ENG204 - Signals and Linear Systems – Assignment 1.1

[2024-08-05 Mon 17:19]

Contents

1 ENG204 - Signals and Linear Systems – Assignment 1.1

1.1 Part A

Cases For Convolution

1.2 Part B

```matlab
% True Answer
T = 50;
A = 0.05; % This amplitude shows the input and output well
xDiscrete = -T:1:2*T;


fDiscrete = zeros(size(xDiscrete));
for i = 1:length(xDiscrete)
    x = xDiscrete(i) + T/2;
    if  x < 0
        fDiscrete(i) = 0;
    elseif -T/2 <= x && x < T/4
        fDiscrete(i) = A^2 * x;
    elseif T/4 <= x && x < T/2
        fDiscrete(i) = A^2 * (-3 * x + T);
    elseif T/2 <= x && x < 3*T/4
        fDiscrete(i) = A^2 * (x - T);
    elseif 3*T/4 <= x && x <= T
        fDiscrete(i) = A^2 * (5 * x - 4 * T);
    else
        fDiscrete(i) = 0;
    end
end

fFlippedDiscrete = zeros(size(xDiscrete));
for i = 1:length(xDiscrete)
    x = -xDiscrete(i) + 1.5 * T;
    if 0 <= x && x < T/4
        fFlippedDiscrete(i) = A^2 * x;
    elseif T/4 <= x && x < T/2
```

```
        fFlippedDiscrete(i) = A^2 * (-3 * x + T);
    elseif T/2 <= x && x < 3*T/4
        fFlippedDiscrete(i) = A^2 * (x - T);
    elseif 3*T/4 <= x && x <= T
        fFlippedDiscrete(i) = A^2 * (5 * x - 4 * T);
    else
        fFlippedDiscrete(i) = 0;
    end
end

% Discrete Approximation
sDiscrete = zeros(size(xDiscrete));
hDiscrete = zeros(size(xDiscrete));
for i = 1:length(xDiscrete)
    x = xDiscrete(i);
    if -T <= x && x < 0
        sDiscrete(i) = 0;
        hDiscrete(i) = 0;
    elseif 0 <= x && x < T/4
        sDiscrete(i) = A;
        hDiscrete(i) = A;
    elseif T/4 <= x && x < 3*T/4
        sDiscrete(i) = -A;
        hDiscrete(i) = -A;
    elseif 3*T/4 <= x && x < T
        sDiscrete(i) = A;
        hDiscrete(i) = A;
    elseif T <= x && x <= 2*T
        sDiscrete(i) = 0;
        hDiscrete(i) = 0;
    end
end

C = conv(sDiscrete, hDiscrete, 'same');

figure;
hold on;
plot(xDiscrete, sDiscrete, 'b', 'DisplayName', 's(x)', 'LineWidth', 1.5);
plot(xDiscrete, hDiscrete, 'r', 'DisplayName', 'h(x)', 'LineWidth', 1.5);
plot(xDiscrete, C, 'k', 'DisplayName', 'Convolution Result', 'LineWidth', 1.5);
plot(xDiscrete, fDiscrete, 'g', 'DisplayName', 'f(x)', 'LineWidth', 1.5);
plot(xDiscrete, fFlippedDiscrete, 'm', 'DisplayName', 'f(x) flipped about y-axis', 'LineWidth', 1.5);

title('Original Functions and Convolution Result');
xlabel('x');
ylabel('Amplitude');
xlim([min(xDiscrete), max(xDiscrete)]);
ylim([min(C), max(C)]);
grid on;
legend;
hold off;
```

1.3 Part C

```
% Load the data from the .p file
% This data is preloaded into a .ods spreadsheet, as octave could not run the .p file
% It has the student ids 652137 and 651790
pkg load io;
```

```
data = odsread("/home/Baley/UTAS/ENG204 - Signals And Linear Systems/Assignment 1.1/Data.ods", 1);
Datat = data(1:end,1);
Datart = data(1:end,2);

A = 1;
T = Datat(end); % Use the last value of Datat as T

% Create Walsh Functions using logical indexing
t = Datat; % Use Datat as the time variable

% Initialize the Walsh functions
s1 = zeros(size(t));
s2 = zeros(size(t));
s3 = zeros(size(t));
s4 = zeros(size(t));

% Define the piecewise functions using logical conditions
s1(t >= 0 & t < T) = A;

s2(t >= 0 & t < T/2) = A;
s2(t >= T/2 & t < T) = -A;

s3(t >= 0 & t < T/4) = A;
s3(t >= T/4 & t < 3*T/4) = -A;
s3(t >= 3*T/4 & t < T) = A;

s4(t >= 0 & t < T/4) = A;
s4(t >= T/4 & t < T/2) = -A;
s4(t >= T/2 & t < 3*T/4) = A;
s4(t >= 3*T/4 & t < T) = -A;

% Matched Filter Bank
h1 = flip(s1); % Flip the Walsh functions for matched filtering
h2 = flip(s2);
h3 = flip(s3);
h4 = flip(s4);

% Perform convolution
D1 = conv(Datart, h1, 'same');
D2 = conv(Datart, h2, 'same');
D3 = conv(Datart, h3, 'same');
D4 = conv(Datart, h4, 'same');

% Plot the results
figure; % Create a new figure
hold on; % Hold on to plot multiple lines
plot(Datat, D1, 'DisplayName', 'Convolution with s1', 'LineWidth', 1.5);
plot(Datat, D2, 'DisplayName', 'Convolution with s2', 'LineWidth', 1.5);
plot(Datat, D3, 'DisplayName', 'Convolution with s3', 'LineWidth', 1.5);
plot(Datat, D4, 'DisplayName', 'Convolution with s4', 'LineWidth', 1.5);
xlabel("Time");
ylabel("Convolution");
title("Convolution Results with Walsh Functions");
grid on; % Add grid for better visibility
legend; % Show legend to identify each plot
hold off; % Release the hold

% 16 bits so the sample time will be 8, as there are two bits per output
```

```
sizeData = length(Datat);
output = "";

for i = 1:8
    idx = round((1/8) * sizeData * i); % Use round to get an integer index
    Max = max([D1(idx), D2(idx), D3(idx), D4(idx)]);
    if Max == D1(idx)
        output = [output, "00"];
    elseif Max == D2(idx)
        output = [output, "01"];
    elseif Max == D3(idx)
        output = [output, "10"];
    elseif Max == D4(idx)
        output = [output, "11"];
    end
end

outputDecimal = bin2dec(strjoin(cellstr(output), ''));
disp(outputDecimal);
```

```
23168
```