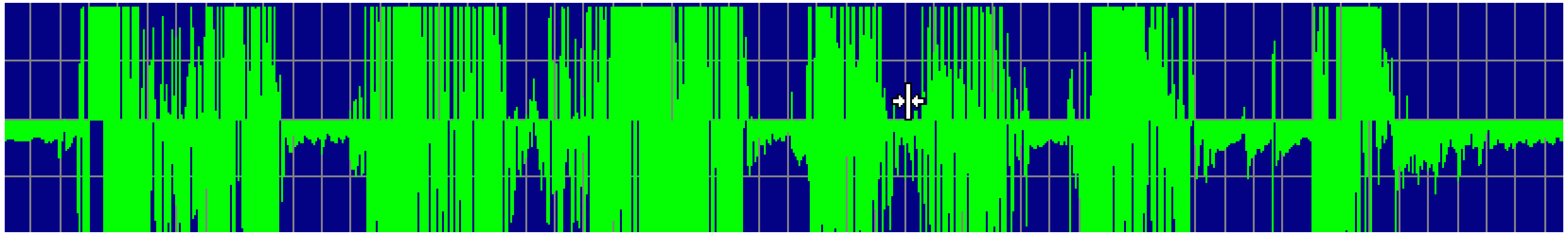
Effect of Signal Quality

- The quality of the final digitized signal depends critically on all the other components:
  - The microphone quality
  - Environmental quality – the microphone picks up not just the subject's speech, but all other ambient noise
  - The electronics performing sampling and digitization
    - Poor quality electronics can severely degrade signal quality
      - *E.g.* Disk or memory bus activity can inject noise into the analog circuitry
  - Proper setting of the recording level
    - Too low a level underutilizes the available signal range, increasing susceptibility to noise
    - Too high a level can cause *clipping*

- Suboptimal signal quality can affect recognition accuracy to the point of being completely useless

Digression: Clipping in Speech Signals

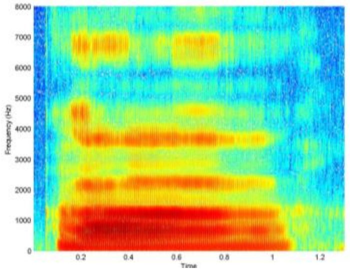
- Clipping is a kind of signal distortion.
- The amplitude of a clipped signal is limited by some threshold(s).
- On oscillograms, clipping usually appears as a cutoff of signal amplitude.
- Clipping can be single-sided (only the top or only the bottom of the signal is cut) and double-sided.
- Clipping and non-linear distortion are the most common and most easily fixed problems in audio recording
  - Simply reduce the signal gain



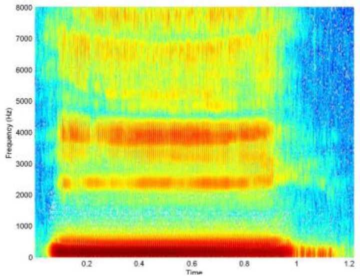
Sound Characteristics are in Frequency Patterns

- Figures below show energy at various frequencies in a signal as a function of time
  - Called a spectrogram

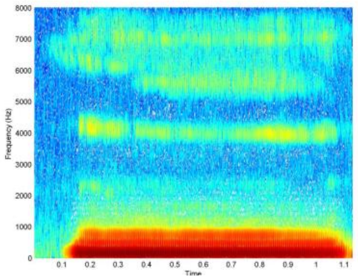
AA



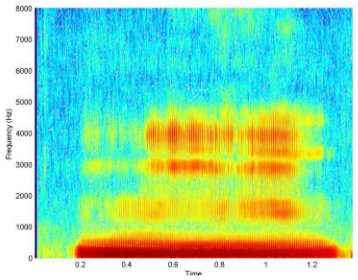
IY



UW



M



- Different instances of a sound will have the same generic spectral structure
- **Features must capture this spectral structure**

Computing "Features"

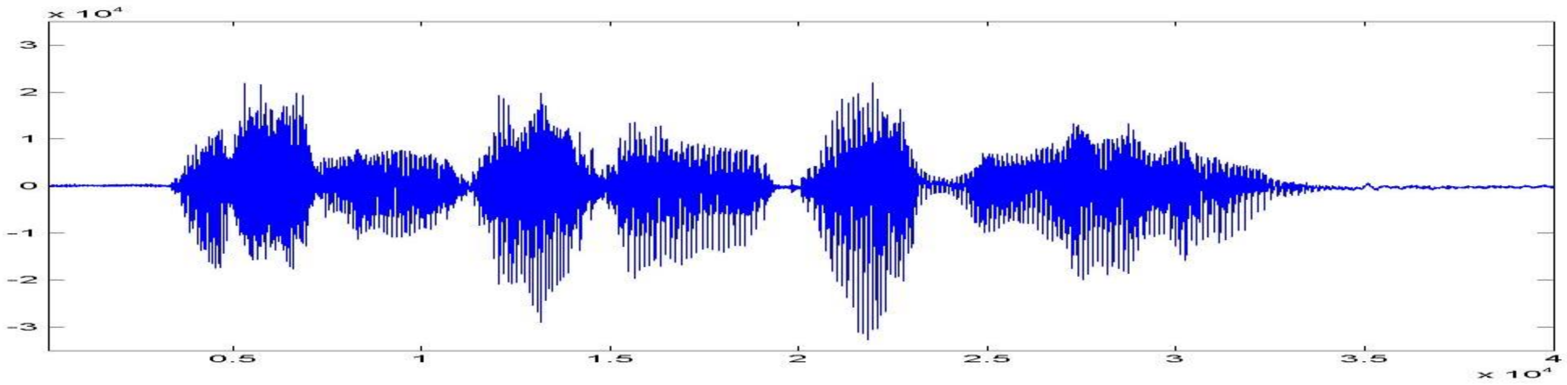
- **Features must be computed that capture the *spectral* characteristics of the signal**
- **Important to capture only the *salient* spectral characteristics of the sounds**
  - Without capturing speaker-specific or other incidental structure
- The most commonly used feature is the ***Mel-frequency cepstrum***
  - Compute the spectrogram of the signal
  - Derive a set of numbers that capture only the salient aspects of this spectrogram
  - Salient aspects computed according to the manner in which humans perceive sounds
- A **cepstrum** is the result of taking the inverse Fourier transform of the logarithm of the estimated spectrum of a signal.

Capturing the Spectrum: The discrete Fourier transform

- Transform analysis: **Decompose a sequence of numbers into a weighted sum of other time series**

- The component time series must be defined
  - For the Fourier Transform, these are complex exponentials

- The analysis determines the weights of the component time series



The complex exponential

- The complex exponential is a complex sum of two sinusoids

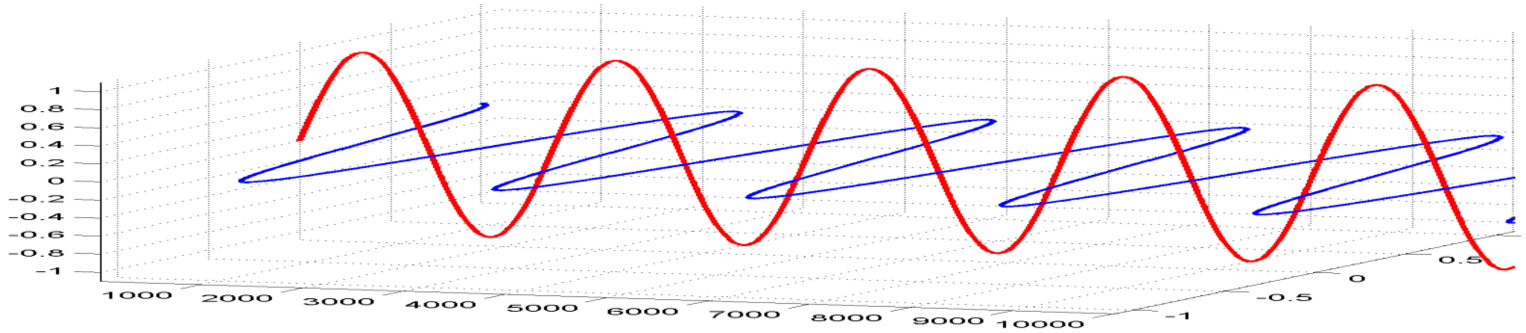
$$e^{j\theta} = \cos\theta + j\sin\theta$$

- The real part is a cosine function
- The imaginary part is a sine function
- A complex exponential time series is a complex sum of two time series

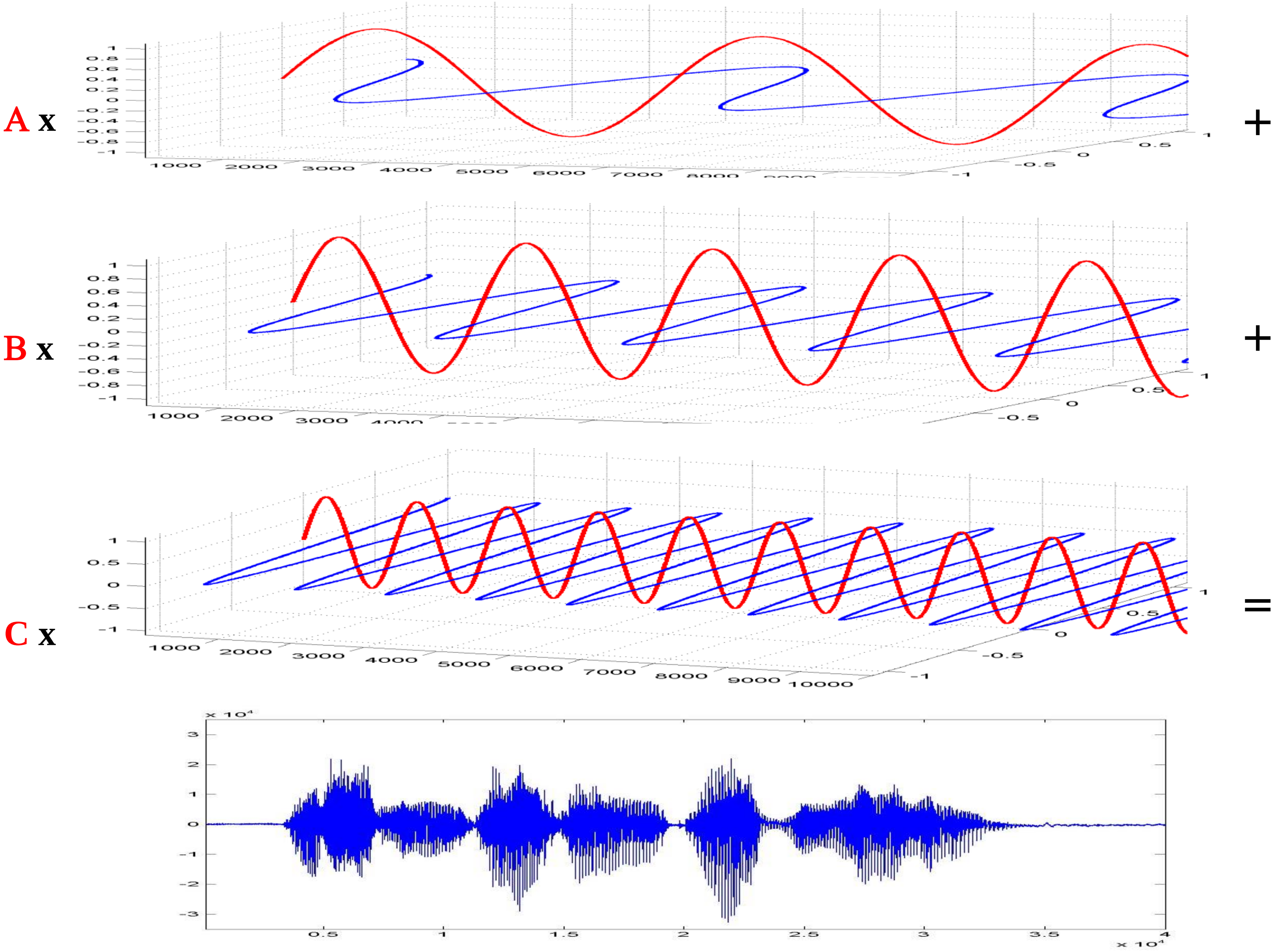
$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$$

- Two complex exponentials of different frequencies are "orthogonal" to each other. i.e.

