Experiment No.: 01

1 Aim

1. To get familiarity with basic commands in MATLAB.
2. To explore the connection between system impulse response and the solution of linear ordinary constant coefficient differential equation.
3. To understand and implement convolution routine for discrete time finite lenght sequences.

2 Software Used

1. MAT LAB

3 Theory

For theory, we can refer the text book: **Alan V.Oppenheim, Alan S.Willsky, Signals and Systems Systems, Second edition, Prentice hall (1997)**.

4 Procedure

Exercise 1 *Generate and plot following signals/functions*

$$x_1(t) = e^{-t}cos(2\pi t) \tag{1}$$
$$x_2(t) = 1 + 1.5cos(2\pi t) - 0.6cos(4\pi t) \tag{2}$$
$$x_3(t) = |cos(2\pi t)| \tag{3}$$

Simulations and calculations on computing devices are performed with a discretized version of the continuous signal. For example, $x(t)$ is represented by a finite length sequence $x(n) = x(t = nT_s)$, where T_s is the sampling time interval. For this exercise use $T_s = 0.001sec$.

Exercise 2 *Convolution of two infinite length sequences $x(n)$ and $h(n)$ is given by*

$$y(m) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(m-k). \tag{4}$$

Rewrite this expression to convolve two finite length sequences/signals. Show the relationship between the lengths of input and output. Write a MATLAB function ($[y] = myconv(x, h)$) that takes input arguments as two finite length sequences and produces convolution of these two sequences as an output. For the input sequence $x(n)$ and system impulse responses $h_1(n)$ and $h_2(n)$,

$$x(n) = [\vec{1}, 1, 1, 1, -1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1] \tag{5}$$
$$h_1(n) = [\vec{1}, 1] \quad and \quad h_2(n) = [\vec{1}, -1],$$

calculate the corresponding output sequences $y_1(n) = x(n) * h_1(n)$ and $y_2(n) = x(n) * h_2(n)$ using your own convolution routine.

Exercise 3 (Analysis of LTI Systems) *Consider an RC low pass filter circuit. The relationship between the input voltage $x(t)$ and the output voltage $y(t)$ is given by the differential equation*

$$x(t) = y(t) + RC\frac{dy(t)}{dt} \tag{6}$$

In order to discretise the differential equation, we approximate the derivative of $y(t)$ sampled at spacing T_s, at a time kT_s, as

$$\frac{dy(t)}{dt} \approx \frac{y_k - y_{k-1}}{T_s} \tag{7}$$

Inserting this approximation into the differential equation of the RC filter lead to the following expression

$$RC\frac{y_k - y_{k-1}}{T_s} + y_k = x_k. \tag{8}$$

This equation can be easily solved in terms of y_k. The solution has the form $y_k = ax_k - by_{k-1}$ where $a = \frac{1}{1+\frac{RC}{T_s}}$ and $b = \frac{-1}{1+\frac{T_s}{RC}}$. Write a MATLAB code to solve this difference equation with input specified by $x(t)$ in terms of two step functions,

$$x_1(t) = u(t) \quad and \quad x_2(t) = -u(t-1) \tag{9}$$

$$x(t) = x_1(t) + x_2(t). \tag{10}$$

Assume $x_1(n)$ and $x_2(n)$ are discrete versions of $x_1(t)$ and $x_2(t)$ with finite length L=2000. Take $T_s = 0.001$ sec.

Exercise 4 *Output of the RC filter discussed in Exercise 3 can also be characterized by impulse response $h(t)$ of the filter. Output of a given LTI system is characterized by the convolution of input with the system impulse response.*

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)\,d\tau. \tag{11}$$

In order to compute convolution discretely, we approximate the integration in the above expression, and we get:

$$y(m) = T_s \sum_{k=-\infty}^{\infty} x(k) \cdot h(m-k). \tag{12}$$

5 Observation

Write/ Plot Your Own With Observation Table (If Required).

6 Analysis of Results

Write Your own.

7 Conclusions

Write Your Own.

Precautions

Observation should be taken properly.

Experiment No.: 07

1 Aim

To compute and plot the Fourier spectra for the aperiodic signals.

