BRI509: Introduction to Brain Signal Processing
Assignment No. 3

CANOY RAYMART JAY
Student ID #: **2020021376**

October 25, 2022

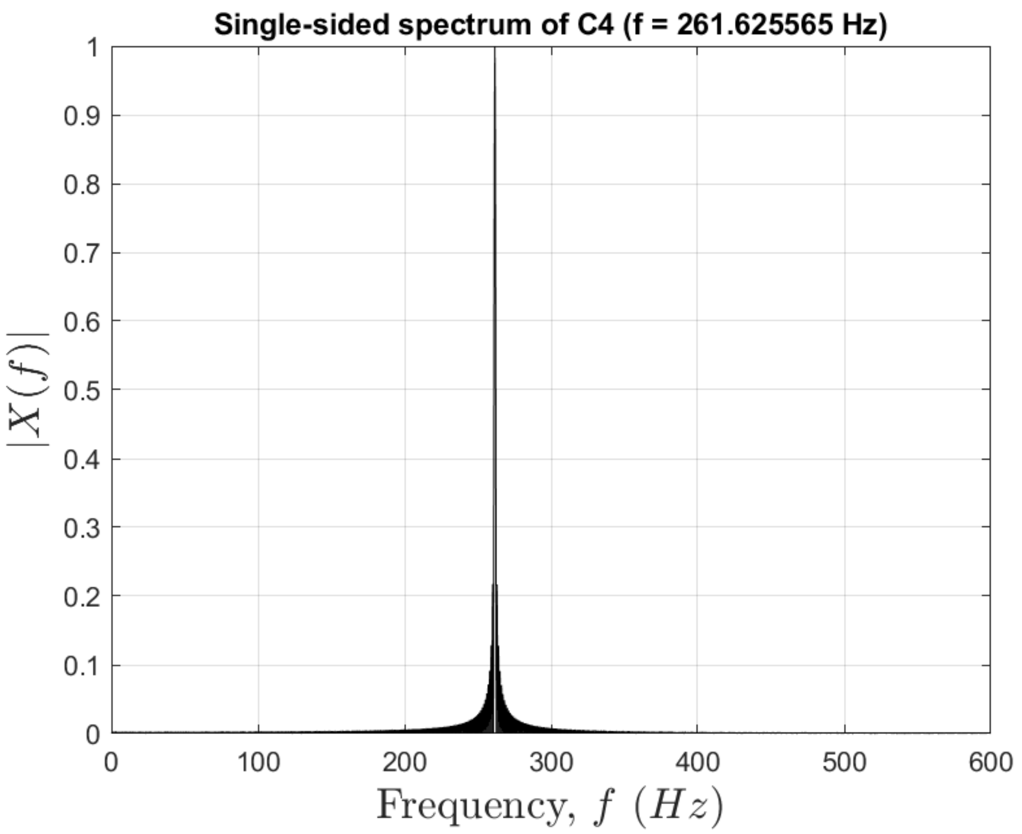
(a) Draw the Bode diagrams of Do(C4), Mi(E4), Sol(G4) and their chord(C4+E4+G4).

Source Code

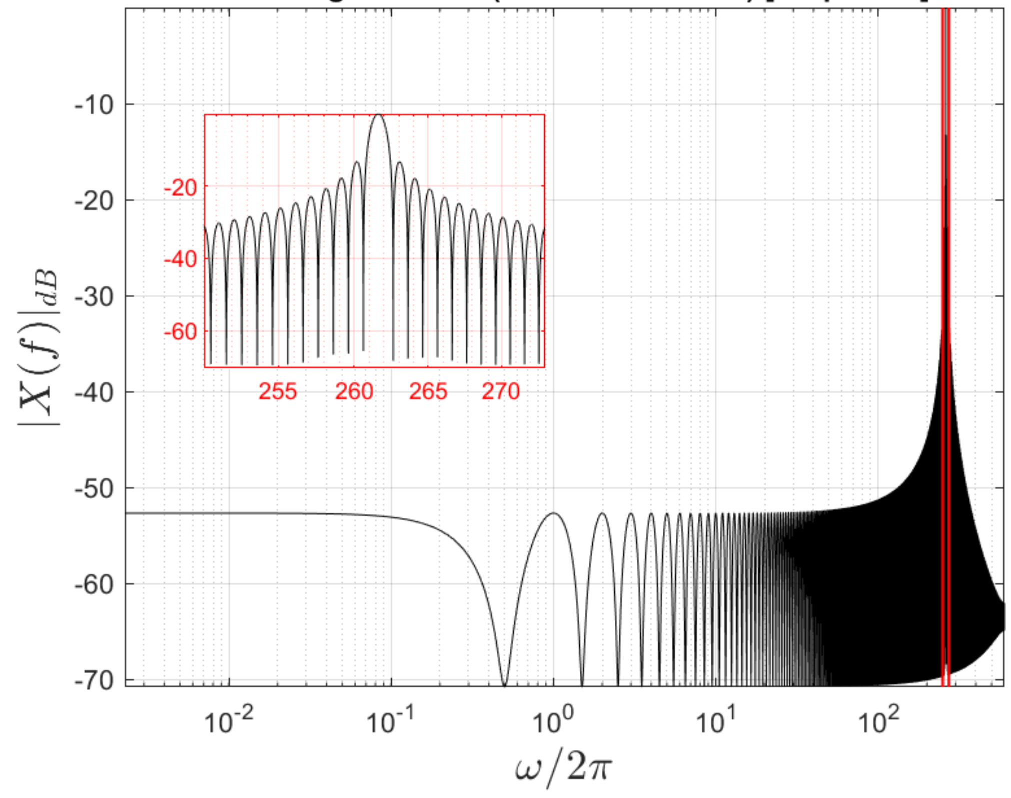
```matlab
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 3
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

% Notes:
% "The frequencies 440 Hz and 880 Hz both correspond to the
    musical note A,
% but one octave apart. In the western musical scale, there
    are 12 notes in
% every octave. These notes are evenly distributed (
    geometrically), so the
% next note above A, which is B flat, has frequency (440 x
    beta), where
% beta is the twelfth root of two."
% Source: https://ptolemy.berkeley.edu/eecs20/week8/scale.html

%% Draw the Bode diagrams of:
%  (a) Do(C4)
%  (b) Mi(E4)
%  (c) Sol(G4)
%  (d) Chord(C4+E4+G4)

% (00) Initialization
clear; clc;

fs = 1200;
T = 1/fs;
dur = 1;
L = dur*fs;
t = (0:L-1)*T;

% (01) Sinusoid signal
beta = 2^(1/12);
beta_exponents = [-9, -7, -5, -4, -2, 0, 2, 3];
note = @(index) sin(2*pi*(440*(beta^(index)))*t);

