МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ УНІВЕРСИТЕТ

Кафедра інформатики та прикладної математики

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни «Моделювання систем»

для студентів спеціальностей 121 «Інженерія програмного забезпечення» та 122 Комп'ютерні науки

Укладачі проф. Колодяжний В.М.,
ст. викладач Козачок Л.М.

2017 рік

ФОРМИРОВАНИЕ И АНАЛИЗ СИГНАЛОВ В СИСТЕМЕ MATLAB

Цель работы. Изучение возможностей MATLAB для формирования анализа сигналов**.**

Задание 1. Изучить материал, изложенный ниже, и повторить примеры и упражнения, приведенные в тексте.

Формирование сигналов

Формирование детерминированных сигналов

Для формирования дискретизированного сигнала, заданного какимлибо выражением, необходимо сначала сформировать вектор дискретных значений времени. Для этого удобно задать значение частоты дискретизации **Fs** (sampling frequency) и использовать обратную величину в качестве шага временного ряда. Пример формирования гармонических колебаний:

```
Fs = 8e3;                   % частота дискретизации 8 кГц
t = 0:1/Fs:0.025;           % 0.025 с дискретных значений времени
A = 2;                      % амплитуда – два вольта
f0 = 1e3;                   % частота 1 кГц
phi = pi/4;                 % начальная фаза 45°
s1=A*cos(2*pi*f0*t+phi);    % гармоническое колебание
alpha1 = 0.1e3;             % скорость затухания
s2 = exp(-alpha1*t) .* s1;  % затухающее гармоническое колебание
subplot(2,1,1);plot(t,s1); subplot(2,1,2);plot(t,s2);
```

На рис. 4.1, приведенном ниже, показаны «осциллограммы» сформированных колебаний.

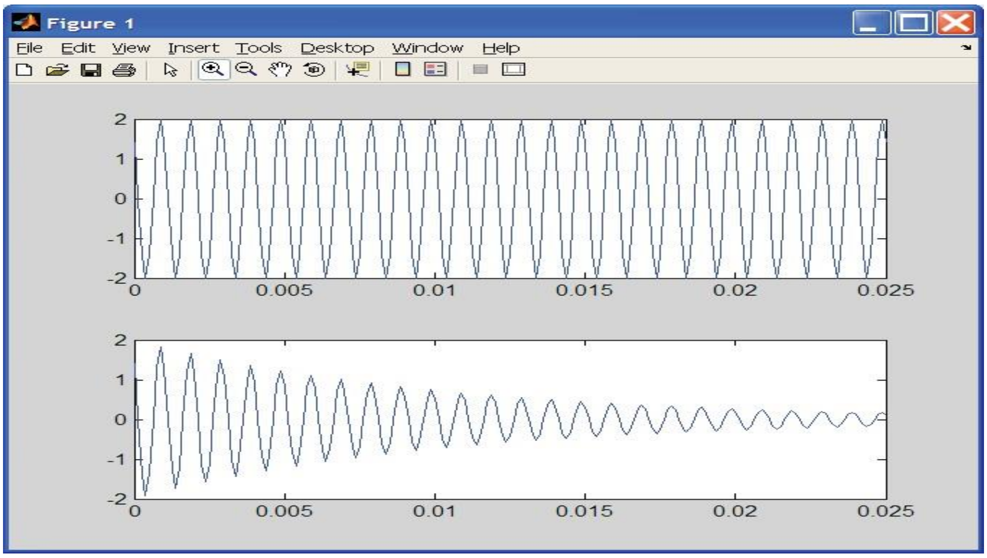

Рис. 4.1

При формировании импульсных сигналов целесообразно использовать операции сравнения, которые возвращают единицу при выполнении неравенства и ноль в противном случае, причем в случае векторного аргумента возвращается вектор результатов сравнения. Сформируем односторонний экспоненциальный импульс (рис. 4.2):

A = 2; Fs = 8e3; t = -
0.005:1/Fs:0.01; alpha = 0.5e3; s
= A * exp(-alpha * t) .* (t >= 0);
stem(t,s);

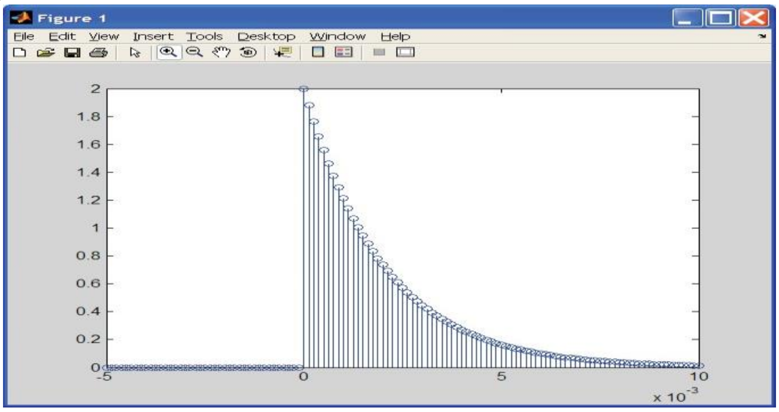

Рис. 4.2

Сформируем прямоугольный импульс, центрированный относительно начала отсчета времени (рис. 4.3):

