

**دانشکده مهندسی برق**

**آزمایشگاه DSP - گزارشکار آزمایش شماره 2**

**موضوع آزمایش:**

**تابع تولید دلتا، پالس و بخش زوج و فرد سیگنال**

**تهیه کننده و نویسنده:**

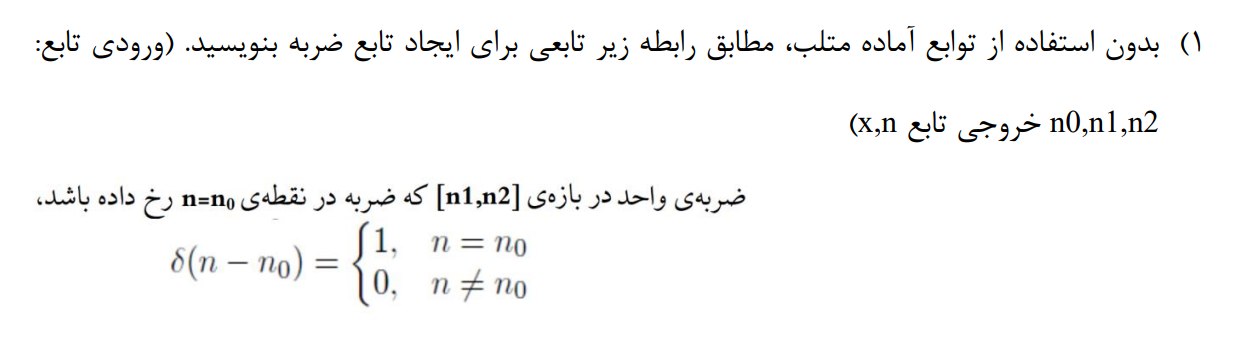
**رضا آدینه پور**

**استاد:**

**جناب آقای دکتر مهدی مقیمی**

**تاریخ تهیه و ارائه:**

**فروردین ماه 1402**

****

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');

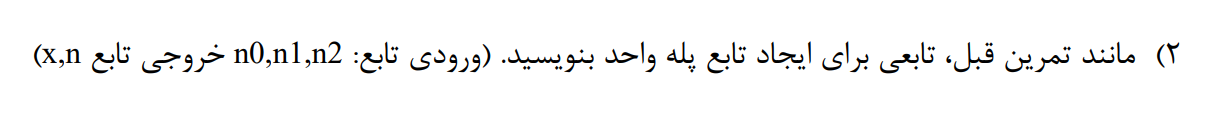
ylabel('$y$', 'Interpreter','latex');

grid on;

grid minor;

خروجی کد به صورت زیر است:



****

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');

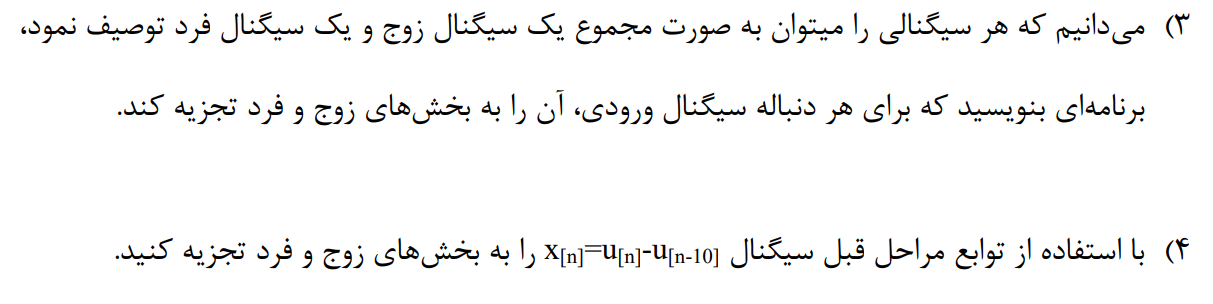
ylabel('$y$', 'Interpreter','latex');

grid on;

grid minor;

خروجی کد به صورت زیر است:

****



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’);

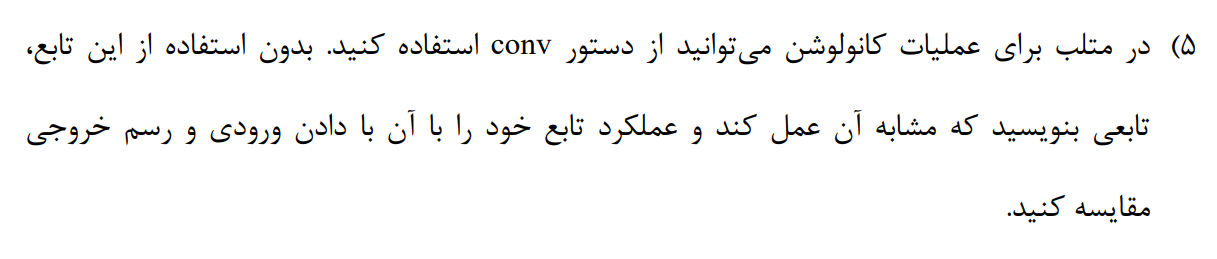
grid on;

grid minor;

end

خروجی کد به صورت زیر است:



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:

