Лабораторная работа № 9. Расчет цифровых фильтров в среде Matlab

Цель работы: провести исследования по применению фильтров для обработки речевых сигналов в системе Matlab.

1. Теоретические сведения

1.1. Расчет коэффициентов цифрового фильтра.

Для расчета коэффициентов $a_k \left(при\ \ k = 0, \ldots, N \right)$ цифрового фильтра (ЦФ), используются следующие расчетные формулы:

$$\mathrm{a}_k = \mathrm{a}_{-k} = \frac{\Omega_c}{\pi} \frac{\sin k\Omega_c}{k\Omega_c} = \frac{\Omega_c}{\pi} S_{\mathrm{a}}(k\Omega_c),$$

где $\Omega_c = \frac{\omega_c}{f_d} = 2\pi \frac{f_c}{f_d}$, $S_a(x) = \sin x/x$, ω_c – частота среза (иногда ее обозначают $\omega_{\hat{a}}$ и называют «верхняя граничная частота»), f_d – частота дискретизации и f_c – частота среза.

Рассмотрим пример ФНЧ 6-го порядка ($N = 3$) для f_c=25 Гц, f_d =100 Гц, коэффициенты фильтра: $\mathrm{a}_0 = 0{,}5$, $\mathrm{a}_1 = \mathrm{a}_{-1} = 1/\pi = 0{,}3183$, $\mathrm{a}_2 = \mathrm{a}_{-2} = 0$, $\mathrm{a}_3 = \mathrm{a}_{-3} = -1/3\pi = -0{,}1061$.

Интересно и полезно проверить, действительно ли эти коэффициенты с точностью до коэффициента $\Delta t = 1/f_d$ совпадают со значениями ИПХ $h(k\Delta t)$: $\mathrm{a}_k = \Delta t \cdot h(k\Delta t)$.

Обратное преобразование Фурье от функции $H(f) = rect\left(\frac{f}{2f_c}\right)$ дает выражение для ИПХ аналогового фильтра:

$$h(t) = 2f_c S_a\left(2\pi f_c t\right),$$

откуда следует

$$h(k\Delta t) = 2f_c\ S_a\left(2\pi f_c k\Delta t\right) = f_d \frac{\Omega_c}{\pi} S_a\left(k\Omega_c\right) = \frac{1}{\Delta t} a_k .$$

Сравнение позволяет обнаружить различие «по вертикали» ИПХ аналогового и цифрового фильтров.

Рассмотрим различие «по горизонтали».

Расстояние между нулями функции *h(t)*: $\Delta\tau = \dfrac{1}{2f_c}$. В рассмотренном примере было принято $f_c = f_d/4 = 1/4\Delta t$, откуда следует $\Delta\tau = 2\Delta t$. Действительно, коэффициент $a_2 = 0$. Ясно также, что все четные коэффициенты должны быть равны нулю.

Для иллюстрирования примера, построим график функции *h(t)* (рис. 1).

```
% График ИПХ непрерывного ФНЧ
dt=1;                          % шаг дискретизации;
dtau=2*dt;                     % расстояние между нулями;
fc=1/(2*dtau);                 % частота среза
t=-3*dtau:0.1:3*dtau;          % время
h=2*fc*sinc(2*fc*t);           % ИПХ
plot(t,h)                      % график
grid on            % сетка на графике
```

