

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

КАФЕДРА КОНСТРУЮВАННЯ ЕОА

Звіт
з лабораторної роботи №3
по курсу
«Цифрове оброблення сигналів»
на тему
«Сигнали та їх властивості. Генерація сигналів в середовищі MatLab»

Лабораторна робота №3

Тема. Сигнали та їх властивості. Генерація сигналів в середовищі MatLab.

Мета: навчитися створювати різні типи сигналів та візуалізувати їх.

Хід роботи

Приклад 1

Генерація сигналу, що представляє суму двох синусоїд – 3 Гц з амплітудою 1 В і 40 Гц з амплітудою 0,25 В. Сигнал, сформований для інтервалів від 0 до 1 с, частота дискретизації - 100 Гц.

```
fs = 100;           % Частота дискретизації 100 Гц
t = 0 : 1 / fs : 1; % Інтервал часу від 0 до 1 с
x = sin(2 * pi * t * 3) + 0.25 * sin(2 * pi * t * 40); % Сума гармонік
plot(t, x);
grid on
```

Результат роботи програми представлений на рисунку 1.

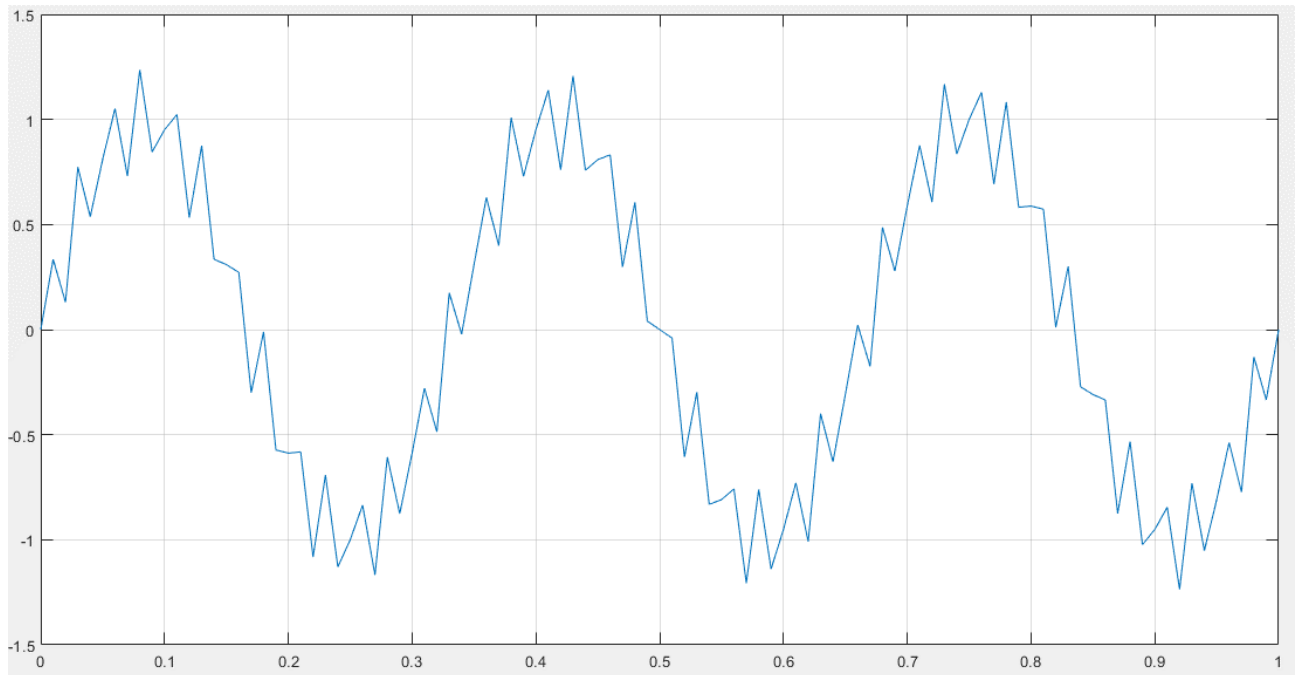


Рисунок 1. Результат роботи програми з прикладу 1.

Приклад 2

Генерація гармонічного сигналу з частотою 1 кГц (частота дискретизації – 8 кГц), амплітуда якого змінюється відповідно до експоненціального закону. Початкова амплітуда сигналу – 2 В, початкова фаза сигналу – 45°.

```

Fs = 8e3;           % Частота дискретизації 8 кГц
t = 0 : 1 / Fs : 1; % 1 секунда дискретних моментів часу
t = t';             % Претворення рядку в стовпчик
A = 2;              % Амплітуда - 2 В
f0 = 1e3;           % Частота - 1 кГц
phi = pi / 4;       % Початкова фаза - 45 град
s1 = A * cos(2 * pi * f0 * t + phi); % Гармонічний сигнал
alpha = 1e3;         % Швидкість затухання експоненти
s2 = exp(-alpha * t) .* s1; % Загасаюча синусоїда
subplot(2, 2, 1);
plot(s2(1 : 100))
subplot(2, 2, 2);
plot(s2(1 : 100), 'r.')
subplot(2, 2, 3);
stem(s2(1:100))
subplot(2, 2, 4);
stairs(s2(1 : 100))

```

Результат роботи програми представлений на рисунку 2.

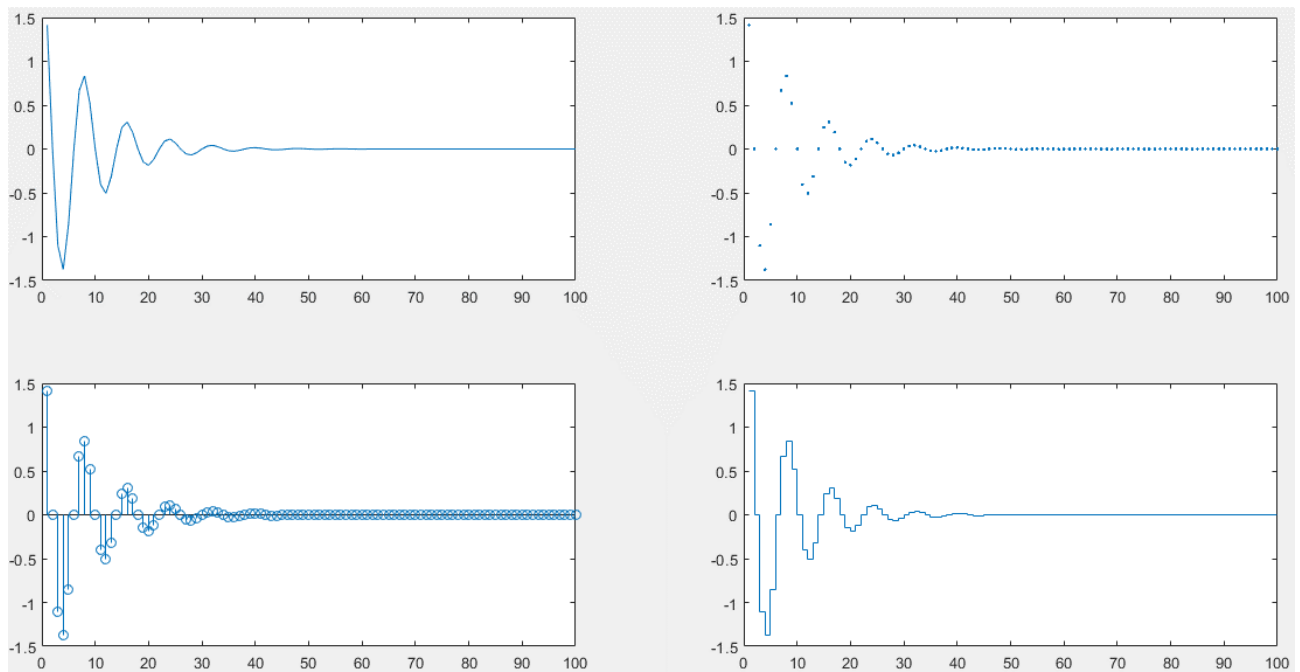


Рисунок 2. Результат роботи програми з прикладу 2.

Приклад 3

Генерація одиночного негативного імпульсу, тривалістю 20 мс і амплітудою – мінус 5 В.

```

Fs = 1e3;
t = -40e-3 : 1 / Fs : 40e-3;
T = 20e-3;
A = 5;
s = -A * rectpuls(t, T);
plot(t, s);
ylim([-6 6]);

```

Результат роботи даної програми:

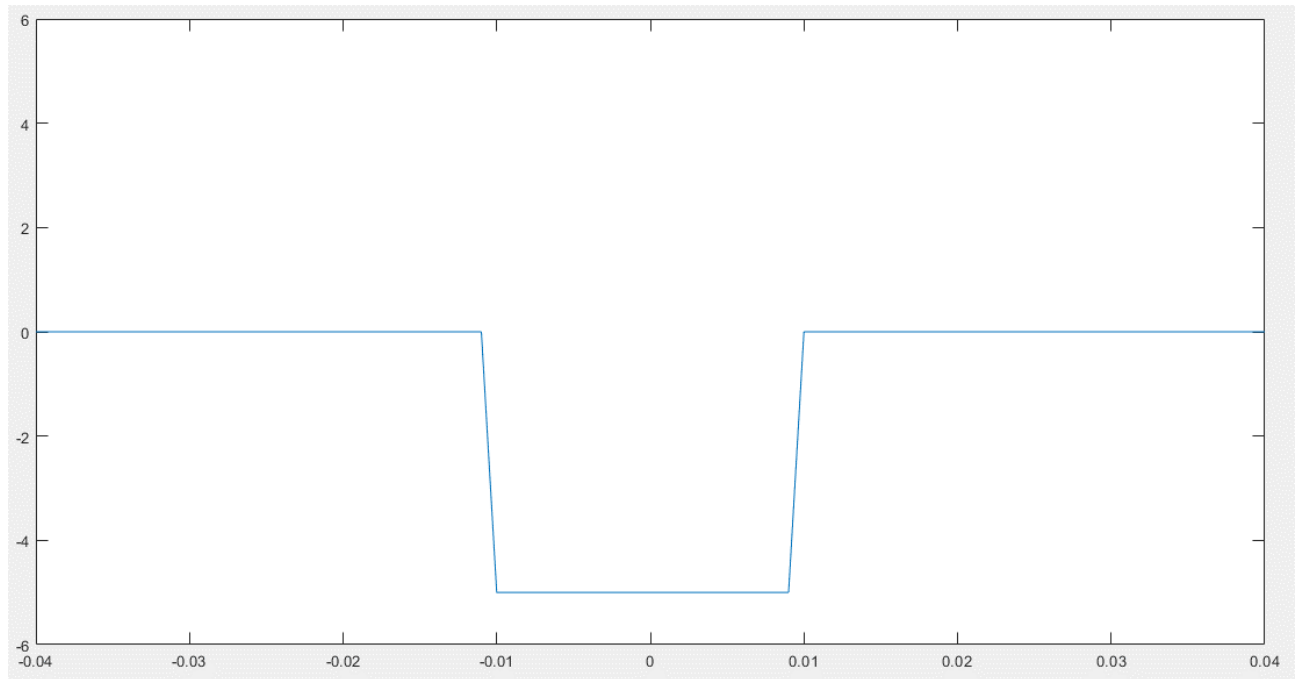


Рисунок 3

Приклад 4

Генерація одиночного трикутного імпульсу, довжиною 60 мс, амплітудою 10 В та зсувом вершини на 15 мс.

```
Fs = 1e3; % Частота дискретизації
t = -50e-3 : 1 / Fs : 50e-3; % Дискретний час
A = 10; % Амплітуда
T1 = 0.5; % Зсув вершини
T2 = 60e-3; % Нижня основа
s = A * tripuls(t, T2, T1); % Генерація імпульсу
plot(t, s)
grid on
```

