BRI509: Introduction to Brain Signal Processing
Assignment No. 1

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

April 6, 2021

1. Explain the following terms briefly.

(a) Sampling property of the impulse

The sampling property is one of the important properties of the unit impulse which follows the equivalence property. According to the equivalence property, the product of an arbitrary function $g(t)$ and the unit impulse $\delta(t - t_0)$ can be written as:

$$g(t)\delta(t - t_0) = g(t_0)\delta(t - t_0). \tag{1}$$

Hence,

$$\begin{aligned} \int_{-\infty}^{\infty} g(t)\delta(t - t_0)dt &= \int_{-\infty}^{\infty} g(t_0)\delta(t - t_0)dt, \\ &= g(t_0) \int_{-\infty}^{\infty} \delta(t - t_0)dt \\ &= g(t_0) \end{aligned} \tag{2}$$

The sampling property of the impulse, therefore, means that in an integral of this type, the impulse will sample the value of the arbitrary function $g(t)$ at time $t = t_0$.

(b) Time invariance

Time invariance is one of the system properties wherein a time-shifted input signal will also cause a time-shifted response. Mathematically, if a system is initially in its zero state and an arbitrary input signal $x(t)$ causes a response $y(t)$, then a time-shifted input signal $x(t-t_0)$ will also cause a time-shifted response $y(t - t_0)$ for any arbitrary t_0.

(c) Causality

Causality is one of the system properties wherein a system responds only during or after the time in which it is excited. It is important to take note that all physical systems are causal because they cannot look into the future and respond before they are being excited.

(d) Linearity and Superposition

Linearity is a property of homogeneous and additive systems. It is important to take note that:

* In a homogeneous system, if the input signal is multiplied by a constant (including complex constants), then the zero-state response is also multiplied by the same constant.

* On the other hand, in an additive system, if the input signal $x_1(t)$ produces a zero-state response $y_1(t)$ and $x_2(t)$ produces $y_2(t)$, then the sum of the two input signals $x_1(t)$ and $x_2(t)$ will also produce a zero-state response which is the sum of the zero-state responses $y_1(t)$ and $y_2(t)$.

Since linearity is a combination of these two properties, then a linear system has these characteristics:

$$\begin{aligned} x_1(t) &\xrightarrow{\text{produces a zero-state response}} y_1(t), \\ x_2(t) &\xrightarrow{\text{produces a zero-state response}} y_2(t), \\ (\alpha x_1(t) + \beta x_2(t)) &\xrightarrow{\text{produces a zero-state response}} (\alpha y_1(t) + \beta y_2(t)) \, . \end{aligned} \tag{3}$$

Superposition, on the other hand, is a consequence of linearity wherein when one input signal is added to another, then the overall response is one of the responses added to the other as well. This implies that in a linear system, the zero-state response to a complicated input signal can be found by breaking the input signal down into simple pieces that add up to the original signal, finding the response to each small piece, and then adding all those responses to find the overall response to the overall complicated input signal.

(e) Zero-input response vs. Zero-state response

If the input signal is $x(t)$, then the zero-input response is the solution to the differential or difference equation, which describes the dynamics of the system, when $x(t)$ is set to zero. This is also called the homogeneous solution.

Zero-output response, on the other hand, is the solution to the differential or difference equation when the system has no stored energy, i.e., when the initial conditions are set to zero.

2. Solve the following simple problems.

(a) What is the fundamental period of $g(t) = 2\cos(300\pi t)$?

Solution:

(i) Note that
$$g(t) = A\cos(2\pi f_0 t) = 2\cos(300\pi t)\,. \tag{4}$$

(ii) By equating the arguments of the sinusoidal functions in eq. (4), the fundamental frequency can be easily solved and can be written as:
$$\begin{aligned} 2\pi f_0 t &= 300\pi t, \\ f_0 &= \frac{300\pi t}{2\pi t}, \\ f_0 &= 150 \ \texttt{cycles/s}. \end{aligned} \tag{5}$$

(iii) Therefore, the fundamental period is
$$\begin{aligned} T_0 &= \frac{1}{f_0}, \\ &\boxed{T_0 = \frac{1}{150}\ \texttt{s}.} \end{aligned} \tag{6}$$

(b) Find and graph the even parts of the function $g(t)$

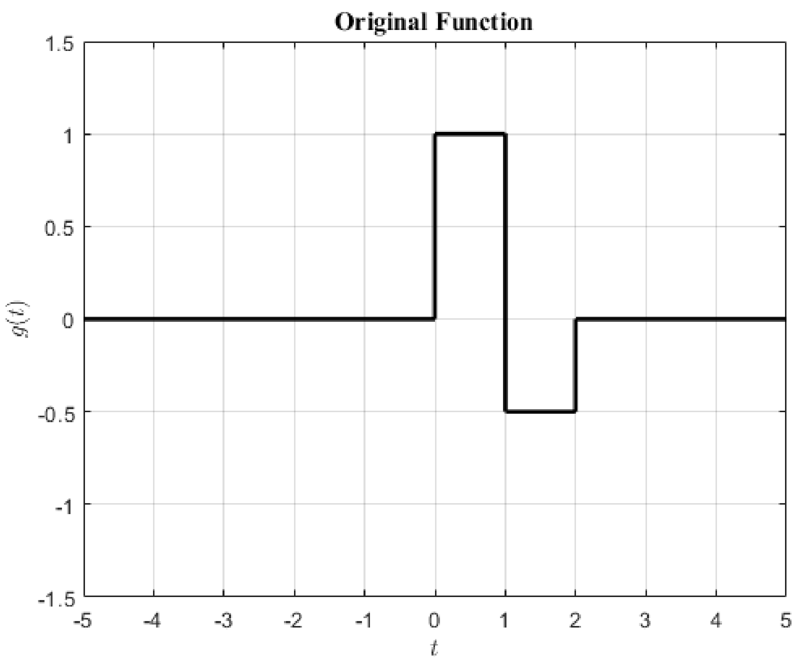


Figure 1: The original function $g(t)$.

