

Hilbert Transform

Table of Contents

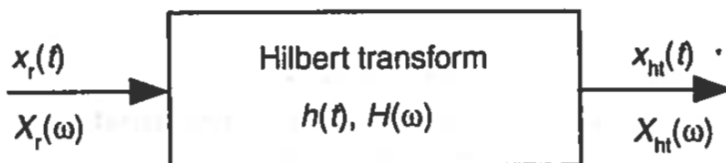
What is Hilbert Transform?	1
Why do we need it?	2
Real Signals as Complex Exponentials	2
Quarter Cycle Shift	3
Table of selected Hilbert transforms	4
Properties of the Hilbert transform	6
Envelope Extraction using Hilbert transform	8
FFT-based Hilbert transform	9
Is the Hilbert transform of a triangular wave signal is another triangular wave signal?	13
ADSI Problem 7.1: Study the hilbert function in MATLAB	15
ADSI Problem 7.2: Hilbert transform of a cosine signal	15
[✓] ADSI Problem 7.3: One-sided spectrum of analytical signals	15
[✓] ADSI Problem 7.4: Swept sine wave	17
[✓] 1. Design swept sine signal	18
[✓] 2. Compute instantaneous frequency	19
[?] ADSI Problem 7.5: Envelope Detection	20
[?] 1. Plot the signal and envelope on the same graph	20
[✓] 2. Describe how noise affects Hilbert transform's ability to recover the envelope	21
[✓] 3. Try with other two signals	23
[»] ADSI Problem 7.8: Amplitude modulated signal	25
[✓] 1. Plot 1 second of the signal and confirm that it is an amplitude modulated signal with a carrier frequency of 20 kHz	25
[»] 2. Use the procedure from Figure 9.8 in the note to frequency shift the signal to 15 kHz. Plot the spectrum of the signal at each step in the procedure	25
Functions	25

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.

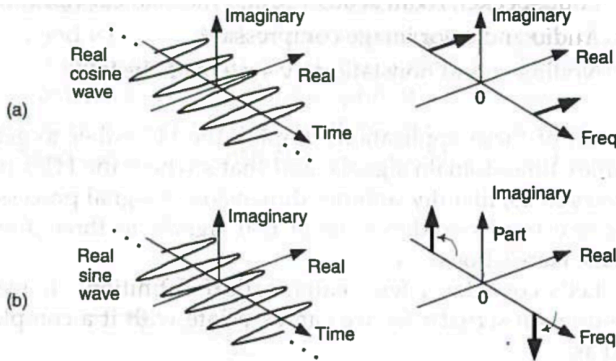


Figure 9-3 The Hilbert transform: (a) $\cos(\omega t)$; (b) its transform is $\sin(\omega t)$.

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 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

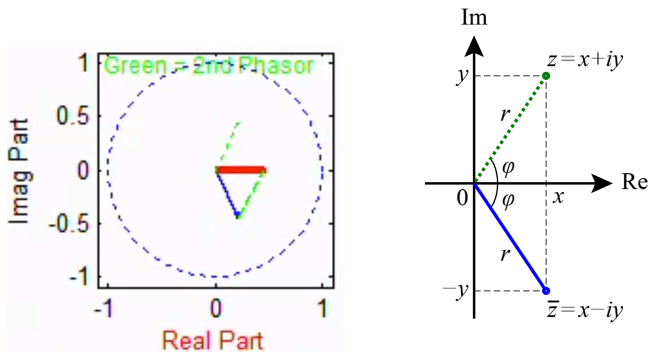
- $x_c(t)$ is known as an *analytic signal* because it has no negative-frequency spectral components. This means that $X_c(e^{j\omega})$ is zero over the negative frequency range. This is also known as a *one-sided spectrum*.
- $j x_i(t)$ is called the *phase-quadrature component*.

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

Real Signals as Complex Exponentials

A real-valued signal can be seen as a single frequency phasor (red) that is composed of the sum of two phasors:

- a positive frequency phasor (blue) rotates counter-clockwise
- a negative frequency phasor (green) rotates clockwise



If we project the sum of the two phasors in the real part, we get the red phasor. However, if we project the sum into the imaginary part, the two phasors cancel out one another and we get zero imaginary component.

This means that a real-valued signal has two complex exponentials (complex conjugate phasors) rotating in opposite directions. The real parts are in phase and sum together, the imaginary parts are opposite polarity, and cancel out, which leaves only a real sinusoid as a result. The negative and positive frequency components **are** both necessary to produce the real signal.

A real cosine signal $x_r(t) = A \cos(\omega_0 t + \phi)$ can be expressed in terms of two complex exponentials using Eulers formula:

$$x_r(t) = A \cos(\omega_0 t + \phi)$$

$$x_r(t) = A \left(\frac{e^{j(\omega t + \phi)} + e^{-j(\omega t + \phi)}}{2} \right)$$

$$x_r(t) = A \left(\frac{1}{2} e^{j(\omega t + \phi)} + \frac{1}{2} e^{-j(\omega t + \phi)} \right)$$

$$x_r(t) = A \left(\frac{1}{2} e^{j\phi} e^{j\omega t} + \frac{1}{2} e^{-j\phi} e^{-j\omega t} \right)$$

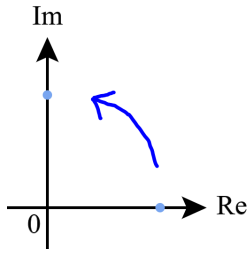
$$x_r(t) = \frac{1}{2} X e^{j\omega t} + \frac{1}{2} X^* e^{-j\omega t} \text{ where } X = A e^{j\phi} \text{ and } X^* = A e^{-j\phi}$$

where X^* is complex conjugate of the phasor X . This shows that a real-valued signal with frequency ω is actually composed for a positive frequency $+\omega$ complex exponential and a negative frequency $-\omega$ complex exponential.

Quarter Cycle Shift

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 $-\frac{\pi}{2}$.

For a continuous time signal $x_r(t)$, the Hilbert transform is defined as:

$$x_{ht}(t) = x_r(t) * \frac{1}{\pi t} \text{ where } * \text{ is the convolution operation}$$

We can see the Hilbert transform as a FIR filter with the impulse response $h(t) = \frac{1}{\pi t}$

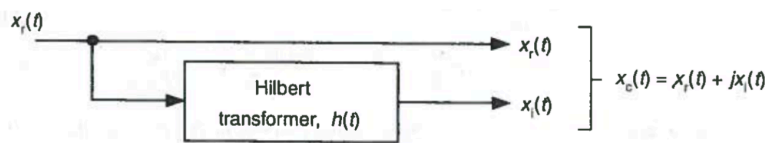


Figure 9-4 Functional relationship between the $x_c(t)$ and $x_r(t)$ signals.

Table of selected Hilbert transforms

In the following table, the frequency parameter ω is real.

Signal $u(t)$	Hilbert transform ^[fn 1] $H(u)(t)$
$\sin(\omega t)$ ^[fn 2]	$\operatorname{sgn}(\omega) \sin\left(\omega t - \frac{\pi}{2}\right) = -\operatorname{sgn}(\omega) \cos(\omega t)$
$\cos(\omega t)$ ^[fn 2]	$\operatorname{sgn}(\omega) \cos\left(\omega t - \frac{\pi}{2}\right) = \operatorname{sgn}(\omega) \sin(\omega t)$
$e^{i\omega t}$	$\operatorname{sgn}(\omega) e^{i\left(\omega t - \frac{\pi}{2}\right)} = -i \cdot \operatorname{sgn}(\omega) e^{i\omega t}$
$\frac{1}{t^2 + 1}$	$\frac{t}{t^2 + 1}$
e^{-t^2}	$2\pi^{-1/2} F(t)$ (see Dawson function)
Sinc function $\frac{\sin(t)}{t}$	$\frac{1 - \cos(t)}{t}$
Rectangular function $\operatorname{rect}(t) = \Pi(t) = \begin{cases} 0, & \text{if } t > \frac{1}{2} \\ \frac{1}{2}, & \text{if } t = \frac{1}{2} \\ 1, & \text{if } t < \frac{1}{2}. \end{cases}$	$\frac{1}{\pi} \ln \left \frac{t + \frac{1}{2}}{t - \frac{1}{2}} \right $
Dirac delta function $\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$	$\frac{1}{\pi t}$
Characteristic Function $\chi_{[a,b]}(t)$	$\frac{1}{\pi} \ln \left \frac{t - a}{t - b} \right $

Real signal	Hilbert transform
$a_1 g_1(t) + a_2 g_2(t); a_1, a_2 \in \mathbb{C}$	$a_1 \hat{g}_1(t) + a_2 \hat{g}_2(t)$
$h(t - t_0)$	$\hat{h}(t - t_0)$
$h(at); a \neq 0$	$\text{sgn}(a) \hat{h}(at)$
$\frac{d}{dt} h(t)$	$\frac{d}{dt} \hat{h}(t)$
$\delta(t)$	$\frac{1}{\pi t}$
e^{jt}	$-je^{jt}$
e^{-jt}	je^{-jt}
$\cos(t)$	$\sin(t)$
$\text{rect}(t)$	$\frac{1}{\pi} \ln (2t + 1)/(2t - 1) $
$\text{sinc}(t)$	$\frac{\pi t}{2} \text{sinc}^2(t/2) = \sin(\pi t/2) \text{sinc}(t/2)$
$1/(1 + t^2)$	$t/(1 + t^2)$

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:

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

(4) The effect of applying the Hilbert transform twice on a zero mean signal $x_r(t)$ is:

$$H[H[x_r(t)]] = -x_r(t)$$

For example, suppose $x_r(t) = \cos(t)$. Computing the Hilbert transform we get:

$$H[\cos(t)] = \cos\left(t - \frac{\pi}{2}\right)$$

Computing the Hilbert transform again, we get:

$$H\left[\cos\left(t - \frac{\pi}{2}\right)\right] = \cos\left(t - \frac{\pi}{2} - \frac{\pi}{2}\right)$$

$$H\left[\cos\left(t - \frac{\pi}{2}\right)\right] = \cos(t - \pi)$$

$$H\left[\cos\left(t - \frac{\pi}{2}\right)\right] = -\cos(t)$$

(5) The energy of a Hilbert transform $x_i(t)$ is the same as the energy of the original signal $x_r(t)$. Hilbert Transform does not change the magnitude of the input signal hence the energy is invariant

Assume that some zero-mean signal $x(k)$ with finite energy

$$\sum_{k=-\infty}^{\infty} x^2(k) = C$$

is Hilbert transformed to give $x_{HT}(k) = H[x(k)]$

Example:

$$x(t) \longrightarrow \boxed{h(t) = \frac{1}{\pi t}} \longrightarrow y(t)$$

- If $x(t) = 4\text{sinc}(2t)$, then evaluate $\int_{-\infty}^{\infty} |y(t)|^2 dt$.
- The integral is energy of $y(t)$.
- Also, $y(t)$ is Hilbert transform of $x(t)$.
- Hence, energy of $y(t) = \text{energy of } x(t)$.
- Energy of $x(t) = 4\text{sinc}(2t)$, is, $E_x = \frac{4^2}{2} = 8$.

(6) The inverse Hilbert transform is $H^{-1}[x_r(t)] = -H[x_r(t)]$:

$$H(\omega) = \begin{cases} -j & \omega > 0 \\ 0 & \omega = 0 \\ j & \omega < 0 \end{cases} \Rightarrow H_{inv}(\omega) = -H(\omega) = \begin{cases} j & \omega > 0 \\ 0 & \omega = 0 \\ -j & \omega < 0 \end{cases}$$

Note: A DC term can't be recreated

Envelope Extraction using Hilbert transform

```
clear variables;
```

The book states: If a real sinewave $x_r(t)$ is amplitude modulated so its envelope contains information, then we can measure the instantaneous envelope $E(t)$ values from an analytic version of the signal using the Equation 9-4:

$$E(t) = |x_c(t)| = \sqrt{x_r(t)^2 + x_i(t)^2}. \quad (9-4)$$

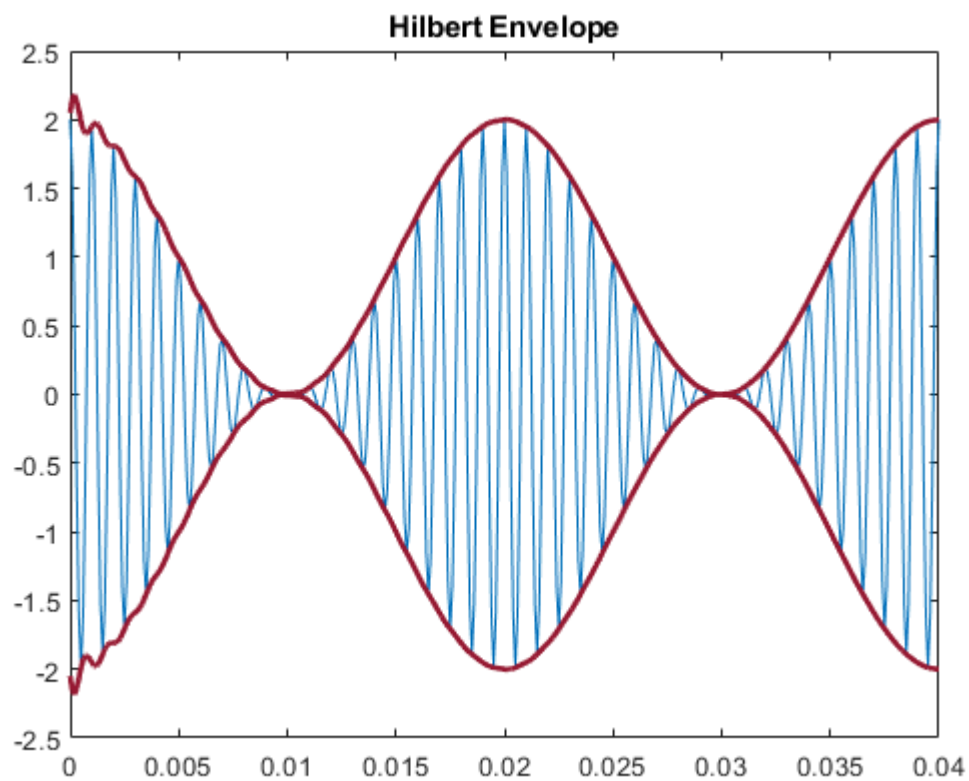
Equation 9-4 states that the envelope of the signal is equal to magnitude of $x_c(t)$.

In MATLAB the complex magnitude of a signal can be computed with the `abs` function.

```
t = 0:1e-4:0.1;
x = (1+cos(2*pi*50*t)).*cos(2*pi*1000*t);

y = hilbert(x);
env = abs(y);
plot_param = {'Color', [0.6 0.1 0.2], 'Linewidth', 2};

plot(t,x)
hold on
plot(t,[-1;1]*env,plot_param{:})
hold off
xlim([0 0.04])
title('Hilbert Envelope')
```

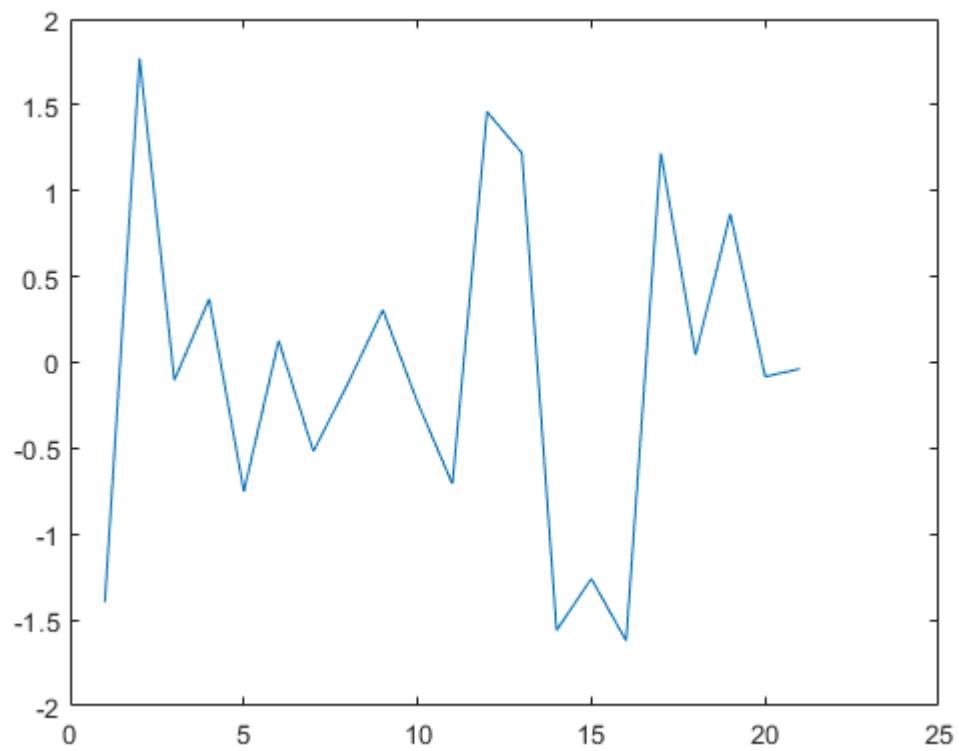
FFT-based Hilbert transform

```
clear variables;
```

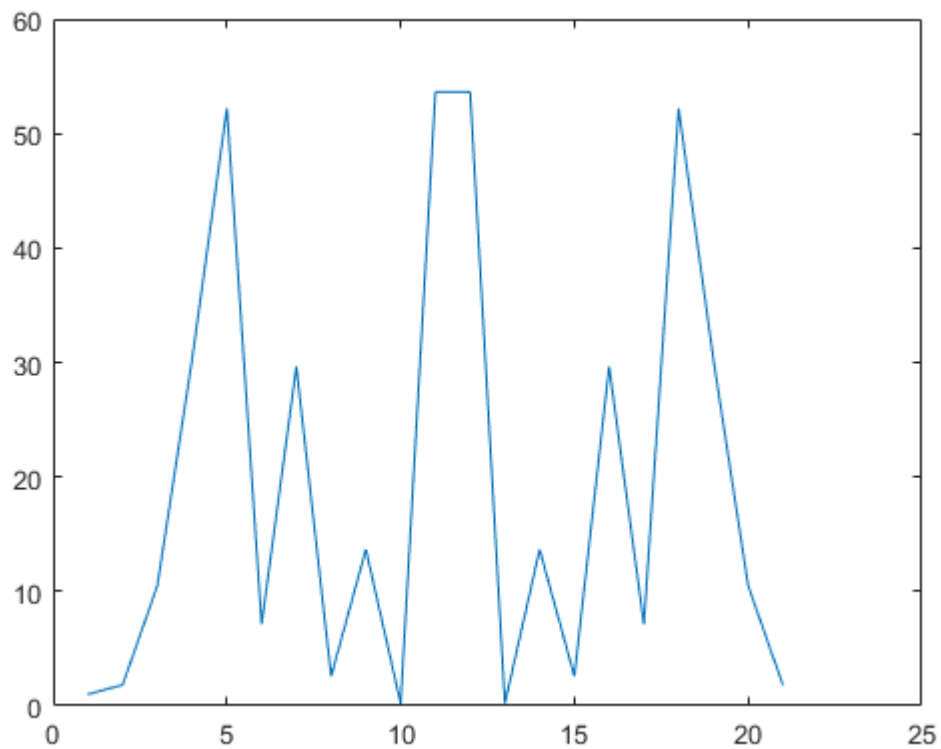
The FFT-based Hilbert transform can be implemented as three steps:

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

```
n = 21;  
  
% Generate a signal  
x = randn(n, 1);  
plot(x)
```



```
% Step 1: Take the FFT  
f = fft(x);  
  
plot(f.*conj(f))
```

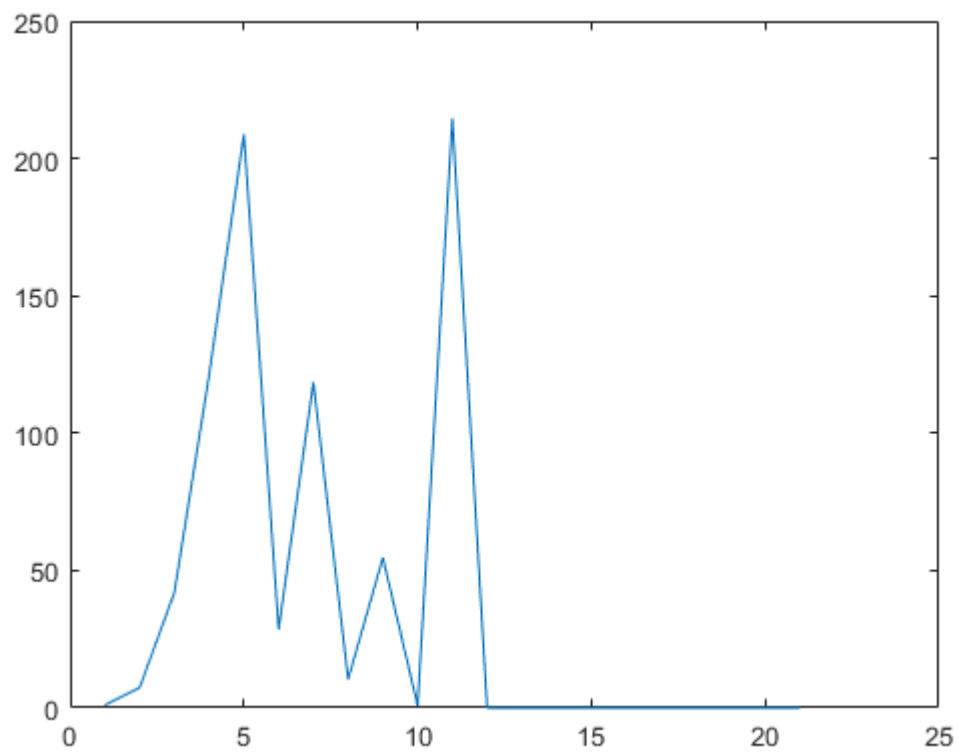


```
% Create a copy of the Fourier coefficients
complexf = 1i*f;

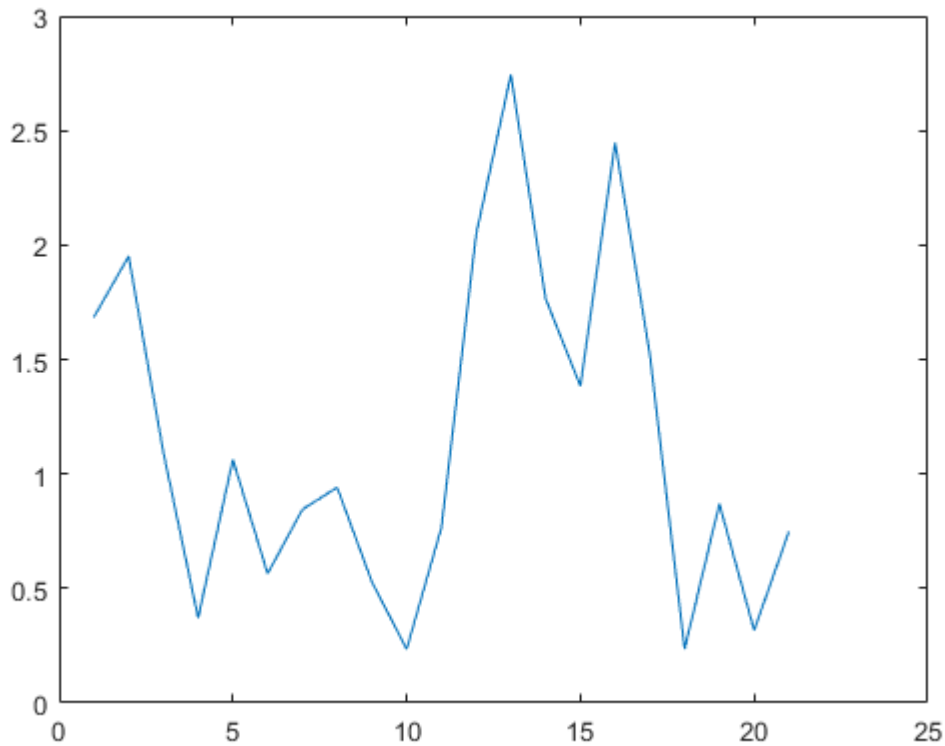
% Identify the indices for the positive and negative frequencies
% because they are rotated in different ways.
pos_freq_idx = 2:floor(n/2) + mod(n, 2);
neg_freq_idx = ceil(n/2) + 1 + ~mod(n, 2):n;

% 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))
```



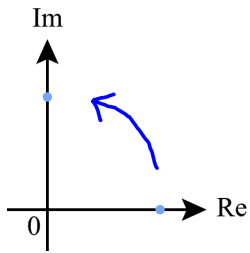
Is the Hilbert transform of a triangular wave signal is another triangular wave signal?

2. The Hilbert transform of a triangular wave signal is another triangular wave signal.

Answer: FALSE.

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 $-\frac{\pi}{2}$.

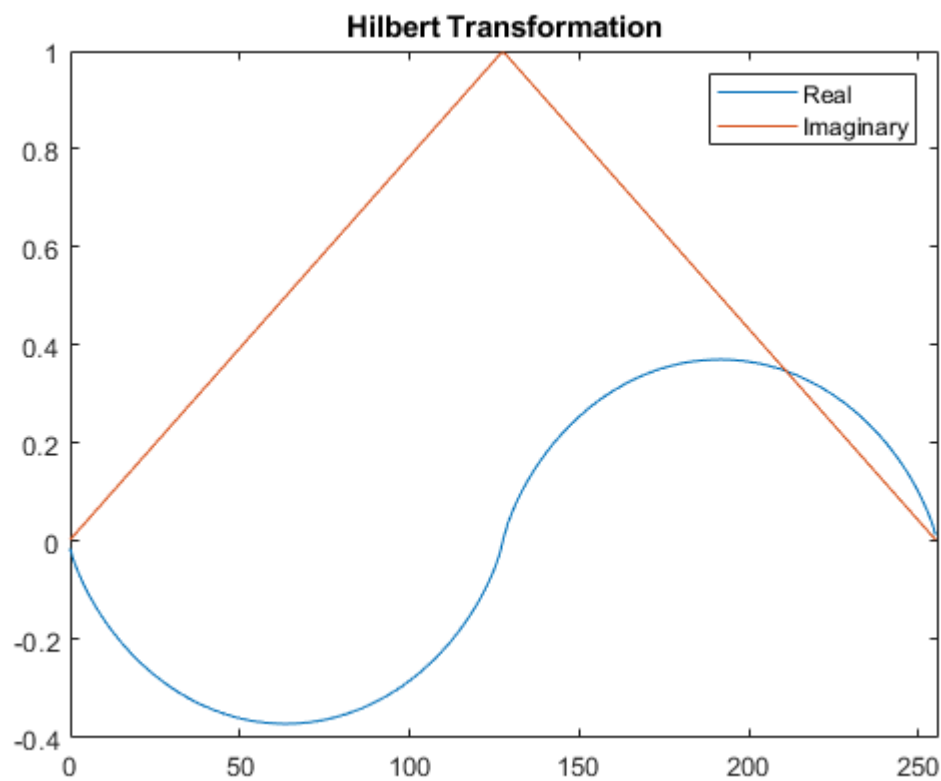
The Hilbert transform of $\cos(\theta)$ is $\sin(\theta)$ i.e., a quarter-cycle shift.



A triangular wave signal can be thought of being composed of multiple frequency components. If each of these is shifted by 90 degrees then the result is not a triangular wave signal.

We can experiment with it.

```
N = 256;
n = 0:N-1;
xr = triang(256);
x = hilbert(xr);
plot(n, imag(x), n, real(x))
legend('Real', 'Imaginary')
title('Hilbert Transformation')
xlim([0, N])
```



ADSI Problem 7.1: Study the hilbert function in MATLAB

A tremendous amount of functionality is found in MATLABs toolboxes and Hilbert transforms are no exception. Type `doc hilbert` in MATLAB and read the documentation.

```
doc hilbert
```

ADSI Problem 7.2: Hilbert transform of a cosine signal

A continuous time signal is given by $x(t) = 4 + 3\cos(\omega t)$. What is the Hilbert transform of this signal?

Signal $u(t)$	Hilbert transform ^[1] $H(u)(t)$	Real signal	Hilbert transform
$\sin(\omega t)$ ^[1]	$\text{sgn}(\omega) \sin(\omega t - \frac{\pi}{2}) = -\text{sgn}(\omega) \cos(\omega t)$	$a_1 g_1(t) + a_2 g_2(t); a_1, a_2 \in \mathbb{C}$	$a_1 \hat{g}_1(t) + a_2 \hat{g}_2(t)$
$\cos(\omega t)$ ^[1]	$\text{sgn}(\omega) \cos(\omega t - \frac{\pi}{2}) = \text{sgn}(\omega) \sin(\omega t)$	$h(t - t_0)$	$\hat{h}(t - t_0)$
$e^{i\omega t}$	$\text{sgn}(\omega) e^{i(\omega t - \frac{\pi}{2})} = -i \cdot \text{sgn}(\omega) e^{i\omega t}$	$h(at); a \neq 0$	$\text{sgn}(a) \hat{h}(at)$
$\frac{1}{t^2 + 1}$	$\frac{t}{t^2 + 1}$	$\frac{d}{dt} h(t)$	$\frac{d}{dt} \hat{h}(t)$
e^{-t^2}	$2\pi^{-1/2} F(t)$ (see Dawson function)	$\delta(t)$	$\frac{1}{\pi t}$
Sinc function $\frac{\sin(t)}{t}$	$\frac{1 - \cos(t)}{t}$	e^{jt}	$-j e^{jt}$
Rectangular function $\text{rect}(t) = \Pi(t) = \begin{cases} 0, & \text{if } t > \frac{1}{2} \\ \frac{1}{2}, & \text{if } t = \frac{1}{2} \\ 1, & \text{if } t < \frac{1}{2} \end{cases}$	$\frac{1}{\pi} \ln \left \frac{t + \frac{1}{2}}{t - \frac{1}{2}} \right $	e^{-jt}	$j e^{-jt}$
Dirac delta function $\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$	$\frac{1}{\pi t}$	$\cos(t)$	$\sin(t)$
Characteristic Function $\chi_{(a,b)}(t)$	$\frac{1}{\pi} \ln \left \frac{t-a}{t-b} \right $	$\text{rect}(t)$	$\frac{1}{\pi} \ln (2t+1)/(2t-1) $
		$\text{sinc}(t)$	$\frac{\pi t}{2} \text{sinc}^2(t/2) = \sin(\pi t/2) \text{sinc}(t/2)$
		$1/(1+t^2)$	$t/(1+t^2)$

$$\begin{aligned}
 H[4 + 3 \cos(\omega t)] &= H[4] + 3H[\cos(\omega t)] \\
 &= 0 + 3\text{sgn}(\omega)\sin(\omega t)
 \end{aligned}$$

[✓] ADSI Problem 7.3: One-sided spectrum of analytical signals

```
clear variables;
```

Consider the signal

$$x(n) = e^{-0.001(n-255)^2} \cos(1.8n) \quad 0 \leq n \leq 511.$$

Calculate the analytical signal in MATLAB and check that the spectrum is one-sided as expected.

Recall that we start with a real-valued signal $x_r(t)$ and we want we need to represent the signal as a complex sinusoid:

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

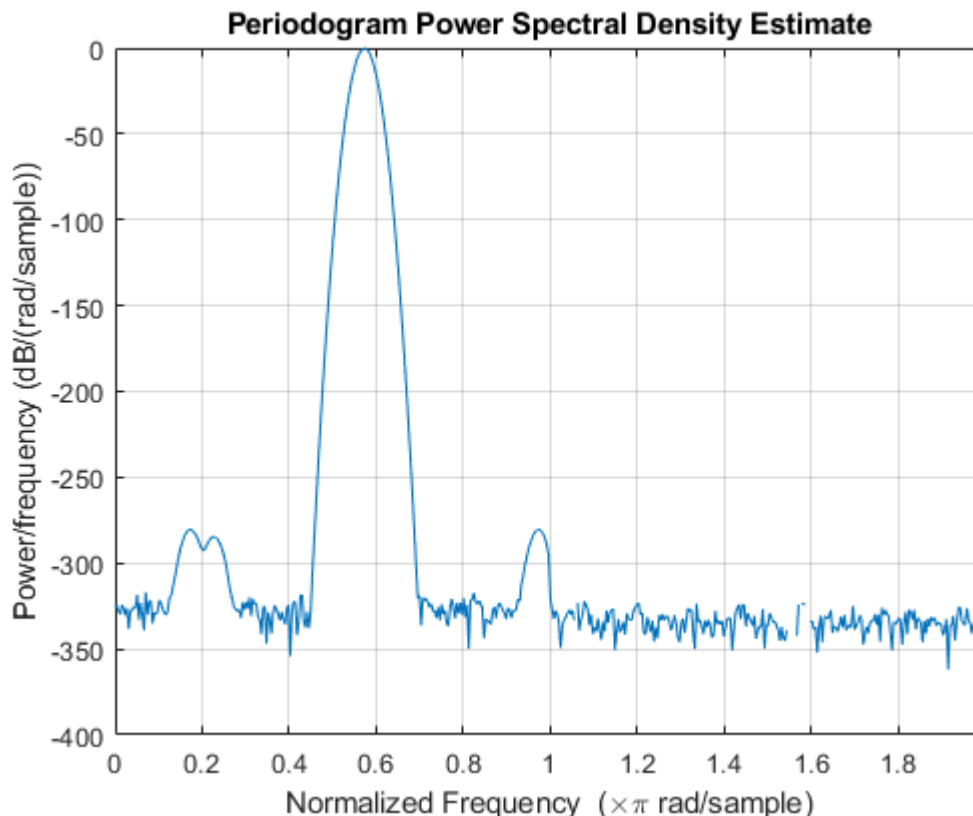
$x_c(t)$ is known as an **analytic signal** because it has no negative-frequency spectral components.

To show that the spectrum is one-sided, we have to show that the frequency response $X_c(e^{j\omega})$ is zero over the negative frequency range.

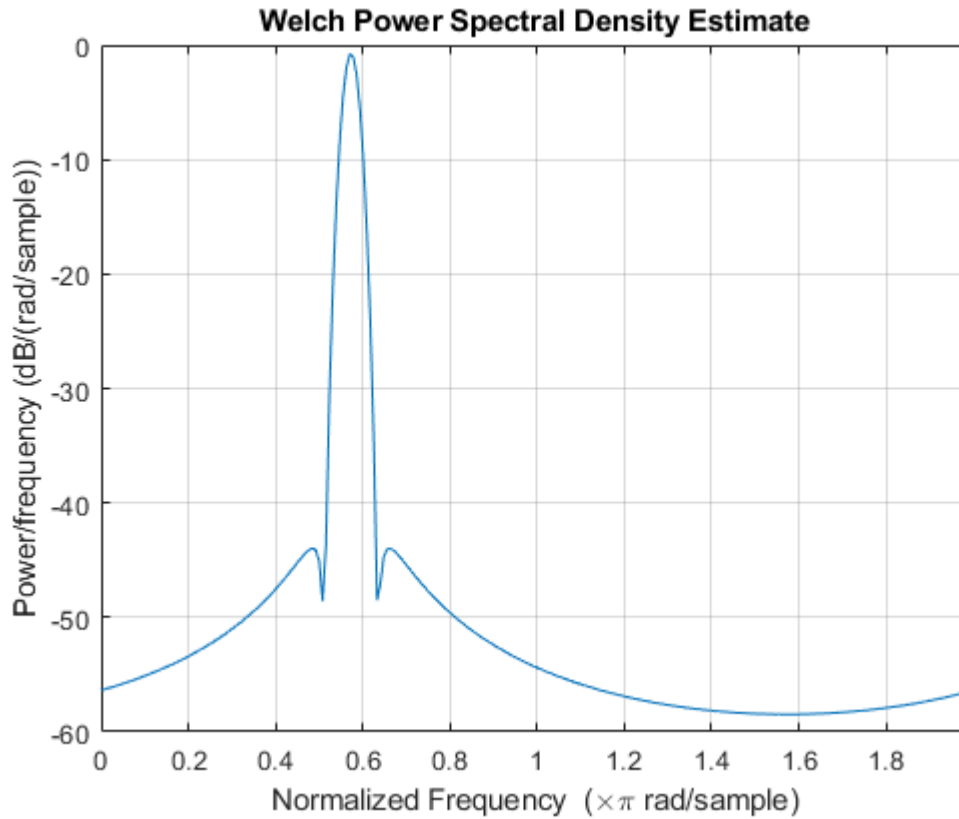
We can do this by estimating the power spectrum density (PSD) of the analytical signal using the periodogram.

In the MATLAB plot, the normalised **positive frequency range** is [0, 1] whereas the **negative frequency range** is between 1 and 2. Notice there is no power spikes in the negative frequency range, only noise.

```
n = (0:511)';  
x = exp(-0.001.*(n-225).^2).*cos(1.8*n);  
  
% Compute the analytic signal of the real signal x(n)  
y = hilbert(x);  
  
% Compute and plot the PSD estimate  
periodogram(y)
```




```
% PSD estimate using the Welch method
pwelch(y)
```



✓ ADSI Problem 7.4: Swept sine wave

In (audio) measurements an exponentially swept sine wave is often used to measure transfer functions and nonlinear distortion, see e.g. Novák et al. “Nonlinear System Identification Using Exponential Swept-Sine Signal” *IEEE Trans. Instrum. Measure.* **59**, 2220-2229, (2010) or “Synchronized Swept-Sine: Theory, Application and Implementation”. *J. Audio Eng. Soc.* **63**, 787-798, (2015). In the continuous time domain an exponential swept sine wave is given by

$$s(t) = \sin \left(2\pi f_1 L \left[\exp \left(\frac{t}{L} \right) - 1 \right] \right)$$

where

$$L = \frac{T}{\ln \left(\frac{f_2}{f_1} \right)}$$

T is the duration of the sweep, f_1 and f_2 are the start and stopping frequencies of the sweep.

```
clear variables;
```

[✓] 1. Design swept sine signal

1. Design a 5 second long swept sine sampled at 48 kHz. The sine should sweep from 50 Hz to 5000 Hz. Play the sound on your pc and check that it performs as expected.

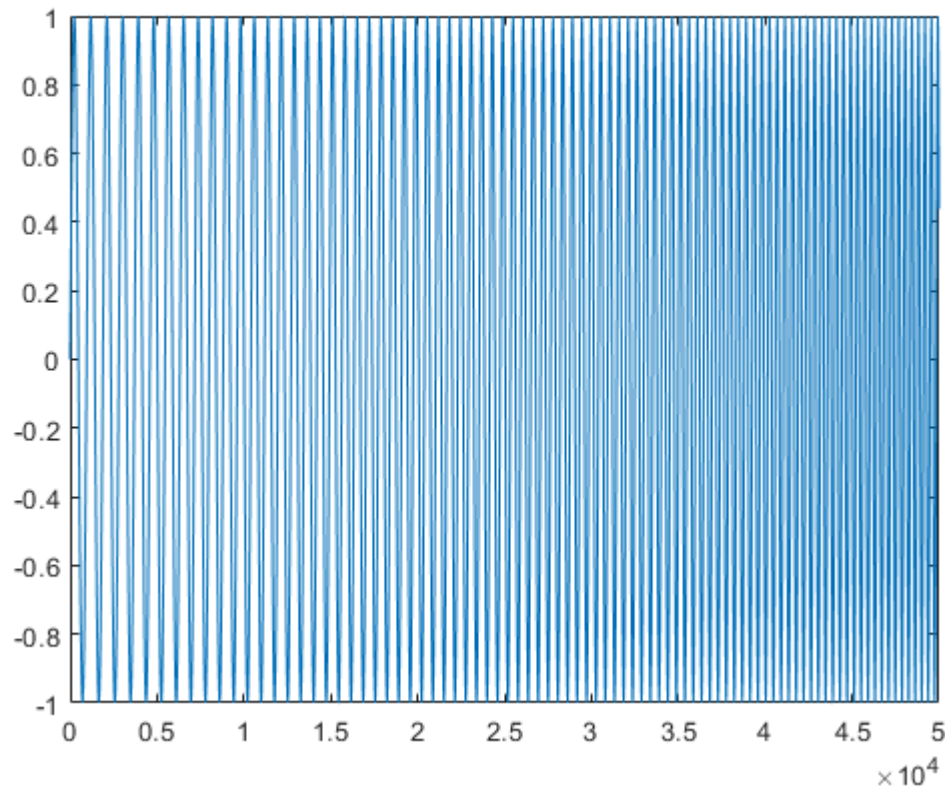
```
Ts = 48000; % Sampling rate
f1 = 50;    % Starting 1/frequency
f2 = 5000;  % Stopping 1/frequency
T = 5*Ts;   % Duration of the sweep
L = T / log(f2/f1);

n = 0:T-1;
s = sin(2*pi * f1 * L * (exp(n/L) - 1) / Ts);

% Write to file
% audiowrite('swept_sine.wav', s, Ts);

% Play sound
soundsc(s, Ts);

% Plot it
plot(n, s)
xlim([0, 50000]);
```



[✓] 2. Compute instantaneous frequency

2. Use equation 9.6 $F(t) = d/dt \tan^{-1} x_i(t)/x_r(t)$ to calculate the instantaneous frequency of the swept sine signal and compare with the real frequency.

$$F(t) = \frac{d}{dt} \phi(t) = \frac{d}{dt} \tan^{-1} \left(\frac{x_i(t)}{x_r(t)} \right). \quad (9-6)$$

```
s_c = hilbert(s); % Compute the analytic signal

% The instantaneous phase can be computed using MATLAB's angle function
inst_phase = angle(s_c);
plot(n, inst_phase, n, unwrap(inst_phase))
legend('Instantaneous phase', 'Unwrapped phase')
xlim([0, 2000])
% Approximate the derivative of the unwrapped instantaneous phase
F = diff(unwrap(inst_phase));

% Plot it
plot(n(2:end)/Ts, F*10)
ylim([0, 6])
```

```

grid on
ylabel('Frequency (kHz)')
xlabel('Time (seconds)')
title('Instantaneous Frequency Estimate')
% Estimate the spectrogram of the waveform using the pspectrum function.
pspectrum(s, Ts, 'spectrogram')

% The instfreq function computes and displays the instantaneous frequency in one step.
instfreq(s, Ts, 'Method', 'hilbert')

```

[?] ADSI Problem 7.5: Envelope Detection

The textbook states that the Hilbert transform can be used to find the envelope of a signal. The validity of this statement is investigated in this problem.

Consider an exponentially decaying oscillating signal given by

$$x(n) = 4e^{-0.01n}\cos(\pi n)$$

```

clear variables;
clf

```

[?] 1. Plot the signal and envelope on the same graph

Plot this signal and use equation 9.4 to plot the envelope on the same graph. Compare the result with the expected outcome.

The books states: If a real sinewave $x_r(t)$ is amplitude modulated so its envelope contains information, then we can measure the instantaneous envelope $E(t)$ values from an analytic version of the signal using the Equation 9-4:

$$E(t) = |x_c(t)| = \sqrt{x_r(t)^2 + x_i(t)^2}. \quad (9-4)$$

Equation 9-4 states that the envelope of the signal is equal to magnitude of $x_c(t)$.

In MATLAB the complex magnitude of a signal can be computed with the abs function.

```

n = (0:511);
x_r = 4*exp(-0.001.*n).*cos(pi*n);

% Compute the analytic signal
x_c = hilbert(x_r);

% Find the envelope by computing the magnitude of the signal
x_envelope = abs(x_c);

% Plot the real signal

```

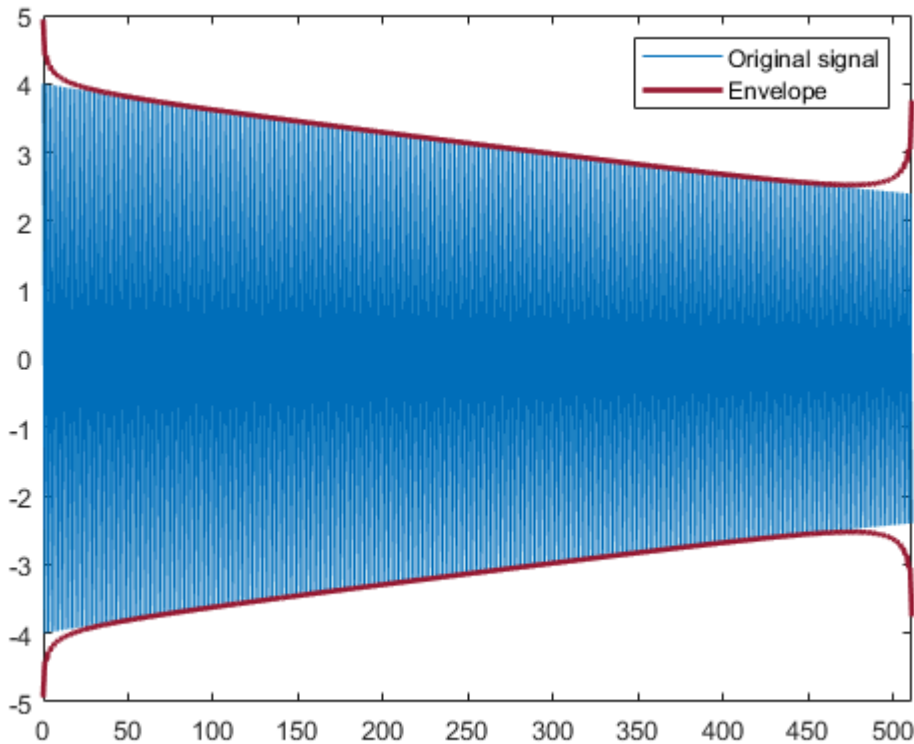
```

plot(n, x_r)
hold on

% Plot the envelope
plot(n, [-1;1]*x_envelope, 'Color', [0.6 0.1 0.2], 'Linewidth', 2)
hold off

legend('Original signal', 'Envelope')
xlim([0, max(n)])

```



[✓] 2. Describe how noise affects Hilbert transform's ability to recover the envelope

In real life noise is always present in measurements.

2. Add white Gaussian noise $WGN \sim (0,1)$ to the signal and repeat. Describe the influence of the noise in the ability of the Hilbert transform to recover the envelope.

```

N = 512;
n = (0:N-1);
w = randn(N, 1)';
x_r = 4*exp(-0.001.*n).*cos(pi*n) + w;

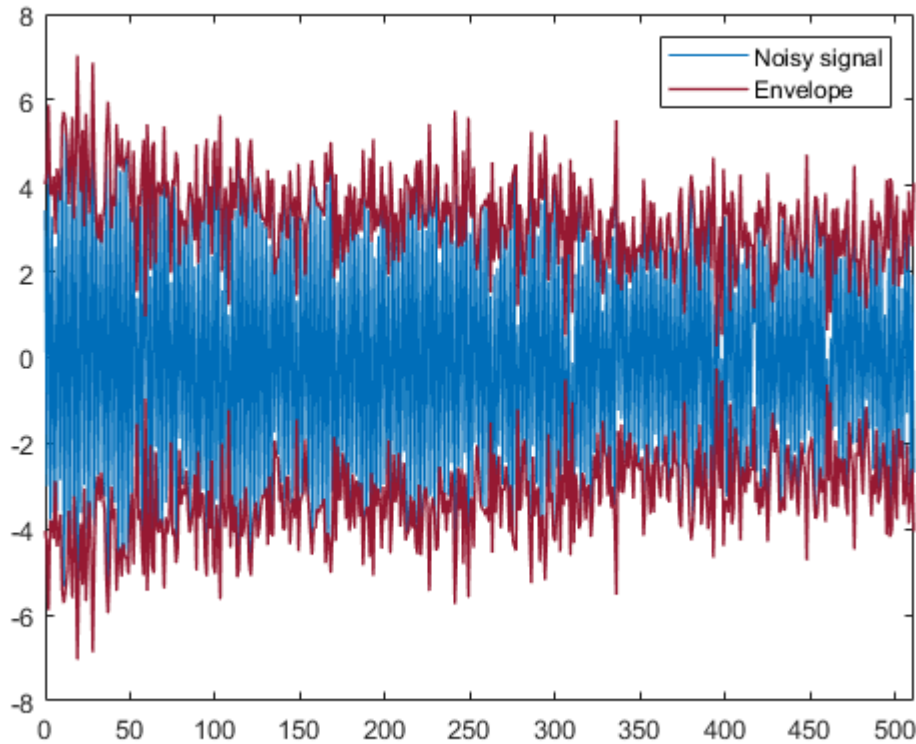
x_c = hilbert(x_r);
x_envelope = abs(x_c);

```

```

plot(n, x_r)
hold on
plot(n, [-1;1]*x_envelope, 'Color', [0.6 0.1 0.2], 'Linewidth',1)
hold off
legend('Noisy signal', 'Envelope')
xlim([0, max(n)])

```



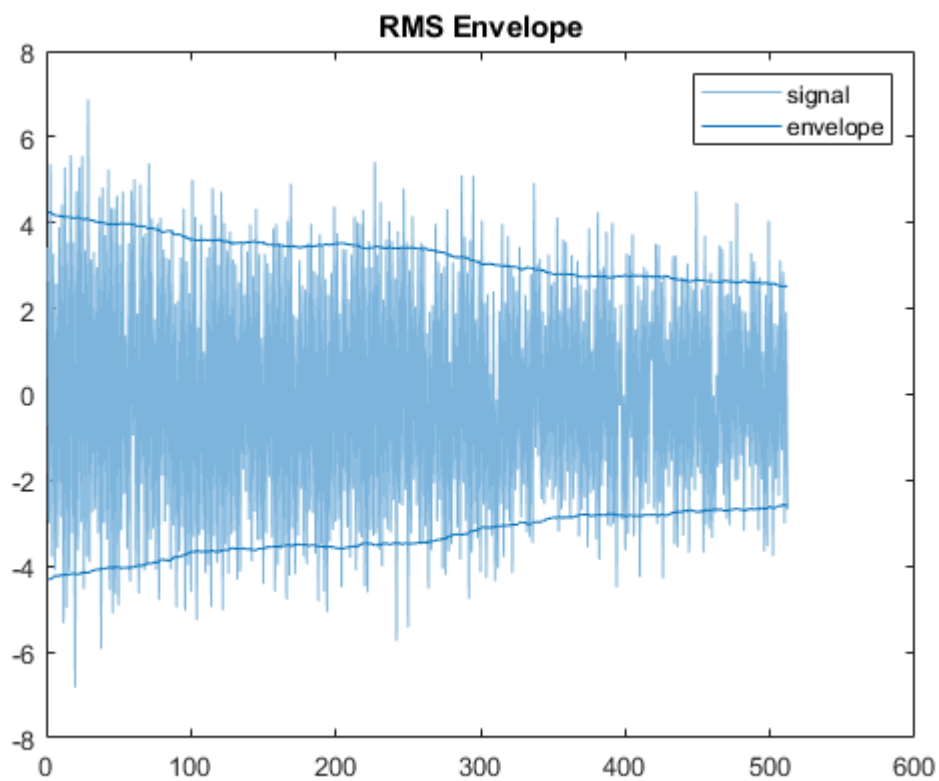
Hilbert transform is sensitive to noise. We observe spurious peaks in the envelope due to noise.

We can generate a moving Root-Mean-Square (RMS) average envelopes by using a sliding window of size W . Using a large window size smooths out the envelope.

```

envelope(x_r, 100, 'rms');

```

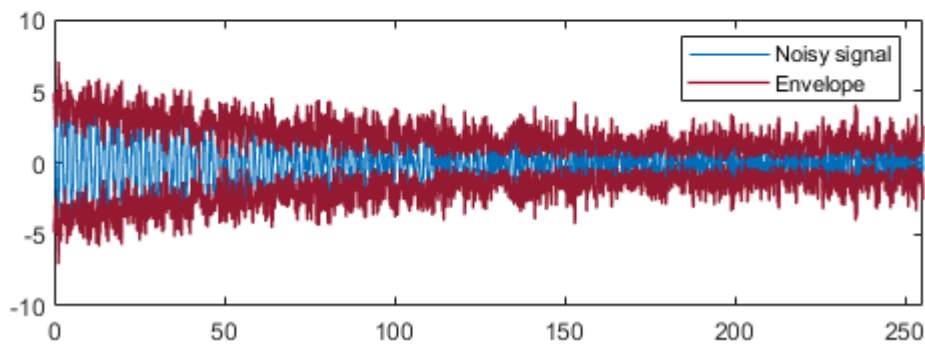
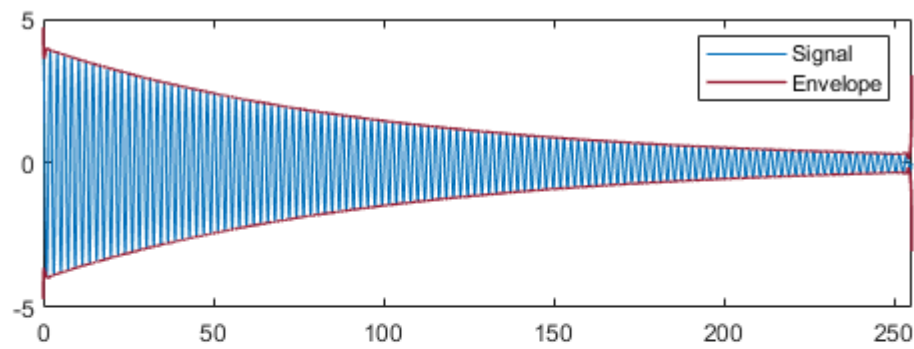


[✓] 3. Try with other two signals

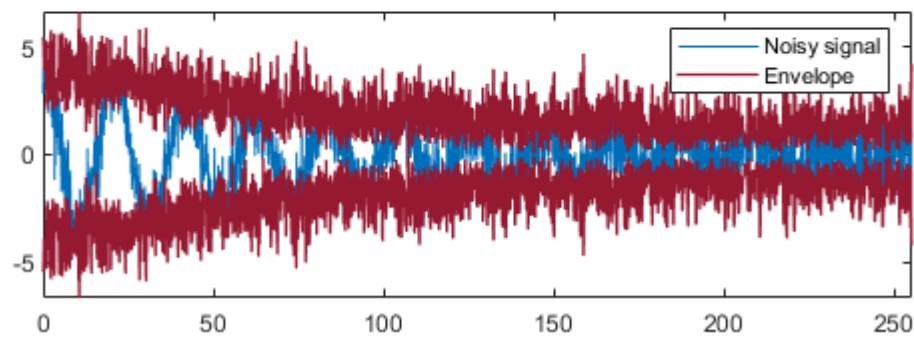
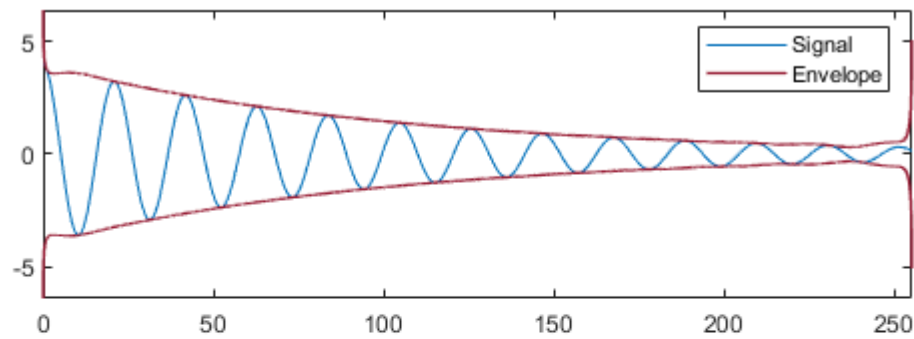
3. Repeat the above two questions for the signals $x_1(n) = 4e^{-0.01n} \cos(3n)$ and $x_2(n) = 4e^{-0.01n} \cos(0.3n)$.

```
clear variables;

N = 256;
n = 0:0.1:N-1;
x = 4*exp(-0.01.*n).*cos(3*n);
plot_signal_and_envelope(n, x)
```



```
x = 4*exp(-0.01.*n).*cos(0.3*n);
plot_signal_and_envelope(n, x)
```



[»] ADSI Problem 7.8: Amplitude modulated signal

Consider an amplitude modulated signal given by

$$x(t) = (1 + 0.2 \cos(2\pi 37t)) \cos(2\pi * 20000t)$$

```
clear variables;
```

[✓] 1. Plot 1 second of the signal and confirm that it is an amplitude modulated signal with a carrier frequency of 20 kHz.

The discrete-time signal $x[n]$ of continuous-time signal $x(t)$ can be represented as:

$$x[n] = s(n \cdot T_s)$$

where

- n is an integer denoting the sample index
- T_s is the sampling period

```
Ts = 8000; % Sampling rate  
n = 0:200;  
x = 1 + 0.2.*cos(2*pi*37*n/Ts).*cos(2*pi*20000*n/Ts);  
plot(n, x)
```

[»] 2. Use the procedure from Figure 9.8 in the note to frequency shift the signal to 15 kHz. Plot the spectrum of the signal at each step in the procedure.

Functions

```
function plot_signal_and_envelope(n, x)  
w = randn(numel(n), 1)';  
x_noisy = x + w;  
  
x_c = hilbert(x);  
x_envelope = abs(x_c);
```

```

x_noisy_c = hilbert(x_noisy);
x_noisy_envelope = abs(x_noisy_c);

clf
figure
ax1 = subplot(2,1,1);
plot(n, x)
hold on
plot(n, [-1;1]*x_envelope, 'Color', [0.6 0.1 0.2], 'Linewidth',1)
hold off
legend('Signal', 'Envelope')
xlim([0, max(n)])

ax2 = subplot(2,1,2);
plot(n, x_noisy)
hold on
plot(n, [-1;1]*x_noisy_envelope, 'Color', [0.6 0.1 0.2], 'Linewidth',1)
hold off
legend('Noisy signal', 'Envelope')
xlim([0, max(n)])
clear figure;
figure
end

```