PHYS 2010 (W20)
Classical Mechanics

2020.03.05
 Relevant reading:
 Knudsen & Hjorth: X

Christopher Bergevin
York University, Dept. of Physics & Astronomy
Office: Petrie 240  Lab: Farq 103
cberge@yorku.ca

Refs:
Three of the four tangled stars are identical. Which one is different?
Looking Ahead...

Superposition

DDHO is a linear system

\[ m\ddot{x} + c\dot{x} + kx = \sum_{n} (m\ddot{x}_n + c\dot{x}_n + kx_n) = \sum_{n} F_n(t) = F_{ext} \]

Periodic forcing & superposition

Fourier analysis

\[ f(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \]

\[ a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) \, dt \quad n = 0, 1, 2, \ldots \]

\[ b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) \, dt \quad n = 1, 2, \ldots \]
Intuitively defined in two different (but equivalent) ways:

1. Time response of ‘system’ when subjected to an impulse
   (e.g., striking a bell w/ a hammer)

2. Fourier transform of resulting response
   (e.g., spectrum of bell ringing)

Ex. Harmonic oscillator

(Important) Note: The Fourier transform of the impulse response is called the *transfer function*
When dealing with linear oscillators (or linear systems in general), superposition takes a domineering position in how we approach analysis and modeling.
Superposition

"Double-slit experiment"

Thomas Young (1773-1829)

→ Wave theory of light
  (via phase interference)
Superposition

Each source:

\[ x_1 = A_1 \cos(\omega t + \alpha_1) \]
\[ x_2 = A_2 \cos(\omega t + \alpha_2) \]

Their sum (at the mic)

\[ x = x_1 + x_2 = A_1 \cos(\omega t + \alpha_1) + A_2 \cos(\omega t + \alpha_2) \]

\[ = A \cos(\omega t + \alpha) \]

A tad messy to solve for the constants there....

If instead we used complex exponentials:

\[ z_1 = A_1 e^{i(\omega t + \alpha_1)} \]
\[ z_2 = A_2 e^{i(\omega t + \alpha_2)} \]

Then:

\[ z = z_1 + z_2 = A_1 e^{i(\omega t + \alpha_1)} + A_2 e^{i(\omega t + \alpha_2)} \]

\[ = e^{i(\omega t + \alpha_1)} [A_1 + A_2 e^{i(\alpha_2 - \alpha_1)}] \]

→ This latter equation, while seemingly intimidating, tells us a lot!

French (1971)
Superposition: "Beats" (adding two different frequencies)

\[ x(t) = A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2) \]

\[ f_1=1, f_2=1.1 \]
\[ A_1=1, A_2=1 \]
\[ \phi_1=0, \phi_2=0 \]

→ Changing (relative) phase affects summation

\[ f_1=1, f_2=1.1 \]
\[ A_1=1, A_2=1 \]
\[ \phi_1=\pi/2, \phi_2=0 \]

→ Changing (relative) amplitudes affects summation

\[ f_1=1, f_2=1.1 \]
\[ A_1=2, A_2=1 \]
\[ \phi_1=0, \phi_2=0 \]
Recall: Lissajou figures

Jules Antoine Lissajous (1822-1880)
Tying this all back together....

One of the more fundamental/canonical problems in all areas of physics...

“mass-on-a-spring”

- One of the more fundamental/canonical problems in all areas of physics...
Fourier & the DDHO

Consider the driven case

\[ m \ddot{x} + c \dot{x} + kx = F_{\text{ext}} \]

where the driving force is a superposition of individual forces

\[ F_{\text{ext}} = \sum_{n} F_{n}(t) \]

\[ m \ddot{x}_{n} + c \dot{x}_{n} + kx_{n} = F_{n}(t) \]

Linearity allows us consider each separately...