A = 2;

Fs = 8e3; T= 0.01; t =
-0.01:1/Fs:0.01; s = A
*** (abs(t) <= T/2);**
stem(t,s)

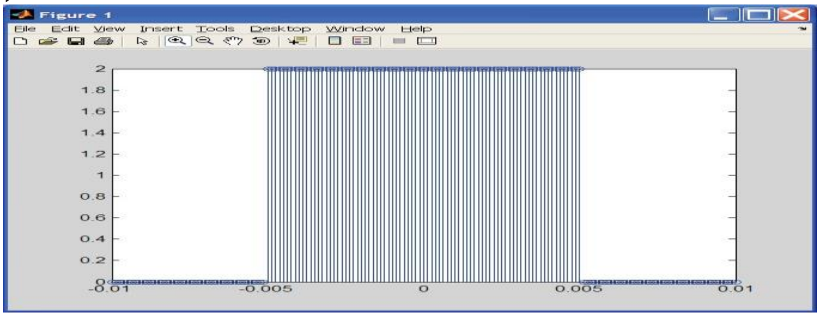

Рис. 4.3

Пример формирования несимметричного треугольного импульса (рис. 4.4):

A = 2;
Fs = 8e3; T= 0.01; t = -
0.01:1/Fs:0.02; s = A * t / T .* (t
>= 0) .* (t <= T); stem(t,s)

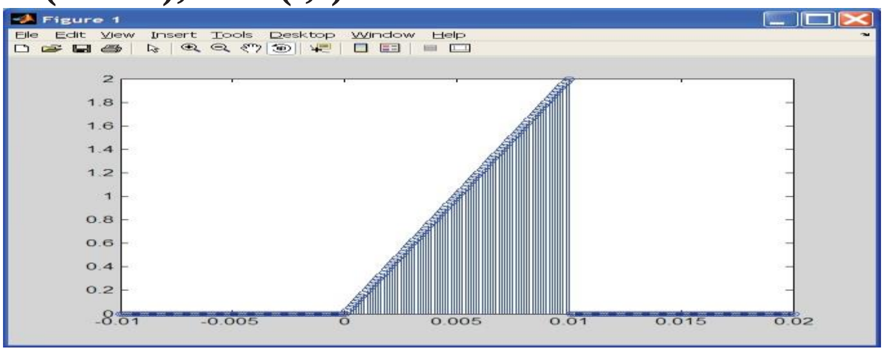

Рис. 4.4

В приведенных примерах следует обратить внимание на использование оператора **.*** для поэлементного перемножения векторов.

В составе системы MATLAB имеется ряд пакетов (**Toolboxes**), которые дополняют ядро системы другими функциями. В пакете **Signal Processing** имеется ряд функций, генерирующих часто встречающиеся на практике импульсные сигналы. В частности, имеются функции **rectpuls** (формирование одиночного прямоугольного импульса), **tripuls** (формирование одиночного треугольного импульса) и **square** (формирование периодической последовательности прямоугольных импульсов). Функция **rectpuls** формирует одиночный прямоугольный импульс с единичной амплитудой: **y = rectpuls(t, width)**. Здесь **t** – вектор значений времени, **width** – ширина (длительность) импульса. Возвращаемый результат **y** – вектор рассчитанных значений сигнала, определяемых по следующей формуле:

$$y = \begin{cases} 1, & -\dfrac{\text{width}}{2} \le t < \dfrac{\text{width}}{2}, \\ 0, & t < -\dfrac{\text{width}}{2},\ t \ge \dfrac{\text{width}}{2}. \end{cases}$$

Сформируем пару разнополярных прямоугольных импульсов, расположенных справа и слева от начала отсчета времени (рис. 4.5).

A = 2;
Fs = 8e3; T= 0.01; t = -0.02:1/Fs:0.02; s = - A* rectpuls(t+T/2, T) + A * rectpuls(t-T/2, T); stem(t, s); ylim([-3 3]) grid on

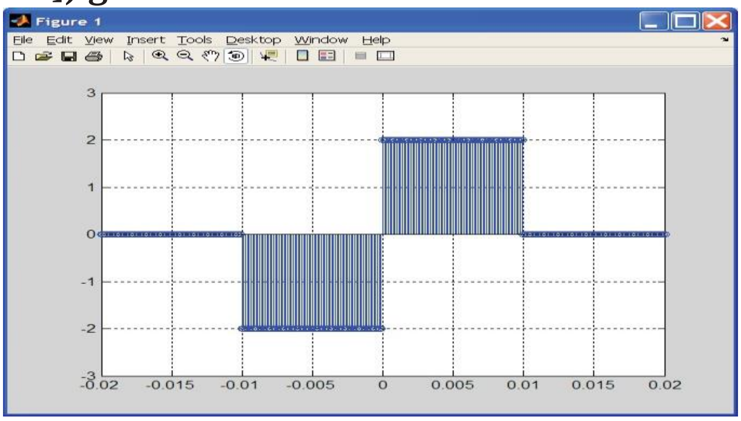

Рис. 4.5

Построим с помощью функции **tripuls** пару разнополярных треугольных импульсов (рис. 4.6).

s = - A* tripuls(t+T/2, T) + A * tripuls(t-T/2, T); stem(t, s); ylim([-3 3]) grid on

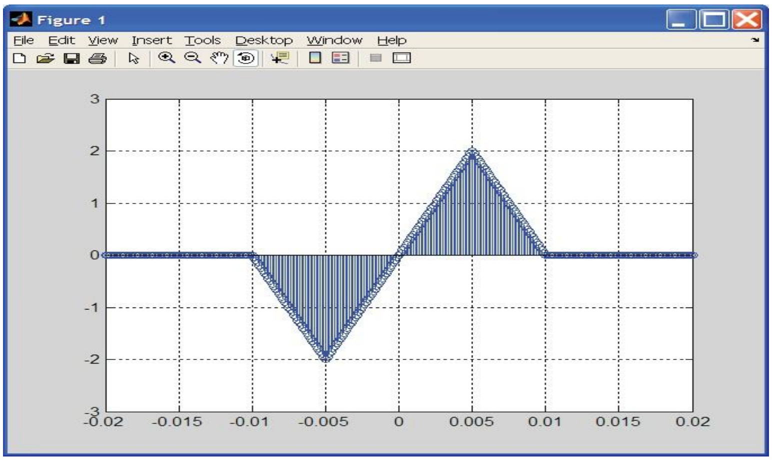

Рис. 4.6

Для формирования последовательности прямоугольных импульсов с периодом **T** служит функция **square: y=square(2*pi*t/T,duty)**. Параметр **duty** – это отношение длительности импульса к периоду (в процентах). По умолчанию значение параметра равно 50, то есть генерируется меандр, в данном случае функция аналогична функции **sin(x)**, но вместо синусоиды формируется импульсная последовательность. Сформируем последовательность однополярных прямоугольных импульсов (рис. 4.7):

A = 2; % амплитуда
Fs = 8e3; % частота дискретизации
t = -0.02:1/Fs:0.02; % дискретное время **f0 = 100;**
% частота следования импульсов
tau = 2.5e-3; % длительность импульсов **s2 =0.5*A*(square(2*pi*t*f0, f0*tau*100)+1); stem(t, s2); ylim([0 2.5]); grid on**

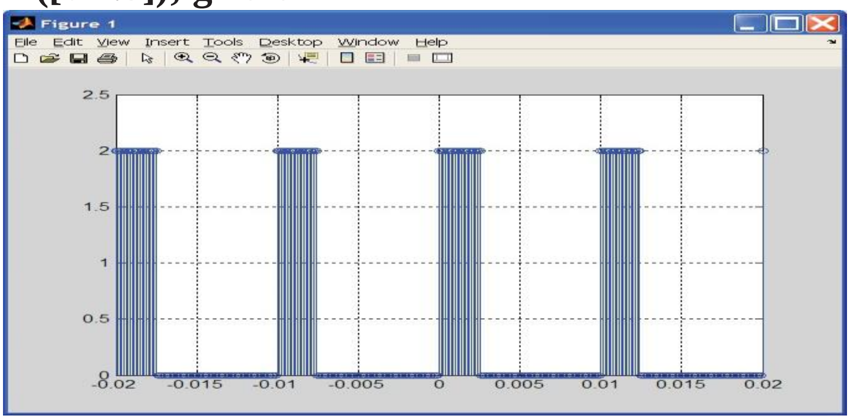

Рис. 4.7

Необходимо отметить, что частоту дискретизации следует выбирать так, чтобы количество отсчетов на интервале длительности импульса было достаточно большим, так как из-за погрешности вычислений число отсчетов на этом интервале может отличаться от требуемого на единицу.

Формирование случайных сигналов

Для генерации случайных чисел служат функции **rand(m, n)** (равномерное распределение на интервале от нуля до единицы) и **randn(m, n)** (нормальное распределение с нулевым математическим ожиданием и единичным среднеквадратическим отклонением). Здесь **m, n** – число строк и столбцов матрицы случайных чисел. Если функции заданы в виде: **rand(size(A)), randn(size(A))**, где **A** – массив, то генерируется массив случайных чисел с размерами, равными размерам массива **A**. Для генерации случайных чисел предназначены также средства пакета расширения **Statistics**, которые поддерживают множество различных законов распределения вероятностей.

