

Laboratory Assignment 4

Fourier Sound Synthesis

PURPOSE

This lab investigates how to use a computer to evaluate the Fourier series for periodic signals and to synthesize audio signals from Fourier series coefficients. The Fourier series should be a familiar mathematical concept. We show how to use the DFT (Discrete Fourier transform) to compute the Fourier series for digital sound synthesis. Using a computer, we can determine the Fourier series coefficients for an arbitrary periodic signal corresponding to a sampled audio signal having a given pitch. Then similarly shaped waveforms having different fundamental frequencies, i.e., different pitch, can be generated to create interesting musical sounds. We explore how these sounds can be modified using ideal filters.

4.1 OBJECTIVES

By the end of this laboratory assignment, you should be able to:

1. Synthesize a signal from a finite number of Fourier series coefficients.
2. Compute Fourier series coefficients for an arbitrary periodic signal (that may not have an analytical solution).
3. Perform and analyze basic ideal filtering operations on periodic signals.

4.2 REFERENCES

Review Topics

1. Fourier series analysis and synthesis

Exploratory Topics

1. Discrete-time Fourier analysis and synthesis
2. Ideal filtering of arbitrary periodic waveforms

Reference

1. Deller, J. R. Jr. *Tom, Dick, and Mary Discover the DFT*. IEEE Signal Processing Magazine, April 1994 pg. 36-50

4.3 LABORATORY PREPARATION

Problems

Question 1. Assume that a significant harmonic is defined to have a magnitude within 50 dB of the magnitude of the highest harmonic. Using this figure of merit, how many harmonics are significant for a square wave? Use the analytic formulation for the FS of a square wave. You may wish to review the definition of dB given in Laboratory Assignment 3.

Question 2. Consider a signal that comprises only two harmonic components: the third harmonic, having a magnitude of two and an angle of $\pi/4$, and the fifth harmonic, having a magnitude of $1/2$ and an angle of $3\pi/4$. Construct 2 periods of this waveform, assuming that its fundamental frequency is 50 Hz. If an ideal lowpass filter is created to remove the fifth harmonic, what would be a reasonable cutoff frequency (in Hz)?

Question 3. Given the waveform $f(t)$ below, determine its Fourier series coefficients a_n analytically in terms of the integer index n , assuming it is periodic outside of the interval shown.

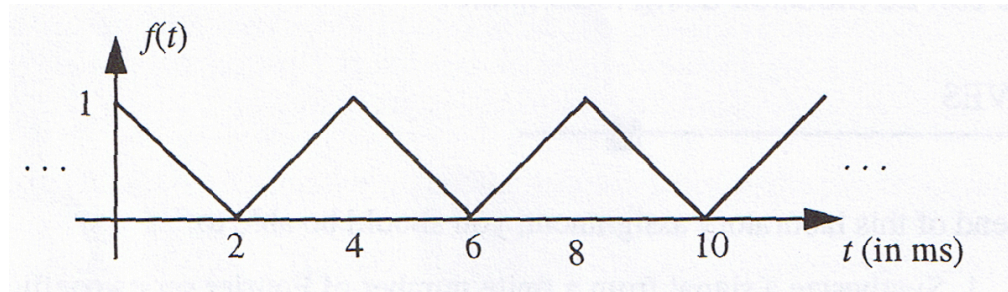


Figure 4.3.1 Waveform for Problem 3

What are the Fourier series coefficients for the signal resulting from filtering $f(t)$ with an ideal LPF that has a cutoff frequency of $3.5\omega_0$, where ω_0 corresponds to the fundamental frequency of $f(t)$?

Question 4. Verify that your Fourier series analysis from Question 3 is correct by plotting the waveform resulting from using the first four Fourier series coefficients. What happens to the waveform shape as you increase and decrease the number of Fourier series coefficients used in the approximation? Identify any differences between the Fourier series approximation and the original signal. Hypothesize, on the basis of your observations, as to whether these differences will disappear if enough of the harmonic frequencies are included in the approximation. Plot the magnitude of the first four Fourier series coefficients as a discrete-time sequence with the corresponding harmonic frequencies indicated on the horizontal axis.

Question 5. Discuss whether the Fourier synthesis method described for the guitar could be applied to different types of instruments, e.g., a trumpet. Explain your reasoning.