... and simply add them all back up!

\[ x(t) = \sum_{n} x_{n}(t) \]

\[ m \ddot{x} + c \dot{x} + kx = \sum_{n} (m \ddot{x}_{n} + c \dot{x}_{n} + kx_{n}) = \sum_{n} F_{n}(t) = F_{\text{ext}} \]

For linear systems, the "door" of superposition swings both ways!
Fourier & the DDHO

Now, can we make a smart choice for what all those "individual" forces are?

$$F_{ext} = \sum_{n} F_n(t)$$

In many instances, the driving force is something that eventually repeats itself*

$$F_{ext}(t) = F_{ext}(t + T)$$

* - Note that much of what we discuss here also usefully applies even when a given "signal" is not strictly periodic (or even remotely close to such!)
Periodic forcing....
Fourier & the DDHO

Now, can we make a smart choice for what all those "individual" forces are?

In many instances, the driving force is something that eventually repeats itself

Then it is fairly natural to represent such a function via an infinite series of sinusoids (i.e., a Fourier series)
Fourier & the DDHO

\[ f(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} \left[ a_n \cos(n\omega t) + b_n \sin(n\omega t) \right] \]

Just like a Taylor series expansion, there are a unique set of coefficients \((a_n \text{ and } b_n)\) that determine the appropriate amplitude and phase of a given frequency term (i.e., \(n\omega\))

\[
a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) \, dt \quad n = 0, 1, 2, \ldots
\]

\[
b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) \, dt \quad n = 1, 2, \ldots
\]

There is a reasonably clear "recipe" for these coefficients...

We can also use complex numbers to manage "bookkeeping" more efficiently....
Fourier analysis

- Deep history throughout mathematics, physics, engineering, biology, ..... 

- Backbone of modern signal processing and linear systems theory 

- Lays at foundation of many modern methodologies in medical imaging (e.g., MRI, CT scans) 

- Builds off the basic idea of a *Taylor series* (which posits we can describe a function as an infinite series of polynomials)

**Basic idea:** Represent ‘signal’ as a sum of sinusoids

**Note:** We focus on 1-D here for clarity, but these ideas generalize to higher dimensions (e.g., 2-D for images)
Key idea: Fourier transform

- Allows one to go from a time domain description (e.g., recorded signal) to a spectral description (i.e., what frequency components make up that signal)

- One axis is time, the other is frequency
- These two are fundamentally tied together
Intuitive connection back to Taylor series:

\[ y(x_1 + \Delta x) \approx y(x_1) + \sum_{n=1}^{N} \frac{1}{n!} \frac{d^n y}{dx^n} \bigg|_{x_1} (\Delta x)^n. \quad (D.2) \]

\[
f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!} (x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n + \cdots
\]

\[
= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n
\]

**Taylor series** → Expand as a (infinite) sum of polynomials

**Different Idea: Fourier series** → Expand as a (infinite) sum of sinusoids
“The exponential function $e^x$ (in blue), and the sum of the first $n+1$ terms of its Taylor series at 0 (in red).”
Fourier series

\[ f(t) = a_o + a_1 \sin (\omega t) + b_1 \cos (\omega t) + \]
\[ + a_2 \sin (2\omega t) + b_2 \cos (2\omega t) + \]
\[ + a_3 \sin (3\omega t) + b_3 \cos (3\omega t) + \cdots \]

\[ = A_0 + A_1 \sin (\omega t + \phi_1) + \]
\[ + A_2 \sin (2\omega t + \phi_2) + \]
\[ + A_3 \sin (3\omega t + \phi_3) + \cdots \]

\[ = \sum_{n=0}^{\infty} A_n \sin (n\omega t + \phi_n) \]

\[ a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) \, dt \quad n = 0, 1, 2, \ldots \]

\[ b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) \, dt \quad n = 1, 2, \ldots \]

\[ = \sum_{n=0}^{\infty} B_n e^{in\omega t} \quad \text{where } B_n \in \mathbb{C}, \ i = \sqrt{-1} \]

Complex #s are much more compact and easier to deal with
e.g., Square "Waves"

Big peak (call it $f_0$) is the fundamental. Smaller peaks are harmonics (e.g., $3f_0$).

