Laboratory 4

Fourier Series and the DFT

4.1 Introduction

As emphasized in the previous lab, sinusoids are an important part of signal analysis. We noted that many signals that occur in the real world are composed of sinusoids. For example, many musical signals can be approximately described as sums of sinusoids, as can some speech sounds (vowels in particular). It turns out that any periodic signal can be written exactly as a sum of amplitude-scaled and phase-shifted sinusoids. Equivalently, we can use Euler's inverse formulas to write periodic signals as sums of complex exponentials. This is a mathematically more convenient description, and the one that we will adopt in this laboratory and, indeed, in the rest of this course. The description of a signal as a sum of sinusoids or complex exponentials is known as the *spectrum* of the signal.

Why do we need another representation for a signal? Isn't the usual *time-domain* representation enough? It turns out that spectral (or *frequency-domain*) representations of signals have many important properties. First, a frequency-domain representation may be simpler than a time-domain representation, especially in cases where we cannot write an analytic expression for the signal. Second, a frequency-domain representation of a signal can often tell us things about the signal that we would not know from just the time-domain signal. Third, a signal's spectrum provides a simple way to describe the effect of certain systems (like *filters*) on that signal. There are many more uses for frequency-domain representations of a signal, and we will examine many of them throughout this course. Spectral representations are one of the most central ideas in signals and systems theory, and can also be one of the trickiest to understand.

Consider the following problem. Suppose that we have a signal that is actually the sum of two different signals. Further, suppose that we would like to separate one signal from the other, but the signals overlap in time. If the signals have frequency-domain representations that do not overlap, it is still possible to separate the two signals. In this way, we can see that frequency-domain representations provide another "dimension" to our understanding of signals.

In this laboratory, we will examine two tools that allow us to use spectral representations. The *Fourier Series* is a tool that we use to work with spectral representations of periodic continuous-time signals. The *Discrete Fourier Transform* (DFT) is an analogous tool for periodic discrete-time signals. Each of these tools allow both *analysis* (the determination of the spectrum of the time-domain signal) and *synthesis* (the reconstruction of the time-

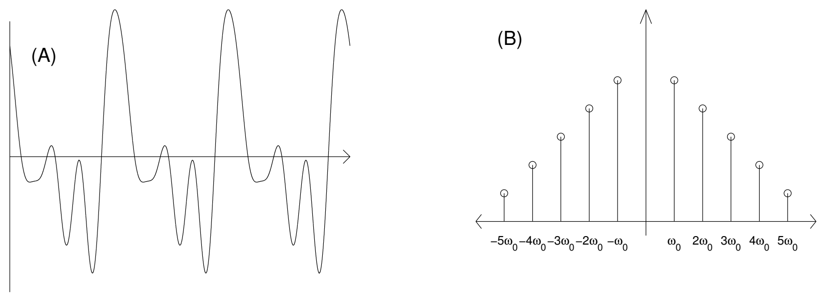

Figure 4.1: (A) A time-domain representation of a signal. (B) A frequency-domain representation of the same signal produced with the Fourier Series.

domain signal from its spectrum). Though you may not be aware of it, you have already performed DFT analysis; the "frequency, amplitude, and phase estimator" system that you implemented in Laboratory 3 actually performs DFT analysis.

4.1.1 "The Questions"

- How can we determine the spectral content of signals?
- How can we separate signals that overlap in time?

4.2 Background

4.2.1 Frequency-domain representations

This section provides an overview to the Fourier series approach of the frequency-domain representation of continuous-time signals.

So far, we have typically thought of signals as time-varying quantities, like $s(t)$. When we plot these signals, we generally place time along the horizontal axis and signal value along the vertical axis. The idea behind the frequency-domain representation of a signal is similar. Rather than plotting signal value versus time, we plot a spectral value versus *frequency*. Doing this involves a *transformation* of the signal. Figure 4.1 shows an example of a time-domain and frequency-domain representation of a signal. Note that we can think of the result of the transform as a signal as well, a signal whose independent variable is frequency rather than time.

The frequency domain representation of a signal (i.e., its *spectrum*) is easy to construct when the signal is composed of a sum of simple complex exponential signals. In this case, the spectrum consists of a few isolated *spectral lines* ("spikes") on the frequency axis *at the frequencies of those complex exponentials*. These spectral lines are complex-valued, and their magnitudes and angles equal the amplitudes and phases of the corresponding complex exponentials. Alternatively, we may draw two separate spectral line plots — one showing the magnitude and the other showing their angles.

If we add more complex exponentials to our signal, then we simply add more spectral lines to its frequency-domain representation. Eventually, if we add enough complex exponentials (possibly an infinite number), we can create *any* signal that we might want. This includes signals that do not look very sinusoidal, like square waves and sawtooth waves. We will use this result for periodic signals in this laboratory assignment.

4.2.2 Periodic Continuous-Time Signals — The Fourier Series

Suppose that we have a periodic continuous-time signal $s(t)$ with period T seconds. We have claimed that *any* such signal can be represented as a sum of complex exponential signals. We now assert that these complex exponentials have harmonically related frequencies. Specifically, their frequencies (in radians per second) form a *harmonic series*

$$\ldots, -3\omega_0, -2\omega_0, -\omega_0, 0, \omega_0, 2\omega_0, 3\omega_0, \ldots \ , \tag{4.1}$$

where

$$\omega_0 \ = \ \frac{2\pi}{T} \tag{4.2}$$

is the *fundamental frequency.* The frequency $k\omega_0$, $k \geq 2$, is called the k-th *harmonic* of the fundamental frequency, or the k-th harmonic frequency for short.

Next we assert that the representation of $s(t)$ in terms of complex exponentials with these frequencies is given by the *Fourier Series synthesis formula*[1]:

$$\begin{aligned} s(t) &= \ldots \alpha_{-2} e^{j\frac{2\pi(-2)}{T}t} + \alpha_{-1} e^{j\frac{2\pi(-1)}{T}t} + \alpha_0 e^{j\frac{2\pi 0}{T}t} + \alpha_1 e^{j\frac{2\pi 1}{T}t} + \alpha_2 e^{j\frac{2\pi 2}{T}t} + \ldots \\ &= \sum_{k=-\infty}^{\infty} \alpha_k e^{j\frac{2\pi k}{T}t} \ , \end{aligned} \tag{4.3}$$

where the α_k's, which are called *Fourier coefficients.* The Fourier coefficients are determined by the *Fourier series analysis formula*

$$\alpha_k = \frac{1}{T} \int_{\langle T_0 \rangle} s(t) e^{-j\frac{2\pi k}{T}t} dt \ , \tag{4.4}$$

where $\int_{\langle T \rangle}$ indicates an integral over any T second interval[2]. In other words, the Fourier synthesis formula shows that the complex exponential component of $s(t)$ at frequency $\frac{2\pi k}{T}$ is

$$\alpha_k e^{j\frac{2\pi k}{T}t} \ . \tag{4.5}$$

Similarly, the Fourier analysis formula shows how the complex exponential components can be determined from $s(t)$, even when no exponential components are evident.

In general, the Fourier coefficients, i.e. the α_k's, are complex. Thus, they have a magnitude $|\alpha_k|$ and a phase or angle $\angle\alpha_k$. The magnitude $|\alpha_k|$ can be viewed as the strength of the exponential component at frequency $k\omega_0 = 2\pi k/T$, while the angle $\angle\alpha_k$ gives the phase of that component. The coefficient α_0 is the *DC term*; it measures the average value of the signal over one period.