4.4 BACKGROUND

Motivation for Frequency Analysis

Much of signal processing, communication and control systems analysis relies upon frequency analysis of relevant signals and systems. Frequency response, bandwidth, and tone control on stereo systems are all concepts based on the idea that a signal can be thought of as comprising sinusoidal components at different frequencies.

Traditional frequency analysis uses two basic transforms: the Fourier series, for periodic waveforms, and the Fourier transform, for aperiodic waveforms. In both cases the information from a signal is converted to a frequency-domain representation; instead of representing the information in the signal as a function of time, it is represented as a function of frequency.

What does plotting the signal content as a function of frequency tell us? In a broad sense, it tells us at what frequencies the signal has energy, just as a prism visually shows the colors - frequencies—present in a light source, or a stereo spectrum analyzer shows the relative loudness of tones in a musical piece. Frequency analysis helps to understand other technologies as well. For example, the electric power in a standard wall outlet contains energy predominately at 60 Hz - AC power. A radio station broadcasts energy around its transmission frequency, which is indicated on your radio station dial. By transforming signal data to be viewed in the frequency domain, we can find out what frequencies are significant in a given signal.

The behavior of systems also can be described using frequency-domain concepts. Systems are characterized by their frequency response, i.e., how they affect the frequency content of an input signal. For example, the tone control on a stereo (bass and treble knobs) has two effects: the bass knob controls the gain (relative loudness) on low-frequency signal components, whereas the treble knob controls the gain on high-frequency signal components. Examining the frequency content of a stereo signal before and after tone control can tell what the two knobs are doing. In a more general sense, the frequency response of any unknown system can be determined from the relative frequency content of signals at its input and output.

This set of experiments deals with three concepts in the context of synthesizing and modifying audio sounds: calculation of the frequency information in arbitrary periodic signals via the Fourier series (FS); the effect of certain systems, called ideal filters, on the FS coefficients and the audio properties of signals; and determination of a system's frequency response based on input and output signal Fourier series spectra.

Why bother with the Fourier series at all? This mathematical representation provides a means for determining the frequency content of periodic signals. For a given periodic signal, the gain (magnitude) and time-shift (phase) of sinusoids at each harmonic frequency define the signal's frequency spectrum. Knowing the frequency content of a

signal allows you to determine how a signal will or should be modified by a physical system. In this lab, we investigate how systems affect the frequency content of a signal in the context of ideal filters having both periodic input and output signals that can be represented by their respective Fourier series.

Scenario

You are interested in getting hired by a small start-up company that develops multimedia application packages for entertainment and education. This company uses advanced digital audio processing in their products to produce high quality sound with low storage requirements, which is the main selling point for their products. You have researched the company's products and financial status and are impressed. This company is interviewing on campus. Knowing that the competition is stiff for the single engineering position available, you would like to make a good impression.

You have played around some with sound synthesis using simple tones and sampled sounds, but recognize the large storage requirements needed to produce high-quality music using only samples of digitized instrument sounds. You have heard of something called Fourier synthesis. Recalling the Fourier series from your coursework, you wonder if these mathematical techniques are used in Fourier synthesis to develop methods for low-cost, high-quality digital sound synthesis. So you decide to explore how to use a computer to perform Fourier series analysis and synthesis to see what kinds of sounds you can create.

Fourier Series and Discrete Fourier Series

The Fourier series representation of a periodic signal should be a familiar concept from introductory engineering coursework. The main concept behind the FS is that periodic signals can be represented as a sum of sinusoidal signals, with the restriction that the resulting sum has the same fundamental frequency as the signal itself— i.e., the same periodicity. A classic example is the construction of a square wave from the FS coefficients; there is a MATLAB demo that does just this: type **fourier** in MATLAB to see the development.

The FS is defined by two equations: an analysis equation, which, from the signal, determines the FS coefficients, and a synthesis equation, which defines how to construct the signal as a sum of sinusoids weighted by the FS coefficients.

1. Fourier Series: Analysis
$$a_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi f_0 t} dt$$

2. Fourier Series: Synthesis
$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi k f_0 t}$$

In both equations (1) and (2), $f_0 = 1/T$, the fundamental frequency of the signal, T is the period of the signal, and the a_k 's are the complex coefficients, each of which weights the k th harmonic (having frequency kf_0). Note that since the above synthesis equation is expressed in terms of complex exponential functions, the index on the summation includes negative values so that the resulting sum defines sinusoids. (Recall that $\cos(\theta) = (e^{j\theta} + e^{-j\theta}) / 2$.) Often the sign associated with the index is associated with the frequency; kf_0 , where k is negative results in a mathematical artifact called a negative frequency.

