یا لطیف

دانشکده مهندسی برق

آزمایشگاه DSP - گزارشکار آزمایش شماره ۲

موضوع آزمایش:

تابع تولید دلتا، پالس و بخش زوج و فرد سیگنال

تهیه کننده و نویسنده:

رضا آدینه پور

استاد:

جناب آقای دکتر مهدی مقیمی

تاریخ تهیه و ارائه:

فروردین ماه ۱۴۰۲

۱) بدون استفاده از توابع آماده متلب، مطابق رابطه زیر تابعی برای ایجاد تابع ضربه بنویسید. (ورودی تابع:

n0,n1,n2 خروجی تابع x,n)

ضربه‌ی واحد در بازه‌ی **[n1,n2]** که ضربه در نقطه‌ی $n=n_0$ رخ داده باشد،

$$\delta(n-n_0)=\begin{cases}1, & n=n_0\\ 0, & n\neq n_0\end{cases}$$

تابع نوشته شده به صورت زیر است:

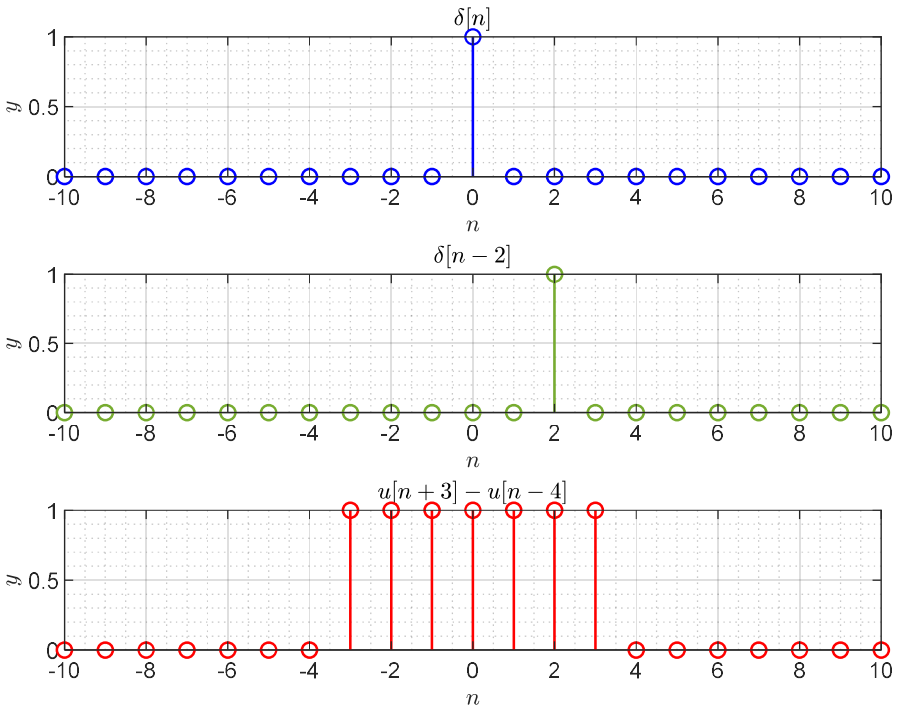
```
function [y] = SS_delta(n, width)

    % SS_delta returns unit impulse function
    % SS_delta(n) retutns the value 0 for n < 0 and n > 0, and 1 for n = 0;
    % SS_delta(n, width) returns the value 1/(2 * width) for -width < n < width,
and 0 for other times.

    if (nargin == 1)
        y = (n == 0);
    else
        y = 1 * ((n >= -width) & (n <= width));
    end
end
```