Once we know the α_k's, the spectrum of $s(t)$ is simply a plot consisting of spectral lines at frequencies $\ldots, -2\omega_0, -\omega_0, 0, \omega_0, 2\omega_0, \ldots$. The spectral line at frequency $k\omega_0$ is drawn with height indicating the magnitude $|\alpha_k|$ and is labeled with the complex value of α_k.

[1] This is the *exponential form* of the Fourier series synthesis formula. There is also a *sinusoidal form*, which is presented later in this section.

[2] Because $s(t)e^{-j\frac{2\pi k}{T}t}$ is periodic with period T, this integral evaluates to the same value for any interval of length T.

Alternatively, two separate spectral line plots can be drawn — one showing the $|\alpha_k|$'s and the other showing the $\angle\alpha_k$'s.

Notice that the Fourier synthesis formula is very similar to the formula given in Lab 3 for the correlation between a sinusoid and a complex exponential. Indeed it has the same interpretation: in computing α_k we are computing the correlation[3] between the signal $s(t)$ and a complex exponential with frequency $2\pi k/T$. Thought of another way, this correlation gives us an indication of *how much* of a particular complex exponential is contained in the signal $s(t)$.

Partial Series

Notice the infinite limits of summation in the synthesis formula (4.3). This tells us that, for the general case, we need an infinite number of complex exponentials to represent our signal. However, in practical situations, such as in this lab assignment, when we use the synthesis formula to determine signal values, we can generally only include a finite number of terms in the sum. For example, if we use only the first N positive and negative frequencies plus the DC term (at $k = 0$), our approximate synthesis equation becomes

$$s(t) \approx \sum_{k=-N}^{N} \alpha_k e^{j\frac{2\pi k}{T}t} \ . \tag{4.6}$$

Fortunately, Fourier series theory shows that this approximation becomes better and better[4] as $N \longrightarrow \infty$. Alternatively, it is known that the mean-squared value of the difference between $s(t)$ and the approximation tends to zero as $N \longrightarrow \infty$. Specifically, it can be shown that

$$\begin{aligned} MS\left(s(t) - \sum_{k=-N}^{N} \alpha_k e^{j\frac{2\pi k}{T}t}\right) &= MS(s(t)) - \sum_{k=-N}^{N} |\alpha_k|^2 \\ &\longrightarrow 0 \text{ as } N \longrightarrow \infty \ . \end{aligned} \tag{4.7}$$

How large must N be for the approximation to be good? There is no simple answer. However, you will gain some idea by the experiments you perform in this lab assignment.

***T*-Second Fourier Series**

If a signal $s(t)$ is periodic with period T, then it is also periodic with period $2T$, and period $3T$, and so on. Thus when applying Fourier series, we have a choice as to the value of T. Often, we will choose T to be the smallest period, i.e. the *fundamental period* of $s(t)$. However, there are also situations where we will not. For example, suppose we wish to perform spectral analysis/synthesis of two or more periodic signals that have different fundamental periods. We could of course form a separate Fourier series for each signal. In this case, each Fourier series will be based on a different harmonic series of frequencies. Wouldn't it be nicer if we could base each series on a common harmonic series of frequencies? We can do this by choosing T to be a multiple of the fundamental periods of both signals.

[3] Actually, here we are computing what we called the *length-normalized correlation*.

[4] It is known that under rather benign assumptions about the signal $s(t)$, the approximation converges to $s(t)$ as $N \longrightarrow \infty$ at all times t where $s(t)$ is continuous, and at times t where $s(t)$ has a jump discontinuity, the approximation converges to the average of the values immediately to the left and right of the discontinuity.

When we want to explicitly specify the value of T that is used in a Fourier series, we will say *T-second Fourier series*. What then is the relationship between Fourier series corresponding to different values of T? To see what is happening, let us compare a T-second Fourier series to a $2T$-second Fourier series. The T-second Fourier series has components at the frequencies

$$\ldots, -2\omega_0, -\omega_0, 0, \omega_0, 2\omega_0, \ldots \ , \tag{4.8}$$

where

$$\omega_0 = \frac{2\pi}{T} \tag{4.9}$$

and the $2T$-second Fourier series has components at the frequencies.

$$\ldots, -2\omega_0', -\omega_0', 0, \omega_0', 2\omega_0', \ldots \ = \ \ldots, -\omega_0, -\frac{\omega_0}{2}, 0, \frac{\omega_0}{2}, \omega_0, \ldots \ , \tag{4.10}$$

where

$$\omega_0' = \frac{2\pi}{2T} = \frac{\omega_0}{2}. \tag{4.11}$$

From this we see that the $2T$-second Fourier series decomposes $s(t)$ into frequency components with half the separation of that of the T-second Fourier series. However, since $s(t)$ is periodic with period T, its spectrum is actually concentrated at frequencies that are multiples of ω_0 (or a subset thereof). Hence, the "additional" coefficients in the $2T$-Fourier series must be zero, and it turns out that the nonzero coefficients are the same as for the T-second Fourier series. Specifically, it can be shown that with α_k and α_k' denoting the T-second and $2T$-second Fourier coefficients, respectively, then

$$\alpha_k' = \begin{cases} \alpha_{k/2}, & k \text{ even} \\ 0, & k \text{ odd} \end{cases} \tag{4.12}$$

In summary, Fourier series analysis/synthesis can be performed over one fundamental period or over any number of fundamental periods. Usually, when Fourier series is mentioned, the desired number of periods interval will be clear from context. In any case, the spectrum is not affected by the choice of T.

Aperiodic Continuous-Time Signals

Next, we briefly discuss how Fourier series can also be applied when the signal $s(t)$ is not periodic. In this case, we can nevertheless determine the spectrum of a finite *segment* of the signal, say from time t_1 to time t_2, by performing Fourier series analysis/synthesis on just this segment. That is, if we find Fourier coefficients

$$\alpha_k = \frac{1}{T} \int_{t_1}^{t_2} s(t) e^{-j\frac{2\pi k}{T}t} dt \ , \tag{4.13}$$

where $T = t_2 - t_1$, then we have

$$s(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{j\frac{2\pi k}{T}t} \ , \ \text{for } t_1 \le t \le t_2 \ . \tag{4.14}$$

This will give us an idea of the frequency content of the signal during the given time interval. It is important to emphasize, however, that the synthesis equation (4.14) is valid *only* when t is between t_1 and t_2. Outside of this time interval, the synthesis formula will not necessarily equal $s(t)$. Instead, it describes a signal that is periodic with period T, called the *periodic extension* of the segment between t_1 and t_2.

Properties of the Fourier Coefficients

We conclude our discussion of the Fourier series with a list of useful properties, some of which have already been mentioned. A few of these will be useful in this lab assignment. The rest are included for completeness. These properties are stated without derivations. However, each can be derived straightforwardly from the analysis and synthesis formulas. Though not required in this laboratory, you may want to confirm some of these properties using the Fourier analysis and synthesis programs described in Section 4.3.

