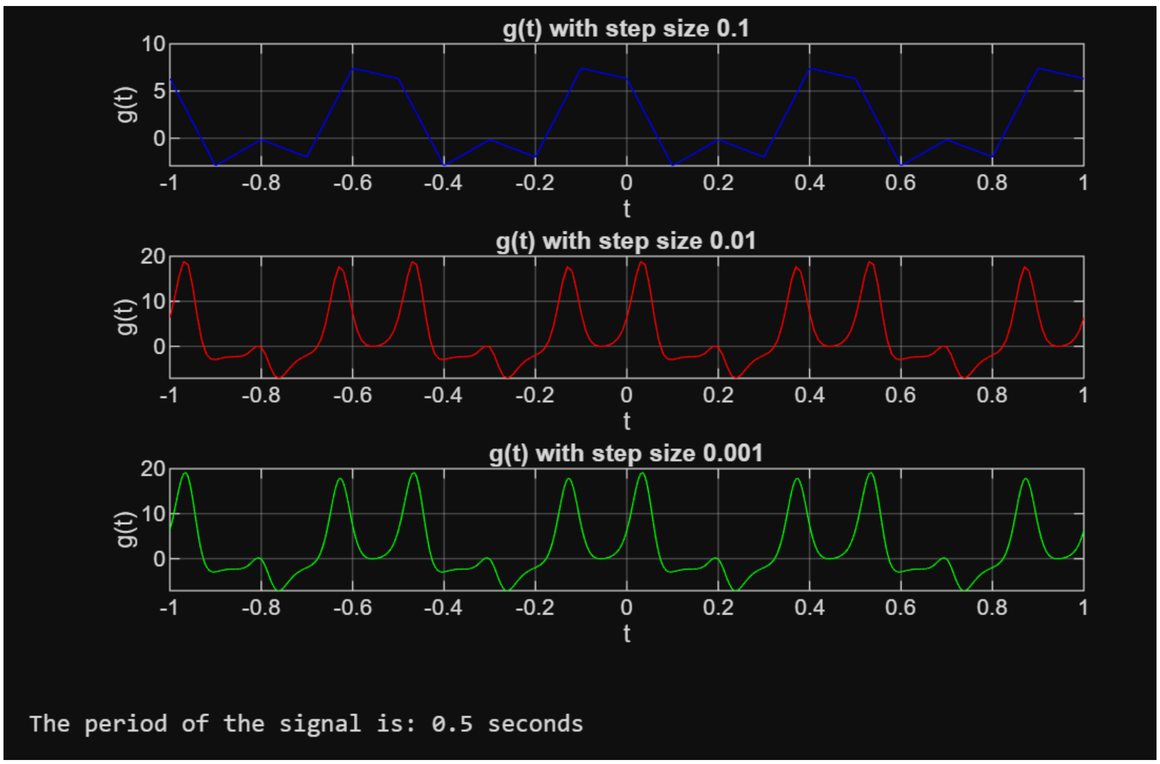
Jason Waseq
ECE103L
Visualizing signal in Matlab
Assignment 2

1. For the following function:
$g(t) = 3\pi \sin(8\pi t + 1.3)\cos(4\pi t - 0.8)e^{\sin(12\pi t)}$
create an m-file that plots the function within the window $t \in [-1,1]$ in a 3-by-1 subplot with steps of t equal to 0.1, 0.01, and 0.001. What is the period of this signal?

Final Answer:



- The period of the signal is 0.5 seconds.

Detailed Solution:
1. Function Definition: The function g(t) is defined using an anonymous function in MATLAB.
2. Time Vectors: Three time vectors are created for the specified step sizes.
3. Subplot Creation: The subplot function is used to create a 3-by-1 grid of plots.

4. Plotting: Each subplot is populated with the corresponding plot of g(t) for each time vector.
5. Period Calculation: The period of the signal is calculated based on the least common multiple of the function’s component’s periods [¼,½,⅙] which is 0.5 seconds.

