Spectrum Representation

Chapter 3

- Extending the investigation of Chapter 2, we now consider signals/waveforms that are composed of multiple sinusoids having different amplitudes, frequencies, and phases

$$\begin{aligned} x(t) &= A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k) \\ &= X_0 + \text{Re}\left\{ \sum_{k=1}^{N} X_k e^{j2\pi f_k t} \right\} \end{aligned} \tag{3.1}$$

where here $X_0 = A_0$ is real, $X_k = A_k e^{j\phi_k}$ is complex, and f_k is the frequency in Hz

- We desire a graphical representation of the parameters in (3.1) versus frequency

The Spectrum of a Sum of Sinusoids

- An alternative form of (3.1), which involves the use of the inverse Euler formula's, is to expand each real cosine into two complex exponentials

$$x(t) = X_0 + \sum_{k=1}^{N} \left\{ \frac{X_k}{2} e^{j2\pi f_k t} + \frac{X_k^*}{2} e^{-j2\pi f_k t} \right\} \tag{3.2}$$

- Note that we now have each real sinusoid expressed as a sum of positive and negative frequency complex sinusoids

Two-Sided Sinusoidal Signal Spectrum: Express $x(t)$ as in (3.2) and then the spectrum is the set of frequency/amplitude pairs

$$\begin{aligned}&\{(0, X_0), (f_1, X_1/2), (-f_1, X_1^*/2), \ldots \\ &\ldots(f_k, X_k/2), (-f_k, X_k^*/2), \ldots \\ &(f_N, X_N/2), (-f_N, X_N^*/2)\}\end{aligned} \tag{3.3}$$

- The spectrum can be plotted as vertical lines along a frequency axis, with height being the magnitude of each X_k or the angle (phase), thus creating either a two-sided magnitude or phase spectral plot, respectively
  - The text first introduces this plot as a combination of magnitude and phase, but later uses distinct plots

---

Example: Constant + Two Real Sinusoids

$$\begin{aligned}x(t) &= 5 + 3\cos(2\pi \cdot 50 \cdot t + \pi/8) \\ &\quad + 6\cos(2\pi \cdot 300 \cdot t + \pi/2)\end{aligned} \tag{3.4}$$

- We expand $x(t)$ into complex sinusoid pairs

$$\begin{aligned}x(t) &= 5 + \frac{3}{2}e^{j\left(2\pi 50 t + \frac{\pi}{8}\right)} + \frac{3}{2}e^{-j\left(2\pi 50 t + \frac{\pi}{8}\right)} \\ &\quad + \frac{6}{2}e^{j\left(2\pi 300 t + \frac{\pi}{2}\right)} + \frac{6}{2}e^{-j\left(2\pi 300 t + \frac{\pi}{2}\right)}\end{aligned} \tag{3.5}$$

- The frequency pairs that define the two-sided *line spectrum* are

$$\begin{aligned}\{(0, 5), (50, 1.5e^{j\pi/8}), (-50, 1.5e^{-j\pi/8}),\\ (300, 3e^{j\pi/2}), -(300, 3e^{-j\pi/2})\}\end{aligned} \tag{3.6}$$

- We can now plot the magnitude phase spectra, in this case with the help of a MATLAB custom function

```
function Line_Spectra(fk,Xk,mode,linetype)
% Line_Spectra(fk,Xk,range,linetype)
%
% Plot Two-sided Line Spectra for Real Signals
%----------------------------------------------------
%       fk = vector of real sinusoid frequencies
%       Xk = magnitude and phase at each positive frequency in fk
%     mode = 'mag' => a magnitude plot, 'phase' => a phase
%              plot in radians
% linetype = line type per MATLAB definitions
%
% Mark Wickert, September 2006; modified February 2009

if nargin < 4
    linetype = 'b';
end

my_linewidth = 2.0;

switch lower(mode) % not case sensitive
    case {'mag','magnitude'} % two choices work
        k = 1;
        if fk(k) == 0
            plot([fk(k) fk(k)],[0 abs(Xk(k))],linetype,...
                'LineWidth',my_linewidth);
            hold on
        else
            Xk(k) = Xk(k)/2;
            plot([fk(k) fk(k)],[0 abs(Xk(k))],linetype,...
```

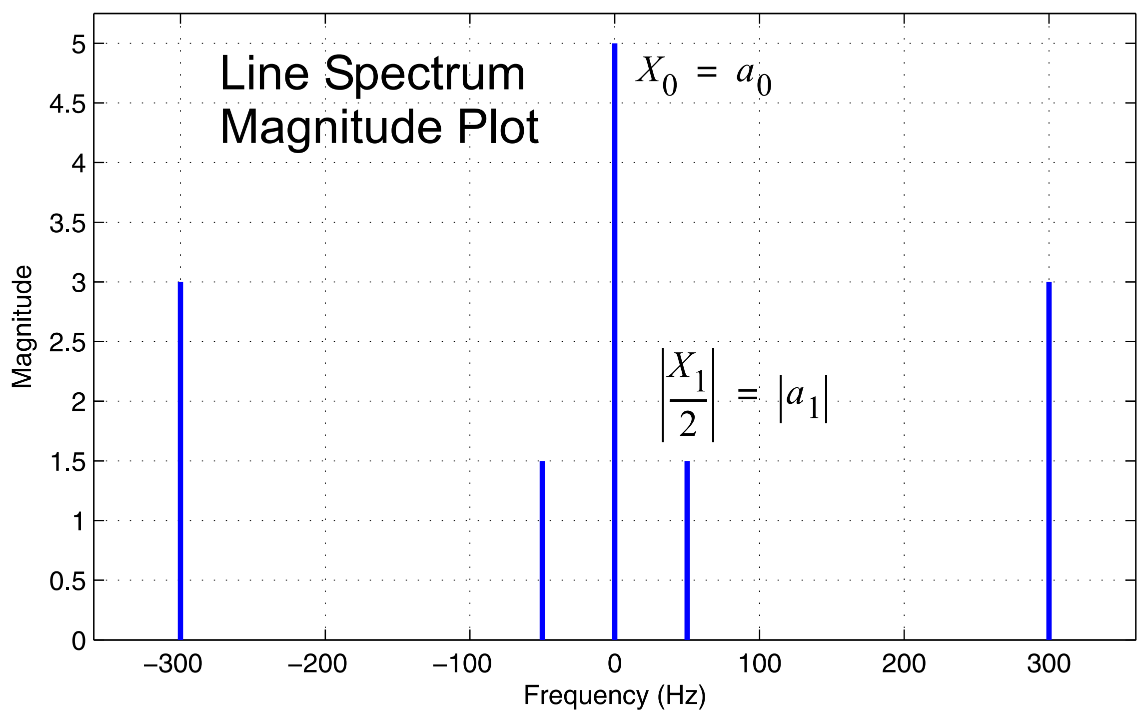
```
                'LineWidth',my_linewidth);
            hold on
            plot([-fk(k) -fk(k)],[0 abs(Xk(k))],linetype,...
                'LineWidth',my_linewidth);
        end
        for k=2:length(fk)
            if fk(k) == 0
                plot([fk(k) fk(k)],[0 abs(Xk(k))],linetype,...
                    'LineWidth',my_linewidth);
            else
                Xk(k) = Xk(k)/2;
                plot([fk(k) fk(k)],[0 abs(Xk(k))],linetype,...
                    'LineWidth',my_linewidth);
                plot([-fk(k) -fk(k)],[0 abs(Xk(k))],linetype,...
                    'LineWidth',my_linewidth);
            end
        end
        grid
       axis([-1.2*max(fk) 1.2*max(fk) 0 1.05*max(abs(Xk))])
        ylabel('Magnitude')
        xlabel('Frequency (Hz)')
        hold off
    case 'phase'
        k = 1;
        if fk(k) == 0
            plot([fk(k) fk(k)],[0 angle(Xk(k))],linetype,...
                 'LineWidth',my_linewidth);
            hold on
        else
            plot([fk(k) fk(k)],[0 angle(Xk(k))],linetype,...
                 'LineWidth',my_linewidth);
            plot([-fk(k) -fk(k)],[0 -angle(Xk(k))],linetype,...
                 'LineWidth',my_linewidth);
            hold on
        end
        for k=2:length(fk)
            if fk(k) == 0
                plot([fk(k) fk(k)],[0 angle(Xk(k))],linetype,...
                    'LineWidth',my_linewidth);
            else
```

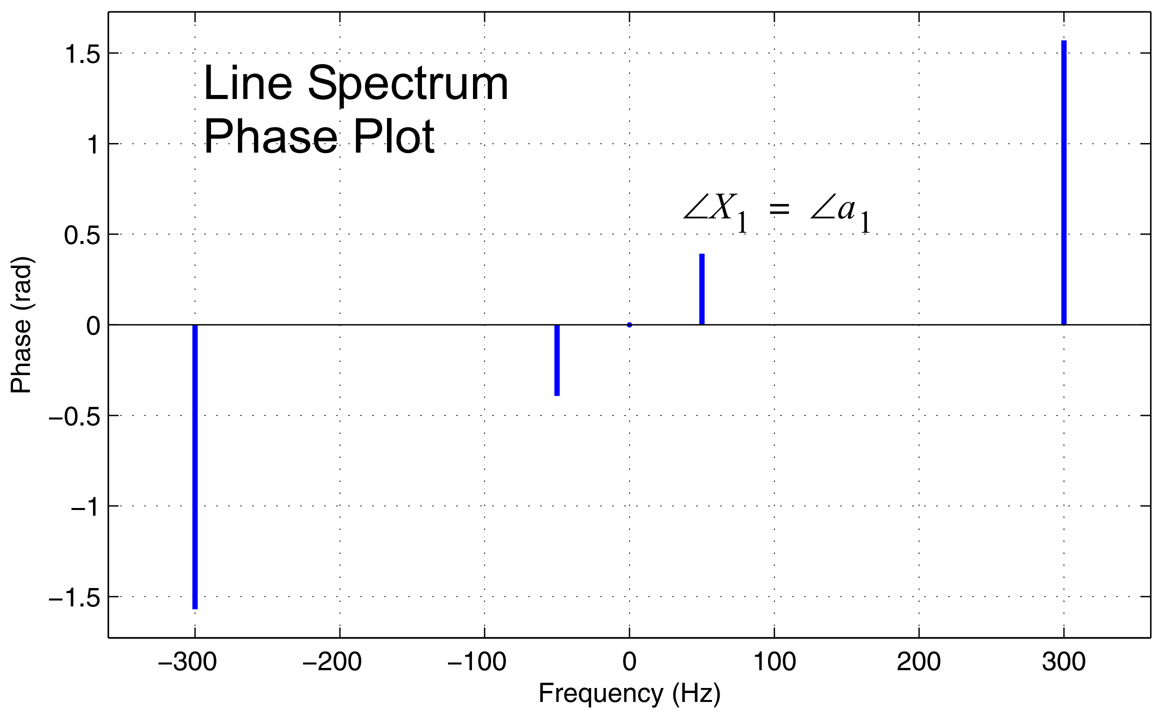
```
            plot([fk(k) fk(k)],[0 angle(Xk(k))],linetype,...
                'LineWidth',my_linewidth);
            plot([-fk(k) -fk(k)],[0 -angle(Xk(k))],...
               linetype,'LineWidth',my_linewidth);
        end
    end
    grid
    plot(1.2*[-max(fk) max(fk)], [0 0],'k');
    axis([-1.2*max(fk) 1.2*max(fk)
        -1.1*max(abs(angle(Xk))) 1.1*max(abs(angle(Xk)))])
    ylabel('Phase (rad)')
    xlabel('Frequency (Hz)')

    hold off
otherwise
    error('mode must be mag or phase')
end
```

- We use the above function to plot magnitude and phase spectra for $x(t)$; **Note** for the `Xk`'s we actually enter $A_k e^{j\theta_k}$

```
>> Line_Spectra([0 50 300],[5 3*exp(j*pi/8) 6*exp(j*pi/2)],'mag')
>> Line_Spectra([0 50 300],[5 3*exp(j*pi/8) 6*exp(j*pi/
2)],'phase')
```





A Notation Change

- The conversion to frequency/amplitude pairs is a bit cumbersome since the factor of $X_k/2$ must be carried for all terms except X_0, therefore the text advocates a more compact spectral form where a_k replaces X_k according to the rule

$$a_k = \begin{cases} X_0, & k = 0 \\ \frac{1}{2}X_k, & k \neq 0 \end{cases} \tag{3.7}$$

- We can then write more compactly the general expression for $x(t)$ as

$$x(t) = \sum_{k=-N}^{N} a_k e^{j2\pi f_k t} \tag{3.8}$$

- The new notations are overlaid in the previous example
- In some cases all of the frequencies in the above sum are related to a common or *fundamental frequency*, via integer multiplication

Beat Notes

- A special case that occurs when we have at least two sinusoids present, is an audio/musical effect known as a *beat note*
- A beat note occurs when we hear the sum of two sinusoids that are very close in frequency, e.g.,

$$x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) \tag{3.9}$$

where $f_1 = f_c - f_\Delta$ and $f_2 = f_c + f_\Delta$

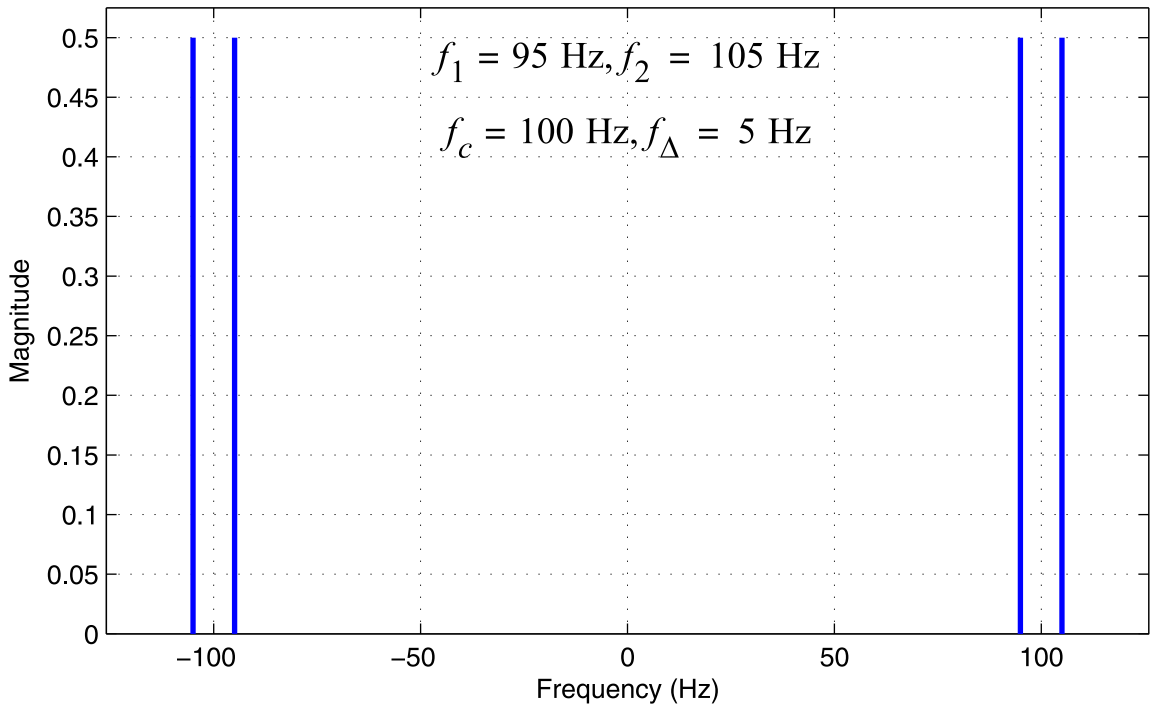
- In this definition

$$\begin{aligned} f_c &= \frac{1}{2}(f_1 + f_2) = \text{center frequency} \\ f_\Delta &= \frac{1}{2}(f_2 - f_1) = \text{deviation frequency} \end{aligned} \tag{3.10}$$

we further assume that $f_\Delta \ll f_c$

Beat Note Spectrum

- Consider `Line_Spectra([95 105],[1 1],'mag')`



- Through the trig double angle formula, or by direct complex sinusoid expansion, we can write that

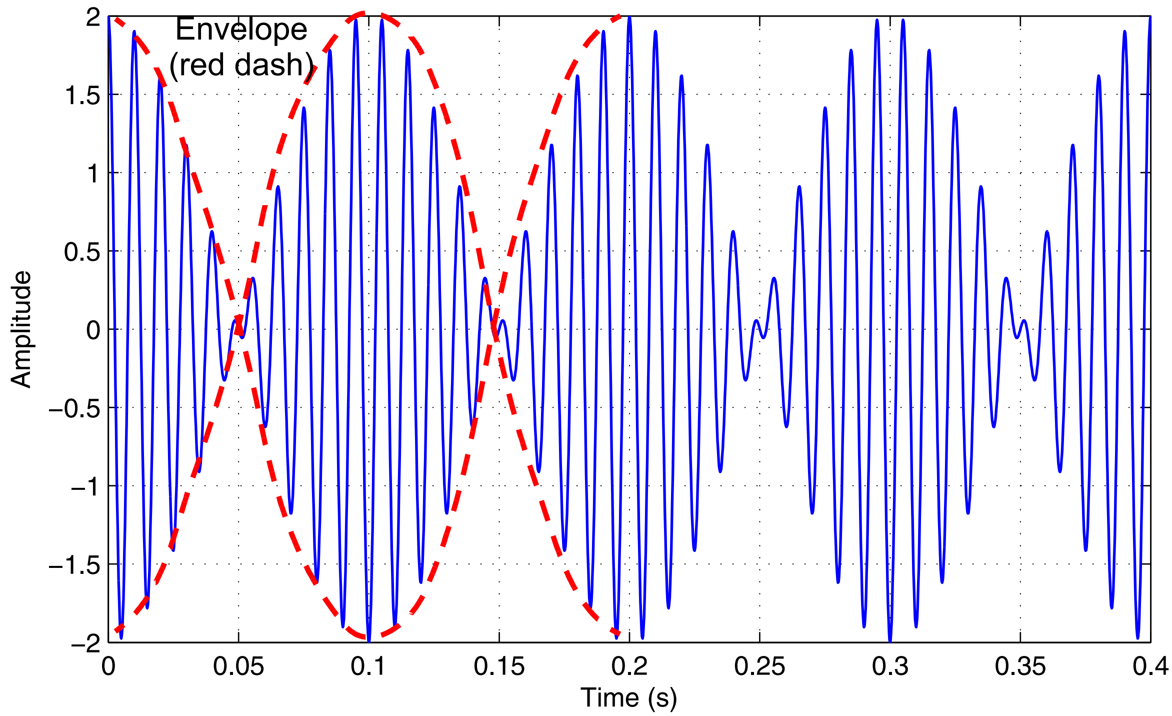
$$\begin{aligned} x(t) &= \cos(2\pi f_1 t) + \cos(2\pi f_2 t) \\ &= \text{Re}\{e^{j2\pi(f_c - f_\Delta)t} + e^{j2\pi(f_c + f_\Delta)t}\} \\ &= \text{Re}\{e^{j2\pi f_c t}[e^{-j2\pi f_\Delta t} + e^{j2\pi f_\Delta t}]\} \\ &= \text{Re}\left\{e^{j2\pi f_c t}[2\cos(2\pi f_{\Delta t})]\right\} \\ &= 2\cos(2\pi f_\Delta t)\cos(2\pi f_c t) \end{aligned} \tag{3.11}$$

- If f_Δ is small compared to f_c, then $x(t)$ appears to have a slowly varying *envelope* controlled by $\cos(2\pi f_\Delta t)$ filled by the rapidly varying sinusoid $\cos(2\pi f_c t)$

Beat Note Waveform

- Consider $f_c = 100$ Hz and $f_\Delta = 5$ Hz

```
>> t = 0:1/(50*100):2/5;
>> x = 2*cos(2*pi*5*t).*cos(2*pi*100*t);
>> plot(t,x)
>> grid
>> xlabel('Time (s)')
>> ylabel('Amplitude')
```



- As f_Δ approaches zero, the envelope fluctuations become slower and slower, and the beat note becomes a steady tone/note; only a single frequency is heard and the line spectrum becomes a single pair of lines at just $\pm f_c$
- With two musicians tuning their instruments, the process of getting $f_\Delta \Rightarrow 0$ is called *in-tune*

Multiplication of Sinusoids

- In the study of beat notes we indirectly encountered sinusoidal multiplication
- Formally we may be interested in

$$x(t) = \cos(2\pi f_1 t) \cdot \cos(2\pi f_2 t) \tag{3.12}$$

- Using trig identity 5 from the notes Chapter 2, we know that

$$\cos\alpha\cos\beta = \frac{1}{2}[\cos(\alpha+\beta) + \cos(\alpha-\beta)] \tag{3.13}$$

- Using this result to expand (3.12) we have that

$$\begin{aligned} x(t) &= \cos(2\pi f_1 t) \cdot \cos(2\pi f_2 t) \\ &= \frac{1}{2}\{\cos[2\pi(f_1 - f_2)t] + \cos[2\pi(f_1 + f_2)t]\} \end{aligned} \tag{3.14}$$

- In words, multiplying two sinusoids of different frequency results in two sinusoids, one at the sum frequency and one at the difference frequency
- For the case where the frequencies are the same, we get

$$x(t) = \cos^2(2\pi f_0 t) = \frac{1}{2}\{1 + \cos[2\pi(2f_0)t]\} \tag{3.15}$$

Amplitude Modulation

- Multiplying sinusoids also occurs in a fundamental radio communications modulation scheme known as *amplitude modulation* (AM)
  - Today AM broadcasting is mostly sports and talk radio

- To form an AM signal we let

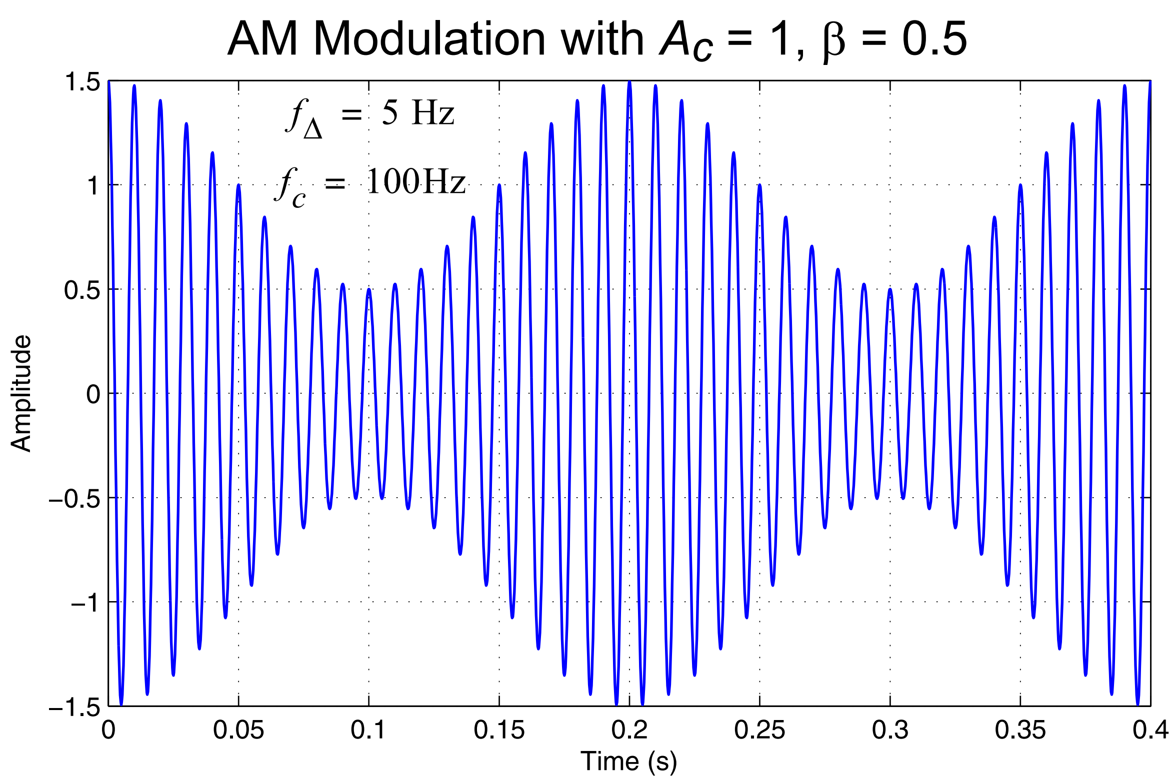
$$x(t) = \underbrace{A_c[1 + \beta m(t)]}_{v(t) \text{ in text}} \cos(2\pi f_c t) \qquad (3.16)$$

  where $m(t)$ is a *message* or information bearing signal, f_c is the *carrier frequency*, and $0 < \beta \le 1$ is the *modulation index*

- The spectral content of $m(t)$ would be say, speech or music (typically low fidelity), such that f_c is much greater that the highest frequencies in $m(t)$

- If $\beta < 1$ the envelope of $x(t)$ never crosses through zero, and the means to recover $m(t)$ from $x(t)$ at a receiver is greatly simplified (so-called envelope detection)

```
>> t = 0:1/(50*100):2/5;
>> x = (1+.5*cos(2*pi*5*t)).*cos(2*pi*100*t);
>> plot(t,x)
```



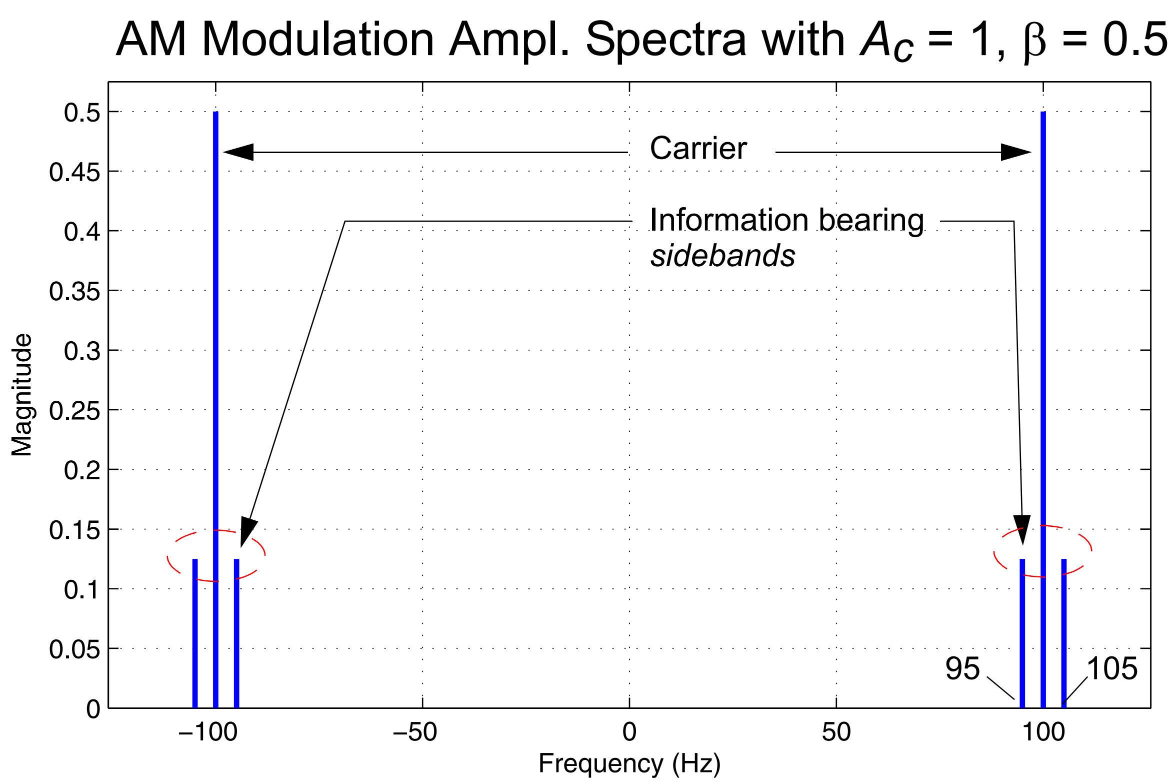
- The spectrum of an AM signal, for $m(t)$ a single sinusoid, can be obtained by expanding $x(t)$ as follows

$$\begin{aligned} x(t) &= A_c[1 + \beta\cos(2\pi f_\Delta t)]\cos(2\pi f_c t) \\ &= A_c\cos(2\pi f_c t) \\ &\quad + \frac{A_c\beta}{2}\{\cos[2\pi(f_c - f_\Delta)t] + \cos[2\pi(f_c + f_\Delta)t]\} \end{aligned} \tag{3.17}$$