```

```matlab
% C major
musical_scale_keys = {'C4', 'D4', 'E4', 'F4', 'G4', 'A4', 'B4', 'C5'};
musical_scale_values = cell(length(musical_scale_keys), 1);

for ind = 1 : length(musical_scale_keys)
    musical_scale_values{ind} = note(beta_exponents(ind));
end

% (02) Sofege for C major
music_octave = containers.Map(musical_scale_keys, musical_scale_values);

%% (03) Bode Diagram
assignment = {'C4', 'E4', 'G4'};
beta_exponents_assigment = [-9, -5, -2];
chord = music_octave('C4') + music_octave('E4') + music_octave('G4');

for i = 1 : 4
    % Fourier Transform
    n = 256*2^(nextpow2(L));
    if i < 4
        x = music_octave(assignment{i});
    else
        x = chord;
    end
    X = fftshift(fft(x, n, 2));

    f = [0: fs/n : fs/2 - fs/n];
    PdB = 20*log10(2*abs(X(end/2+1:end)./L));
    phase = rad2deg(angle(X(end/2+1:end)));

    A = [ones(size(f, 2), 1), [1:1:size(f, 2)]'];
    theta = pinv(A'*A)*(A'*f');

    if i < 4
        f_center_index = round((440*(beta^(beta_exponents_assigment(i))) - theta(1))/theta(2));
    else
        f_center_index = 144017;
    end
    figure,
    p1 = plot([0: fs/n : fs/2 - fs/n], 2*abs(X(end/2+1:end)./L), 'k')
    grid on
    xlim([0 fs/2])
    ylim([0 1])
    xlabel('Frequency, $f$ ($Hz$)', 'Interpreter', 'Latex', 'FontSize', 16)
    ylabel('$|X(f)|$', 'Interpreter', 'Latex', 'FontSize', 16)
    if i < 4
        title(sprintf('Single-sided spectrum of %s (f = %2f Hz)', assignment{i}, 440*(beta^(beta_exponents_assigment(i)))))
    else
        title('Single-sided spectrum of chord (C4+E4+G4)')
    end
    saveas(gcf, sprintf('FFT_Prob%02d.png', i))

    figure;
    ax1 = axes();
    plot(ax1, f, PdB, 'k');
    hold on
    if i < 4
        plot(ax1, f(f_center_index - 5000)*ones(1200, 1), linspace(min(PdB), max(PdB), 1200), 'r', 'LineWidth', 1);
        plot(ax1, f(f_center_index + 5000)*ones(1200, 1), linspace(min(PdB), max(PdB), 1200), 'r', 'LineWidth', 1);
    else
        plot(ax1, f(f_center_index - 35000)*ones(1200, 1), linspace(min(PdB), max(PdB), 1200), 'r', 'LineWidth', 1);
        plot(ax1, f(f_center_index + 35000)*ones(1200, 1), linspace(min(PdB), max(PdB), 1200), 'r', 'LineWidth', 1);
    end
    set(gca, 'xscale', 'log')
    grid on
    xlim([min(f) max(f)])
    ylim([min(PdB) max(PdB)])
    xlabel('$\omega / 2\pi$', 'Interpreter', 'Latex', 'FontSize', 16)
    ylabel('$|X(f)|_{dB}$', 'Interpreter', 'Latex', 'FontSize', 16)
    if i < 4
        title(sprintf('Bode Diagram of %s (f = %2f Hz) [Amplitude]', assignment{i}, 440*(beta^(beta_exponents_assigment(i)))))
    else
        title('Bode Diagram of chord (C4+E4+G4) [Amplitude]')
    end
    ax2 = axes('Position', [0.2 0.5 0.3 0.3]);
    ax2.XColor = 'red';
    ax2.YColor = 'red';
    if i < 4
        plot(ax2, f(f_center_index - 5000:f_center_index+5000), PdB(f_center_index - 5000:f_center_index+5000), 'k')
    else
        plot(ax2, f(f_center_index - 35000:f_center_index+35000), PdB(f_center_index - 35000:f_center_index+35000), 'k')