کد نوشته شده برای تست تابع به صورت زیر است:

```
%           ******************************************************
%          **   course         : DSP-Lab                         **
%         ***   HomeWork       : 02                              ***
%        ****   Topic          : Customize Function              ****
%        ****   AUTHOR         : Reza Adinepour                  ****
%         ***   Student ID:    : 9814303                         ***
%          **   Github         : github.com/reza_adinepour/      **
%           ******************************************************
clear; clc; close all;

%% part1: test delta dirac function
N1 = -10;
N2 = 10;
n = N1:N2; %sequence range

f1 = SS_delta(n);
f2 = SS_delta(n - 2);
f3 = SS_delta(n, 3); %create pulse with pulse width = 2 * 3 + 1

figure('Name','delta dirac function');
subplot(3, 1, 1);
stem(n, f1, 'LineWidth', 1, 'Color', 'b');
title('$\delta[n]$', 'Interpreter','latex');
```

```
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(3, 1, 2);
stem(n, f2, 'LineWidth', 1, 'color', '#77AC30');
title('$\delta[n - 2]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(3, 1, 3);
stem(n, f3, 'LineWidth', 1, 'Color', 'r');
title('$u[n + 3]-u[n - 4]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;
```

خروجی کد به صورت زیر است:



۲) مانند تمرین قبل، تابعی برای ایجاد تابع پله واحد بنویسید. (ورودی تابع: n0,n1,n2 خروجی تابع x,n)

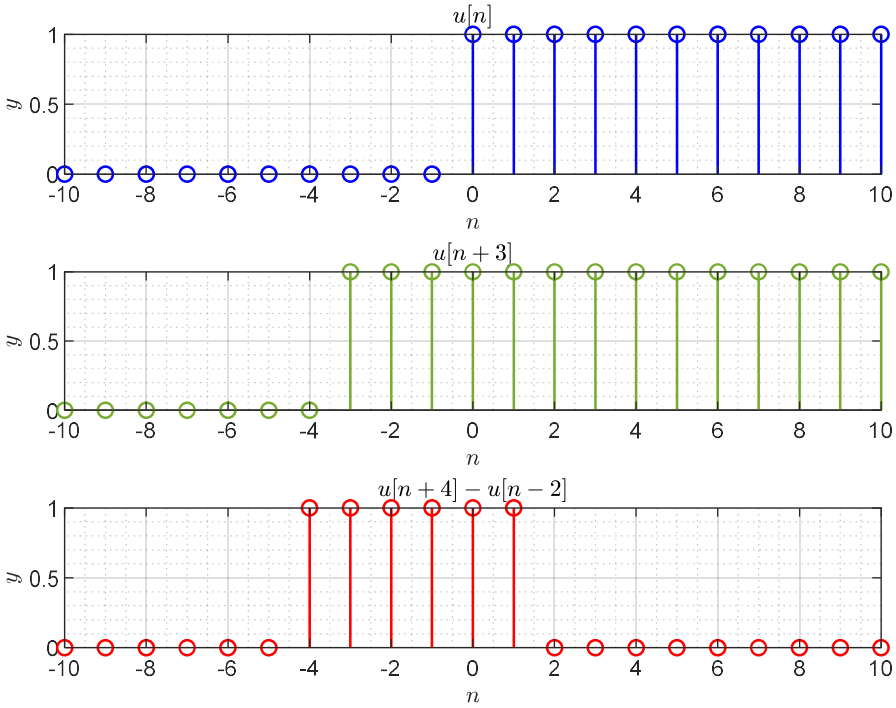
تابع نوشته شده به صورت زیر است:

```
function [y] = SS_u(n)

    y = 1.*(n >= 0); % return 1 for n >= 0 and 0 for others.

end
```

کد نوشته شده برای تست تابع به صورت زیر است:

```
%% part2: test pulse function
f4 = SS_u(n);
f5 = SS_u(n + 3);
f6 = SS_u(n + 4) - SS_u(n - 2);

figure('Name','pulse function');
subplot(3, 1, 1);
stem(n, f4, 'LineWidth', 1, 'Color', 'b');
title('$u[n]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(3, 1, 2);
stem(n, f5, 'LineWidth', 1, 'color', '#77AC30');
title('$u[n + 3]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(3, 1, 3);
stem(n, f6, 'LineWidth', 1, 'Color', 'r');
title('$u[n + 4]-u[n - 2]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;
```

خروجی کد به صورت زیر است:



۳) می‌دانیم که هر سیگنالی را میتوان به صورت مجموع یک سیگنال زوج و یک سیگنال فرد توصیف نمود، برنامه‌ای بنویسید که برای هر دنباله سیگنال ورودی، آن را به بخش‌های زوج و فرد تجزیه کند.