1. (Fourier series analysis) The T-second Fourier series analysis of a periodic signal $s(t)$ with period T produces a set of Fourier coefficients α_k, $k = \ldots, -2, -1, 0, 0, 1, 2, \ldots$, which are, in general, complex valued.

2. (Frequency components) If α_k are the coefficients of the T-second Fourier series of the periodic signal $s(t)$ with period T, then the frequency or spectral component of $s(t)$ at frequency $\frac{2\pi k}{T}$ is $\alpha_k e^{j\frac{2\pi k}{T}t}$.

3. (DC component) The coefficient α_0 equals the average or DC value of $s(t)$.

4. (One-to-one relationship) There is a one-to-one relationship between periodic signals and Fourier coefficients. Specifically, if $s(t)$ and $s'(t)$ are distinct[5] periodic signals, each periodic with period T, then their T-second Fourier coefficients are not entirely identical, i.e. $\alpha_k \neq \alpha'_k$ for at least one k. It follows that one can recognize a periodic signal from its Fourier coefficients (and its period).

5. (Conjugate symmetry) If $s(t)$ is a real-valued signal, i.e. its imaginary part is zero, then for any integer k

$$\alpha_{-k} = \alpha_k^* \tag{4.15}$$
$$|\alpha_{-k}| = |\alpha_k| \tag{4.16}$$
$$\angle\alpha_{-k} = -\angle\alpha_k \ . \tag{4.17}$$

6. (Conjugate pairs) If α_k's are the T-second Fourier coefficients for a real-valued signal $s(t)$, then for any k the sum of the complex exponential components of $s(t)$ corresponding to α_k and α_{-k} is a sinusoid at frequency $2\pi k/T$. Specifically, using the inverse Euler relation,

$$\alpha_k e^{j\frac{2\pi k}{T}t} + \alpha_{-k} e^{-j\frac{2\pi k}{T}t} = 2|\alpha_k|\cos(\frac{2\pi k}{T}t + \angle\alpha_k) \ . \tag{4.18}$$

7. (Sinusoidal form of the Fourier synthesis formula) The previous property leads to the sinusoidal form of the Fourier synthesis formula:

$$s(t) = \alpha_0 + \sum_{k=-\infty}^{\infty} 2|\alpha_k|\cos(\frac{2\pi k}{T}t + \angle\alpha_k) \ . \tag{4.19}$$

[5]By "distinct", we mean that $s(t)$ and $s'(t)$ are sufficiently different that $s(t) \neq s'(t)$ for all times t in some interval with (t_1, t_2), with nonzero length. They are *not* "distinct" if they differ only at a set of isolated points. To see why we need this clarification, observe that if $s(t)$ and $s'(t)$ differ only at time t_1, then they have the same Fourier coefficients, because integrals, such as those defining Fourier coefficients, are not affected by changes to their integrands at isolated points. Likewise, $s(t)$ and $s'(t)$ will have the same Fourier coefficients if they differ only at isolated times $t_1, t_2, \ldots$. However, if $s(t) \neq s'(t)$ for all t in an entire interval, no matter how small, then $\alpha_k \neq \alpha'_k$ for at least one k.

8. (Linear combinations) If $s(t)$ and $s'(t)$ have T-second Fourier coefficients α_k and α'_k, respectively, then $as(t) + bs'(t)$ has T-second Fourier coefficients $a\alpha_k + b\alpha'_k$.

9. (Fourier series of elementary signals) The following lists the T-second Fourier coefficients of some elementary signals.

   (a) Complex exponential signal: $s(t) = e^{j\frac{2\pi m}{T}t} \quad \Longrightarrow$

   $$\alpha_k = \begin{cases} 1, & k = m \\ 0, & k \neq m \end{cases} . \tag{4.20}$$

   (b) Cosine: $s(t) = \cos(\frac{2\pi m}{T}t) \quad \Longrightarrow$

   $$\alpha_k = \begin{cases} \frac{1}{2}, & k = \pm m \\ 0, & k \neq \pm m \end{cases} . \tag{4.21}$$

   (c) Sine: $s(t) = \sin(\frac{2\pi m}{T}t) \quad \Longrightarrow$

   $$\alpha_k = \begin{cases} -\frac{j}{2}, & k = m \\ \frac{j}{2}, & k = -m \\ 0, & k \neq \pm m \end{cases} . \tag{4.22}$$

   (d) General sinusoid: $s(t) = \cos(\frac{2\pi m}{T}t + \phi) \quad \Longrightarrow$

   $$\alpha_k = \begin{cases} \frac{1}{2}e^{j\phi}, & k = m \\ \frac{1}{2}e^{-j\phi}, & k = -m \\ 0, & k \neq \pm m \end{cases} . \tag{4.23}$$

10. (T-second Fourier series) If a periodic signal $s(t)$ has period T and T-second Fourier coefficients α_k, then the nT-second Fourier coefficients are

$$\alpha'_k = \begin{cases} \alpha_{k/n}, & k = \text{ multiple of } n \\ 0, & \text{else} \end{cases} \tag{4.24}$$

11. (Parseval's relation) If α_k's are the T-second Fourier coefficients for signal $s(t)$, then the mean-squared value of $s(t)$, equivalently the power, equals the sum of the squared magnitudes of the Fourier coefficients. That is,

$$MS(s) = \frac{1}{T}\int_{\langle T\rangle} |s(t)|^2\, dt = \sum_{k=-\infty}^{\infty} |\alpha_k|^2 \tag{4.25}$$

12. (Uncorrelatedness/orthogonality of complex exponentials) The T-second correlation between complex exponential signals $e^{j\frac{2\pi m}{T}t}$ and $e^{j\frac{2\pi n}{T}t}$, $m \neq n$, is zero. This property is used in the derivation of the previous and other properties.

4.2.3 Periodic Discrete-Time Signals — The Discrete Fourier Transform

This section overview the discrete Fourier transform approach to the frequency-domain representation of discrete-time signals.

Consider a periodic discrete-time signal $s[n]$ with period N. As with continuous-time signals, we wish to find its frequency-domain representation, i.e. its spectrum. That is, we wish to represent $s[n]$ as a sum of *discrete-time* complex exponential signals. Again, by analogy to the continuous-time case we will use frequencies that are multiples of

$$\widehat{\omega}_0 = \frac{2\pi}{N} \ . \tag{4.26}$$

However, unlike the continuous-time case, we now use only a finite number of such frequencies. Specifically, we use the N harmonically related frequencies:

$$0, \widehat{\omega}_0, 2\widehat{\omega}_0, \ldots, (N-1)\widehat{\omega}_0 \ . \tag{4.27}$$

The reason is that any complex exponential signal with the frequency $k\widehat{\omega}_0$ is in fact identical to a complex exponential signal with one of the N frequencies listed above[6]. Notice that this set of frequencies ranges from 0 to $\frac{2\pi(N-1)}{N}$, which is just a little less than 2π.