Since the signals in which we are interested are real-valued functions of time, the Fourier coefficients form complex conjugate pairs, $a_k = (a_{-k})^*$.

The Fourier series also can be expressed as a sum of sinusoids,

$$x(t) = a_0 + 2 \sum_{k=1}^{\infty} |a_k| \cos(2\pi k f_0 t + \angle a_k)$$

Note that the magnitude of each Fourier series coefficient determines the amplitude of the sinusoid at the corresponding harmonic frequency, and the phase of the FS determines the phase shift of the sinusoid.

If the signal is sampled to yield a discrete-time signal, we can use the discrete Fourier Series for analysis and synthesis. In this case, integration is replaced by summation in the FS analysis equation, and the signal periodicity is expressed in terms of an integer number of samples N rather than time T .

3. Discrete Fourier Series: Analysis
$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

4. Discrete Fourier Series: Synthesis
$$x[n] = \sum_{k=0}^{N-1} a_k e^{j2\pi kn/N}$$

Note that in the discrete case only a finite number of DFS coefficients are used. Since the data is sampled, there is a restriction on the frequencies that can be represented, limited to half of the sampling rate. This gives a maximum discrete-time frequency of $\Omega = \pi k N$, corresponding to a continuous-time frequency of $\omega = \Omega/T_s$ where T_s is the sampling period. The remaining terms in the sum correspond to the negative frequencies values, shifted by $+2\pi$, as needed to create sinusoids from complex exponentials.

These two equations are readily implemented in MATLAB: simply write a script that evaluates the sums for all n and k values. There are, however, more efficient methods for computing the DFS using Fast Fourier Transforms (FFT), which will be discussed later.

Interpreting Fourier Series

So now we have a set of Fourier series coefficients for a signal, and we can recreate the signal from its coefficients. What does this mean?

We are interested in knowing a few things about a signal: what range of frequencies does it contain? Which of these have the most energy? How much energy is stored in a particular frequency? How can the signal be changed by altering its frequency content? The FS representation provides us with a means for answering these questions.

To help answer these questions, we can plot the magnitude of the FS coefficients as a DT sequence where the horizontal axis is used to indicate the harmonic frequency (kf_0) associated with each coefficient. Consider the FS representation of a periodic square wave, which is commonly used as an example in lecture or as homework. It contains only odd harmonics ($k = 1, 3, 5, \text{etc.}$) and the amplitude of each harmonic is proportional to the frequency ($1/k$). The frequency spectrum is plotted in Figure 4.4.1 for a 5-Hz periodic square wave.

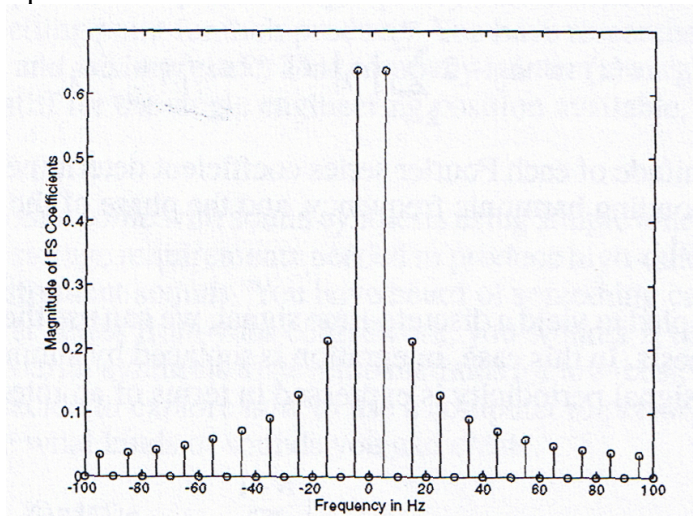


Figure 4.4.1 FS Coefficients for Square Wave

It is clear from Figure 4.4.1 that most of the energy is contained in the fundamental or first harmonic, which is at 5 Hz. The energy at each higher harmonic drops off rapidly. If we were to truncate this series by using only the first 10 harmonics to represent the signal, all of the energy above 55 Hz would be lost.