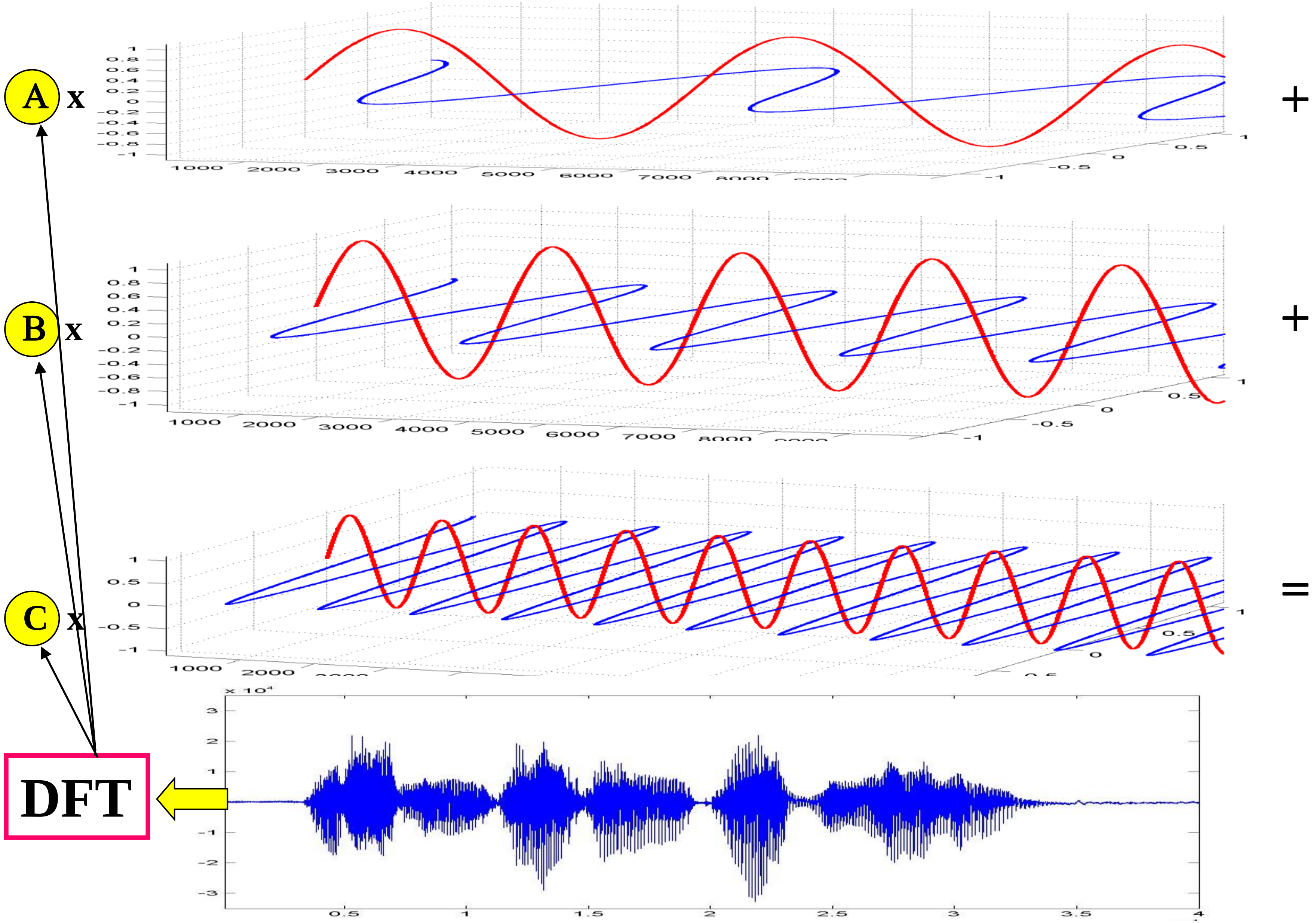
$$\int_{-\infty}^{\infty} e^{j\alpha t} e^{j\beta t} dt = 0 \qquad \text{if } \alpha \neq \beta$$



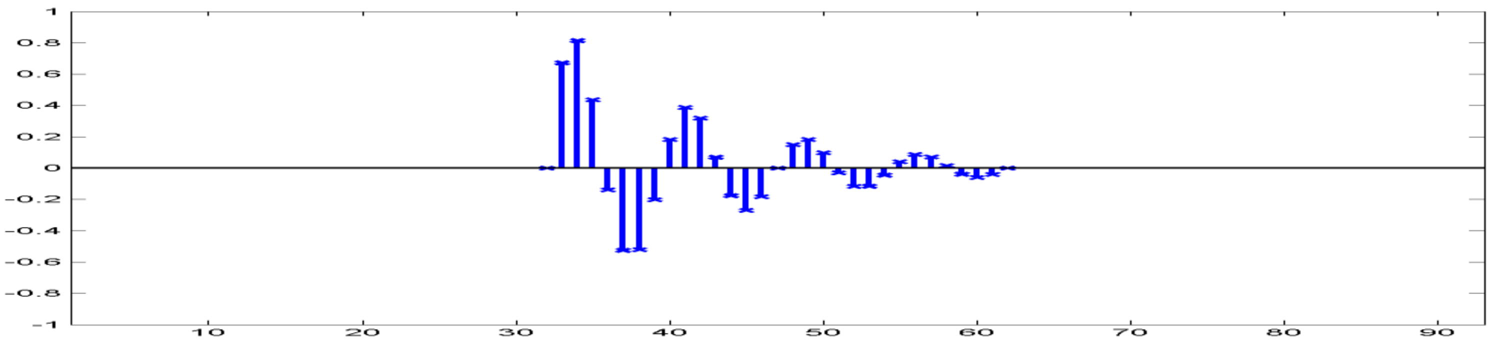
The discrete Fourier transform



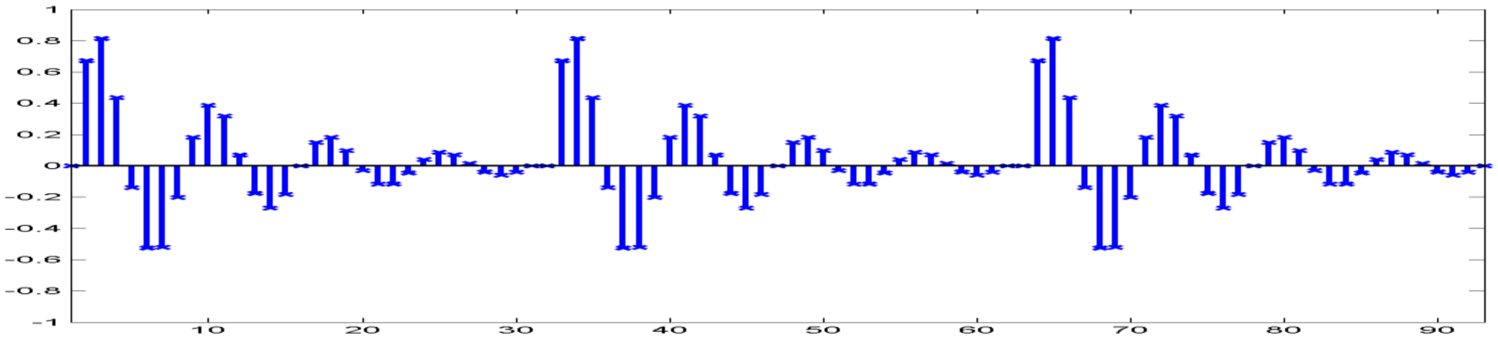
The discrete Fourier transform



The discrete Fourier transform



- The discrete Fourier transform of the above signal actually computes the Fourier spectrum of the periodic signal shown below
  - Which extends from –infinity to +infinity
  - The period of this signal is 31 samples in this example



The discrete Fourier transform

- Discrete Fourier transform coefficients are generally complex
  - $e^{j\theta}$ has a real part $\cos\theta$ and an imaginary part $\sin\theta$

$$e^{j\theta} = \cos\theta + j\sin\theta$$

  - As a result, every X[k] has the form

$$X[k] = X_{real}[k] + jX_{imaginary}[k]$$

- A magnitude spectrum represents only the magnitude of the Fourier coefficients

$$X_{magnitude}[k] = \mathrm{sqrt}(X_{real}[k]^2 + X_{imag}[k]^2)$$

- A power spectrum is the square of the magnitude spectrum

$$X_{power}[k] = X_{real}[k]^2 + X_{imag}[k]^2$$

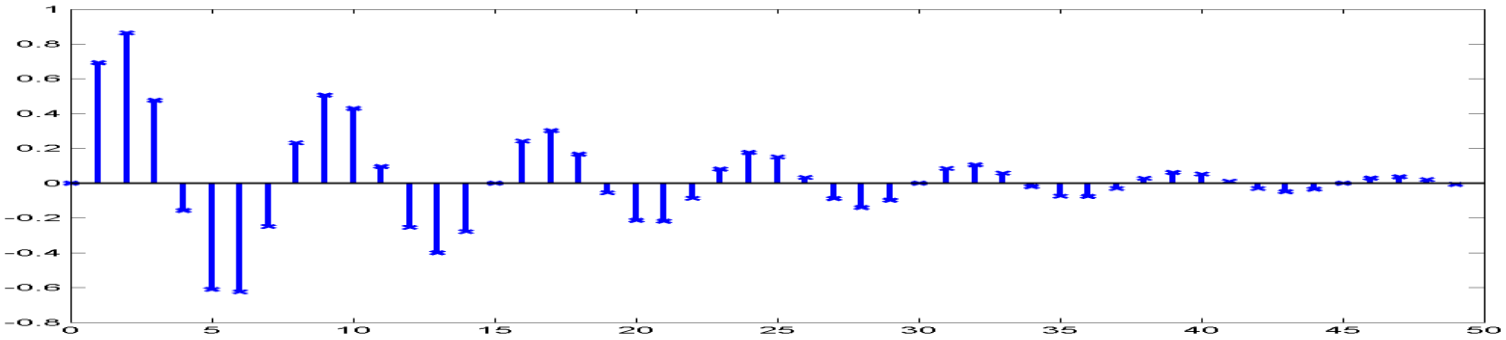
- **For speech recognition, we usually use the magnitude or power spectrum**

The discrete Fourier transform

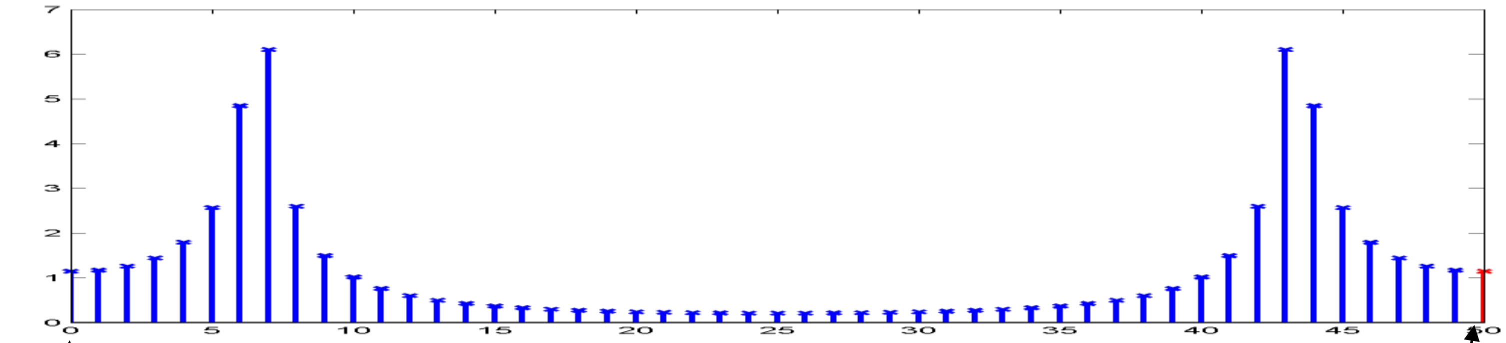
- **A discrete Fourier transform of an M-point sequence will only compute M unique frequency components**
  - i.e. the DFT of an M point sequence will have M points
  - The M-point DFT represents frequencies in the continuous-time signal that was digitized to obtain the digital signal
- The 0^{th} point in the DFT represents 0Hz, or the DC component of the signal
- The $(M-1)^{th}$ point in the DFT represents (M-1)/M * the sampling frequency
- **All DFT points are uniformly spaced on the frequency axis between 0 and the sampling frequency**

The discrete Fourier transform

- A 50 point segment of a decaying sine wave sampled at 8000 Hz



- The corresponding 50 point magnitude DFT. The 51st point (shown in red) is identical to the 1st point.

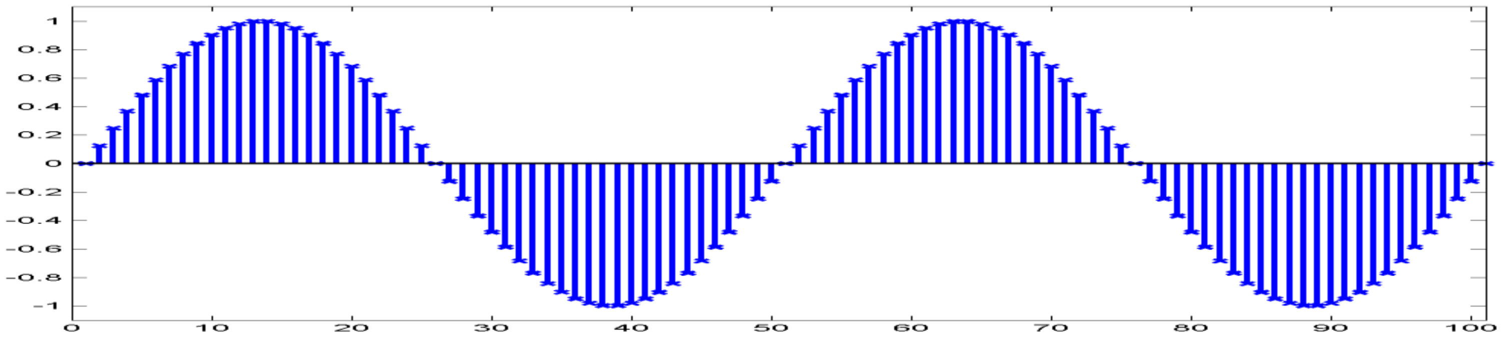


Sample 0 = 0 Hz

Sample 50 is the 51st point
It is identical to Sample 0

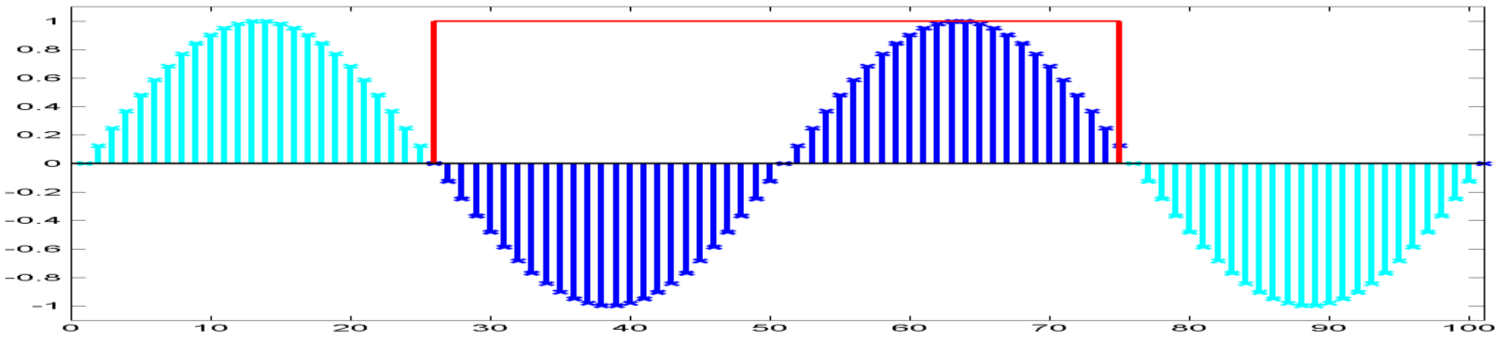
Sample 50 = 8000Hz

Windowing



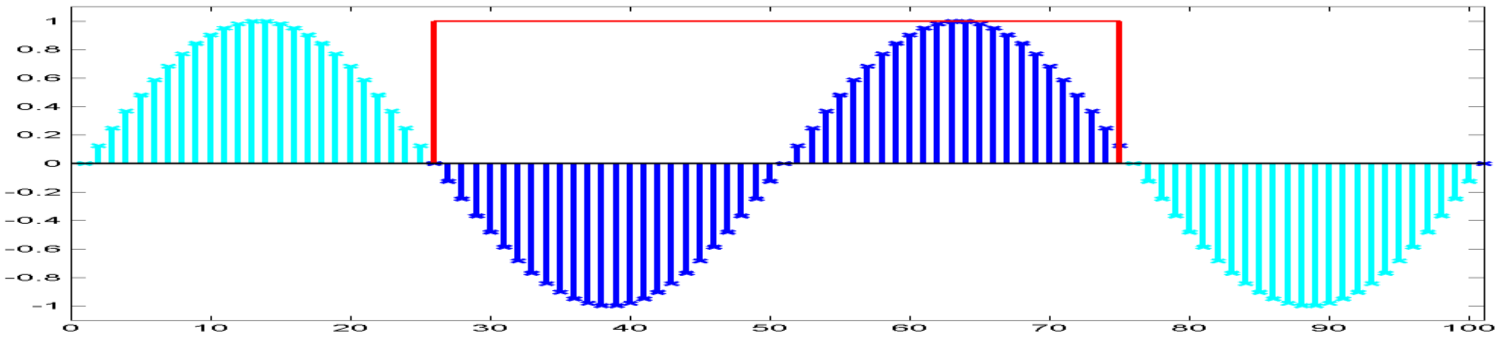
- The DFT of one period of the sinusoid shown in the figure computes the Fourier series of the entire sinusoid from –infinity to +infinity

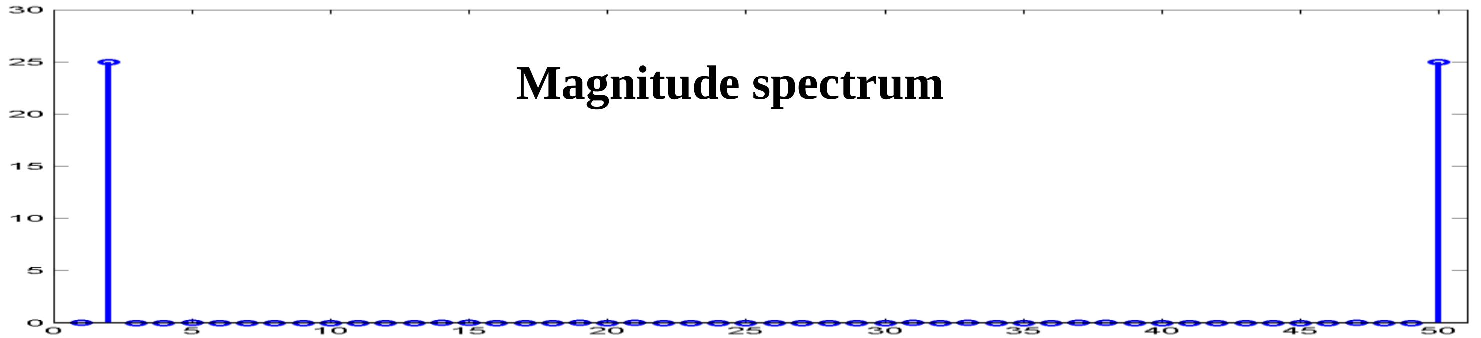
Windowing



- The DFT of one period of the sinusoid shown in the figure computes the Fourier series of the entire sinusoid from –infinity to +infinity

Windowing

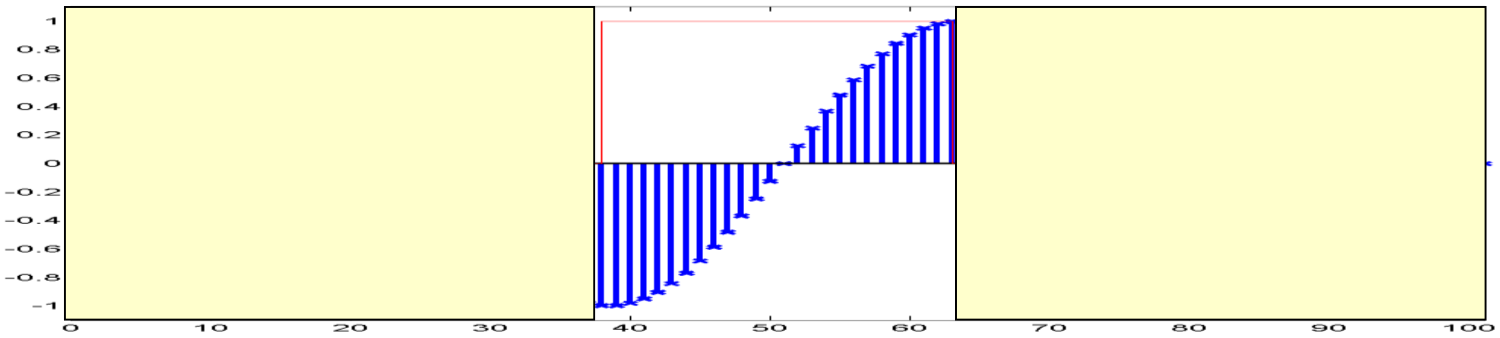




Magnitude spectrum

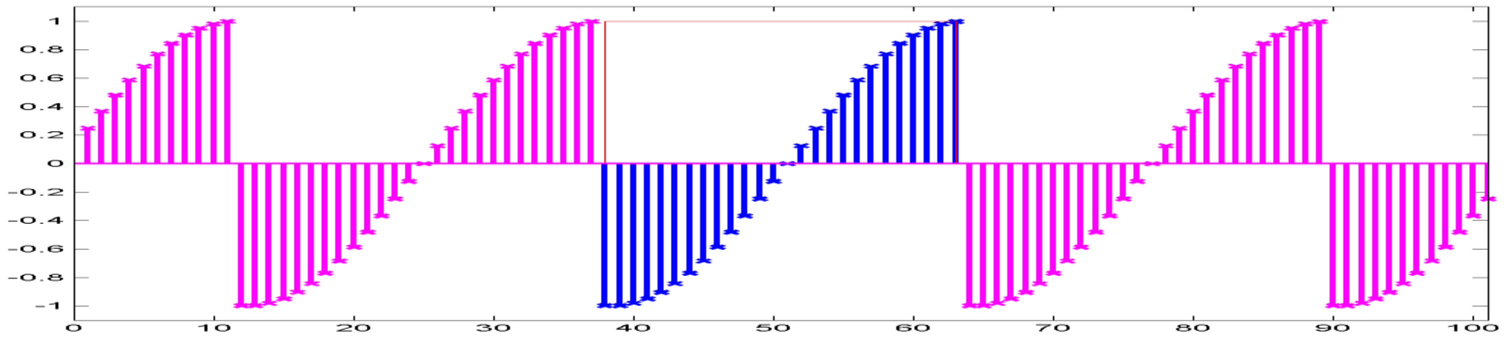
- The DFT of one period of the sinusoid shown in the figure computes the Fourier series of the entire sinusoid from –infinity to +infinity

Windowing



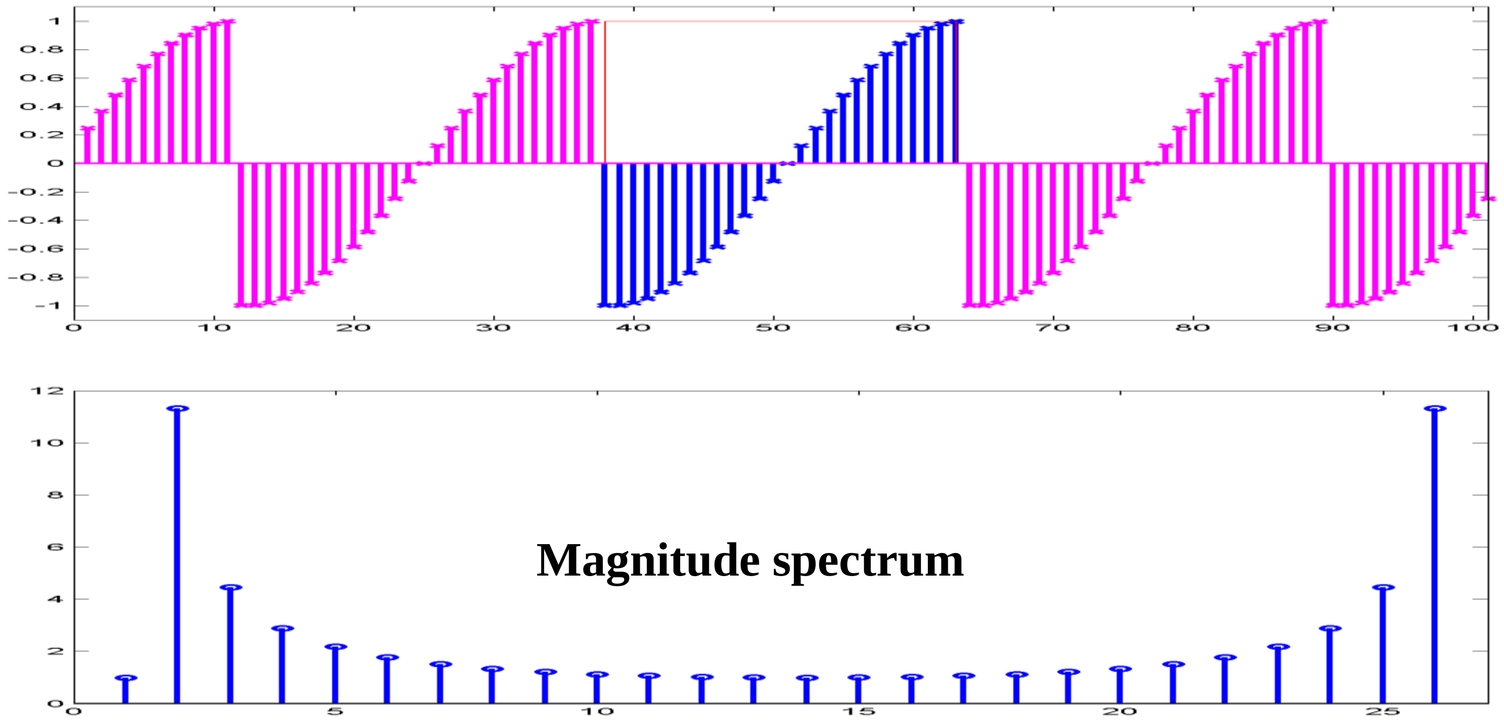
- The DFT of *any* sequence computes the Fourier series for an infinite repetition of that sequence

Windowing



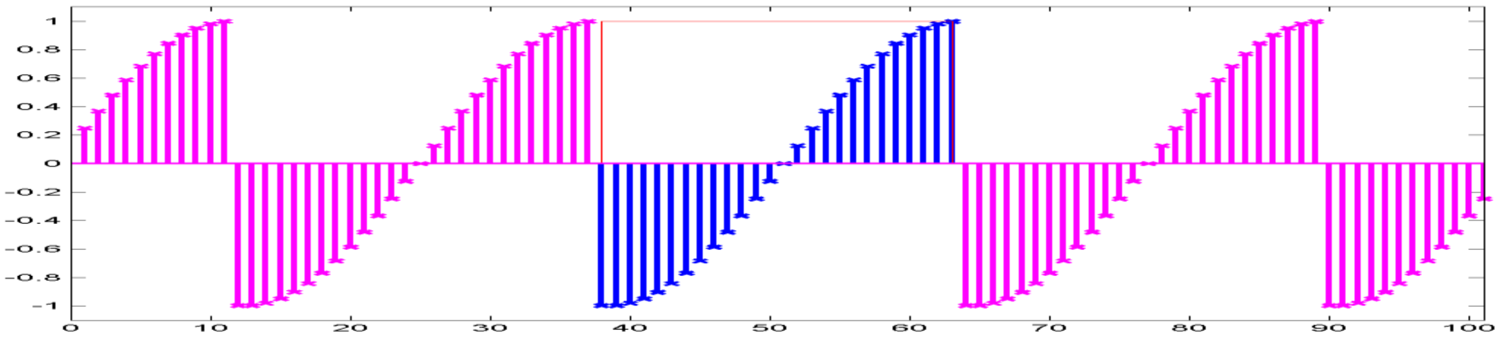
- The DFT of *any* sequence computes the Fourier series for an infinite repetition of that sequence