Для оценки вида закона распределения вероятностей значений случайного процесса используется гистограмма. Вызов соответствующей функции имеет следующий синтаксис **N = hist(Y, M)**. Здесь **Y** – вектор сигнала; **M** – число столбцов гистограммы (по умолчанию M =10). **N** – вектор, который показывает, сколько элементов содержится в каждом столбце гистограммы. Если выходной параметр функции **hist** не указывается, то строится гистограмма. Построим гистограмму значений случайного процесса с равномерным распределением вероятностей (рис. 4.8):

```
Fs = 8e3; t
= 0:1/Fs:3;
x5=rand(size(t)); % дискрет. белый шум с равномер.
распределением hist(x5,100);
```

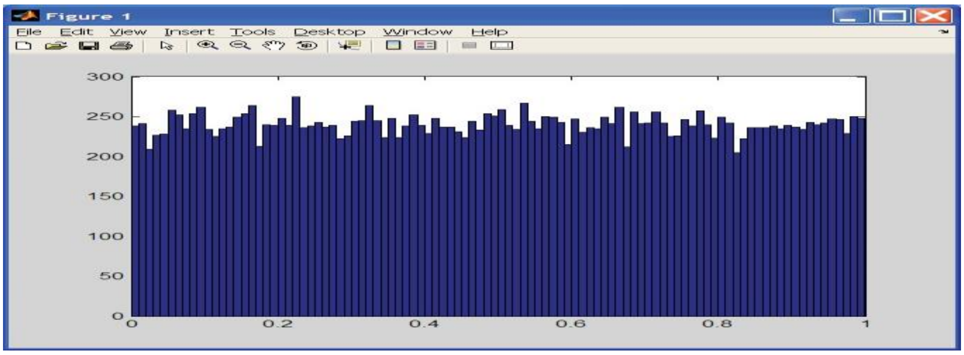

Рис. 4.8

Аналогично строим гистограмму для случайного процесса с нормальным распределением (рис. 4.9):

Fs = 8e3; t
= 0:1/Fs:3;
x6 = randn(size(t)); % дискрет. белый шум с норм. распределением **hist(x6,100);**

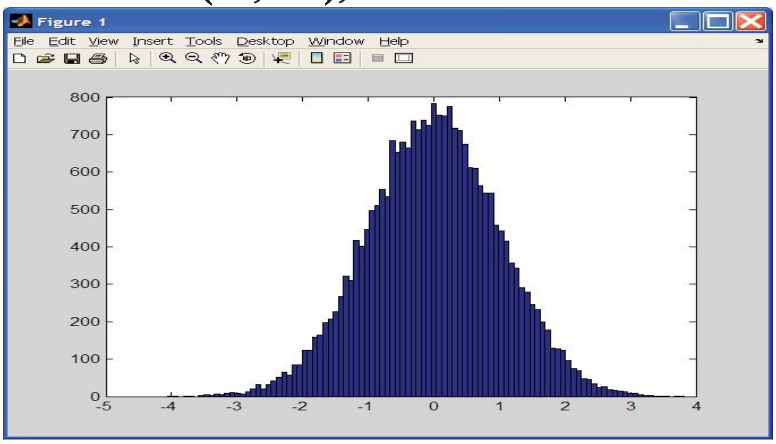

Рис. 4.9

В задачах, связанных с обработкой сигналов, для генерации дискретного нормального белого шума удобнее использовать функцию **wgn** (white Gaussian noise) пакета **Communications**, поскольку она позволяет в явном виде задавать уровень генерируемого шума. Синтаксис вызова функции следующий: **y=wgn(m,n,p,imp,state,’powertype’,**
’outputtype’);

Здесь **m** и **n** – как и ранее, размеры генерируемой матрицы, а **p** – мощность генерируемого шума в единицах, задаваемых параметром **'powertype'** (по умолчанию – в децибелах). Остальные параметры являются необязательными и имеют значения по умолчанию.

Параметр **imp** задает импеданс нагрузки в омах (предполагается, что генерируются отсчеты случайного ***напряжения*** на этой нагрузке). По умолчанию используется импеданс нагрузки, равный 1 Ом.

Целочисленный параметр **state** позволяет принудительно задавать начальное состояние генератора гауссовских случайных чисел (функция ***randn***). По умолчанию используется текущее состояние.

Строковый параметр **‘powertype’** задает единицы измерения мощности, использованные при указании параметра **p**. Возможны следующие значения:

– **'dBW'** – мощность **p** задается в децибелах, значению 0 дБ соответствует мощность 1 Вт;

– **' dBm'** – мощность p задается в децибелах, значению 0 дБ соответствует мощность, равная 10^{-3} Вт;

– **'linear'** – мощность **p** задается в ваттах, дисперсия генерируемого шума равна **p*imp**.

Строковый параметр **'outputtype'** позволяет задавать генерацию вещественного или комплексного шума. Возможны значения **'real'** (вещественный шум; генерируется по умолчанию) и **'complex'** (комплексный шум). Если генерируется комплексный шум, его вещественная и мнимая части имеют мощности p/2.