Solution:

(i) This function can be represented mathematically as:
$$\boxed{g(t) = \begin{cases} 0, & t < 0 \\ 1, & 0 \le t \le 1 \\ -\frac{1}{2}, & 1 < t \le 2 \\ 0, & t > 2 \end{cases}}$$

(ii) The even and odd parts of the function $g(t)$ can be written as

$$g_e(t) = \frac{g(t) + g(-t)}{2}, \quad \texttt{and} \quad g_o(t) = \frac{g(t) - g(-t)}{2}, \tag{7}$$

respectively, where $g(-t)$ is the time-reversed function of $g(t)$. This time-reversed function can be represented mathematically as:

$$\boxed{g(-t) = \begin{cases} 0, & t > 0 \\ 1, & -1 \le t \le 0 \\ -\frac{1}{2}, & -2 \le t < -1 \\ 0, & t < -2 \end{cases}}$$

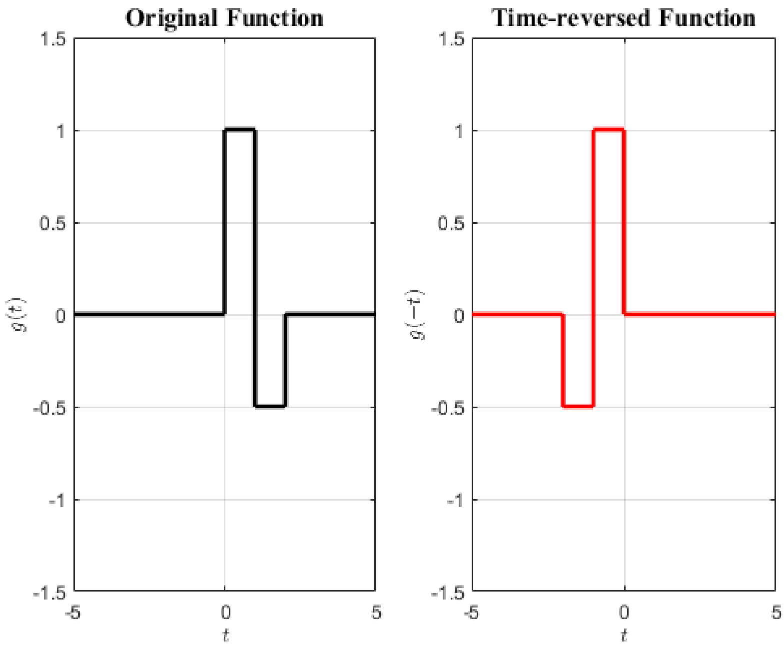


Figure 2: The graphs of the original function $g(t)$ and the time-reversed function $g(-t)$.

(iii) The even part, $g_e(t)$, of the function $g(t)$, therefore, can be calculated as

$$
\begin{aligned}
g_e(t) &= \frac{g(t)+g(-t)}{2} \\
g_e(t) &= \frac{1}{2}\begin{cases} 0, & t < -2 \\ -\frac{1}{2}, & -2 \le t < -1 \\ 1, & -1 \le t < 0 \\ 2, & t = 0 \\ 1, & 0 <\le 1 \\ -\frac{1}{2}, & 1 < t \le 2 \\ 0, & t > 2 \end{cases} \\
g_e(t) &= \begin{cases} 0, & t < -2 \\ -\frac{1}{4}, & -2 \le t < -1 \\ \frac{1}{2}, & -1 \le t < 0 \\ 1, & t = 0 \\ \frac{1}{2}, & 0 <\le 1 \\ -\frac{1}{4}, & 1 < t \le 2 \\ 0, & t > 2 \end{cases}
\end{aligned}
\tag{8}
$$

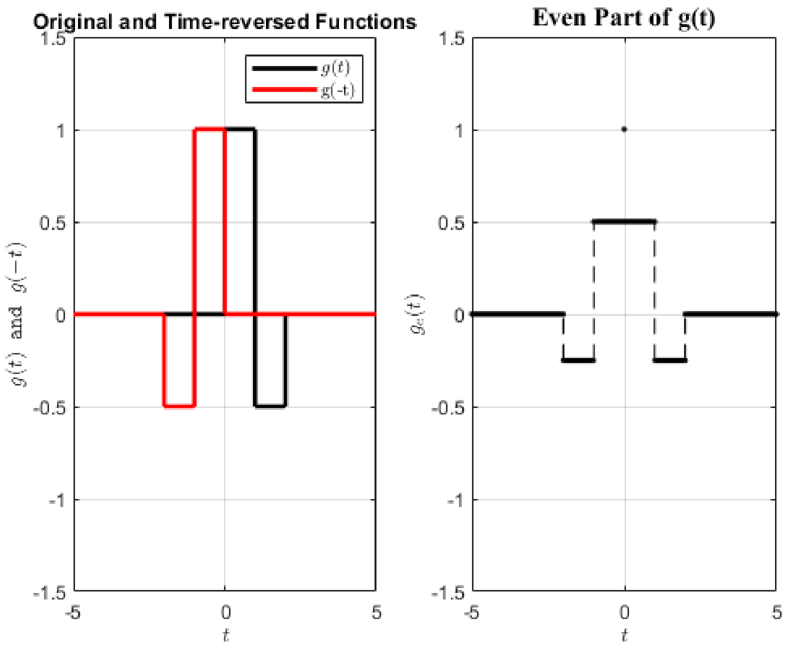


Figure 3: The graphs of the original function $g(t)$, time-reversed function $g(-t)$, and the **even part**, $g_e(t)$, of the of the function $g(t)$.

(iv) The odd part, $g_o(t)$, of the function $g(t)$ can, therefore, be calculated as:

$$g_o(t) = \frac{g(t) - g(-t)}{2}$$
$$g_o(t) = \frac{1}{2}\begin{cases} 0, & t < -2 \\ 0 - \left(-\frac{1}{2}\right), & -2 \leq t < -1 \\ 0 - 1, & -1 < t < 0 \\ 1 - 1, & t = 0 \\ 1 - 0, & 0 \leq t < 1 \\ -\frac{1}{2} - 0, & 1 < t \leq 2 \\ 0, & t > 2 \end{cases} \tag{9}$$
$$g(t) = \begin{cases} 0, & t < -2 \\ \frac{1}{4}, & -2 \leq t < -1 \\ -\frac{1}{2}, & -1 < t < 0 \\ 0, & t = 0 \\ \frac{1}{2}, & 0 \leq t < 1 \\ -\frac{1}{4}, & 1 < t \leq 2 \\ 0, & t > 2 \end{cases}$$