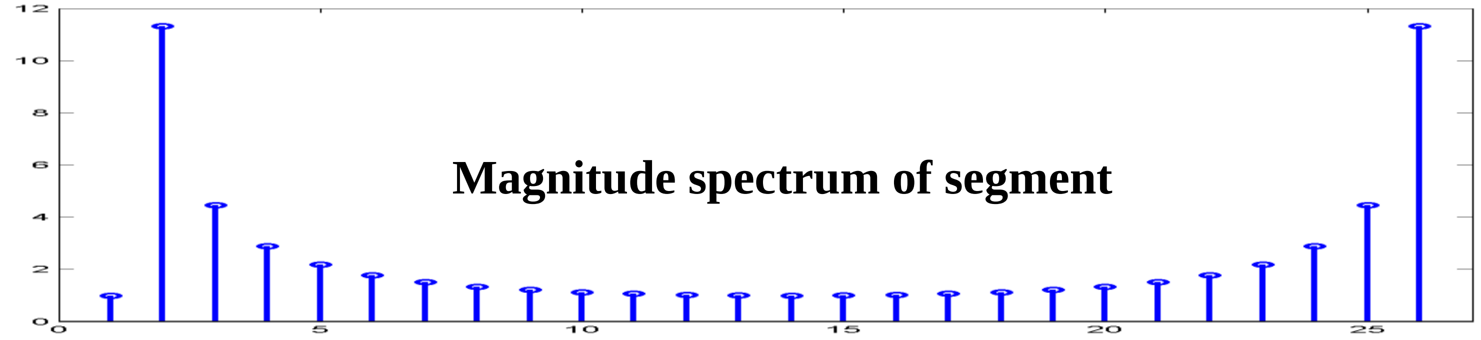
Windowing

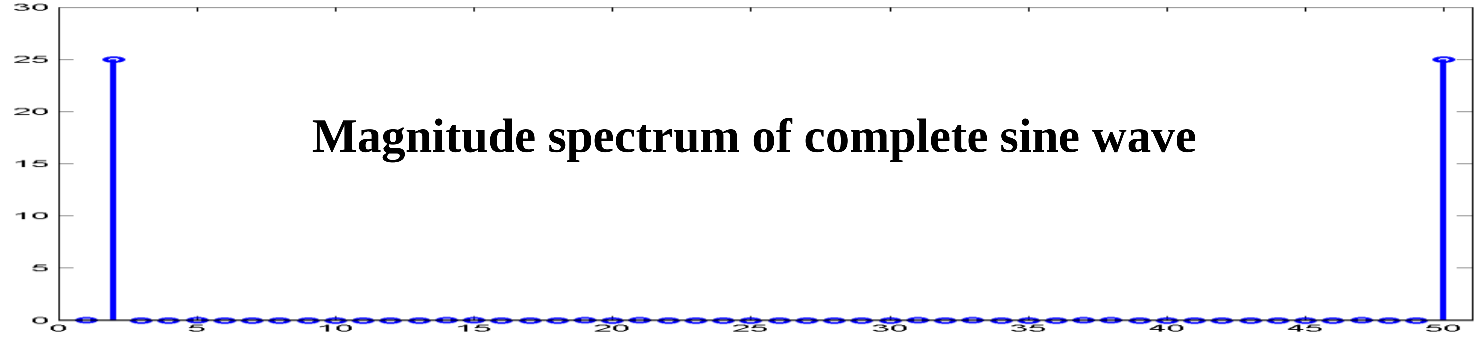


- The DFT of *any* sequence computes the Fourier series for an infinite repetition of that sequence
- The DFT of a partial segment of a sinusoid computes the Fourier series of an inifinite repetition of that segment, and not of the entire sinusoid
- This will not give us the DFT of the sinusoid itself!

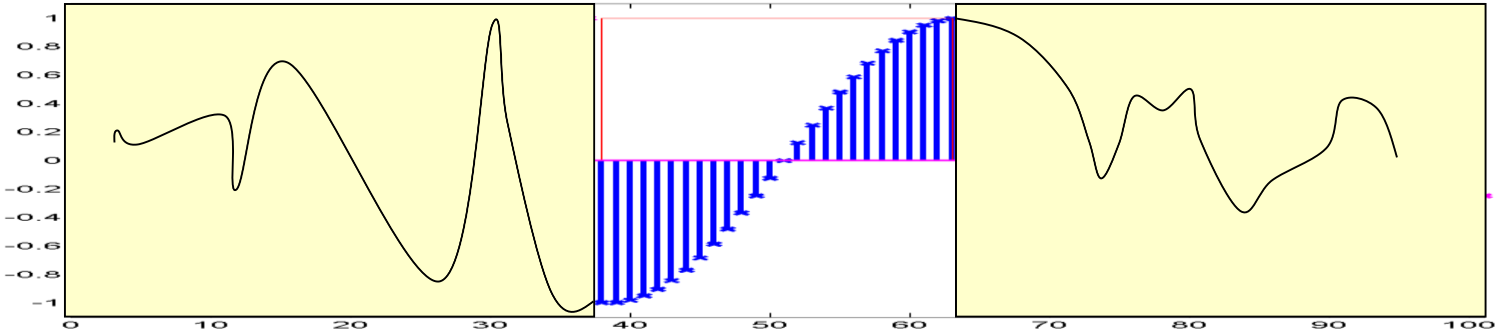
Windowing




Magnitude spectrum of segment

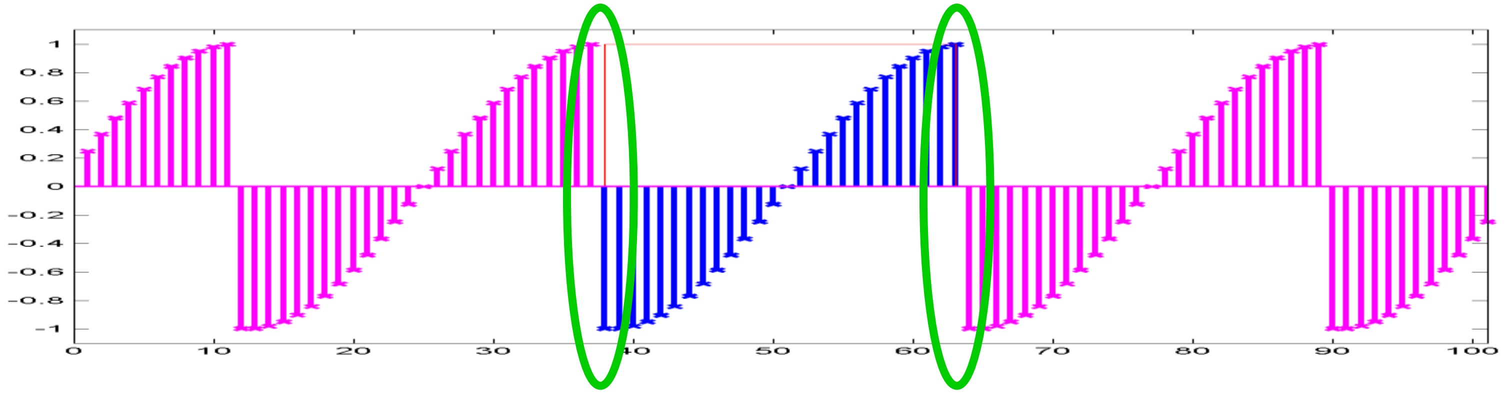

Magnitude spectrum of complete sine wave

Windowing



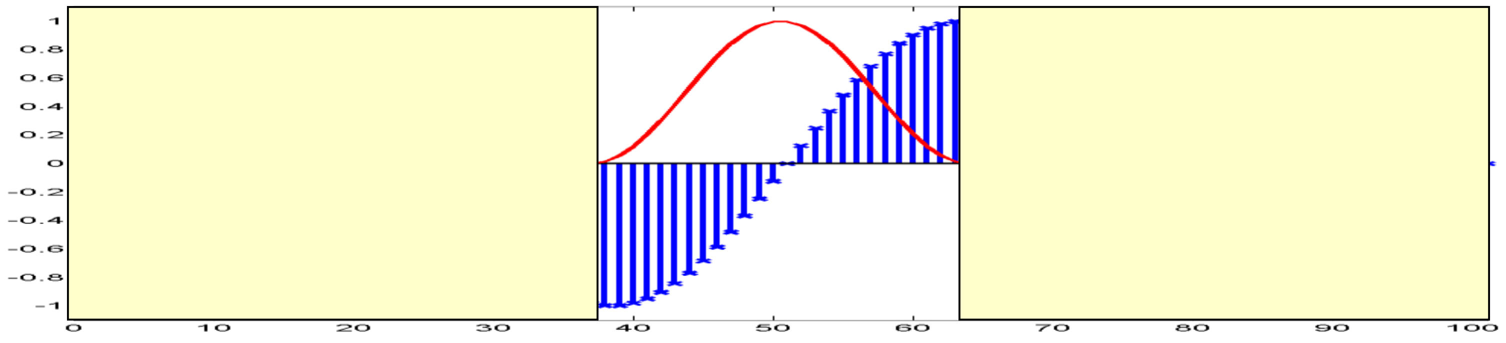
- The difference occurs due to two reasons:
- The transform cannot know what the signal actually looks like outside the observed window
  - We must infer what happens outside the observed window from what happens inside

Windowing



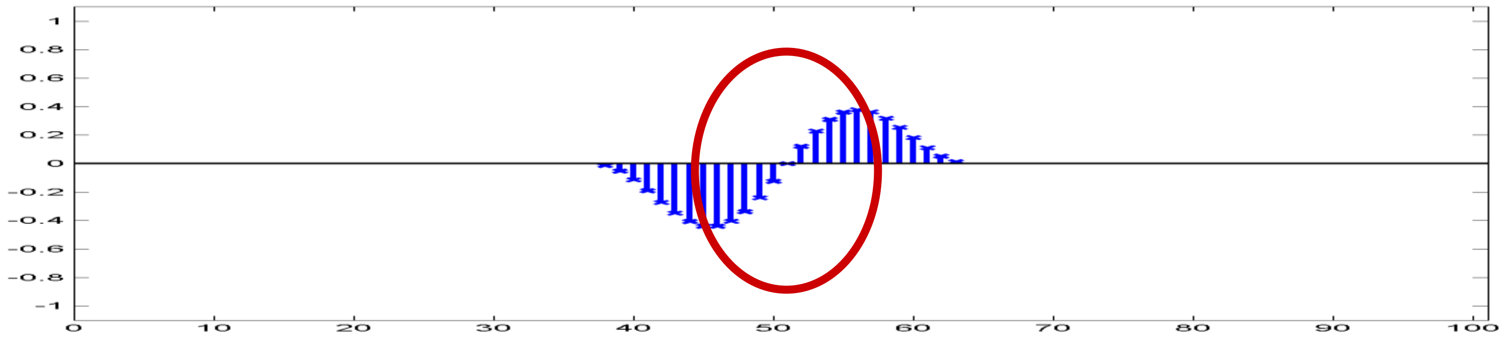
- The difference occurs due to two reasons:
- The transform cannot know what the signal actually looks like outside the observed window
  - We must infer what happens outside the observed window from what happens inside
- The implicit repetition of the observed signal introduces large discontinuities at the points of repetition
  - This distorts even our measurement of what happens at the boundaries of what has been reliably observed
  - The actual signal (whatever it is) is unlikely to have such discontinuities

Windowing



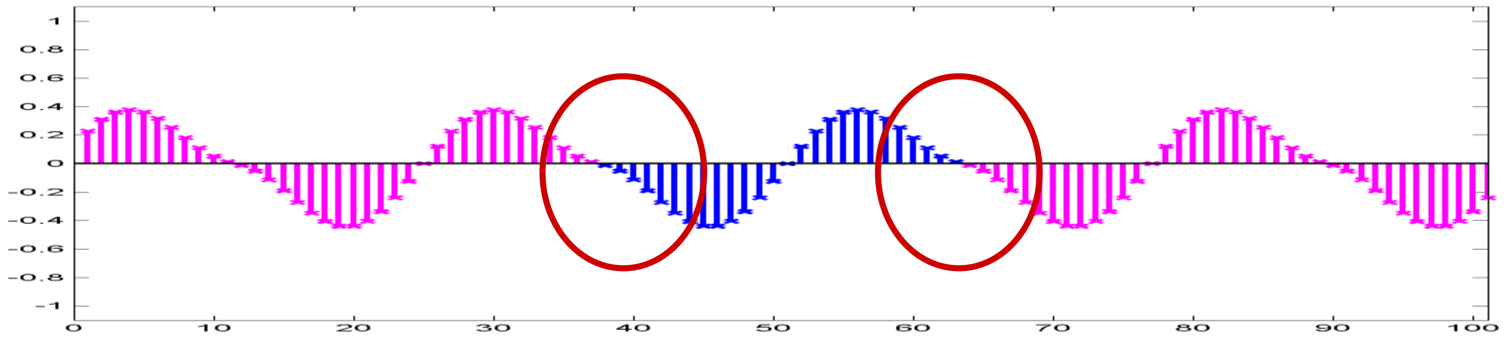
- While we can never know what the signal looks like outside the window, we can try to minimize the discontinuities at the boundaries
- We do this by multiplying the signal with a *window* function
  - We call this procedure windowing
  - We refer to the resulting signal as a “windowed” signal