Digital Fourier Series Analysis and Synthesis

Since we will be using MATLAB on a computer, we will focus on how to compute Fourier spectra for discrete-time signals resulting from sampling continuous-time signals. This process is facilitated by the DFT (Discrete Fourier Transform), implemented in MATLAB using the command `fft`, which uses the computationally efficient method called the Fast Fourier Transform (FFT).

The Fourier transform is an analytical method for determining the frequency content of continuous-time analytic signals, which may or may not be periodic. Since aperiodic signals do not have a fundamental period, they may have power at a continuum of frequencies, rather than only at harmonic frequencies. Thus all possible frequencies can be considered in analysis and synthesis. The continuous-time FT is an integral evaluated over the entire signal, from the beginning to the end of time.

However, to use a computer to evaluate the spectrum (Fourier transform) of a signal represented by a vector of time-domain samples, it is neither possible nor desirable to numerically evaluate this integral for all time. Instead of approximating the Fourier transform integral by a summation over infinite time, we only sum over a finite number of samples. However, the number of samples used in the time domain corresponds to the number of “samples” of the signal’s continuous frequency spectrum. The frequency resolution of the FT is limited to the number of points used in the computation. If we want to analyze for higher frequencies, we must sample at a higher rate since the uppermost frequency is limited by the sampling rate; if we want to analyze more densely we have smaller frequency steps we must take a longer time sequence.

The analysis and synthesis equations that result from this sampling of the signal and its spectrum define what is called the Discrete Fourier Transform:

$$5. \quad \text{DFT: Analysis } X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

$$6. \quad \text{DFT: Synthesis } x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N}$$

In these equations, $2\pi nk/N \equiv \omega_k$. Note that these equations look virtually identical to those defining the DFS. The only major difference is that the frequencies corresponding to each sampled Fourier transform value, or Fourier coefficient $X[k]$, are not necessarily related to a fundamental period. To calculate the DFS coefficients using the DFT we need to define N such that it corresponds as closely as possible to one period of the signal, which is defined by the sample values stored in $x[n]$.

The DFT in MATLAB is not computed using loops that implement the summation equations above; rather, the Fast Fourier Transform (FFT) is used. The FFT essentially streamlines the DFT computations to have a minimal number of multiplications by removing redundancy from the DFT calculation. The MATLAB implementation of the FFT automatically chooses the most efficient way to perform the calculation based on the length N selected for the transform.

Fourier Analysis Using MATLAB

Assume you have a signal $x[n]$ in the vector \mathbf{x} . The DFT of the signal is computed and stored in \mathbf{X} by entering $\mathbf{X} = \mathbf{fft}(\mathbf{x})$. The first value, $\mathbf{X}(1)$, has the value of a_0 , corresponding to the “average” of the signal.

The best way to understand how to interpret the results you obtain using MATLAB is via example. You can run the examples by simply typing the filename indicated. We suggest that you explore and understand these examples prior to attempting the laboratory assignment.

E_4_1.m: In this example, the DFT of a sum of two cosines is displayed as a function of frequency. Centering about an integer index of zero is accomplished using **fftshift**.

E_4_2.m: In this example, the DFT of a length- N vector \mathbf{x} is computed using different length DFTs. The vector \mathbf{x} represents the DT signal $x[n]$ obtained by sampling the signal $x(t) = \cos(20\pi t)$ every nT_s seconds, where $T_s = 0.01$ s. The DFTs computed using different lengths N , corresponding respectively to (1) five periods of data, (2) five periods of data padded with zeros, and (3) 4.5 periods of data, are computed. You should examine the results and note the impact of the DFT length. When you are familiar with Fourier transform properties, you may wish to revisit this example and see if you can analytically understand these differences.

Fourier Series Analysis of Real Signals40

The Fourier series is a good analytical tool, and students always get to solve interesting problems like finding the Fourier series of a square wave or a triangle wave. What about using the Fourier series on a real signal?