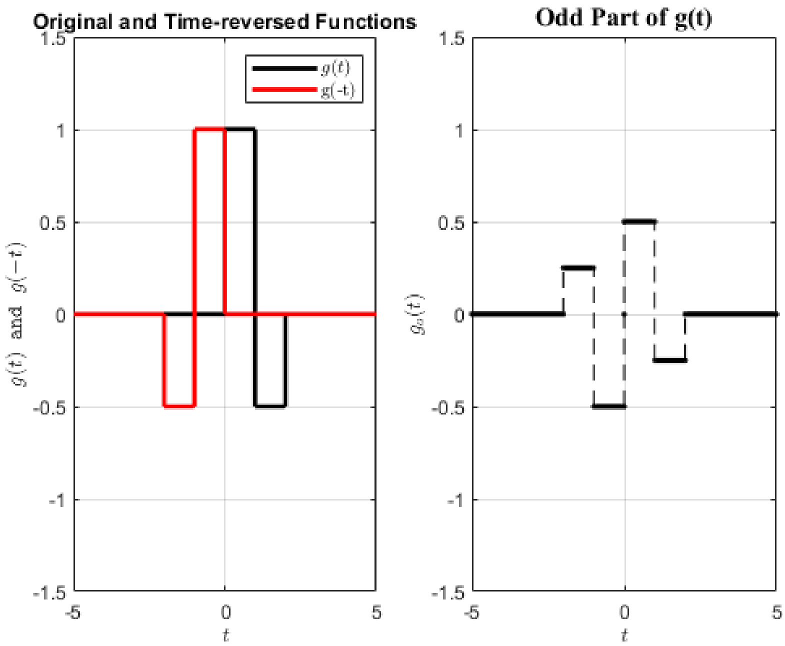


Figure 4: The graphs of the original function $g(t)$, time-reversed function $g(-t)$, and the **odd part**, $g_0(t)$, of the of the function $g(t)$.

(c) What is the numerical value of the following accumulation?

$$\sum_{n=-5}^{10} \delta_3[n] \tag{10}$$

Solution:

(i) Note that impulse train is represented as $\delta_N = \sum_{m=-\infty}^{\infty} \delta[n - mN]$. If $N = 3$, the impulse train becomes $\boxed{\delta_3[n] = \sum_{m=-\infty}^{\infty} \delta[n - m(3)]}$. In the domain $-5 \leq n \leq 10$, this impulse train can be plotted as:

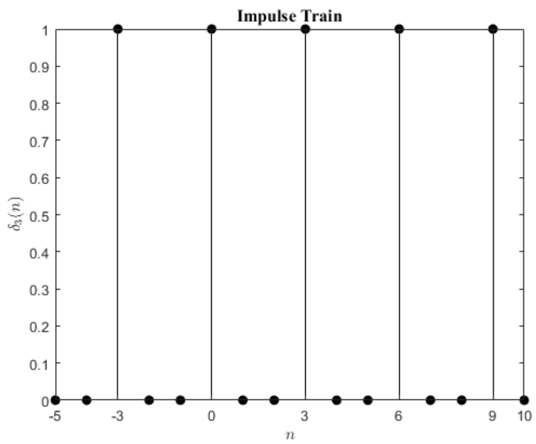


Figure 5: The plot of the impulse train $\delta_3[n] = \sum_{m=-\infty}^{\infty} \delta[n - m(3)]$.

(ii) The numerical value of the accumulation is shown in Fig. (6). It should be noted that it is assumed that the accumulation before $n = -5$ is zero.

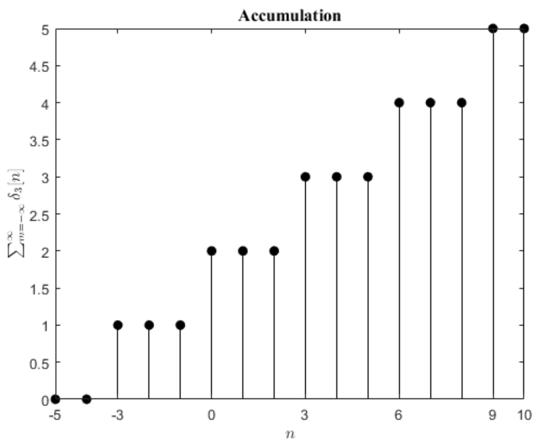


Figure 6: The accumulation of the impulse train $\delta_3[n]$

(d) Find the average signal power of the periodic signal $x(t)$.

$$x(t) = A\cos(2\pi f t) \tag{11}$$

Solution:

$$\begin{aligned}
P_x(t) &= \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} |x(t)|^2 dt \\
&= \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} |A\cos(2\pi f_0 t)\,|^2 dt \\
&= \frac{2|A|^2}{T_0}\int_{0}^{T_0/2} \cos^2(2\pi f_0 t)\, dt
\end{aligned} \tag{12}$$

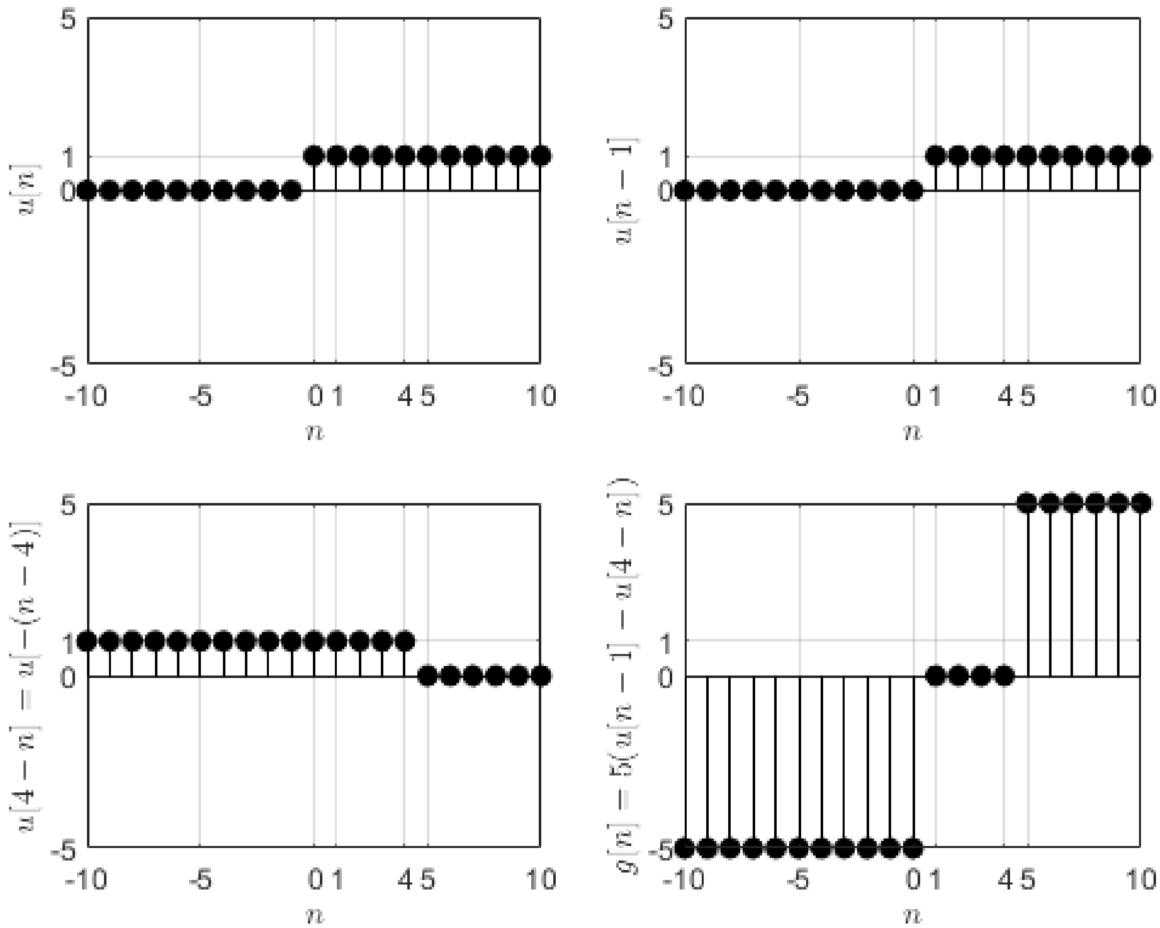
Note that $\cos(2A) = 2\cos^2(A) - 1$. Therefore, $\cos^2(A) = \frac{1}{2}(\cos(2A)+1)$.

