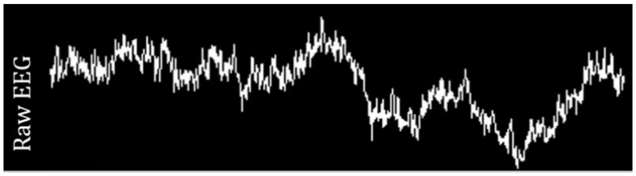
Hilbert Transform

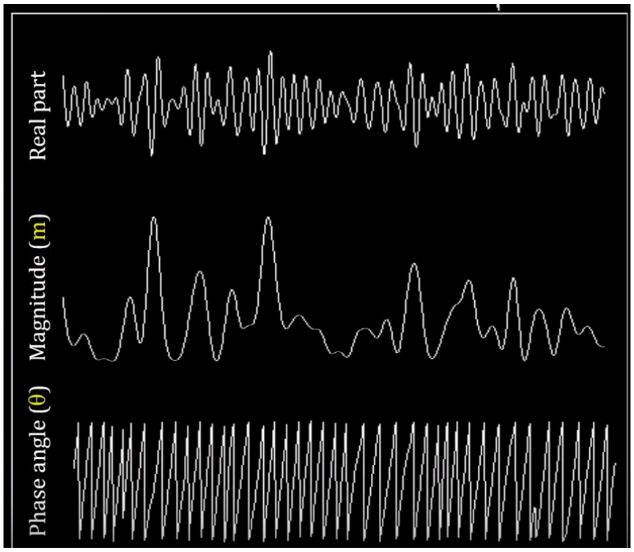
Table of Contents

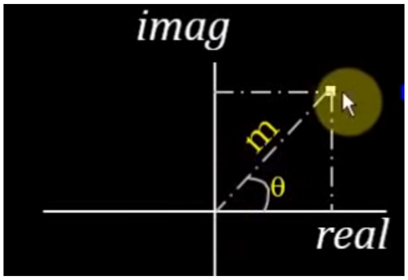
What is Hilbert Transform?

The discrete Hilbert transform is a process used to generate complex-valued signals from real-valued signals.

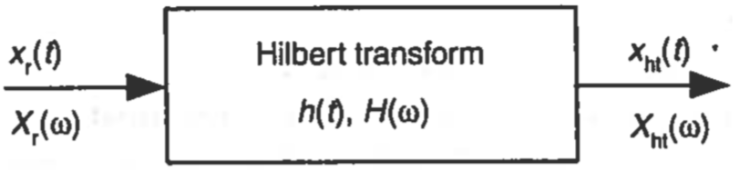
It is a method for extracting the magnitude and phase information from a real-valued signal like a EEG data:







Notation used to define the continuous Hilbert transform:



$x_r(t)$ = a real continuous time-domain input signal
$h(t)$ = the time impulse response of a Hilbert transformer
$x_{ht}(t)$ = the HT of $x_r(t)$, ($x_{ht}(t)$ is also a real time-domain signal)
$X_r(\omega)$ = the Fourier transform of real input $x_r(t)$
$H(\omega)$ = the frequency response (complex) of a Hilbert transformer
$X_{ht}(\omega)$ = the Fourier transform of output $x_{ht}(t)$
ω = continuous frequency measured in radians/second
t = continuous time measured in seconds.

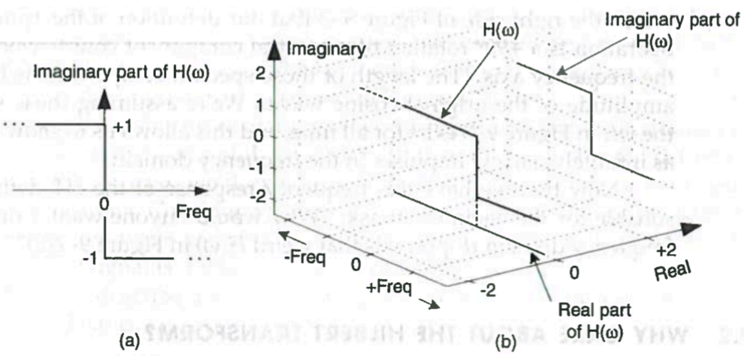


Figure 9-2 The complex frequency response of $H(\omega)$.