Let's look at analyzing the vibrations of an acoustic guitar string. If a string is plucked at a single point, we expect to see a triangular shape appear on the string, with the highest point occurring where the string was plucked. By doing some analysis of this ideal vibrating string, the Fourier Series coefficients are

$$C_n = \frac{L^2 h}{2n^2 \pi^2} \sin\left(\frac{n\pi}{L}\right)$$

where L is the proportion of the string length from the point where the string was plucked to the bridge (e.g., $1/5$) and h is the initial displacement. It should be noted that the C_n values fall off at $1/n^2$, with zeros at the L th harmonic. Since the pluck point occurs at $1/L$, no energy is imparted to that particular frequency (and its harmonics). In Figure 4.4.2 the string shape for a pluck point of $1/4$ is shown (with a period of 225 samples).

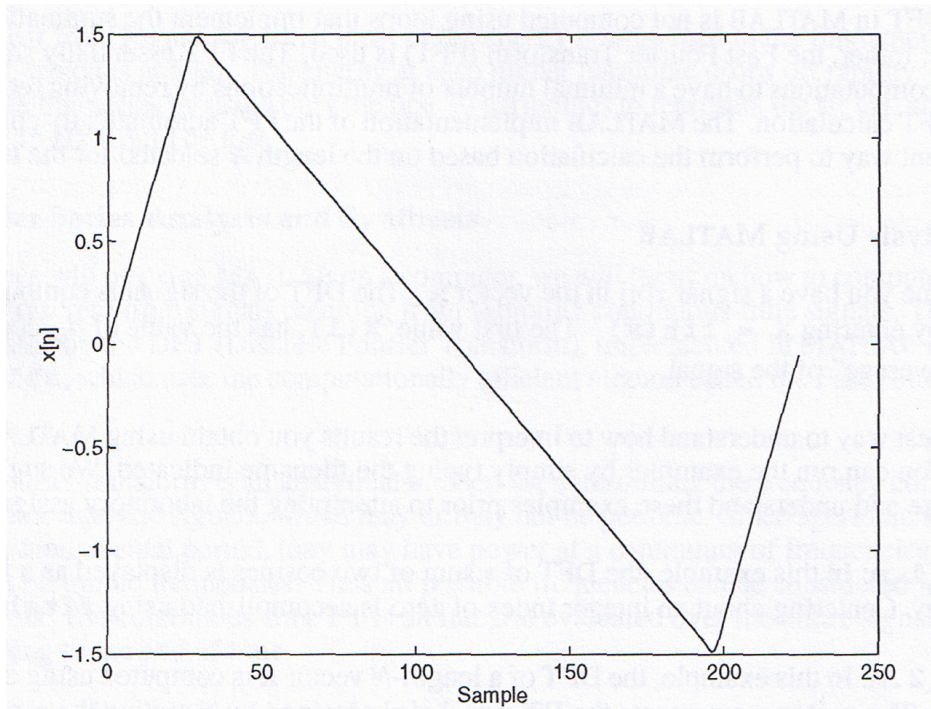


Figure 4.4.2 Ideal Plucked String, One Period

If we sample the sound resulting from plucking an acoustic guitar string, however, we will not see a triangular waveform. Instead, we will see a waveform that looks like it could be a triangle wave, but is smoothed out and has some distortions and bumps, as shown in Figure 4.4.3.

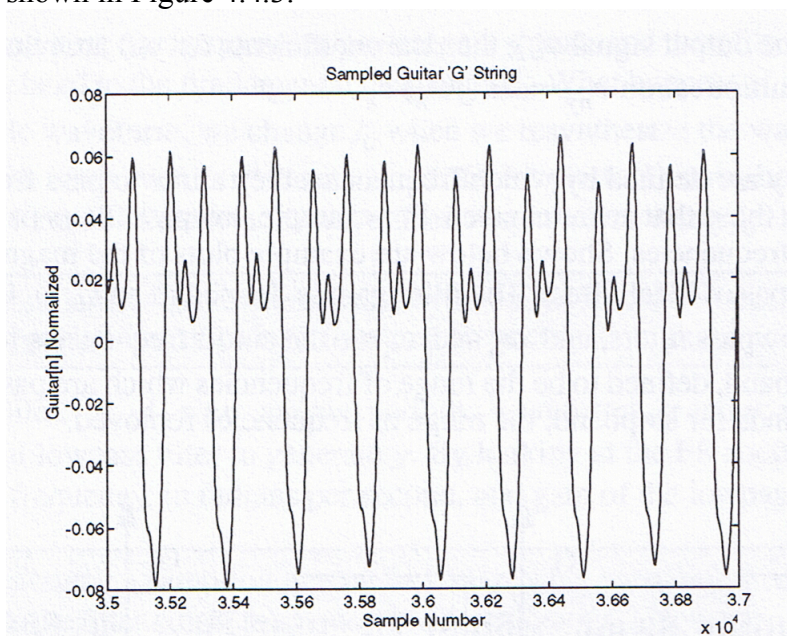


Figure 4.4.3 A Sampled Guitar String

This signal is still periodic, so we can find a Fourier series representation for it and compare it to the Fourier series for an ideal plucked string to see what impact the guitar