Результат роботи даної програми:

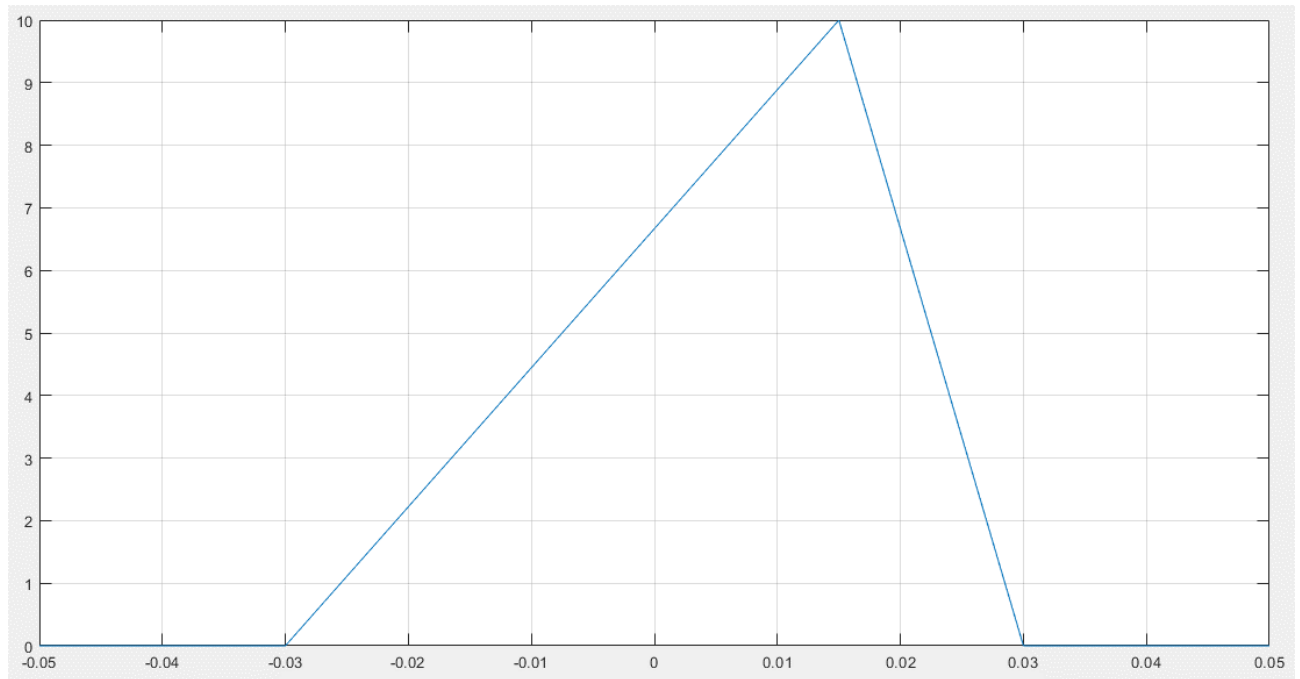


Рисунок 4

Приклад 5

Генерація одиночного радіоімпульсу з частотою = 4 кГц, шириною спектру = 400 Гц, амплітудою = 1 В, рівнем вимірювання ширини спектра = -20 дБ.

```
Fs = 16e3; % Частота дискретизації 16 кГц
t = -10e-3 : 1 / Fs : 10e-3; % Довжина радіоімпульсу
Fc = 4e3; % Опорна частота 4 кГц
bw = 0.1; % Відносна ширина спектру 0.1
bwr = -20; % Рівень = -20 дБ
s = gauspuls(t, Fc, bw, bwr); % Генерація радіоімпульсу
plot(t, s);
grid on
```

Результат роботи даної програми:

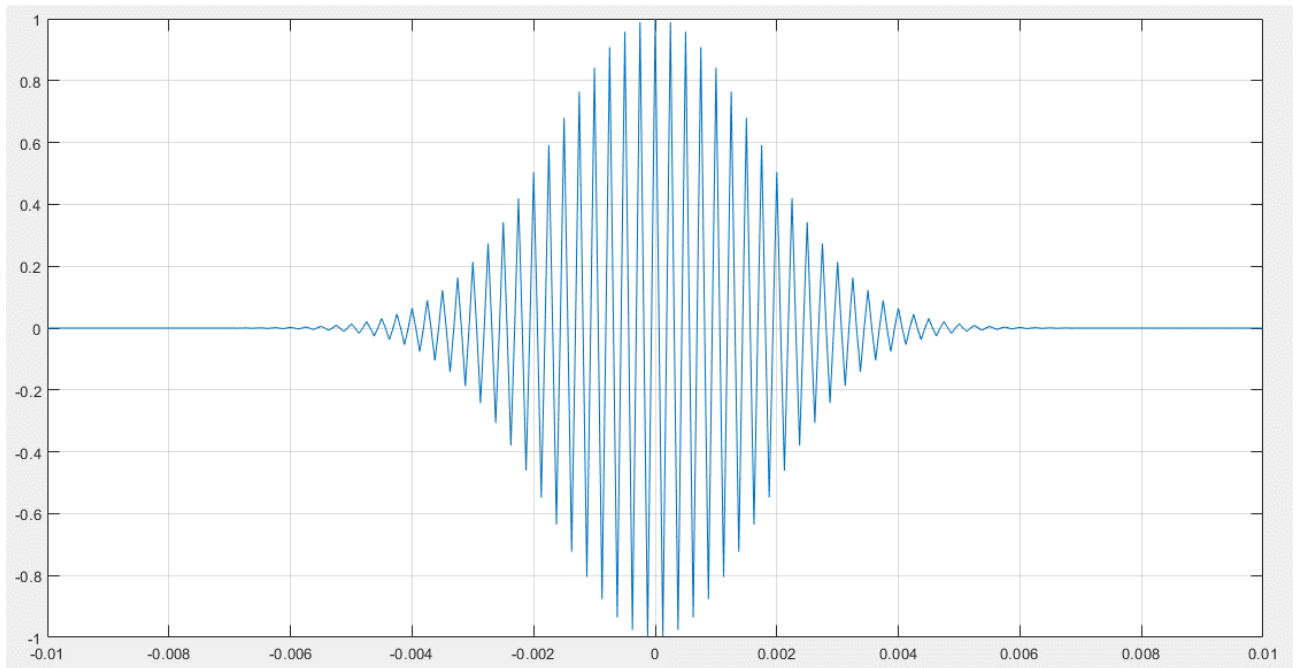


Рисунок 5

Практична частина

1. Відкрийте програму, яка генерує синусоїдальний сигнал з частотою 1 кГц (Sin.m). Програма генерує 64 відліки сигналу з частотою дискретизації 8 кГц.

Генерований синус сигнал описується формулою:

$$x(n) = A \sin \left[\left(2\pi \frac{f}{f_s} \right) n + \varphi \right]$$

де f – частота генеруючого сигналу; f_s – частота дискретизації; φ – фаза, а A – амплітуда генеруючого сигналу.

```
% Generating 64 samples of x(t) = sin(2 * pi * f * t) with a
% Frequency of 1 kHz, and sampling frequency of 8 kHz.
N = 64;                % Define Number of samples
n = 0 : N - 1;         % Define vector n = 0, 1, 2, 3, ... 62, 63
f = 1000;              % Define the frequency
fs = 8000;             % Define the sampling frequency
x = sin(2 * pi * (f / fs) * n); % Generate x(t)
plot(n, x);            % Plot x(t) vs. t
grid;
title('Sinewave [f = 1 kHz, fs = 8 kHz]');
xlabel('Sample Number');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на рисунку:

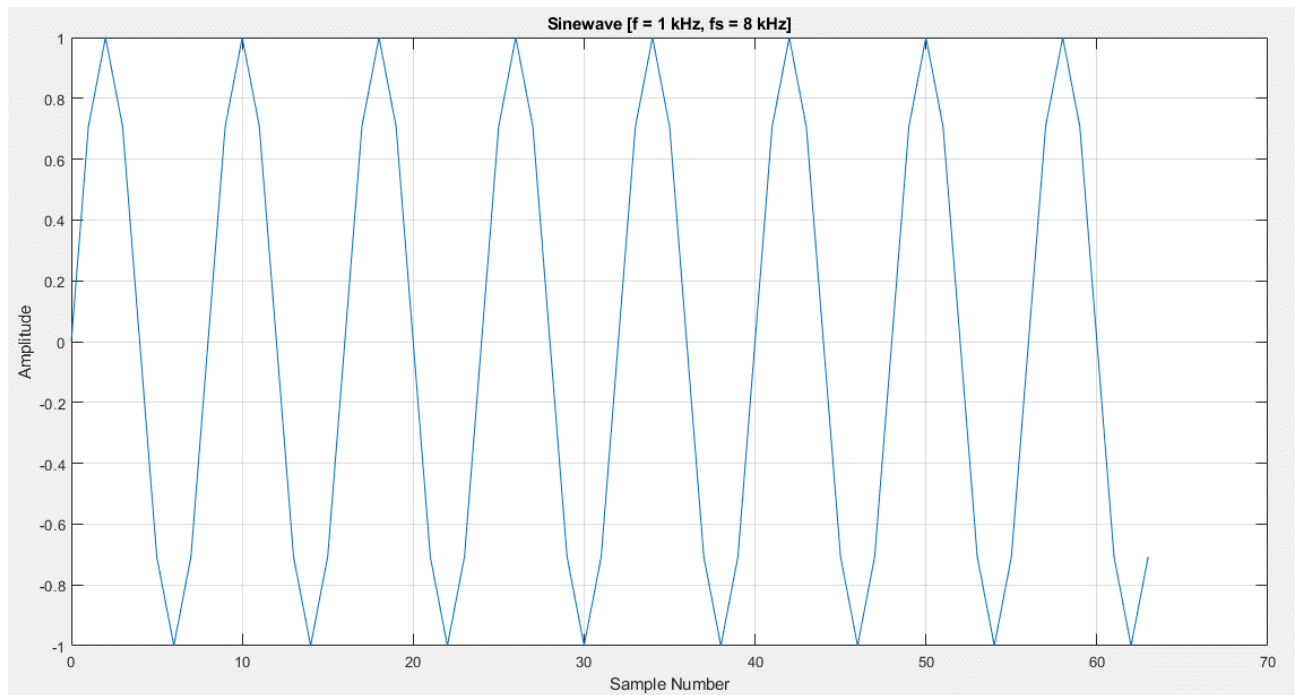


Рисунок 1.1

Завдання

- 1) Згенеруйте 64 відліки косинусоїдального сигналу з частотою 25 Гц, амплітудою 1,5 В і фазовий зсув = 0 при частоті дискретизації 400 Гц.

Файл Cos.m :