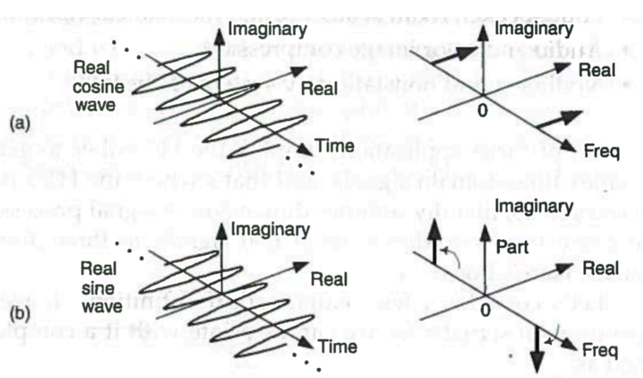


Figure 9-3 The Hilbert transform: (a) $\cos(\omega t)$; (b) its transform is $\sin(\omega t)$.

Why do we need it?

Typically, we start with a real-valued signal which can be conceptualised as:

$$x_r(t) = A\cos(\omega\, t)$$

Given a real-value signal, we cannot estimate or extract the phase angle and the power. What we need to represent the signal as a complex sinusoid

$$x_c(t) = A\, e^{j\omega t}$$

$$x_c(t) = A\cos(\omega\, t) + j\, A\sin(\omega t)$$

$$x_c(t) = x_r(t) + j\, x_i(t)$$

Using complex signals of the real signals simplifies and improves the performance of many signal processing operations.

The answer is: we need to understand the HT because it's useful in so many complex-signal (quadrature) processing applications. A brief search on the Internet reveals HT-related signal processing techniques being used in the following applications:

- Quadrature modulation and demodulation (communications)
- Automatic gain control (AGC)
- Analysis of two- and three-dimensional complex signals
- Medical imaging, seismic data and ocean wave analysis
- Instantaneous frequency estimation
- Radar/sonar signal processing, and time-domain signal analysis using wavelets
- Time difference of arrival (TDOA) measurements
- High definition television (HDTV) receivers
- Loudspeaker, room acoustics, and mechanical vibration analysis
- Audio and color image compression
- Nonlinear and nonstationary system analysis.

Definition

Typically, we start with a real-valued signal which can be conceptualised as:

$$x_r(t) = A\cos(\omega\, t)$$

Given a real-value signal, we cannot estimate or extract the phase angle and the power. What we need to represent the signal as a complex sinusoid using Euler's formula:

$$x_c(t) = A\, e^{j\omega t}$$

$$x_c(t) = A\cos(\omega\, t) + j\, A\sin(\omega t)$$

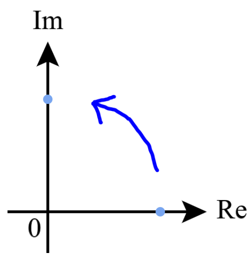
$x_c(t) = x_r(t) + j\,x_i(t)$

where

- $x_c(t)$ is known as an *analytic signal* because it has no negative-frequency spectral components.
- $j\,x_i(t)$ is called the *phase-quadrature component*.

The phase-quadrature component can be obtained by a 90 degree (*quarter-cycle*) shift on the complex plan.

The Hilbert transform of $\cos(\theta)$ is $\sin(\theta)$ i.e., a quarter-cycle shift.



The Hilbert transform rotates the Fourier coefficients in the complex space which converts the real components into the imaginary component. The quarter-cycle shifted signal is then added to the original signal.

In other words, a Hilbert transform takes each frequency component present in the original signal and shifts its phase by $-\frac{\pi}{2}$.