body has on the waveform. In order to do this, we have to extract a single period of the signal. Once we have that, we can apply **fft** to find the FS coefficients.

We can then pose the question “How does the body change the sound of a plucked string from that of an ideal?” Equivalently we could ask, “How does the guitar body modify the harmonics of the plucked string?” The impact of a system on the frequency content of a signal is called its frequency response, and the process of modifying this frequency content is called filtering.

Filtering Operations

To determine the frequency response of the guitar, we are interested in how the magnitudes of the input FS coefficients are modified by multiplicative gains, and how the phases of the FS coefficients are modified by additive terms. If we assume that the effect of the guitar body can be roughly approximated by a CT linear, time-invariant system, then the FS for the periodic output signal $y(t)$ can only contain the same harmonic frequencies as the input $x(t)$; thus it can be expressed as the Fourier series given below, where the a_k are the FS coefficients and T_0 is the fundamental period of $x(t)$

$$y(t) = H(j0)a_0 + 2 \sum_{n=1}^{\infty} |H(j\omega_n)| |a_n| \cos\left(\frac{2\pi n}{T}t + \angle a_n + \angle H(j\omega_n)\right)$$

The function $H(j\omega)$ is called the system frequency response. Note that the magnitude and phase of the frequency response, when evaluated at the harmonic frequencies, determine the gain and phase shift, respectively, that are introduced by the system at each harmonic. Thus only the magnitude and phase of the input FS terms will be modified by the filter, and the amount of the modification depends upon the frequency. Note that if the equation above is viewed as the Fourier series for the output signal $y(t)$, then the coefficients for $y(t)$ are related to those for $x(t)$ by the complex multiplication $c_{ny} = H(j\omega_n)c_n$

Filters generally are defined by which frequencies they allow to pass from input to output, or equivalently by those that are attenuated. For example, lowpass filters pass low frequencies and attenuate high frequencies. Shown below are example plots of the magnitude $|H(j\omega)|$ vs. ω for the different types of ideal filters. The filter gain is defined to be G , ω_c is the cutoff frequency for high- and lowpass filters, and ω_h and ω_l are the cutoff frequencies for bandpass filters. PB stands for passband, defined to be the range of frequencies which are passed through to the output, and SB stands for stopband, the range of frequencies removed.

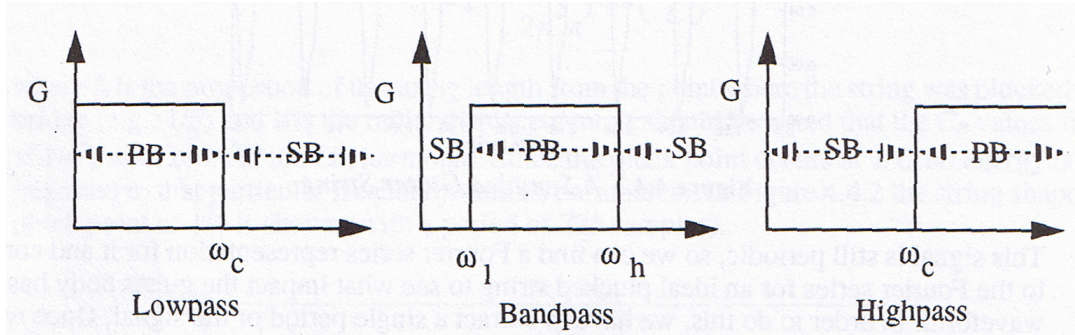


Figure 4.4.4 ideal filters $|H(j\omega)|$ vs. ω

For the ideal filters shown, the magnitude of the filter frequency response, $|H(j\omega)|$ is equal to the gain G in the passband and zero in the stopband. The phase of the frequency response is assumed to be zero, which implies that the filter does not cause any time shift.