$$\begin{aligned}
P_x(t) &= \frac{2|A|^2}{T_0}\int_{0}^{T_0/2} \frac{1}{2}(\cos(4\pi f_0 t) + 1)\, dt \\
&= \frac{|A|^2}{T_0}\left[\int_{0}^{T_0/2} \cos(4\pi f_0 t) dt + \int_{0}^{T_0/2} dt\right] \\
&= \frac{|A|^2}{T_0}\left[\frac{1}{4\pi f_0}\left(\sin(4\pi f_0 t)\Big|_{0}^{T_0/2} + \frac{T_0}{2}\right]\right. \\
&= \frac{|A|^2}{T_0}\left[\frac{1}{4\pi f_0}(\sin(2\pi) - \sin(0)) + \frac{T_0}{2}\right] \\
&= \frac{|A|^2}{T_0}\left(\frac{T_0}{2}\right) \\
&\boxed{P_x = \frac{|A|^2}{2}}
\end{aligned} \tag{13}$$

(e) Graph the following function:

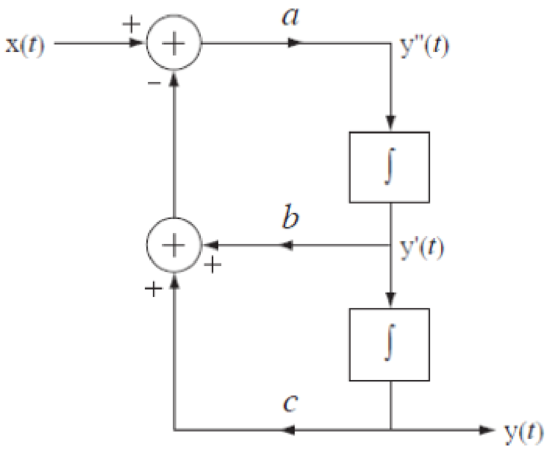
$$g[n] = 5u[n-1] - u[4-n] \tag{14}$$

Solution:



3. **Solve the following problems**

(a) **Find the zero-input response, the response when $x(t) = 0$, of the system in the figure if the initial value of $y(t)$ is $y(0) = 1$, and the initial rate of change of $y(t)$ is $y^{'}(t = 0) = 0$, $a = 1$, $b = 0$, and $c = 4$.**



Solution:

(i) The differential equation which represents this system can be written as:

$$\begin{aligned} y^{''}(t) &= a\left[x(t) - \left(by^{'}(t) + cy(t)\right)\right] \\ y^{''}(t) &= (1)\left[x(t) - \left((0)y^{'}(t) + (4)y(t)\right)\right] \\ y^{''}(t) &= x(t) - 4y(t) \\ \boxed{y^{''}(t) + 4y(t) = x(t)} \end{aligned} \tag{15}$$

(ii.) **Zero-input response (Homogeneous solution):**

(1.) Let $y_h(t) = Ke^{\lambda t}$.

(2.) Substituting this initial guess into the differential equation will give the eigenvalues:

$$\begin{aligned} y_h^{''}(t) + 4y_h(t) &= 0 \\ \lambda^2 Ke^{\lambda t} + 4Ke^{\lambda t} &= 0 \\ Ke^{\lambda t}(\lambda^2 + 4) &= 0 \\ \lambda^2 &= -4 \\ \lambda_{1,2} &= \pm i2 \end{aligned} \tag{16}$$

(3.) The homogeneous solution can now be written as

$$y_h(t) = K_1 e^{i2t} + K_2 e^{-i2t} \tag{17}$$

(4.) The constants K_1 and K_2 can be found by applying the initial conditions.

Initial condition 1: At $t = 0$, $y_h(0) = 1$.

$$\begin{aligned} y_h(0) &= K_1 e^0 + K_2 e^0 \\ 1 &= K_1 + K_2 \end{aligned} \tag{18}$$

Initial condition 2: At $t = 0$, $y_h'(0) = 0$.

$$\begin{aligned} y_h'(0) &= i2K_1e^0 - i2K_2e^0 \\ 0 &= i2K_1 - i2K_2 \end{aligned} \tag{19}$$

Writing this system of linear equations in matrix form:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ i2 & -i2 \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \tag{20}$$

Therefore,

$$\begin{aligned} \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ i2 & -i2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} K_1 \\ k_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \end{aligned} \tag{21}$$

(5.) Therefore, the zero-input response (or the homogeneous solution is)

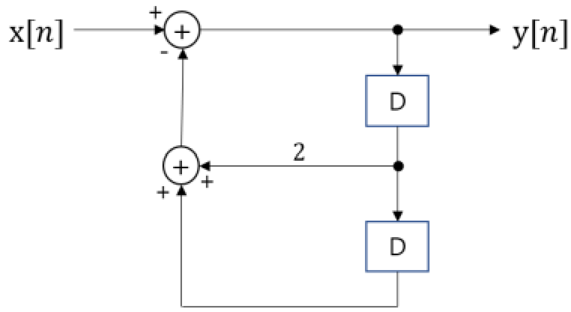
$$\begin{aligned} y_h(t) &= \frac{1}{2}e^{i2t} + \frac{1}{2}e^{-i2t} \\ y_h(t) &= \frac{e^{i2t} + e^{-i2t}}{2} \\ &\boxed{y_h(t) = \cos(2t)} \end{aligned} \tag{22}$$

If the system is in its zero state before $t = 0$, then the zero-input response can be written as:

$$\begin{aligned} y_h(t) &= \cos(2t), \quad t \geq 0 \\ y_h(t) &= \cos(2t)\, u(t), \end{aligned} \tag{23}$$

where $u(t)$ is the unit-step function.