MATLAB Code:

```
% Define the function g(t)
g = @(t) 3*pi*sin(8*pi*t + 1.3).*cos(4*pi*t - 0.8).*exp(sin(12*pi*t));

% Define the time ranges for different step sizes
t1 = -1:0.1:1;
t2 = -1:0.01:1;
t3 = -1:0.001:1;

% Create a figure
figure;

% Plot for step size 0.1
subplot(3, 1, 1);
plot(t1, g(t1), 'b');
title('g(t) with step size 0.1');
xlabel('t');
ylabel('g(t)');
grid on;

% Plot for step size 0.01
subplot(3, 1, 2);
plot(t2, g(t2), 'r');
title('g(t) with step size 0.01');
xlabel('t');
ylabel('g(t)');
grid on;

% Plot for step size 0.001
subplot(3, 1, 3);
plot(t3, g(t3), 'g');
```

```
title('g(t) with step size 0.001');
xlabel('t');
ylabel('g(t)');
grid on;

% Calculate the period of the signal
% The period can be determined from the sine and cosine components
T = 0.5;
disp(['The period of the signal is: 0.5 seconds']);
```

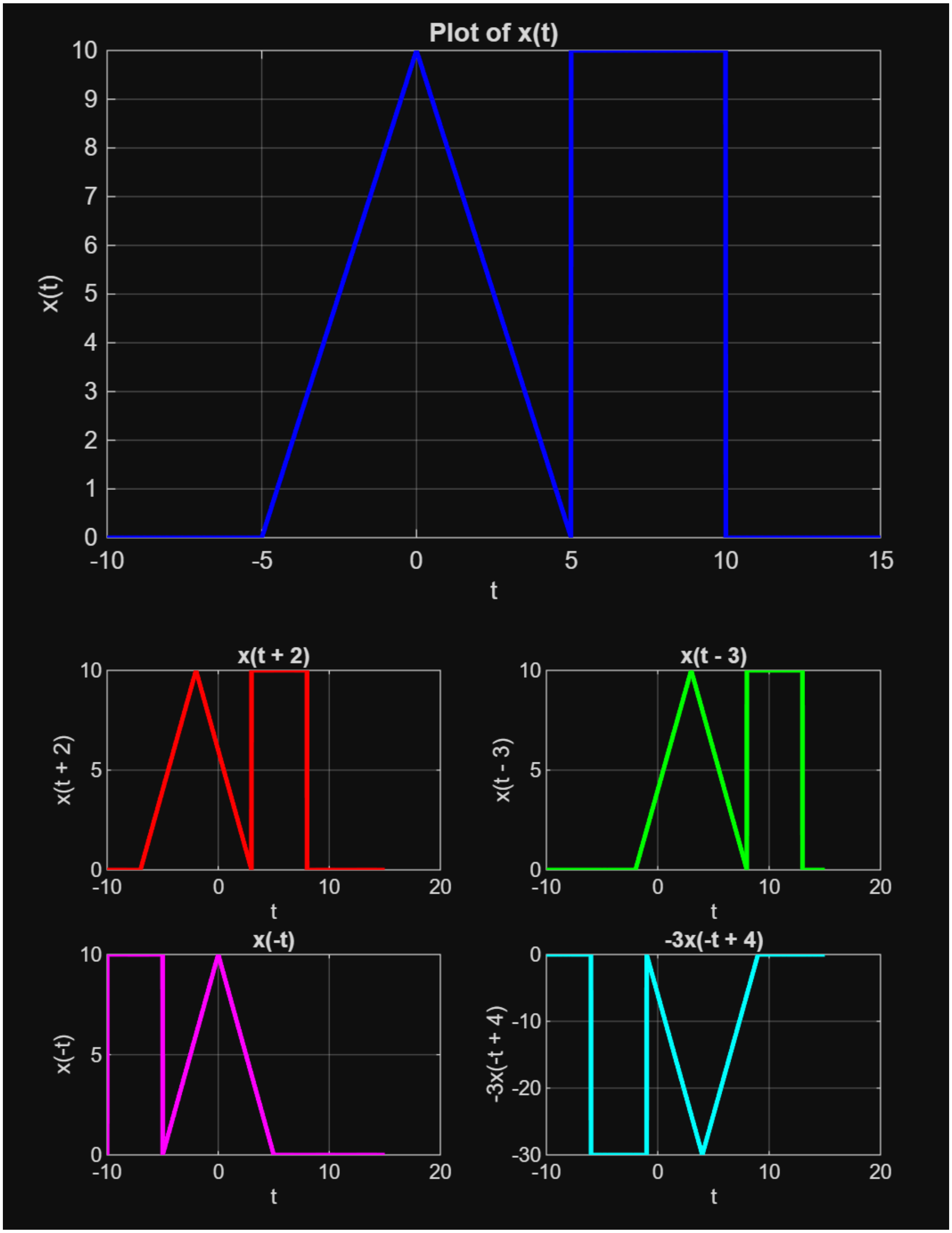
2. For the following function

$$x(t) = \begin{cases} -2|t| + 10, & t \in [-5,5) \\ 10, & t \in [5,10) \\ 0, & \text{elsewhere} \end{cases}$$

create an m-file that plots the function x(t) within the window $t \in [-10,15]$. Also create a separate figure that has 4 subplots in 2-by-2 arrangement with the following signals:

(a) x(t+2)
(b) x(t-3)
(c) x(-t)
(d) -3x(-t+4)

Final Answer:



The MATLAB code provided will generate the required plots for the function x(t) and its transformations.

Detailed Solution:

1. Function Definition: The piecewise function x(t) is defined using an anonymous function in MATLAB. The function uses logical indexing to apply the correct formula based on the value of t.
2. Time Vector: A time vector t is created from -10 to 15 with a small step size (0.01) for smooth plotting.
3. First Figure: The first figure plots x(t) over the specified range.
4. Second Figure with Subplots: A new figure is created with a 2-by-2 arrangement of subplots for the transformed signals:
    - x(t + 2)
    - x(t - 3)
    - x(-t)
    - -3x(-t + 4)

MATLAB Code:

```
% Define the piecewise function x(t)
x = @(t) (-2*abs(t) + 10) .* (t >= -5 & t < 5) + ...
     10 * (t >= 5 & t < 10) + ...
     0 * (t < -5 | t >= 10);

% Define the time vector for plotting
t = -10:0.01:15; % Fine resolution for smooth plotting

% Create the first figure for x(t)
figure;
plot(t, x(t), 'b', 'LineWidth', 2);
title('Plot of x(t)');
xlabel('t');
ylabel('x(t)');
grid on;

% Create a new figure for the transformed signals
figure;

% Define the transformed signals
subplot(2, 2, 1);
```

```
plot(t, x(t + 2), 'r', 'LineWidth', 2);
title('x(t + 2)');
xlabel('t');
ylabel('x(t + 2)');
grid on;

subplot(2, 2, 2);
plot(t, x(t - 3), 'g', 'LineWidth', 2);
title('x(t - 3)');
xlabel('t');
ylabel('x(t - 3)');
grid on;

subplot(2, 2, 3);
plot(t, x(-t), 'm', 'LineWidth', 2);
title('x(-t)');
xlabel('t');
ylabel('x(-t)');
grid on;

subplot(2, 2, 4);
plot(t, -3*x(-t + 4), 'c', 'LineWidth', 2);
title('-3x(-t + 4)');
xlabel('t');
ylabel('-3x(-t + 4)');
grid on;
```

3. Consider the signal $x(t) = te^{-0.15t}, -20 \leq t \leq 20$. Plot

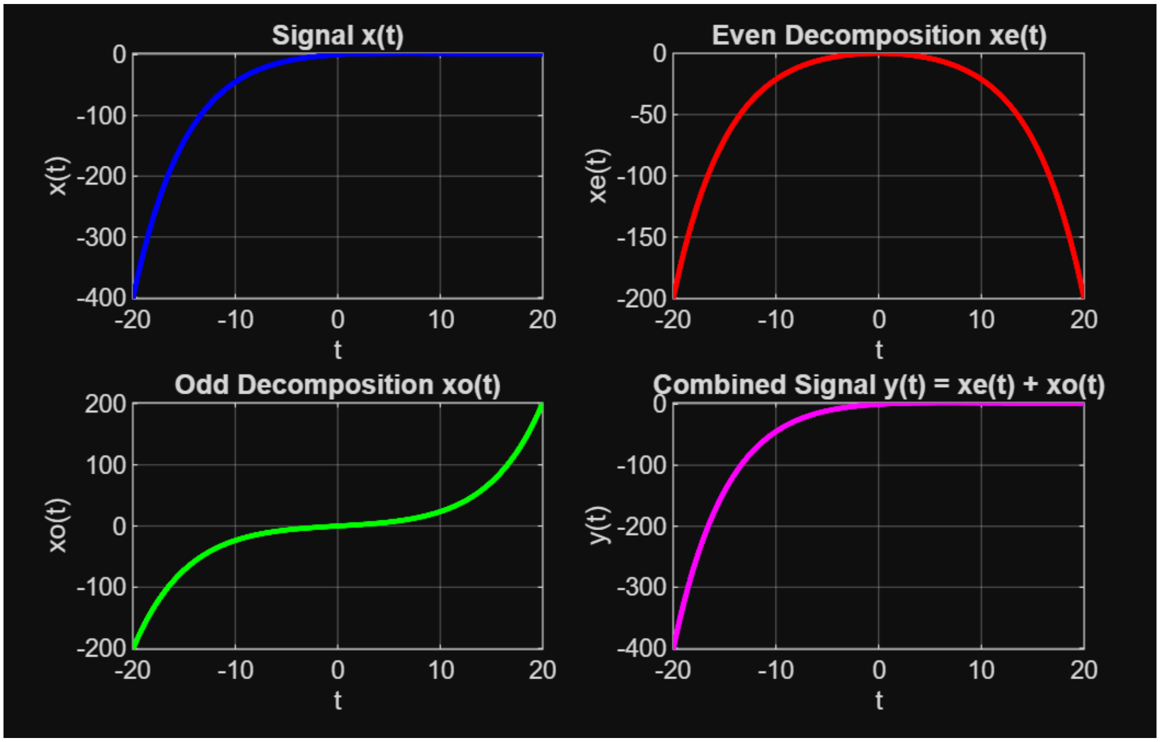
(a) The signal $x(t)$

(b) The even decomposition $x_e(t)$ of $x(t)$

(c) The odd decomposition $x_o(t)$ of $x(t)$

(d) The signal $y(t)=x_e(t)+x_o(t)$

Final Answer:



The MATLAB code provided will generate the required plots for the signal x(t), its even and odd decompositions, and the combined signal y(t).

Detailed Solution:

1. Signal Definition: The signal (x(t)) is defined as x(t) = te^(-0.15t).
2. Time Vector: A time vector t is created from -20 to 20 with a small step size (0.01) for smooth plotting.
3. Plotting: The first subplot displays the original signal (x(t)).
4. Even and Odd Decompositions:
    - The even part (x_e(t)) is calculated using the formula (x_e(t) = (x(t) + x(-t))/2).
    - The odd part (x_o(t)) is calculated using the formula (x_o(t) = (x(t) - x(-t))/2).
5. Combined Signal: The combined signal y(t) is simply the sum of the even and odd components.
6. Final Plots: The even decomposition, odd decomposition, and combined signal are plotted in the remaining subplots.

MATLAB Code:

% Define the time vector

```
t = -20:0.01:20; % Time range from -20 to 20

% Define the signal x(t)
x = @(t) t .* exp(-0.15 * t); % x(t) = te^(-0.15t)

% Calculate the signal x(t)
xt = x(t);

% Plot x(t)
figure;
subplot(2, 2, 1);
plot(t, xt, 'b', 'LineWidth', 2);
title('Signal x(t)');
xlabel('t');
ylabel('x(t)');
grid on;

% Calculate even and odd decompositions
xe = @(t) (x(t) + x(-t)) / 2; % Even part
xo = @(t) (x(t) - x(-t)) / 2; % Odd part

% Calculate the even and odd signals
xe_t = xe(t);
xo_t = xo(t);

% Plot the even decomposition xe(t)
subplot(2, 2, 2);
plot(t, xe_t, 'r', 'LineWidth', 2);
title('Even Decomposition xe(t)');
xlabel('t');
ylabel('xe(t)');
grid on;

% Plot the odd decomposition xo(t)
subplot(2, 2, 3);
plot(t, xo_t, 'g', 'LineWidth', 2);
title('Odd Decomposition xo(t)');
```

```
xlabel('t');
ylabel('xo(t)');
grid on;

% Calculate the combined signal y(t) = xe(t) + xo(t)
y_t = xe_t + xo_t;

% Plot the combined signal y(t)
subplot(2, 2, 4);
plot(t, y_t, 'm', 'LineWidth', 2);
title('Combined Signal y(t) = xe(t) + xo(t)');
xlabel('t');
ylabel('y(t)');
grid on;
```

4. For the signal g(x) in problem 1, calculate the energy of the signal in the window $t \in [0.25,0.75]$. Also calculate the power of the signal.

Final Answer:

```
Energy of the signal in the interval [0.25, 0.75]: 27.3775
Power of the signal in the interval [0.25, 0.75]: 54.755
```

- The energy of the signal in the interval [0.25, 0.75] is calculated using the integral.
- The power of the signal in the same interval is the energy divided by the interval length.

Detailed Solution

1. Signal Definition: The function g(t) is defined using an anonymous function in MATLAB.
2. Energy Calculation: The integral function is used to compute the integral of (|g(t)|^2) over the interval [0.25, 0.75].
3. Power Calculation: The average power is calculated by dividing the energy by the duration of the interval (b - a).
4. Results Display: The results for energy and power are displayed in the MATLAB command window.

MATLAB Code:

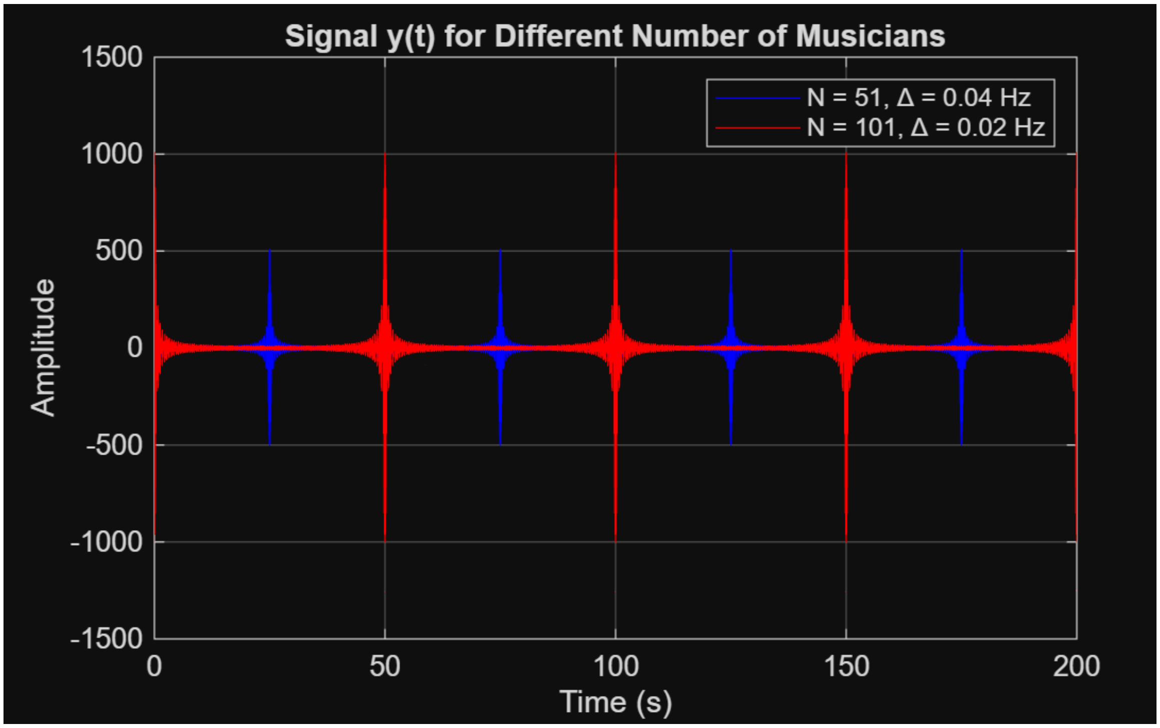
```
% Define the signal g(t)
g = @(t) 3*pi*sin(8*pi*t + 1.3).*cos(4*pi*t - 0.8).*exp(sin(12*pi*t));

% Define the time interval
a = 0.25;
b = 0.75;

% Calculate the energy of the signal
energy = integral(@(t) abs(g(t)).^2, a, b);

% Calculate the power of the signal
power = energy / (b - a);

% Display the results
disp(['Energy of the signal in the interval [0.25, 0.75]: ', num2str(energy)]);
disp(['Power of the signal in the interval [0.25, 0.75]: ', num2str(power)]);
```

5. Suppose N different musicians in an orchestra are trying to play a pure tone, a sinusoid of frequency 160 Hz. Assume the N players while trying to play the pure tone (160 Hz) end up playing tones separated by Δ Hz, so the overall sound they produced is:

$$y(t) = \sum_{i=1}^{N} 10 \cos(2\pi f_i t)$$

where the f_i are the frequencies from 159 to 161 Hz. Generate the signal y(t), $0 \leq t \leq 200$ sec considering that each musician is playing a unique frequency. First assume the number of musicians to be N = 51 with Δ= 0.04 Hz, and then N=101 with Δ= 0.02 Hz. Plot y(t) for the two cases on the same figure.

Final Answer:



The MATLAB code provided will generate the signal y(t) for both cases and plot them on the same figure, allowing for a visual comparison of the effects of different numbers of musicians and frequency separations.

Detailed Solution:

1. Time Vector: A time vector t is created from 0 to 200 seconds with a sampling rate of 1000 Hz for smooth plotting.
2. Frequency Calculation: For both cases, the frequencies are calculated based on the number of musicians and the frequency separation Δ.
3. Signal Generation: The signal (y(t)) is computed by summing the contributions from each musician's frequency.
4. Plotting: Both signals are plotted on the same figure, with different colors for clarity.

MATLAB Code:

```
% Define the time vector
t = 0:0.001:200; % Time from 0 to 200 seconds with a sampling rate of 1000 Hz

% Case 1: N = 51, Δ = 0.04 Hz
N1 = 51;
```

```
delta1 = 0.04;
f1 = 159 + (0:N1-1) * delta1; % Frequencies from 159 Hz to 161 Hz
y1 = zeros(size(t)); % Initialize y(t)

% Generate the signal y(t) for N = 51
for i = 1:N1
   y1 = y1 + 10 * cos(2 * pi * f1(i) * t);
end

% Case 2: N = 101, Δ = 0.02 Hz
N2 = 101;
delta2 = 0.02;
f2 = 159 + (0:N2-1) * delta2; % Frequencies from 159 Hz to 161 Hz
y2 = zeros(size(t)); % Initialize y(t)

% Generate the signal y(t) for N = 101
for i = 1:N2
   y2 = y2 + 10 * cos(2 * pi * f2(i) * t);
end

% Plotting the results
figure;
plot(t, y1, 'b', 'DisplayName', 'N = 51, Δ = 0.04 Hz');
hold on;
plot(t, y2, 'r', 'DisplayName', 'N = 101, Δ = 0.02 Hz');
title('Signal y(t) for Different Number of Musicians');
xlabel('Time (s)');
ylabel('Amplitude');
legend show;
grid on;
hold off;
```