4.5 LABORATORY EXPERIMENT

Evaluating the DFS and IDFS

Problem 1. Write a MATLAB function to evaluate the DFS of a signal using E ft. Test your function by evaluating the FS coefficients of both a 50-Hz square wave and a 50-Hz sine wave. Use an 8 kHz sampling rate and a value of N corresponding to exactly 1 period.

Plot the FS coefficients vs. frequency. Verify that your results are correct by comparing your results to the expected analytical results.

Problem 2. Write a MATLAB function that can synthesize a signal from FS coefficients given the fundamental frequency. Attempt to generate the signals from Problem 1 using your function. You will probably need to use **real** if you use **ifft** to remove small imaginary components generated by roundoff.

What happens as you increase the number of FS coefficients used to synthesize the signal? Does this observation make sense given your theoretical understanding of FS?

Problem 3. We know that the frequencies present in a signal, and therefore in its FS, are harmonically related to the fundamental frequency f_0 . What happens if, after analyzing a given periodic waveform, we change f_0 when we resynthesize the waveform to essentially replicate the waveform “shape” at a different pitch? Determine the first 12 FS coefficients for a 50-Hz sawtooth wave, then resynthesize using $f_0 = 100$ Hz.

What similarities and differences do you observe between the original and resynthesized signals? Consider both time and frequency domains.

Problem 4. In the file **P_4_4.mat** are two periodic waveforms, **x** and **y**; **x** has been filtered with an ideal lowpass filter to generate **y**. By looking at the FS coefficients, determine the cutoff frequency, in radians per second, and gain of the lowpass filter.

If you assume a sampling rate of 44.1 kHz, what harmonic frequencies are present? What is the filter cutoff frequency as a function of f_0 ? How exact is your estimate of the cutoff frequency (ie, in what range of values could it be)?

Problem 5. Construct a MATLAB function that plots the magnitude of the Fourier series coefficients for a periodic square wave as a function of frequency. Normalize by dividing the magnitude of all coefficients by a_0 . Fix the width of the rectangular pulses to be $A = 1$ s, and let the period T be a variable. Assume a sampling rate of 100Hz. Generate the plot for $T = 2$ s. Repeat the process several times, doubling the period each time.

Plot your results using the same frequency scale. What do you observe? What might be the implications of these results on the accuracy of the method you used in Problem 3 to resynthesize using a different fundamental frequency?

Some Aspects of DFS

Problem 6. Analyzing Acoustic Guitar Data

In **P_4_6.mat**, a waveform corresponding to an ideal plucked string is stored in vector **x_ideal**. The signal resulting from sampling the sound of an acoustic guitar in response to a plucked string is stored in **x_sampled**. Compute the Fourier series coefficients for **x_ideal**, and contrast the results to the mathematical formula on page 40. Determine the pluck point L . Compute the Fourier series coefficients for **x_sampled**, and compare the Fourier series coefficients of **x_ideal** and **x_sampled**. Calculate the change in magnitude and phase for each coefficient from ideal to sampled data, and plot these values. What do these tell you about the frequency response of the guitar body? Create a function that synthesizes the sampled guitar sound from the FS coefficients for the ideal waveform, given the magnitude and phase changes found above.

If you had to design a filter that implemented the changes, how would you do it? For which pitches (ie, at what frequencies) can you be sure that your filter accurately models the guitar's response?

Problem 7. Creating Beethoven's Fifth Using Fourier Synthesis

In Lab 3, you synthesized Beethoven's Fifth Symphony by concatenating sine waves of different frequencies. Recreate the first four notes of this musical score by pitch-shifting the sampled guitar string using the method described in Problem 3 and the Fourier series coefficients that you computed in Problem 6.

Do the notes you've created sound better than the times you used before? Does it sound like it was played on a guitar? Why or why not?

Problem 8. Identifying Filter Frequency Responses

In **P_4_8.mat**, there are two periodic waveforms, **x** and **y**; **x** has been filtered, creating **y**. From the FS coefficients of **x** and **y**, determine as much as possible about the frequency response of the filter. Try changing the FS components of **x** with the frequency response you've found, and compare the results to **y**.

How much do the FS coefficients tell you - do you know the entire frequency response of the filter? How accurate is your determination of the response?