2 Software Used

1. MATLAB

3 Theory

Aperiodic Signals: The signals which may be repetitive but only over a finite interval. These signals are analyzed by means of the Fourier Transform.
Fourier series: A Fourier series is an expansion of a periodic function terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality relationships of the sine and cosine functions. The computation and study of Fourier series is known as harmonic analysis and is extremely useful as a way to break up an arbitrary periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical. For a periodic signal, one that repeats exactly every, say, T seconds, there is a decomposition that we can use, Fourier series decomposition, to put the signal in this form. If the signals are not periodic we can extend the Fourier series approach and do another type of spectral decomposition of a signal called a Fourier Transform.
Fourier Transform: The Fourier transform expresses a function of time (a signal) as a function of frequency. This is similar to the way in which a musical chord can be expressed as the amplitude (or loudness) of its constituent notes. The Fourier transform of a function of time itself is a complex-valued function of frequency, whose complex modulus represents the amount of that frequency present in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency. The Fourier transform is called the frequency domain representation of the original signal. The term Fourier transforms refers to both the frequency domain representation and the mathematical operation that associates the frequency domain representation to a function of time. The Fourier transform is not limited to functions of time, but in order to have a unified language, the domain of the original function is commonly referred to as the time domain. For many functions of practical interest one can define an operation that reverses this: the inverse Fourier transformation, also called Fourier synthesis, of a frequency domain representation combines the contributions of all the different frequencies to recover the original function of time. Functions that are localized in the time domain have Fourier transforms that are spread out across the frequency domain and vice versa. The critical case is the Gaussian function, of substantial importance in probability theory and statistics as well as in the study of physical phenomena exhibiting normal distribution (e.g., diffusion), which with appropriate normalizations goes to itself under the Fourier transform. Joseph Fourier introduced the transform in his study of heat transfer, where Gaussian functions appear as solutions of the heat equation.

4 Procedure

Note: Students have to do both magnitude plot & Phase Plot.

Create your own function $X = mydft(x, t_o, t_s)$ which will generate DFT of x.

Exercise:1 The Fourier transform (FT) of an aperiodic continuous-time signal x(t) is given by

$$X(\omega) = \int_{-\infty}^{\infty} \mathrm{x}(t)e^{-j\omega t}\,\mathrm{d}t \tag{1}$$

In numerical computations, the data must be finite. Let us consider the signal $x(t)$ of finite duration T_0 . We approximate the FT of the finite duration signal $x(t)$. The Fourier transform (FT) of an aperiodic continuous-time signal $x(t)$ is given by

$$\begin{aligned} X(\omega) &= \int_{0}^{T_0} \mathrm{x}(t)e^{-j\omega t}\,\mathrm{d}t \\ &= \lim_{T_s \to \infty} \sum_{k=0}^{N_0-1} x(kT_s)e^{-j\omega T_s}T_s; \end{aligned} \tag{2}$$

Where Ts denotes the sampling interval of the signal x(t) and $N = T_0/T_s$ is the total number of samples. Let us consider the samples of $X()$ at regular interval of ω_0. If X_r is the r^{th} sample, then from Equation 2, we obtain

$$\begin{aligned} X_r &= \sum_{k=0}^{N_0-1} T_s x(kT_s)e^{-jr\omega_0 kT_s} \\ &= \sum_{k=0}^{N_0-1} x_k e^{-jr\Omega_0 k}; \end{aligned} \tag{3}$$

where $x_k = T_s\ x(kT_s)$, $X_r = X(r\omega_0)$ and $\Omega_0 = \omega_0 T_s$. Use MATLAB to compute the FT of the following signal:

$$x_1(t) = e^{-2t}\ u(t) \tag{4}$$

where $u(t)$ denotes the continuous-time unit step function.

5 Observation

Plot DFT for these two cases alongside FT using the inbuild command $X = fft(x)$ with observation Table.

1. $T_0 = 4$ sec, $T_s = 1/64$ sec.
2. $T_0 = 8$ sec, $T_s = 1/32$ sec.

6 Analysis of Results

Write Your own.

7 Conclusions

Write Your Own.

Precautions

Observation should be taken properly.