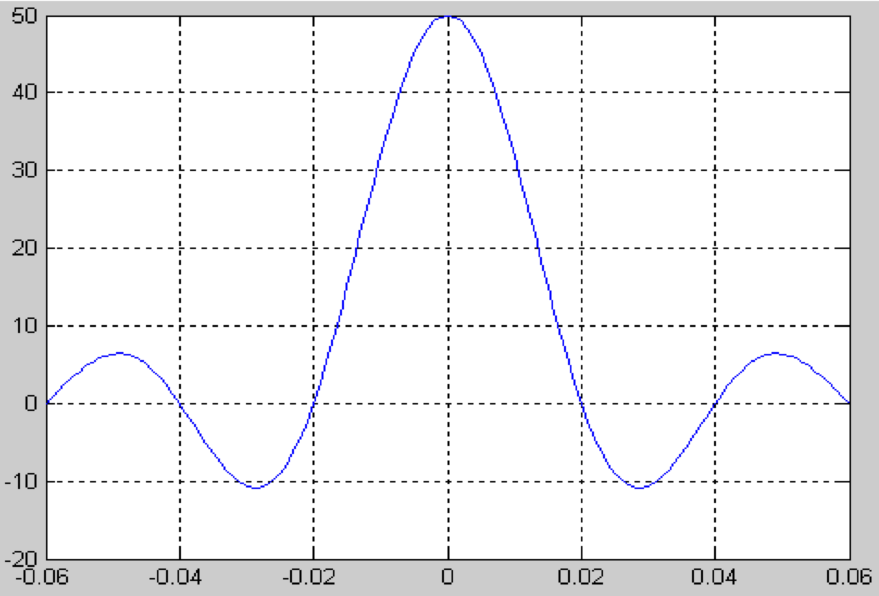

Рис. 8.4. График ИПХ непрерывного ФНЧ

Построим частотную характеристику рассчитанного фильтра (рис.2):

$$H_d(\omega) = a_0 + 2\sum_{k=1}^{N} a_k \cos\omega k\Delta t.$$

```
a0=0.5;
ak=[0.3183 0 -0.1061];     % коэффициенты фильтра с 1 по 3
```

```
dt=1;  N=3;  df=0.02;
f=-0.5:df:1.5;      % диапазон частот
sum=0;  for k=1:N,
     sum=sum+ak(k)*cos(2*pi*f*k*dt); end;
H=a0+2*sum;           % ЧХ;
plot(f,H);
grid on
```

Как видно из графика на рис. 8.5, частотная характеристика фильтра 3-го порядка существенно отличается от прямоугольника с относительной частотой среза $f_c / f_d = 1/4$. Полагая, что причина тому малый порядок фильтра, рассмотрим случай $N = 9$ и досчитаем недостающие коэффициенты ИПХ с нечетными номерами (коэффициенты с четными номерами равны нулю) (рис.8.6):

$a_5 = 0{,}0637$; $a_7 = -0{,}0455$; $a_9 = 0{,}0354$.

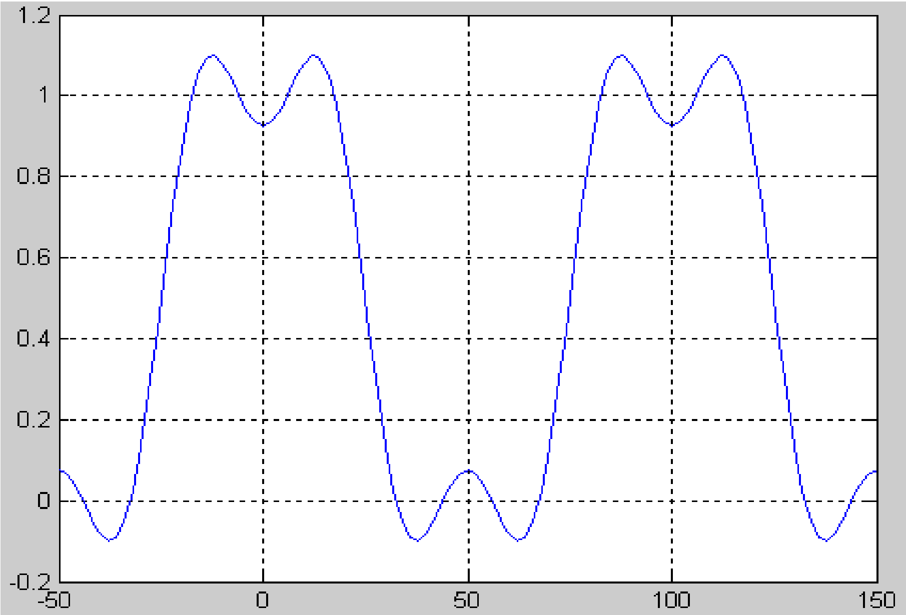

Рис. 8.6. Частотная характеристика ФНЧ 3-го порядка (N = 3)

```
% Частотная характеристика ФНЧ
a0=0.5; ak=[0.3183 0 -0.1061 0 0.0637 0 -0.0455 0 0.0354]; % коэффициенты
фильтра с 1 по 9
dt=1; N=9; df=0.02; f=-0.5:df:1.5;
for k=1:N,
      sum=sum+ak(k)*cos(2*pi*f*k*dt);
end
H=a0+2*sum;    plot(f,H);   grid on
```