Пример генерации матрицы 1x **length(t)** вещественного шума мощностью 3 Вт на нагрузке 75 Ом:

```
Fs = 8e3; t = 0:1/Fs:3; y = wgn(1,
length(t), 3, 75, ‘linear’, ‘real’); plot(t,y) fi
gure hist(y,100);
```

Спектральный анализ

Прямое и обратное дискретное преобразование Фурье (ДПФ)

ДПФ является основой спектрального анализа сигналов. Инструкция **y = fft(x)** – вычисляет прямое ДПФ для вектора **x**; если **x** – матрица, преобразование производится для каждого ее столбца по отдельности; **y=fft(x, N)** – предварительно приводит исходные данные к размеру **N**, урезая их или дополняя нулями. Инструкции **x = ifft(y)** и **x = = ifft(y, N)** – аналогичные варианты вызова для функции обратного ДПФ.

Для ускорения вычислений используется быстрое преобразование Фурье (БПФ). Пример определения БПФ для гармонического колебания (рис. 4.10):

```
Fs = 8e3;
f0=100; t =
0:1/Fs:0.01;
x = sqrt(2)*sin(2*pi*f0*t);
F=fft(x); stem(abs(F))
```

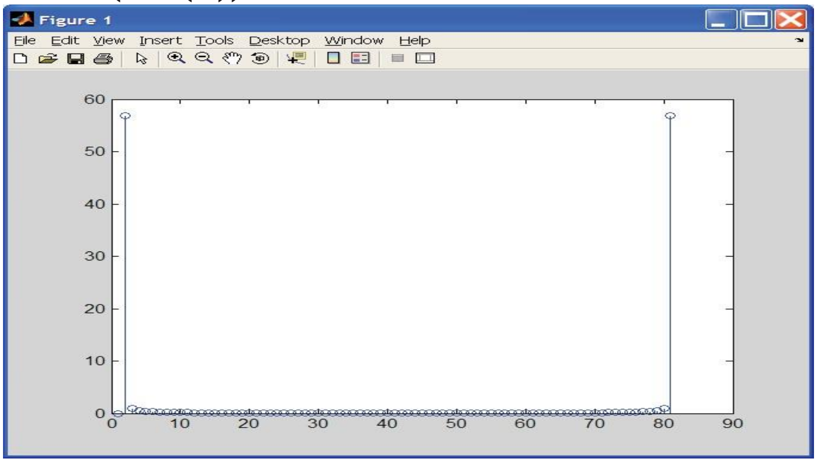

Рис. 4.10

При спектральном анализе рассматривается фрагмент (сегмент) сигнала на некотором интервале времени. Для выделения сегмента из всего сигнала используются весовые функции (временные окна). Использование весовых функций повышает точность спектрального анализа. Система MATLAB содержит (в пакете **Signal Processing**) целый ряд стандартных весовых функций.

Все весовые функции принимают в качестве параметра требуемую длину вектора (**n**), которая должна быть целым положительным числом, и возвращают вектор-столбец **w**. Рассмотрим для примера две функции.

Функция **boxcar**, реализующая «прямоугольное окно», введена в MATLAB лишь для полноты набора весовых функций, так как она соответствует отсутствию операции взвешивания: **w=boxcar(n)**. Возвращаемый вектор заполнен единицами: **w = ones(n,1)**.

Функция **hamming** реализует окно Хэмминга: **w = hamming (n,'sfl ag')**.

Строковый параметр **'sfl ag'** позволяет выбрать режим расчета окна. При значении **'symmetric'**, принятом по умолчанию, генерируется симметричное окно, для которого **w(k)=w(n +1 – k)**. Оно используется при расчете фильтров. При значении **'periodic'** создается слегка несимметричное окно, которое используется при спектральном анализе.

Спектрограмма

Если спектр сигнала меняется во времени, то для оценки спектра целесообразно использовать спектрограмму сигнала. Спектрограммой (spectrogram) сигнала называется *его мгновенный спектр,* зависящий от времени. Для вычисления спектрограммы вектор сигнала разбивается на сегменты (в общем случае с перекрытием). Для каждого сегмента вычисляется спектр с помощью функции **fft**. Набор спектров всех сегментов и образует спектрограмму. Для вычисления спектрограммы служит функция **spectrogram**.

Синтаксис вызова функции: **[S,F,T]=spectrogram (x,window, noverlap, nfft,Fs)**, где **x** – вектор сигнала; **window** – вектор весовой функции (если вместо вектора используется целое число, то используется весовая функция по умолчанию – функция Хэмминга соответствующей длины); **noverlap** – величина перекрытия соседних сегментов сигнала; **nfft** – число точек преобразования Фурье; **Fs** – частота дискретизации. **S –** матрица, каждая колонка которой содержит (**nfft/2+1**) отсчетов спектра для данного момента времени (если **nfft –** нечетное число, количество отсчетов равно (**nfft+1)/2**). Число колонок **k=fi x((nx-noverlap)/(length(window)noverlap))**, где **nx** – длина вектора сигнала. Параметр **F** – вектор частот, **T** – вектор моментов времени, его длина равна **k**.

Если выходные параметры функции не указываются (**spectrogram(x, window,noverlap, nfft, Fs)**), то строится трехмерный график спектральной плотности мощности в координатах: время, частота, уровень.

Обязательным входным параметром функции является вектор значений сигнала **x**, остальные параметры имеют значения по умолчанию, которые используются, если в качестве параметра указана пустая матрица (**[]**) или если несколько последних параметров при вызове опущены. Пример (рис. 4.11):

```
[V,fs,b]=wavread('9m.wav');
spectrogram(V,256,128,[],fs,'yaxis');
fi gure
spectrogram(V,256,128,1024,fs,'yaxis');
```

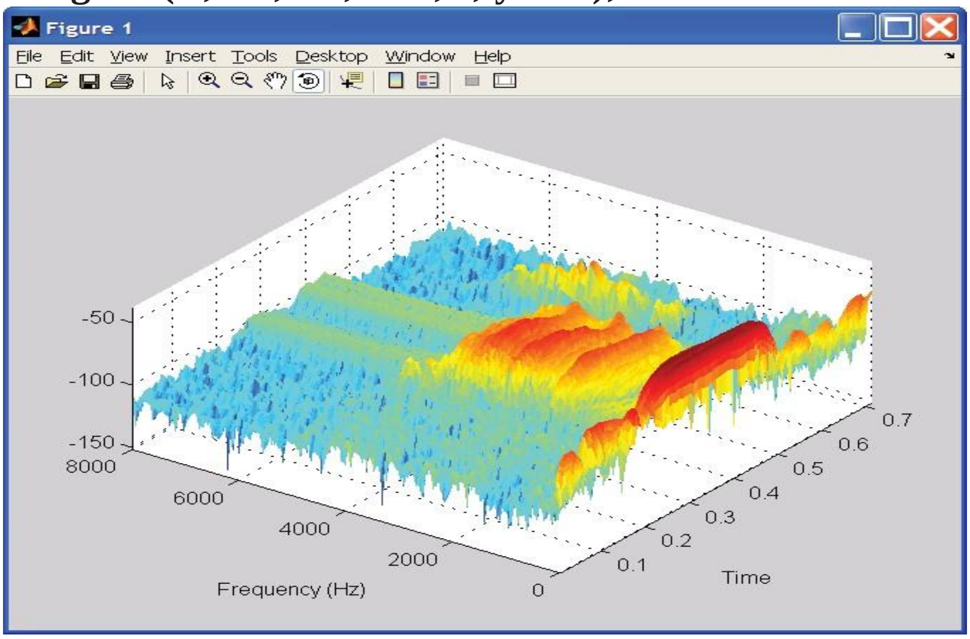

Рис. 4.11

Здесь во второй строке примера для параметра **nfft** используется значение по умолчанию – максимальное издвух чисел: 256 и 2^k. Значение k таково, что выполняется условие **2^k > window.**

Периодограмма

Периодограммой называется оценка спектральной мощности сигнала, полученная по N отсчетам одной реализации случайного процесса. Периодограмма используется для оценки спектра мощности стационарного случайного процесса. Для вычисления периодограммы предназначена функция **periodogram**. Синтаксис ее вызова следующий: **[Pxx, f] = periodogram(x, window, Nfft, Fs, 'range')**.

Обязательным входным параметром является **x** – вектор отсчетов сигнала. Остальные параметры имеют значения по умолчанию, которые используются, если в качестве параметра указана пустая матрица [] или если некоторое количество параметров (начиная с последнего) опущены при вызове.

Вектор **window** должен содержать коэффициенты используемого окна (при этом говорят о модифицированной периодограмме). По умолчанию используется прямоугольное окно.

Параметр **Nfft** задает размерность БПФ, используемого для вычисления периодограммы. По умолчанию этот параметр равен максимальному из двух чисел: 256 и 2^k. Значение k таково, что выполняется условие $\mathbf{2^k > length(x)}$. Входной сигнал, умноженный на окно, приводится к размеру **Nfft** (обрезается либо дополняется нулями).

Параметр **Fs** – частота дискретизации в герцах. Значение по умолчанию равно 2π. Строковый параметр **'range'** определяет частотный диапазон для возвращаемого вектора **Pxx**. Возможны два значения:

'twosided' – векторы **Pxx** и **f** имеют длину **Nfft** и соответствуют полному диапазону частот 0... **Fs**. Этот вариант используется по умолчанию, если **x** содержит комплексные отсчеты;

'onesided' – векторы **Pxx** и **f** имеют длину **ceil ((Nfft + 1)/2)** и соответствуют половинному диапазону частот 0... **Fs/2**. Этот вариант используется по умолчанию в случае вещественного вектора **x**.

Параметр **'range'** может быть указан в списке параметров в любом месте после **window**.

Возвращаемые параметры: **Pxx** – вектор значений спектральной плотности мощности, **f** – вектор значений частот, использованных для расчета. Шаг между соседними элементами этого вектора равен **Fs/Nfft**, первый элемент равен нулю. Если выходные параметры при вызове не указаны, то функция строит график спектральной плотности мощности. Пример оценки спектральной плотности мощности зашумленного гармонического колебания (рис. 4.12):

```
Fs = 1000; t = 0:1/Fs:3; x =
cos(2*pi*t*200)+randn(size(t));
periodogram(x,[],[],Fs);
```

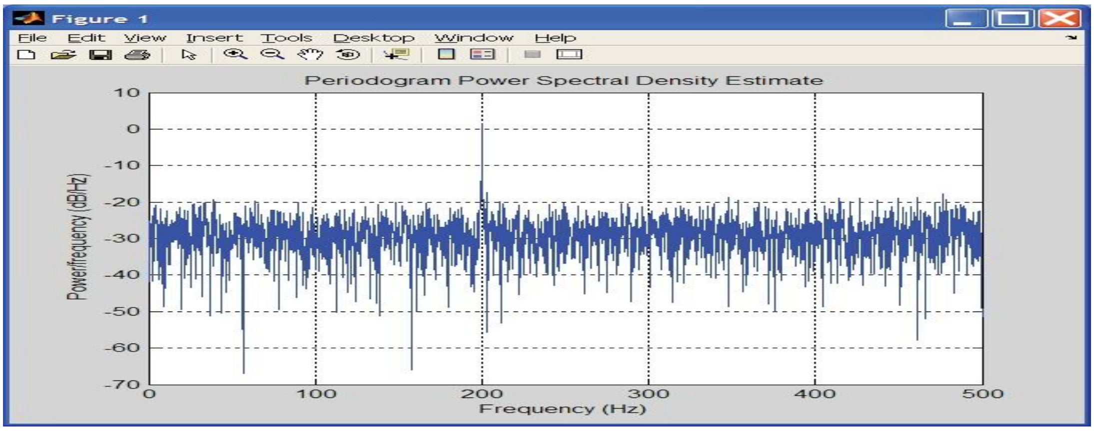

Рис. 4.12

Здесь используется значение окна по умолчанию.

Вычисление периодограммы по методу Уэлча

При вычислении периодограммы по длинному фрагменту случайного сигнала она оказывается весьма изрезанной, что затрудняет оценку формы спектра. Для уменьшения изрезанности необходимо применить усреднение, которое реализуется методом Уэлча. Вычисления при использовании метода Уэлча организуются следующим образом:

1. Вектор отсчетов сигнала делится на перекрывающиеся сегменты. Как правило, используется перекрытие на 50 %. Каждый сегмент умножается на используемую весовую функцию.
2. Для взвешенных сегментов вычисляются модифицированные периодограммы.
3. Периодограммы всех сегментов усредняются.

Следует отметить, что метод Уэлча для оценки спектра использует меньшие по размеру сегменты сигнала, поэтому разрешающая способность спектрального анализа (возможность различить две рядом расположенные спектральные линии) в этом случае снижается. Для реализации метода Уэлча используется функция **pwelch.** Синтаксис вызова функции следующий: **[Pxx, f] = pwelch(x, Nwin, Noverlap, Nfft, Fs, 'range')**.

Обязательным входным параметром является **x** – вектор отсчетов анализируемого сигнала. Все остальные параметры имеют значения по умолчанию, которые используются, если при вызове в качестве параметра указана пустая матрица ([]) или если несколько последних параметров опущено.

Параметр **Nwin** управляет выбором окна, используемого для анализа. Если **Nwin** – число, используется окно Хэмминга указанной длины, если вектор, то данный вектор используется в качестве окна. По умолчанию используется окно Хэмминга, длина которого выбирается так, чтобы с учетом заданного перекрытия (см. ниже) сигнал оказался разделенным на восемь фрагментов.

Параметр **Noverlap** задает (в отсчетах) перекрытие соседних фрагментов сигнала, для которых вычисляются периодограммы. По умолчанию перекрытие равно половине длины окна.

Параметр **Nfft** задает размерность БПФ, используемого для вычисления периодограммы. По умолчанию **Nfft** равно максимальному из двух чисел: 256 и 2^k. Значение **k** таково, что выполняется условие $\mathbf{2^k > Nwin}$**,** где **Nwin** – длина фрагмента сигнала (длина используемого окна).

Параметр **Fs** указывает частоту дискретизации сигнала. Это значение используется для нормировки рассчитанного спектра мощности, а также при расчете возвращаемого вектора **f** и для оцифровки графика. По умолчанию значение этого параметра равно 2π.