Experiment No.: 08

1 Aim

1. Implementation of discrete Fourier transform (DFT) and inverse DFT (IDFT) algorithm.
2. Implementation of autocorrelation and cross correlation algorithm.

2 Software Used

1. MATLAB

3 Theory

The **discrete Fourier transform** (DFT) converts a finite list of equally spaced samples of a function into the list of coefficients of a finite combination of complex sinusoids, ordered by their frequencies, that has those same sample values. It can be said to convert the sampled function from its original domain to the frequency domain. The input samples are complex numbers, and the output coefficients are complex as well. The frequencies of the output sinusoids are integer multiples of a fundamental frequency, whose corresponding period is the length of the sampling interval. The combination of sinusoids obtained through the DFT is therefore periodic with that same period.

The sequence of N complex numbers $x_0, x_1, x_2, \cdots, x_{N-1}$ is transformed into an N-periodic sequence of complex numbers:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi\, k\, n/N}; \qquad k \in \mathbb{Z} \tag{1}$$

Inverse discrete Fourier transform (IDFT) is given as

$$X[n] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, e^{j2\pi\, k\, n/N}; \qquad n \in \mathbb{Z} \tag{2}$$

In signals and systems analysis, the relationships between signals often indicate whether the physical phenomena which produced the signals are related or whether one signal is a modified version of the other. The relationship between two signals indicates whether one depends on the other, both depend on some common phenomenon, or they are independent. Correlation function indicates how correlated two signals are as a function of how much one of them is shifted in time.

The autocorrelation function (ACF) of a random signal describes the general dependence of the values of the samples at one time on the values of the samples at another time. Consider a random process $x(t)$ (i.e. continuous-time), its autocorrelation function is computed as

$$R_{xx}(\tau) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)dt; \tag{3}$$

where T is the period of observation. $R_{xx}(\tau)$ is always real-valued and an even function. For sampled signal, the ACF is defined as

$$R_{xx}[m] = \frac{1}{N}\sum_{n=1}^{N-m+1} x[n]x[n+m-1]; \qquad m = 1, 2, ..., N+1 \tag{4}$$

where N is the number of samples. The cross-correlation function (CCF) measures the dependence of the values of one signal on another signal. For two wide sense stationary (WSS) processes $x(t)$ and $y(t)$, the CCF is defined as

$$R_{xx}(\tau) = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)y(t+\tau)dt; \tag{5}$$

where T is the period of observation.
For sampled signal, the CCF is defined as

$$R_{xx}[m] = \frac{1}{N}\sum_{n=1}^{N-m+1} x[n]y[n+m-1]; \qquad m = 1, 2, ..., N+1 \tag{6}$$

where N is the number of samples. ACF is simply a special case of the CCF.

4 Procedure

1. Write a simple MATLAB program to perform the DFT. Use your program to determine and plot the DFT of the following signal:

   $$x(t) = \cos(2000\pi t) + \cos(800\pi t)$$

   Let $F_s = 8000Hz$ and $N = 128$ and plot the magnitude and phase of $X(k)$.

   Write a simple MATLAB routine to perform the inverse DFT. Use $x_1[n] = \{1, 1, 1, 1\}$, and $N = 4$, determine the DFT. Plot the Magnitude and phase spectra of the $x_1[n]$. Use the IDFT to transfer the DFT results (i.e. $X[k]$ sequence) to its original sequence.

2. Take a speech signal and plot the magnitude and phase spectra.

3. Write a simple MATLAB program to perform the CCF. Use your routine to plot the crosscorrelation of the following signal:

   $$\begin{aligned} x(t) &= \sin(2\,\pi\, f\, t), \\ y(t) &= x(t) + w(t), \end{aligned} \tag{7}$$

   where $w(t)$ is a zero-mean, unit variance of the Gaussian random process. Sampled version is written as

   $$\begin{aligned} x[n] &= \sin\left(\frac{2\,\pi\, f\, n}{F_s}\right), \\ y[n] &= x[n] + w[n], \end{aligned} \tag{8}$$

   Let f = 1Hz, F_s = 200Hz, N = 1024 and $w[n]$ is generated by $randn(1, N)$ MATLAB function.

5 Observation

Write/ Plot Your Own With Observation Table (If Required).