We now assert that the representation of $s[n]$ in terms of complex exponentials with the above frequencies is given by the *discrete-time Fourier series synthesis formula* or as we will usually call it, the *the Discrete Fourier Transform (DFT) synthesis formula*

$$\begin{aligned} s[n] &= S[0]e^{j\frac{2\pi 0}{N}n} + S[1]e^{j\frac{2\pi 1}{N}n} + S[2]e^{j\frac{2\pi 2}{N}n} + \ldots + S[N-1]e^{j\frac{2\pi(N-1)}{N}n} \\ &= \sum_{k=0}^{N-1} S[k]e^{j\frac{2\pi k}{N}n} \ , \end{aligned} \tag{4.28}$$

where the $S[k]$'s, which are called *DFT coefficients*, are determined by the *DFT analysis formula*

$$S[k] = \frac{1}{N}\sum_{\langle N\rangle} s[n]e^{-j\frac{2\pi k}{N}n} \ , \qquad k = 0, 1, 2, 3, \ldots, N-1 \tag{4.29}$$

where $\langle N\rangle$ indicates a sum over any N consecutive integers[7], e.g. the sum over $0, \ldots, N$.

As with the continuous-time Fourier series, the DFT coefficients are, in general, complex. Thus, they have a magnitude $|S[k]|$ and a phase or angle $\angle S[k]$. The magnitude $|S[k]|$ can be viewed as the strength of the exponential component at frequency $k\widehat{\omega}_0 = 2\pi k/N$, while $\angle S[k]$ is the phase of that component. The coefficient $S[0]$ is the *DC term*; it measures the average value of the signal over one period.

Once we know the $S[k]$'s, the spectrum of $s[n]$ is simply a plot consisting of spectral lines at frequencies $0, \widehat{\omega}_0, 2\widehat{\omega}_0, \ldots, (N-1)\widehat{\omega}_0$. The spectral line at frequency $k\widehat{\omega}_0$ is drawn with height indicating the magnitude $|S[k]|$ and is labeled with the complex value of $S[k]$.

[6] If $k\widehat{\omega}_0$ is not in this range, then $k = mN + l$ where $m \neq 0$ and $0 \leq l < N$. It then follows that the complex exponential with this frequency is $e^{j\frac{2\pi k}{N}n} = e^{j\frac{2\pi(mN+l)}{N}n} = e^{j2\pi mn}e^{j\frac{2\pi l}{N}n} = e^{j\frac{2\pi l}{N}n}$, which is an exponential with one of the N frequencies in the list above.

[7] Because $s[n]e^{-j\frac{2\pi k}{N}n}$ is periodic with period N, the sum is the same for any choice of N consecutive integers.

Alternatively, two separate spectral line plots can be drawn — one showing the $|S[k]|$'s and the other showing the $\angle S[k]$'s.

Since the sums in the synthesis and analysis formulas are finite, there are no convergence-of-partial-sum issues, such as those that arise for the continuous-time Fourier series.

Often the DFT coefficients $S[0], \ldots, S[N]$ are said to be the "DFT of the signal $s[n]$" and the process of computing them via the analysis equation (4.29) is called "taking the DFT" of $s[n]$. Conversely, applying the synthesis equation (4.28) is often called "taking the inverse DFT" of $S[0], \ldots, S[N]$.

Notice that the DFT analysis formula (4.29) is identical to equation (3.45) in Lab 3. That is, in computing the set of correlations between a signal $s[n]$ and the various complex exponentials in Lab 3, we were actually taking the DFT of $s[n]$. Indeed, it continues to be helpful to view the DFT analysis as the process of correlating $s[n]$ with various complex exponentials. Those correlations that lead to larger magnitude coefficients indicate frequencies where the signal has larger components.

In some treatments, the DFT analysis and synthesis formulas differ slightly from those given above in that the $1/N$ factor is moved from the analysis formula to the synthesis formula[8], or replaced by a $1/\sqrt{N}$ factor multiplying each formula. All of these approaches are equally valid. The choice between them is largely a matter of taste. For example, our approach is the only one for which $S[0]$ equals the average signal value. For the other approaches, the average is $S[0]$ multiplied by a known constant. The only cautionary note is that one should never use the analysis formula from one version with the synthesis formula from another. In this course, we will always use the analysis and synthesis formulas shown above.

Although we will always take $0, \widehat{\omega}_0, 2\widehat{\omega}_0, \ldots, (N-1)\widehat{\omega}_0$ as the analysis frequencies produced by the DFT, it is important to point out that every frequency $\widehat{\omega}$ in the upper half of this range, i.e. between π and 2π, is equivalent to a frequency $\widehat{\omega} - 2\pi$, which lies between $-\pi$ and 0. By "equivalent," we mean that a complex exponential with frequency $\widehat{\omega}$ with $\pi < \widehat{\omega} < 2\pi$ equals the complex exponential with frequency $\widehat{\omega} - 2\pi$. Thus, it is often useful to think of frequencies in the upper half of our designated range as representing frequencies in the range $-\pi$ to 0.

For example, let us look at the DFT of a sinusoidal signal, $s[n] = \cos(\frac{2\pi m}{N} n)$, with $0 < m < \frac{N}{2}$. The DFT coefficients, $S[k]$, are given by

$$(S[0], \ldots, S[N-1]) = (0, \ldots, 0, 1/2, 0, \ldots, 0, 1/2, 0, \ldots, 0), \tag{4.30}$$

where $S[m] = S[N-m] = 1/2$ and $S[k] = 0$ for other k's. In the synthesis formula, the coefficient $S[m]$ multiplies the complex exponential $e^{j\frac{2\pi m}{N}n}$, and the coefficient $S[N-m]$ multiplies the complex exponential $e^{j\frac{2\pi(N-m)}{N}n} = e^{-j\frac{2\pi m}{N}n}$. Thus, these two coefficients can be viewed as multiplying exponentials at frequencies $\pm\frac{2\pi m}{N}$, which by the inverse Euler formula sum to yield $s[n] = \cos(\frac{2\pi m}{N} n)$.

N-point DFT

As with continuous-time signals, if a discrete-time signal $s[n]$ is periodic with period N, then it also periodic with period $2N$, and period $3N$, and so on. Thus, when applying the DFT, we have a choice as to the value of N. Sometimes we choose it to be the the smallest

[8] The *DSP First* textbook does this in Chapter 9.

period, i.e. the fundamental period, but sometimes we do not. When we want to explicitly specify the value of N used in a DFT, we will say *N-point DFT*.

The relationship between the N-point and $2N$-point DFT is just like the relationship between the T-second and $2T$-second Fourier series. That is, whereas the N-point DFT has components at frequencies

$$0, \widehat{\omega}_0, 2\widehat{\omega}_0, \ldots, (N-1)\widehat{\omega}_0 \ , \tag{4.31}$$

the $2N$-point DFT has components at the frequencies

$$0, \widehat{\omega}\prime_0, 2\widehat{\omega}_0', \ldots, (2N-1)\widehat{\omega}_0' \ = \ 0, \frac{\widehat{\omega}_0}{2}, \widehat{\omega}_0, \ldots, (2N-1)\frac{\widehat{\omega}_0}{2} \ . \ . \tag{4.32}$$

where

$$\widehat{\omega}_0 \ = \ \frac{2\pi}{2N} \ = \ \frac{\omega_0}{2} \tag{4.33}$$

From this we see that the separation between frequency components has been halved. Moreover, it can be shown that the relationship between the original and new coefficients is

$$S'[k] = \begin{cases} S[k/2], & k \text{ even} \\ 0, & k \text{ odd} \end{cases} \tag{4.34}$$

In summary, DFT analysis/synthesis can be performed over one fundamental period or over any number of fundamental periods. Usually, when the DFT is mentioned, the desired number of periods interval will be clear from context. In any case, the spectrum is not affected by the choice of N.