Ignore this other (symmetrical) half for now (arises as a mathematical redundancy).
% Visually demonstrate the build-up of a square wave
% by adding successive (user-specified) terms of the Fourier series
% expansion; also quantifies the Gibbs phenomenon

clear

% array of # of terms to compute (1 2 4 8 15 25 100 500)
P.order= [1 2 4 8 15 25 100 500];
% period (1)
P.tau= 1;
% peak-to-peak amplitude (1)
P.A= 1;
% total # of points per interval (must be even?) (1000)
P.M= 10000;
% time to pause between displaying new iterates [s] (0.5)
P.pause= 0.5;

% time array
\[ t = \text{linspace}(-1.5 \cdot P.tau, 1.5 \cdot P.tau, 3 \cdot P.M); \]
% create sawtooth baseline
\[ \text{squareT} = \text{repmat}(P.A \cdot ([\text{zeros}(P.M/2,1); \text{ones}(P.M/2,1)]-0.5), 3, 1)'; \]

% use a loop to add in the terms
for mm=1:numel(P.order)
    tempN= P.order(mm);
    squareF= 0; % dummy initial indexer
    for nn= 1:tempN
        nextTerm= (2*P.A/pi)*(1/(2*nn-1))*sin((2*nn-1)*(2*pi/P.tau)*t);  % create next term in series
        squareF= squareF+ nextTerm;
    end
    % --- estimate "overshoot"
    [M, indx]= max(squareF);
    disp(['Overshoot ratio ~', num2str(M/max(squareT))]);
end

% visualize
figure(1); clf;
\[ h1 = \text{plot}(t, \text{squareT}, 'b-', 'LineWidth', 2); \]
hold on; grid on; xlabel('x'); ylabel('y'); ylim(0.65*P.A*[-1 1]);
\[ h2 = \text{plot}(t, \text{squareF}, 'r--', 'LineWidth', 2); \]
legend([h1 h2], 'square', 'Fourier series', 'Location', 'NorthWest');
title(['(truncated) Fourier reconstruction of square wave w/ ', num2str(tempN), ' terms']);
pause(P.pause);
Recall: Impulses

\[ \delta(x) = \begin{cases} 
+\infty, & x = 0 \\
0, & x \neq 0 
\end{cases} \]

\[ \int_{-\infty}^{\infty} \delta(x) \, dx = 1 \]

Delta functions are much more ubiquitous in your lives (and science) than you might think/realize......
clear; clf;

% --------------------------------
SR= 44100;         % sample rate [Hz]
Npoints= 8192;     % length of fft window (# of points) [should ideally be 2^N]
                % [time window will be the same length]
INDXon= 1000;     % index at which click turns 'on' (i.e., go from 0 to 1)
INDXoff= 1001;   % index at which click turns 'off' (i.e., go from 1 to 0)
% --------------------------------
dt= 1/SR;  % spacing of time steps
freq= [0:Npoints/2];    % create a freq. array (for FFT bin labeling)
freq= SR*freq./Npoints;
t=[0:1/SR:(Npoints-1)/SR]; % create an appropriate array of time points
% build signal
clktemp1= zeros(1,Npoints); clktemp2= ones(1,INDXoff-INDXon);
signal= [clktemp1(1:INDXon-1) clktemp2 clktemp1(INDXoff:end)];
% ------------------------------
% *******
% plot "final" time waveform of signal
if 1==1
    figure(3); clf; plot(t*1000,signal,'ko-','MarkerSize',5)
    grid on; hold on; xlabel('Time [ms]'); ylabel('Signal'); title('Time Waveform')
end
% *******
% now compute/plot FFT of the signal
sigSPEC= rfft(signal);
% MAGNITUDE
figure(1); clf;
subplot(211); plot(freq/1000,db(sigSPEC),'ko-','MarkerSize',3)
hold on; grid on; ylabel('Magnitude [dB]'); title('Spectrum (or "Look Up Table")')
% PHASE
subplot(212); plot(freq/1000,cycs(sigSPEC),'ko-','MarkerSize',3)
xlabel('Frequency [kHz]'); ylabel('Phase [cycles]'); grid on;
% *******
% now make animation of click getting built up, using the info from the FFT
sum= zeros(1,numel(t)); % (initial) array for reconstructed waveform
inclV=[1:30,floor(linspace(31,floor(0.9*numel(freq)),100)),...
     floor(linspace(0.9*numel(freq),numel(freq),20))];
figure(2); clf; grid on;
for nn=1:numel(freq)
    sum= sum+ abs(sigSPEC(nn))*cos(2*pi*freq(nn)*t + angle(sigSPEC(nn)));
    if ismember(nn,inclV),  plot(t,sum,'LineWidth',2); grid on; xlabel('Time [s]');
    legend([{'Highest freq= ',num2str(freq(nn)/1000),' kHz'}])
    pause(3/(nn));  end
end
Signal is an ‘impulse’ (i.e., a delta function)