- Continuing the AM example with $A_c = 1$ and $\beta = 0.5$, we have

$$\begin{aligned} x(t) &= \cos(2\pi 100 t) \\ &\quad + \frac{1}{4}\{\cos[2\pi(95)t] + \cos[2\pi(105)t]\} \end{aligned} \tag{3.18}$$

```
>> Line_Spectra([95 100 105],[1/4 1 1/4],'mag')
```



Periodic Waveforms

- We have been talking about signals composed of multiple sinusoids, but until now we have not mentioned anything about these signals being periodic
- Recall that a signal is periodic if there exists some T_0 such that $x(t+T_0) = x(t)$
  - The smallest T_0 that satisfies this condition is the *fundamental period* of $x(t)$

Example: $x(t) = 2\cos(2\pi 8t)\cos(2\pi 10t)$

- Expanding we have

$$x(t) = \cos(2\pi 18t) + \cos(2\pi 2t), \tag{3.19}$$

  which has component sinusoids at 2 Hz and 18 Hz
- The fundamental period is $T_0 = 0.5\,\text{s}$, with $f_0 = 1/T_0 = 2\,\text{Hz}$ being the *fundamental frequency*
- Since $18 = 9 \times 2$, we refer to the 18 Hz term as the 9th harmonic

---

- When a signal composed of multiple sinusoids is periodic, the component frequencies are integer multiples of the fundamental frequency, i.e., $f_k = kf_0$, in the expression

$$x(t) = A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k) \tag{3.20}$$

- The fundamental frequency is the largest f_0 such that

$f_k = mf_0$, m an integer, $k = 1, 2, \ldots, N$, or in mathematical terms the *greatest common divisor*

$$f_0 = \gcd\{f_k\}, k = 1, 2, \ldots, N \tag{3.21}$$

- In the example with $f_1 = 2$ and $f_2 = 18$ the largest divisor of {2,18} is 2, since 2/2 and 18/2 both result in integers, but there is no larger value that works

---

Example: Suppose $\{f_k\} = \{3, 7, 9\}$ Hz

- The fundamental is $f_0 = 1$ Hz since 7 is a prime number

Nonperiodic Signals

- In the world of signal modeling both periodic and nonperiodic signals are found
- In music, or least music that is properly tuned, periodic signals are theoretically what we would expect
- It does not take much of a frequency deviation among the various components to make a periodic signal into a nonperiodic signal

---

Example: Three Term Approximation to a *Square Wave*[1]

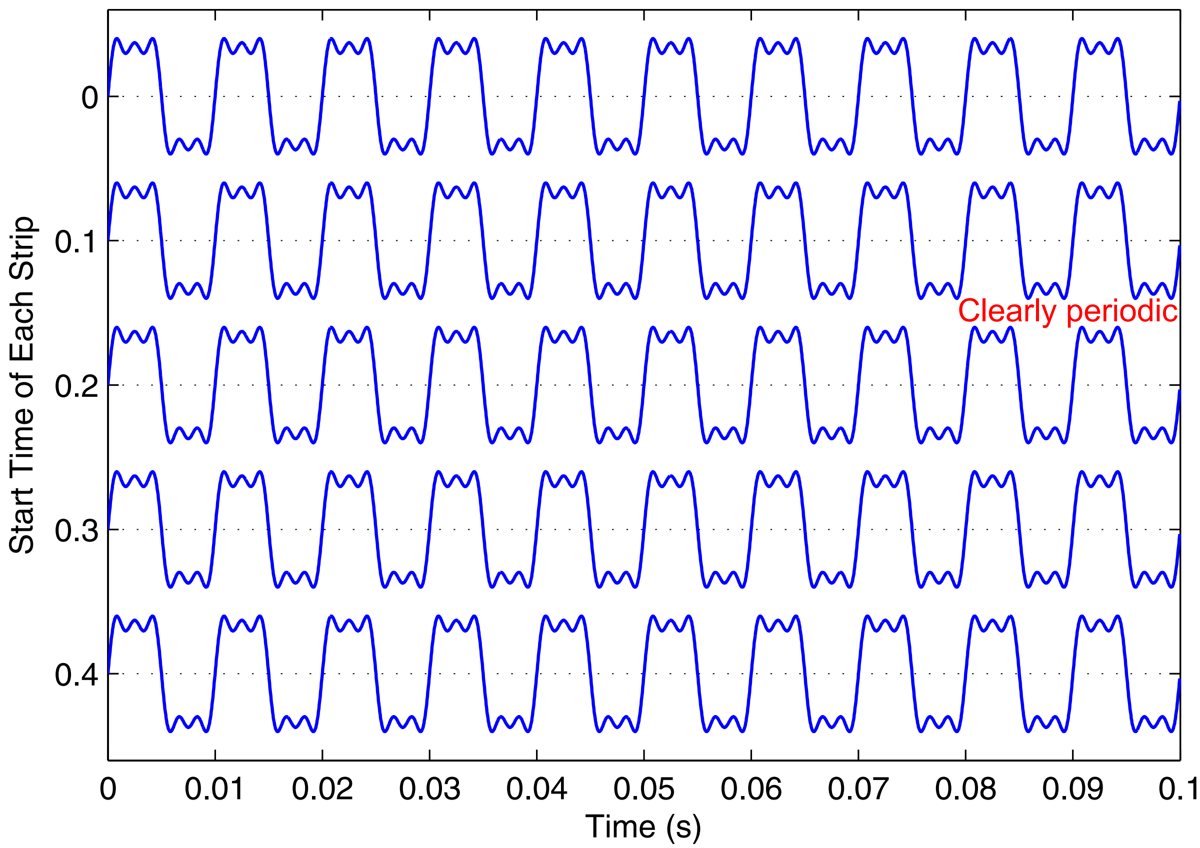
$$x_p(t) = \sin[2\pi(100)t] + \frac{1}{3}\sin[2\pi(300)t] + \frac{1}{5}\sin[2\pi(500)t]$$

- This signal is composed of 1st, 3rd, and 5th harmonic components; fundamental is 100 Hz

1. More on this later in the chapter.

- We plot this waveform using MATLAB

```
>> t = 0:1/(50*500):0.1;
>> x_per = sin(2*pi*100*t)+1/3*sin(2*pi*300*t)+...
        1/5*sin(2*pi*500*t);
>> strips(x_per,.1,50*500)
>> xlabel('Time (s)')
>> ylabel('Start Time of Each Strip')
```

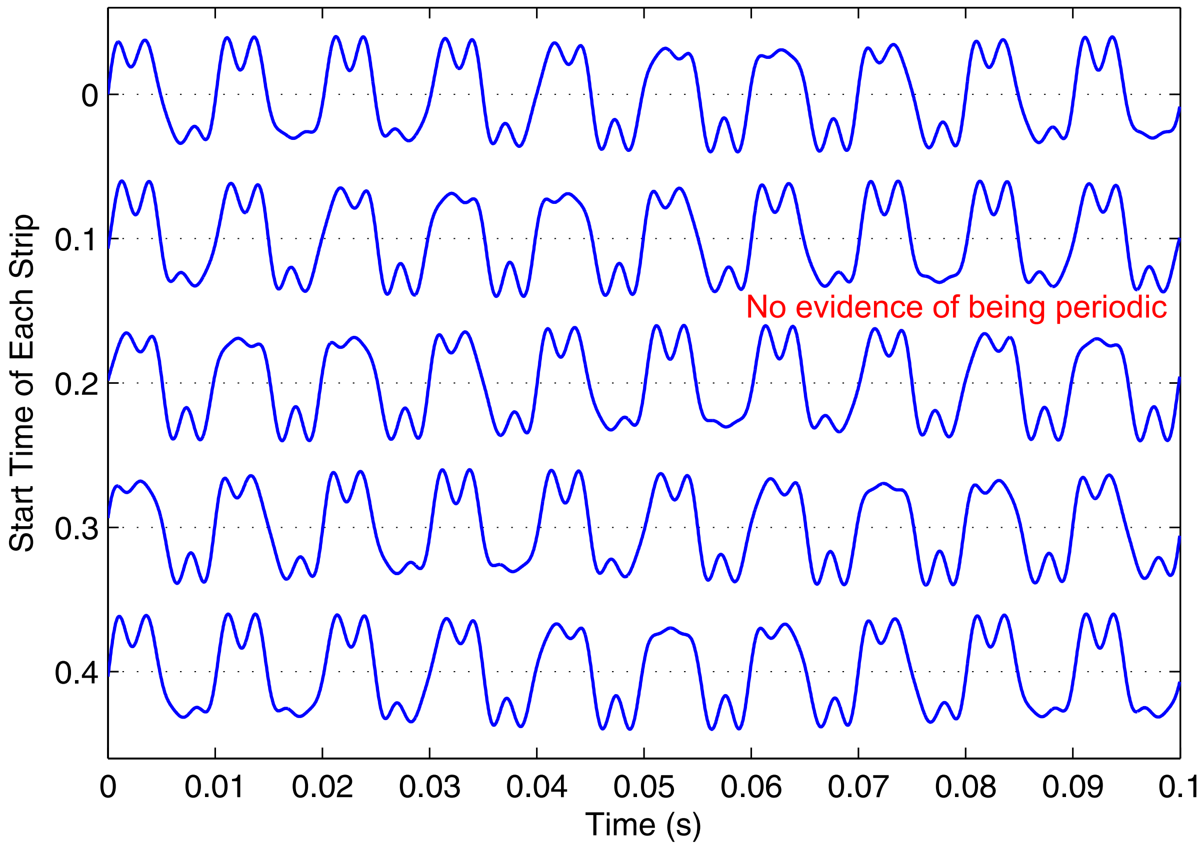


- To make this signal nonperiodic we tweak the frequencies of the 3rd and 5th harmonics

$$x_{np}(t) = \sin[2\pi(100)t] + \frac{1}{3}\sin[2\pi(\sqrt{89999})t] + \frac{1}{5}\sin[2\pi(\sqrt{249999})t]$$

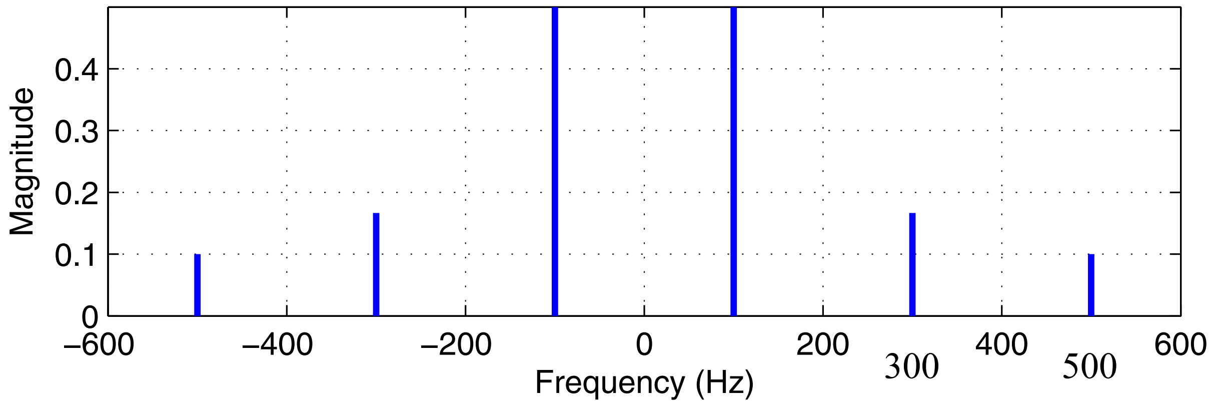
```
>> x_nper = sin(2*pi*100*t)+...
```

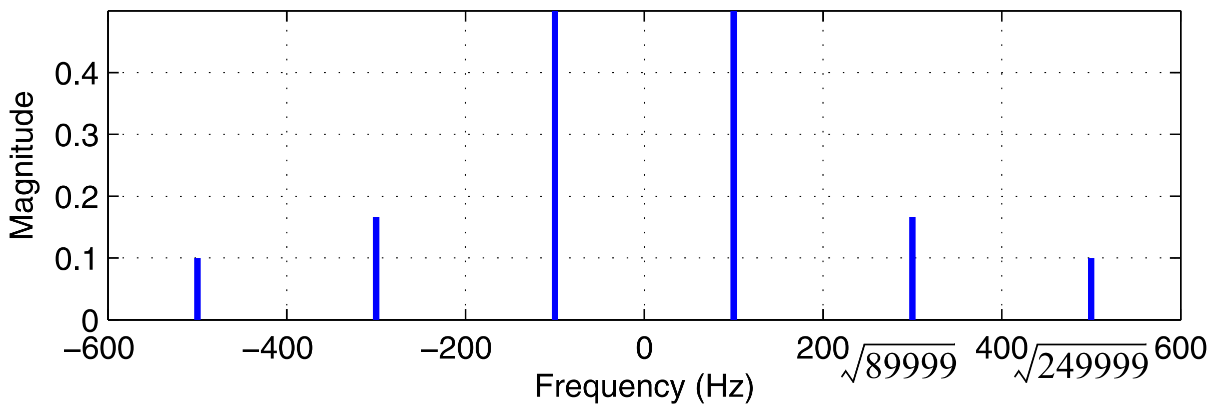
```
                1/3*sin(2*pi*sqrt(89999)*t)+...
                1/5*sin(2*pi*sqrt(149999)*t);
>> strips(x_nper,.1,50*500)
>> xlabel('Time (s)')
>> ylabel('Start Time of Each Strip')
```



- It is interesting to note that the line spectra of both signals is very similar, in particular the magnitude spectra as shown below

```
>> subplot(211)
>> Line_Spectra([100 300 500],[1 1/3 1/5],'mag')
>> subplot(212)
>> Line_Spectra([100 sqrt(89999) sqrt(249999)],...
        [1 1/3 1/5],'mag')
```





Fourier Series

Through the study of Fourier[1] series we will learn how any periodic signal can be represented as a sum of harmonically related sinusoids.

- The *synthesis* formula is

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j(2\pi/T_0)kt} \tag{3.22}$$

where T_0 is the period

- The *analysis* formula will determine the a_k from $x(t)$

1. French mathematician who wrote a thesis on this topic in 1807.

- For $x(t)$ a real signal, we see that $a_{-k} = a_k^* = \text{conj}(a_k)$ and then we can write that

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos[(2\pi/T_0)kt + \phi_k], X_k = A_k e^{j\phi_k} \quad (3.23)$$

Fourier Series: Analysis

- To obtain a Fourier series representation of periodic signal $x(t)$ we need to evaluate the Fourier integral

$$a_k = \frac{1}{T_0}\int_0^{T_0} x(t) e^{-j(2\pi/T_0)kt} dt \quad (3.24)$$

  where T_0 is the fundamental period

- As a special case note that the DC component of $x(t)$, given by a_0, is

$$a_0 = \frac{1}{T_0}\int_0^{T_0} x(t) dt \quad (3.25)$$

  – We call a_0 the average value since it finds the area under $x(t)$ over one period divided (normalized) by T_0

Fourier Series Derivation

- Since working with complex numbers is a relatively new concept, it might seem that proving (3.24) which involves complex exponentials, is out of reach for this course; not so
- The result of (3.24) can be established through a careful step-by-step process

- We begin with the property that integration of a complex exponential over an integer number of periods is identically zero, i.e.,

$$\int_0^{T_0} e^{j(2\pi/T_0)kt} dt = 0 \tag{3.26}$$

  – Verify Version #1:

$$\int_0^{T_0} e^{j(2\pi/T_0)kt} dt = \left.\frac{e^{j(2\pi/T_0)kt}}{j(2\pi k/T_0)}\right|_0^{T_0} = \frac{e^{j(2\pi/T_0)kT_0} - 1}{j(2\pi k/T_0)} = 0$$

  since $e^{j2\pi k} = 1$ for any integer $k = 1, 2, \ldots$

  – Verify Version #2: Expand the integrand using Euler's formula

$$\int_0^{T_0} e^{j(2\pi/T_0)kt} dt = \int_0^{T_0} \left\{ \cos\left(\left[\left(\frac{2\pi}{T_0}\right)kt\right]\right) + j\sin\left(\left[\left(\frac{2\pi}{T_0}\right)kt\right]\right) \right\} dt$$

$$= 0 + j0 = 0$$

  since integrating over one or more complete cycles of sin/cos is always zero

- Regardless of the harmonic number k, all complex exponentials of the form $v_k(t) = \exp[j(2\pi k/T_0)t]$, repeat with period T_0, i.e.,

$$\begin{aligned} v_k(t+T_0) &= e^{j\left(\frac{2\pi k}{T_0}\right)(t+T_0)} \\ &= e^{j\left(\frac{2\pi k}{T_0}\right)t} e^{j\left(\frac{2\pi k}{T_0}\right)T_0} \\ &= e^{j\left(\frac{2\pi k}{T_0}\right)t} \cancelto{1}{e^{j2\pi k}} \\ &= v_k(t) \end{aligned}$$

Orthogonality Property

$$\boxed{\int_0^{T_0} v_k(t) v_l^*(t) dt = \begin{cases} 0, & k \neq l \\ T_0, & k = l \end{cases}} \tag{3.27}$$

– Note:

$$v_l^*(t) = \{\exp[j(2\pi l/T_0)t]\}^* = \exp[-j(2\pi l/T_0)t]$$

– Proof:

$$\begin{aligned} \int_0^{T_0} v_k(t) v_l^*(t) dt &= \int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)kt} e^{-j\left(\frac{2\pi}{T_0}\right)lt} dt \\ &= \int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)(k-l)t} dt \end{aligned}$$

– When $k = l$ the exponent is zero and the integral reduces to

$$\int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)(k-l)t} dt = \int_0^{T_0} e^{j0} dt = \int_0^{T_0} dt = T_0$$

– When $k \neq l$, but rather some integer, say m, we have

$$\int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)(k-l)t} dt = \int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)mt} dt = 0$$

Final Step: We now have enough tools to comfortably prove the Fourier analysis formula.

- We take the Fourier synthesis formula, multiply both sides by $v_l^*(t)$ and integrate over one period T_0

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j(2\pi/T_0)kt}$$

$$x(t)e^{-j\left(\frac{2\pi}{T_0}\right)lt} = \sum_{k=-\infty}^{\infty} a_k e^{j\left(\frac{2\pi}{T_0}\right)kt} e^{-j\left(\frac{2\pi}{T_0}\right)lt}$$

$$\int_0^{T_0} x(t)e^{-j\left(\frac{2\pi}{T_0}\right)lt} dt = \int_0^{T_0} \left\{ \sum_{k=-\infty}^{\infty} a_k e^{j\left(\frac{2\pi}{T_0}\right)kt} e^{-j\left(\frac{2\pi}{T_0}\right)lt} \right\} dt$$

$$= \sum_{k=-\infty}^{\infty} a_k \left\{ \int_0^{T_0} e^{j\left(\frac{2\pi}{T_0}\right)kt} e^{-j\left(\frac{2\pi}{T_0}\right)lt} dt \right\}$$

- Due to the orthogonality condition, the only surviving term is when $k = l$, and here the integral is T_0

- We are left with

$$\int_0^{T_0} x(t) e^{-j\left(\frac{2\pi}{T_0}\right) lt} \, dt = a_l T_0$$

or

$$a_l = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-j\left(\frac{2\pi}{T_0}\right) lt} \, dt$$

and we have completed the proof!

Summary

$$a_k = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-j\left(\frac{2\pi}{T_0}\right) kt} \, dt \quad \textbf{Analysis}$$

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j(2\pi/T_0)kt} \quad \textbf{Synthesis} \tag{3.28}$$

Spectrum of the Fourier Series

- The spectrum associated with a Fourier series representation is consistent with the earlier discussion of two-sided line spectra
- The frequency/amplitude pairs are

$$\{(0, a_0), (\pm f_0, a_{\pm 1}), (\pm 2f_0, a_{\pm 2}), \ldots, (\pm kf_0, a_{\pm k}), \ldots\} \tag{3.29}$$

Example: $x(t) = \cos^2[2\pi(1500)t]$

- This signal has a Fourier series representation that we can obtain directly by expanding $\cos^2$

$$\cos^2[2\pi(1500)t] = \left\{\frac{e^{j2\pi 1500t} + e^{-j2\pi 1500t}}{2}\right\}^2$$

$$= \left(\frac{e^{j2\pi 3000t} + 2 + e^{-j2\pi 3000t}}{4}\right)$$

$$= \underbrace{\frac{1}{2}}_{k=0} + \underbrace{\frac{1}{4}e^{j2\pi 3000t}}_{k=2} + \underbrace{\frac{1}{4}e^{-j2\pi 3000t}}_{k=-2}$$

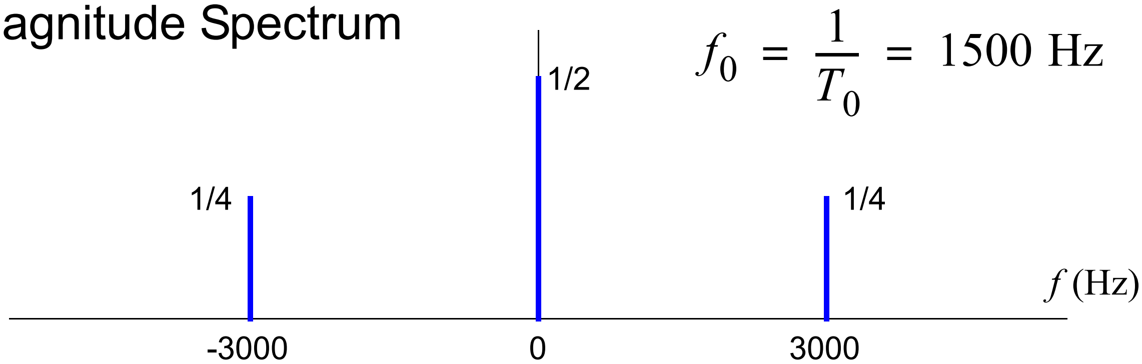
Fourier Series Coeff.

- By comparing the above with the general Fourier series synthesis formula, we see that relative to $f_0 = 1/T_0 = 1500\,\text{Hz}$

$$a_k = \begin{cases} 1/2, & k = 0 \\ 1/4, & k = \pm 2 \\ 0, & \text{otherwise} \end{cases}$$

Magnitude Spectrum

$$f_0 = \frac{1}{T_0} = 1500\ \text{Hz}$$



Fourier Analysis of Periodic Signals

We can synthesize an approximation to some periodic $x(t)$ once we have an expression for the Fourier coefficients $\{a_k\}$ using the first N harmonics

$$x_N(t) = \sum_{k=-N}^{N} a_k e^{j(2\pi/T_0)kt}. \tag{3.30}$$

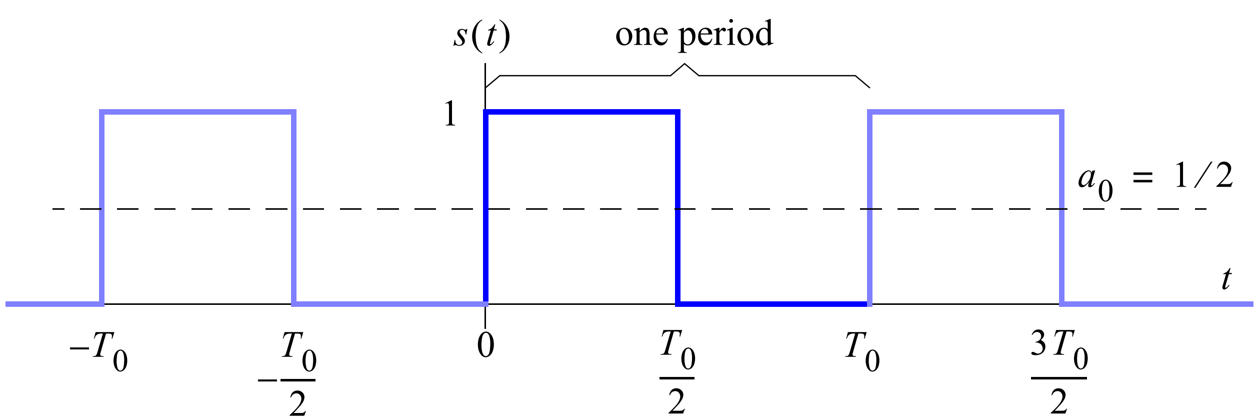
- We can then implement the plotting of this approximation using MATLAB

The Square Wave

- Here we consider a signal which over one period is given by

$$s(t) = \begin{cases} 1, & 0 \le t < T_0/2 \\ 0, & T_0/2 \le t < T_0 \end{cases} \tag{3.31}$$

  – This is actually called a 50% duty cycle square wave, since it is *on* for half of its period



- We solve for the Fourier coefficients via integration (the Fourier integral)

$$
\begin{aligned}
a_k &= \frac{1}{T_0}\int_0^{T_0/2}(1)e^{-j(2\pi/T_0)kt}dt + 0 \\
&= \frac{1}{T_0}\left[\frac{e^{-j(2\pi/T_0)kt}}{-j(2\pi/T_0)k}\right]\Bigg|_0^{T_0/2} = \frac{1-e^{-j\pi k}}{j2\pi k}
\end{aligned}
\tag{3.32}
$$

- Notice that $e^{-j\pi} = -1$, so

$$
a_k = \frac{1-(-1)^k}{j2\pi k} \text{ for } k \neq 0 \tag{3.33}
$$

and for $k = 0$ we have

$$
a_0 = \frac{1}{T_0}\int_0^{T_0/2}(1)e^{-j0}dt = \frac{1}{2} \text{ (DC value)} \tag{3.34}
$$

  – This is the average value of the waveform, which is dependent upon the 50% aspect (i.e., halfway between 0 and 1)

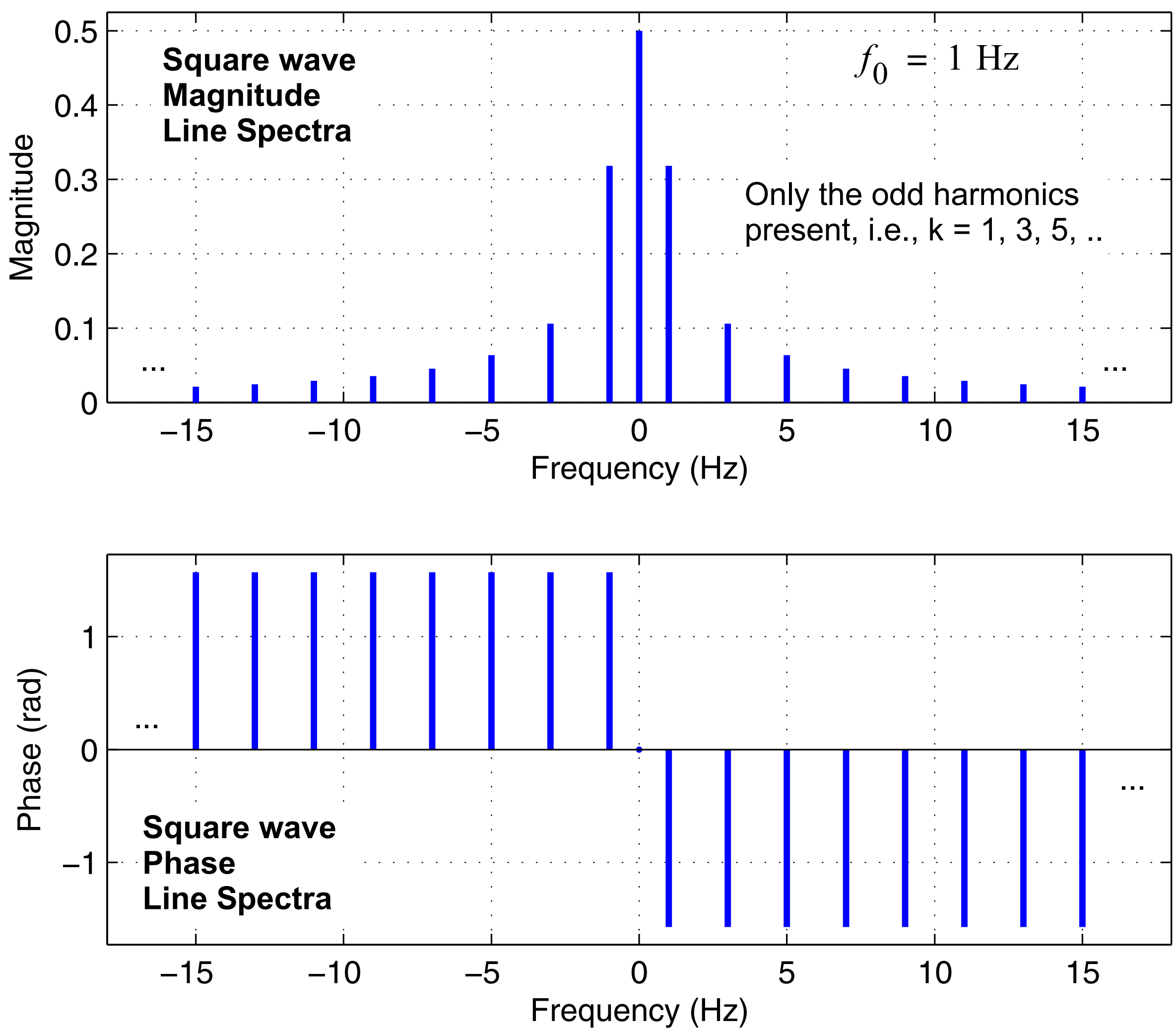
- In summary,

$$
\boxed{a_k = \begin{cases} \frac{1}{2}, & k = 0 \\ \frac{1}{j\pi k}, & k = \pm 1, \pm 3, \pm 5, \ldots \\ 0, & k = \pm 2, \pm 4, \pm 6, \ldots \end{cases}} \tag{3.35}
$$

Spectrum for a Square Wave

- We can plot the square wave amplitude spectrum using the `Line_Spectrum()` function, by converting the coefficients from a_k back to X_k

```
>> N = 15; k = 1:2:N; % odd frequencies
>> Xk = 2./(j*pi*k); % Xk's at odd freqs, Xk = 2*ak
>> k = [0 k]; % augment with DC value
>> Xk = [1/2 Xk]; % X0 = a0
>> subplot(211)
>> Line_Spectra(1*k,Xk,'mag')
>> subplot(212)
>> Line_Spectra(1*k,Xk,'phase')
```



Synthesis of a Square Wave

- We can synthesize a square wave by forming a partial sum, say up to the 15th harmonic; $N = 15$ in (3.30)
- First we modify syn_sin() for Fourier series modeling

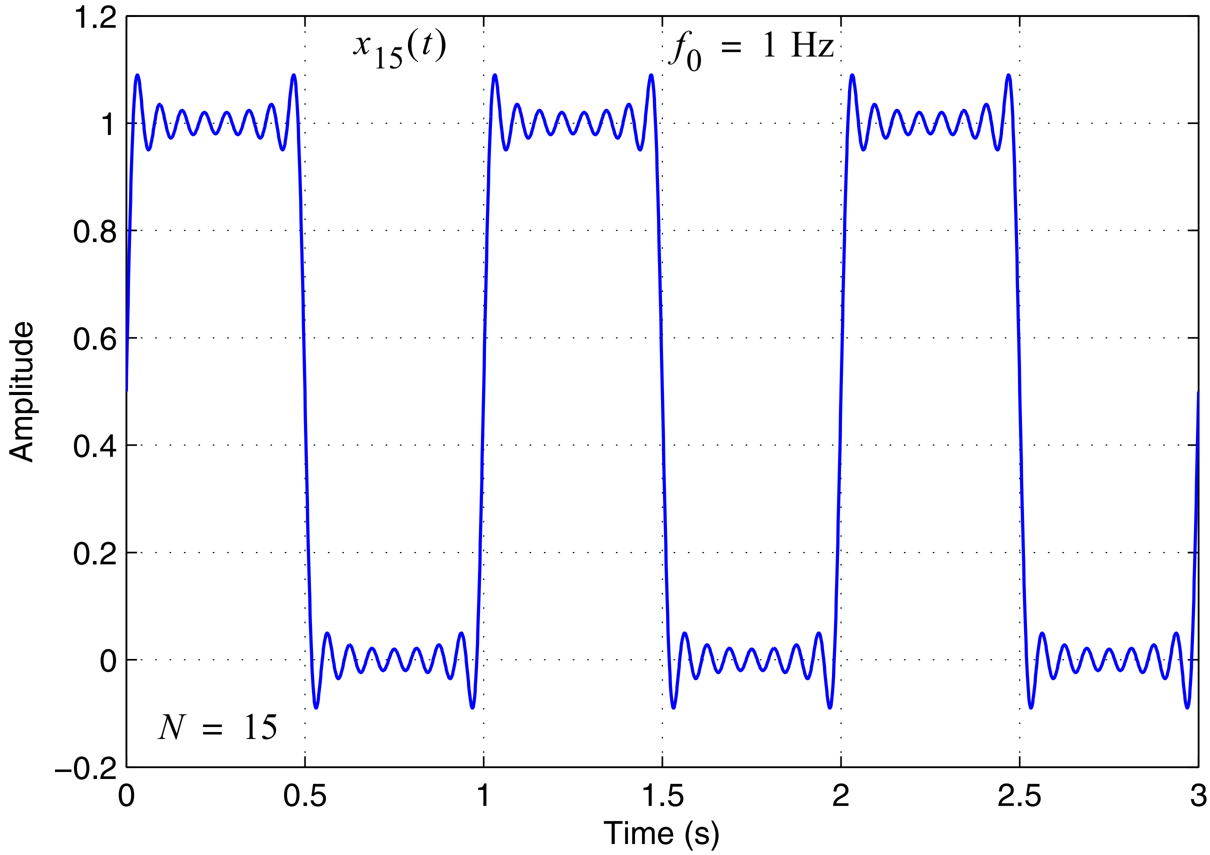
```
function [x,t] = fs_synth(fk, ak, fs, dur, tstart)
% [x,t] = fs_synth(fk, ak, fs, dur, tstart)
%
% Mark Wickert, September 2006

if nargin < 5,
    tstart = 0;
end

t = tstart:1/fs:dur;
x = zeros(size(t));
for k=1:length(fk)
    x = x + ak(k)*exp(j*2*pi*fk(k)*t);
end
```

- The code used to produce simulation results for $x_{15}(t)$:

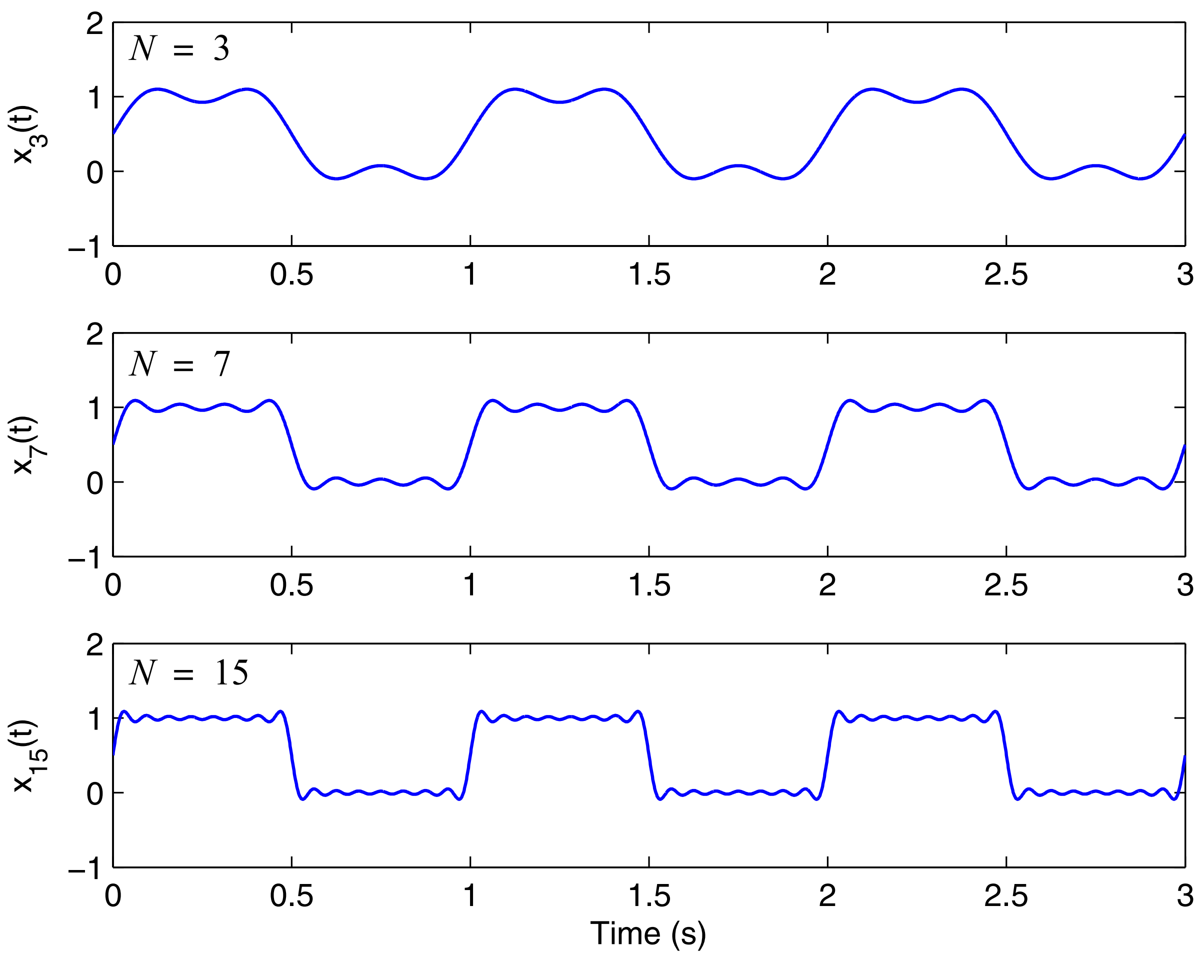
```
>> N = 15; k = -N:2:N;
>> ak = 1./(j*pi*k);
>> fk = 1*[0 k];
>> ak = [1/2 ak];
>> [x,t] = fs_synth(fk, ak, 50*15, 3);
>> plot(t,real(x)) % note x is not purely real
>> grid            % due to numerical imperfections
>> xlabel('Time (s)')
>> ylabel('Amplitude')
```



- With the $N = 15$ approximation, we observe that there is *ringing* or *ears* as the waveform makes discontinuous steps from 0 to 1 and 1 back to 0
- This behavior is known as the *Gibbs phenomenon*, and comes about due to the discontinuity of the ideal square wave
- The next plot shows that regardless of N, the ringing persists with about a 9% overshoot/undershoot at the transition points
- The frequency of the rings increases as N increases

```
>> N = 3; k = -N:2:N;
>> ak = 1./(j*pi*k); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x3,t] = fs_synth(fk, ak, 50*15, 3);
>> N = 7; k = -N:2:N;
```

```
>> ak = 1./(j*pi*k); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x7,t] = fs_synth(fk, ak, 50*15, 3);
>> N = 15; k = -N:2:N;
>> ak = 1./(j*pi*k); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x15,t] = fs_synth(fk, ak, 50*15, 3);
>> subplot(311); plot(t,real(x3))
>> ylabel('x_3(t)')
>> subplot(312); plot(t,real(x7))
>> ylabel('x_7(t)')
>> subplot(313); plot(t,real(x15))
>> ylabel('x_15(t)')
>> xlabel('Time (s)')
```



- A limitation of Fourier series is that it cannot handle discontinuities very well, real physical waveforms do not have discontinuities to the extreme found in mathematical models

---

Example: Frequency Tripler

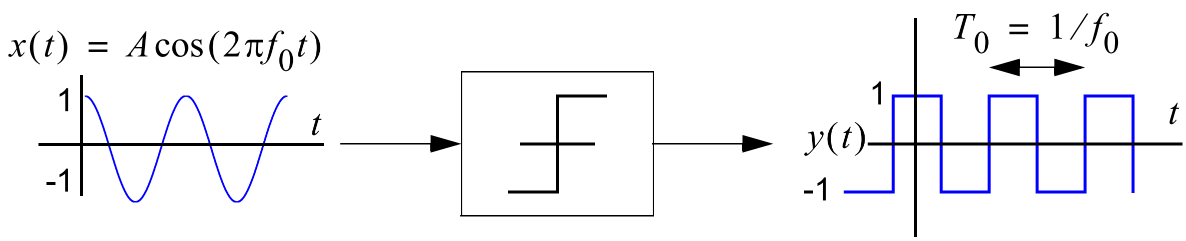
- Suppose we have a sinusoidal signal

$$x(t) = A\cos(2\pi f_0 t)$$

and we would like to obtain a sinusoidal signal of the form

$$y(t) = B\cos(2\pi(3f_0)t)$$

- The systems aspect of this example is that we can convert $x(t)$ into a square wave centered about zero, by passing the signal through a *limiter* (like a comparator)

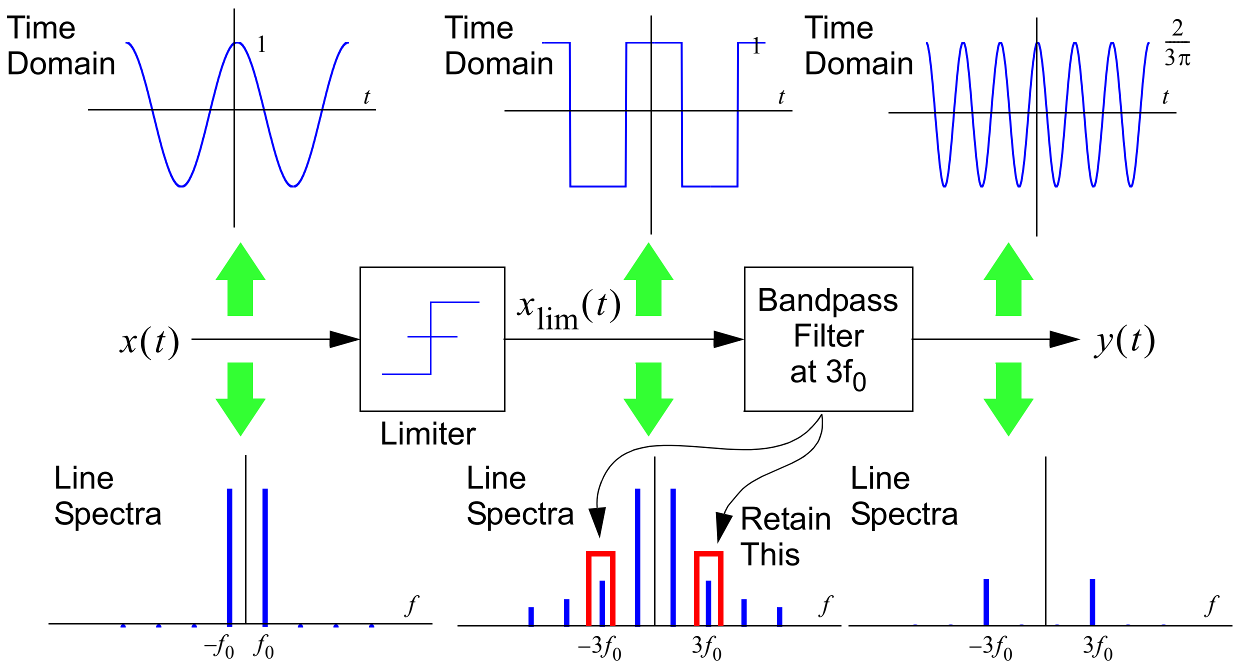


- The output signal $y(t)$ is very similar to $s(t)$, that is

$$y(t) = 2s(t + T_0/4) - 1$$

- The Fourier series coefficients of the $y(t)$ square wave and the $s(t)$ square wave are related via an amplitude shifting and time shifting property
- Without going into the details, it can be said that the $\{a_k\}$ coefficients for $k \neq 0$ still only exist for k odd, and have a scale factor of the form C/k

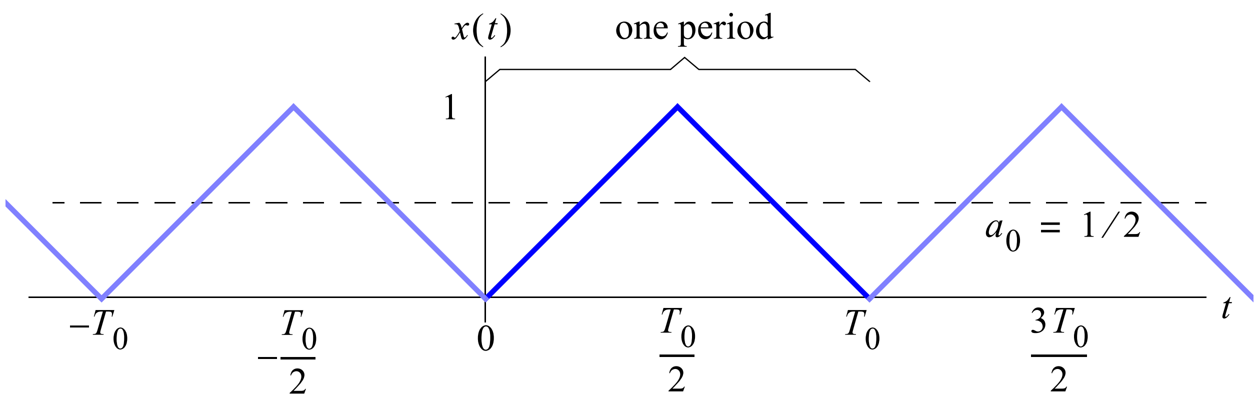
- Note that $a_0 = 0$ why?
- The Fourier coefficients that contribute to $B\cos(2\pi(3f_0)t)$ are at k = -3 and 3
- Knowing that the line spectra consists of all of the odd harmonics, means that in order to obtain just the 3rd harmonic we need to design a filter that will allow just this signal to pass (a *bandpass filter*)
- A system block diagram with waveforms and line spectra is shown below



Triangle Wave

- Another waveform of interest is the *triangle wave*

$$x(t) = \begin{cases} 2t/T_0, & 0 \le t < T_0/2 \\ 2(T_0 - t)/T_0, & T_0/2 \le t < T_0 \end{cases} \tag{3.36}$$



- We use the Fourier analysis formula to obtain the $\{a_k\}$ coefficients, starting with the DC term

$$a_0 = \frac{1}{T_0}\int_0^{T_0} x(t)dt = \frac{1}{T_0} \times \text{area} = \frac{1}{T_0} \cdot \frac{T_0}{2} = \frac{1}{2} \tag{3.37}$$

- The remaining terms are found using integration

$$\begin{aligned} a_k &= \frac{1}{T_0}\int_0^{T_0/2} \left\{\frac{2t}{T_0}\right\} e^{-j(2\pi/T_0)kt} dt \\ &+ \frac{1}{T_0}\int_{T_0/2}^{T_0} \left\{\frac{2(T_0 - t)}{T_0}\right\} e^{-j(2\pi/T_0)kt} dt \end{aligned} \tag{3.38}$$

- To evaluate this integral we must use integration by parts, or from a mathematical handbook[1] lookup the result that

$$\int x e^{ax} dx = \frac{e^{ax}}{a}\left(x - \frac{1}{a}\right)$$

1. Murray R. Spiegel, *Mathematical Handbook of Formulas and Tables*, 2nd ed., Schaum's Outlines, McGraw Hill, New York, 1999.

- The symbolic engine of Mathematica can also solve this

$$\text{I1} = \frac{1}{T}\,\text{Integrate}\left[\frac{2\,t}{T}\,\text{Exp}\left[-j\,\frac{2\,\pi}{T}\,k\,t\right],\ \left\{t,\ 0,\ \frac{T}{2}\right\}\right]$$

$$-\frac{e^{-i\,k\,\pi}\,\left(-1+e^{i\,k\,\pi}-i\,k\,\pi\right)}{2\,k^2\,\pi^2}$$

$$\text{I2} = \frac{1}{T}\,\text{Integrate}\left[\frac{2\,(T-t)}{T}\,\text{Exp}\left[-j\,\frac{2\,\pi}{T}\,k\,t\right],\ \left\{t,\ \frac{T}{2},\ T\right\}\right]$$

$$\frac{e^{-2\,i\,k\,\pi}\,\left(-1+e^{i\,k\,\pi}\,(1-i\,k\,\pi)\right)}{2\,k^2\,\pi^2}$$

ak = FullSimplify[I1 + I2]

$$-\frac{e^{-i\,k\,\pi}\,\left(-1+\text{Cos}[k\,\pi]\right)}{k^2\,\pi^2}$$

- From the above Mathematica result, we note that $e^{-jk\pi} = (-1)^k$ and $\cos(k\pi) = (-1)^k$, so

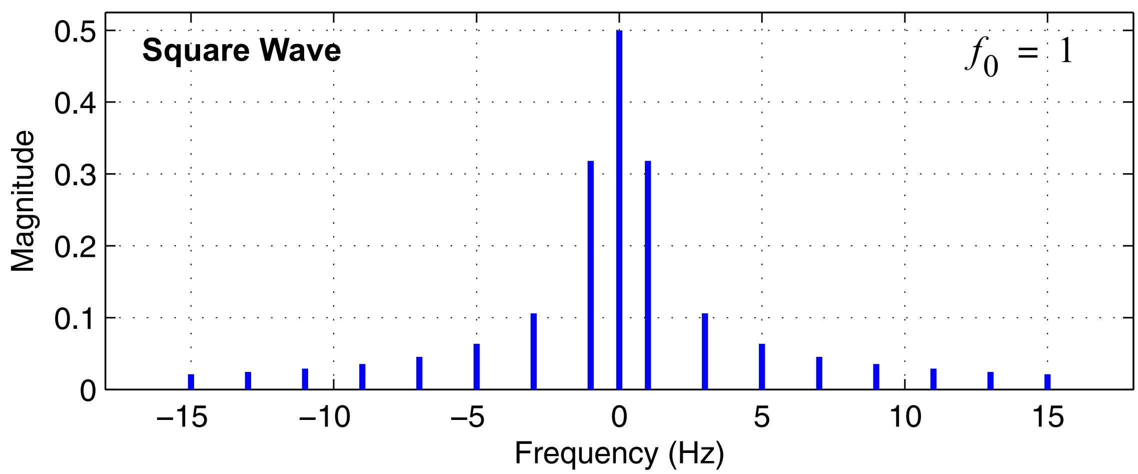
$$a_k = -\frac{(-1)^k[(-1)^k - 1]}{k^2\pi^2} = \begin{cases} \dfrac{2}{k^2\pi^2}, & k = \pm 1, \pm 3, \pm 5, \ldots \\ 1/2, & k = 0 \\ 0, & k = \pm 2, \pm 4, \pm 6, \ldots \end{cases} \tag{3.39}$$

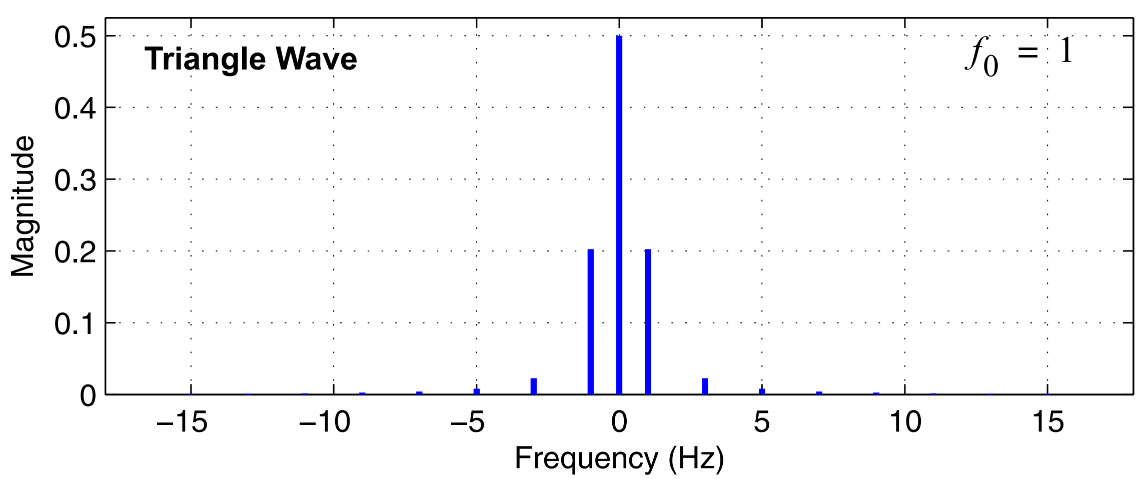
since $(-1)^k[(-1)^k - 1] = 2$ when k is odd and zero otherwise

Triangle Wave Spectrum

- Compare the line spectra for a triangle wave and square wave out to the 15th harmonic

```
>> N = 15; k = 1:2:N;
>> Xk = 2./(j*pi*k); Xk = [1/2 Xk];
>> k = [0 k];
>> subplot(211)
>> Line_Spectra(1*k,Xk,'mag')
>> N = 15; k = 1:2:N;
>> Xk = -4./(pi^2*k.^2); Xk = [1/2 Xk];
>> k = [0 k];
>> subplot(212)
>> Line_Spectra(1*k,Xk,'mag')
```





- Note that the spectral lines drop off with $1/k^2$ for the triangle wave, compared with just $1/k$ for the square wave