Windowing



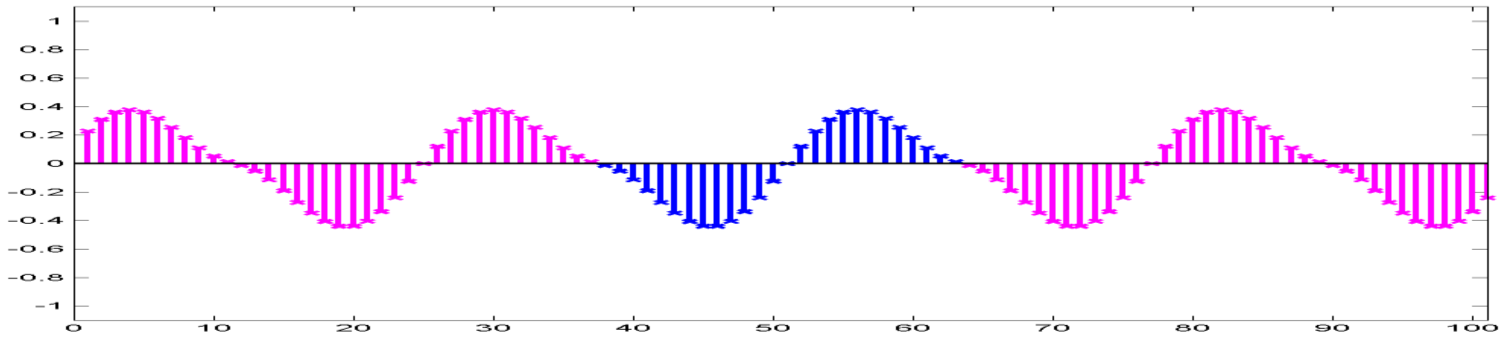
- While we can never know what the signal looks like outside the window, we can try to minimize the discontinuities at the boundaries
- We do this by multiplying the signal with a *window* function
  - We call this procedure windowing
  - We refer to the resulting signal as a “windowed” signal
- Windowing attempts to do the following:
  - Keep the windowed signal similar to the original in the central regions

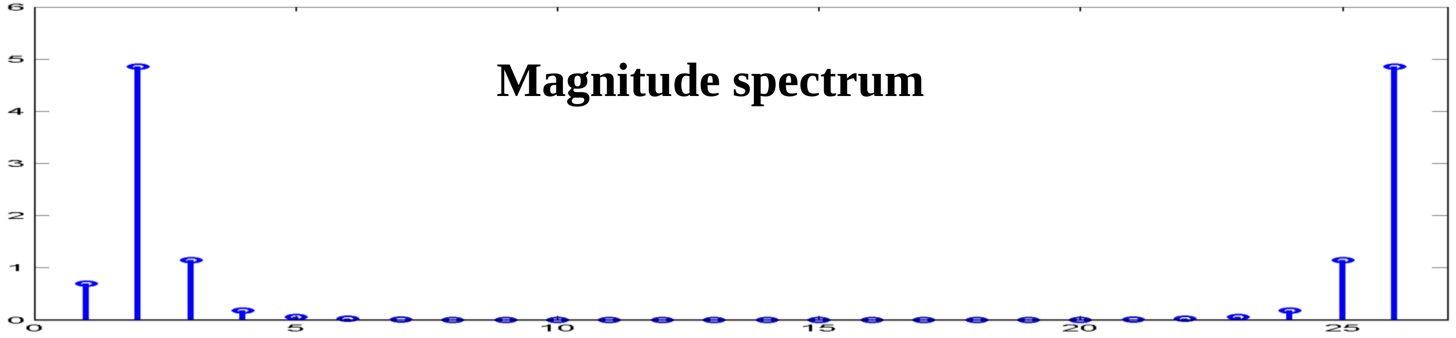
Windowing



- While we can never know what the signal looks like outside the window, we can try to minimize the discontinuities at the boundaries
- We do this by multiplying the signal with a *window* function
  - We call this procedure windowing
  - We refer to the resulting signal as a “windowed” signal
- Windowing attempts to do the following:
  - Keep the windowed signal similar to the original in the central regions
  - Reduce or eliminate the discontinuities in the implicit periodic signal

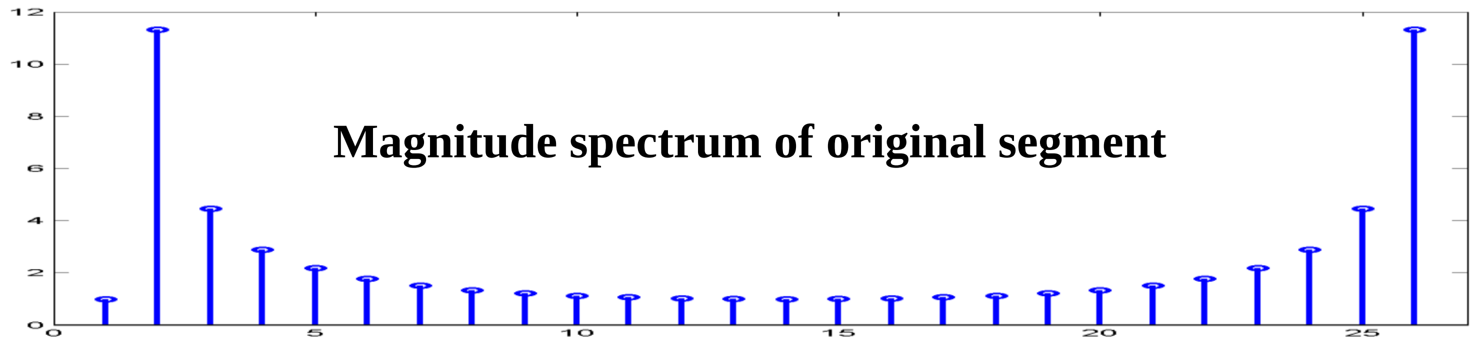
Windowing

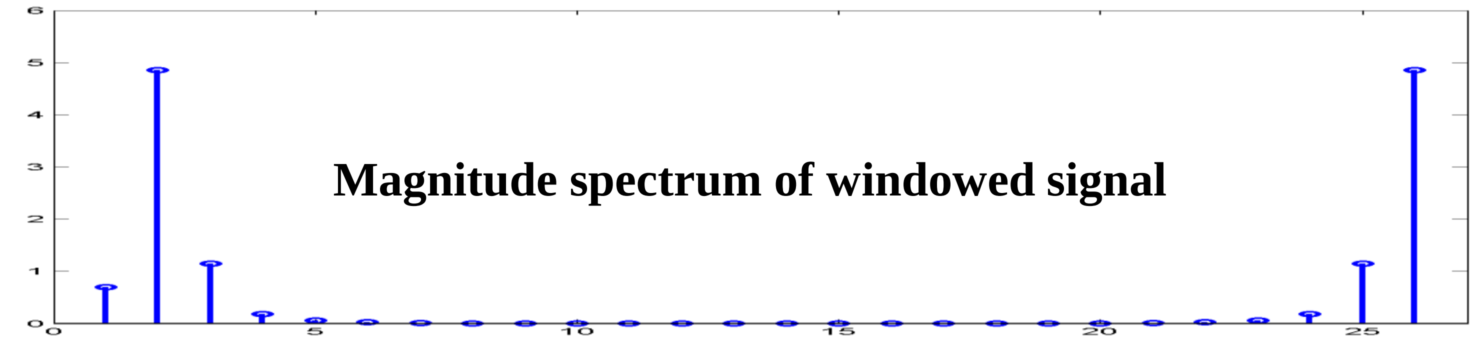


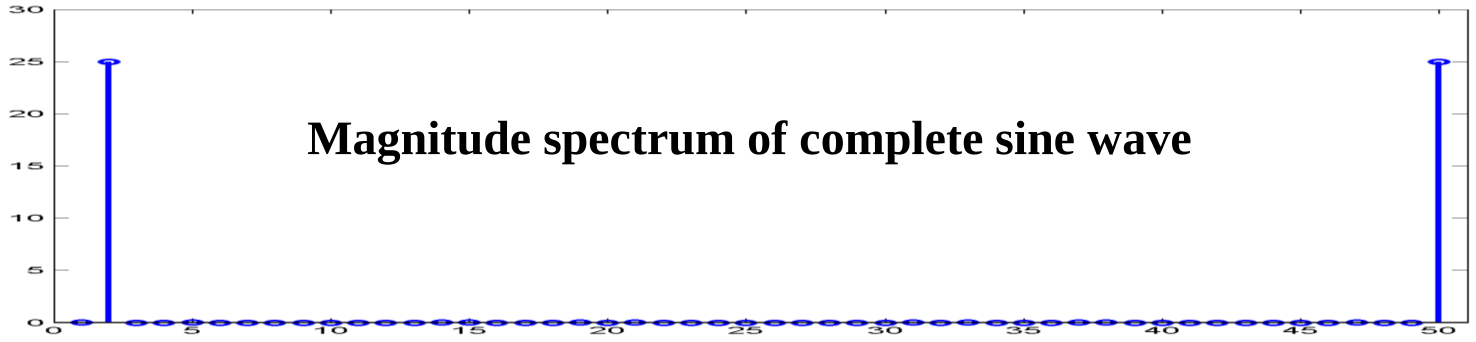


- The DFT of the windowed signal does not have any artefacts introduced by discontinuities in the signal
- Often it is also a more faithful reproduction of the DFT of the complete signal whose segment we have analyzed

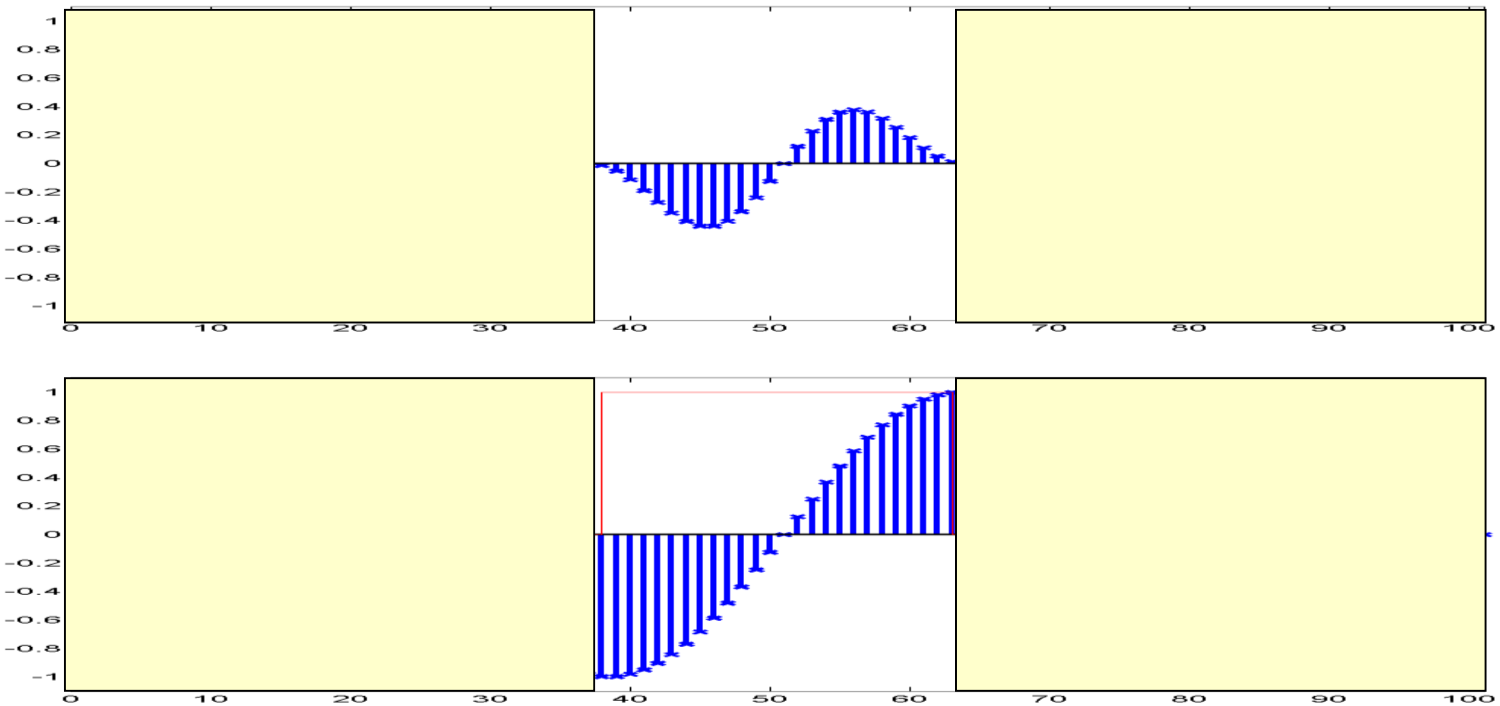
Windowing


Magnitude spectrum of original segment
0
2
4
6
8
10
12
0
5
10
15
20
25


Magnitude spectrum of windowed signal
0
1
2
3
4
5
6
0
5
10
15
20
25

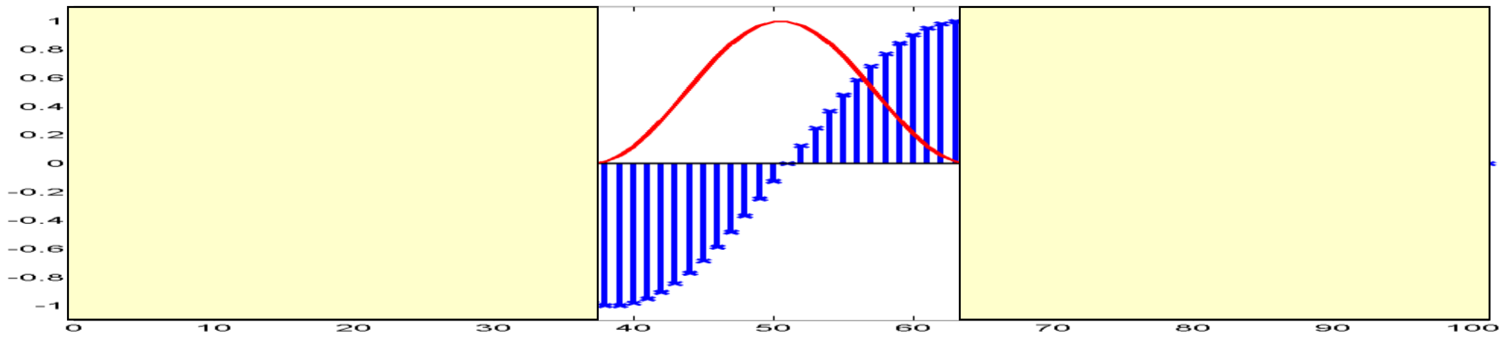

Magnitude spectrum of complete sine wave
0
5
10
15
20
25
30
0
5
10
15
20
25
30
35
40
45
50