۴) با استفاده از توابع مراحل قبل سیگنال $x_{[n]}=u_{[n]}-u_{[n-10]}$ را به بخش‌های زوج و فرد تجزیه کنید.

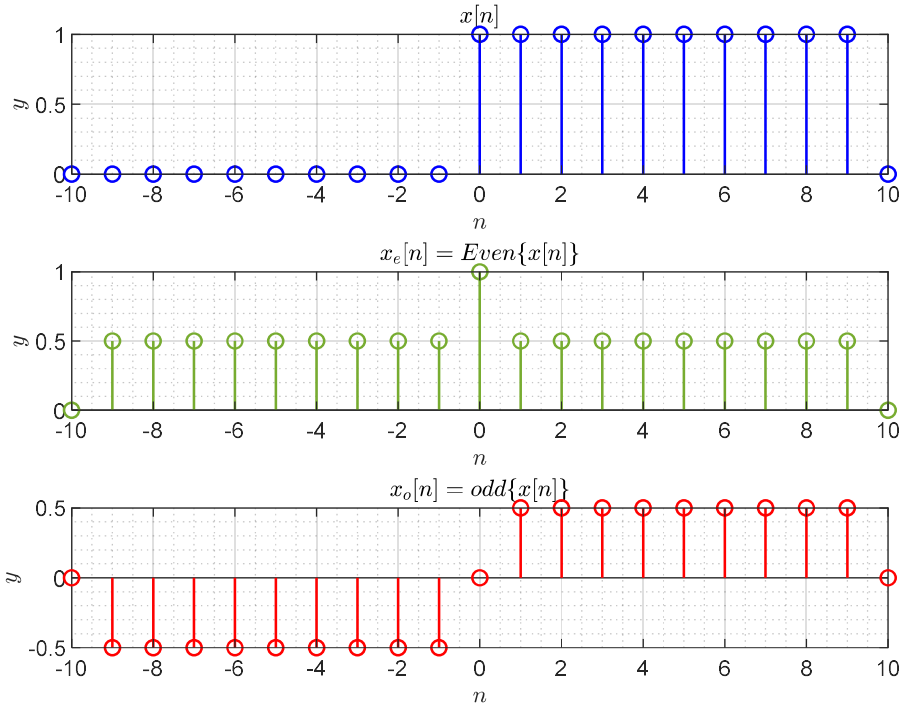
تابع نوشته شده به صورت زیر است:

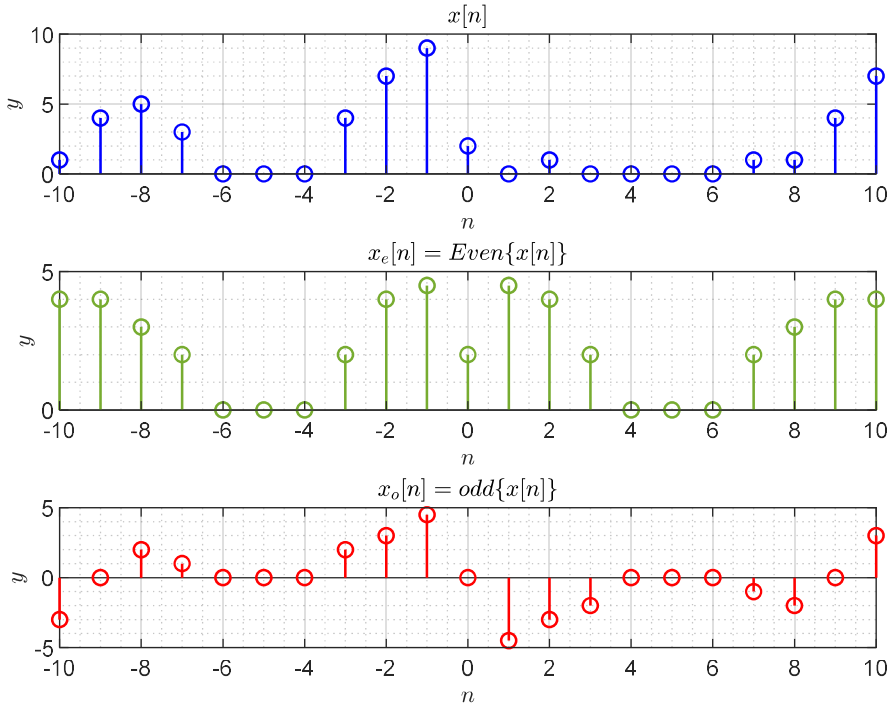
```
function [ye, yo] = SS_eop(x)
    % This function takes a discrete signal x and returns its even and odd parts.
    % The output is two vectors, even and odd.

    ye = 0.5 * (x + fliplr(x));
    yo = 0.5 * (x - fliplr(x));
end
```