Aperiodic Discrete-Time Signals

Next, we briefly discuss how the DFT can also be applied when the signal $s[n]$ is not periodic. In this case, we can nevertheless determine the spectrum of a finite *segment* of the signal, say from time n_1 to time n_2, by performing DFT analysis/synthesis on just this segment. That is, if we find DFT coefficients

$$S'[k] = \frac{1}{N} \sum_{\langle N \rangle} s[n] e^{-j\frac{2\pi k}{N}n} \ , \qquad k = 0, 1, 2, 3, \ldots, N-1 \tag{4.35}$$

where $N = n_2 - n_1$, then we have

$$s[n] = \sum_{k=0}^{N-1} S'[k] e^{j\frac{2\pi k}{N}n} \ , \qquad k = 0, 1, 2, 3, \ldots, N-1 \ . \tag{4.36}$$

This will give us an idea of the frequency content of the signal during the given time interval. It is important to emphasize, however, that the synthesis equation (4.36) is valid *only* at times n from n_1 to n_2. Outside of this time interval, the synthesis formula will not necessarily equal $s[n]$. Instead, it describes a signal that is periodic with period N, called the *periodic extension* of the segment from n_1 to n_2.

Approximating Fourier series coefficients with the DFT

Frequently, we are interested in finding the spectrum of some continuous-time signal $s(t)$, but for practical reasons, we sample the signal and work with the resulting discrete-time signal $s[n]]$. Can we find, at least approximately, the spectrum of $s(t)$ by working with the discrete-time signal $s[n]$? As discussed below there is a close relationship between the Fourier series coefficients of $s(t)$ and the DFT of $s[n]$.

Suppose $s(t)$ is periodic with period T, and suppose we sample $s(t)$ with sampling interval $T_s = T/N$, where N is an integer, resulting in the discrete-time signal $s[n] = s(nT_s)$, which is easily seen to be periodic with period N. Let α_k denote the T-second Fourier coefficients of $s(t)$, and let $S[k]$ denote the N-point DFT of $s[n]$. Then it can be shown that if N is very large, then

$$\alpha_k \approx S[k] \ , \ \text{when } k << N \tag{4.37}$$

Moreover, it can be shown that if it should happen that $s(t)$ has no spectral components at frequencies greater than $1/(2T_s)$, then

$$\alpha_k = \begin{cases} S[k], & 0 \le k \le N/2 \\ S[N-k+1], & -N/2 \le k < 0 \\ 0, & |k| > N/2 \end{cases} \tag{4.38}$$

The above two equation shows how the DFT can be used to compute, at least approximately, the Fourier series coefficients. In fact, the Fourier series analysis program described in the in the MATLAB section of this assignment uses the DFT to compute the Fourier coefficients.

Properties of the DFT coefficients

The following are a number of useful properties of the DFT with which you should be familiar. A few of these will be useful in this lab assignment. Others will be used in future assignments. These properties are stated without derivations. However, each can be derived straightforwardly from the analysis and synthesis formulas. Though not required in this laboratory, you may want to confirm some of these properties using the DFT analysis and synthesis programs described in Section 4.3.

1. (DFT analysis) The N-point DFT of a periodic signal $s[n]$ with period N produces a vector of N DFT coefficients $S[0], \ldots, S[N-1]$, which are, in general, complex valued. Equivalently, the coefficients may be considered to be determined by a set of N signal samples.

2. (Frequency components) If $S[k]$ is N-point DFT of the periodic signal $s[n]$ with period N, then the frequency or spectral component of $s[n]$ at frequency $\frac{2\pi k}{N}$ is $S[k]e^{j\frac{2\pi k}{N}n}$. The component of the signal at frequency $\frac{-2\pi k}{N}$ is $S[N-k]e^{-j\frac{2\pi k}{N}N}$.

3. (DC component) The coefficient $S[0]$ equals the average value or DC value of $s[n]$.

4. (One-to-one relationship) There is a one-to-one relationship between discrete-time signals with period N (equivalently, sequences of N signal samples) and sequences of N DFT coefficients. Specifically, if $s[n]$ and $s'[n]$ are distinct periodic signals with period N, i.e. $s[n] \neq s'[n]$ for some value of n, then their N-point DFT coefficients are not entirely identical, i.e. $S[k] \neq S'[k]$ for at least one k. It follows that one can recognize a discrete-time periodic signal from its DFT coefficients (and N).

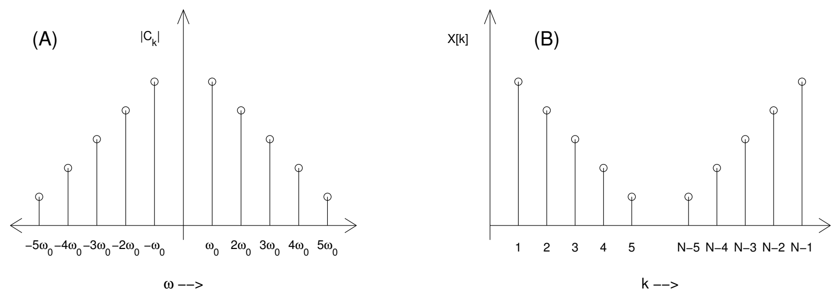

Figure 4.2: (A) The magnitude of the Fourier Series coefficients α_k for a periodic continuous-time signal. (B) The DFT of a periodic discrete-time version of the same signal. Note that the origin for the Fourier Series coefficients is in the middle of the plot, but the origin for the DFT is to the left.

5. (Conjugate symmetry) If $s[n]$ is a real-valued signal, i.e. its imaginary part is zero, then for any integer k

$$S[N-k] = S^*[k] \tag{4.39}$$
$$|S[N-k]| = |S[k]| \tag{4.40}$$
$$\angle S[N-k] = -\angle S[k]\ . \tag{4.41}$$

   These facts indicate that we are usually only interested in the first half of the DFT coefficients. In particular, note that when we plot the DFT, the location of the origin and the appearance of the symmetry is different than when we plot the Fourier Series. See Figure 4.2 for an example of the relation between the two.

6. (Conjugate pairs) If $S[k]$ is the N-point DFT of a real-valued signal $s[n]$, then for any k the sum of the complex exponential components of $s[n]$ corresponding to $S[k]$ and $S[N-k]$ is a sinusoid at frequency $2\pi k/N$. Specifically, using the inverse Euler relation,

$$S[k]e^{j\frac{2\pi k}{N}n} + S[N-k]e^{-j\frac{2\pi k}{N}n} = 2|S[k]|\cos(\frac{2\pi k}{N}n + \angle S[k])\ . \tag{4.42}$$

7. (Linear combinations) If $s[n]$ and $s'[n]$ have N-point DFT $S[k]$ and $S'[k]$, respectively, then $as[n] + bs'[n]$ has N-point DFT $aS[k] + bS'[k]$.

8. (Sampled continuous-time signals) If the discrete-time signal $s[n]$ comes from sampling a continuous-time signal $s(t)$ with sampling interval T_s, i.e. if $s[n] = s(nT_s)$, then the continuous-time frequency represented by DFT coefficient $S[k]$ is $\frac{2\pi k}{N} f_s$, where $f_s = 1/T_s$ samples per second is the sampling rate.