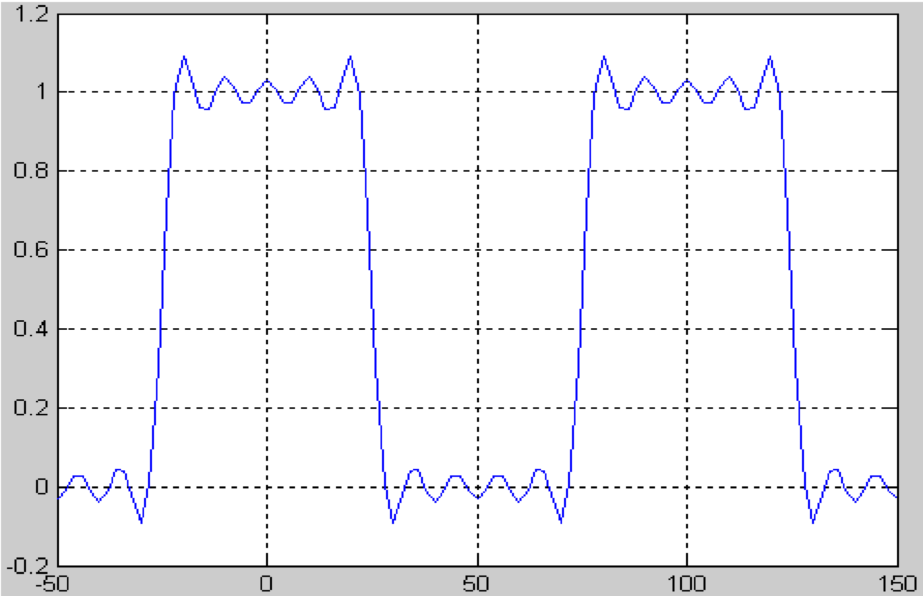

Рис. 8.7. Частотная характеристика ФНЧ 3-го порядка (N=9)

Как следует из рис.8.7, ЧХ действительно становится более прямоугольной, но степень осцилляций слева и справа частоты среза не уменьшается.

Применим треугольное окно к ИПХ фильтра 18-го порядка ($N = 9$) (рис.8.8).

```
% Частотная характеристика ФНЧ
a0=0.5; ak=[0.3183 0 -0.1061 0 0.0637 0 -0.0455 0 0.0354];  fd=100;  dt=1/fd;
N=9; r=1:N;
k(r)=ak(r).*(1-r/N);  df=0.02*fd;   f=-0.5:df:1.5;
sum=0; for k=1:N,
sum=sum+ak(k)*cos(2*pi*f*k*dt); end
H=a0+2*sum;
plot(f,H); grid on
```

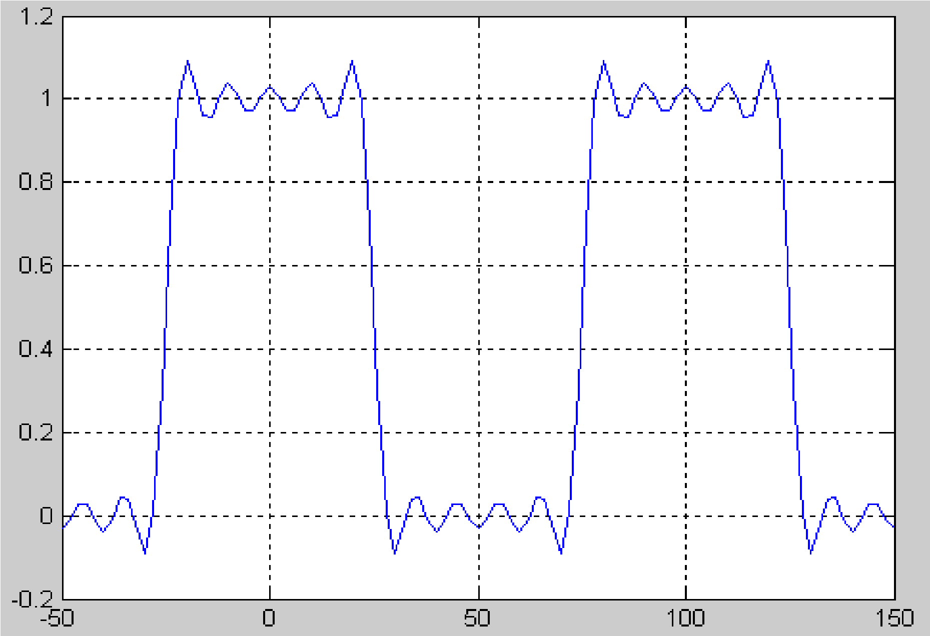

Рис.8.8. Частотная характеристика ФНЧ 18-го порядка (*N*=*9*)

Как и следовало ожидать, осцилляции уменьшились, однако за это пришлось «заплатить» уменьшением крутизны склонов ЧХ в районе частоты среза.

1.2. Расчет цифровых фильтров в среде Matlab

В среде MATLAB цифровые фильтры можно рассчитывать, по меньшей мере, тремя способами: из командного окна; с помощью пакета *sptool*; с помощью пакета *fdatool*.

1. *Расчет цифровых фильтров из командного окна*

Функция *fir1* реализует вычисления по методу обратного преобразования Фурье с использованием окон

a=firl(n,Wn,’ftype’,window,’normalization’),

где *n* – порядок фильтра (количество коэффициентов равно *n+1*), его лучше задавать четным (так как для некоторых типов фильтров нечетное *n* хотя и можно задавать, но результат будет такой, как если бы задавался порядок на единицу больше);

Wn – относительная частота среза (по отношению к частоте Найквиста, которая принимается равной единице); представляет собой вектор из двух чисел, если фильтр полосовой или режекторный; является вектором из *m* пар чисел, если фильтр многополосный, из *m* полос;

‘ftype’ – тип фильтра (если отсутствует – ФНЧ; ‘high’ – ФВЧ; ‘stop’ – режекторный; ‘DC-1’ – многополосный пропускающий; ‘DC-0’ – многополосный режекторный);

window – вектор-столбец из *n+1* элементов (по умолчанию применяется окно Хэмминга hamming(n+1));

‘normalization’ – нормировка ИПХ (по умолчанию значение ‘scale’ – единичное значение коэффициента передачи в центре полосы пропускания; ‘noscale’ – нормировка не производится).

Пример:

% расчет коэффициентов КИХ-фильтра с нормализацией.

window=rectwin(7) % синтез прямоугольного окна из 7 отсчетов

a=fir1(6,0.5,window);

Результат:

a =[– 0,1148 0,0000 0,3443 0,5409 0,3443 0,0000 – 0,1148].

Сравнивая эти результаты с рассчитанными вручную коэффициентами, нетрудно увидеть разницу. Например, ручные расчеты дают $a_0=0,5$, тогда как в Matlab мы получили $a_0 = 0,5409$. Естественно предположить, что причиной тому проводимая по умолчанию нормировка ИПХ. Проверяем это предположение, задавая в программе значение ’noscale’ для параметра нормализации:

window=rectwin(7); a=fir1(6,0.5,window,’noscale’);

Результат:

a = [– 0,1061 0,0000 0,3183 0,5000 0,3183 0,0000 – 0,1061].

2. *Расчет цифровых фильтров с помощью пакета sptool.*

Для активизации пакета нужно в командном окне набрать команду sptool или открывать его следующим образом: start → toolboxes → More...→ Signal Processing → Signal Processing tool (sptool); Затем в появившемся окне в колонке кнопок Filters нажать кнопку New.

В появившемся окне Filter Designer:

- задать частоту дискретизации 100 Гц;

- выбрать в позиции Design Method выбрать FIR - значение Kaiser Window FIR (выбираем из 3-х вариантов: Equiripple FIR, Least Square FIR и Kaiser Window FIR);

- отключить флажок Minimum Order; задать Order=6; задать Type=lowpass; задать Passband Fc = 25;

- отключить флажок Autodesigne; закрыть окно Filter Designer;

- в окне sptool в колонке Filters нажать кнопку View;

- в появившемся окне Filter Viewer наблюдаем графики АЧХ, ФЧХ, ИПХ (ИПХ наблюдаем после активизации соответствующего флажка).

Мы выбрали в позиции Design Method значение Kaiser Window FIR. Кроме данного алгоритма, есть еще два алгоритма: Equiripple FIR, Least Square FIR. Из всех трех алгоритмов только алгоритм Кайзера реализует метод обратного преобразования Фурье с весовым окном Кайзера. При значении параметра «0» окно Кайзера превращается в обычное прямоугольное окно.

Как показывает эксперимент, рассчитанные таким образом коэффициенты ФНЧ оказываются ненормированными, т.е. в точности равными вычисленным вручную.

Чтобы узнать значения коэффициентов, нужно:

- активизировать график ИПХ, щелкнув по нему мышкой;
- активизировать вертикальные маркеры (кнопкой, расположенной под меню);
- поместить один из маркеров (всего имеется 2 маркера – 1-й изображается сплошной вертикальной линией. 2-й - пунктирной) напротив нужного отсчета ИПХ;
- рассчитать значение отсчета ИПХ в специальном окне.

Другой способ, в зависимости от версии Matlab - нажать на специальную кнопку Filter coefficients.

3. Расчет фильтра $\{a_k\}$ с помощью пакета fdatool

Для активизации пакета нужно в командном окне набрать команду fdatool. Затем в появившемся окне в разделе Designe Filter задать:

- Designe Method: FIR=Window;
- Window Specifications: Window=Rectangular;
- Filter order: Specify order=6;
- частоту дискретизации (Fs=100 Гц);
- Filter Type=lowpass;
- Passband Fc=25;
- с помощью кнопок под меню включаем режим просмотра коэффициентов фильтра.

В результате проведенных расчетов убеждаемся, что здесь по умолчанию производится нормирование ИПХ (см. выше fir1).

2. Задания и методические указания по выполнению работы

Необходимо выполнить следующее:

1. С помощью микрофонной гарнитуры введите в компьютер речевой сигнал (Фамилию студента) с параметрами: длительность – несколько секунд; частота дискретизации 11025 Гц. Для ввода сигнала в компьютер удобно использовать программу «Звукозапись» из раздела «Стандартные-Развлечения».
2. С помощью команды y = wavread('filename') импортируйте речевой сигнал в среду Matlab, постройте график сигнала с помощью команды plot.

3. Запустите программу sptool и импортируйте сигнал y в её среду.

4. Визуализируйте и прослушайте введенный сигнал с помощью инструментария программы sptool**.**

5. Синтезируйте НЧ фильтр с окном Кайзера: минимального порядка; граничная частота полосы пропускания 1500 Гц; неравномерность в полосе пропускания 1 дБ; граничная частота полосы задержания 2 000 Гц; минимальное затухание в полосе задержания –40 дБ.

6. Примените синтезированный НЧ фильтр к вашему речевому сигналу.

7. Прослушайте сигнал, полученный в результате НЧ фильтрации, постройте его график и ответьте на вопросы:

а) как и почему изменилось звучание речевого сигнала после фильтрации?

б) что произойдет, если граничные частоты полос пропускания и задержания уменьшить вдвое? Экспериментально проверьте свои предположения.

8. Подберите оптимальные, с вашей точки зрения, параметры НЧ фильтра. Попробуйте обосновать свой выбор.