

1 Analog and Discrete-Time Sinusoids

Properties of Analog and Discrete-Time Sinusoids

	$x_a(t) = A \cos(\Omega t + \theta) \quad -\infty < t < \infty$ $\Omega = 2\pi F, \quad [\Omega] = \text{radians/second}, \quad [F] = \text{cycles/second or Hertz}$
Periodicity	$x_a(t)$ is periodic for $-\infty < \Omega < \infty$
Uniqueness	$\cos(\Omega_1 t) = \cos(\Omega_2 t)$ for all t if and only if $\Omega_1 = \Omega_2$
Frequency range	Unlimited frequency range: Increasing Ω increases the oscillations
	$x(n) = A \cos(\omega n + \theta) \quad \dots, -2, -1, 0, 1, 2, \dots$ $\omega = 2\pi f, \quad [\omega] = \text{radians/sample}, \quad [f] = \text{cycles/sample}$
Periodicity	$x(n)$ is periodic if and only if $f = \frac{k}{N}$ (rational)
Uniqueness	$\cos(\omega_1 n) = \cos(\omega_2 n)$ for all n if and only if $\omega_2 = \omega_1 + 2\pi k$
Frequency range	Limited frequency range: Highest rate of oscillation is $\omega = \pm\pi$ or $f = \pm\frac{1}{2}$

Definition: The *Fundamental period* is the smallest integer N that satisfies $x(n + N) = x(n)$.

Example: Calculate the fundamental period of $x(n) = \cos\left(2\pi\frac{12}{36}n\right)$.

Solution: Express f as a ratio of relatively prime integers.

$$f = \frac{12}{36} = \frac{1}{3} \Rightarrow N = 3 \tag{1}$$

Example: Calculate the fundamental period of $x(n) = \cos\left(2\pi\frac{13}{36}n\right)$.

Solution:

$$f = \frac{13}{36} \Rightarrow N = 36 \tag{2}$$

Principle: Small changes in the frequency can lead to large changes in the fundamental period.

Example: (Uniqueness of sinusoids)

$$x(n) = \cos\left(2\pi\frac{4}{3}n\right) \tag{3}$$

$$= \cos\left(2\pi\left(1 + \frac{1}{3}\right)n\right) \tag{4}$$

$$= \cos\left(2\pi n + 2\pi\frac{1}{3}n\right) \tag{5}$$

$$= \cos\left(2\pi\frac{1}{3}n\right) \tag{6}$$

Conclusion: In general, discrete-time sinusoids are not unique.

$$\cos(2\pi(f_0 + k)n) = \cos(2\pi f_0 n) \quad \text{for all integers } k, n \tag{7}$$

$$\cos((\omega_0 + 2\pi k)n) = \cos(\omega_0 n) \quad \text{for all integers } k, n \tag{8}$$

Example: (Limited frequency range) Consider a discrete-time sinusoid with frequency $\omega = \pi + \epsilon$ that is just slightly larger than π .

$$\cos((\pi + \epsilon)n) = \cos((\pi + \epsilon)n - 2\pi n) \tag{9}$$

$$= \cos((-\pi + \epsilon)n) \tag{10}$$

$$= \cos(-(\pi - \epsilon)n) \tag{11}$$

$$= \cos((\pi - \epsilon)n) \tag{12}$$

Conclusion: In discrete-time, a cosinusoid with frequency $\pi + \epsilon$ is equal to a cosinusoid with frequency $\pi - \epsilon$. Therefore, $\pi + \epsilon$ is a lower frequency than π . Now, let $\epsilon \rightarrow 0$ and we see that π is the highest frequency in discrete-time. Since, $\omega = \pi$ radians per sample is the same as $f = \frac{1}{2}$ cycles per sample we have the bounds,

$$-\pi \leq \omega \leq \pi \quad -\frac{1}{2} \leq f \leq \frac{1}{2}. \quad (13)$$

2 Sampling

So far we have only compared the properties of analog and discrete-time sinusoids. Now, let's consider sampling analog sinusoids to create discrete-time sinusoids.

Let's assume we use uniform sampling, which means that $x(n) = x_a(t)$ when $t = T_s n$ where T_s is called the sample period. Note that $x(n)$ is only defined for n in the integers. The sample frequency or sample rate is defined to be $F_s = \frac{1}{T_s}$. Consider sampling a sinusoidal signal.

$$x_a(t) = \cos(2\pi Ft) \quad (14)$$

$$x(n) = x_a(T_s n) = \cos(2\pi F T_s n) \quad (15)$$

$$= \cos\left(2\pi \frac{F}{F_s} n\right) \quad (16)$$

The discrete-time frequency of the sampled sinusoid is $f = \frac{F}{F_s}$. The highest frequency represented in discrete-time is $\frac{1}{2}$ cycles per sample. Hence, if we want to avoid aliasing we must have (assuming $F > 0$)

$$f = \frac{F}{F_s} < \frac{1}{2} \quad \Rightarrow \quad F < \frac{F_s}{2} = \frac{1}{2T_s} \quad \text{or} \quad \Omega < \pi F_s = \frac{\pi}{T_s} \quad (17)$$

In general, an analog sinusoid with frequency F will alias to a discrete-time sinusoid with frequency f given by

$$f = \frac{F}{F_s} - \left\lfloor \frac{F}{F_s} + \frac{1}{2} \right\rfloor \quad (18)$$

3 Reconstruction

Sampling Theorem: If the highest frequency contained in an analog signal $x_a(t)$ is F_{max} and the signal is sampled at a rate $F_s = \frac{1}{T_s} > 2F_{max}$, then $x_a(t)$ can be exactly recovered from its sample values using the interpolation function

$$g(t) = \frac{\sin(2\pi F_s t)}{2\pi F_s t} \quad (19)$$

Thus $x_a(t)$ may be expressed as

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT_s) g(t - nT_s) \quad (20)$$

Proof: Assume that $F_s > 2F_{max}$. This means that

$$X_a(j\Omega) = 0 \quad \text{for} \quad |\Omega| > \frac{\pi}{T_s} \quad (21)$$

The samples $x_a(nT_s)$ can be extracted by modulating $x_a(t)$ by the periodic impulse train $p(t)$ given by

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \xleftrightarrow{\mathcal{F}} P(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta\left(\Omega - \frac{2\pi k}{T_s}\right) \quad (22)$$

Let $x_d(t) = x_a(t)p(t)$ be the output of the modulator.

$$x_d(t) = \sum_{n=-\infty}^{\infty} x_a(nT_s) \delta(t - nT_s) \xleftrightarrow{\mathcal{F}} X_d(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_a\left(j\Omega - j\frac{2\pi k}{T_s}\right) \quad (23)$$

Note that $x_d(t)$ is zero except at the sampling instants nT_s . The k^{th} sample value can be extracted by integrating $x_d(t)$ from $(k - \frac{1}{2})T_s$ to $(k + \frac{1}{2})T_s$. Notice that the spectrum of $X_d(j\Omega)$ is the sum of shifted replicas of $X_a(j\Omega)$. If $x_a(t)$ is bandlimited as in (21), then these shifted replicas do not overlap and $x_a(t)$ can be recovered by an ideal low-pass filter. This is the approach that we take next. However, if $x_a(t)$ violates (21) by having signal content above $\frac{\pi}{T_s}$, then these high frequency regions overlap when $X_a(j\Omega)$ is shifted and summed to produce $X_d(j\Omega)$. This overlapping in the frequency-domain is called aliasing. Aliasing is an irreversible effect.

Define an interpolation filter $g(t)$ by

$$g(t) = \frac{\sin(2\pi F_s t)}{2\pi F_s t} \xleftrightarrow{\mathcal{F}} G(j\Omega) = \begin{cases} T_s & \text{for } |\Omega| < \frac{\pi}{T_s}, \\ 0 & \text{for } |\Omega| \geq \frac{\pi}{T_s} \end{cases} \quad (24)$$

Due to (24) and (21) we have

$$X_a(j\Omega) = G(j\Omega)X_d(j\Omega) \quad (25)$$

In the time-domain, $x_a(t)$ is given by the following convolution

$$x_a(t) = \int_{-\infty}^{\infty} x_d(\tau)g(t - \tau)d\tau \quad (26)$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_a(nT_s)\delta(\tau - nT_s)g(t - \tau)d\tau \quad (27)$$

$$= \sum_{n=-\infty}^{\infty} x_a(nT_s) \int_{-\infty}^{\infty} \delta(\tau - nT_s)g(t - \tau)d\tau \quad (28)$$

$$= \sum_{n=-\infty}^{\infty} x_a(nT_s)g(t - nT_s) \quad (29)$$

4 Simulation of Continuous-time Sinusoidal Signals in Discrete-time Using Matlab

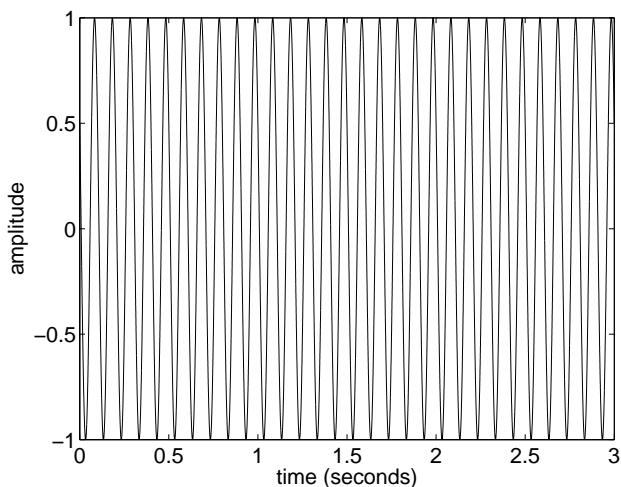
4.1 Time Domain Simulation

Suppose you want to simulate the analog signal $c(t) = \cos(\Omega_c t + \theta_c) = \cos(2\pi F_c t + \theta_c)$ for over the time interval $0 \leq t \leq 2$ in Matlab where $F_c = 10$ Hz and $\theta_c = \frac{\pi}{3}$ radians. According to the sampling theorem, $c(t)$ can be represented by discrete-time samples provided the sample rate F_s is more than twice the largest frequency in $c(t)$. The highest frequency (and only frequency) in $c(t)$ is $F_c = 10$ Hz. Therefore the sample frequency must be greater than 20 Hz to avoid aliasing. If visualization or demonstration is the purpose of the simulation, it is a good idea to choose the sample rate 10 (or more) times faster than is required by the sampling theorem. On the other hand, if representation by discrete-time samples is all that is required, then sampling at 21 Hz will do.

Let's suppose that we choose $F_s = 5000$ Hz as the sample rate. In Matlab, we will work with the discrete-time sequence $c_n = c(nT)$ instead of $c(t)$ where $T = \frac{1}{F_s}$. Before generating the sequence c_n , we must generate the "time" vector $t_n = nT$ to represent time over the interval $[0, 3]$ seconds. The following Matlab commands will accomplish the construction of c_n .

```
fs = 5000; % Sample frequency
T = 1/fs; % Sample period
t_n = [0:T:3]; % Time vector with samples spaced T seconds apart
fc = 10; % Frequency of the sinusoid
theta = pi/3; % Phase of the sinusoid
c_n = cos(2*pi*fc*t_n + theta); % Construct the sinusoid
plot(t_n,c_n); % Plot the sinusoid
```

The final command plots the sinusoid. Here is what the plot looks like.



You can zoom in and use the time axis to measure the period of the sinusoid. Invert the period measurement and verify that the frequency really is 10 Hz. You can also measure the frequency of the sinusoid by looking for a peak of the spectrum of the signal.

4.2 Viewing the Spectrum of the Sampled Signal

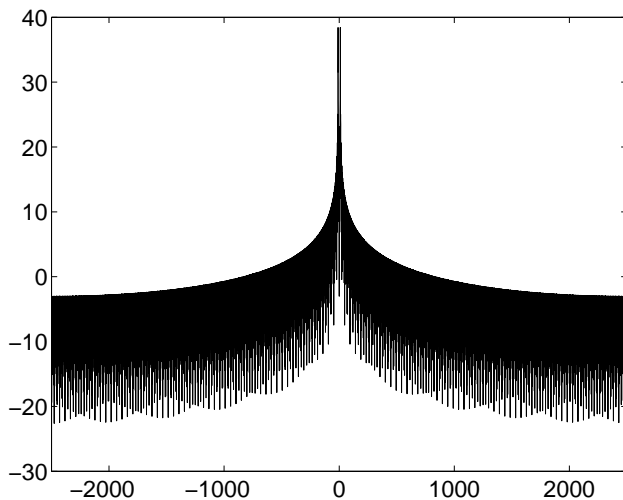
Matlab's `fft` function may be used to compute the discrete Fourier transform (DFT) of the sampled signal. Recall that discrete frequencies occupy the interval $[-\frac{1}{2}, \frac{1}{2}]$. The correspondence between discrete-time frequencies f and continuous-time frequencies F is given by the formula,

$$F = f * F_s.$$

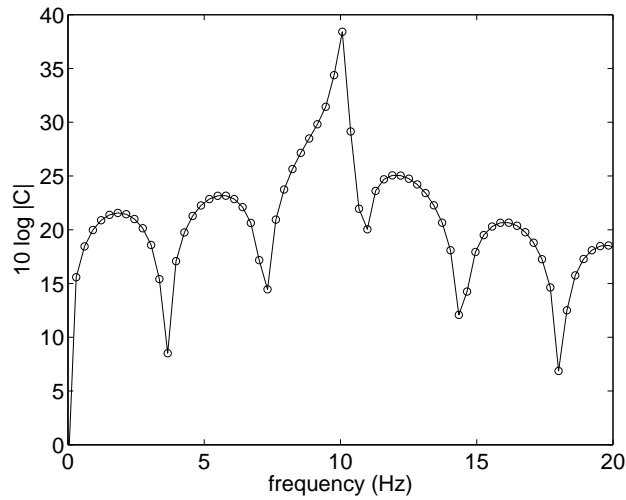
If we want frequency components in the sampled signal to appear in a plot of the spectrum at their true continuous time frequencies (assuming no aliasing), we have to generate a frequency vector that covers the range $[-\frac{1}{2}F_s, \frac{1}{2}F_s)$ with steps of F_s/N which is the sample frequency times the FFT bin width. Plotting the spectrum can be accomplished in Matlab using the following commands.

```
N = 2^14; % FFT size
f = ([0:N-1]/N - 0.5)*fs; % The frequency vector for plotting
C = fftshift(fft(c_n,N)); % Compute the FFT and rearrange the output
plot(f,10*log10(abs(C))); % Plot the magnitude of the spectrum on a
% log scale
```

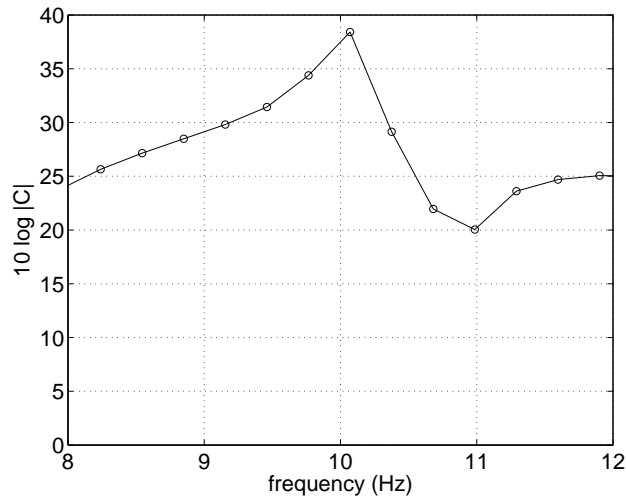
Here is what the spectrum looks like. Note that the x -axis has the correct frequency labeling and scaling.



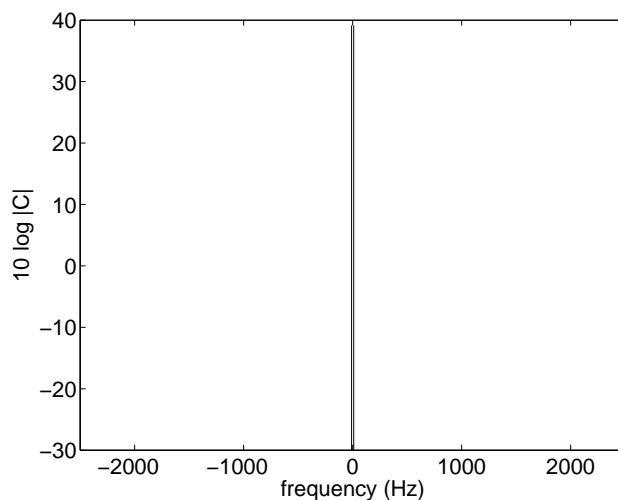
It is difficult to see the details of the spectrum around 10 Hz. The plot below shows the details around 10 Hz. Circles have been added around the data points which are connected by straight lines by the plot functions.



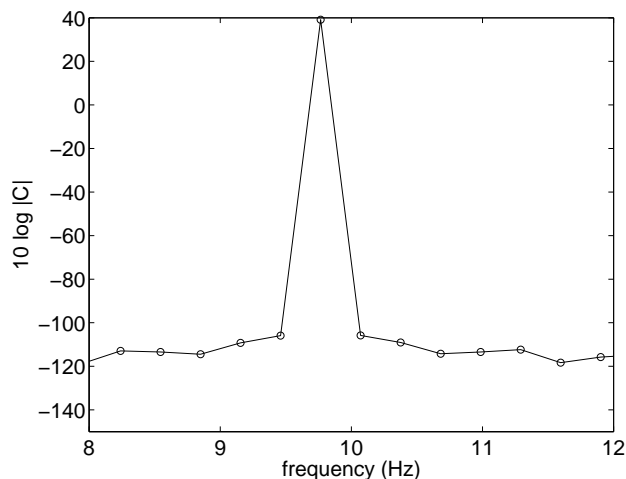
It appears that there is a peak in the spectrum at 10 Hz. However, zooming in to the interval [8, 12] Hz as shown in the figure below reveals that the peak is not actually at 10 Hz. This is due to the “spectral leakage” phenomenon.



If the frequency of the continuous-time signal is changed to 9.765625 Hertz (which is a FFT bin center frequency mapped back to a continuous time frequency), then the plot looks like this.



Note that except for the two peaks (which are unresolvable because they are at low frequency, the spectrum is effectively zero (less than -100 dB). Here is what the picture looks like zoomed to the frequency range 8 to 12 Hz.



Next, we explain how to construct sinusoids with frequencies which are DFT bin center frequencies.

4.3 Constructing Bin Center Discrete-time Sinusoids

Spectral leakage due to processing finite length data records smears the spectral lines of pure tones across several bins around the tonal frequency. Sinusoids with frequencies which are FFT bin centers are not smeared however. The FFT bin center frequencies are $0, \frac{1}{N}, \frac{2}{N}, \dots, \frac{N-1}{N}$ cycles per sample. The “fundamental range” can be shifted by $\frac{1}{2}$ to the following set of frequencies, $-\frac{1}{2}, -\frac{1}{2} + \frac{1}{N}, \dots, -\frac{1}{2} + \frac{N-1}{N}$. Using the correspondence between continuous and discrete frequencies, the corresponding continuous time frequencies that are not smeared are given by

$$F_{\text{bin center}} = -\frac{1}{2}F_s + \frac{k}{N}F_s \quad k = 0, 1, \dots, N - 1. \tag{30}$$

Exercises:

1. What is the frequency (in Hertz) of the sinusoidal signal $x(t) = \sin(19\pi^2 t)$? If this signal is sampled at a rate of 3 samples per second, what is the sampled signal $x(n/3)$ and what is the perceived frequency?
2. What is the fundamental period (in samples) of $\cos(\frac{n}{3})$?
3. Simulate the continuous-time signal $c(t) = \cos(2\pi F_c t + \theta_c)$ in Matlab using a sample rate $F_s = 15$ Hz. Let $\theta_c = 2\pi/3$ radians. Let $N = 2^{12}$ be the size of the FFT that will be used for analyzing the sampled signal.
 - (a) Choose F_c so that the sampled sinusoid $c_n = c(nT)$ has a discrete time frequency that is an FFT bin center frequency ($F_c = 0$ is not allowed). Remember that this is a demonstration. So, we want F_s to be much greater than F_c . What value of F_c did you choose? What FFT bin center frequency did you choose?
 - (b) How many seconds of data should you generate in Matlab so that you will have exactly N samples of the sinusoid?
 - (c) Generate the sinusoid in Matlab and plot it. The x -axis should be properly scaled so that the units are in seconds.
 - (d) Using the `fft` function, compute the spectrum and plot it. The x -axis in this plot should be properly scaled so that the units are in Hertz. Does your plot appear as predicted?