9. (DFT of elementary signals) The following lists the N-point DFT of some elementary signals.

   (a) Complex exponential signal: $s[n] = e^{j\frac{2\pi m}{N}n} \quad \Longrightarrow$

$$(S[0], \ldots, S[N-1]) = (0, \ldots, 0, 1, 0, \ldots, 0)\ , \tag{4.43}$$

   where the nonzero coefficient is $S[m]$.

(b) Cosine: $s[n] = \cos\left(\frac{2\pi m}{N} n\right) \quad \Longrightarrow$

$$(S[0], \ldots, S[N-1]) = (0, \ldots, 0, \frac{1}{2}, 0, \ldots, 0, \frac{1}{2}, 0, \ldots, 0) \ , \tag{4.44}$$

where the nonzero coefficients are $S[m]$ and $S[N-m]$.

(c) Sine: $s[n] = \sin\left(\frac{2\pi m}{N} n\right) \quad \Longrightarrow$

$$(S[0], \ldots, S[N-1]) = (0, \ldots, 0, -\frac{j}{2}, 0, \ldots, 0, \frac{j}{2}, 0, \ldots, 0) \ , \tag{4.45}$$

where the nonzero coefficients are $S[m]$ and $S[N-m]$.

(d) General sinusoid: $s[n] = \cos\left(\frac{2\pi m}{N} n + \phi\right) \quad \Longrightarrow$

$$(S[0], \ldots, S[N-1]) = (0, \ldots, 0, \frac{1}{2}e^{j\phi}, 0, \ldots, 0, \frac{1}{2}e^{-j\phi}, 0, \ldots, 0) \ , \tag{4.46}$$

where the nonzero coefficients are $S[m]$ and $S[N-m]$.

(e) Not quite periodic sinusoid: $s[n] = \cos\left(\frac{2\pi(m+\epsilon)}{N} n\right)$ where $(m+\epsilon)$ is non-integer $\Longrightarrow$ The resulting $S[k]$'s will all be nonzero[9], typically with small magnitudes except those corresponding to frequencies closest to $\frac{2\pi(m+\epsilon)}{N}$.

(f) Period contains unit impulse period: $s[n] = (1, 0, \ldots, 0) \quad \Longrightarrow$

$$(S[0], \ldots, S[N-1]) = \left(\frac{1}{N}, \ldots, \frac{1}{N}\right) \ . \tag{4.47}$$

10. (N-point DFT) If $S[k]$ is the N-point DFT of the periodic signal $s[n]$ with period N, then the mN-point DFT coefficients are

$$S[k] = \begin{cases} S[k/m], & k = \text{ multiple of } m \\ 0, & \text{else} \end{cases} \tag{4.48}$$

11. (Parseval's relation) If $S[k]$ is the N-point DFT of $s[n]$, then

$$MS(x) = \frac{1}{N} \sum_{\langle N \rangle} |s[n]|^2 = \sum_{k=0}^{N-1} |S[k]|^2 \ . \tag{4.49}$$

This shows that the power in the signal $s[n]$ equals the energy of the DFT coefficients.

12. (Uncorrelatedness/orthogonality of complex exponentials) The N-point correlation between complex exponential signals $e^{j\frac{2\pi m}{N} n}$ and $e^{j\frac{2\pi l}{N} n}$, $m \neq l$, is zero. This property is used in the derivation of the previous one.

[9] This is the same effect that you saw in lab 3 when you ran `fape` over a non-integer number of periods of the sinusoid.

4.2.4 Separating Signals Based on Differing Harmonic Series

We've already suggested that there are many nearly-periodic signals that occur in the real world, with two notable examples being many musical signals and vowels in speech signals. These sort of signals can be analyzed using the Fourier Series or the DFT (applied to samples). We will use the DFT, principally because if we wanted to use the Fourier series, we would anyway approximately compute the Fourier coefficients with the DFT. In particular, let us consider a note played on a musical instrument like a flute or clarinet. Such a signal is nearly periodic with some fundamental period. If the note is played at "concert pitch," for instance, it has a fundamental frequency of 440 Hz and a fundamental period of 1/440 seconds. Few musical signals, though, are purely sinusoidal. From our development of the Fourier series, we know that a periodic signal can be described as a sum of complex exponentials (or sinusoids) with harmonically-related frequencies. That is, the spectrum of our musical note is composed of a *harmonic series*. In particular, if the fundamental frequency is 440 Hz, higher harmonics will be at 880 Hz, 1320 Hz, 1760 Hz, and so on. Figure 4.3 shows a stemp plot of the DFT of an example harmonic series.

Suppose that we have two instruments playing different notes (i.e., the two signals have different fundamental periods) at the same time. The signal coming from each instrument is a single harmonic series, but a listener "hears" a signal which is the sum of these two signals. By the linear combination properties of the Fourier Series and DFT, we know that the spectrum of the combined signal is simply the sum of the spectra of the separate signals. We can use this property to separate the two signals in the frequency-domain, even though they overlap in the time-domain.

Suppose that we wish to simply remove one of the notes from the combined signal. We'll assume that we have recorded and sampled the signal, so we're working in discrete-time. We'll also assume that the combined signal is also periodic[10] with some (fairly long) fundamental period N_0. If we take the N_0-point DFT of a segment of the combined signal, we can identify the coefficients that make up each harmonic series. Then, we simply zero-out all of the coefficients corresponding to the harmonics of the note we wish to remove. When we resynthesize the signal with the inverse DFT, the resulting signal will contain only one of the two notes.

[10] In the "real-world," this is a somewhat questionable assumption. However, we can approximate this behavior quite well by simply using a long DFT. In this case, each harmonic may be "spread" over several DFT coefficients, so to remove a harmonic we need to zero-out all of coefficients associated with it. This spreading behavior is the same as what you saw in Lab 3 when running `fape` over non-periodic signals.

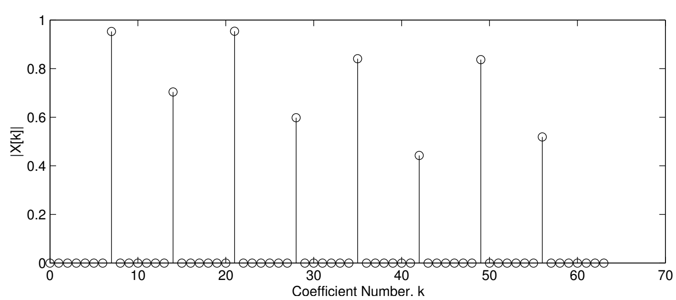

Figure 4.3: The DFT of a harmonic series. Note that only the first half of the DFT coefficients are shown in this figure.

We can extend this procedure to more complicated signals, like melodies with many notes. In this case, we simply analyze and resynthesize each note individually. Of course, with more simultaneously-sounding notes and more complicated music, this procedure becomes rather difficult. In this lab, we will implement this procedure to remove a "corrupting" note held throughout a simple, easily analyzed melody. Though somewhat idealized, the problem should help to motivate the use of the DFT and the frequency domain.