کد نوشته شده برای تست تابع به ازای دو ورودی متفاوت به صورت زیر است:

```
%% part3: even and odd part of signals
f7 = [1 4 5 3 0 0 0 4 7 9 2 0 1 0 0 0 0 1 1 4 7];
% f7 = SS_u(n) - SS_u(n - 10);
if(length(f7) ~= length(n))
    disp('length not be same!')
else
    [f7e, f7o] = SS_eop(f7);

    figure('Name','even and odd part');
    subplot(3, 1, 1);
    stem(n, f7, 'LineWidth', 1, 'Color', 'b');
    title('$x[n]$', 'Interpreter','latex');
    xlabel('$n$', 'Interpreter','latex');
    ylabel('$y$', 'Interpreter','latex');
    grid on;
    grid minor;

    subplot(3, 1, 2);
    stem(n, f7e, 'LineWidth', 1, 'color', '#77AC30');
    title('$x_e[n] = Even\{x[n]\}$', 'Interpreter','latex');
    xlabel('$n$', 'Interpreter','latex');
    ylabel('$y$', 'Interpreter','latex');
    grid on;
    grid minor;

    subplot(3, 1, 3);
    stem(n, f7o, 'LineWidth', 1, 'Color', 'r');
    title('$x_o[n] = odd\{x[n]\}$', 'Interpreter','latex');
    xlabel('$n$', 'Interpreter','latex');
    ylabel('$y$', 'Interpreter','latex');
    grid on;
    grid minor;
end
```

خروجی کد به صورت زیر است:





۵) در متلب برای عملیات کانولوشن می‌توانید از دستور conv استفاده کنید. بدون استفاده از این تابع، تابعی بنویسید که مشابه آن عمل کند و عملکرد تابع خود را با آن با دادن ورودی و رسم خروجی مقایسه کنید.

تابع نوشته شده به صورت زیر است:

```
function [y, n_y] = SS_conv(h,x, xll, xul, hll, hul)
    n_x = xll:xul;
    n_h = hll:hul;
    y = conv(h,x);
    n_y = n_x(1) + n_h(1):n_x(end) +n_h(end);
end
```

کد نوشته شده برای تست تابع به صورت زیر است:

```
%% part4: convolution of two sequence
[f8, n_y] = SS_conv(f6, f5, N1, N2, N1, N2);
f9 = conv(f6, f5);

figure('Name','convolution');
subplot(4, 1, 1);
stem(n, f6, 'LineWidth', 1, 'Color', 'b');
title('$u[n + 4]-u[n - 2]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(4, 1, 2);
stem(n, f5, 'LineWidth', 1, 'color', '#77AC30');
title('$u[n + 3]$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(4, 1, 3);
stem(n_y, f8, 'LineWidth', 1, 'Color', 'r');
title('$y[n]=x[n]*h[n]\ with\ my\ function$', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;

subplot(4, 1, 4);
stem(n_y, f9, 'LineWidth', 1, 'Color', 'black');
title('$y[n]=x[n]*h[n]$\ with\ matlab\ function', 'Interpreter','latex');
xlabel('$n$', 'Interpreter','latex');
```

```
ylabel('$y$', 'Interpreter','latex');
grid on;
grid minor;
```

خروجی کد به صورت زیر است: