на правах рукописи

МИНОБРНАУКИ РОССИИ

Нижнекамский химико-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Казанский национальный исследовательский технологический университет»
НХТИ ФГБОУ ВО «КНИТУ»
Факультет управления и автоматизации
Кафедра Автоматизации технологических процессов и производств

ЛАБОРАТОРНАЯ РАБОТА

ПО ТЕМЕ: СОЗДАНИЕ И ОБРАБОТКА СИГНАЛОВ В ПРОГРАММНОЙ СИСТЕМЕ MATHCAD

Нижнекамск, 2016

Цель работы: Изучение принципов создания сигналов различной формы и их последующего анализа и обработки в программной системе MathCAD (пакет расширения Signal Processing Toolbox)

Методические указания по теоретической части

Пакет расширения Signal Processing Toolbox системы MATHLAB содержит порядка 150 функций, реализующих следующие задачи:

1) создание сигналов различного типа, в том числе модулированных;
2) создание окон фильтрации и спектрального анализа;
3) реализация прямого и обратного преобразований Фурье, в том числе быстрого преобразования Фурье;
4) реализация дискретного косинусного и других преобразований сигналов;
5) оценка спектральной плотности мощности сложных сигналов;
6) анализ линейных систем и цепей;
7) фильтрация сигналов;
8) моделирование работы различных фильтров и вычисление их характеристик и др.

Для обзора функций пакета Signal Processing Toolbox нужно выполнить в командной строке MATHLAB команду >> **help signal**. Справку по любой функции можно получить с помощью команды **>> help name_function**.

I. Работа с комплексными числами

При обработке сигналов часто встречаются комплексные числа. В MATHLAB имеются обширные средства для работы с ними. В пакете Signal Processing Toolbox некоторые средства продублированы:

- вычисление модуля комплексного числа – abs;
- вычисление фазы комплексного числа – angle;
- группирование комплексных чисел – cplxpair (x, [tol]): комплексные числа из вектора x размещаются в порядке возрастания действительной части, комплексно-сопряженные числа размещаются рядом, причем вначале те, у которых отрицательная мнимая часть, действительные числа из x размещаются в конец последовательности в возрастающем порядке. Необязательный параметр tol задает минимальное значение мнимой части, при которой число считается действительным. По умолчанию tol=100×eps, eps- погрешность численных расчетов.

II. Моделирование сигналов

Моделирование зашумленных сигналов в пакете Signal Processing Toolbox обеспечивается с помощью генератора случайных чисел. Инициализация генератора случайных чисел осуществляется командой

>> randn (‘static’, 0);

1) Генерация матрицы случайных чисел размерности m×n с нормальным законом распределения с нулевым средним и единичной дисперсией осуществляется функцией **randn ([m n])**.

Пример генерации сложного зашумленного сигнала (рис.1), содержащего 2 синусоидальные компоненты – первая (основная) с амплитудой 1 и частотой 1

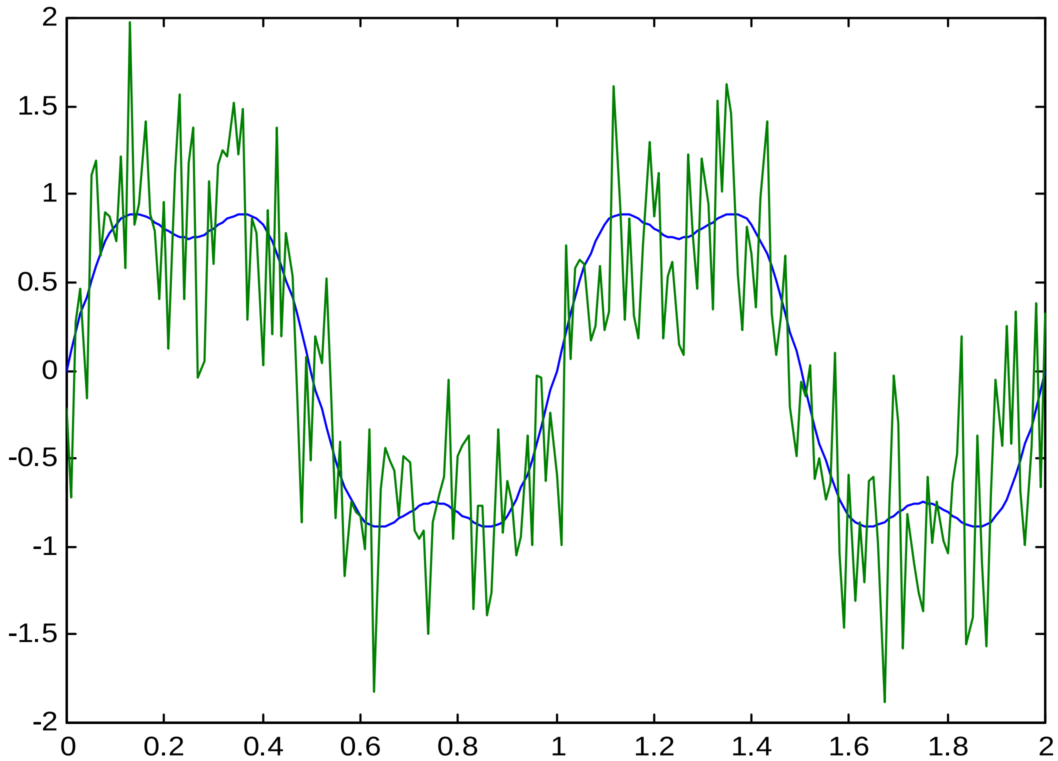

Рис.1 График исходного сложного и зашумленного сигналов.

Гц; вторая с амплитудой 0.25 и частотой 3 Гц.:

\>> t= (0 : 0.01 : 2)' ;

\>> y= sin (2*pi*1*t) + 0.25* sin (2*pi*3*t);

\>> randn (‘state’, 0);

\>> yn= y + randn (size(t));

\>> plot (t, y, t, yn);

Команда plot строит графики сигналов y(t) и yn(t) (рис.1). Этот пример наглядно показывает, насколько шум искажает форму сигнала. По виду зашумленного сигнала уже практически невозможно определить наличие второй компоненты сигнала – третьей гармоники. Поэтому основной задачей пакета Signal Processing Toolbox является фильтрация сигналов и, соответственно, построение различного типа фильтров.

2) Функция **y=chirp (t, f0, t1, f1 [, ‘method’, phi])** формирует дискретные значения косинусоидального сигнала с частотой f0 в начальный момент t до f1 в конечный момент t1. По умолчанию t=0, f0=0, f1=100. Необязательный параметр phi (по умолчанию 0) задает начальную фазу сигнала. Необязательный параметр ‘method’ задает закон изменения частоты (по умолчанию method=linear) и может быть следующим:

- linear – линейный закон изменения частоты $f_j(t) = f_0 + a \cdot t, a = \frac{f_1 - f_0}{t_1}$,
- quadratic – квадратичный закон изменения частоты $f_j(t) = f_0 + a \cdot t^2, a = \frac{f_1 - f_0}{t_1}$;
- logarithmic – логарифмический закон изменения частоты $f_j(t) = f_0 + 10^{a \cdot t}, a = \frac{\log(f_1 - f_0)}{t_1}, f_1 > f_0$;

3) Функция **y=diric (x, n)** формирует вектор значений сигнала, представленного функцией Дирихле: $\operatorname{diric}(x, n) = \begin{cases} -1^{\frac{x \cdot (n-1)}{2\pi}}, x = 0, \pm 2\pi, \pm 4\pi, ... \\ \frac{\sin(nx/2)}{n \sin(x/2)}, \text{при других значениях } x \end{cases}$. Размерность вектора y равна размерности x. Функция diric периодическая, при этом период кратен 2π при нечетных n и 4π при четных.
4) Функция **yi= gauspuls (t, fc, bw [, bwr]) или [yi, yq]= gauspuls (t, fc, bw [, bwr]) или [yi, yq, ye]= gauspuls (t, fc, bw [, bwr]) или tc= gauspuls (‘cutoff’,t, fc, bw , bwr, tpe)** формирует синусоиду, модулированную по амплитуде функцией Гаусса. Функция может использоваться в трех видах. В первом случае она создает вектор отчетов yi для моментов времени, заданных в векторе t. Параметр fc задает частоту синусоиды, bw – ширину полосы частот сигнала. По умолчанию fc=1000, bw=0.5. Необязательный параметр bwr (его значение отрицательное) задает сигнал единичной амплитуды с частотой fc и шириной полосы частот bw, причем граница полосы частот задается ослаблением амплитуды на заданное число децибел bwr (по умолчанию – 6дБ). Во втором случае дополнительно возвращается вектор отчетов сигнала, фаза которого сдвинута на 90 °, а в третьем – в выходном параметре ye возвращается вектор отчетов огибающей сигнала. Четвертая форма задания функции формирует время отсечения tc, которое определяется по спаду амплитуды от максимального уровня до уровня tpe дБ (по умолчанию – 60 дБ).
5) Функция **y= gmonopuls (t, fc)** генерирует вектор отчетов y Гауссового моноимпульса для заданного вектора отчетов времени t, а функция **tc = gmonopuls (‘cutoff’, fc)** возвращает интервал времени tc, отсчитанного во время спада амплитуды от максимального значения до минимального.
6) Функция **y=pulstran (t, d, ‘func’ [, p1, p2,…])** генерирует отчеты импульсных сигналов различной формы, которая задается параметром ‘func’:
   - gauspuls – синусоида, модулированная по закону Гаусса;
   - rectpuls – прямоугольный импульс;
   - tripuls – треугольный импульс.

Вектор y вычисляется для отчетов времени, заданных вектором t, по формуле y= func (t – d(1)) + func (t – d (2)) +… Число импульсов в заданном интервале времени равно length (d). Необязательные параметры p1, p2,… позволяют задавать дополнительные параметры обращения к **‘func’**, например типа func (t – d(1), p1, p2,…). При записи функции в виде **y=pulstran (t, d, p [, f_S])** можно задать частоту дискретизации f_S (по умолчанию 1 Гц).

7) Функция **y= sawtooth (t, [, width])** генерирует вектор пилообразных и треугольных колебаний, уровень которых меняется от –1 до 1 на периоде 2π. Если задан необязательный параметр width, то импульс на интервале от 0 до 2π* width нарастает в указанных пределах, а на интервале от 2π* width до 2π уменьшается от 1 до –1.
8) Функция **y = sinc (t)** генерирует вектор (матрицу) сигнала по формуле $y = \begin{cases} 1 \text{ при } t = 0 \\ \sin(\pi \cdot t) \text{ при } t \neq 0 \end{cases}$. Размер вектора (матрицы) y совпадает с размерностью t.
9) Функция **y = square (t [,duty])** генерирует вектор сигнала прямоугольной формы с периодом 2π для моментов времени из вектора t. Положительная полуволна импульсов равна +1, а отрицательная –1. Необязательный параметр duty (по умолчанию 50) задает продолжительной положительной части полуволны колебаний в % от периода.
10) Функция **y = tripuls (T [,w [,s]])** генерирует вектор значений треугольного апериодического импульса единичной амплитуды, центрированного относительно T=0. Параметр w задает ширину импульса (по умолчанию 1), а параметр –1<s<1 задает асимметрию импульса (по умолчанию 0).
11) Функция **y = vco (x, fc, fs)** (управляемый напряжением источник) генерирует вектор косинусоидального сигнала с частотной модуляцией. Параметр fs задает среднюю частоту сигнала с единичной амплитудой. Вектор управляющего воздействия должен содержать действительные значения воздействия в диапазоне от –1 до 1. При этом отклонение меняется от 0 до 2*fs. Размер вектора y совпадает с размерностью x. В форме **y = vco (x, [Fmin, Fmax], fs)** можно задать изменение частоты от Fmin до Fmax. Желательно, чтобы изменение частоты не превышало fs/2. Аргумент x может быть матрицей.

III. Функции задания окон

Для точного представления сигналов необходимо выполнять операции спектрального анализа и синтеза для бесконечного числа гармоник периодического сигнала или учета спектральной плотности мощности на бесконечном диапазоне частот. Поэтому, как правило, реальный численный спектральный анализ и синтез практически невозможны. Но большинство реальных сигналов имеют ограниченный спектр, следовательно, для них возможен спектральный анализ в ограниченном частотном диапазоне. Средства ограничения частотного спектра или временной области задания сигналов называются **окнами**. Окна могут быть различного типа и характеризуются графическими зависимостями своих коэффициентов и различными специфическими параметрами. Наиболее широко используются Гауссовы окна, т.к. они дают малые искажения спектра сигнала в процессе его ограничения в окнах. Использование окон лежит в основе кратковременного или оконного преобразования Фурье.

Пакет расширения Signal Processing Toolbox имеет ряд функций для задания окон. Как правило, они не имеют самостоятельного значения и применяются при выполнении спектрального анализа и синтеза. Все эти функции создают вектор-столбец коэффициентов окна соответствующего типа.

1) Функция **w = bartlett (n)** формирует вектор w коэффициентов n-точечного окна Бартлетта. Эти коэффициенты рассчитываются по формулам:

$$w(k) = \begin{cases} \dfrac{2(k-1)}{n-1} \text{ при } 1 \le k \le \dfrac{n+1}{2} \\ \dfrac{2-2(k-1)}{n-1} \text{ при } \dfrac{n+1}{2} \le k \le n \end{cases} \quad \text{при нечетном } n$$

$$w(k) = \begin{cases} \dfrac{2(k-1)}{n-1} \text{ при } 1 \le k \le \dfrac{n}{2} \\ \dfrac{2(n-k-1)}{n-1} \text{ при } \dfrac{n}{2} \le k \le n-1 \end{cases} \quad \text{при четном } n.$$

2) Функция **w = blackman (n [,'sflag'])** формирует вектор w коэффициентов n-точечного окна Блекмана, вычисляемые по формуле $w(k) = 0.42 - 0.5\cos\left(2\pi\dfrac{k-1}{n-1}\right) + 0.8\cos\left(4\pi\dfrac{k-1}{n-1}\right),\ k = 1,2,...,n.$

Необязательный параметр sflag может иметь следующие значения:

- symmetric – задает симметричное окно (используется по умолчанию);
- periodic – вычисляет окно для (n+1) точки, но возвращает только первые n точек.

3) Функция **w= boxcar (n)** формирует вектор w коэффициентов n-точечного прямоугольного окна, вычисляемое как w= ones (n, 1).

4) Функция **w= chebwin (n, r)** формирует вектор w коэффициентов n-точечного окна Чебышева с пульсациями на уровне r дБ в полосе задержания относительно амплитуды в полосе пропускания.

5) Функция **w= hamming (n [, 'sflag'])** формирует вектор w коэффициентов n-точечного окна Хемминга, вычисляемый по формуле: $w(k) = 0.54 - 0.46\cdot\cos\left(\dfrac{2\cdot\pi\cdot k}{n-1}\right), k = 0,1,....,n-1.$ Необязательный параметр sflag имеет такой же смысл, что и в функции blackman.

6) Функция **w= hanning (n [, 'sflag'])** формирует вектор w коэффициентов n-точечного окна Хенна (Хеннинга), вычисляемый по формуле: $w(k) = 0.5\cdot\left(1-\cos\left(\dfrac{2\cdot\pi\cdot k}{n-1}\right)\right), k = 1,2,....,n-1.$ Необязательный параметр sflag имеет такой же смысл, что и в функции blackman.

7) Функция **w= kaiser (n, β)** формирует вектор-столбец w коэффициентов n-точечного окна Кайзера. Параметр β задает затухание боковых лепестков окна.
8) Функция **w= triang (n)** формирует вектор w коэффициентов n-точечного треугольного окна. При четном n это окно совпадает с окном Бартлетта, за исключением того, что при k=0 и k=1 его значение равно 0. При нечетном n коэффициенты треугольного окна вычисляются по формуле:

$$w(k)=\begin{cases}\dfrac{2k-1}{n-1}\ \text{при}\ 1\le k\le\dfrac{n+1}{2}\\ \dfrac{2(m-k+1)}{n-1}\ \text{при}\ \dfrac{n+1}{2}\le k\le n\end{cases}.$$

IV. Дискретное быстрое преобразование Фурье (БПФ)

а) Функция **y=fft(x [, n])** реализует алгоритм БПФ, при котором вектор y вычисляется по формуле: $y(k+1)=\sum\limits_{n=0}^{N-1} x(n+1)\cdot e^{-j\cdot 2\pi\cdot k/N}$, где N = length(x).

Если N-степень число 2, то используется высокоэффективный алгоритм БПФ для вещественных или комплексных данных. Если N<n, то недостающие элементы x дополняются нулями.

б) При выполнении прямого БПФ спектральные компоненты, близкие к нулевой частоте, группируются по краям спектрограммы. Функция **y=fftshift(x)** осуществляет перегруппировку элементов выходного вектора преобразования Фурье так, что эти компоненты оказываются в центре графика.

в) Функция **y=ifft(x [, n])** реализует обратное БПФ, при котором вектор y вычисляется по формуле: $y(t)=\frac{1}{N}\sum\limits_{k=0}^{N-1} x(k+1)\cdot e^{-j\cdot 2\pi/N}$, где N = length(x).

Если N-степень число 2, то используется высокоэффективный алгоритм БПФ для вещественных или комплексных данных. Если N<n, то недостающие элементы x дополняются нулями.

г) Функция **A=dftmtx(n)** возвращает матрицу дискретного преобразования Фурье размерности n×n, такую, что матричное выражение y=A·x задает прямое дискретное преобразование Фурье.

д) Функция **Y=fft2(X [, m, n])** реализует прямое двумерное БПФ для матрицы X, приведенной к матрице размера m×n. Если x имеет иной размер, то она дополняется 0 или усекается до этого размера. Результат преобразования – матрицаY с комплексными элементами размерности m×n или размерности матрицы X (при отсутствии параметров m, n).

е) Функция **Y=ifft2(X [, m, n])** реализует обратное двумерное дискретное БПФ.

V. Реализация методов спектрального анализа сигналов

Спектральный анализ сигналов – важнейшая задача пакета Signal Processing Toolbox, которая реализуется большим числом методов. Все методы вычисляют спектральную плотность мощности (СПМ) сигналов. Функции

спектрального анализа начинаются с буквы p. Входные параметры функций спектрального анализа следующие:
- x – вектор значений сигнала;
- p – порядок модели;
- nfft- число отчетов сигнала, используемых при БПФ (по умолчанию 256);
- f – частота (в Гц);
- w – круговая частота;
- fs – частота дискретизации;
- ‘range’ – параметр указания диапазона частот;
- 'squared' – параметр, отменяющий вывод графика СПМ в децибелах и задающий график в прямоугольной системе координат.

Некоторые входные параметры могут отсутствовать, в этом случае их нужно задавать как пустую матрицу [ ] для указания использования соответствующего параметра по умолчанию.

Выходные параметры этих функций следующие:
- freq – вектор частот, для которых осуществляется оценка СПМ;
- Pxx – вектор-столбец оценки СПМ, размер которого при четном nfft = nfft/2+1, а при нечетном (nfft+1)/2.

Если x содержит комплексные данные, то СПМ оценивается для всех частот и число элементов Pxx=nfft. Все функции имеют внутреннее обращение к функции plot (f, Pxx) и строят графики зависимости СПМ от частоты.

1) Вычисление СПМ методом Бурга осуществляется функцией **pburg**, которая имеет следующие варианты:
- **Pxx= pburg (x, p);**
- **[Pxx, w]= pburg (x, p);**
- **[Pxx, w]= pburg (x, p, nfft);**
- **[Pxx, f]= pburg (x, p, nfft, fs);**
- **[Pxx, f]= pburg (x, p, nfft, fs, ‘range’);**
- **[Pxx, w]= pburg (x, p, nfft, ‘range’);**
- **pburg (…)**;

2) Вычисление СПМ ковариационным методом осуществляется функцией **pcov**, которая имеет следующие варианты:
- **Pxx= pcov (x, p);**
- **[Pxx, w]= pcov (x, p);**
- **[Pxx, w]= pcov (x, p, nfft);**
- **[Pxx, f]= pcov (x, p, nfft, fs);**
- **[Pxx, f]= pcov (x, p, nfft, fs, ‘range’);**
- **[Pxx, w]= pcov (x, p, nfft, ‘range’);**
- **pcov (…);**

3) Вычисление СПМ модифицированным ковариационным методом осуществляется функцией **pmcov**, которая имеет следующие варианты:
- **Pxx= pmcov (x, p);**
- **[Pxx, w]= pmcov (x, p);**
- **[Pxx, w]= pmcov (x, p, nfft);**
- **[Pxx, f]= pmcov (x, p, nfft, fs);**

- **[Pxx, f]= pmcov (x, p, nfft, fs, ‘range’);**
- **[Pxx, w]= pmcov (x, p, nfft, ‘range’);**
- **pmcov (…);**

4) Вычисление СПМ многооконным методом осуществляется функцией **pmtm**, которая имеет следующие варианты:

- **[Pxx, w]= pmtm (x, nw);**
- **[Pxx, w]= pmtm (x, nw, nfft);**
- **[Pxx, f]= pmtm (x, nw, nfft, fs);**
- **[Pxx, Pxxc, f]= pmtm (x, nw, nfft, fs, p);**
- **[Pxx, Pxxc, f]= pmtm (x, nw, nfft, fs);**
- **[Pxx, Pxxc, f]= pmtm (x, e, v, nfft, fs, p);**
- **[Pxx, Pxxc, f]= pmtm (x, dpss_params, nfft, fs, p, ‘method’);**
- **[Pxx, Pxxc, f]= pmtm (x, dpss_params, nfft, fs, p, ‘method’,‘range’);**
- **pmtm (…);**

Эта функция работает с вещественными данными и использует совокупность ортогональных окон. Параметр nw (по умолчанию 4) задает спектрально-временное разрешение, которое рекомендуется выбирать равным 2, 2.5, 3, 3.5. Параметр ‘method’ позволяет задать метод вычисления СПМ:

- adapt – адаптивный нелинейный алгоритм Томсона комбинации индивидуальных оценок (используется по умолчанию);
- unity – линейная комбинация индивидуальных оценок с весами, равными 1;
- eigen – линейная комбинация индивидуальных оценок с весами, задаваемыми собственными значениями;

5) Вычисление СПМ методом Уэлча осуществляется функцией **pwelch**, которая имеет следующие варианты:

- **[Pxx, w]= pwelch (x);**
- **[Pxx, w]= pwelch (x, nwin);**
- **[Pxx, w]= pwelch (x, nwin, noverlap);**
- **[Pxx, w]= pwelch (x, nwin, noverlap, nfft);**
- **[Pxx, f]= pwelch (x, nwin, noverlap, nfft, fs);**
- **[Pxx, f]= pwelch (x, nwin, noverlap, nfft, fs, ‘range’);**
- **pwelch (…)**;

Необязательный целочисленный параметр nwin задает длину окна Хэмминга. Если nwin – двухэлементный вектор, то он задает размеры прямоугольного окна.

Пример: randn (‘state’, 0); fs=1000; t= 0: 1/fs: 3;
x= cos (2*pi*t*200)+randn (size(t));
pwelch (x, 33, 32, [ ], fs, ‘twosided’)

6) Вычисление СПМ методом собственных значений осуществляется функцией **peig**, которая имеет следующие варианты:

- **[s, w]= peig (x, p);**
- **[s, w]= peig (x, p, nfft);**
- **[s, f]= peig (x, p, nfft, fs);**
- **[s, f]= peig (x, p, nfft, fs, ‘corr’);**
- **[s, f]= peig (x, p, nfft, fs, nwin,noverlap);**

- **[s, f]= peig (x, p, nfft, fs, nwin,noverlap, ‘range’);**
- **[s, f]= peig (…);**
- **peig (…)**;

7) Вычисление СПМ методом Юла-Уокера осуществляется функцией **pyulear**, которая имеет следующие варианты:

- **Pxx= pyulear (x, p);**
- **[Pxx, w]= pyulear (x, p, nfft);**
- **[Pxx, f]= pyulear (x, p, nfft, fs);**
- **[Pxx, f]= pyulear (x, p, nfft, fs, ‘range’);**
- **[Pxx, w]= pyulear (x, p, nfft, ‘range’);**
- **pyulear (…);**

VI. Средства визуализации спектра сигналов

Статистика сигналов

К таким средствам относятся графики периодограмм, спектральной плотности и спектограмм.

1) Для построения периодограмм (зависимостей спектральной мощности дБ/Вт от частоты) служит функция **periodogram**:

- **[Pxx, w]= periodogram (x);**
- **[Pxx, w]= periodogram (x, window);**
- **[Pxx, w]= periodogram (x, window, nfft);**
- **[Pxx, f]= periodogram (x, window, nfft, fs);**
- **[Pxx, …]= periodogram (x, …, ‘range’);**
- **periodogram (…)**

Частота может задаваться как угловая f или в герцах. Параметр ‘range’ может быть следующим:

- ‘twosided’- вычисляет двухполосную спектральную мощность в частотном диапазоне от 0 до fs. При задании вместо этого параметра пустого вектора [ ] частотный диапазон от 0 до 1. Если не используется спецификация fs, то диапазон от 0 до 2;
- ‘onesided’ - вычисляет однополосную спектральную мощность в частотном диапазоне, задаваемом для вещественных компонентов вектора x. Для x с вещественными элементами этот параметр используется по умолчанию.

2) Для построения графиков спектральной плотности служит графическая команда psdplot (Pxx, w [, ‘units’, ‘yscale’, ‘title’]). Параметр ‘units’ может иметь значение ‘rad/sample’ (по умолчанию) или ‘Hz’. Параметр ‘yscale’, задающий масштаб по вертикали, может иметь значение ‘db’ (по умолчанию) для построения графиков в логарифмическом масштабе или ‘linear’ для графика в линейном масштабе. Параметр ‘title’ – титульная надпись на графике.

Пример построения графика зависимости спектральных составляющих сигнала от нормализованной частоты для зашумленного косинусоидального сигнала с частотой 150 Гц:

t= 0: 0.001: 0.3; x= cos(2*pi*t*150)+0.2*randn (size(t));

[Pxx, w]= periodogram (x, [ ], ‘onesided’, 512);

psdplot (Pxx, w, ‘’.’’, ‘График спектральной плотности мощности’).

3) Для визуализации БПФ служит функция построения спектрограмм **specgram**. Спектрограмм является весьма информативной характеристикой сигнала и часто позволяет выявить самые тонкие его особенности. Спектрограмма строится в плоскости частота-время, при этом амплитуда каждой спектральной составляющей определяет цвет построения каждой точки спектрограммы. Существуют следующие формы записи функции:

а) **B=specgram (x)** вычисляет спектрограмму сигнала с отчетами в векторе x. При этом nfft= min (256, length(x)), fs=2., window –окно Хэмминга с длиной nfft и numoverlap= length(window)/2;

б) **[B, f]=specgram (x, nfft)**;

в) **[B, f, t]=specgram (x, nfft, fs)**;

г) **B=specgram (x, nfft, fs, window [, numoverlap]);**

д) **B=specgram (x, f, fs, window [, numoverlap]);**

е) **specgram (…)** строит спектрограмму в текущем окне. Наряду с амплитудами спектральных составляющих B может возвращаться вектор частот БПФ f и вектор времени t. Размер t равен числу столбцов B. Параметр numoverlap задает число отчетов, на которое происходит перекрытие блоков. При x с комплексными элементами, B будет содержать комплексные компоненты с числом строк nfft.

Пример: load mtlb; specgram (mtlb, 512, fs, kaiser(500, 5), 475)
title(‘Спектрограмма звуковых колебаний’)

В этом примере с жесткого диска считывается файл mtlb и затем строится спектрограмма, использующая окно Кайзера.

VII. Изменение частоты дискретизации

1) **Децимацией** сигналов называется уменьшение частоты их дискретизации в заданное число раз r. В пакете Signal Processing Toolbox для этого дискретный сигнал превращается в непрерывный с помощью фильтра того или иного типа, и с сигнала на выходе фильтра берутся новые выборки с заданной частотой. Децимация сигнала, заданного дискретными отсчетами в векторе x, реализуется функцией **y= decimate (x, t [, n, ‘fir’]).** При отсутствии параметров n и ‘fir’ используется фильтр Чебышева 8-го порядка. Параметр n задает порядок фильтра, который не рекомендуется брать выше 13 из-за возможной численной неустойчивости. При использовании параметра ‘fir’ применяется 30-точечный КИХ-фильтр. В этом случае n задает длину КИХ-фильтра.

Пример:
```
t= 0: 0.00025: 0.3;
x= sin(2*pi*30*t)+t;
stem (x(1:120)), axis ([0 120 –2 2])
title (‘Исходный сигнал’); figure
y= decimate (x, 4);
stem ( y(1:30))
title (‘Сигнал после децимации’)
```

Децимация может использоваться для уменьшения спектра сигнала и его сжатия с некоторой потерей точности восстановления.

2) **Интерполяция** сигналов заключается в вычислении значений сигнала в промежутках между его отсчетами. Это может использоваться для решения обратной децимации задачи – увеличения числа отсчетов сигнала, которое в пакете Signal Processing Toolbox реализуется функцией

y= interp (x, r [, l, alpha])   [y, b]= interp (x, r , l, alpha)

Целочисленный параметр r указывает во сколько раз увеличивается число отсчетов исходного векторы x. При этом вначале в исходную последовательность вносятся нулевые элементы и она расширяется, а затем она обрабатывается НЧ-фильтром. Необязательные параметры l и alpha задают порядок фильтра и частоту отсечки. Выходной параметр b – вектор с коэффициентами фильтра.

3) **Рациональное изменение частоты дискретизации** осуществляется функцией **y= redsample(x, p, q, [, n, beta])   y= redsample(x, p, q, b)**, которая позволяет задать изменение частоты дискретизации в p/q раз, где p и q –целые числа. Исходнвая последовательность x обрабатывается КИХ-фильтром с окном Кайзера. Сигнал на выходе фильтра дискретизируется с новой частотой. Если x- матрица, то обрабатываются ее столбцы, т.е. по существу ряд сигналов. Необязательный параметр n задает число отсчетов по обе стороны от элемента исходной выборки. При этом порядок фильтра пропорционален n. Можно задать вектор коэффициентов фильтра b – в этом случае для перевыборки отсчетов будет использоваться НЧ КИХ-фильтр с заданными коэффициентами. Параметр beta задает параметр окна Кайзера, который по умолчанию равен 5.

В варианте [**y, b]= redsample(x, p, q)** кроме вектора (матрицы) сигнала y с измененной частотой дискретизации возвращается вектор коэффициентов фильтра b, используемого для перевыборки отсчетов сигнала.

VIII. Модуляция и демодуляция сигналов

1) Создание модулированных сигналов, которые широко используются в технике связи, осуществляется функцией **y=modulate (x, fc, fs, ‘method’ [,opt])** или **[y, t]=modulate (x, fc, fs).** Эта функция генерирует вектор y отсчетов модулированного сигнала с несущей частотой fc и частотой дискретизации fs. Модулирующий сигнал задается отсчетами в векторе x. Выходной параметр t –вектор отсчетов времени. Параметр ‘method’ (по умолчанию am) может иметь следующие значения:

-amdsb-sc или am - амплитудная модуляция с двойной боковой полосой и подавленной несущей y=x.*cos(2*pi*fc*t);

-amdsb-tc или am - амплитудная модуляция с обеими боковыми полосами, частично подавленной несущей и изменяемой глубиной модуляции y=(x-opt).*cos(2*pi*fc*t). Здесь скалярный параметр opt задает степень подавления несущей – при opt=-1 несущая не подавляется и коэффициент модуляции =100%. По умолчанию opt= min(min(x)). Модулирующий сигнал при этом имеет только положительные значения с нулевым минимальным значением;

-amssb-sc - амплитудная модуляция на одной боковой полосе y=x.*cos(2*pi*fc*t)+ imag (hilbert(x))>* sin (2*pi*fc*t);

-fm - частотная модуляция y=cos(2*pi*fc*t+opt*cumsum(x)), где cumsum- прямоугольная аппроксимация интеграла от x. По умолчанию значение параметра opt=(fc/fs)*2*pi/max(max(x)). Максимальное отклонение частоты не превосходит fc;

-pm - фазовная модуляция y=cos(2*pi*fc*t+opt*x). По умолчанию opt= pi/max(max(x)). При этом максимальное отклонение по фазе не превышает π радиан;

-pwm – широтно-импульсная модуляция для элементов массива x, представленных значениями от 0 до 1, представляющая ширину импульса в относительных единицах к периоду. Применяется выравнивание импульсов слева. Для выравнивания по центру необходимо length(y)= length(x)*fs/fc;

-ptm – фазо-импульсная модуляция для массива подобного описанному для функции pwm (значения элементов дают время начало импульсов в долях от периода). Параметр opt задает длительность импульсов в долях периода. По умолчанию opt=0.1, length(y)= length(x)*fs/fc;

-qam – квадратурно-импульсная модуляция

y=x.*cos(2*pi*fc*t+opt.*sin(2*pi*fc*t). Здесь параметр opt должен представлять массив того же размера, что и x.

2) Демодуляция сигналов осуществляется функцией demod:

- x=demod(y, fc, fs, ‘method’ [, opt]);

- x=demod(y, fc, fs, ‘pwn’,‘centeredd’);

- [x1, x2]=demod(y, fc, fs, ‘qam’)

3) Пофрагментный вывод сигналов. Многие модулированные и иные сигналы имеют довольно сложную форму. Для ее детального анализа предназначена графическая функция **strips(x)**. Она строит график сигналов по фрагментам, содержащим по 250 отсчетов вектора x. Если x –матрица, то отображаются отдельными фрагментами ее столбцы. Вариант функции strips(x, sd, fs) позволяет задать отображение ряда фрагментов с длиной в sd секунд для сигнала с частотой дискретизации fs. А вариант strips(x, sd, fs, scale) позволяет задать еще и масштаб по вертикали scale.

Задание для самостоятельного выполнения

1. Сгенерировать сигналы, указанные в разделе моделирование сигналов.
2. Сгенерировать сигнал $x(t) = \sin\left(2 \cdot \pi \cdot 5 \cdot N_{вар} \cdot t\right) + 0.5 \cdot \sin\left(2 \cdot \pi \cdot 10 \cdot N_{вар} \cdot t\right), \ t \in [0,1]$. Получить вектор ДПФ сигнала. Построить графики исходного сигнала, а также АЧХ и ФЧХ сигнала.
3. Сгенерировать зашумленный сигнал. Построить график сигнала, периодограмму и спектрограмму сигнала.
4. Сгенерировать сигнал $y = t^2, \ t \in [0,2]$. Осуществить процесс перевыборки сигнала (изменить частоту дискретизации в 3/2 раза). Построить график.

5. Осуществить частотную модуляцию косинусоидального сигнала. Построить многофрагментный график модулированного сигнала.

Контрольные вопросы

1. Как осуществляется прямое и обратное преобразование Фурье, в том числе быстрое в системе MathCAD?
2. Что такое периодограмма и спектрограмма?
3. Как можно изменить частоту дискретизации сигнала?
4. Как осуществляется модуляция сигналов в системе MathCAD?
5. Как можно построить многофрагментный график модулированного сигнала?