4.3 Some MATLAB commands for this lab

- **Fourier Series Synthesis in** MATLAB**:** The function `fourier_synthesis` is a function that we provide to compute the approximate T-second Fourier series synthesis formula, equation (4.6). Its inputs are the period T and a set of $2N+1$ Fourier coefficients. Its output is the synthesized signal. The calling command is

  ```
  >>   [ss,tt] = fourier_synthesis(CC, T, periods, Ns);
  ```

  where `CC` is a vector containing the Fourier coefficients, `T` is the interval (in seconds) over which the Fourier series is applied. `periods` is the (integer) number of periods to include in the resynthesis; `periods` defaults to a value of 1 if not provided. The optional parameter `Ns` specifies how many samples per period to include in the output signal.

  It is assumed that `CC` contains the coefficients $\alpha_{-N} \ldots \alpha_N$. ($N$ is implicitly determined from the length of `CC`.) Thus, `CC` has length $2N+1$, the `CC(n)` element contains the Fourier series coefficient α_{n-N-1}. Further, note that the α_0 coefficient falls at `CC(N+1)`.

  The two returned parameters are the signal vector `ss` and the corresponding signal support vector `tt`.

- **Fourier Series Analysis in** MATLAB**:** The function `fourier_analysis` is the complement to `fourier_synthesis`. It performs T-second Fourier series analysis on an input signal. The calling command is

  ```
  >> [CC,ww] = fourier_analysis(ss,T,N);
  ```

  where `ss` is a vector containing the signal samples, `T` is the interval T in seconds over which the Fourier series is to be computed, and `N` is the number of positive harmonics to include in the analysis. (`2N+1` is the total number of harmonics.) It is assumed that `ss` contains samples of the signal to be analyzed over the interval $[0, T]$.

  The outputs are the vectors `CC`, which contains the $2N+1$ Fourier coefficients[11], and `ww`, which contains the frequencies (in Hertz) associated with each Fourier coefficient.

- **DFT Analysis in** MATLAB**:** In order to calculate an N-point DFT using MATLAB, we use the `fft` command[12]. The specific calling command is

[11] Because `fourier_analysis` is given only samples of the desired continuous-time signal, it cannot compute the Fourier coefficients exactly. Rather it computes an approximation by using the DFT.

[12] FFT stands for the *Fast Fourier Transform*, which is a fast implementation of the DFT. Calculating the DFT from its definition requires $O(N^2)$ computations, but the FFT only requires $O(N \log N)$. Additionally, the FFT is faster when N is equal to a power of two (i.e., $N = 256, 512, 1024, 2048$, etc.).


```
>>   XX = fft(xx)/length(xx);
```

This computes the N-point DFT of the signal vector `xx` , where N is the length of `xx`, and where the signal is assumed to have support $0, 1, \ldots, N-1$. Since the MATLAB command `fft` does not include the factor $1/N$ in the analysis formula, as in equation (4.29), we must divide by `length(xx)` to obtain the N DFT coefficients `XX`.

- **DFT Synthesis in MATLAB:** The synthesis equation for the DFT is computed with the command `ifft`. If we have computed the DFT using the above command, we must also remember to multiply the result by N:

```
>> xx = ifft(XX)*length(XX);
```

Note that the `ifft` command will generally return complex values even when the synthesis should exactly be real. However, the imaginary part should be negligible (i.e., less than 1×10^{-14}). You can eliminate this imaginary part using the `real` command.

- **Indexing the DFT:** Since MATLAB begins its indexing from 1 rather than 0, remember to use the following rules for indexing the DFT:

$$\begin{aligned} X[0] &\Rightarrow \texttt{X(1)} \\ X[1] &\Rightarrow \texttt{X(2)} \\ X[k] &\Rightarrow \texttt{X(k+1)} \\ X[N-k] &\Rightarrow \texttt{X(N-k+1)} \\ X[N-1] &\Rightarrow \texttt{X(N)} \end{aligned}$$

4.4 Demonstrations in the Lab Section

- Approximating signals as sums of sinusoids, as in Problem 1.
- "Mapping out" this week's background section
- Relating the Fourier Series to the DFT
- T-second Fourier Series and the N-point DFT
- The DFT in MATLAB

4.5 Laboratory Assignment

1. (Building signals from sinusoids) In this problem, you will "hand tune" the amplitudes and phases of three sinusoids so that their sum matches a "target" periodic signal as well as possible. The signals are considered to be continuous-time. One could do this task analytically or numerically using the Fourier series analysis formula, but we want you to gain the insight that results from doing it manually. A graphical MATLAB program has been written to facilitate this procedure.

Download the files **sinsum.m** and **sinsum.fig** and execute **sinsum**[13]. MATLAB will bring up a GUI window with three sinusoids (colored, dotted lines), the sum of these three sinusoids (the black, dashed line), and a target periodic signal (the black, solid line). The frequencies of the sinusoids are ω_0, $2\omega_0$, and $3\omega_0$, where ω_0 is the fundamental frequency of the target signal.

As stated earlier, the goal of this problem is to adjust the amplitudes and phases of the three sinusoids to approximate the target signal as closely as possible. You can enter the amplitude and phase for each sinusoid in the spaces provide in the GUI window, or using the mouse, you can click-and-drag each sinusoid to change its amplitude and phase. In addition to displaying the three sinusoids, their sum, and the target signal, the GUI window also shows the mean-squared error between the sum and the target signals.

Use `sinsum.m` to hand tune the amplitudes and phases of the three sinusoids to make the mean-squared error as small as you can.

(Hint: You should be able achieve an MSE less than 0.24. You will receive +2 bonus points if you can achieve an MSE less than 0.231.)

(Hint: In attempting to minimize the MSE you might try to adjust one sinusoid to minimize the MSE, then another, then another. After doing all three, go back and see if readjusting them in a "second round" has any benefits.)

- [16(+2)] Include the resulting figure window in your report. (On Windows systems, use the "Copy to Clipboard" button to copy the figure, then you can simply paste it into a Word or similar document. There is also a "Print Figure" button for other systems if you can't get access to a PC.)

Food for thought[14]*: Did you try the procedure suggested in the hint above, in which you tune each sinusoid one at a time and then return to each for a "second round" of tuning? If so, can you explain why the second round did or did not lead to any improvements? (Hint: Consider Fourier series property 12.)*

Food for thought: By executing `sinsum(1)`*,* `sinsum(2)`*, and* `sinsum(3)`*, you can match different signals with sinusoids. Find MSE's that are as small as possible for each of these other signals.*

2. (Applying Fourier series synthesis) In this problem you will simply apply `fourier_synthesis` to a given set of Fourier coefficients and find the resulting continuous-time signal. Download the file `fourier_synthesis.m`. Use it to generate an approximation to the signal with the following Fourier coefficients:

$$\alpha_k = \begin{cases} -\left(\frac{2}{\pi k}\right)^2 & k = \pm1, \pm3, \pm5, \ldots \\ 0 & k = 0, \pm2, \pm4, \ldots \end{cases} \tag{4.50}$$

Let $T = 0.1$ seconds, and generate 5 periods of the signal. Use $N = 20$, giving you 41 Fourier series coefficients. (Hint: First, define a frequency support vector, `kk=-20:20`. Then, generate `CC` from `kk` and set all even harmonics to zero.)

[13]Note that this function will *only* work under MATLAB 6 and higher. It is highly recommend that you use a Windows-based PC for this problem, since you need to copy the figure window into your report. Using the Windows clipboard simplifies this task significantly.

[14]"Food for thought" items are not required to be read or acted upon. There is no extra credit for involved. However, if you include something in your report, your GSI will read and comment on it. Alternatively, you can discuss "food for thought topics" in office hours.

- Use `stem` to plot the magnitude of the Fourier coefficients. Use your `kk` vector as the x-axis.
- Use `plot` to plot samples of the continuous-time signal that `fourier_synthesis` returns versus time in seconds.
- What kind of signal is this?

3. (Applying Fourier series analysis) In this problem you will use the Fourier series analysis and synthesis formula to see how the accuracy of the approximate synthesis formula (4.6) depends on N.

   Download the files `lab4_data.mat` and `fourier_analysis.m`. `lab4_data.mat` contains the variables `step_signal` and `step_time`, which are the signal and support vectors for the samples of a continuous-time periodic signal with fundamental period $T_0 = 1$ second. Note that there are $N_s = 16384$ samples in one fundamental period. (`step_signal` and `step_time` include several fundamental periods, but you'll be dealing with only one period in several parts of this problem. As such, you might find it useful to create a one-period version of `step_signal`.)

   (a) (Look at the signal to be analyzed) First, let us examine `step_signal`.
   - Use `plot` to plot `step_signal` versus its support vector.
   - Compute the mean-squared value of `step_signal`.

   (b) (Perform FS analysis) Use `fourier_analysis` to perform a T_0 second Fourier series analysis over *a single period* of `step_signal` with with $N = 50$.
   - Use `subplot` and `stem` to plot the magnitude and phase of the resulting Fourier series coefficients. Make sure that your x-axis is given in frequency.

   (c) (Resynthesize FS approximations) Use `fourier_analysis` and `fourier_synthesis` to generate an approximations of `step_signal` with $N = 25$, 50, 100, and 200. (Perform T_0-second Fourier analysis and synthesis over a single period of the signal for each N. Be sure to resynthesize a single period with `Ns` = 16384 samples.)
   - Use `plot` and `subplot` to plot your resynthesized signals for each N in separate panels of a subplot array.
   - Calculate the mean-squared error of the resynthesis for each value of N.
   - Compute the sum of the squared magnitudes of `CC` for each value of N.
   - Find and document a relationship between the mean-squared errors and the sum of squared magnitudes of `CC` you have computed. (Hint: Consider the mean-squared value that you computed for `step_signal`. You might also want to look in the Properties of Fourier Coefficients subsection.)

   (d) (Meet an MSE target) Find the smallest value of N for which the mean-squared error of the resynthesis is less than 0.5% of the mean-squared value of `step_signal`.
   - Include this value in your report.

   Food for thought: Try repeating Part (b) with the Fourier analysis performed over two fundamental periods of the signal, and compare to the previous answer to Part (b). Do the new Fourier coefficients turn out as expected?

4. (Using the DFT to describe a signal as a sum of discrete-time sinusoids) In this problem, you will simply apply the DFT to a particular discrete-time signal, which is also contained in `lab4_data.mat`, namely, `signal_id`. `signal_id` is considered to be a periodic discrete-time signal with fundamental period $N_0 = 128 =$ `length(signal_id)`. Take the N_0-point DFT of `signal_id`.

   - Use `stem` to plot the magnitude of the DFT versus the DFT coefficient index, k.
   - Use the DFT to describe `signal_id` as a sum of discrete-time sinusoids. That is, for each sinusoid, give the amplitude, frequency (in radians per sample), and phase.

5. (Use the DFT to remove undesired components from a signal) In this problem you will use the technique described in Section 4.2.4 to eliminate a noise signal from a desired signal. This signal, `melody`, is also contained in `lab4_data.mat`. This variable contains samples of a continuous-time signal sampled at rate $f_s = 8192$ samples/second. It contains a simple melody with one note every 1/2 second. Unfortunately, this melody is corrupted by another "instrument" playing a constant note throughout. We would like to remove this second instrument from the signal, and we will use the DFT to do so.

   It is a good idea to begin by listening to `melody` using the `soundsc` command.

   (a) (Examine DFT of first note) In order to remove the corrupting instrument, we need to determine where it lies in the frequency domain. Let's begin by looking at just the first note (i.e the first 0.5 seconds or 4096 samples). This "note" consists of the sum of two notes — one is the first note of the melody, the other is the constant note from the corrupting instrument. Each of these notes has components forming a harmonic series. The fundamental frequencies of these harmonic series are different, which is the key to our being able to remove the corrupting note. Take the DFT of the first 0.5 seconds (4096 samples) of the signal.

      - Use `stem` to plot the magnitude of the DFT for the first note.
      - Identify the frequencies contained in each of the two harmonic series present in signal. What are the fundamental frequencies?

   (b) (Examine DFT of second note) By comparing the spectra of the first two notes, we can identify the corrupting instrument. Take the DFT of the second 0.5 seconds (samples 4097 through 8192).

      - Use `stem` to plot the magnitude of the DFT for the second 0.5 seconds.
      - What are the fundamental frequencies (in Hz) of the two harmonic series in this note?
      - We know that the melody changes from the first note to the second, but the corrupting instrument does not. Thus, by comparing the harmonic series found in this and the previous part, identify which fundamental frequency belongs to the melody and which to the corrupting instrument.

   (c) (Identify the DFT coefficients of the corrupting signal) In order to remove the "corrupting" instrument, we simply need to zero-out the coefficients corresponding to the harmonics of the note from the corrupting instrument. This is done directly on the DFT coefficients of each 0.5 seconds of the signal. Then, we resynthesize the signal from the modified DFT coefficients.

- Based on this, and your results from the previous parts of this problem, which DFT coefficients need to be set to zero in order to remove the corrupting instrument from this signal? (Hint: Remember the conjugate pairs.)

(d) (Complete the function that removes the corrupting instrument) Finally, we'd like to remove the corrupting instrument from our melody. Download the file `fix_melody.m`. This function contains the code that you'll use to remove the corrupting instrument from the melody signal. For each note of the melody, the function takes the DFT, zeros out the appropriate coefficients (which you must provide), and resynthesizes the signal.

- Complete the function by setting the variable `zc` equal to a vector containing the DFT coefficients that must be zeroed-out.
- Execute the function using the command

  ```
  >> result = fix_melody(melody);
  ```

  Listen to the resulting signal. Have you successfully removed the corrupting instrument?

(e) (Check your result with the spectrogram) Finally, we'd like to be able to visually check our result. Download the function `melody_check.m`. `melody_check` produces an image called a *spectrogram* that you can use to check your work. Basically, the spectrogram works by taking the DFT of many short segments of a signal and arranging them as the columns of an image. Note that the x-axis is time and the y-axis is frequency. The color of each point on the image represents the strength of the spectral component (in decibels) at that time and frequency. The dark horizontal bands show the presence of sinusoidal components in the signal at the associated times.

- Execute `melody_check` by passing it `melody`. Include the resulting figure in your report.
- Can you identify the components of the corrupting instrument on this spectrogram?
- Now, execute `melody_check` by passing it `result`. Include the resulting figure in your report.
- Compare the spectrogram of `melody` to the spectrogram of `result`. What differences do you see? Is this what you expect to see?

6. On the front page of your report, please provide an estimate of the average amount of time spent outside of lab by each member of the group.