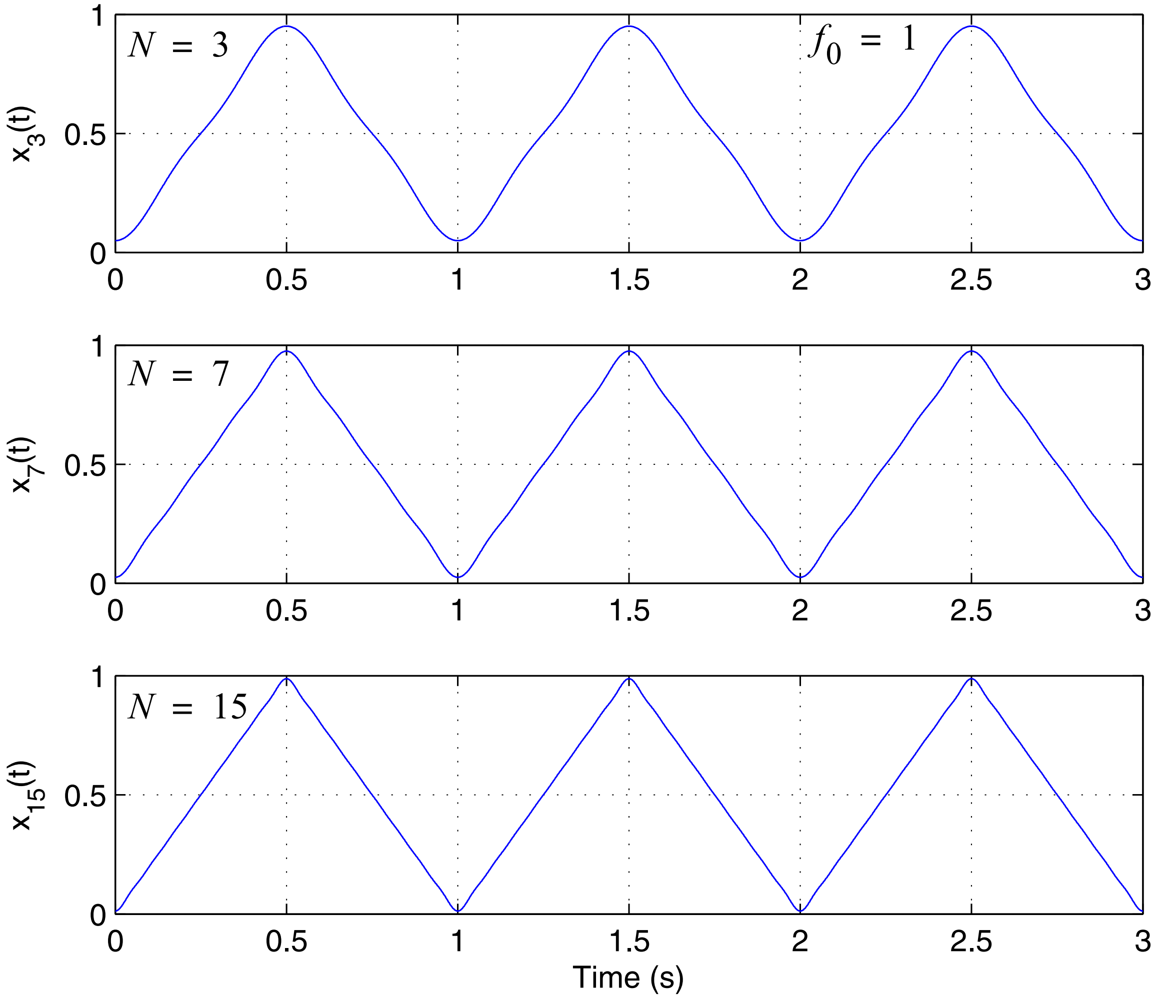
- The relative smoothness of the triangle wave results in the faster spectrum decrease

Synthesis of a Triangle Wave

- As with the square wave, we can synthesize size a triangle wave by forming a partial sum, say for $N = 3, 7, 15$

```
>> N = 3; k = -N:2:N;
>> ak = -2./(pi^2*k.^2); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x3,t] = fs_synth(fk, ak, 50*15, 3);
>> subplot(311); plot(t,real(x3)); grid
>> ylabel('x_3(t)')
>> N = 7; k = -N:2:N;
>> ak = -2./(pi^2*k.^2); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x7,t] = fs_synth(fk, ak, 50*15, 3);
>> subplot(312); plot(t,real(x7)); grid
>> ylabel('x_7(t)')
>> N = 15; k = -N:2:N;
>> ak = -2./(pi^2*k.^2); ak = [1/2 ak];
>> fk = 1*[0 k];
>> [x15,t] = fs_synth(fk, ak, 50*15, 3);
>> subplot(313); plot(t,real(x15))
>> grid
>> ylabel('x_{15}(t)');  xlabel('Time (s)')
```

- The triangle wave is continuous, so we expect the convergence of the partial sum $x_N(t)$ to be much better than for the square wave

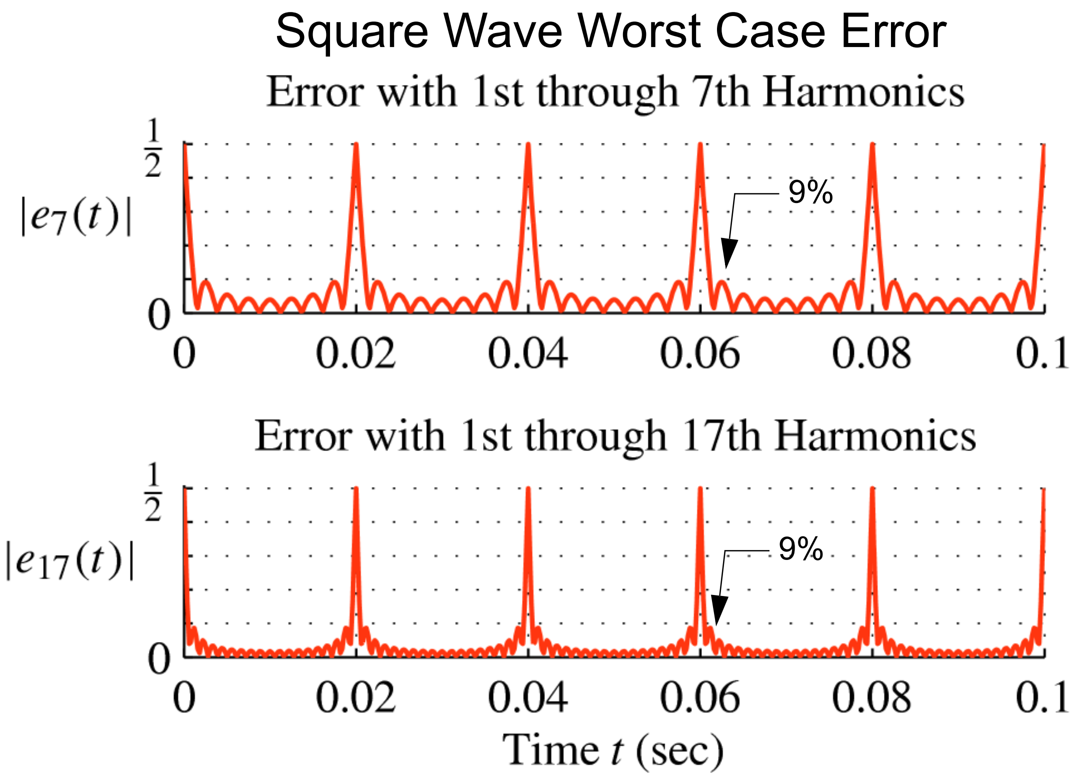


Convergence of Fourier Series

- For both the square wave and the triangle wave we have considered synthesis via the approximation $x_N(t)$
- We know that the approximation is not perfect, in particular for the square wave with the discontinuities, increasing N did not seem to result in that much improvement
- We can define the error between the true signal $x(t)$ and the approximation $x_N(t)$, as $e_N(t) = x(t) - x_N(t)$
- The worst case error can be defined as

$$E_{\text{worst}} = \max_{t \in [0,\, T_0]} |x(t) - x_N(t)| \tag{3.40}$$

- We can then plot this for various N values



- For the square wave the maximum error is always 1/2 the size of the jump, and the overshoot, either side of the jump, is always 9% of the jump

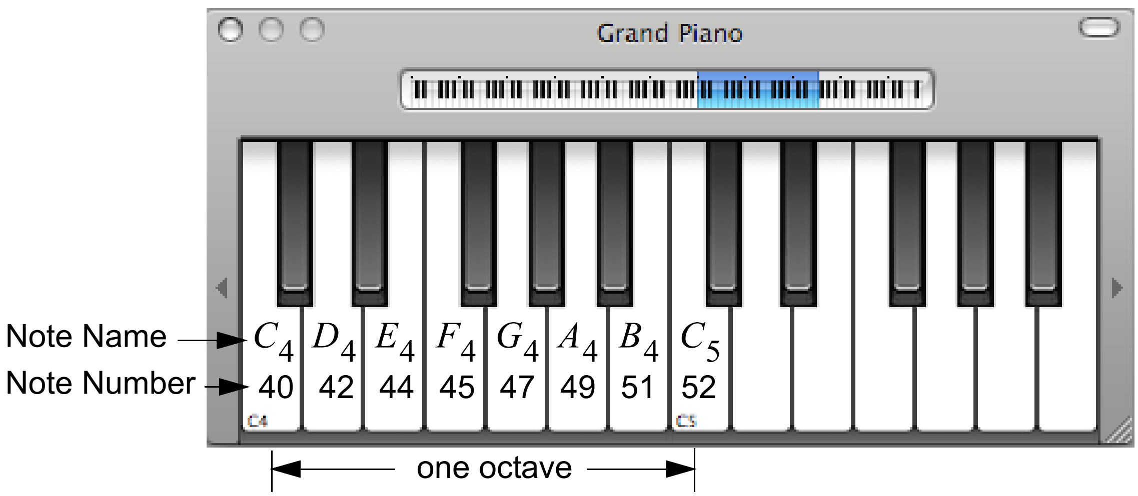
Time–Frequency Spectrum

- The past modeling and analysis has dealt with signals having parameters such as amplitude, frequency, and phase that *do not* change with time
- Most real world signals have parameters, such as frequency, that *do* change with time

- Speech and music are prime examples in our everyday life

Stepped Frequency

- A piano has 88 keys, with 12 keys per octave
  - An octave corresponds to the doubling of pitch/frequency
  - From one octave to the next there are 8 pitch steps, but there are also half steps (*flats* and *sharps*)



- A constant frequency ratio is maintained between all notes

$$r^{12} = 2 \Rightarrow r = 2^{1/12} = 1.0595$$

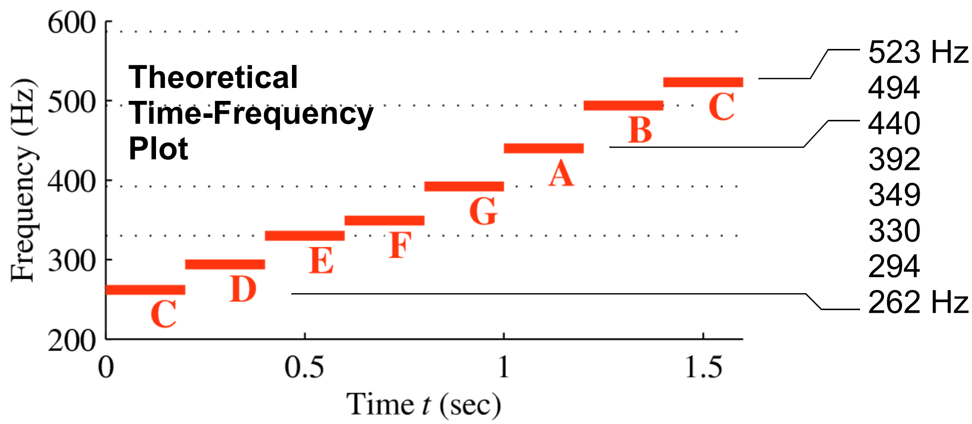
- The note A above middle C is at 440 Hz (tuning fork frequency) and is key number 49 of 88, so

$$f_{\text{middle C}} = f_{C_4} = 440 \times 2^{(40-49)/12} \cong 261.6 \text{ Hz}$$

- The C one octave above middle C is at key number 52, so

$$f_{C_5} = 440 \times 2^{(52-49)/12} \cong 523.3 \text{ Hz} = 2 \times 261.6$$

- A time-frequency plot can be used to display playing the notes in the C-major scale



Spectrogram Analysis

- The *spectrogram* is used to perform a time–frequency analysis on a signal, that is a plot of frequency content versus time, for a signal that has possibly time-varying frequencies
- When using MATLAB's signal processing toolbox, the function `specgram()` and `spectrogram()` are available for this purpose
  - The spfirst toolbox also has the function `plotspec()`
  - Both specgram() and plotspec() plot frequency versus time, whereas spectrogram() plots time versus frequency
  - The basic function interface to specgram() and plotspec is

    ```
    >> specgram(x,N_window,fsamp)
    >> plotspec(x,N_window,fsamp)
    ```

    where `N_window` is the length of the spectrum analysis window, typically 256, 512, or 1024, depending upon the desired frequency resolution and the rate at which the frequency content is changing

Example: C–Major Scale

- The MATLAB function `C_scale.m`, given below, is used to create the C–major scale running from middle C to one octave above middle C