Строковый параметр **'range'** определяет частотный диапазон для возвращаемого вектора **Pxx**. Возможны два значения:

'twosided' – векторы **Pxx** и **f** имеют длину **Nfft** и соответствуют полному диапазону частот **0...Fs**. Этот вариант используется по умолчанию, если **x** содержит комплексные отсчеты;

'onesided' – векторы **Pxx** и **f** соответствуют половинному диапазону частот **0...Fs/2**. Этот вариант используется по умолчанию в случае вещественного вектора **x**.

Параметр **'range’** может быть указан в списке параметров в любом месте после **Noverlap**.

Возвращаемые параметры: **Pxx** – вектор значений спектральной плотности мощности, **f** – вектор значений частот, использованных для расчета. Шаг между соседними элементами этого вектора равен **Fs/Nfft**, первый элемент равен нулю. Если выходные параметры при вызове не указаны, функция строит график спектральной плотности мощности с помощью функции **psdplot**.

Расчет спектра производится следующим образом. Анализируемый сигнал **x** делится на перекрывающиеся фрагменты согласно параметрам **Nwin** и **Noverlap**. Для каждого фрагмента вычисляется модифицированная периодограмма с использованием заданных окна и размерности БПФ Полученный набор модифицированных периодограмм усредняется. Пример (рис. 4.13):

```
Fs = 1000; t = 0:1/Fs:3; f0=200;
x=cos(2*pi*f0*t)+randn(size(t));
pwelch(x,[],[],[],Fs,'onesided');
```

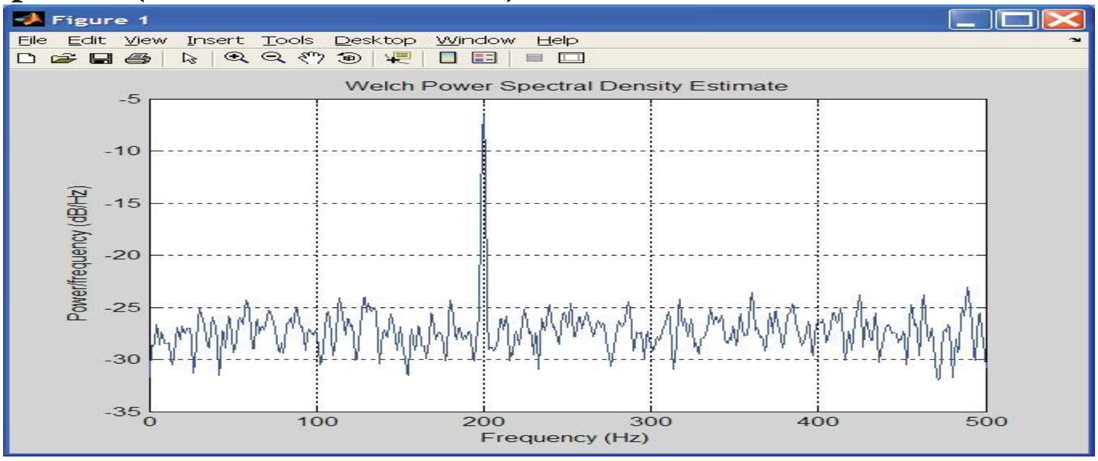

Рис. 4.13

Задание 2

1. Сформировать последовательность из трех импульсов, представляющих собой положительные полуволны синусоиды. Частота синусоиды 0,5 кГц, период следования импульсов 10 мс, частота дискретизации 10 кГц.
2. Сформировать сигнал, представляющий собой сумму гармонического сигнала с частотой 2 кГц и амплитудой 2 В и белого гауссова шума, который имеет уровень на 20 дБ меньший сигнала. Частота дискретизации 10 кГц, интервал времени (0…1) с.
3. Определить периодограмму, спектрограмму, периодограмму, построенную по методу Уэлча, для сигнала, представляющего собой сумму гармонического сигнала с частотой 1 кГц и амплитудой 4 В и белого гауссова шума, который имеет уровень на 10 дБ меньший сигнала. Частота дискретизации 10 кГц, интервал времени (0…1) с. Использовать окно Хэмминга длительностью 200 и с перекрытием 20 отсчетов.

4. Построить таблицу изученных команд и функций с пояснениями по их использованию.

Содержание отчета

1. Распечатка истории команд.
2. Тексты решений задач и графики по итогам решения.
3. Таблица изученных команд и функций с пояснениями по их использованию.

Контрольные вопросы

1. Ответить на вопросы преподавателя по использованию изученных команд и функций.
2. Дать сравнительную характеристику функций **pwelch** и **periodogram** по разрешающей способности спектрального анализа.
3. По спектрограмме заданного звукового файла (запись голосовой команды) определить частоты основных формант и частоту основного тона для вокализованных звуков.
4. В каких случаях при анализе сигнала целесообразно использовать спектрограмму, а в каких случаях – периодограмму?