(b) Find the response of the system in the figure if $x[n] = u[n]$ and the system is in its zero state before time $n = 0$.



Solution:

(i) The difference equation which describes the system can be written as:

$$\begin{aligned} y[n] &= x[n] - (2y[n-1] + y[n-2]) \\ y[n] + 2y[n-1] + y[n-2] &= x[n] \end{aligned} \tag{24}$$

(ii) **Homogeneous Solution:**

(1.) Let $y_h[n] = Kz^n$.

(2.) Substituting this guess into the homogeneous difference equation will give the eigenvalues.

$$\begin{aligned} y_h[n] + 2y_h[n-1] + y_h[n-2] &= 0 \\ Kz^n + 2Kz^{n-1} + Kz^{n-2} &= 0 \\ Kz^{n-2}\left(z^2 + 2z + 1\right) &= 0 \\ z^2 + 2z + 1 = 0 \end{aligned} \tag{25}$$

The eigenvalues are

$$\begin{aligned} z &= \frac{-2 \pm \sqrt{2^2 - 4(1)(1)}}{2(1)} \\ z &= -1 \end{aligned} \tag{26}$$

(3.) Therefore, the homogeneous solution is given by

$$\boxed{y_h[n] = K_h(-1)^n} \tag{27}$$

(iii) **Particular Solution:**

(1.) The input signal is $x[n] = u[n]$, which is a constant; hence, all its differences are zero. The particular solution is, therefore, also a constant

$$y_p[n] = K_p. \tag{28}$$

(2.) Substituting this into the difference equation will give

$$\begin{aligned} K_p + 2K_p + K_p &= 1 \\ 4K_p &= 1 \\ K_p &= \frac{1}{4} \end{aligned} \tag{29}$$

(3.) The particular solution is given by

$$\boxed{y_p[h] = \frac{1}{4}} \tag{30}$$

(iv.) **Complete Solution:**

$$\begin{aligned} y[n] &= y_h[n] + y_p[n] \\ y[n] &= K_h(-1)^n + \frac{1}{4} \end{aligned} \tag{31}$$

(v.) The system is in its zero state before $n = 0$; hence, $y[-1] = 0$ and $y[-2] = 0$. At $n = 0$, therefore, $y[0] = 1$. Applying the initial condition $y[0] = 1$ gives

$$\begin{aligned} y[0] &= K_h(-1)^0 + \frac{1}{4} \\ 1 &= K_h + \frac{1}{4} \\ K_h &= \frac{3}{4} \end{aligned} \tag{32}$$

(vi.) Therefore, the response of the system is

$$\begin{aligned} y[n] &= \begin{cases} \frac{3}{4}(-1)^n + \frac{1}{4}, & n \geq 0 \\ 0, & n < 0 \end{cases} \\ &\boxed{y[n] = \left(\frac{3}{4}(-1)^n + \frac{1}{4}\right) u[n]} \end{aligned} \tag{33}$$

4. MATLAB Coding

(a) Graph the function combinations with MATLAB.

$$\begin{aligned} x_1(t) &= e^{-t}\sin(20\pi t) + e^{-t/2}\sin(19\pi t) \\ x_2(t) &= \texttt{rect}(t)\cos(20\pi t) \end{aligned} \quad (34)$$

Source Code

```
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

%% Part 1: x_{1}(t) = e^(-t)*sin(20*pi*t) + e^(-t/2)*sin(19*pi*t)
fs = 500;
t1 = [-2: 1/fs : 6];
x1 = exp(-t1).*sin(20*pi*t1) + exp(-t1/2).*sin(19*pi*t1);

figure;
subplot(1, 2, 1)
p = plot(t1, x1, 'k');
set(p, 'LineWidth', 1)
xlabel('$t$', 'Interpreter', 'latex');
ylabel('$x_{1}(t)$', 'Interpreter', 'latex');
grid on
title('x_1(t) = exp(-t)sin(20\pit) + exp(-t/2)sin(19\pit)', 'FontName', 'Times', 'FontSize', 12)

%% Part 2: x_{2}(t) = rect(t)*cos(20*pi*t)
fs = 500;
t2 = [-2: 1/fs : 6];
x2 = prob4a_rect(t2).*cos(20*pi*t2);

subplot(1, 2, 2)
p = plot(t2, x2, 'k');
set(p, 'LineWidth', 1)
xlabel('$t$', 'Interpreter', 'latex');
ylabel('$x_{2}(t)$', 'Interpreter', 'latex');
grid on
title('x_2(t) = rect(t)cos(20\pit)', 'FontName', 'Times', 'FontSize', 12)
```

Unit-rectangle Function

```
function y = prob4a_rect(t)
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376
%
% Description: Unit rectangle function
% Equation: rect(t) = us(t + 0.5) - us(t - 0.5);

y = us(t + 0.5) - us(t - 0.5);
end

function y = us(t)
% Description: Unit-step Function
% us(t) = (sgn(t) + 1)/2 = (0).*(t < 0) + (1).*(t >= 0),
    where
% sgn(t) = -1.*(t<0) + 0.*(t = 0) + 1.*(t >= 0)
y = (sign(t) + 1)/2;
end
```

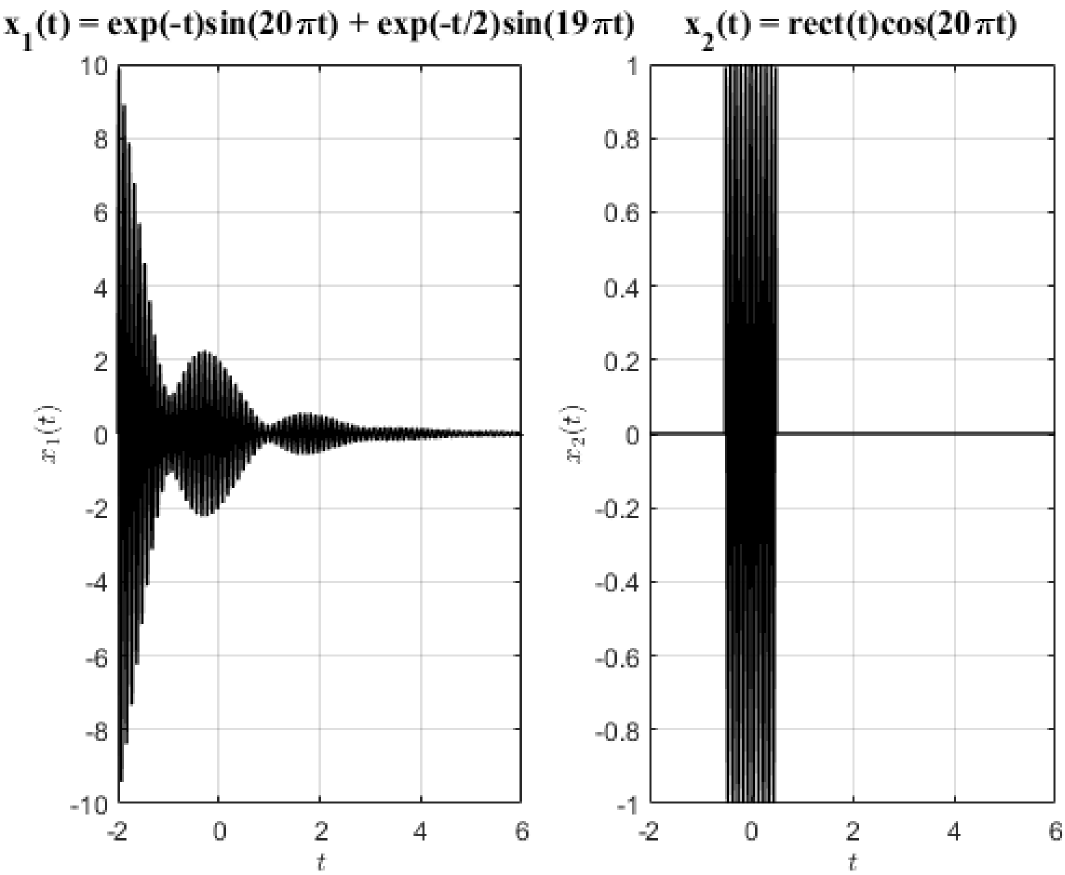


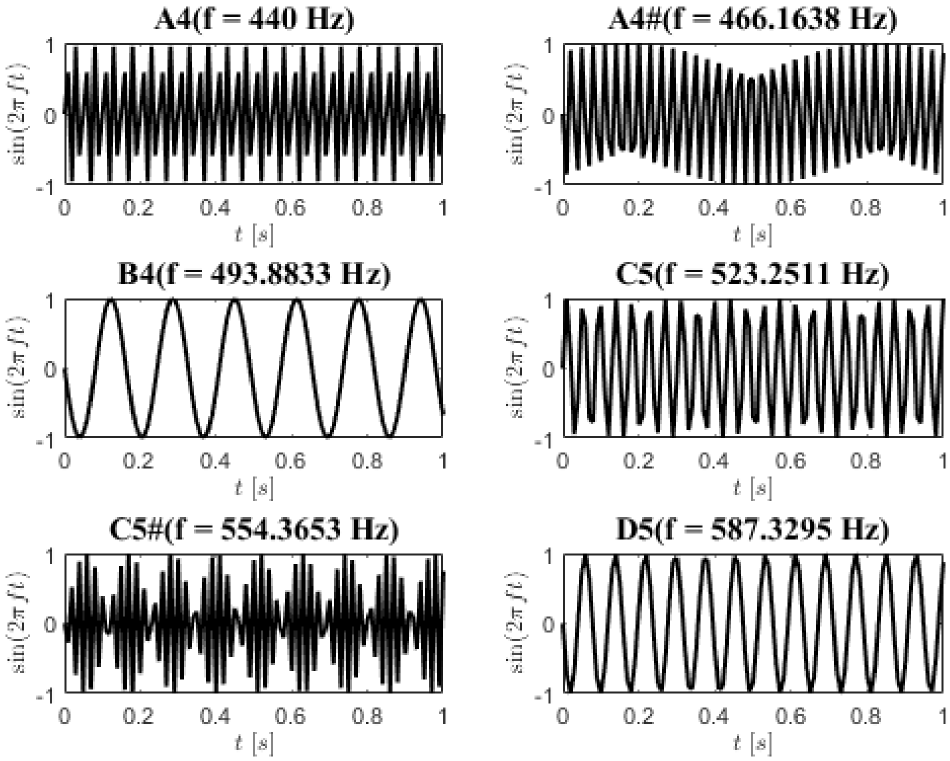
Figure 7: The graphs of $x_1(t) = e^{-t}\sin(20\pi t) + e^{-t/2}\sin(19\pi t)$ and $x_2(t) = \texttt{rect}(t)\cos(20\pi t)$.

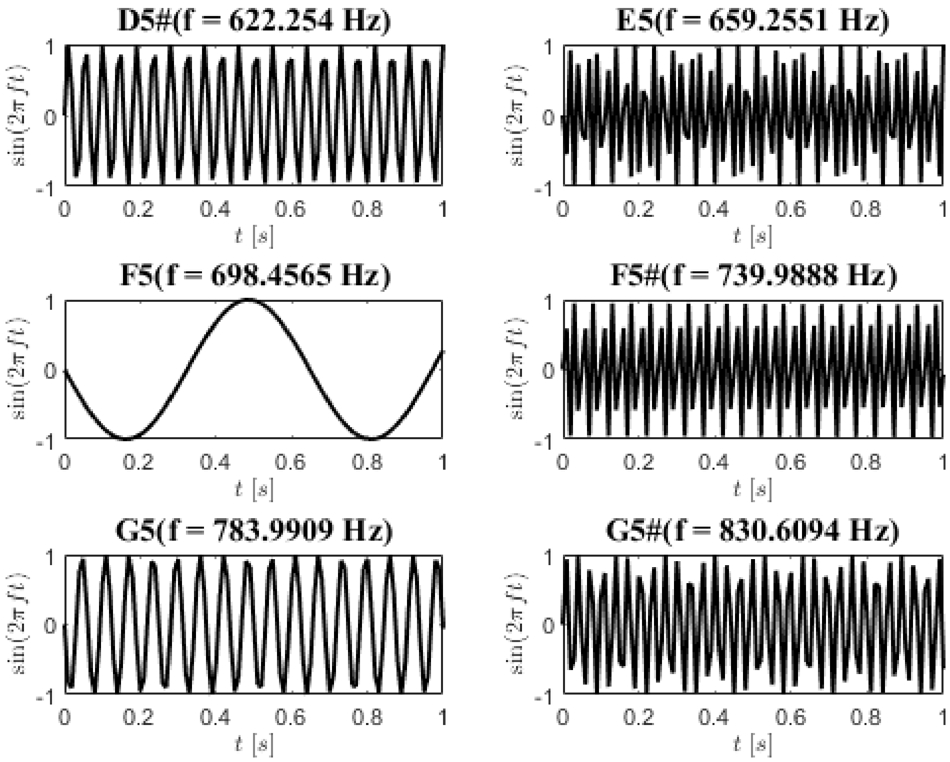
(b) Make sinusoid whose fundamental frequencies are the normal scale, 440 Hz (A4), 466.1 Hz (A4#), 493.8 Hz (B4), 523.22 Hz (C5), 554.36 Hz (C5), 587.33 Hz (D5), 622.25 Hz (D5#), 659.26 Hz (E5), 698.46 Hz (F5), 739.99 Hz (F5#), 784.00 Hz (G5), 830.60Hz (G5#), 880 Hz (A5).

Source Code

```
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% 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

%% Part 1: Visualization
clear; clc; clf;

fs = 100;                         % sampling rate (small value for clear visualization)
dur   = 1;                        % duration, s
t = [0 : 1/fs : dur];             % One-second duration

% Sinusoid Signal
% y(t) = sin(2*pi*f*t), where f = 440*beta^(n - 1), beta = 2^(1/12), and
% n is the index, i.e., n = [1, 13]
beta = 2^(1/12);
note = @(index) sin(2*pi*(440*(beta^(index - 1)))*t); % a sinusoid function

% musical_scale_keys: Names of the notes
musical_scale_keys = {'A4', 'A4#', 'B4', 'C5', 'C5#', 'D5', 'D5#', 'E5', 'F5', 'F5#', 'G5', 'G5#', 'A5'};
% musical_scale_values: An empty 13x1 cell, and will hold the notes' values
musical_scale_values = cell(length(musical_scale_keys), 1);

% Assigns the values of each note to musical_scale_values
for ind = 1 : length(musical_scale_keys)
```

```
    musical_scale_values{ind} = note(ind);
end

% Dictionary-type container -- keys: notes's names; values: notes' values
music_octave = containers.Map(musical_scale_keys, musical_scale_values);

% Plotting is done is done depending on the number of elemets in
% musical_scale_keys
set_NUM = floor(length(musical_scale_keys)/(3*2)); % (3x2) plots
for set_num = 1 : set_NUM
    disp(['set = ' num2str(set_num)])
    figure;
    for ind = (floor(length(musical_scale_keys)/2))*(set_num - 1) + 1 : set_num*floor(length(musical_scale_keys)/2)
        disp(['ind = ' num2str(ind)])
        subplot(floor(length(musical_scale_keys)/4), 2, ind - (set_num - 1)*floor(length(musical_scale_keys)/2))
        p = plot(t, music_octave(musical_scale_keys{ind}), 'k');
        set(p, 'LineWidth', 1.5);
        yticks([-1, 0, 1]);
        xlabel('$t \; [s]$', 'Interpreter', 'latex');
        ylabel('$\sin (2 \pi f t)$', 'Interpreter', 'latex');
        title([musical_scale_keys{ind}, '(f = ', num2str(440*(beta^(ind - 1))), ' Hz)'], 'FontName', 'Times', 'FontSize', 12);
    end
end

figure,
p = plot(t, music_octave(musical_scale_keys{13}), 'k');
set(p, 'LineWidth', 1.5);
yticks([-1, 0, 1]);
xlabel('$t \; [s]$', 'Interpreter', 'latex');
ylabel('$\sin (2 \pi f t)$', 'Interpreter', 'latex');
title([musical_scale_keys{13}, '(f = ', num2str(440*(beta^(13 - 1))), ' Hz)'], 'FontName', 'Times', 'FontSize', 12);
```


A4(f = 440 Hz)
A4#(f = 466.1638 Hz)
B4(f = 493.8833 Hz)
C5(f = 523.2511 Hz)
C5#(f = 554.3653 Hz)
D5(f = 587.3295 Hz)
$\sin(2\pi f t)$
t [s]


D5#(f = 622.254 Hz)
E5(f = 659.2551 Hz)
F5(f = 698.4565 Hz)
F5#(f = 739.9888 Hz)
G5(f = 783.9909 Hz)
G5#(f = 830.6094 Hz)
$\sin(2\pi f t)$
t [s]

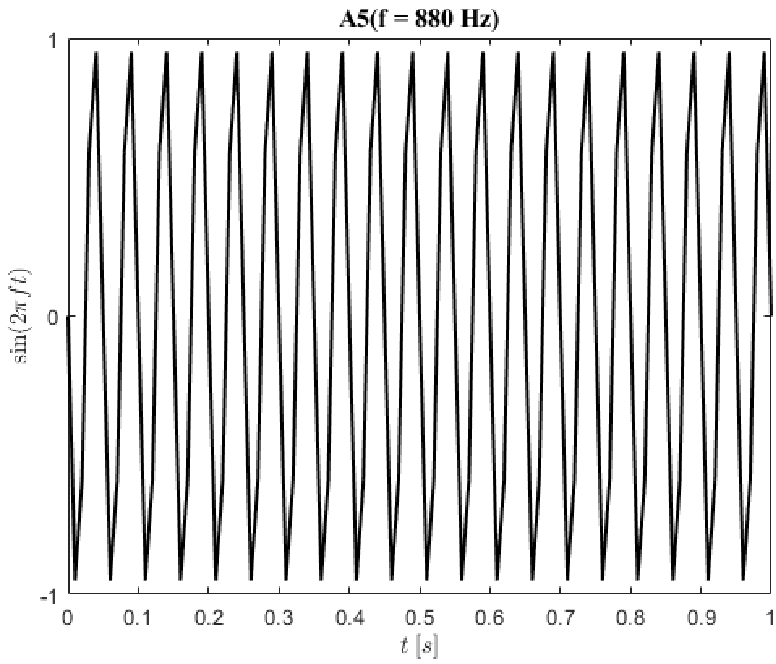


Figure 8: Graphs of the note scales from `A4` to `A5`.

Part 2: Make an MP3 file containing the normal scale `A4` to `A5` one second each.

> **Note:**
>
> Instead of saving the file as a *.mp3 file, I have decided to just save it as a *.mp4 file because there is an `audiowrite()` function in MATLAB which can readily do this. However, *.mp3 is not supported by the `audiowrite()`.

Source Code

```matlab
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

%% Part II: MP3 file containing scale A4~A5 one second each
clear; clc;

fs = 48000;                    % Accepted framerate for the audiowrite function
dur  = 1;                      % duration, s
t = [0 : 1/fs : dur];          % One-second duration

% Sinusoid Signal
% y(t) = sin(2*pi*f*t), where f = 440*beta^(n - 1), beta = 2^(1/12), and
% n is the index, i.e., n = [1, 13]
beta = 2^(1/12);
note = @(index) sin(2*pi*(440*(beta^(index - 1)))*t); % a sinusoid function

% musical_scale_keys: Names of the notes
musical_scale_keys = {'A4', 'A4#', 'B4', 'C5', 'C5#', 'D5', 'D5#', 'E5', 'F5', 'F5#', 'G5', 'G5#', 'A5'};
% musical_scale_values: An empty 13x1 cell, and will hold the notes' values
musical_scale_values = cell(length(musical_scale_keys), 1);

% Assigns the values of each note to musical_scale_values
for ind = 1 : length(musical_scale_keys)
    musical_scale_values{ind} = note(ind);
end

% Dictionary-type container -- keys: notes's names; values: notes' values
music_octave = containers.Map(musical_scale_keys, musical_scale_values);

mp4_file = [];
filename = 'Assignment1_2020021376_CanoyRaymartJay_prob4b_part2.mp4';
for ind = 1 : length(musical_scale_keys)
    mp4_file = [mp4_file; [music_octave(musical_scale_keys{ind}) zeros(1, 100)]'];
end

% Just for listening the sound
soundsc(mp4_file, fs)

% Saving the sound as an mp4 file
audiowrite(filename, mp4_file, fs)
```

Part 3: Make an MP3 file containing the /`do`/, /`re`/, /`mi`/, ..., /`ti`/, /`do`/.

Note:

Instead of saving the file as a *.mp3 file, I have decided to just save it as a *.mp4 file because there is an `audiowrite`() function in MATLAB which can readily do this. However, *.mp3 is not supported by the `audiowrite`().

Source Code: **A Major**

```
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

%% Part 3: A Major
clear; clc;

fs = 48000;                      % Accepted framerate for the
    audiowrite function
dur  = 1;                        % duration, s
t = [0 : 1/fs : dur];            % One-second duration

% Sinusoid Signal
% y(t) = sin(2*pi*f*t), where f = 440*beta^(n - 1), beta =
     2^(1/12), and
% n is the index, i.e., n = [1, 13]
beta = 2^(1/12);
note = @(index) sin(2*pi*(440*(beta^(index - 1)))*t); % a
    sinusoid function

% musical_scale_keys: Names of the notes
musical_scale_keys = {'A4', 'A4#', 'B4', 'C5', 'C5#', 'D5'
    , 'D5#', 'E5', 'F5', 'F5#', 'G5', 'G5#', 'A5'};
% musical_scale_values: An empty 13x1 cell, and will hold
    the notes' values
musical_scale_values = cell(length(musical_scale_keys), 1)
    ;

% Assigns the values of each note to musical_scale_values
for ind = 1 : length(musical_scale_keys)
    musical_scale_values{ind} = note(ind);
end

% Dictionary-type container -- keys: notes's names; values
    : notes' values
music_octave = containers.Map(musical_scale_keys,
    musical_scale_values);

```

```
% A-Major
solfege_A_Major = {'A4', 'B4', 'C5#', 'D5', 'E5', 'F5#', '
    G5#', 'A5'};

for ind = 1 : length(solfege_A_Major)
    index = strcmp(solfege_A_Major(ind), musical_scale_keys
    );
    solfege_index(ind) = find(index);
end

solfege_a_major = [];
for k1 = 1:length(solfege_index)
     solfege_a_major = [solfege_a_major; [note(
    solfege_index(k1))  zeros(1,100)]'];
end

soundsc(solfege_a_major, fs);
filename_a_major = '
    Assignment1_2020021376_CanoyRaymartJay_prob4b_Part3_AMajor
    .mp4';
audiowrite(filename_a_major, solfege_a_major, fs);
```

Source Code: A Minor

```
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 1
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

%% Part 3: A Minor
clear; clc;

fs = 48000;                      % Accepted framerate for the
    audiowrite function
dur  = 1;                        % duration, s
t = [0 : 1/fs : dur];            % One-second duration

% Sinusoid Signal
% y(t) = sin(2*pi*f*t), where f = 440*beta^(n - 1), beta =
     2^(1/12), and
% n is the index, i.e., n = [1, 13]
beta = 2^(1/12);
note = @(index) sin(2*pi*(440*(beta^(index - 1)))*t); % a
    sinusoid function

% musical_scale_keys: Names of the notes
musical_scale_keys = {'A4', 'A4#', 'B4', 'C5', 'C5#', 'D5'
    , 'D5#', 'E5', 'F5', 'F5#', 'G5', 'G5#', 'A5'};
% musical_scale_values: An empty 13x1 cell, and will hold
    the notes' values
musical_scale_values = cell(length(musical_scale_keys), 1)
    ;

% Assigns the values of each note to musical_scale_values
for ind = 1 : length(musical_scale_keys)
     musical_scale_values{ind} = note(ind);
```

```
end

% Dictionary-type container -- keys: notes's names; values: notes' values
music_octave = containers.Map(musical_scale_keys, musical_scale_values);

% A-Minor
solfege_A_Minor = {'A4', 'B4', 'C5', 'D5', 'E5', 'F5', 'G5', 'A5'};

for ind = 1 : length(solfege_A_Minor)
   index = strcmp(solfege_A_Minor(ind), musical_scale_keys);
   solfege_index(ind) = find(index);
end

solfege_a_minor = [];
for k1 = 1:length(solfege_index)
    solfege_a_minor = [solfege_a_minor; [note(solfege_index(k1))  zeros(1,100)]'];
end

soundsc(solfege_a_minor, fs)
filename_a_minor = 'Assignment1_2020021376_CanoyRaymartJay_prob4b_Part3_AMinor.mp4';
audiowrite(filename_a_minor, solfege_a_minor, fs);
```

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/rjcanoy03/BRI509/tree/Assignment%231

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