Spectral representation has flat amplitude and a ‘group delay’

Reconstruct waveform by adding sinusoids (only lowest frequency here)

Now the first 15 terms are included

→ Eventually all the sinusoids add up such that things cancel out everywhere except at the point of the impulse!
Summary

Superposition

DDHO is a linear system

\[ m\ddot{x} + c\dot{x} + kx = \sum_n (m\ddot{x}_n + c\dot{x}_n + kx_n) = \sum_n F_n(t) = F_{\text{ext}} \]

Periodic forcing & superposition

Fourier analysis

\[ f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} \left[ a_n \cos(n\omega t) + b_n \sin(n\omega t) \right] \]

\[ a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) \, dt \quad n = 0, 1, 2, \ldots \]

\[ b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) \, dt \quad n = 1, 2, \ldots \]
% ### EXspecREP3.m ###       10.29.14
% Example code to just fiddle with basics of discrete FFTs and connections
% back to common real-valued time waveforms
% --> Demonstrates several useful concepts such as 'quantizing' the frequency
% Requires: rfft.m, irfft.m, cycs.m, db.m, cyc.m
% -----
% Stimulus Type Legend
% stimT= 0 - non-quantized sinusoid
% stimT= 1 - quantized sinusoid, one un-quantized sinusoid
% stimT= 2 - two quantized sinusoids
% stimT= 4 - click I.e., an impulse
% stimT= 5 - noise (uniform in time)
% stimT= 6 - chirp (flat mag.)
% stimT= 7 - noise (Gaussian; flat spectrum, random phase)
% stimT= 8 - exponentially decaying sinusoid (i.e., HO impulse response)