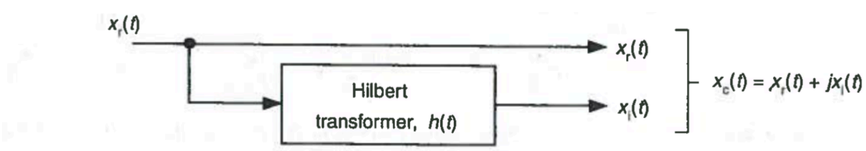


Figure 9-4 Functional relationship between the $x_c(t)$ and $x_r(t)$ signals.

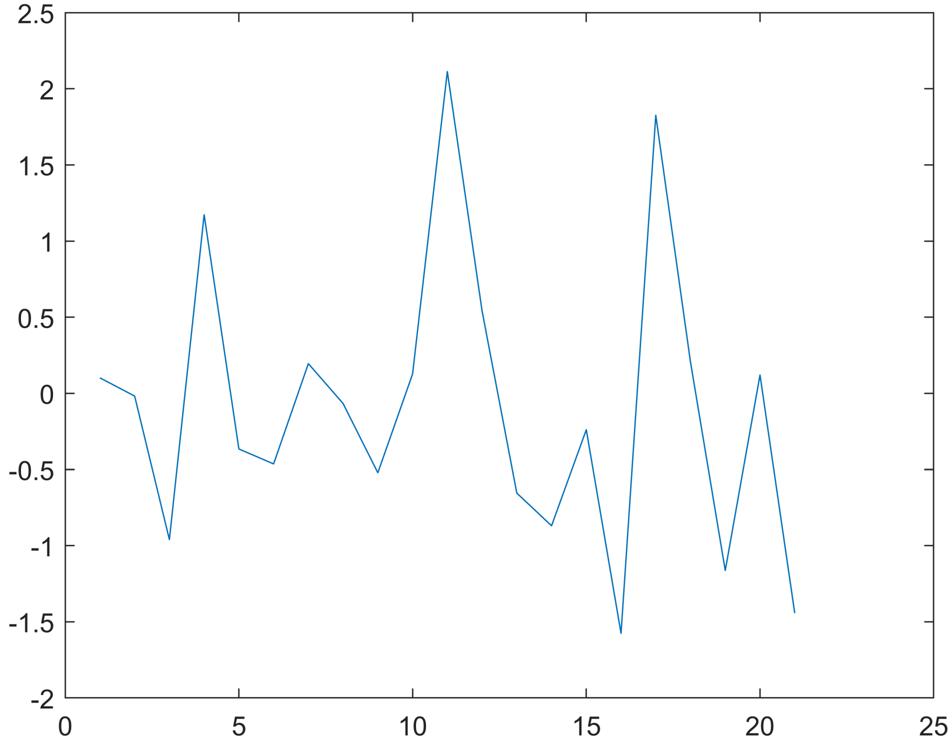
FFT-based Hilbert transform

The FFT-based Hilbert transform can be implemented as three steps:

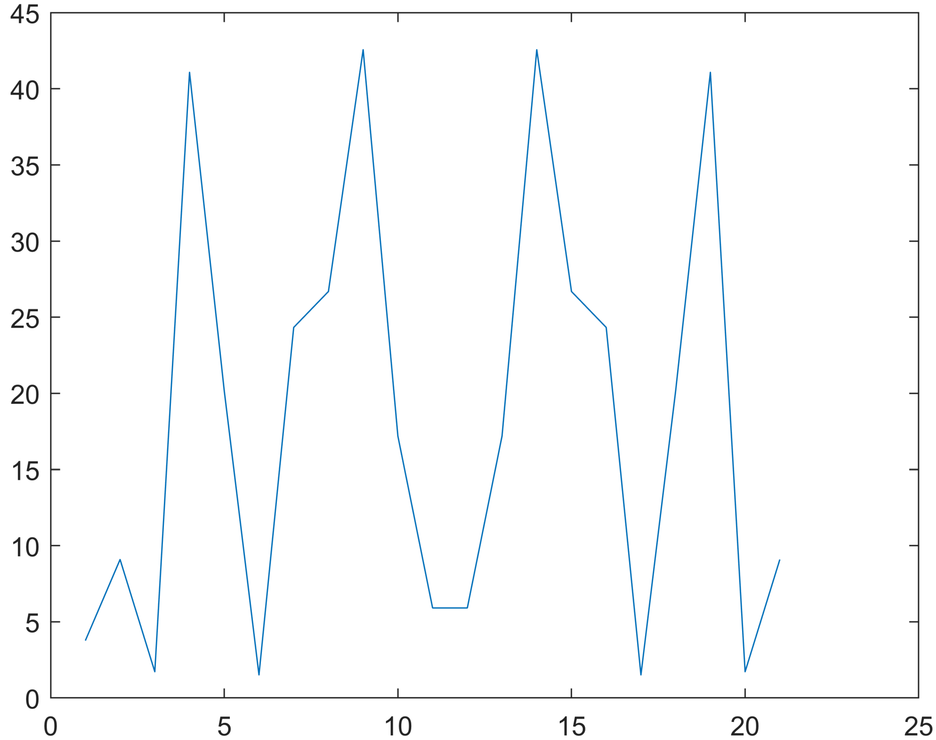
1. Take the FFT of a time-series signal
2. Rotate the Fourier coefficients
3. Take the inverse FFT of the rotated Fourier coefficients to get a time-series signal

```
clear variables;

n = 21;

% Generate a signal
```

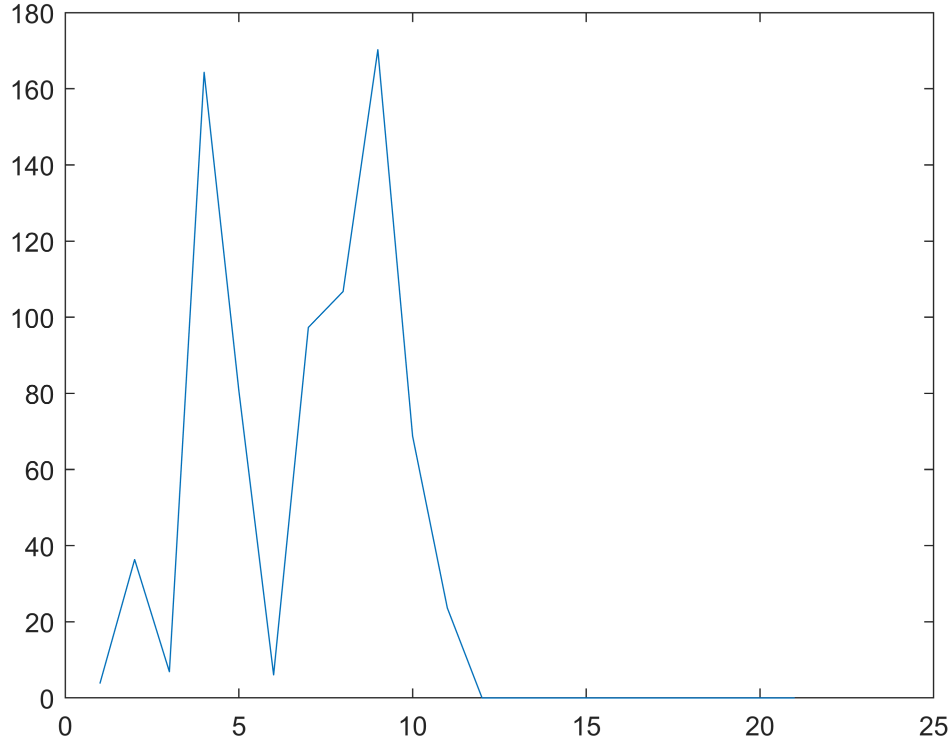
```
x = randn(n, 1);
plot(x)
```



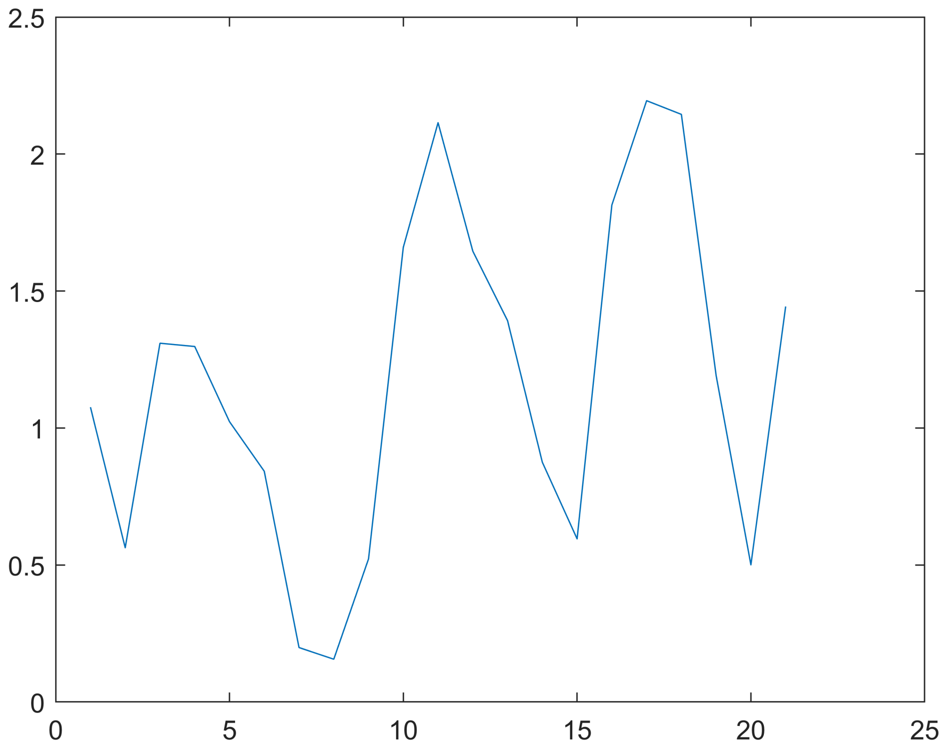

```
% Step 1: Take the FFT
f = fft(x);

plot(f.*conj(f))
```




```
% Create a copy of the Fourier coefficients
complexf = 1i*f;

% Identify the indices for the positive and negative frequencies
% because they are rotated in different ways.
pos_freq_idx = 2:floor(n/2) + mod(n, 2);
neg_freq_idx = ceil(n/2) + 1 + ~mod(n, 2):n;

% Rotate the Fourier coefficients by computing i*A*sin(wt) component.
% Note! Positive frequencies are rotated counter-clockwise while
% negative frequencies are rotated clockwise.
f(pos_freq_idx) = f(pos_freq_idx) + -1i*complexf(pos_freq_idx);
f(neg_freq_idx) = f(neg_freq_idx) +  1i*complexf(neg_freq_idx);

plot(f.*conj(f))
```




```
hilbert_x = ifft(f);

plot(abs(hilbert_x))
```



Properties of the Hilbert transform

(1) A signal $x_r(t)$ and its Hilbert transform $x_i(t)$ have the same power density spectrum.

(2) A signal $x_r(t)$ and its Hilbert transform $x_i(t)$ have the same autocorrelation function

(3) A signal $x_r(t)$ and its Hilbert transform $x_i(t)$ are mutually orthogonal so we can write:

$$\int_{-\infty}^{\infty} x_r(t)x_i(t)\,\mathrm{dt} = 0$$

(4) If $x_i(t)$ is a Hilbert transform of a signal $x_r(t)$ then the Hilbert transform of signal $x_i(t)$ is $-x_r(t)$ i.e. the Hilbert transform of the Hilbert transform yields the negative signal:

if $H[x_r(t)] = x_i(t)$ then $H[x_i(t)] = -x_r(t)$