```
% Generating 64 samples of  $x(t) = \cos(2 * \pi * f * t + \phi)$  with a  
% Frequency of 25 Hz, and sampling frequency of 400 Hz.  
N = 64;           % Define Number of samples  
A = 1.5;          % Амплітуда  
n = 0 : N - 1;    % Define vector n = 0, 1, 2, 3, ... 62, 63  
f = 25;           % Define the frequency  
fs = 400;         % Define the sampling frequency  
phi = 0;          % фазовий зсув  
x = A * cos(2 * pi * (f / fs) * n + phi); % Generate x(t)  
plot(n, x);       % Plot x(t) vs. t  
grid;  
title('Cosewave [f = 25 Hz, fs = 400 Hz]');  
xlabel('Sample Number');  
ylabel('Amplitude');
```

Результат роботи даної програми:

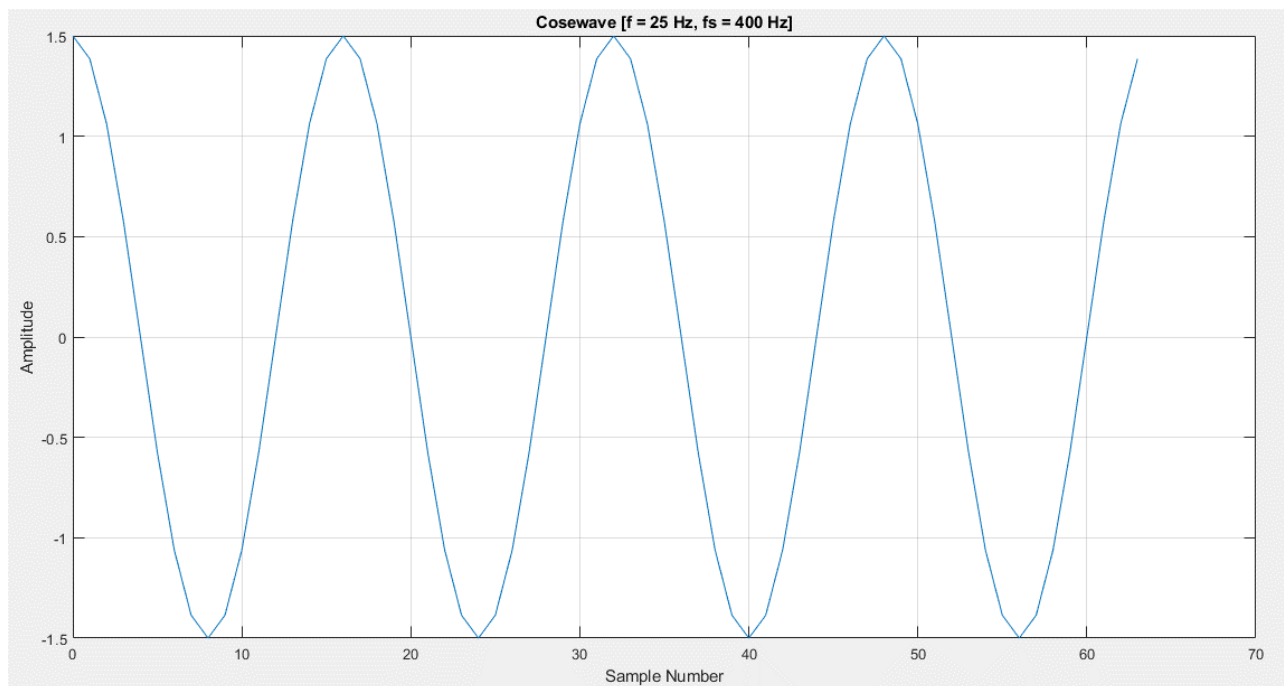


Рисунок 1.2

- 2) Для згенерованого в завданні 1 сигналу додайте фазовий зсув $= 45^\circ$ (тобто $\pi / 4$).

Зміни в файлі Cos.m :

```
phi = pi / 4;      % фазовий зсув
```

Результат роботи програми:

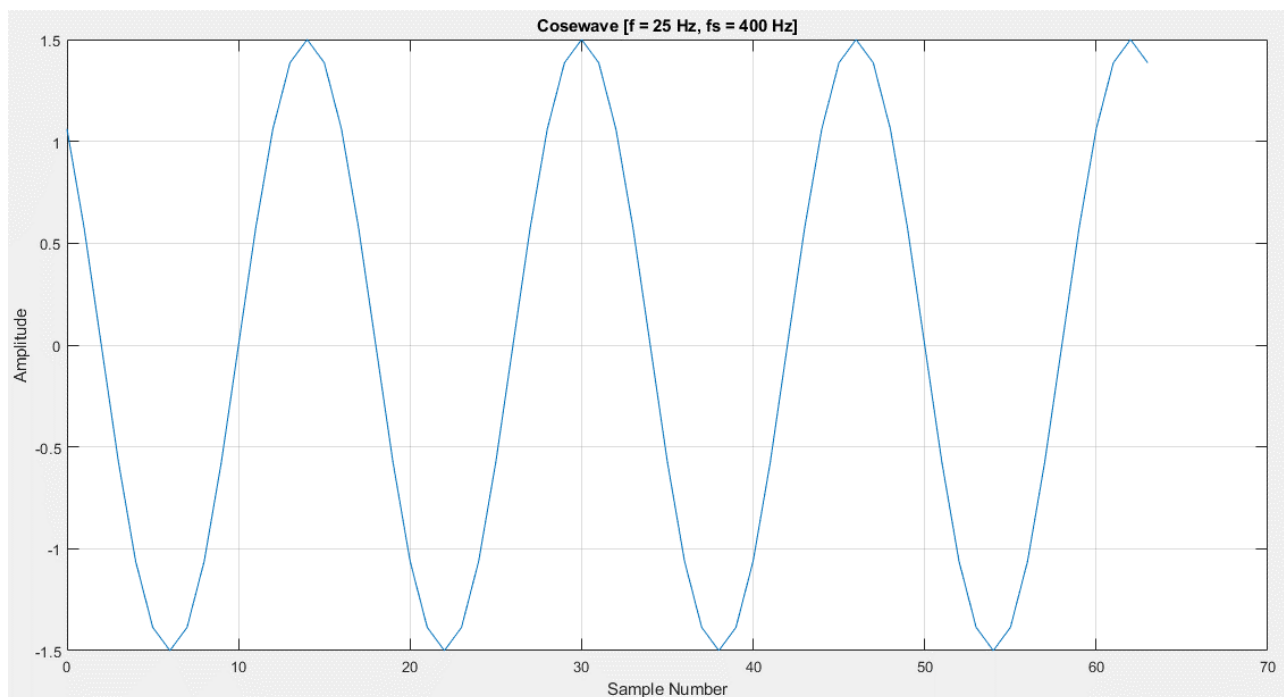


Рисунок 1.3

2. Відкрийте програму, яка генерує експонентний сигнал (Exp.m), який описується формулою:

$$x = e^{(-0.1*t)}$$

Змінна t змінюється в інтервалі від 0 до 40 мс з кроком 0.1 мс.

```
% Generating the signal x(t) = exp(-0.1t)
t = 0 : 0.1 : 40;
x = exp(-0.1 * t);
plot(t, x);
grid;
title('Exponential Signal');
xlabel('Time [ms]');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на рисунку:

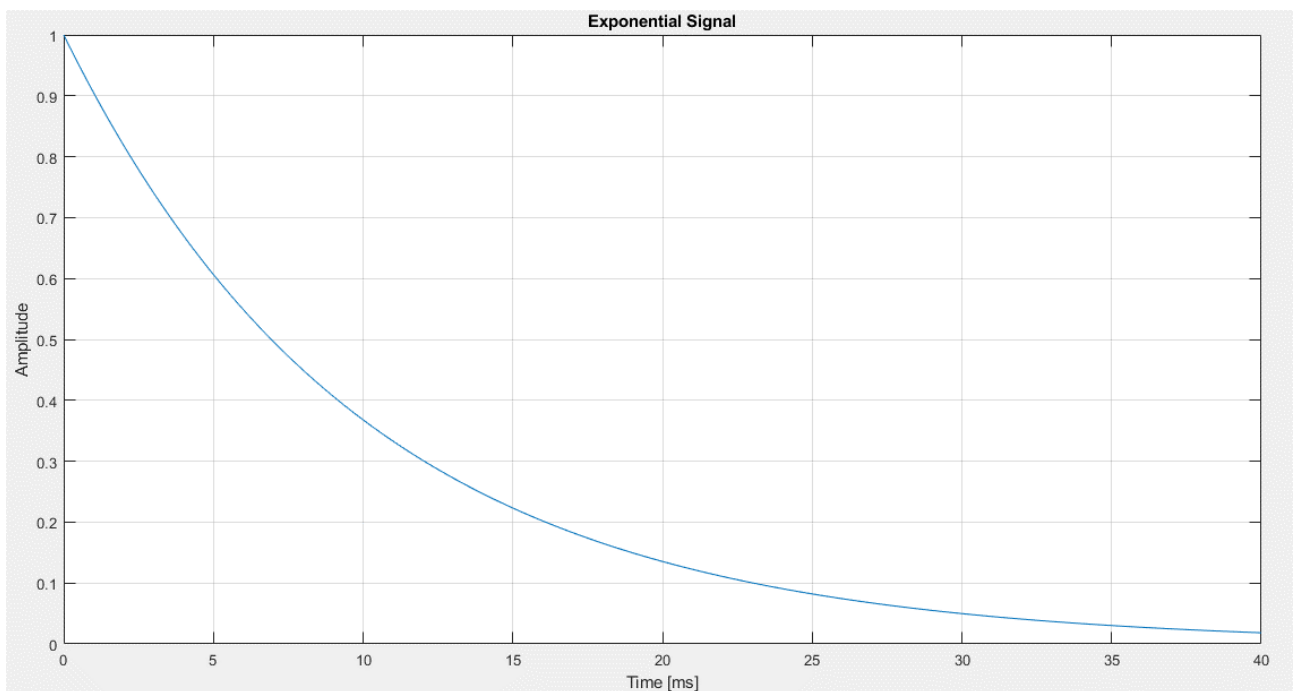


Рисунок 2.1

Завдання

Згенеруйте сигнал, який описується формулою: $x(t) = e^{-0.1t} \sin(0.6t)$ для змінної t , яка змінюється на інтервалі від 0 до 40 мс з кроком 0,1 мс.

Зміни в файлі Exp.m :

```
x = exp(-0.1 * t) .* sin(0.6 * t);
```

Результат роботи програми:

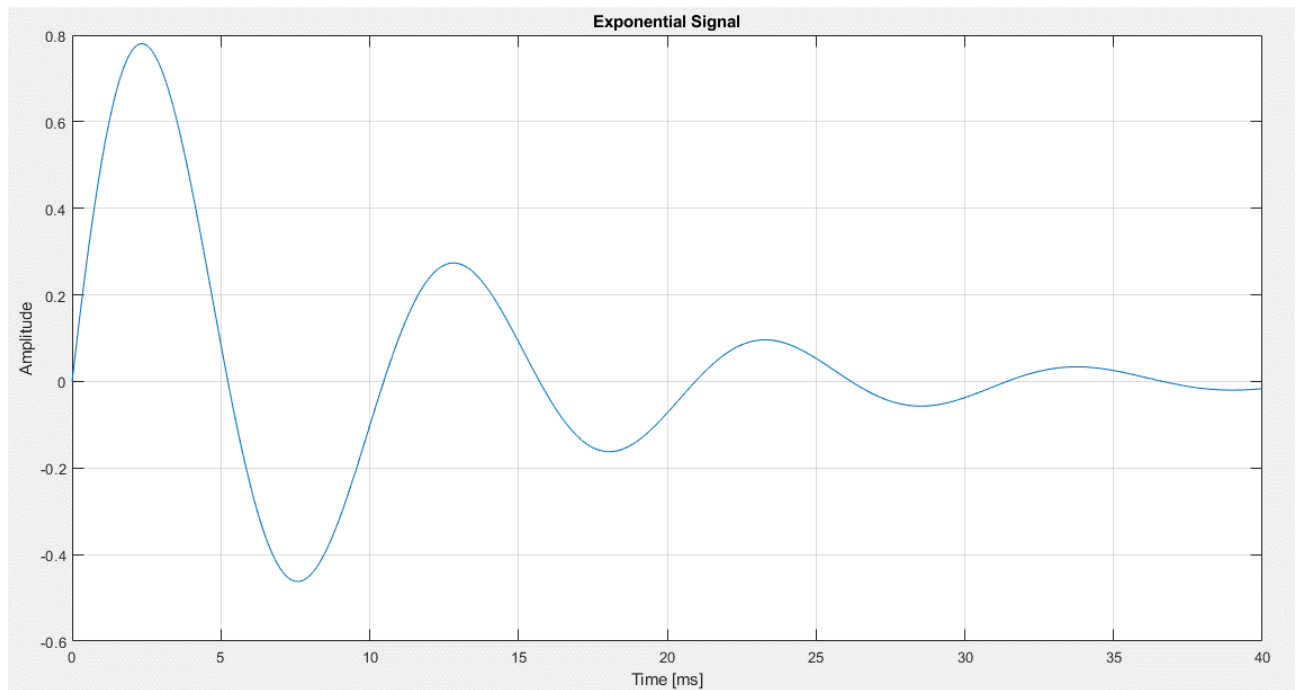


Рисунок 2.2

3. Відкрийте програму, яка генерує дельта-імпульс (Delta.m). Імпульс описується наступною формулою:

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

```
% Generating 64 Samples of a unit impulse signal
N = 64;                                % Define the number of samples
n = -(N / 2) : ((N / 2) - 1);         % Define a vector of sample numbers
x = zeros(1, N);                       % Define a vector of zeros
x((N / 2) + 1) = 1.0;                  % Make the first sample to be 1 (i.e.at t = 0)
plot(n, x);                            % Plot the impulse
grid;
title('A Unit Impulse Signal');
xlabel('Sample Number');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на рисунку:

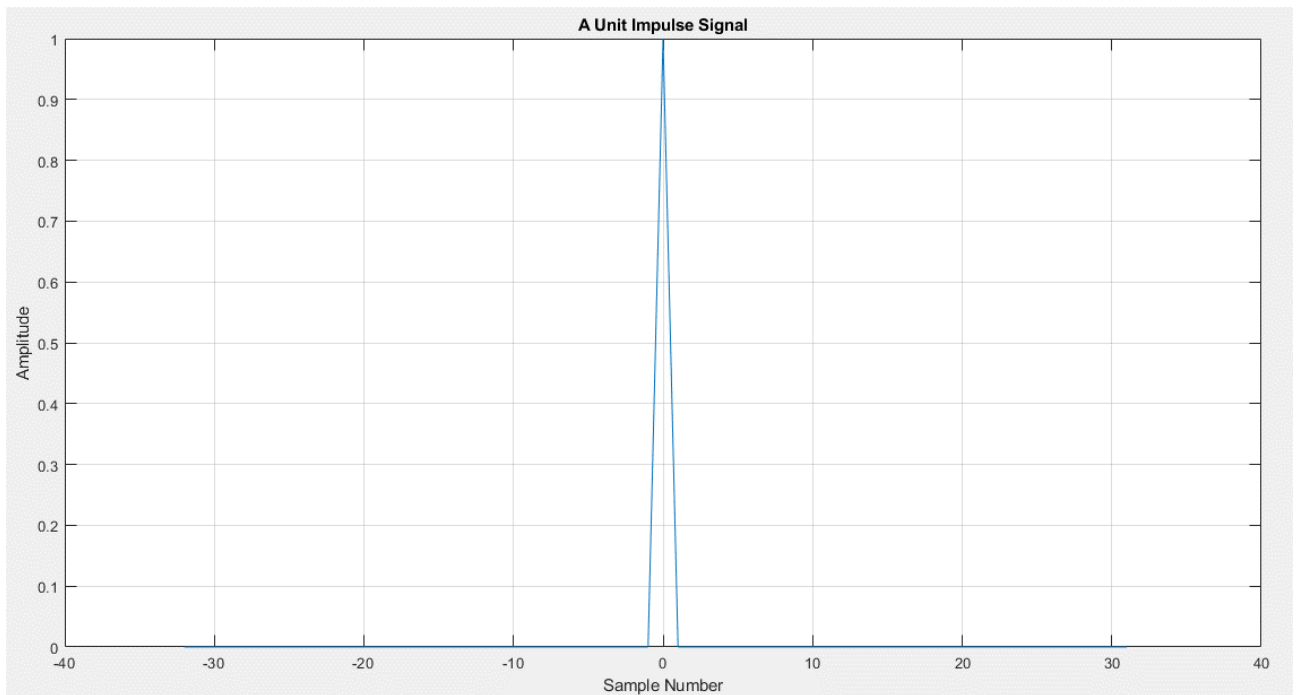


Рисунок 3.1

Завдання

На інтервалі, який складається із 40 відліків (від -20 до 19), згенеруйте наступні сигнали:

1) $x(n) = 2\delta(n - 10)$;

```
function my_Delta()  
n = -20 : 19;  
x = 2 * d(n - 10);  
plot(n, x);  
grid;  
end  
  
function y = d(n)  
y = zeros(1, size(n, 2));  
y(size(n, 2) / 2 + 1) = 1.0;  
end
```

Результат роботи даної програми:

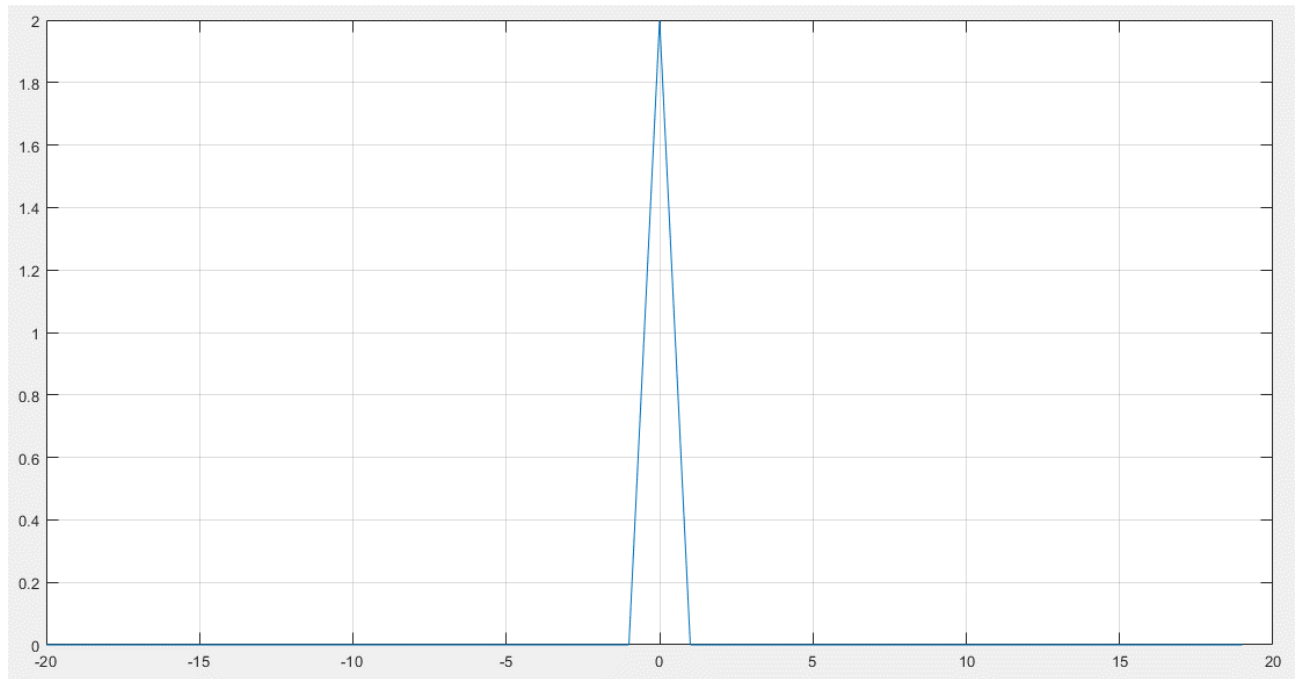


Рисунок 3.2

2) $x(n) = 5\delta(n - 10) + 2.5\delta(n - 20);$

Зміни в файлі my_Delta.m :

```
x = 5 * d(n - 10) + 2.5 * d(n - 20);
```

Результат роботи програми:

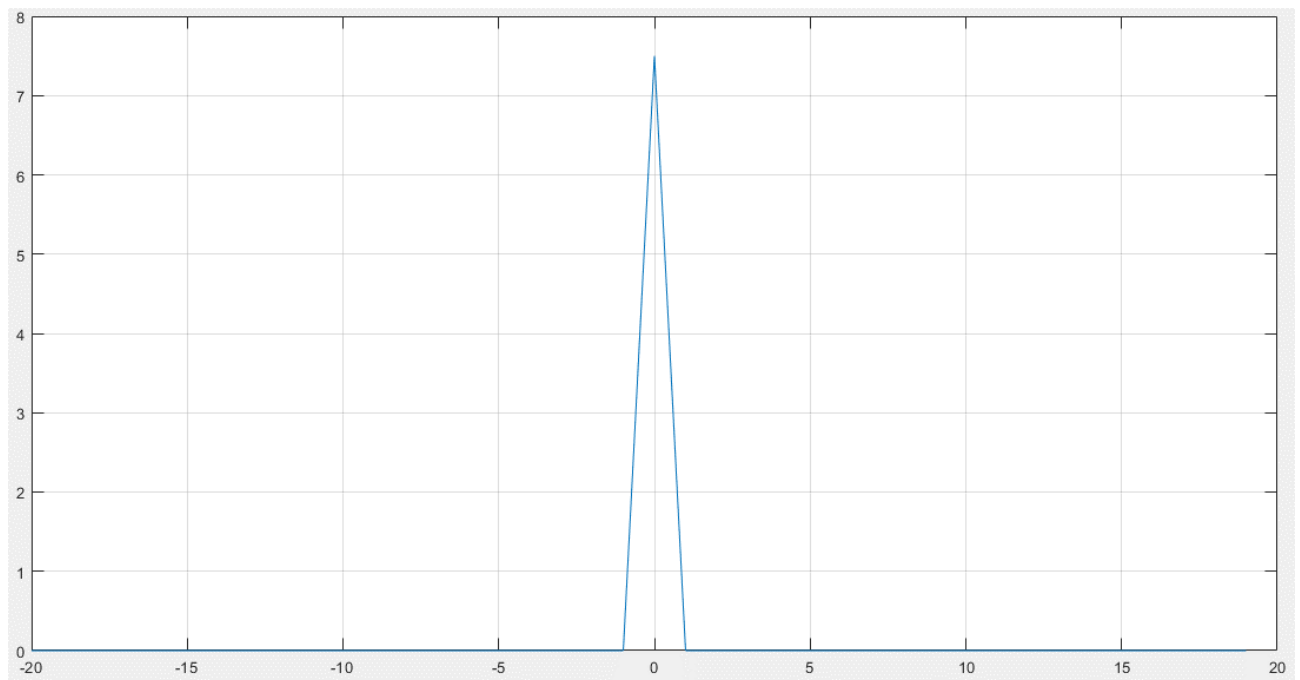


Рисунок 3.3

4. Відкрийте програму, яка генерує сигнал типу «скачок» (Step.m). Цей сигнал описується наступною формулою:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

```
% Generates 40 samples of a unit step signal, u(n)
N = 40; % Define the number of samples
n = -20 : 20; % Define a suitable discrete time axis
u = [zeros(1, (N / 2) + 1), ones(1, (N / 2))]; % Generate the signal
plot(n, u); % Plot the signal
axis([-20, 20, -0.5, 1.5]); % Scale axis
grid;
title('A Unit Step Signal');
xlabel('Sample Number');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на рисунку:

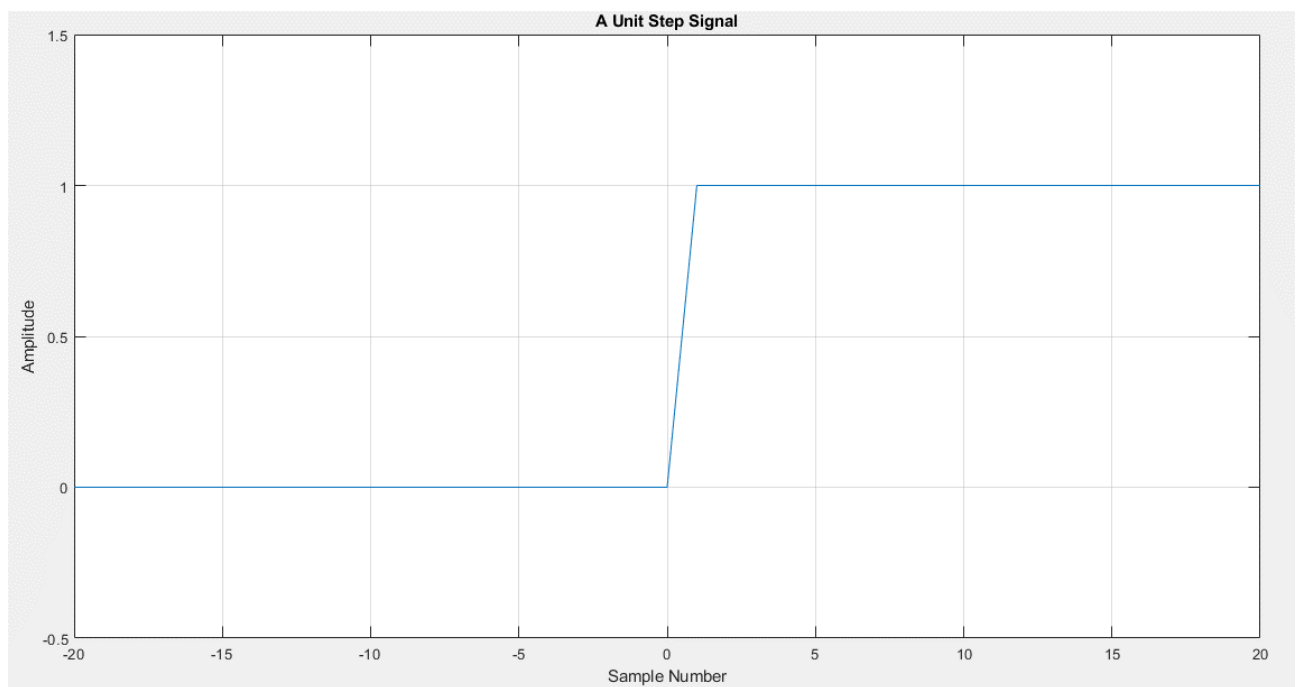


Рисунок 4.1

Завдання

На інтервалі, що складається з 41 відліку (від -20 до 20), згенеруйте наступні сигнали:

1) $x(n) = u(n) - u(n - 1)$;

```

function my_Step()
n = -20 : 20;
x = u(n) - u(n - 1);
plot(n, x);
grid;
end

function y = u(n)
y = zeros(1, size(n, 2));
for i = 1 : size(n, 2)
    if(n(i) >= 0)
        y(i) = 1;
    end
end
end
end

```

Результат роботи даної програми:

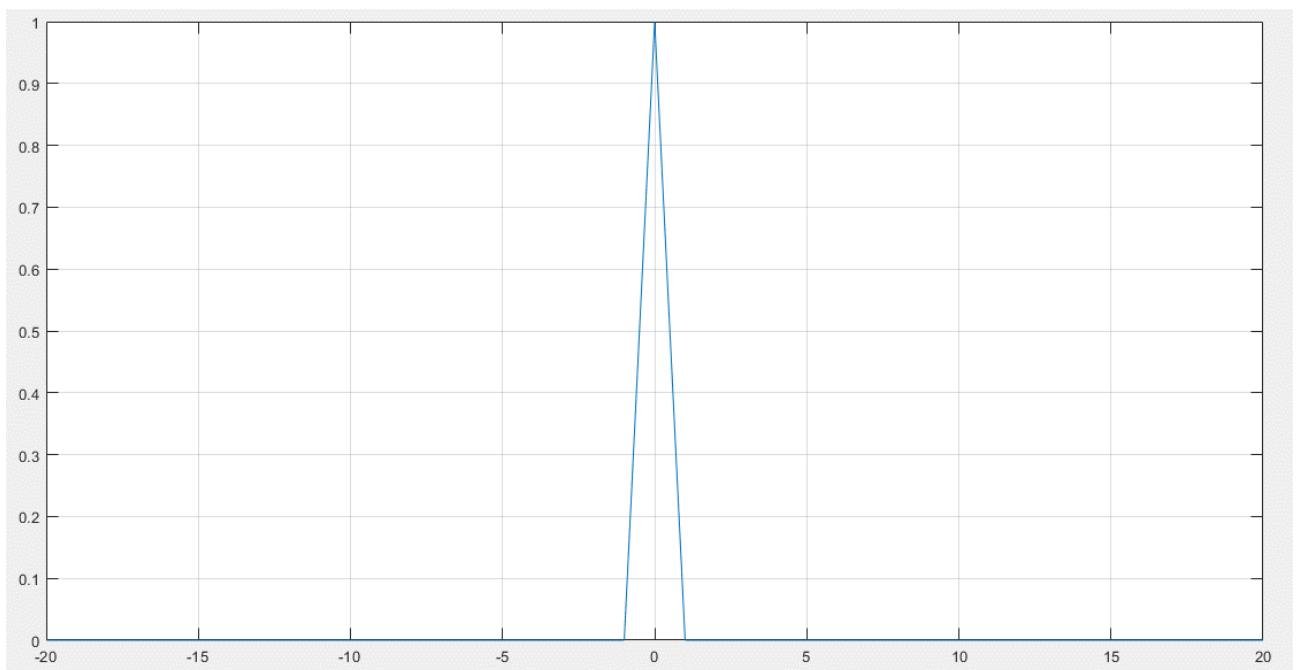


Рисунок 4.2

2) $g(n) = u(n - 1) - u(n - 5);$

Зміни в файлі my_Step.m :

```

x = u(n - 1) - u(n - 5);

```

Результат роботи даної програми:

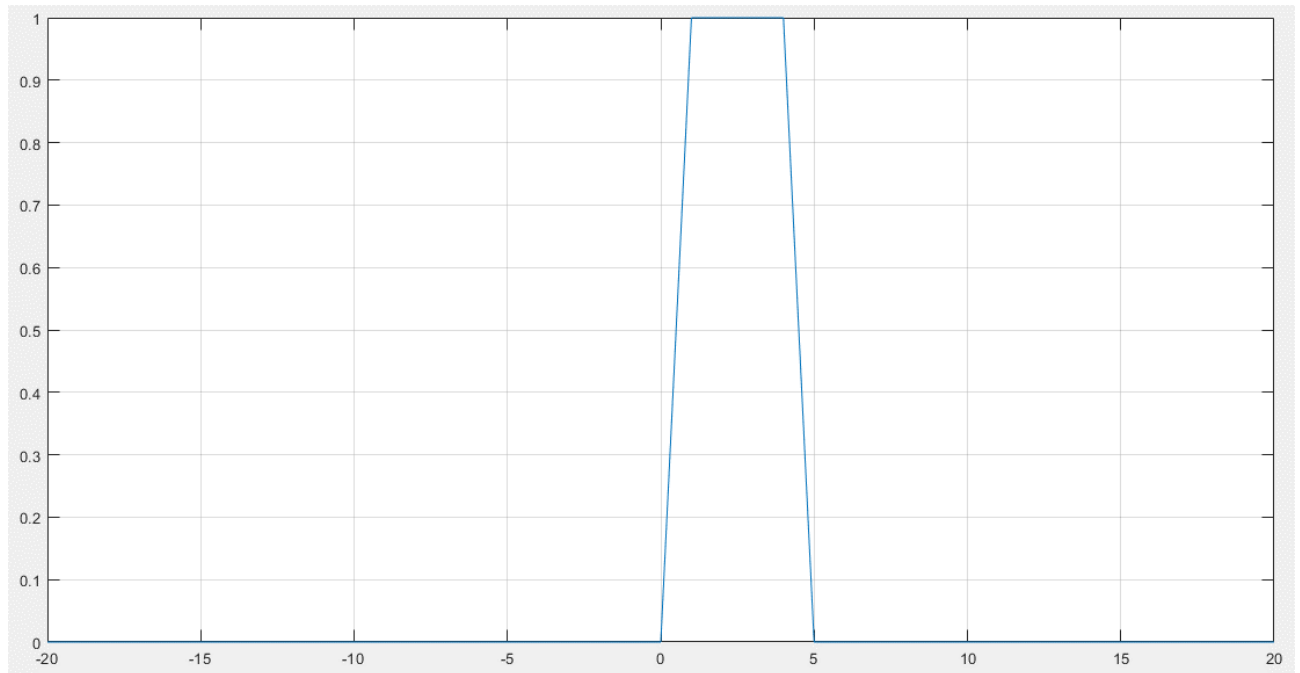


Рисунок 4.3

5. Відкрийте програму, яка генерує сигнал (Sinc.m). Сигнал задається на інтервалі від -5 до 5 мс і описується формулою:

$$x(t) = \sin \frac{\pi * x}{\pi * x}$$

```
% Generating the signal x(t) = sin(pi * x) / (pi * x)
t = linspace(-5, 5);    % Generating time array
x = sinc(t);            % Our function
plot(t, x);
grid;
title('Sinc Function');
xlabel('Time [ms]');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на малюнку:

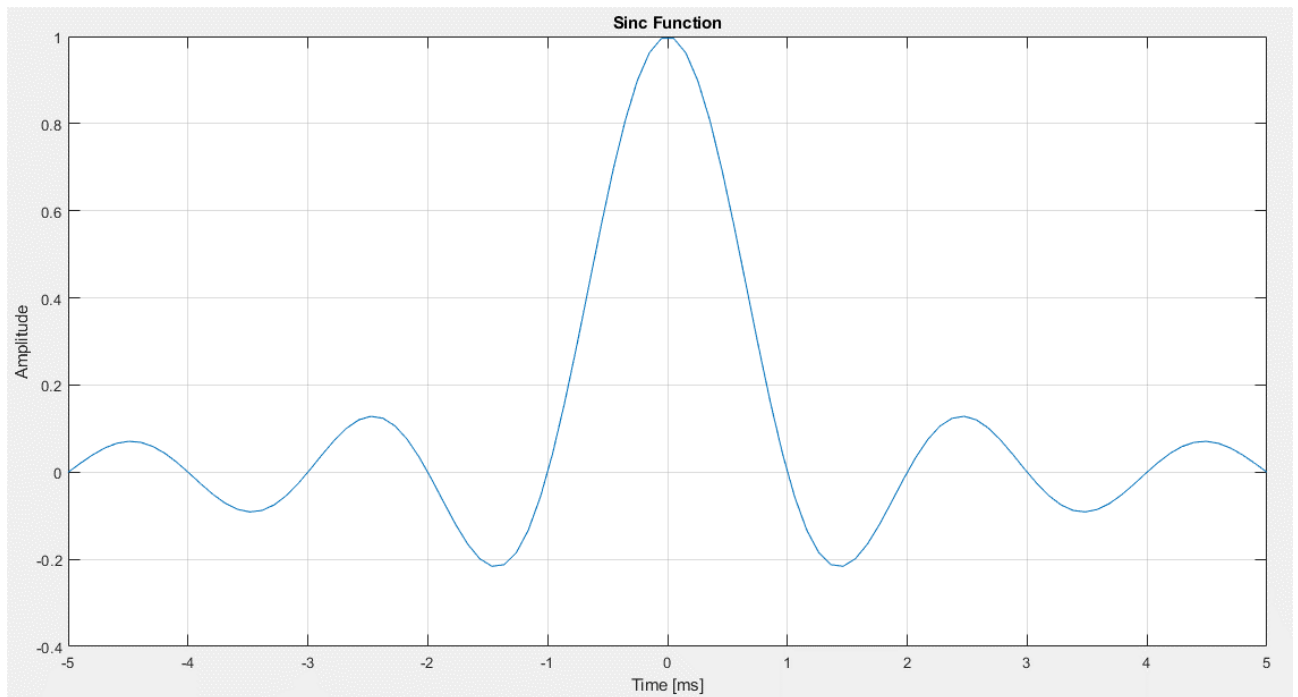


Рисунок 5.1

Завдання

Для даного сигналу змініть часовий інтервал:

- 1) від -10 до 10 мс;