clear; clf;
% --------------------------------
SR= 44100;         % sample rate [Hz]  
Npoints= 8192;     % length of fft window (# of points) [should ideally be 2^N] (time window will be the same length)
stimT= 8;   % Stimulus Type (see legend above)
stimT= 8;   % note: other stimulus parameters can be changed below
% --------------------------------
% spacing of time steps
dt= 1/SR;  
% quantize the freq. (so to have an integral # of cycles in time window)
fmt= SR*fmem; % Frequency (for waveforms w/ tones) [Hz]
fmt= fmem*fmt; % specify f2/f2 ratio (for waveforms w/ two tones)
% Note: Other stimulus parameters can be changed below
% --------------------------------
% compute stimulus
if stimT==0 % non-quantized sinusoid
  signal= cos(2*pi*fmt*t);
  disp(sprintf(' 
 *Stimulus* - (non-quantized) sinusoid, f = %g Hz 
', fmt));
  disp(sprintf('specified freq. = %g Hz', fmt));
elif stimT==1     % quantized sinusoid
  signal= cos(2*pi*fmttmem*t);
  disp(sprintf(' 
 *Stimulus* - quantized sinusoid, f = %g Hz 
', fmttmem));
  disp(sprintf('specified freq. = %g Hz', fmttmem));
  disp(sprintf('quantized freq. = %g Hz', fmttmem));
elseif stimT==2     % one quantized sinusoid, one un-quantized sinusoid
  signal= cos(2*pi*fmttmem*t) + cos(2*pi*fmttmem2*t);
  disp(sprintf(' 
 *Stimulus* - two sinusoids (one quantized, one not) 
'));
elseif stimT==3     % two quantized sinusoids
  fmt2= ceil(fmt2/dfmem)*dfmem; % quantized start/end freqs. (necessary?)
  fmt2SQ= ceil(fmt2SQ/dfmem)*dfmem; % LINEAR sweep rate
  fmt2EQ= ceil(fmt2EQ/dfmem)*dfmem;
  signal = sin(2*pi*fmt2*t)' + cos(2*pi*fmt2*t); 
  disp(sprintf(' 
 *Stimulus* - two sinusoids (both quantized) 
'));
elseif stimT==4     % click
  alpha= 500;     % index at which click turns 'on' (starts at 1)
  signal= exp(-alpha*t).*sin(2*pi*fmttmem*t);
  disp(sprintf(' 
 *Exponentially decaying (quantized) sinusoid*  
'));
elseif stimT==7     % noise (Gaussian)
  Asize=Npoints/2 +1;
  % create array of complex numbers w/ random phase and unit magnitude
  for n=1:Asize
    theta= rand*2*pi;
    N2(n)= exp(i*theta);
  end
  N2=N2';
  % now take the inverse FFT of that using Chris' irfft.m code
  tNoise=irfft(N2);
  if (abs(min(tNoise)) > max(tNoise))
    tNoise= tNoise/abs(min(tNoise));
  else
    tNoise= tNoise/max(tNoise);
  end
  signal= tNoise;
  disp(sprintf(' 
 *Noise* - Gaussian, flat-spectrum 
'));
elseif stimT==8 % exponentially decaying cos
  alpha= 500;
  signal= exp(-alpha*t).*sin(2*pi*fmttmem*t);
  disp(sprintf(' 
 *Exponentially decaying (quantized) sinusoid*  
'));
endif
% ------------------------------
% *******
figure(1); clf % plot time waveform of signal
plot(t*1000,signal,'k.-','MarkerSize',5); grid on; hold on;
xlabel('Time [ms]'); ylabel('Signal'); title('Time Waveform')
% *******
% now plot rfft of the signal
% NOTE: rfft just takes 1/2 of fft.m output and normalizes
sigSPEC= rfft(signal);
figure(2); clf; % MAGNITUDE
subplot(211)
plot(freq/1000,db(sigSPEC),'ko-','MarkerSize',3)
hold on; grid on;
ylabel('Magnitude [dB]')
title('Spectrum')
subplot(212)
plot(freq/1000,cycs(sigSPEC),'ko-','MarkerSize',3)
hold on;
ylabel('Phase [cycles]'); grid on;
% -------
% play the stimuli as an output sound?
if (1==1),  sound(signal,SR);   end
% -------
% compute inverse Fourier transform and plot?
if 1==1
  figure(1);
  signalINV= irfft(sigSPEC);
  plot(t*1000,signalINV,'rx','MarkerSize',4)
  legend('Original waveform','Inverse transformed')
endif
% ---
Fourier transforms of basic (1-D) waveforms

stimT = 0 - non-quantized sinusoid

SR = 44100;  % sample rate [Hz]
Npoints = 8192;  % length of fft window

Note: The phase is ‘unwrapped’ in all the spectral plots

- Magnitude shows a peak at the sinusoid’s frequency
Fourier transforms of basic (1-D) waveforms

stimT= 3 – two quantized sinusoids

- Magnitude shows two peaks (note the ‘beating’ in the time domain)
Fourier transforms of basic (1-D) waveforms

stimT = 4 - click (i.e., an impulse)

➢ Click has a flat magnitude (This is also a good place to mention the concept of a ‘group delay’)

Click has a flat magnitude (This is also a good place to mention the concept of a ‘group delay’)
Fourier transforms of basic (1-D) waveforms

stimT = 5 - noise (uniform distribution)

- Magnitude is flat-ish (on log scale), but actually noisy. Phase is noisy too.
Fourier transforms of basic (1-D) waveforms

stimT = 7 - noise (Gaussian distribution)

- Magnitude is flat just like an impulse (i.e., flat), but the phase is random
Fourier transforms of basic (1-D) waveforms

→ Remarkable that the magnitudes are identical (more or less) between two signals with such different properties. The key difference here is the phase: *Timing is a critical piece of the puzzle!*
Fourier transforms of basic (1-D) waveforms

stimT = 6 - chirp (flat mag.)

Hard to see on this timescale, but frequency is changing (increasing) with time
Fourier transforms of basic (1-D) waveforms

stimT= 8 - exponentially decaying sinusoid

Time domain

Spectral domain

➢ This seems to look familiar....