    end
    set(gca, 'xscale', 'log', 'XColor', 'red', 'YColor', 'red')
    grid on
    ylim([min(PdB) max(PdB)])
    saveas(gcf, sprintf('Bode_diagram%02d.png', i))

    figure;
    ax1 = axes();
    plot(ax1, f, phase, 'k');
    hold on
    if i < 4
        plot(ax1, f(f_center_index - 5000)*ones(1200, 1), linspace(min(phase), max(phase), 1200), 'r', 'LineWidth', 1);
        plot(ax1, f(f_center_index + 5000)*ones(1200, 1), linspace(min(phase), max(phase), 1200), 'r', 'LineWidth',
   1);
    else
        plot(ax1, f(f_center_index - 35000)*ones(1200, 1),
   linspace(min(phase), max(phase), 1200), 'r', 'LineWidth',
   1);
        plot(ax1, f(f_center_index + 35000)*ones(1200, 1),
   linspace(min(phase), max(phase), 1200), 'r', 'LineWidth',
   1);
    end
    set(gca, 'xscale', 'log')
    grid on
    xlim([min(f) max(f)])
    ylim([min(phase) max(phase)])
    xlabel('$\omega / 2\pi$', 'Interpreter', 'Latex', '
   FontSize', 16)
    ylabel('$\angle X(f)$ (deg)', 'Interpreter', 'Latex', '
   FontSize', 16)
    if i < 4
        title(sprintf('Bode Diagram of %s (f = %2f Hz) [Phase]
   ', assignment{i}, 440*(beta^(beta_exponents_assigment(i))))
   )
    else
        title('Bode Diagram of chord (C4+E4+G4) [Phase]')
    end
    ax2 = axes('Position', [0.177 0.16 0.3 0.3]);
    ax2.XColor = 'red';
    ax2.YColor = 'red';
    if i < 4
        plot(ax2, f(f_center_index - 5000:f_center_index+5000)
   , phase(f_center_index - 5000:f_center_index+5000), 'k')
    else
        plot(ax2, f(f_center_index - 35000:f_center_index
   +35000), phase(f_center_index - 35000:f_center_index+35000)
   , 'k')
    end
    set(gca, 'xscale', 'log', 'XColor', 'red', 'YColor', 'red'
   )
    grid on
    ylim([min(phase) max(phase)])
    saveas(gcf, sprintf('Bode_diagram_phase%02d.png', i))
end


%% (04) Audio Files
% (00) Initialization
clear; clc;

fs = 48000;
T = 1/fs;
dur = 1;
L = dur*fs;
t = (0:L-1)*T;

% (01) Sinusoid signal
beta = 2^(1/12);
beta_exponents = [-9, -7, -5, -4, -2, 0, 2, 3];
note = @(index) sin(2*pi*(440*(beta^(index)))*t);

% C major
musical_scale_keys = {'C4', 'D4', 'E4', 'F4', 'G4', 'A4', 'B4'
    , 'C5'};
musical_scale_values = cell(length(musical_scale_keys), 1);

for ind = 1 : length(musical_scale_keys)
    musical_scale_values{ind} = note(beta_exponents(ind));
end

% (02) Sofege for C major
music_octave = containers.Map(musical_scale_keys,
    musical_scale_values);

assignment = {'C4', 'E4', 'G4'};
chord = music_octave('C4') + music_octave('E4') + music_octave
    ('G4');

for i = 1 : 4
    if i < 4
        mp3_file = [zeros(1, 100) music_octave(assignment{i})
    zeros(1, 100)]';
        audiowrite(sprintf('
    Assigment3_2020021376_CanoyRaymartJay_%s.mp4', assignment{i
    }), mp3_file, fs)
    else
        mp3_file = [zeros(1, 100) chord zeros(1, 100) zeros(1,
     100)]';
        audiowrite(sprintf('
    Assigment3_2020021376_CanoyRaymartJay_%s.mp4', 'chord'),
    mp3_file, fs)
    end
end
```

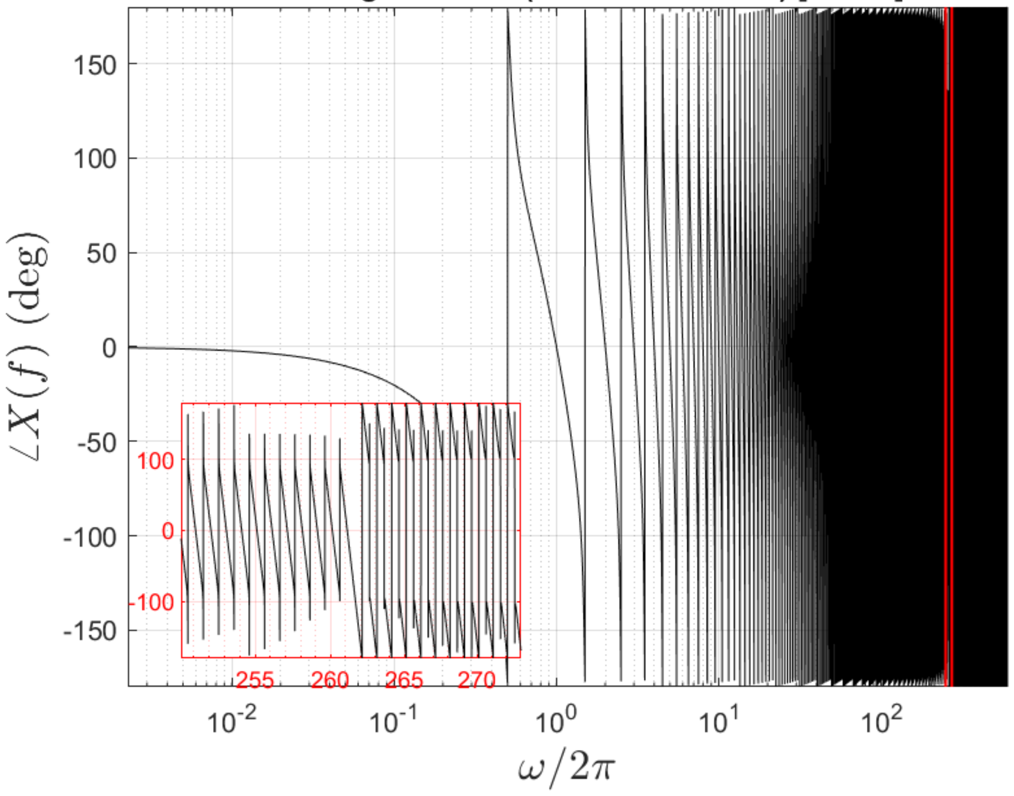
(b) Bode diagram of C4



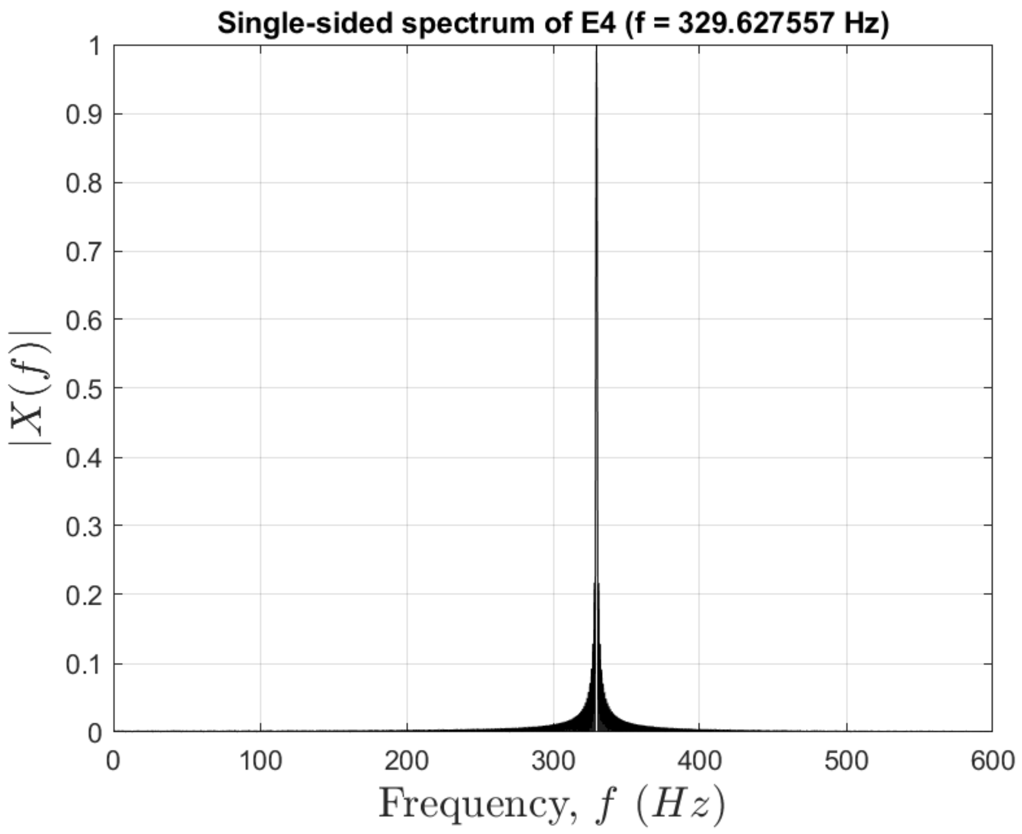
Bode Diagram of C4 (f = 261.625565 Hz) [Amplitude]

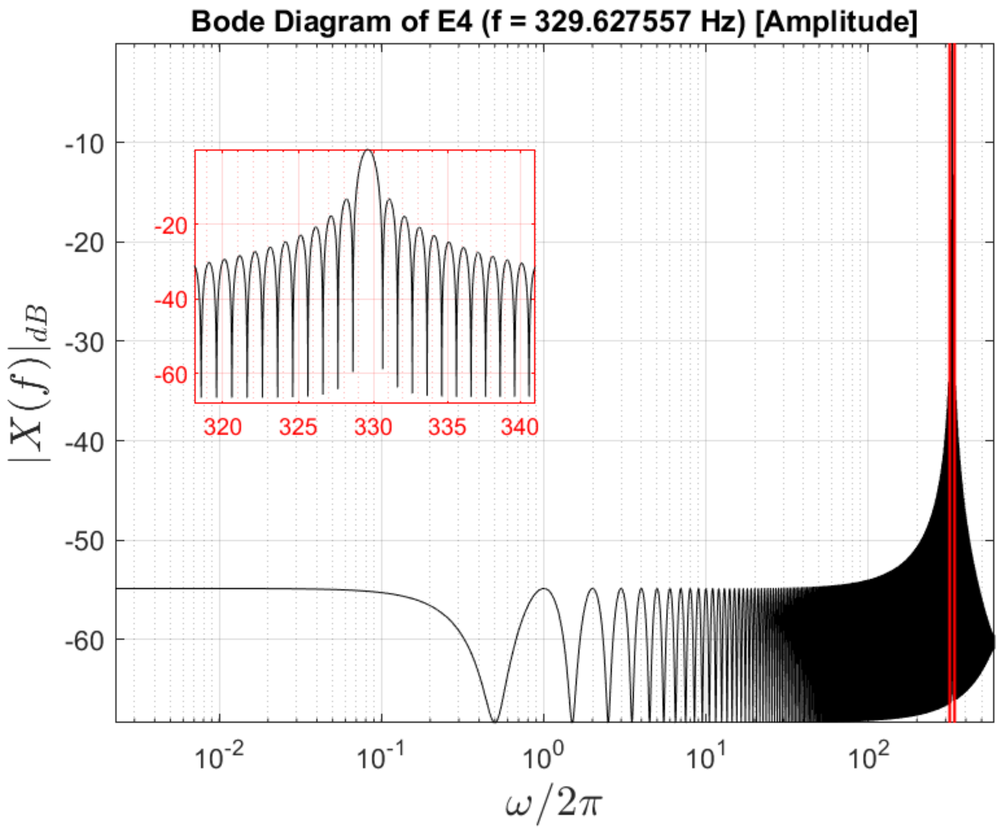


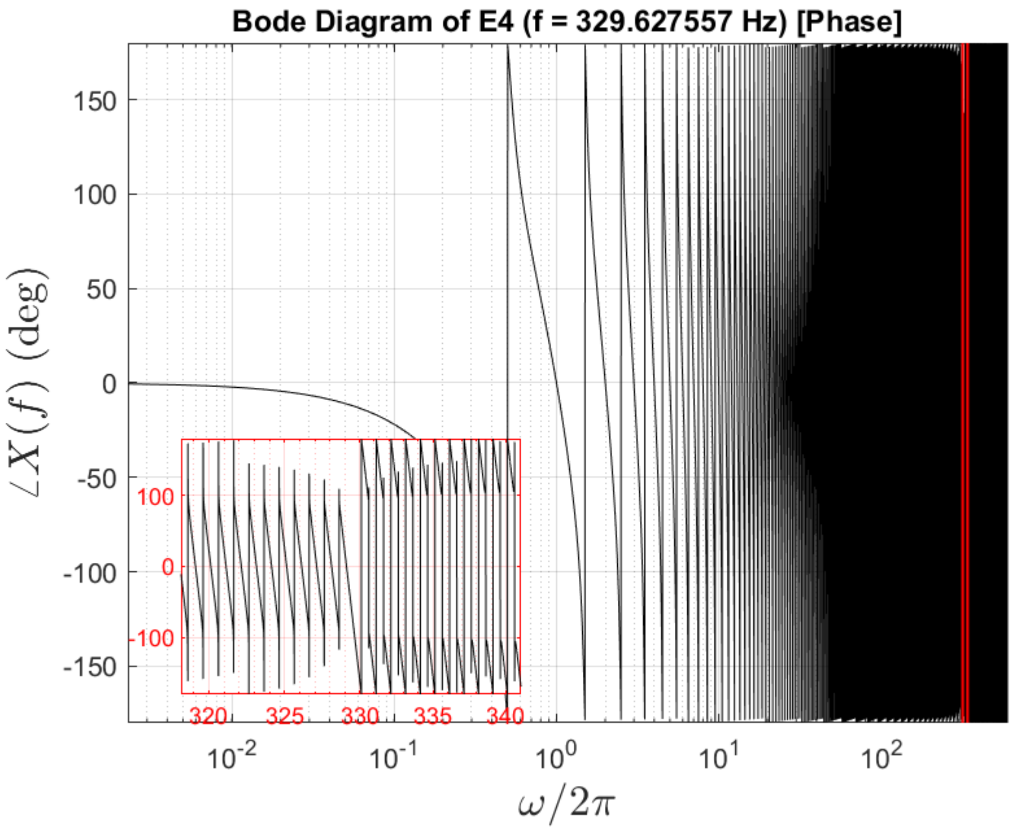
Bode Diagram of C4 (f = 261.625565 Hz) [Phase]



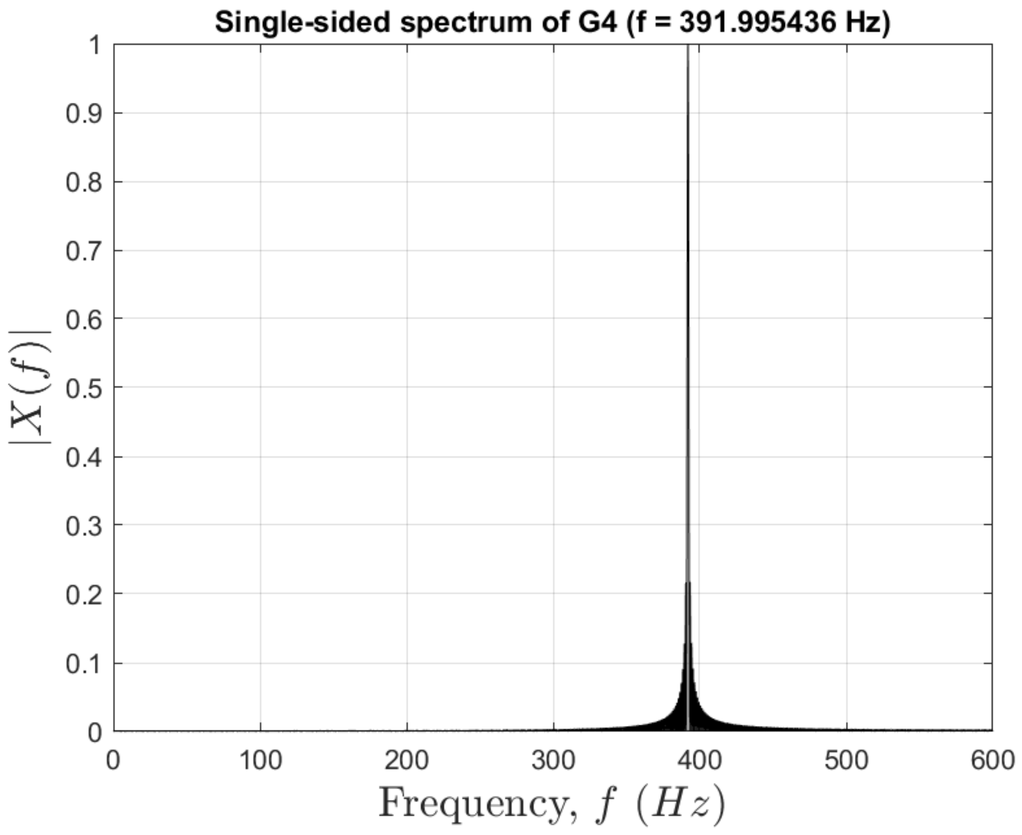
(c) Bode diagram of E4

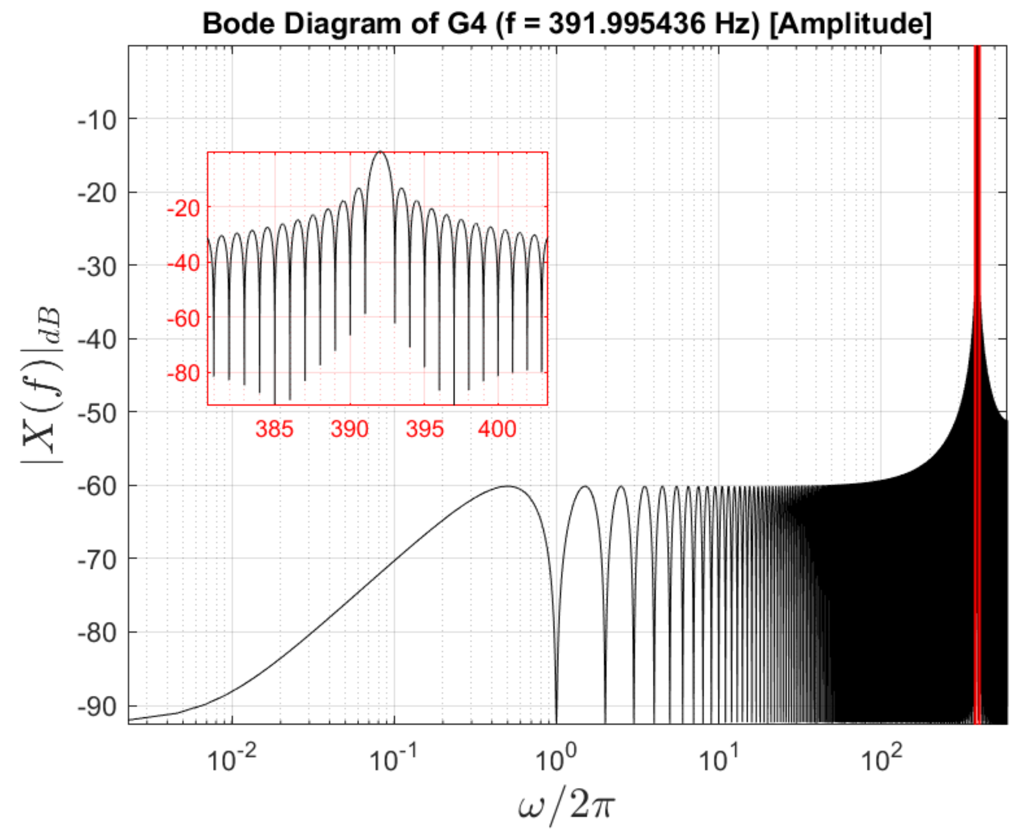


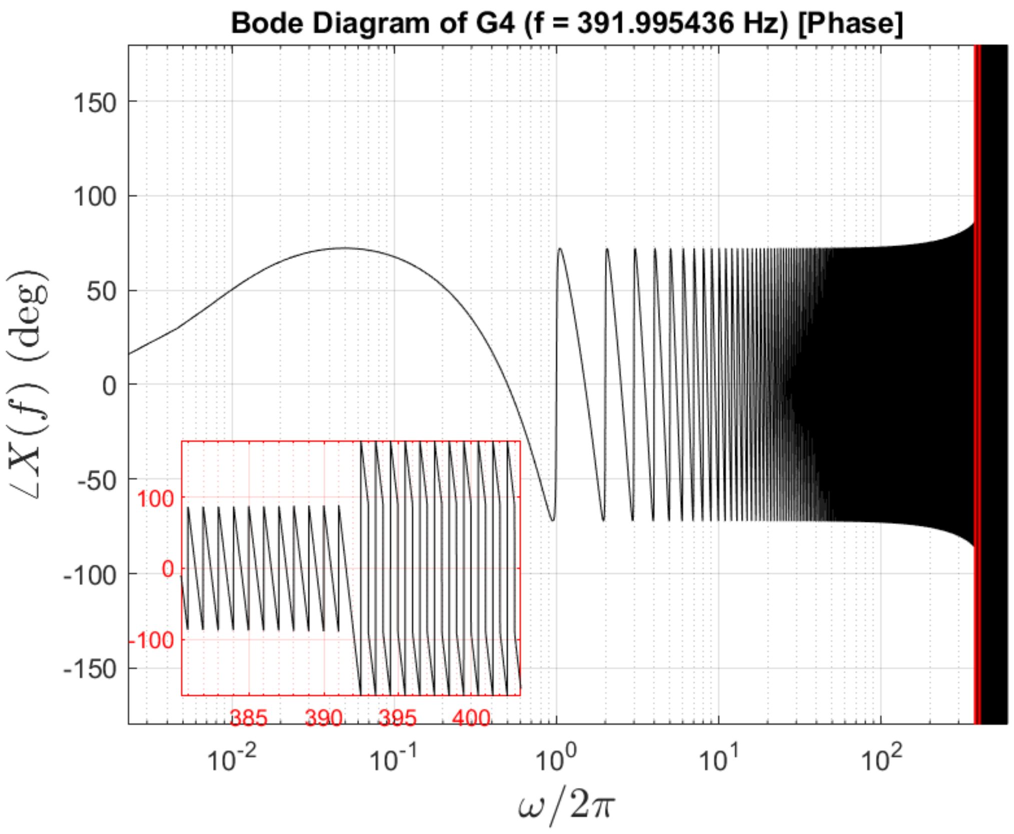




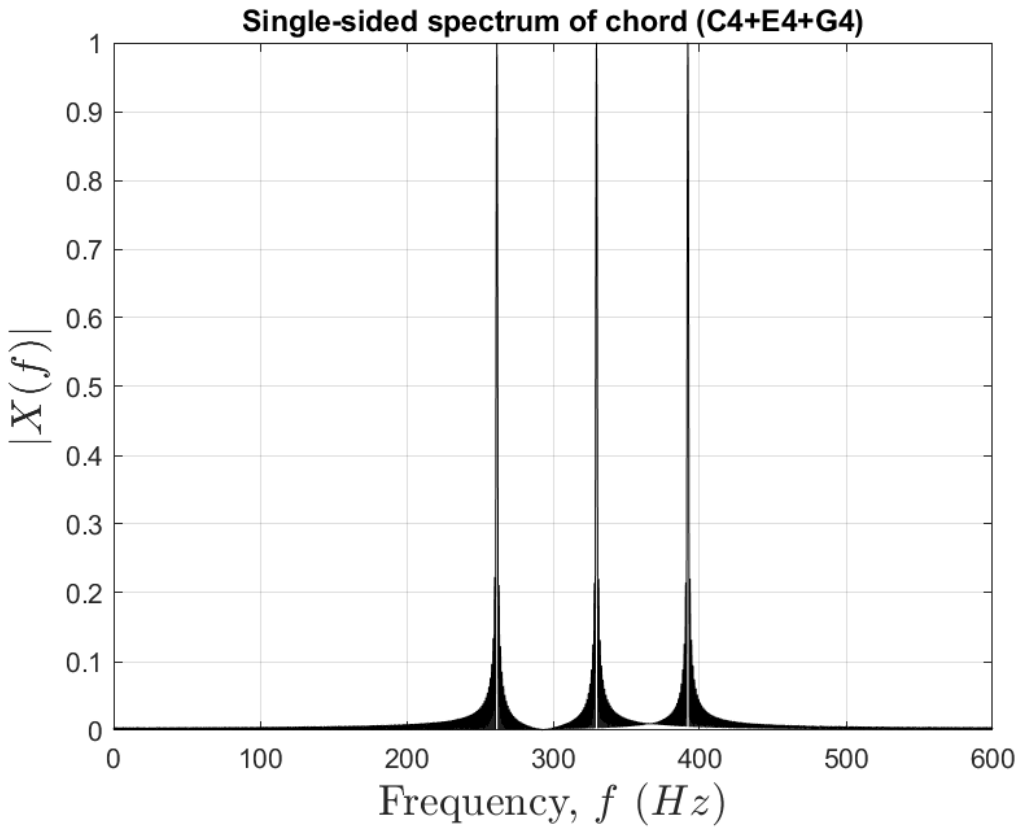
(d) Bode diagram of G4

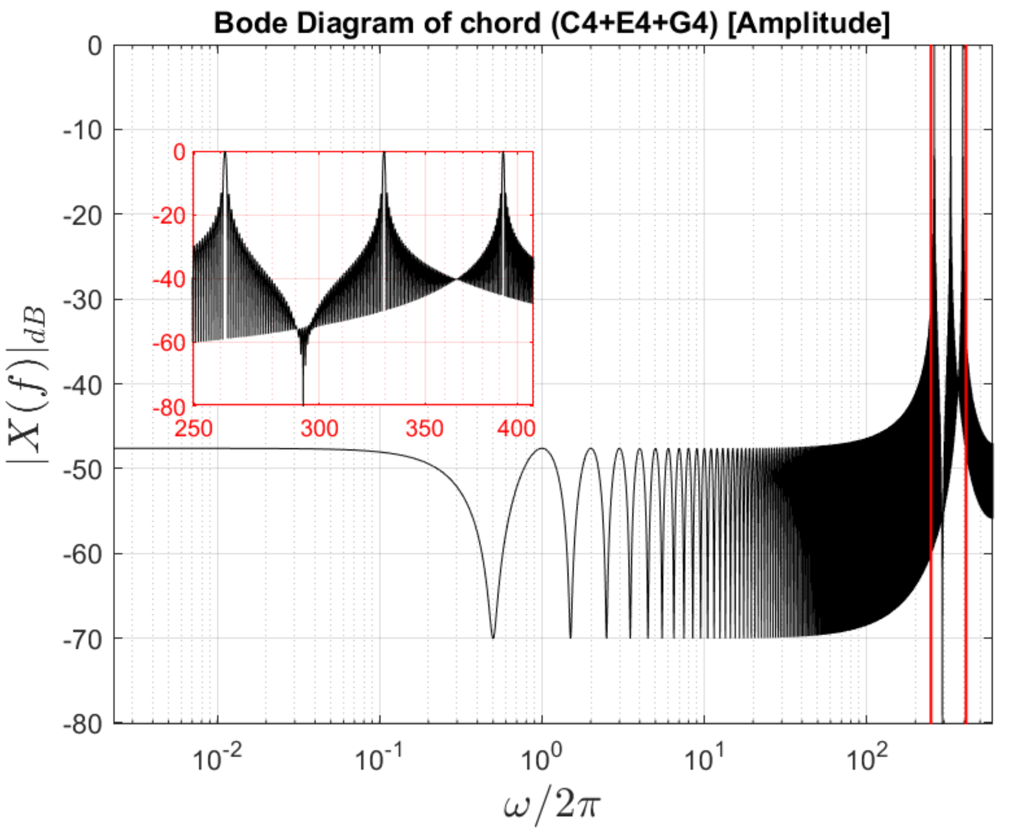


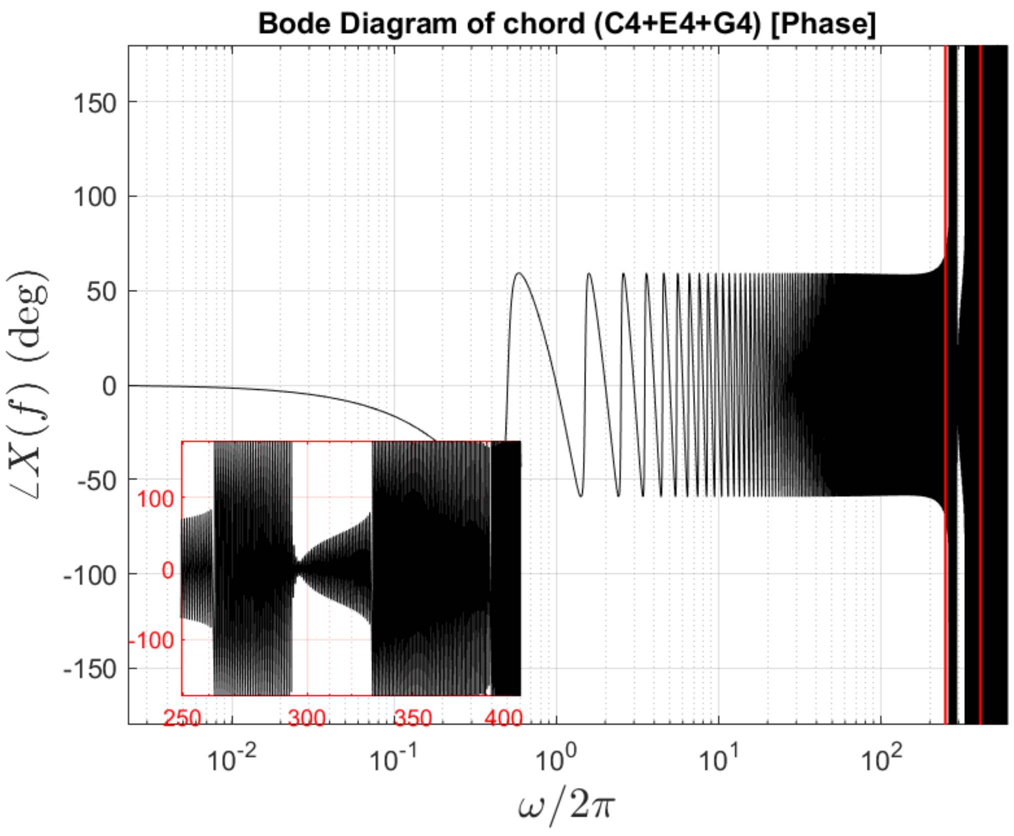

Bode Diagram of G4 (f = 391.995436 Hz) [Amplitude]
$|X(f)|_{dB}$
$\omega/2\pi$
-10
-20
-30
-40
-50
-60
-70
-80
-90
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
-20
-40
-60
-80
385
390
395
400


Bode Diagram of G4 (f = 391.995436 Hz) [Phase]
$\angle X(f)$ (deg)
$\omega/2\pi$
150
100
50
0
-50
-100
-150
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
100
0
-100
385
390
395
400

(e) Bode diagram of C4+E4+G4







Note: All the files were uploaded on GitHub

All the files in this document were uploaded on Github, and can be accessed at:

`https://github.com/recanoy/KoreaUniversity_BRI509`

If there are errors in the solution or codes kindly email,
recanoy@korea.ac.kr.