Windowing



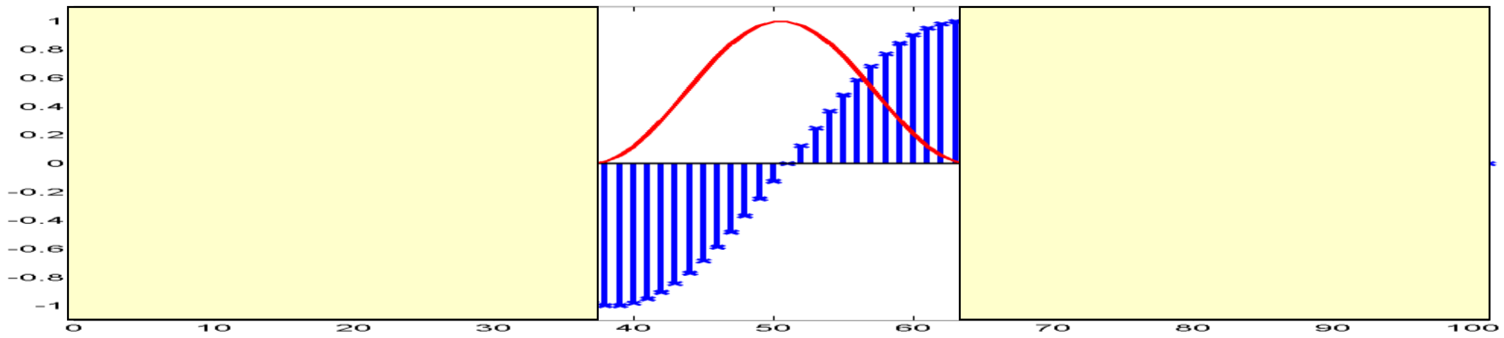
- Windowing is not a perfect solution
  - The original (unwindowed) segment is identical to the original (complete) signal within the segment
  - The windowed segment is often not identical to the complete signal anywhere
- Several windowing functions have been proposed that strike different tradeoffs between the fidelity in the central regions and the smoothing at the boundaries

Windowing

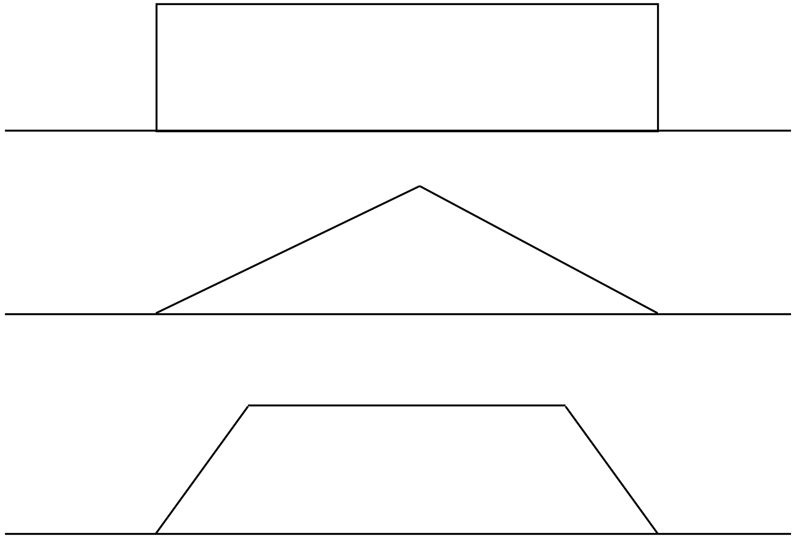


- Cosine windows:
  - Window length is M
  - Index begins at 0
- Hamming: $w[n] = 0.54 - 0.46 \cos(2\pi n/M)$
- Hanning: $w[n] = 0.5 - 0.5 \cos(2\pi n/M)$
- Blackman: $w[n] = 0.42 - 0.5 \cos(2\pi n/M) + 0.08 \cos(4\pi n/M)$

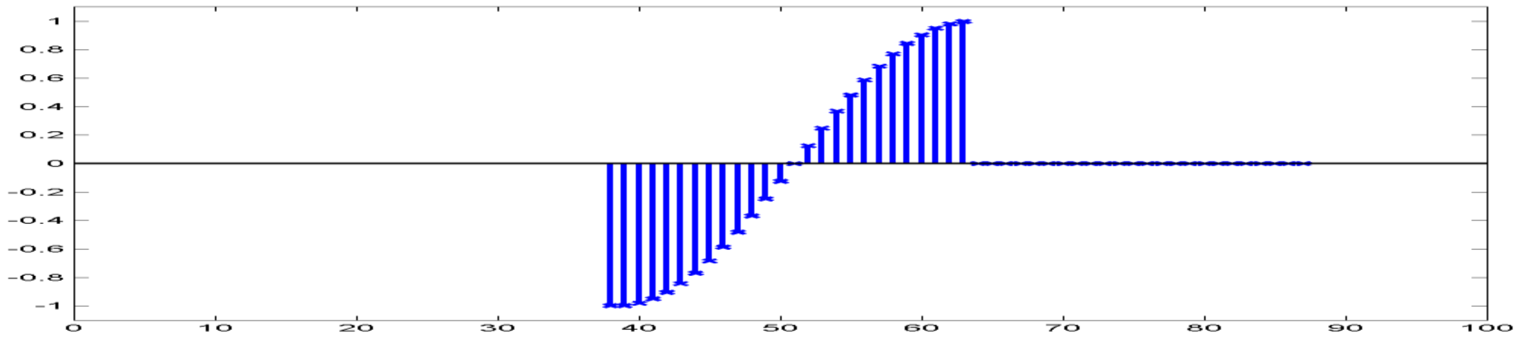
Windowing



- Geometric windows:
  - Rectangular (boxcar):
  - Triangular (Bartlett):
  - Trapezoid:

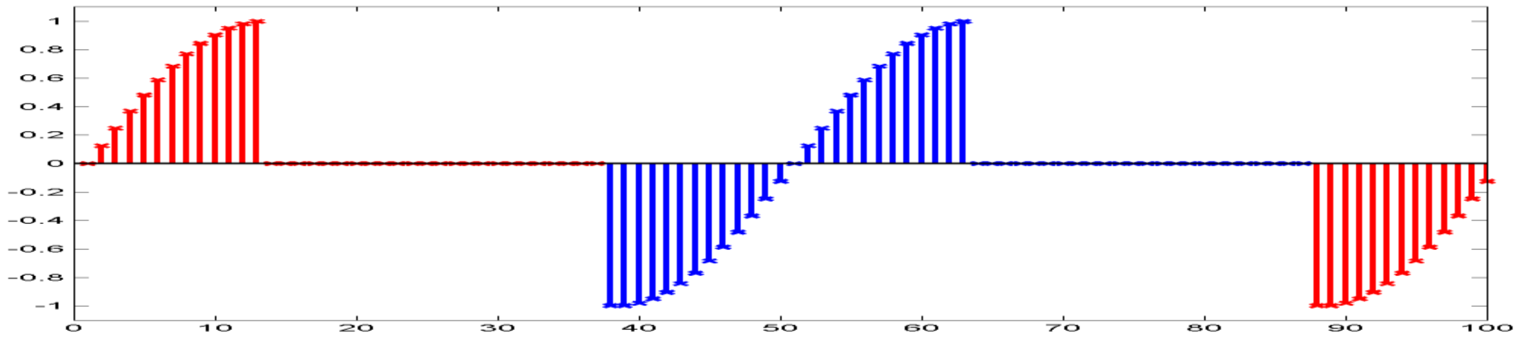


Zero Padding



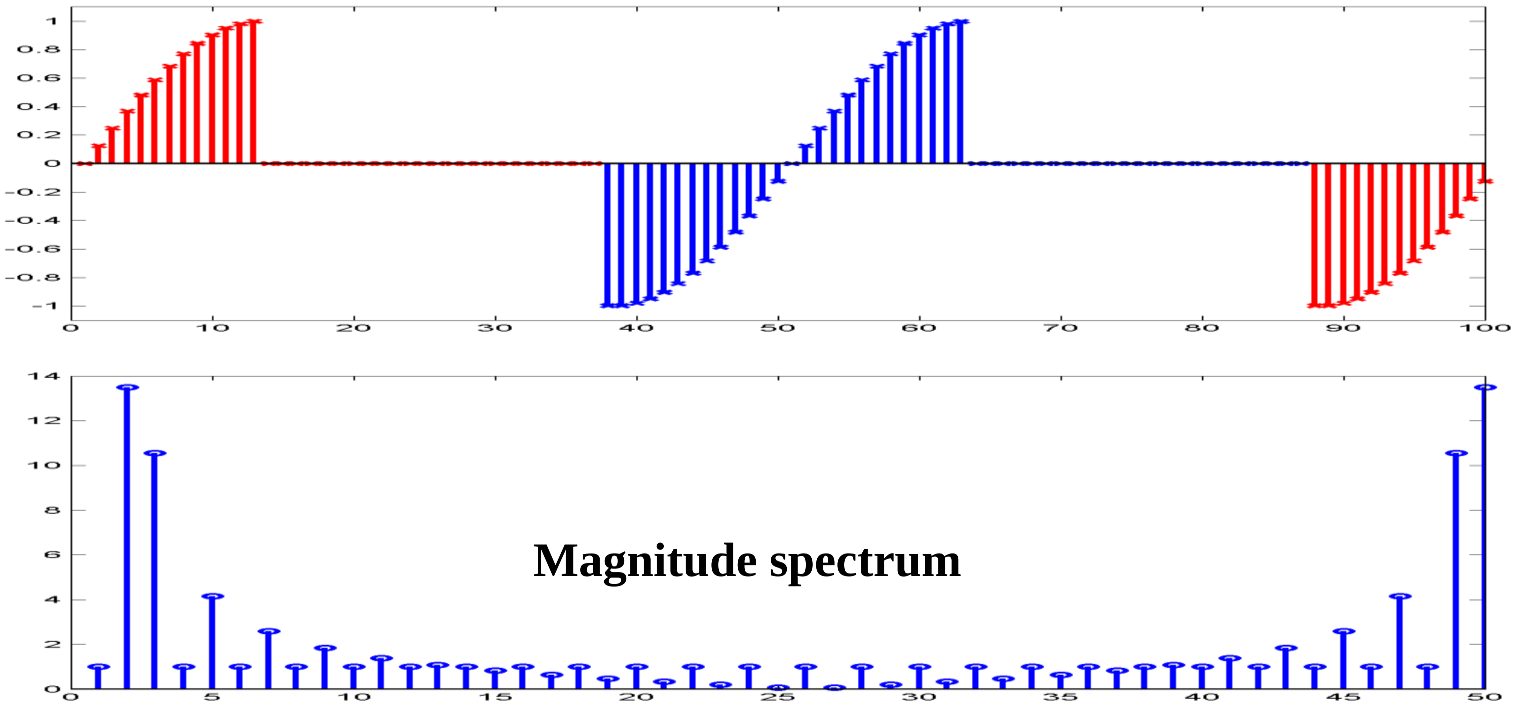
- **We can pad zeros to the end of a signal to make it a desired length**
  - Useful if the FFT (or any other algorithm we use) requires signals of a specified length
- The consequence of zero padding is to change the periodic signal whose Fourier spectrum is being computed by the DFT

Zero Padding



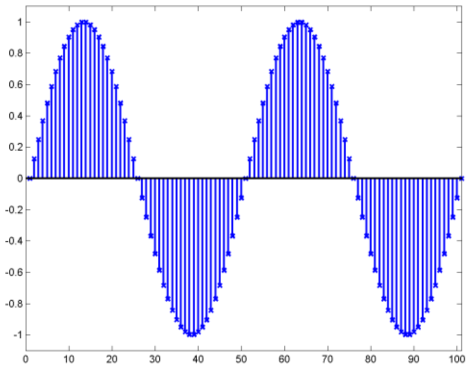
- **We can pad zeros to the end of a signal to make it a desired length**
  - Useful if the FFT (or any other algorithm we use) requires signals of a specified length
- The consequence of zero padding is to change the periodic signal whose Fourier spectrum is being computed by the DFT

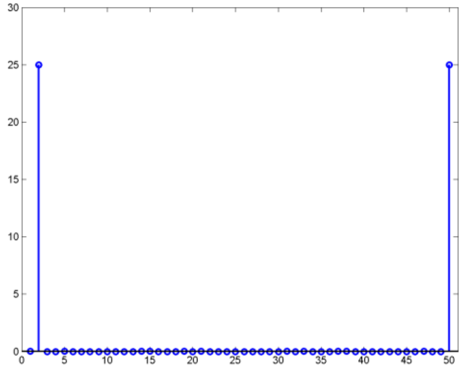
Zero Padding

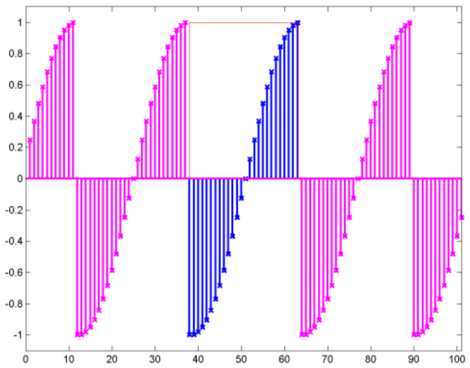


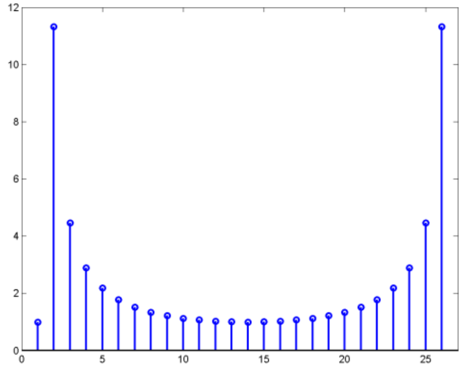
- The DFT of the zero padded signal is essentially the same as the DFT of the unpadded signal, with additional spectral samples inserted in between
  - It does not contain any additional information over the original DFT
  - It also does not contain less information

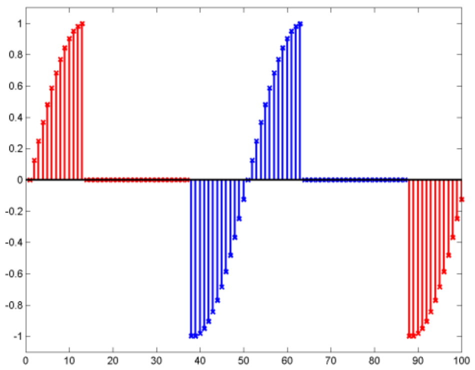
Magnitude spectra

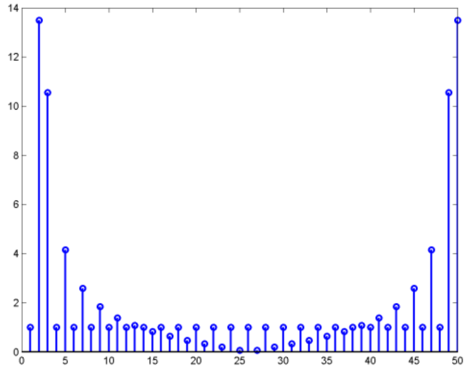






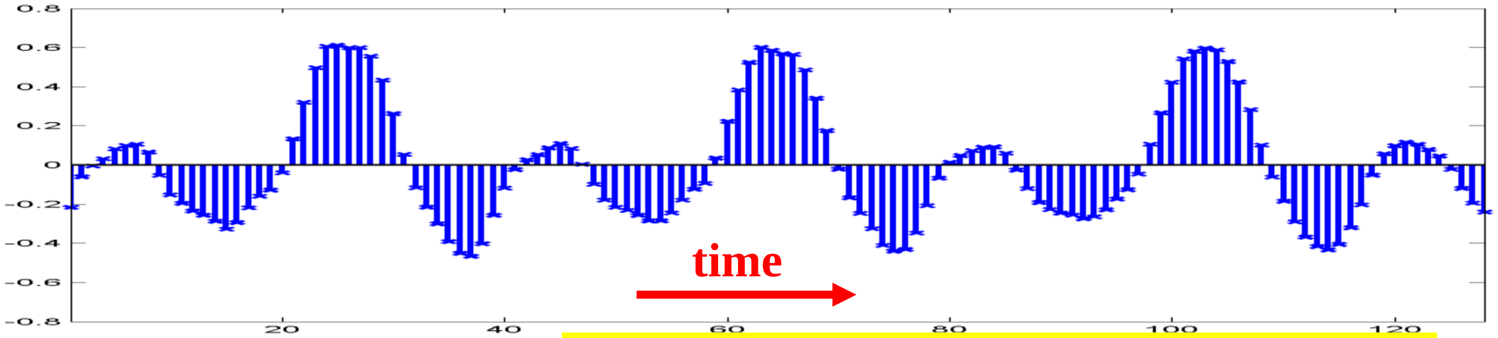




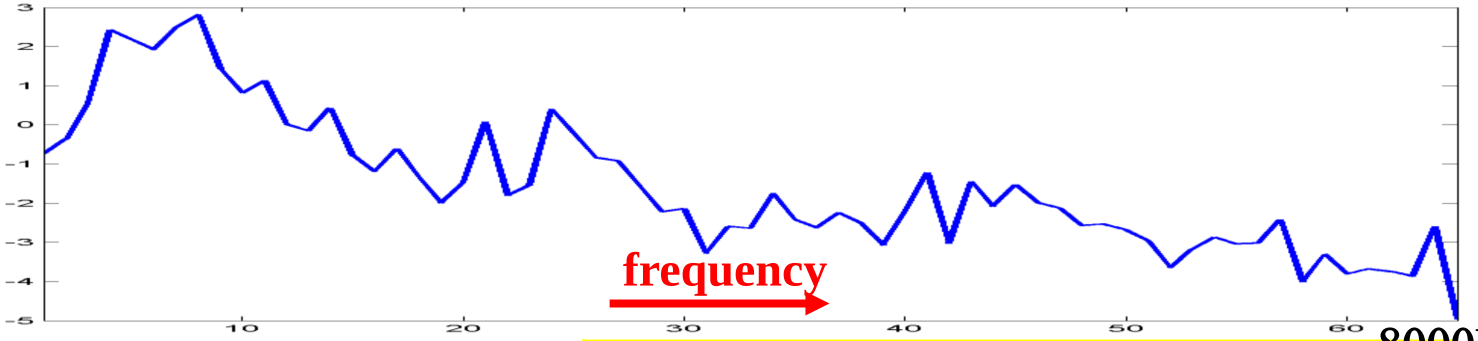


Zero padding a speech signal

128 samples from a speech signal sampled at 16000 Hz

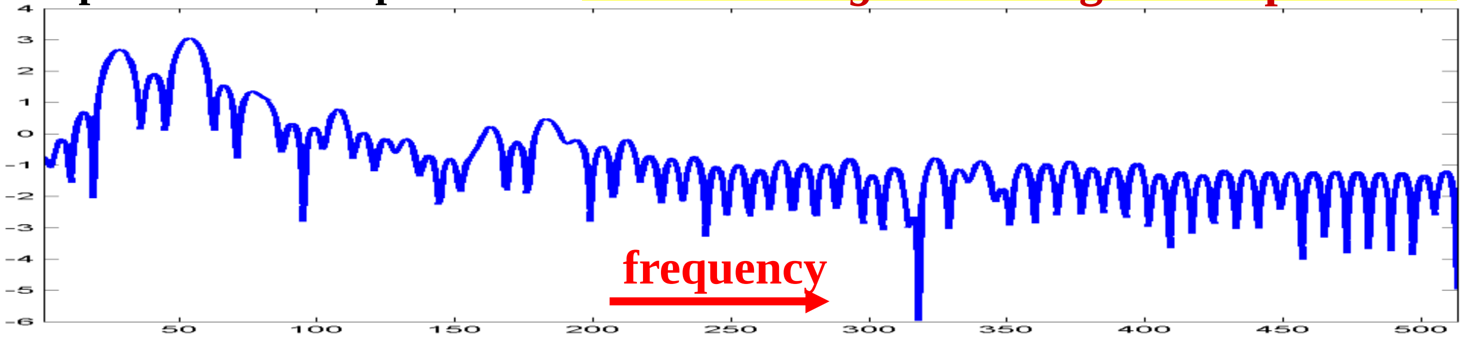


The first 65 points of a 128 point DFT. **Plot shows *log* of the magnitude spectrum**



8000Hz

The first 513 points of a 1024 point DFT. **Plot shows *log* of the magnitude spectrum**



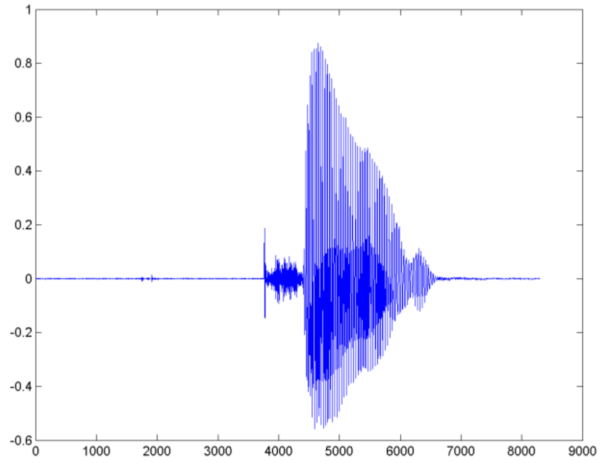
8000Hz

Pre-emphasizing a speech signal

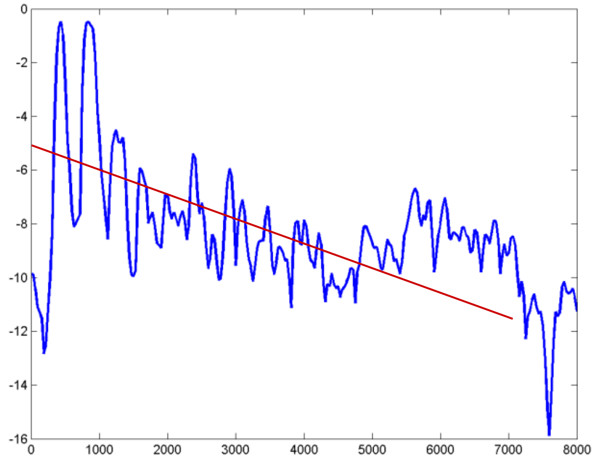
- The spectrum of the speech signal naturally has lower energy at higher frequencies

- This can be observed as a downward trend on a plot of the logarithm of the magnitude spectrum of the signal

- For many applications this can be undesirable
  - E.g. Linear predictive modeling of the spectrum

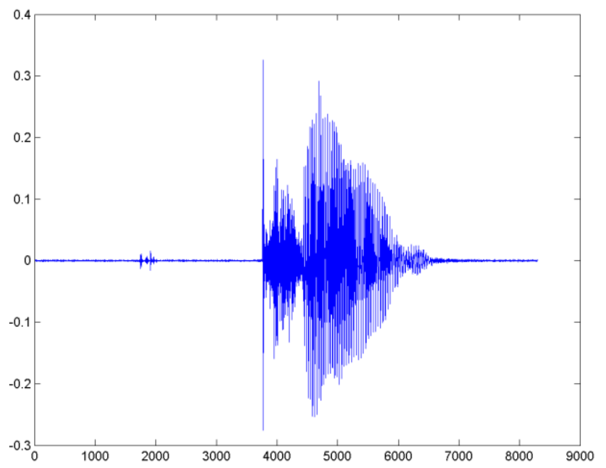


Log(average(magnitude spectrum))



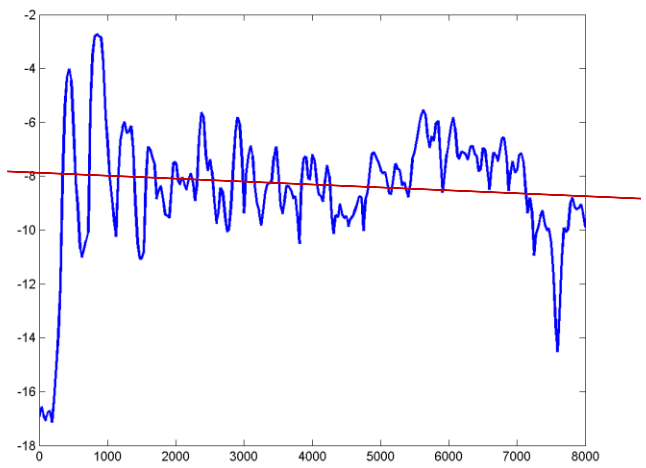
Pre-emphasizing a speech signal

- This spectral tilt can be corrected by pre-emphasizing the signal
  - $s_{preemp}[n] = s[n] - \alpha * s[n-1]$
  - Typical value of $\alpha = 0.95$



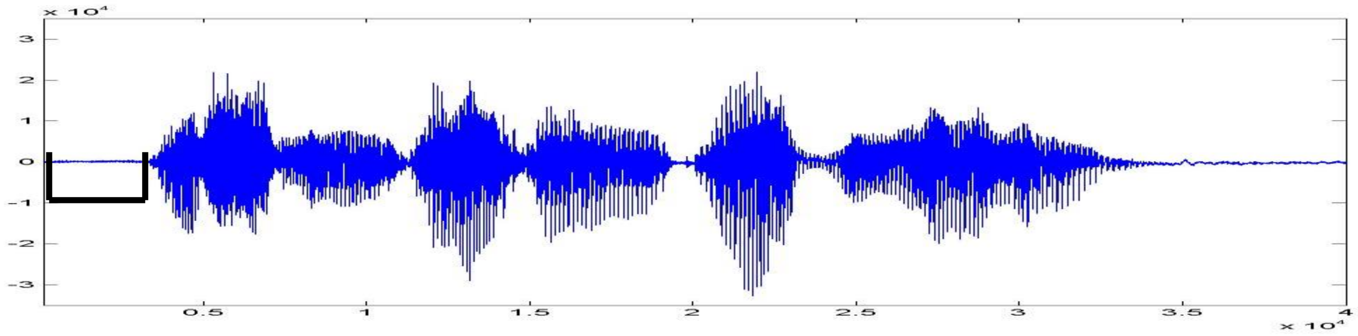
- This is a form of differentiation that boosts high frequencies

Log(average(magnitude spectrum))



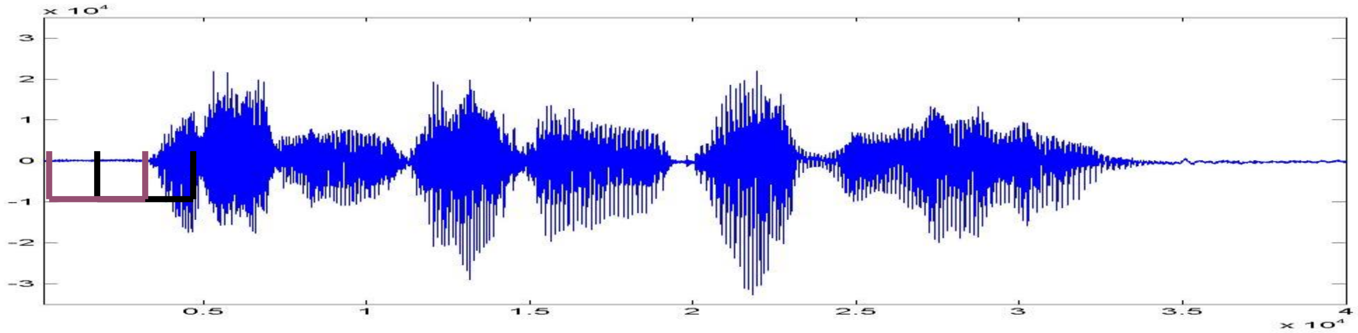
- This spectrum of the pre-emphasized signal has a more horizontal trend
  - Good for linear prediction and other similar methods

The process of parametrization



The signal is processed in segments.
Segments are typically 25 ms wide.

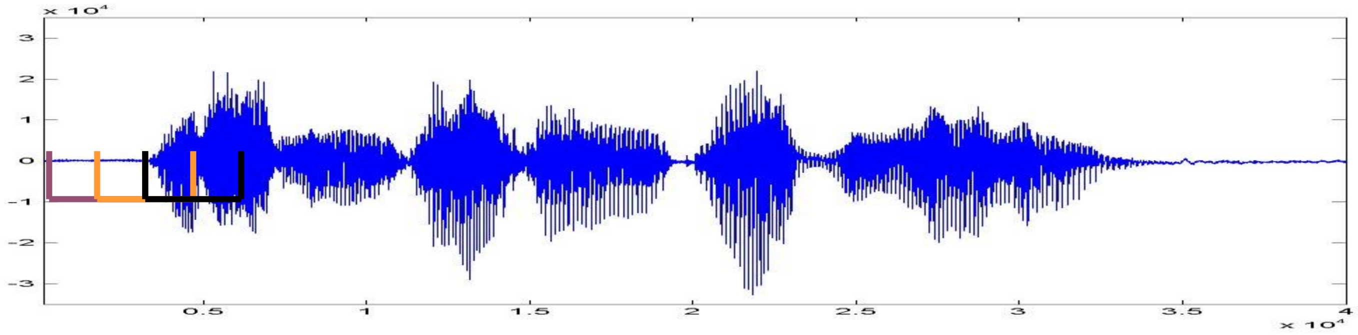
The process of parametrization



The signal is processed in segments. Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

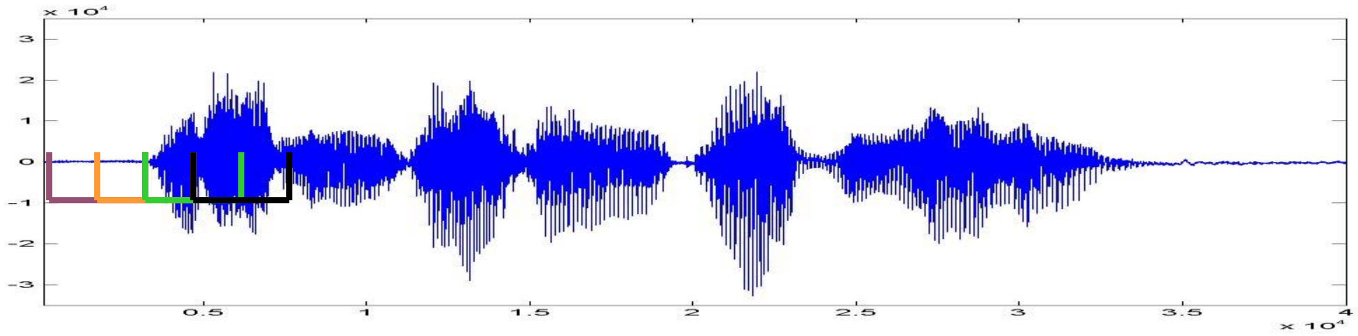
The process of parametrization



The signal is processed in segments.
Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

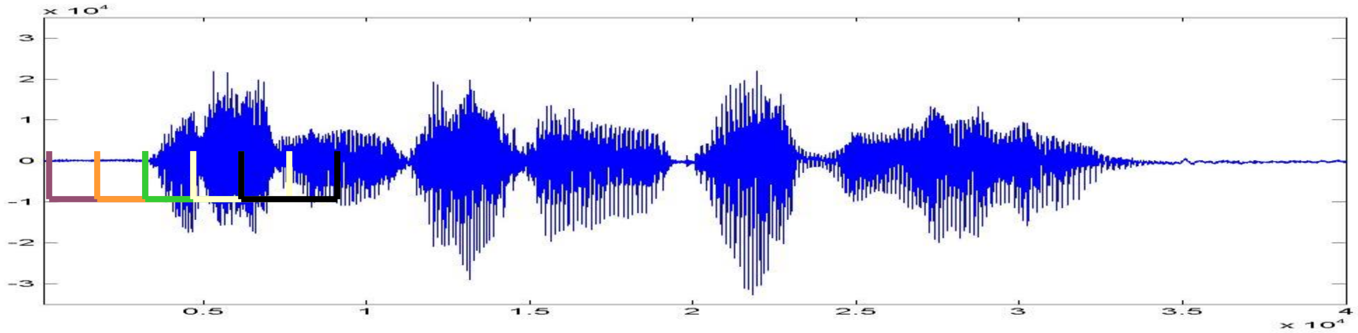
The process of parametrization



The signal is processed in segments. Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

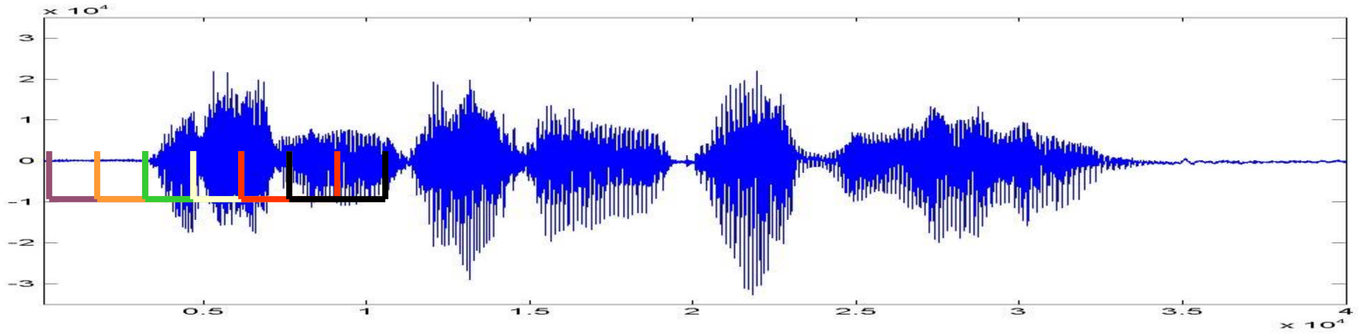
The process of parametrization



The signal is processed in segments. Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

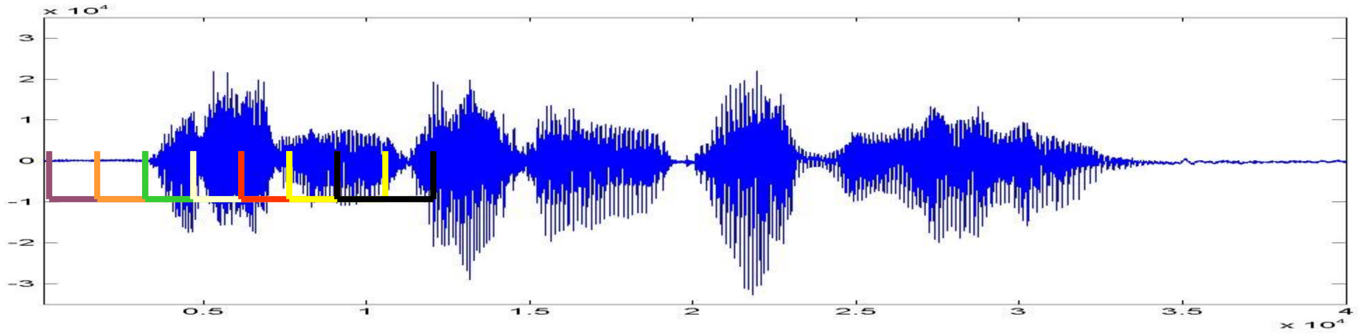
The process of parametrization



The signal is processed in segments. Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

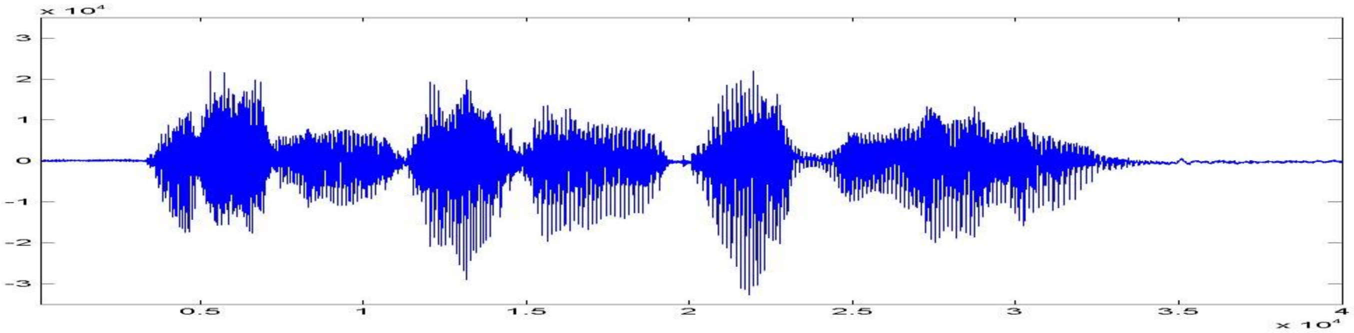
The process of parametrization

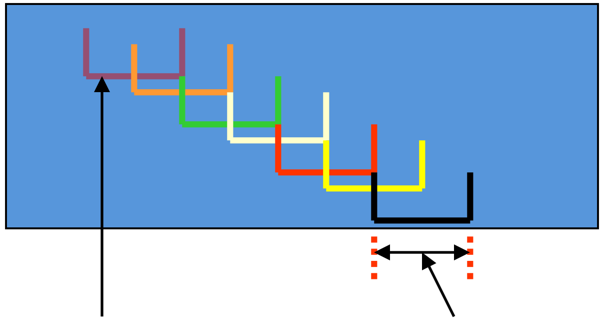


The signal is processed in segments. Segments are typically 25 ms wide.

Adjacent segments typically overlap by 15 ms.

The process of parametrization



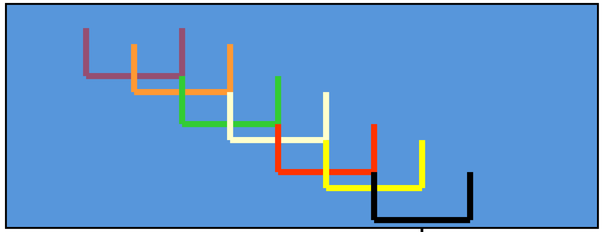


Segments shift every 10 milliseconds

Each segment is typically 20 or 25 milliseconds wide

Speech signals do not change significantly within this short time interval

The process of parametrization



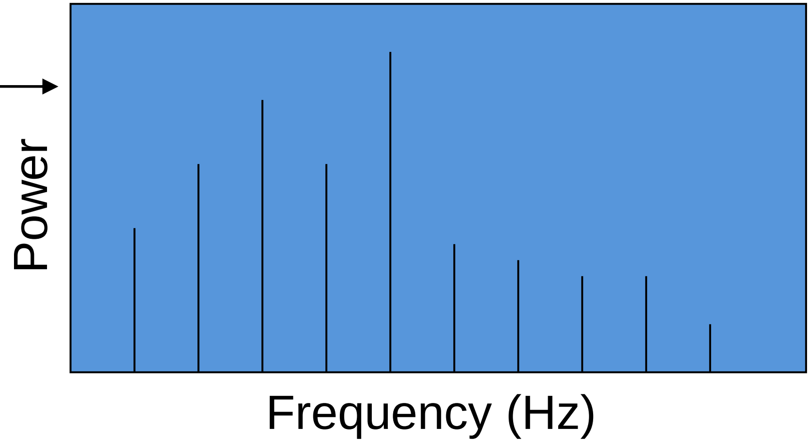
Each segment is pre-emphasized

Pre-emphasized segment

The pre-emphasized segment is windowed

Pre-emphasized and windowed segment

The DFT of the segment, and from it the power spectrum of the segment is computed



= power spectrum

Final Project

Image by
kirkh.deviantart.com

Final Project

- **Team members from 3 (Min) to 5 (Max).**
- **Submit the names and short proposal (1 page) by 28 March (Deadline).**
- **The final project should have a short printed report at max 5 pages + Code.**
- **Prepare short presentation (5-10 slides) for max 10 minutes for all the team members.**
- **The short report should contains the following (Must):**
  1. ***Abstract***
  2. ***Short Introduction (Max 1 page)***
  3. ***Framework diagram***
  4. ***Problem statement***
  5. ***Objectives***
  6. ***The proposed System (Max 2 page)***
  7. ***Conclusion***
  8. ***References and citations along the text.***

Note: **try to avoid copy and paste as much as you can in the report code as well.**

Final Project

- **Topics:**
  1. **Speech Recognition for English language (at least 10 different words).**
  2. **Speech Recognition for Arabic language (at least 10 different words).**
  3. **Speech Recognition for English alphabets.**
  4. **Speech Recognition for Arabic alphabets.**
  5. **Speaker Verification and Identification - English (at least 3 different speakers).**
  6. **Speaker Verification and Identification - Arabic (at least 3 different speakers).**
  7. **Implement speech chatbot.**
  8. **Voice Control System for Smart Home.**