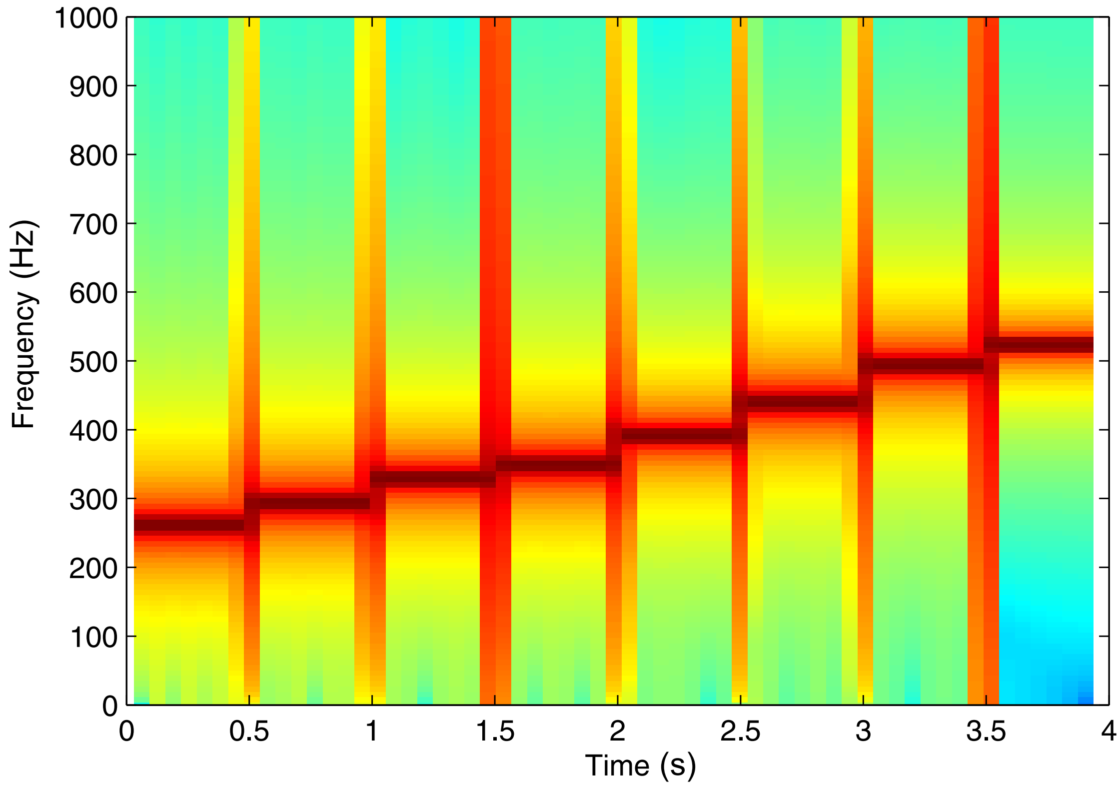
```
function [x,t] = C_scale(fs,note_dur)
% [x,t_final] = C_scale()
%
% Mark Wickert

% Generate octave middle C
pitch = [262 294 330 349 392 440 494 523];
N_pitch = length(pitch);

% Create a vector of frequencies
f = pitch(1)*ones(1,fix(note_dur*fs));
for k=2:N_pitch
    f = [f pitch(k)*ones(1,fix(note_dur*fs))];
end
t = [0:length(f)-1]/fs;
x = cos(2*pi*f.*t);
```

- We now call the function and plot the results using the `specgram` function

```
>> [x,t] = C_scale(8000,.5);
>> specgram(x,1024,8000);
>> axis([0 4 0 1000]) % reduce the frquency axis
```



- In this example the note duration is 0.5 s
- There is also a large smear of spectral information seen as the scale progression steps from note-to-note
- This is due to the way the spectrogram is computed
  – The analysis window straddles note changes, so a transient is captured where the pitch is jumping from one frequency to the next

Frequency Modulation: Chirp Signals

In the previous example we have seen how a sinusoidal waveform can have time varying frequency by stepping the frequency. Frequency modulation or angle modulation, provides another view on this subject within a particular mathematical framework.

Chirped or Linearly Swept Frequency

- A chirped signal is created when we sweep the frequency, according to some function, from a starting frequency to an ending frequency
- A constant frequency sinusoid is of the form

$$x(t) = \mathrm{Re}\{Ae^{j(\omega_0 t + \phi)}\} = A\cos(\omega_0 t + \phi) \tag{3.41}$$

- The argument of (3.41) is a time varying angle, $\psi(t)$, that is composed a linear term and a constant, i.e.,

$$\psi(t) = \omega_0 t + \phi = 2\pi f_0 t + \phi \tag{3.42}$$

- The units of $\psi(t)$ is radians
- If we differentiate $\psi(t)$ we obtain the instantaneous frequency

$$\omega_i(t) = \frac{d\psi(t)}{dt} = \omega_0 \text{ rad/s} \tag{3.43}$$

or by dividing by 2π the instantaneous frequency in Hz

$$f_i(t) = \frac{1}{2\pi}\frac{d\psi(t)}{dt} = f_0 \text{ Hz} \tag{3.44}$$

- The function $\psi(t)$ can take on different forms, but in particular it may be quadratic, i.e.,

$$\psi(t) = 2\pi\mu t^2 + 2\pi f_0 t + \phi \text{ rad} \tag{3.45}$$

which has corresponding instantaneous frequency

$$f_i(t) = 2\mu t + f_0 \text{ Hz} \tag{3.46}$$

- In this case we have a linear chirp, since the instantaneous frequency varies linearly with time

---

Example: Chirping from 100 to 1000 Hz in 1 s

- The beginning and ending times are $t_1 = 0\text{ s}$ and $t_2 = 1\text{ s}$
- We need to have

$$\begin{aligned} f_i(0) &= f_0 = 100 \text{ Hz} \\ f_i(1) &= 2\mu \cdot 1 + 100 = 1000 \text{ Hz} \end{aligned} \tag{3.47}$$

so $\mu = 900/2 = 450$

- Finally,

$$f_i(t) = 900t + 100 \text{ Hz}, \ 0 \le t \le 1 \text{ s} \tag{3.48}$$

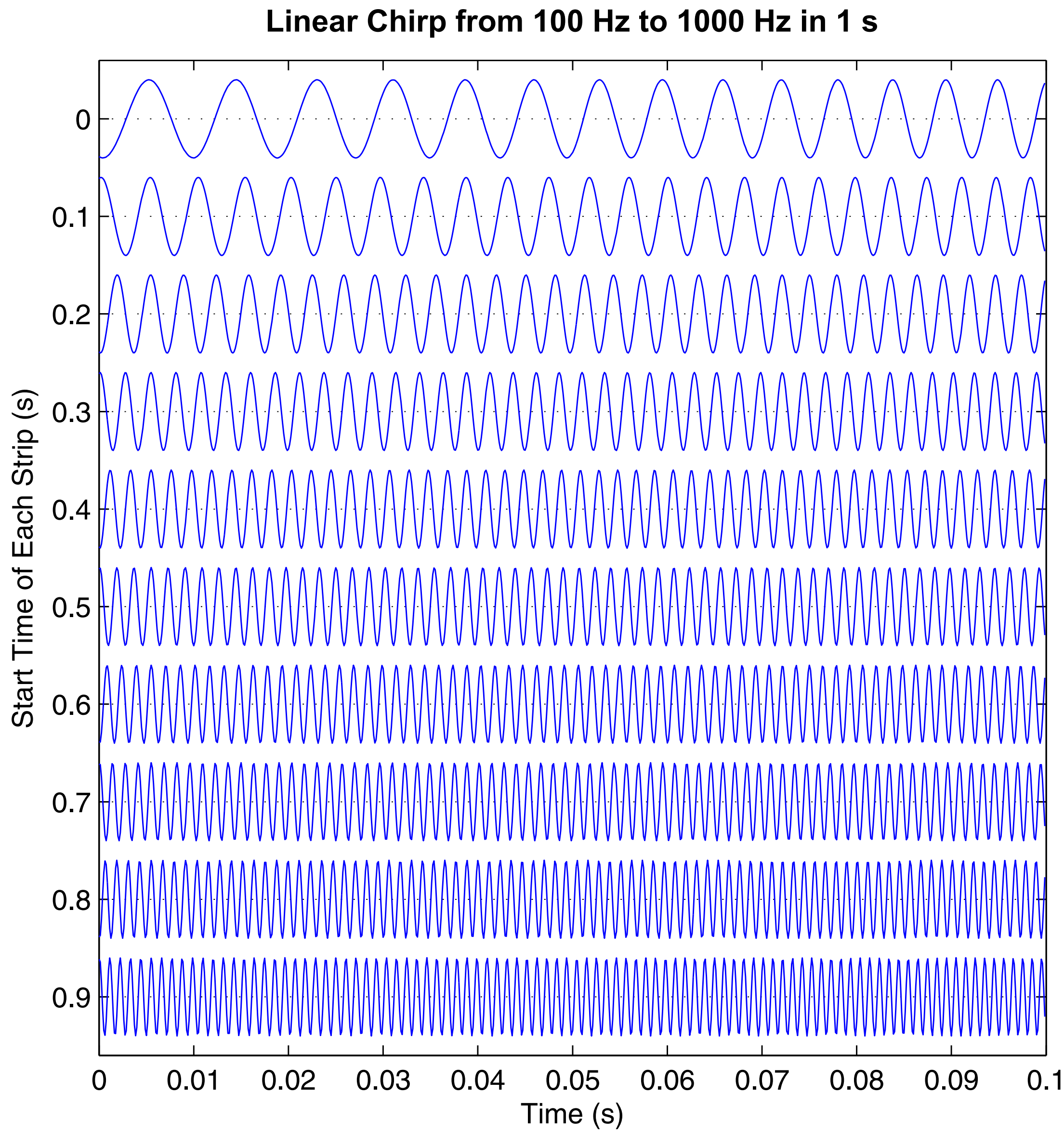
- The phase, $\psi(t)$, is

$$\psi(t) = 2\pi \cdot 450t^2 + 2\pi \cdot 100t + \phi \text{ rad} \tag{3.49}$$

- We can implement this in MATLAB as follows:

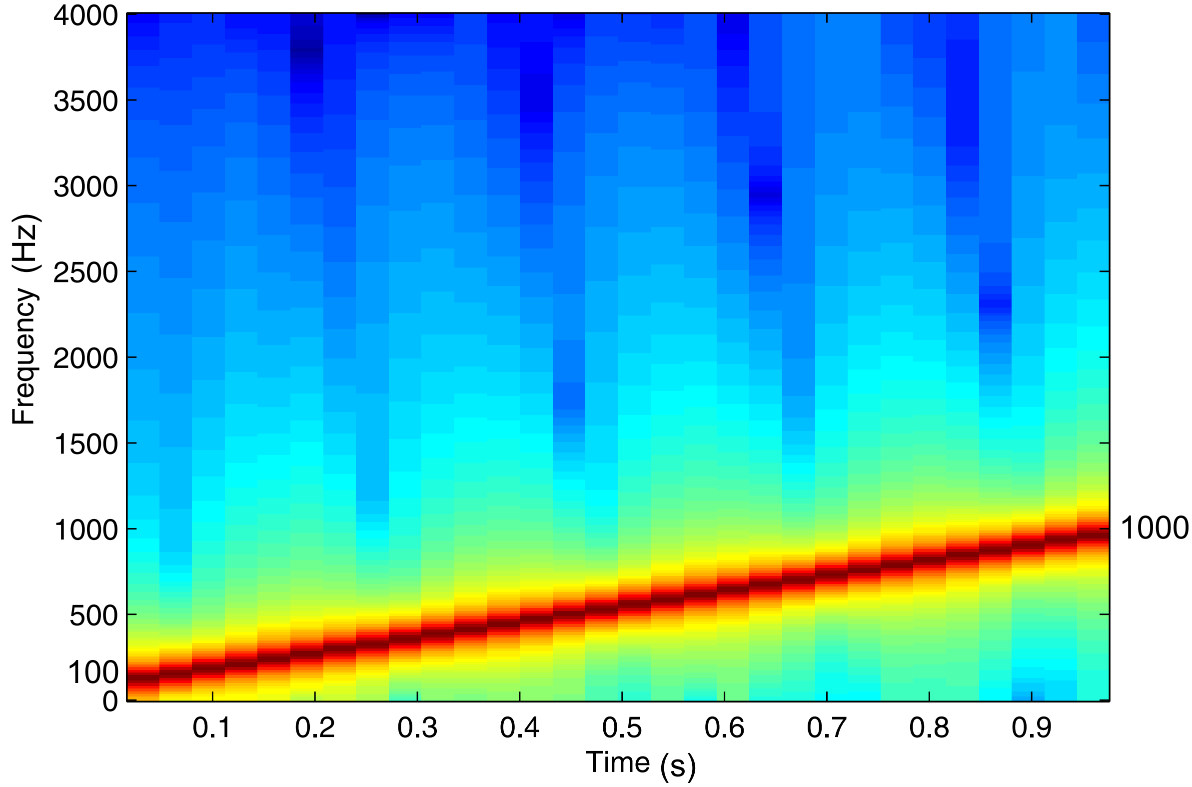
```
>> t = 0:1/8000:1;
```

```
>> x = cos(2*pi*450*t.^2 + 2*pi*100*t + 2*pi*rand(1,1));
>> plot(t,x)
>> strips(x,.2,8000)
>> xlabel('Time (s)')
>> ylabel('Start Time of Each Strip (s)')
```



- Using the `specgram` function we can obtain the time–frequency relationship

```
>> specgram(x,512,8000);
```



Summary

- The spectral representation of signals composed of sums of sinusoids was the main focus of this chapter
- The two-sided line spectra is the means to graphically display the spectra
- The concept of fundamental period and frequency was introduced, along with harmonic number
- The Fourier series was found to be a power tool for both analysis and synthesis of periodic signals

- For sinusoids with time-varying parameters, in particular frequency, the spectrogram is a useful graphical display tool
- Stepped frequency signals, such as a scale being played on a keyboard, is particularly clear when viewed as a spectrogram
- Frequency modulation, in particular linear chirp signals were briefly introduced