```
t = linspace(-10, 10); % Generating time array
```

Результат роботи програми:

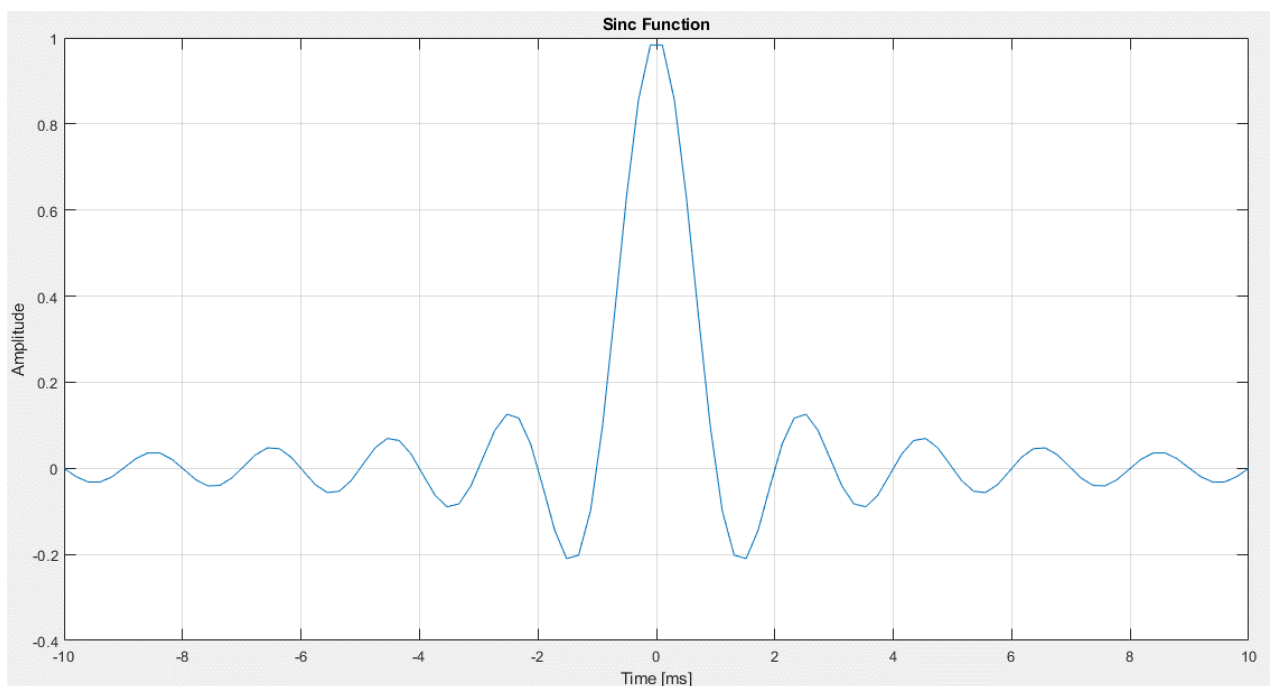


Рисунок 5.2

2) від 0 до 20 мс.

```
t = linspace(0, 20);    % Generating time array
```

Результат роботи програми:

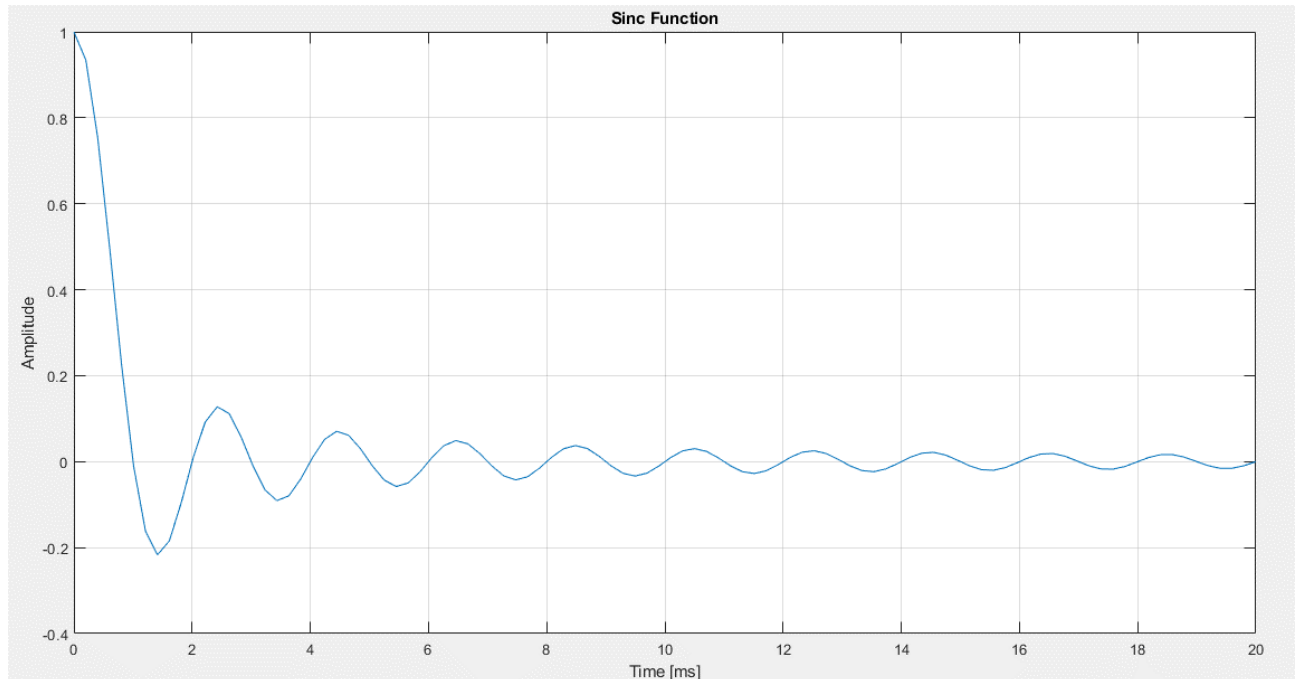


Рисунок 5.3

6. Відкрийте програму, яка генерує радіоімпульс (Gauss.m). Несуча частота радіоімпульсу - 4кГц, тривалість радіоімпульсу - 20мс (від -10 до 10 мс), форма обвідної амплітуди радіоімпульсу - гауссіан, відношення тривалості згенерованого сигналу до тривалості радіоімпульсу - 0,1 (заповнення).

```
% Generating the radiopulse
fs = 16e3;                      % Sampling frequency 16 kHz
fc = 4e3;                       % Carrier frequency of radiopulse 4 kHz
t = -10e-3 : 1 / fs : 10e-3;    % Width of radiopulse from -10 to 10 ms
bw = 0.1;                       % BandWidth of radiopulse
x = gausspuls(t, fc, bw);        % Our radiopulse
plot(t, x);
grid;
title('Radiopulse Function');
xlabel('Time [s]');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на рисунку:

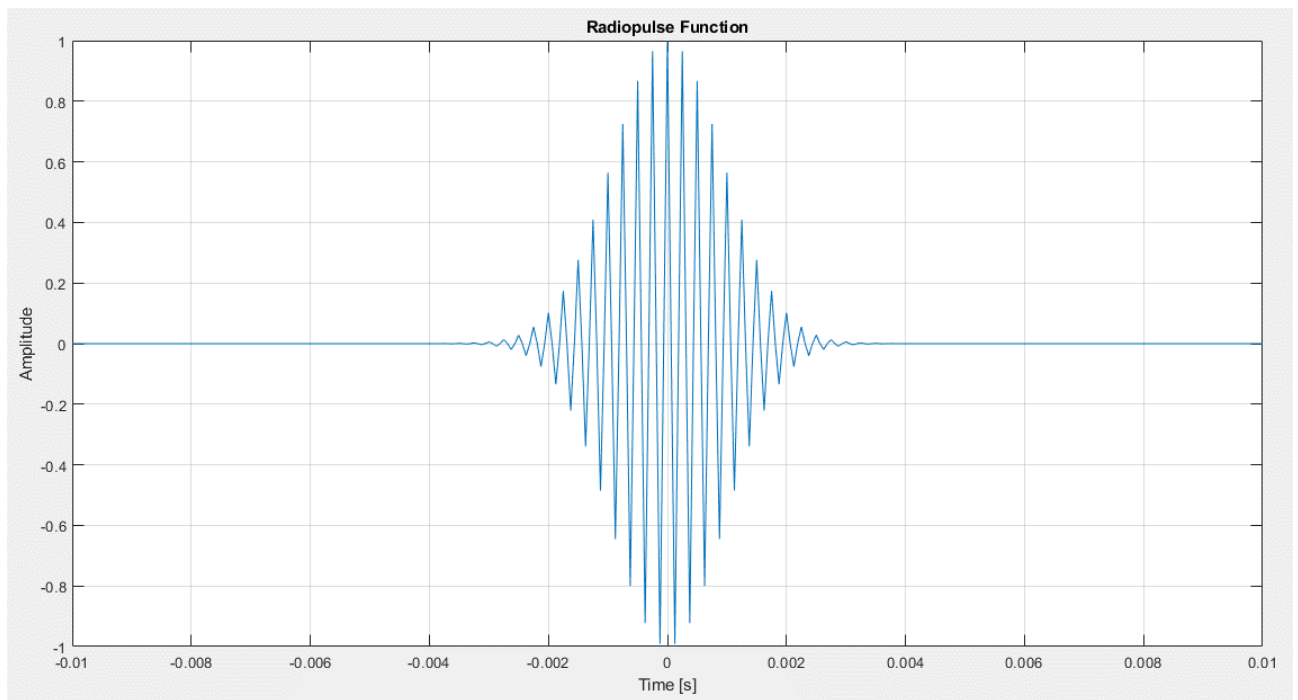


Рисунок 6.1

Завдання

Для даного сигналу змініть:

- 1) відношення тривалості згенерованого сигналу до тривалості радіоімпульсу;

```
bw = 0.05; % BandWidth of radiopulse
```

- 2) опорну частоту радіоімпульсу.

```
fc = 2.5e3; % Carrier frequency of radiopulse 3 kHz
```

Результат роботи даної програми:

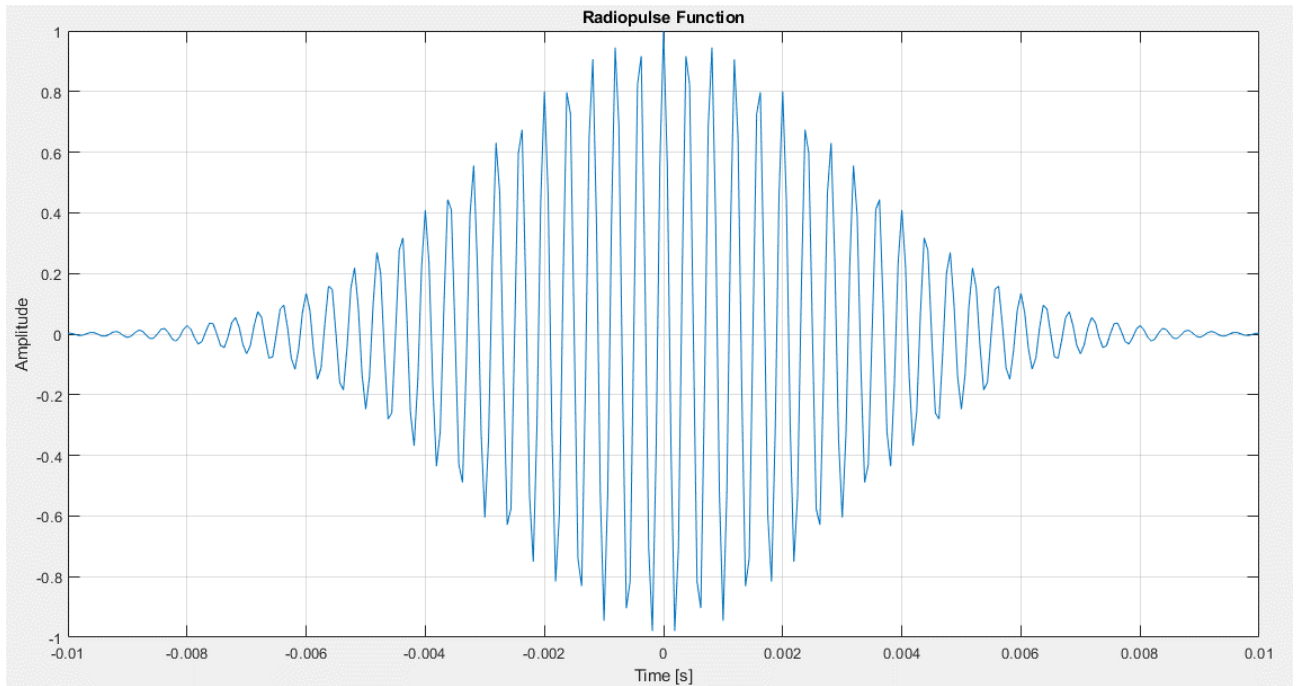


Рисунок 6.2

7. Відкрийте програму, яка генерує послідовність радіоімпульсів з затухаючої амплітудою (Train.m). Несуча частота радіоімпульсу – 10 кГц, заповнення радіоімпульсу – 50%, частота повторення імпульсів – 1 кГц, частота дискретизації – 50 кГц. Амплітуда кожного наступного імпульсу послаблюється в 0,8 разів відносно попереднього.

```
% Generating train of pulse

% Time vector for sample frequency 50 kHz
t = 0 : 1 / 50e3 : 10e-3;

% Delay frequency for pulse 1 kHz. Amplitude attenuate by 0.8
d = [0 : 1 / 1e3 : 10e-3; 0.8 .^ (0 : 10)]';

% Generating Gauspuls with frequency 10 kHz and bandwidth 50%
y = pulstran(t, d, 'gauspuls', 10e3, 0.5);

plot(t, y)
grid;
title('PulseTrain Function');
xlabel('Time [ms]');
ylabel('Amplitude');
```

Результатом роботи даної програми буде сигнал, який представлений на малюнку:

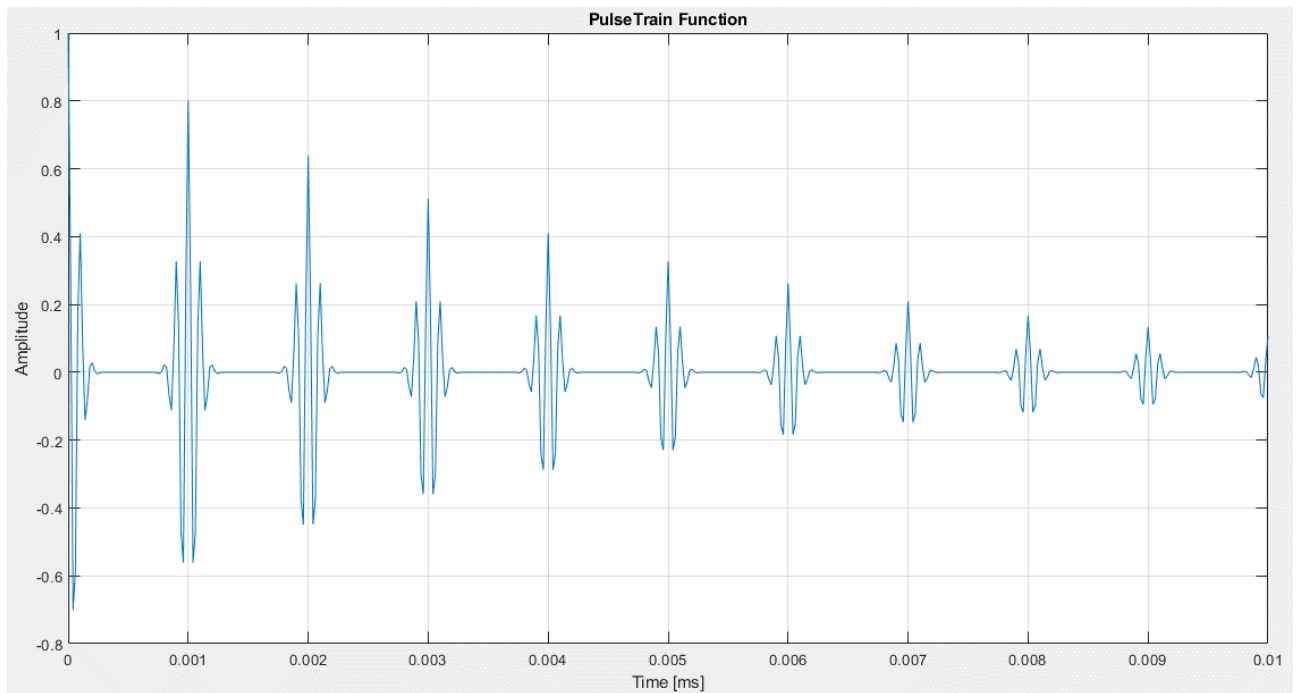


Рисунок 7.1

Завдання

Створіть програму, яка генерує послідовність з п'яти імпульсів $\text{sinc} \frac{x}{x}$, розташованих рівномірно на інтервалі відліків від 0 до 1000.

```
t = 0 : 0.1 : 1000;  
d = [0 : 200 : 1000]';  
y = pulstran(t, d, 'sinc');  
plot(t,y)  
grid;
```

Результат роботи даної програми:

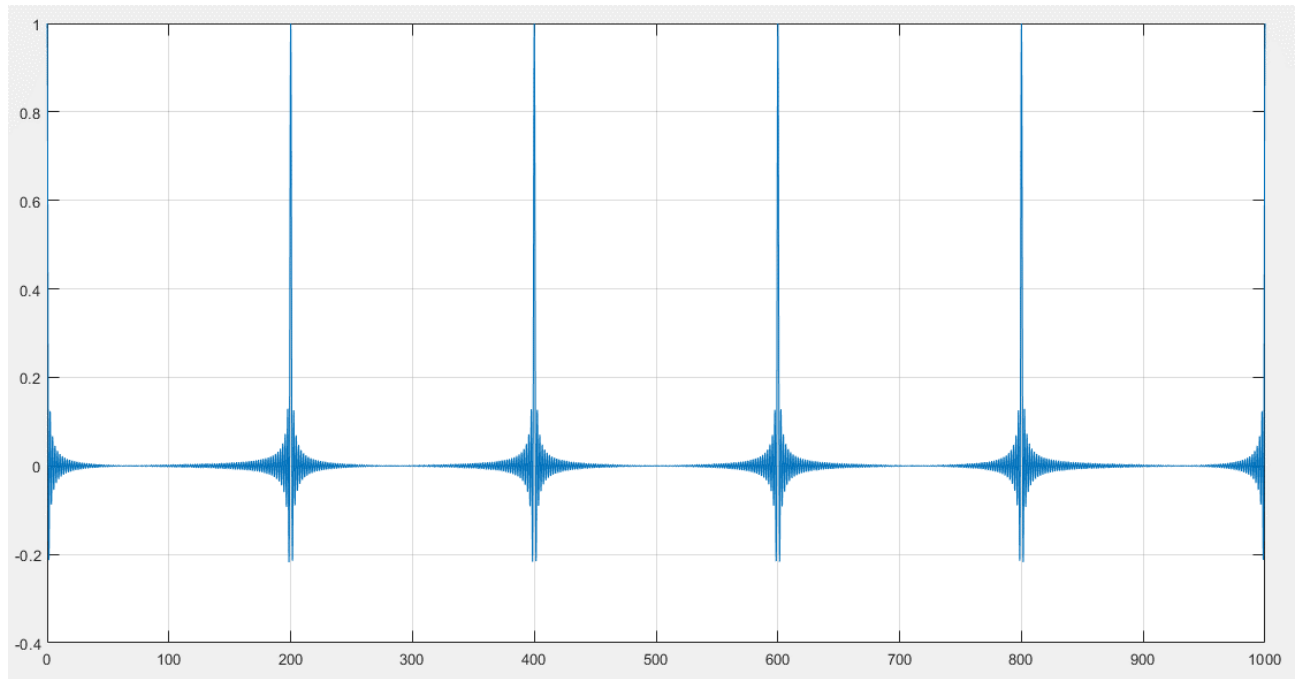


Рисунок 7.2

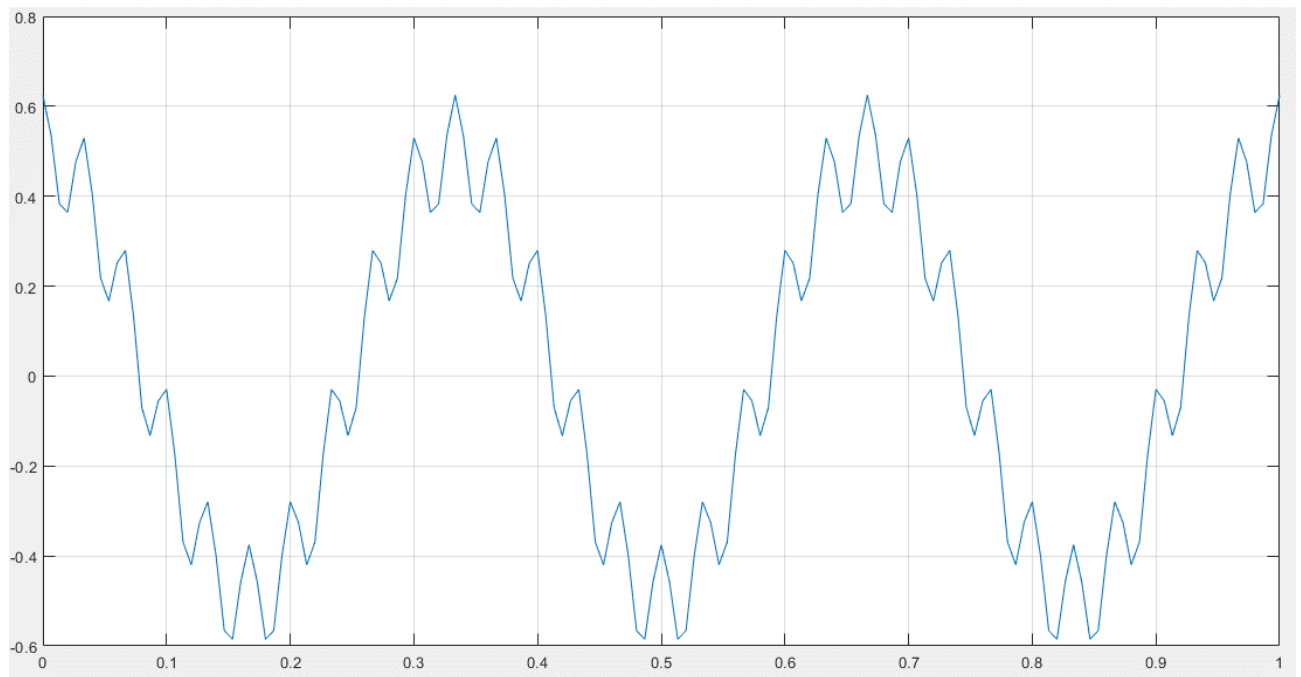
Завдання

1. Опрацюйте основні приклади, викладені вище, в системі MatLab.
2. Змініть параметри сигналів, які генеруються в М-файлах із прикладів № 1 і № 2. Для кожного з прикладів створіть по два варіанти власної реалізації.