6 Analysis of Results

Write Your own.

7 Conclusions

Write Your Own.

Precautions

Observation should be taken properly.

Experiment No.: 09

1 Aim

1. To generate two periodic signals $x_1(t)$ and $x_2(t)$.
2. To compute and plot the Fourier spectra for the aforementioned periodic signals.
3. To illustrate the Gibb's phenomenon.

2 Software Used

1. MATLAB

3 Theory

Fourier series is a way to represent a wave-like function as the sum of simple sine waves. More formally, it decomposes any periodic function or periodic signal into the sum of a (possibly infinite) set of simple oscillating functions, namely sines and cosines (or, equivalently, complex exponentials). Fourier series make use of the orthogonality relationships of the sine and cosine functions. The computation and study of Fourier series is known as harmonic analysis and is extremely useful as a way to break up an arbitrary periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical.In particular, since the superposition principle holds for solutions of a linear homogeneous ordinary differential equation, if such an equation can be solved in the case of a single sinusoid, the solution for an arbitrary function is immediately available by expressing the original function as a Fourier series and then plugging in the solution for each sinusoidal component. In some special cases where the Fourier series can be summed in closed form, this technique can even yield analytic solutions.

The Fourier Series of a periodic signal $x(t)$ with period T is given by

$$x(t) = \sum_{k=-\infty}^{+\infty} D_k \, e^{j\,k\,\omega_0\,t}; \qquad \omega_0 = \frac{2\pi}{T} \tag{1}$$

where the Fourier Series coefficient D_k is calculated by

$$D_k = \frac{1}{T} \int_T x(t)\, e^{-j\,k\,\omega_0\,t} dt. \tag{2}$$

In order to compute the D_k discretely, we approximate the aforementioned finite integral.

$$D_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n\,T_s)\, e^{-j\,k\,\Omega_0\,n}; \qquad \Omega_0 = \omega_o\,T_s \tag{3}$$

where T_s denotes the sampling interval and $N = \frac{T}{T_s}$ is the number of samples in one period T.

Gibb's phenomenon: The peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity: the partial sum of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as the frequency increases, but approaches a finite limit. The Gibb's phenomenon involves both the fact that Fourier sums overshoot at a jump discontinuity, and that this overshoot does not die out as the frequency increases.

4 Procedure

Exercise 1 *For this exercise we consider the periodic signals $x_1(t)$ and $x_2(t)$ defined , respectively as*

$$x_1(t) = e^{-t/2}; \qquad 0 \le t \le 1 \tag{4}$$

and

$$x_2(t) = \begin{cases} 1; & 0 \le t \le T/2 \\ -1; & T/2 < t \le T \end{cases} \tag{5}$$

where $T = 2$.
For the numerical computation of D_k, we use $N = 256$. Compute 10 coefficients of the periodic signals $x_1(t)$ and $x_2(t)$ and also plot the Fourier spectra.

Exercise 2 *Let $x_M(t)$ be the approximation of the original periodic signal $x(t)$. It is defined as*

$$x_M(t) = \sum_{k=-M}^{M} D_k\, e^{j\,k\,\omega_0\,t}. \tag{6}$$

Plot $x_M(t)$ as a function of time for $M = 19$ and 99.

5 Observation

Write/ Plot Your Own With Observation Table (If Required).

6 Analysis of Results

Write Your own.

7 Conclusions

Write Your Own.

Precautions

Observation should be taken properly.

Experiment No: 10

1 Aim

1. To Simulate Continuous-time Sinusoidal Signals in Discrete-time.
2. To illustrate DSB-SC modulation and demodulation.
3. To illustrate FM modulation and demodulation.

2 Software Used

1. MATLAB.

3 Theory

3.1 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 $F_c = 10Hz$ and $\theta_c = \pi/3$ radians. According to the sampling theorem, $c(t)$ can be represented by discrete-time samples provided the sample rate Fs is more than twice the largest frequency in $c(t)$. The highest frequency (and only frequency) in $c(t)$ is $F_c = 10Hz$. Therefore the sample frequency must be greater than $20Hz$ 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 $21Hz$ will do.

Let's suppose that we choose $F_s = 5000Hz$ as the sample rate. In Matlab, we will work with the discrete-time sequence $c_n = c(nT)$ instead of $c(t)$ where $T = 1/F_s$. 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 appart
$fc = 10;$ % 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 $10Hz$. You can also measure the frequency of the sinusoid by looking for a peak of the spectrum of the signal.

3.2 Viewing the Spectrum of the Sampled Signal

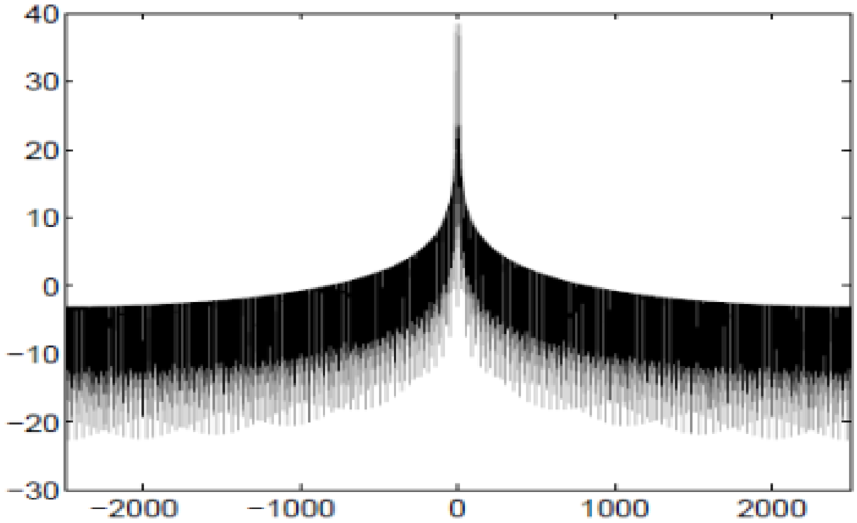
Matlab's fft function may be used to compute the discrete Fourier transform (DFT) of the sampled signal. Recall that discrete frequencies occupy the interval $[\frac{-1}{2}, \frac{1}{2}]$. The correspondence between discrete-time frequencies f and continuous-time frequencies F is given by the formula,

$$F = f * F_s$$

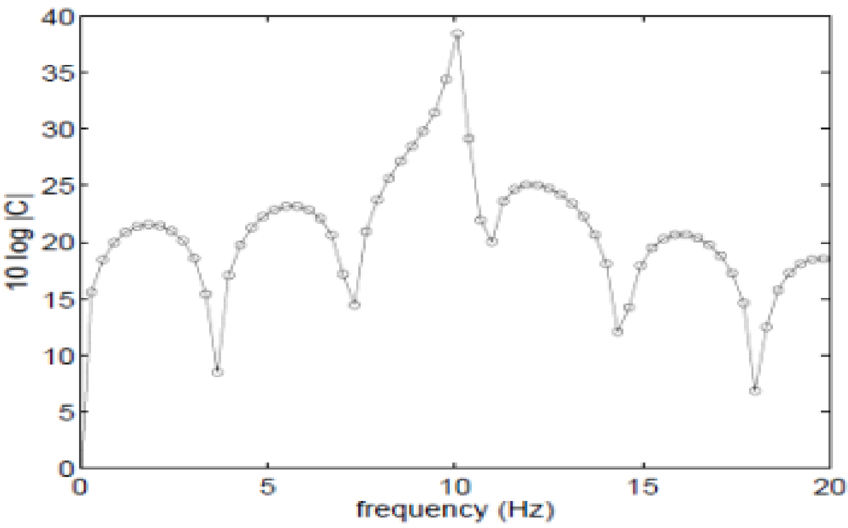
If we want frequency components in the sampled signal to appear in a plot of the spectrum at their true continuous time frequencies (assuming no aliasing), we have to generate a frequency vector that covers the range $[\frac{-1}{2}F_s, \frac{1}{2}F_s]$ with steps of F_s/N which is the sample frequency 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 labelling and scaling.



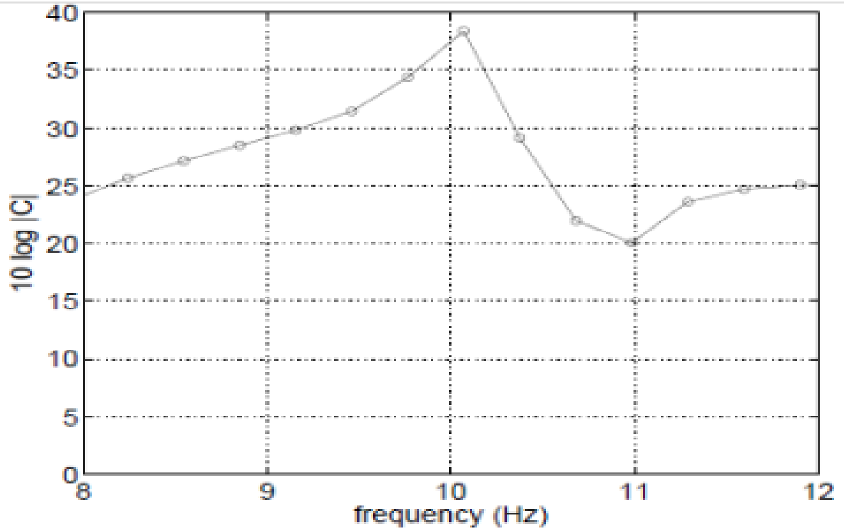
It is difficult to see the details of the spectrum around $10Hz$. The plot below shows the details around $10Hz$. Circles have been added around the data points which are connected by straight lines by the plot functions.

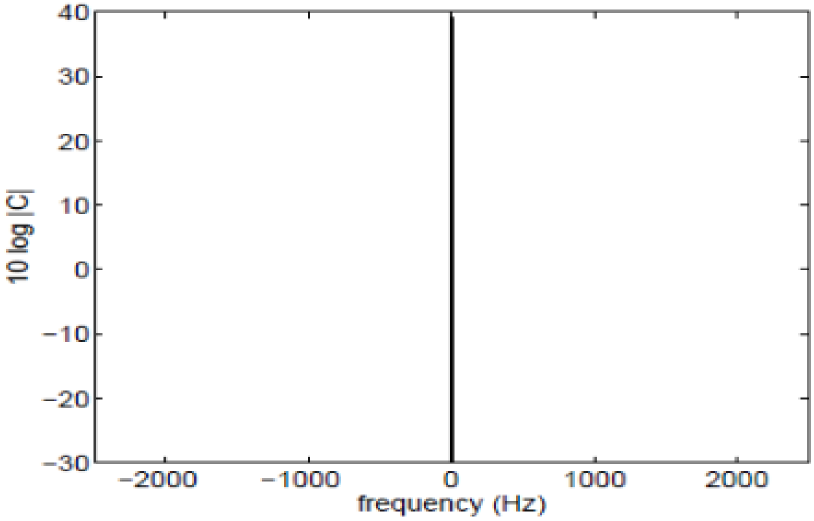


It appears that there is a peak in the spectrum at $10Hz$. However, zooming in to the interval $[8, 12]$ Hz as shown in the figure below reveals that the peak is not actually at $10Hz$. This is due to the "spectral leakage" phenomenon.

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 irresolvable because they are at low frequency, the spectrum is effectively zero (less than $-100dB$). Here is what the picture looks like zoomed to the frequency range 8 to $12Hz$.





3.3 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 $0, \frac{1}{N}, \frac{2}{N}, \cdots, \frac{(N-1)}{N}$ cycles per sample. The "fundamental range" can be shifted by 1/2 to the following set of frequencies, $-\frac{1}{2}, -\frac{1}{2}+\frac{1}{N}, \cdots, -\frac{1}{2}+\frac{(N-1)}{N}$. Using the correspondence between continuous and discrete frequencies, the corresponding continuous time frequencies that are not smeared are given by

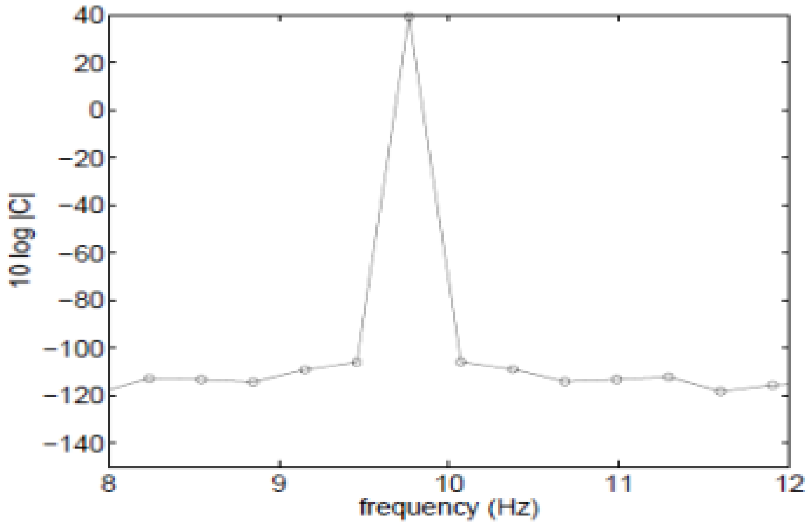
$$F_{bin\ center} = -\frac{1}{2}F_s + \frac{k}{2}F_s; \qquad k = 0, 1, \cdots, N-1.$$

4 Procedure

Exercise 1

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 $(\frac{n}{3})$.



3. Simulate the continuous-time signal $c(t) = cos(2\pi F_c t + \theta_c)$? in Matlab using a sample rate $Fs = 15Hz$. 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 Fc 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?

Exercise 2

DSB-SC modulation and demodulation

The message signal $m(t)$ in the interval $[-0.04, 0.04]$ is given as

$$m(t) = \triangle\left(\frac{t + 0.01}{0.01}\right) - \triangle\left(\frac{t - 0.01}{0.01}\right); \tag{1}$$

where $\triangle\left(\frac{t}{\tau}\right)$ is a triangular function of width τ , the carrier signal $c(t)$ is given as

$$c(t) = cos(2\pi f_c\ t); \tag{2}$$

where f_c is the carrier frequnecy and the DSB-SC signal is defined as

$$\phi(t) = m(t)\ cos(2\pi f_c\ t) \tag{3}$$

The demodulated signal $e(t) = \phi(t) \times 2cos(2\pi f_c\ t + \theta)$, where $2cos(2\pi f_c\ t + \theta)$ is a local carrier generated at the demodulator. $\hat{m}(t)$ denotes the recovered message signal after low-pass filtering. Coherent demodulation is also implemented with a finite impulse response (FIR) low-pass filter of order 40. The low-pass filter at the demodulator has bandwidth of $150Hz$.

Write a simple MATLAB routine to illustrate DSB-SC modulation and demodulation schemes for $f_c = 300Hz$ and an arbitrary θ. Plot $m(t)$, $\phi(t)$, $e(t)$ and $\hat{m}(t)$ in time and frequency domains. In the ideal coherent demodulation we assume that the phase of the local carrier is equal to the phase of the carrier (i.e. $\theta = 0$). If that is not case- i.e. if there exists a phase shift θ between the local carrier and the carrier- how would the demodulation process change?

Exercise 3

FM modulation and demodulation
Once again, we use the same message signal $m(t)$ and carrier signal $c(t)$. The FM signal is defined as

$$\phi_{FM}(t) = \cos\left(2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\alpha)d\alpha\right); \tag{4}$$

where k_f is the FM coefficient. We first integrate the message signal and then use (4) to find $\phi_{FM}(t)$. To demodulate the FM signal, a differentiator is first applied to convert the frequency-modulated signal an amplitude- and frequency modulated signal (i.e. $d(t) = \frac{d\phi_{FM}(t)}{dt}$. By applying envelope detection (non-coherent AM demodulation), we recover the message signal $\hat{m}(t)$.

Write a simple MATLAB routine to illustrate FM modulation and demodulation schemes for $f_c = 300Hz$ and $k_f = 80$. Plot $m(t)$, $\phi_{FM}(t)$, $d(t)$ and $\hat{m}(t)$ in time and frequency domains.

5 Observation

Write/ Plot Your Own With Observation Table (If Required).

6 Analysis of Results

Write Your own.

7 Conclusions

Write Your Own.

Precautions

Observation should be taken properly.