Приклад 1:

```
fs = 150;  
t = 0 : 1 / fs : 1;  
x = 0.5 * cos(2 * pi * t * 3) + 0.125 * cos(2 * pi * t * 180);  
plot(t, x);  
grid on
```

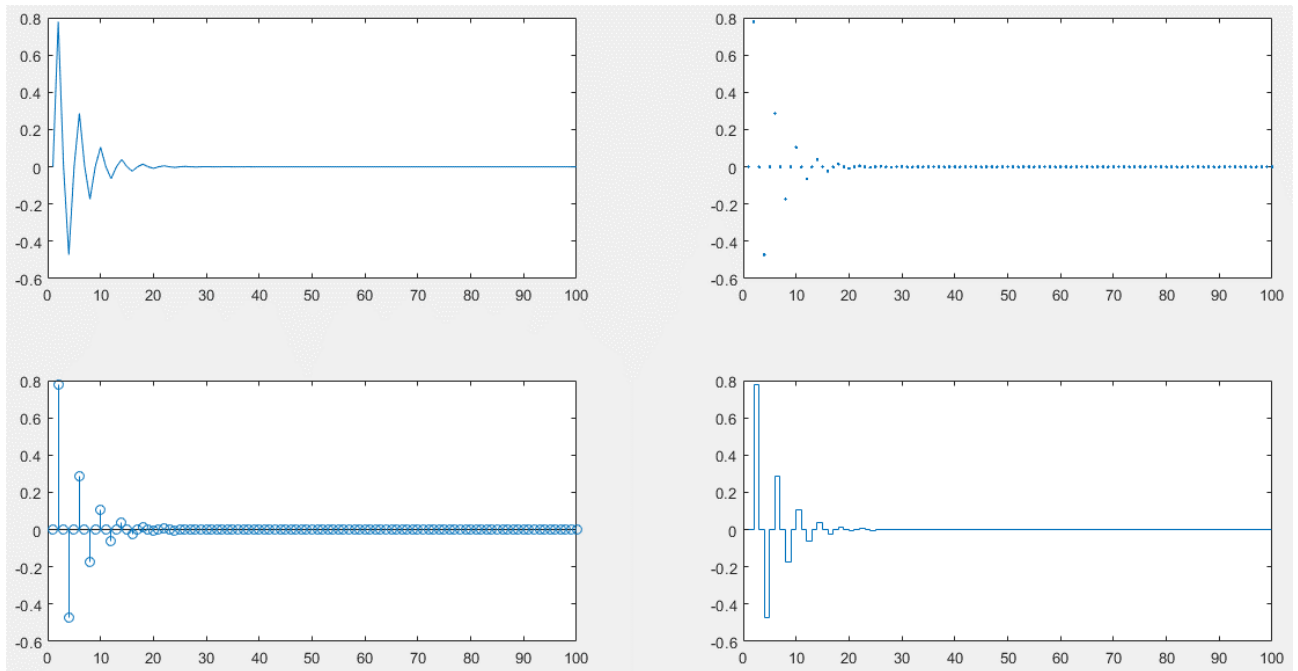
Результат роботи програми:



Приклад 2:

```
Fs = 4e3;  
t = 0 : 1 / Fs : 1;  
t = t';  
A = 1;  
f0 = 1e3;  
alpha = 1e3;  
  
s1 = A * sin(2 * pi * f0 * t);  
s2 = exp(-alpha * t) .* s1;  
  
subplot(2, 2, 1);  
plot(s2(1: 100))  
subplot(2, 2, 2);  
plot(s2(1 : 100), 'r')  
subplot(2, 2, 3);  
stem(s2(1 : 100))  
subplot(2, 2, 4);  
stairs(s2(1 : 100))
```

Результат роботи програми:



3. За допомогою функції `rectpuls` створіть одиночний біполярний імпульс, зображений на рисунку 3.

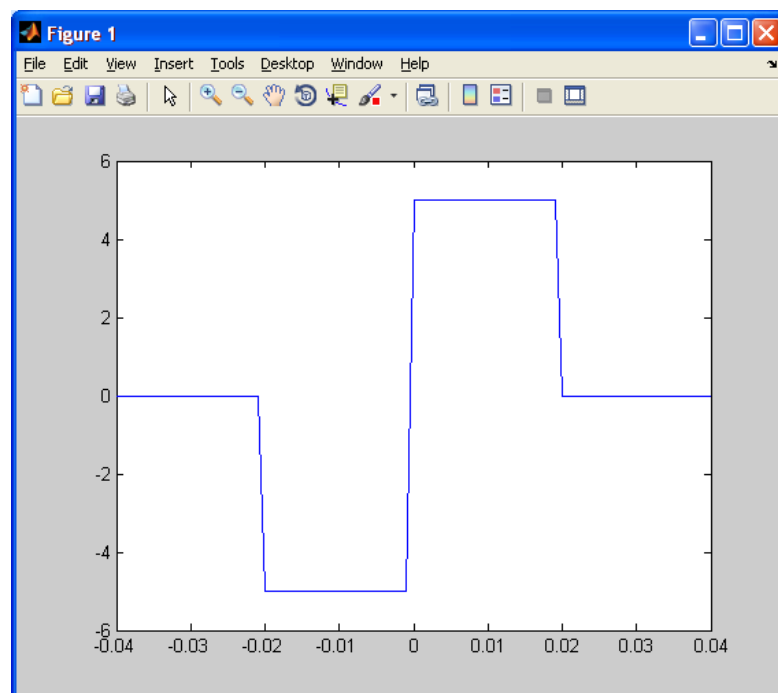
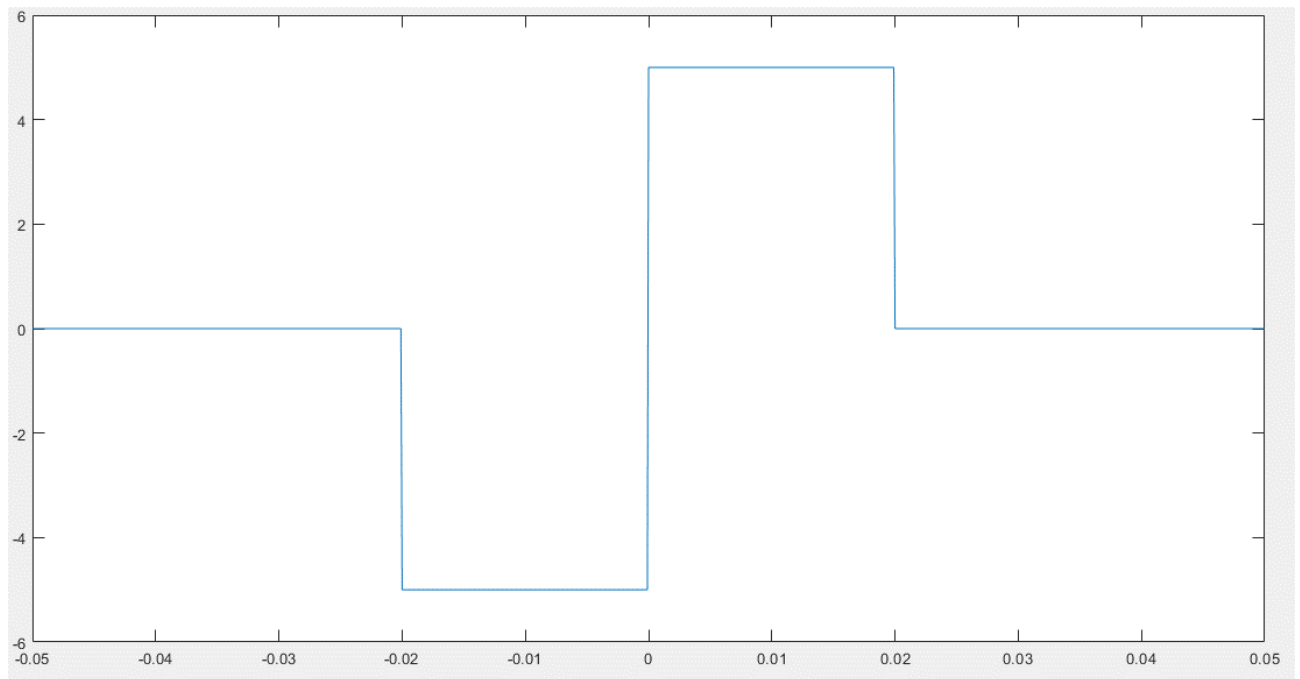


Рисунок 3. Одиночний біполярний імпульс.

```
Fs = 10000;  
t = -0.05 : 1 / Fs : 0.05;  
T = 0.02;  
A = 5;  
  
s = -A * rectpuls(t + T / 2, T) + A * rectpuls(t - T / 2, T);  
  
plot(t, s)  
axis([-0.05, 0.05, -6, 6])
```

Результат роботи програми:



- 4. За допомогою функції `tripuls` створіть одиночний трапецеїдальний імпульс, зображений на рисунку 4.**

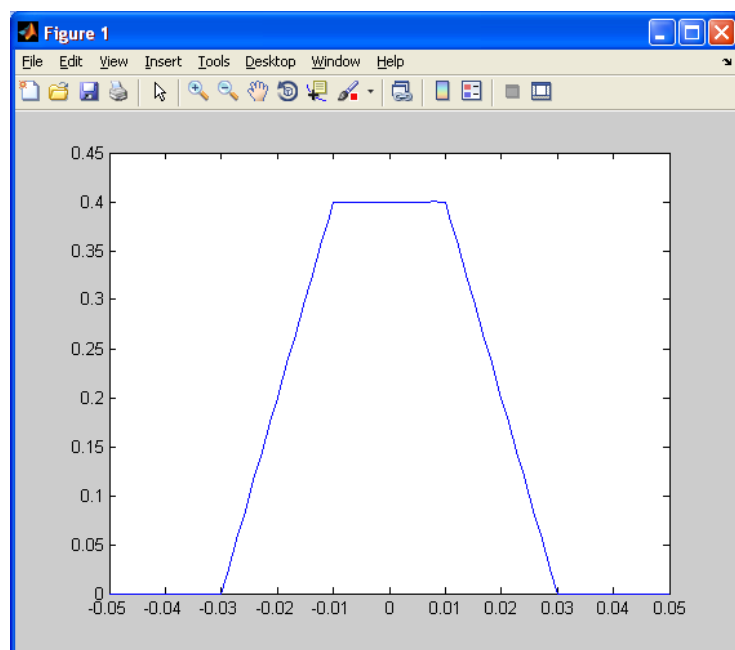


Рисунок 4. Одиночний трапецеїдальний імпульс.


```

Fs = 10e3;
t = -0.1 : 1 / Fs : 0.1;
A = 1;
w = 60e-3;
offset = 0;
offsety = 0.4;

s = A * tripuls(t - offset, w);
for i = 1 : size(s, 2)
    if(s(i) > offsety)
        s(i) = offsety;
    end
end

plot(t, s);
axis([-0.05, 0.05, 0, 0.45])

```

Результат роботи програми:

