

Министерство образования и науки Российской Федерации

ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ (ТУСУР)

КАФЕДРА ТЕЛЕКОММУНИКАЦИЙ И ОСНОВ РАДИОТЕХНИКИ
(ТОР)

УТВЕРЖДАЮ
Зав. кафедрой ТОР

_____ А.Я. Демидов

Методические указания к лабораторным работам

по дисциплине «Цифровая обработка сигналов» (5 семестр)
специальности «Инфокоммуникационные технологии и системы связи»
(код 210700.62)

РАЗРАБОТЧИКИ:

_____ А.А. Гельцер,
доцент каф. ТОР

_____ Р.Р. Абенов,
аспирант каф. ТОР

_____ Е.В. Рогожников,
аспирант каф. ТОР

Содержание

ВВЕДЕНИЕ	2
ЛАБОРАТОРНАЯ РАБОТА №1.	2
ЛАБОРАТОРНАЯ РАБОТА №2	9
ЛАБОРАТОРНАЯ РАБОТА №3	12
ЛАБОРАТОРНАЯ РАБОТА №4	16
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	23
ПРИЛОЖЕНИЕ	24

ВВЕДЕНИЕ

Лабораторные работы по дисциплине «Цифровая обработка сигналов» предназначены для освоения системы математических вычислений Scilab. Эта система похожа на систему MatLab, но в отличие от нее она является открытой и бесплатной. Пакет программ системы Scilab и ее расширения можно найти в интернете. Причем пакет постоянно обновляется и совершенствуется.

Имеются подробные руководства и книги на русском языке по системе Scilab [1,2], многие из которых можно найти также в сети интернет. Для овладения возможностями всего пакета при математических расчетах потребуется значительное количество времени. В то же время при решении задач в студенческих работах часто необходимо выполнять лишь простые вычисления и построения графиков. Быстро освоить приемы выполнения таких задач (от простых к более сложным) можно изучая примеры программ, которые приведены в каждой лабораторной работе. Примеры содержат подробные пояснения и соответствуют направлению индивидуальных работ по дисциплине «Цифровая обработка сигналов».

Кроме этого желательно дополнительно знакомиться с соответствующими разделами книг по системе Scilab [2].

Результаты выполненных лабораторных работ оформить в виде отчета, содержащего коды программ, графики и пояснения к ним.

ЛАБОРАТОРНАЯ РАБОТА №1.

Изучение основ работы в пакете Scilab

Задание на работу:

Часть №1. Построение графиков функций.

Цель: научиться создавать программы построения графиков функций (от простых к более сложным).

Пример 1. Построить график косинусоиды $y = \cos x$, $0 \leq x \leq 2\pi$.

Ниже даны пояснения, комментарии (курсив) и коды программы, написанные шрифтом Courier New.

```
// - это символ комментария. Замечание: для
// разделения целого и десятичной дроби используется
// не запятая, а точка.
x=[0:0.1:2*%pi]; //%pi - число  $\pi$ ; x определяется в
// виде массива точек равномерного разбиения
// сегмента [0;2*%pi] с шагом 0.1 (или вектора с
// компонентами 0,0.1,0.2,...,6.2 ). В конце команды
// ставим «;», иначе произойдет вывод в командном
// окне всех компонент вектора x
plot(x,sin(x)) // строим простой график (simple
// plot)
```

Эту программу (две строки) можно набрать в командном окне и запустить на выполнение нажатием клавиши Enter. Появится графическое окно с графиком косинусоиды. Для удобства программирования в Scilab предусмотрен текстовый редактор SciNotes, который находится во вкладке «Инструменты». После написания скрипта его можно запустить нажатием кнопки F5 (или кнопкой «Сохранить и выполнить» на панели инструментов). Это окно удобно использовать для сохранения программы в виде отдельного файла с каким-либо именем и расширением .sce. Чтобы перед запуском написанного кода автоматически удалять все открытые графические окна, необходимо в начале кода вписать строку:

```
xdel(winsid()); //закрываем все графические окна
```

Чтобы очистить командное окно, используется команда clc, которую тоже удобно писать в самом начале кода.

Пример 2. Построить графики функций $y=\cos x$ и $y=\sin 2x$ на одном рисунке.

```
x=[0:0.1:2*%pi];
plot(x, cos(x), x, sin(2*x)) //построение двух
// графиков в одном окне
```

Пример 3. Назначение правого положения оси y

```
x=[0:0.1:2*%pi];
plot(x, sin(x))
a=gca(); //присвоение символу «a» роли оператора
// назначения
```

```
a.y_location = "right"; // назначение на графике
// правого положения оси y
```

Индивидуальное задание.

Построить графики функции $y = y(x)$ при различных положениях оси y , если $y_1=x^3$, $y_2=3x^2+1$, $y_3=\text{arctg}(x)$, $y_4=\sin(x^2)$, $y_5=\cos(2x+3)$, $y_6=\lg(4-x)$, $-2 \leq x \leq 3$.

Таблица 1. Индивидуальное задание №1 по вариантам.

Вариант	1	2	3	4	5	6	7	8	9
Y	y_1+y_2	y_1+y_3	y_1+y_4	y_1+y_5	y_2+y_3	y_2+y_4	y_2+y_5	y_3+y_4	y_3+y_5
Вариант	10	11	12	13	14	15	16	17	18
Y	$y_1 \times y_2$	$y_1 \times y_3$	$y_1 \times y_4$	$y_1 \times y_5$	$y_2 \times y_3$	$y_2 \times y_4$	$y_2 \times y_5$	$y_3 \times y_4$	$y_3 \times y_5$
Вариант	19	20	21	22	23	24	25	26	27
y	y_4+y_5	$y_4 \times y_5$	y_1+y_6	y_2+y_6	y_3+y_6	y_4+y_6	y_5+y_6	$y_1 \times y_6$	$y_2 \times y_6$

Часть №2. Построение усложненных графиков функций.

Цель: научиться создавать программы построения более сложных графиков функций.

Пример 4. Формирование кривой графика функции определенного вида

```
x=1:10; //x представлен вектором с компонентами
// 1, 2, 3, ..., 10 plot(x, x.*x, '*суа--')
```

Здесь `.*` – знак поэлементного способа умножения x на x , в результате получаем вектор x^2 с компонентами 1, 4, 9, ..., 100;

`'*суа--'` – эта запись относится к области `LineStyle` и задает вид кривой графика. Первый символ записи `*` внутри кавычек характеризует тип маркера кривой (описание маркеров находится в соответствующей таблице ПРИЛОЖЕНИЯ). Запись «суа» определяет цвет кривой (цвет выбирается в соответствующей таблице ПРИЛОЖЕНИЯ 1), символ «`--`» задает тип линии.

Пример 5. Задание толщины всех линий на графике

```
x=[0:0.1:2*%pi];
xset("thickness",2); //установка толщины всех линий
```

```
// графика выбором числа (число 2 задает линии
// толщиной в 2 пикселя)
plot(x, sin(x))
```

Пример 6. Задание ширины и высоты графического окна

```
f4=scf(4); //функция scf создает графическое окно с
// номером 4 и делает
//его текущим
f4.figure_size = [600, 400]; //установка ширины и
// высоты графического окна
// с номером 4 в единицах экрана. Если монитор имеет
// разрешение 1024x768,
// то оно может быть максимальным размером
// графического окна.
plot() //вывод некоторых графиков по умолчанию
clf(f4); очистка графического окна с номером 4
dim=xget("wdim"); //вывод в командном окне размера
// текущего графического окна
```

Индивидуальное задание.

Построить графики трех функций в одном окне, выбирая для них оригинальные маркеры, цвета, типы и толщины линий. Функции брать из лабораторной работы №1. Причем, если $y=y_1+y_2$, то y_1 – первая функция, y_2 – вторая функция, y – третья функция.

Часть №3. Построение графиков непериодических и периодических функций (сигналов).

Цель: научиться создавать программы построения кусочно-заданных непериодических и периодических сигналов (функций).

Пример 7. Представление сигналов по условию первой задачи индивидуального задания

```
E = 1; //амплитуда сигнала
Ts = 1; //промежуток времени наблюдения сигнала
//Представление данного сигнала в виде кусочно-заданной
//функции
Ts=1; E=1;
function s = sz(t) //function - endfunction -
// оператор создания
// функции
if ((t >=0) & (t <=2*Ts/3)) then s = -E;
```

```

//условный оператор «если-то»
elseif ((t >2*Ts/3) & (t <=Ts)) then s =
Е; //условный оператор «иначе-если-то»
end endfunction
////////// Вывод графика заданного сигнала//////////
t = [0: Ts/100: Ts];
clf(); //очистка графического окна с номером 0 (по
// умолчанию)
plot(t,sz); // построение графика заданного сигнала
zoom_rect([min(t),-1.1,max(t),1.1]); //установка
// границ окна вдоль осей координат оператором
// zoom_rect[xmin,ymin,xmax,ymax]
xlabel('Grafic sz(t)', 't', 'sz(t)', boxed=1);
// отображение названий графика, оси абсцисс, оси
// ординат;
// помещение названий в рамку только при
// boxed=1
xgrid(); //установка сетки в графическом окне
///// Формирование периодического сигнала и вывод его графика/////
t=0:Ts/100:3*Ts;
y = t-(floor(t/Ts))*Ts; //y - линейно изменяющаяся от
// 0 до Ts периодическая переменная с периодом Ts;
// floor() - операция выделения наибольшего целого,
// меньшего аргумента
for i = 1:length(y); //for-end - оператор цикла,
// length(y) - определяет количество компонент
// вектора y
a=sz(y(i)); //дискретные значения периодически
// продолженного заданного сигнала
sp(i)=a; // периодически продолженный заданный сигнал
end
scf(1); // функция scf создает графическое окно с
// номером 1 и делает его текущим
clf(1);
plot(t,sp); //вывод периодически продолженной функции
// (периодической функции)
zoom_rect([min(t),-1.1,max(t),1.1]);
xgrid();
/// Формирование непериодического сигнала и вывод его графика ///
t=(-2*Ts:Ts/100:3*Ts);
function f = s0(t) // задание непериодической функции
// (заданной функции, продолженной нулями)
if ((t <0) | (t > Ts)) then f = 0;
else f = sz(t);

```

```

end
endfunction
scf(2);
clf(2);
plot(t,s0); // вывод непериодической функции
zoom_rect([min(t), -1.1, max(t), 1.1]);
xgrid();
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

Индивидуальное задание.

Построить графики заданной функции, ее непериодического и периодического продолжения. Исходные данные взять из таблицы.

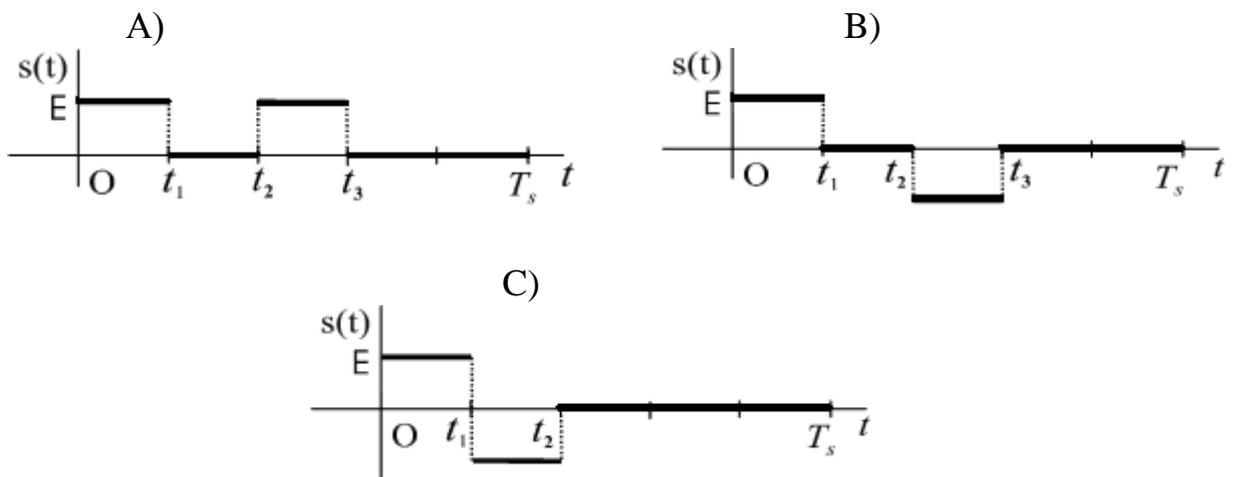


Рисунок 1. Форма сигналов

$$t_i = k_i(T_s/5)$$

k – формат сигнала (А, В или С), k_i – коэффициенты

Таблица 2. Индивидуальное задание №3 по вариантам.

Вариант	Шифр сигнала $k k_1 k_2 k_3$	Вариант	Шифр сигнала $k k_1 k_2 k_3$
1	A 1 2 3	16	B 1 4 5
2	A 1 2 4	17	B 2 3 4
3	A 1 2 5	18	B 2 3 5
4	A 1 3 4	19	B 2 4 5
5	A 1 3 5	20	B 3 4 5
6	A 1 4 5	21	C 1 2
7	A 2 3 4	22	C 1 3
8	A 2 3 5	23	C 1 4
9	A 2 4 5	24	C 1 5
10	A 3 4 5	25	C 2 3
11	B 1 2 3	26	C 2 4
12	B 1 2 4	27	C 2 5
13	B 1 2 5	28	C 3 4
14	B 1 3 4	29	C 3 5
15	B 1 3 5	30	C 4 5

ЛАБОРАТОРНАЯ РАБОТА №2

Шумы квантования в АЦП – оценка и анализ

Одна из наиболее существенных составляющих ошибки при измерениях с помощью АЦП - погрешность квантования – является результатом самого процесса преобразования. Погрешность квантования - это погрешность, вызванная значением шага квантования и определяемая как $\frac{1}{2}$ величины наименьшего значащего разряда (LSB) [3]. Она не может быть исключена в аналого-цифровых преобразованиях, так как является неотъемлемой частью процесса преобразования, определяется разрешающей способностью АЦП и не меняется от АЦП к АЦП с равным разрешением.

Передаточная характеристика АЦП - это функция зависимости кода на выходе АЦП от напряжения на его входе. Такой график представляет собой кусочно-линейную функцию из $2N$ "ступеней", где N - разрядность АЦП. Каждый горизонтальный отрезок этой функции соответствует одному из значений выходного кода АЦП (см. рис. 7). Если соединить линиями начала этих горизонтальных отрезков (на границах перехода от одного значения кода к другому), то идеальная передаточная характеристика будет представлять собой прямую линию, проходящую через начало координат.

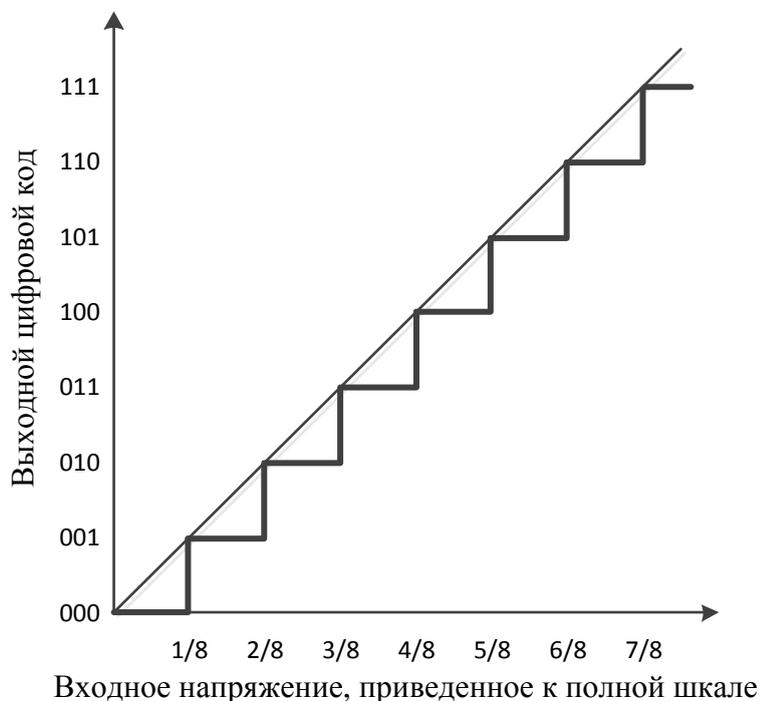


Рисунок 2. Идеальная передаточная характеристика 3-разрядного АЦП

Рис. 2 иллюстрирует идеальную передаточную характеристику для 3-х разрядного АЦП с контрольными точками на границах перехода кода. Выходной код принимает наименьшее значение (000b) при значении входного сигнала от 0 до 1/8 полной шкалы (максимального значения кода этого АЦП). Также следует отметить, что АЦП достигнет значения кода полной шкалы (111b) при 7/8 полной шкалы, а не при значении полной шкалы. Таким образом, переход в максимальное значение на выходе происходит не при напряжении полной шкалы, а при значении, меньшем на наименьший значащий разряд (least significant bit, LSB), чем входное напряжение полной шкалы. Передаточная характеристика может быть реализована со смещением $-1/2$ LSB. Это достигается смещением передаточной характеристики влево, что смещает погрешность квантования из диапазона $-1 \dots 0$ LSB в диапазон $-1/2 \dots +1/2$ LSB.

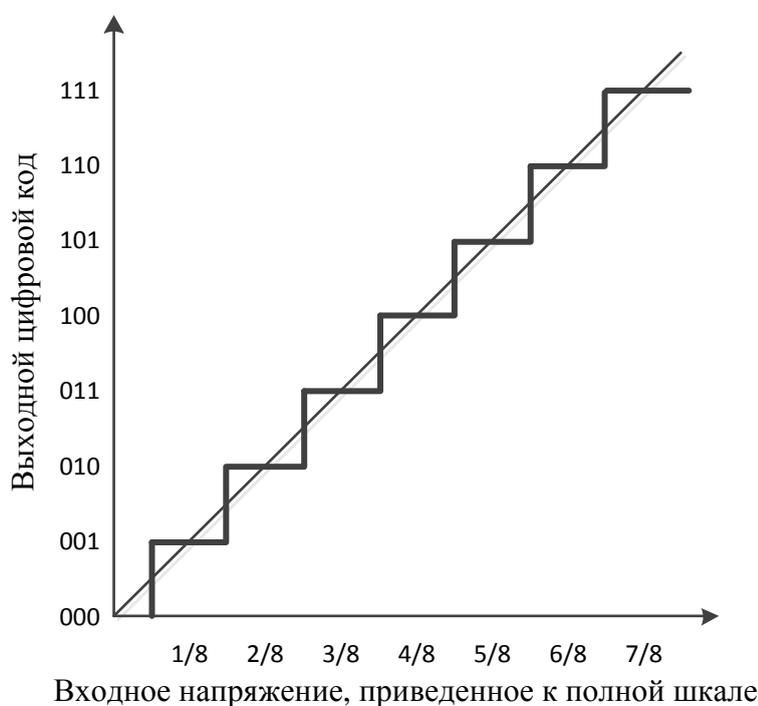


Рисунок 3. Передаточная характеристика 3-разрядного АЦП со смещением на $-1/2$ LSB для уменьшения погрешности квантования на $1/2$ LSB.

Задание на работу:

Часть 1. Формирование опорного сигнала

Сформировать опорный гармонический сигнал $s_1(t)$, амплитудой 0,5 В, со средним значением 0,5 В, с частотой N кГц, при этом корректно выбрать частоту дискретизации сигнала.

Число периодов в наблюдаемой реализации – не менее 10 и не более 100. Привести расчёт количества точек для выбранного (целого) количества периодов сигнала.

Указать частоту дискретизации, длительность реализации сигнала, шаг между отсчётами в частотной и во временной области.

Часть 2. Квантование сигнала

Устройство квантования имеет разрядность 4 бита. Пропустить сигнал, полученный в п. 1.1 через квантователь. Привести сигнал до квантования и после на одном графике.

Найти и построить сигнал ошибки между исходным сигналом и квантованным. Построить исходный и сигнал ошибки на одном графике.

Построить спектр опорного сигнала и спектр квантованного сигнала (в логарифмическом масштабе). Сравнить степень зависимости.

Найти статистические характеристики сигнала ошибки (шума квантования) – среднее значение и СКО. Найти отношение сигнал/шум квантованного сигнала. Сравнить его с расчётным значением.

Пример 7. Построение амплитудного спектра сигнала

```
clc
xdel(winsid())
Tm=5; // Длительность сигнала (с)
Fd=512; // Частота дискретизации (Гц)
Ak=0.5; // Постоянная составляющая (В)
A=1; // Амплитуда синусоиды (В)
F=13; // Частота сигнала (Гц)
Phi=0; // Начальная сигнала (Рад)
T=0:1/Fd:Tm; // Массив отсчетов времени
N=length(T); // Количество точек в массиве отсчетов
// времени
Fs=0:Fd/N:Fd-Fd/N; // Массив отсчетов частоты
Signal=Ak+A*sin(2*%pi*F*T);
///// Спектральное представление сигнала /////
FftS=abs(fft(Signal)); // Амплитуды преобразования
```

```
// Фурье сигнала
FftS=2*FftS./N;// Нормировка спектра по амплитуде
FftS(1)=FftS(1)/2;// Нормировка постоянной
// составляющей в спектре
figure// Создание нового графического окна
clf// Очистка графического окна
plot(Fs(1:200),FftS(1:200))// Построение первых 200
// точек амплитудного спектра
```

Для моделирования квантователя сигнала следует использовать условный оператор «если-то», который был описан в первой работе.

Часть 3. Исследование шумов квантования

Повторить пункт 2.1, 2.3, 2.4 для квантователей с разрядностью 8, 12, 16 бит. Построить зависимость отношения сигнал/шум от разрядности квантователя (от 4 до 16 бит).

Сделать выводы о природе шума квантования, его влиянии на квантуемый сигнал и мер по его уменьшению.

ЛАБОРАТОРНАЯ РАБОТА №3

Многоскоростная обработка сигналов

Децимацией (прореживанием, сокращением) цифровых данных принято называть уплотнение данных с удалением избыточной информации. Последнее имеет место, если шаг дискретизации данных был установлен излишне подробным и сигнала. Информация высокочастотной части сигнала может быть ненужной, если основная энергия полезной части сигнала заключена в низкочастотной области. Децимация может потребоваться и в том случае, если массивы данных представлены с разным шагом дискретизации [4].

На рисунке 4а показан процесс децимации в 2 раза без использования ФНЧ, в результате высокочастотные гармоники проявляются в децимированном сигнале. Использование ФНЧ 2б позволяет устранить гармоники до децимации и они не проявятся. Очевидно, что чем выше коэффициент подавления ФНЧ в полосе заграждения, тем меньше будет наложение.

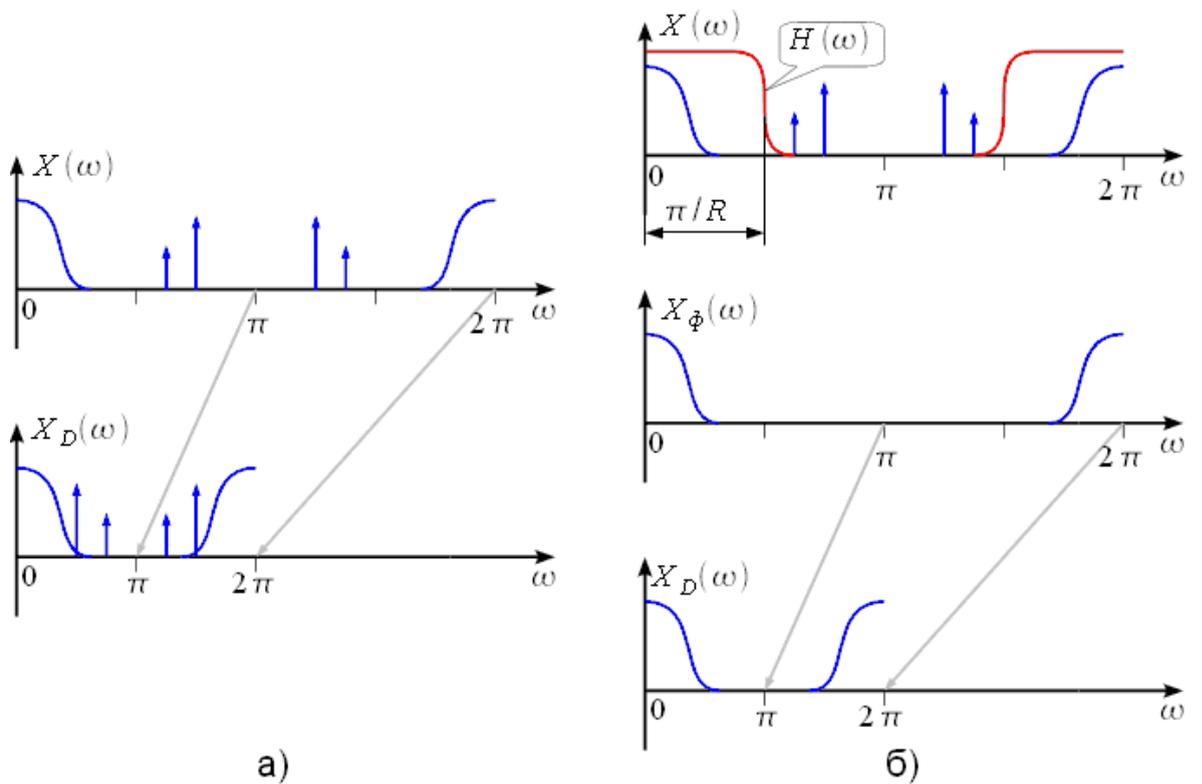


Рисунок 4: Наложение и его устранение при помощи ФНЧ

Рассмотрим теперь использование фильтров в задаче интерполяции сигналов. Фильтр интерполятор обеспечивает увеличение частоты дискретизации сигнала в R раз, где R — коэффициент интерполяции. Схема фильтра интерполятора и пояснение принципа его работы представлена на рисунке 4.

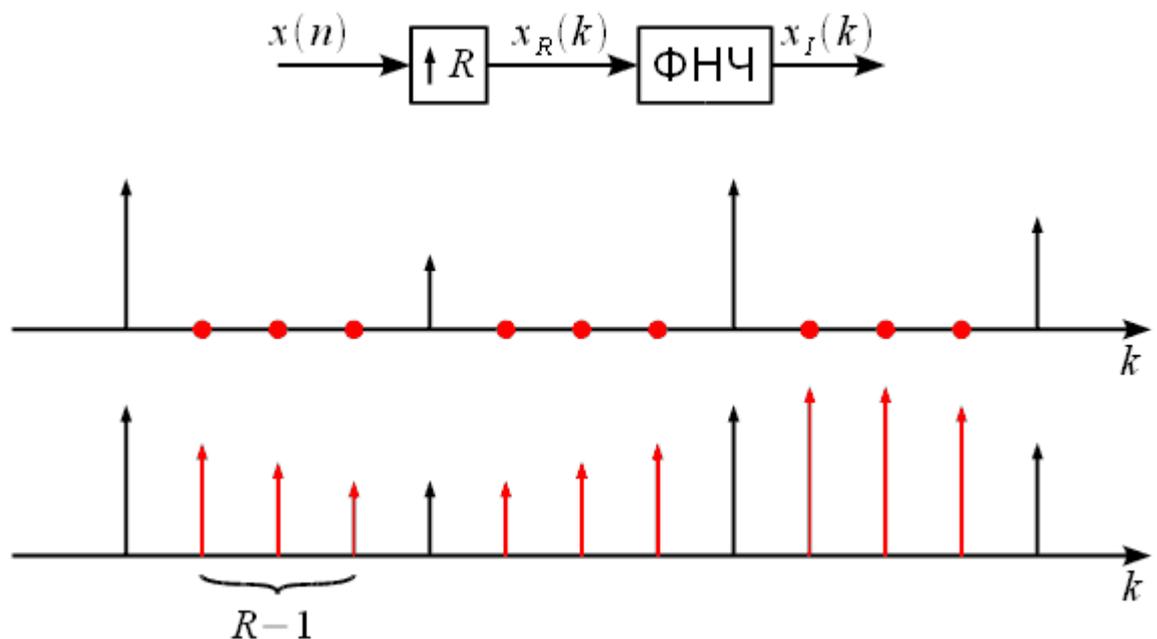


Рисунок 5: Фильтр интерполятор и принцип его работы

Исходный сигнал $x(n)$ (на графиках рисунка 8 показанный черным) поступает на интерполятор, который увеличивает частоту дискретизации в R раз, путем вставки $R-1$ нулей между отсчетами исходного сигнала (на верхнем рисунка 8 графике показаны красными точками). В результате получается сигнал $x_R(k)$, дополненный нулями, который поступает на ФНЧ и который «поднимает нули». В частотной области данный процесс можно представить, как показано на рисунке 5.

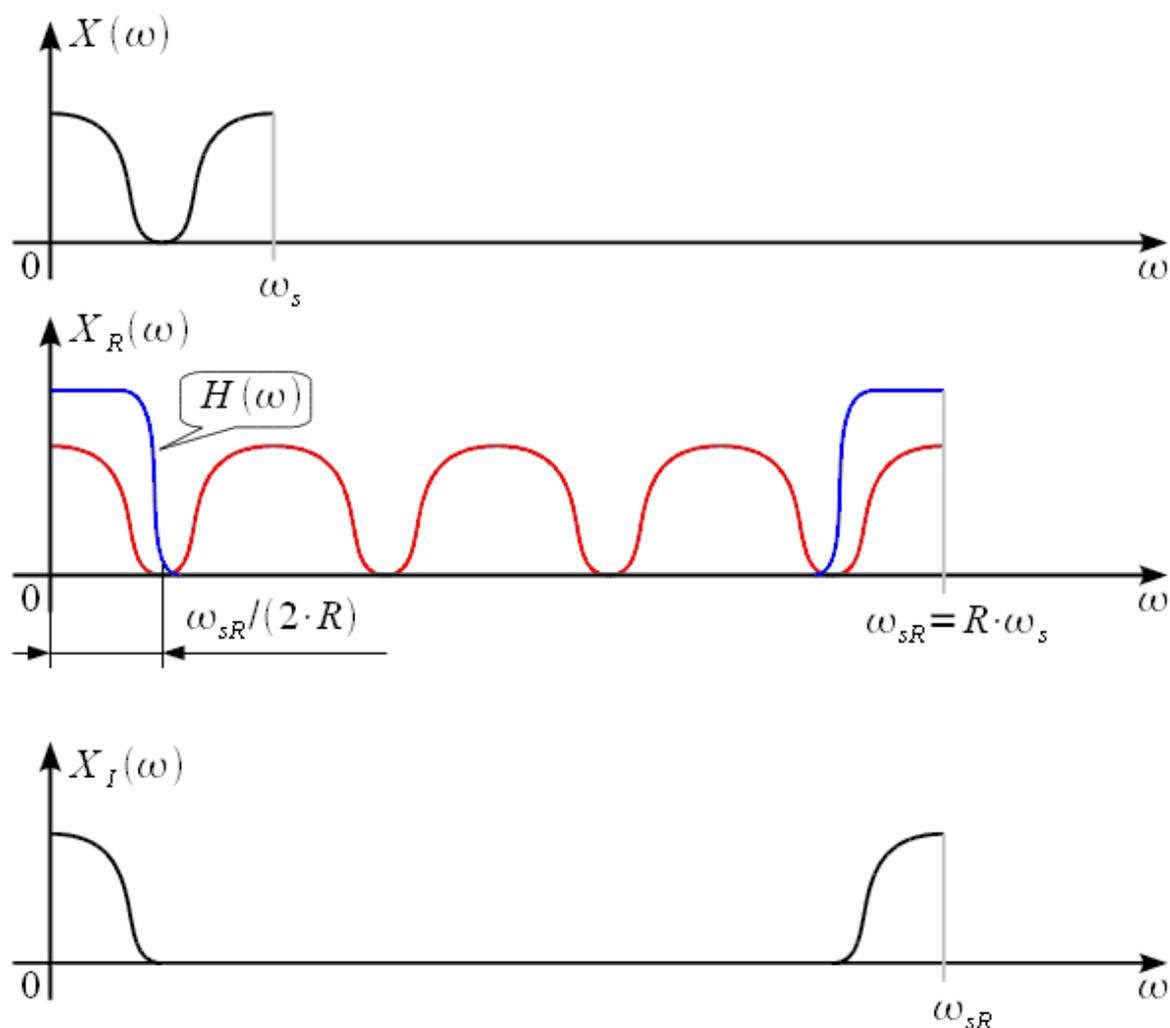


Рисунок 6: Процесс интерполяции в частотной области

Задание на работу:

Часть 1. Формирование опорного сигнала

Сформировать опорный сигнал $s(t)$, представляющий собой сумму гармонических сигналов с частотами N кГц, $N+3$ кГц и $N+6$ кГц (N – номер варианта) и амплитудой 1 В. Частота дискретизации равна 1 МГц.

К опорному сигналу добавить сигнал помехи, состоящий из гармонических сигналов с частотами 75, 80, 85, 90 и 95 кГц и амплитудой 0,5 В.

Число периодов в наблюдаемой реализации (по максимальной частоте опорного сигнала) – не менее 10 и не более 100. Привести расчёт количества точек для выбранного (целого) количества периодов сигнала. Построить спектр сигнала. Указать полосу частот полезного сигнала, длительность реализации сигнала, шаг между отсчётами в частотной и во временной области.

Часть 2. Децимация

2.1 Провести децимацию сформированного сигнала с шагом 10. Показать на графике спектр децимированного сигнала.

2.2 Пропустить опорный сигнал через цифровой фильтр с помощью функции `ffilt`. Провести децимацию сигнала с шагом 10. Показать на графике спектр децимированного сигнала после фильтрации.

Часть 3. Интерполяция

3.1 Провести интерполяцию полученного сигнала с повышением частоты дискретизации до 2 МГц. Построить спектр.

3.2 Пропустить сигнал через цифровой фильтр, при этом выбрать необходимую частоту среза. Показать на графике спектр полученного сигнала после фильтрации. Построить в одном графическом окне сигнал до фильтрации и после.

Часть 4. Исследование многокаскадной децимации и интерполяции

4.1 Повторить пункт 2.1, 2.2, 3.1 и 3.2 с помощью двух каскадов, при этом оптимальным образом выбрать шаги децимации и интерполяции, а также фильтры защиты от наложения спектра для каждого каскада.

Пример 8. Децимация производится с помощью простых операций над матрицами:

```

clc
xdel(winsid())
Fd=512;// Частота дискретизации (Гц)
T=0:1/Fd:Tm;// Массив отсчетов времени
A=0.5;// Амплитуда синусоиды (В)
Signal=A*sin(2*%pi*F*T); // Сигнал
figure
clf
plot2d3(T(1:60),Signal(1:60));// построение сигнала
// в виде прямых линий

```

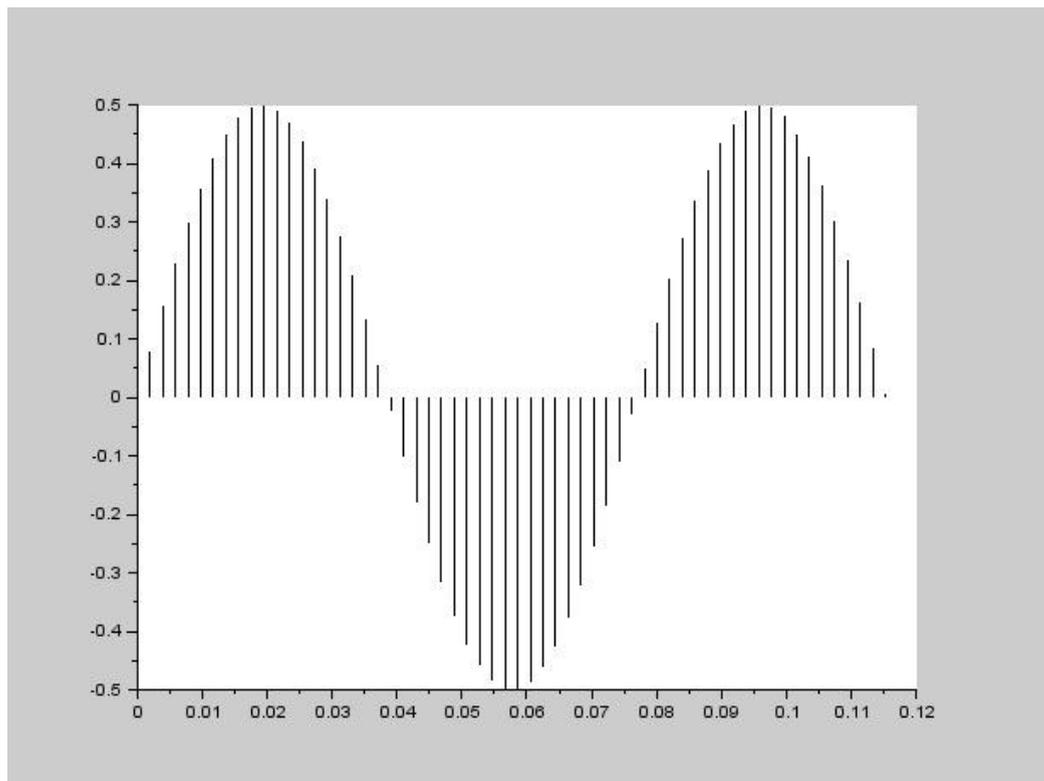


Рисунок 6. Форма сигнала до децимации

```

Signal_dec=Signal(1:5:60);// децимация сигнала в 5
// раз
figure
clf
plot2d3(T(1:5:60),Signal_dec(1:12));// построение
// сигнала, децимированного в 5 раз

```

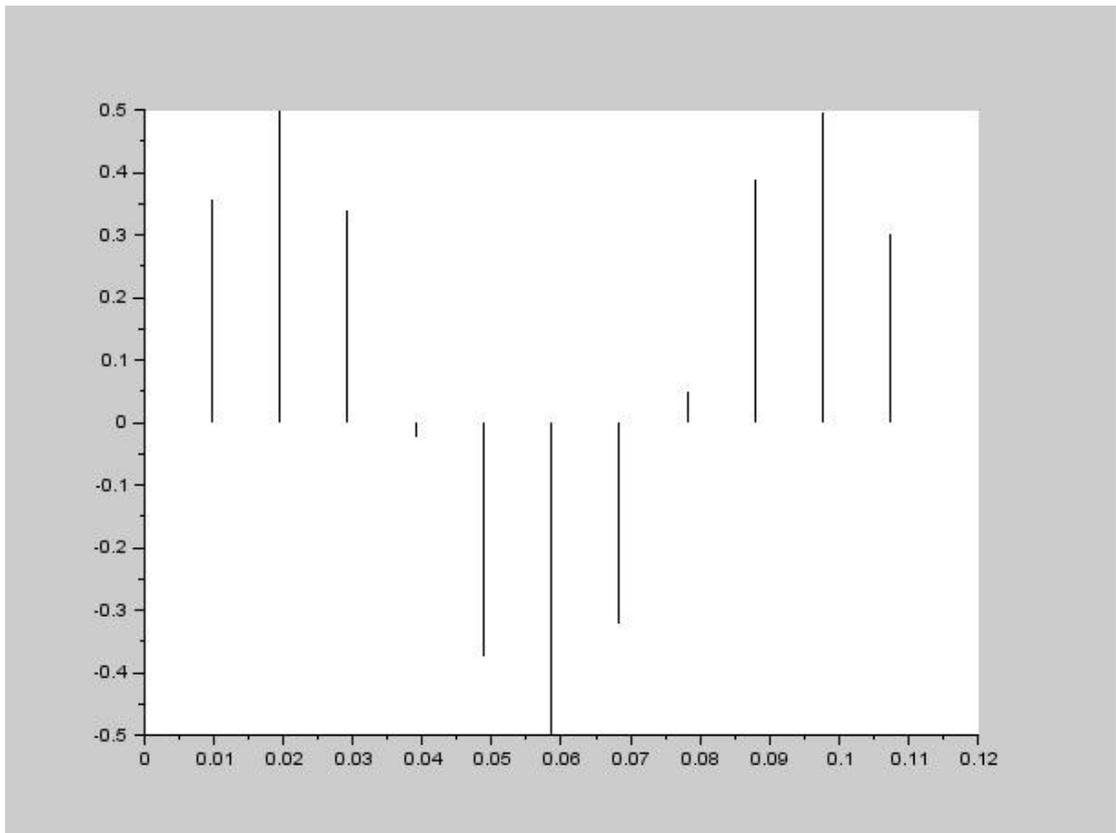
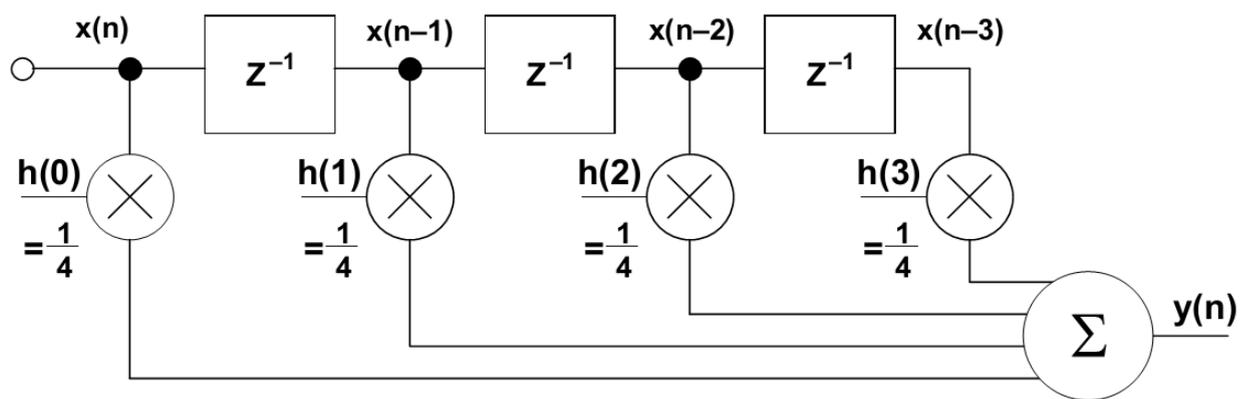


Рисунок 7. Форма децимированного сигнала

ЛАБОРАТОРНАЯ РАБОТА №4

Фильтр скользящего среднего

Элементарной формой КИХ-фильтра является фильтр скользящего среднего (moving average), показанный на рисунке 8. Фильтры скользящего среднего популярны для сглаживания данных, например, для анализа стоимости акций и т.д. Входные отсчеты $x(n)$ пропускаются через ряд регистров памяти (помеченных z^{-1} в соответствии с представлением элемента задержки при z -преобразовании). В приведенном примере имеется четыре каскада, соответствующих 4-точечному фильтру скользящего среднего. Каждый отсчет умножается на 0,25, и результаты умножения суммируются для получения значения скользящего среднего, которое подается на выход $y(n)$. На рисунке также представлено общее уравнение фильтра скользящего среднего на N точек. Вновь обращаем внимание, что N относится к числу точек при вычислении фильтра, а не к разрешающей способности АЦП или ЦАП, как в предыдущих разделах.



$$y(n) = h(0) x(n) + h(1) x(n - 1) + h(2) x(n - 2) + h(3) x(n - 3)$$

$$= \frac{1}{4} x(n) + \frac{1}{4} x(n - 1) + \frac{1}{4} x(n - 2) + \frac{1}{4} x(n - 3)$$

$$= \frac{1}{4} [x(n) + x(n - 1) + x(n - 2) + x(n - 3)]$$

Для N-точечного

фильтра скользящего среднего $y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n - k)$

Рисунок 8. 4-точечный фильтр скользящего среднего

С учетом равенства коэффициентов, наиболее простой путь исполнения фильтра скользящего среднего представлен на рисунке 9. Обратите внимание, что первым шагом является запоминание первых четырех отсчетов $x(0)$, $x(1)$, $x(2)$, $x(3)$ в регистрах. Эти величины суммируются и затем умножаются на 0,25 для получения первого выхода $y(3)$. Обратите внимание, что начальные значения выходов $y(0)$, $y(1)$ и $y(2)$ некорректны, потому что, пока отсчет $x(3)$ не получен, не все регистры заполнены.

Когда получен отсчет $x(4)$, он суммируется с результатом, а отсчет $x(0)$ вычитается из результата. Затем новый результат должен быть умножен на 0,25. Поэтому вычисления, требуемые для получения нового значения на выходе, состоят из одного суммирования, одного вычитания и одного умножения, независимо от длины фильтра скользящего среднего.

$$\begin{aligned}
 y(3) &= 0.25 \left[\begin{array}{c} x(3) + x(2) + x(1) + x(0) \end{array} \right] \\
 y(4) &= 0.25 \left[\begin{array}{c} x(4) + x(3) + x(2) + x(1) \end{array} \right] \\
 y(5) &= 0.25 \left[\begin{array}{c} x(5) + x(4) + x(3) + x(2) \end{array} \right] \\
 y(6) &= 0.25 \left[\begin{array}{c} x(6) + x(5) + x(4) + x(3) \end{array} \right] \\
 y(7) &= 0.25 \left[\begin{array}{c} x(7) + x(6) + x(5) + x(4) \end{array} \right]
 \end{aligned}$$

-
- **Вычисление каждого выходного значения требует**
- **1 умножения, 1 сложения и 1 вычитания**

Рисунок 9. Схема исполнения фильтра скользящего среднего

Реакция 4-точечного фильтра скользящего среднего на ступенчатое воздействие представлена на рисунке 10. Обратите внимание, что фильтр скользящего среднего не имеет выброса по фронту входного сигнала. Это делает его полезным в приложениях обработки сигналов, где требуется фильтрация случайного белого шума при сохранении характера входного импульса. Из всех возможных линейных фильтров фильтр скользящего среднего дает самый низкий уровень шума при заданной крутизне фронта импульса. Это показано на рис.6.9, где уровень шума понижается по мере увеличения числа точек. Существенно, что время реакции фильтра на ступенчатое воздействие от 0 % до 100 % равно произведению общего количества точек фильтра на период дискретизации.

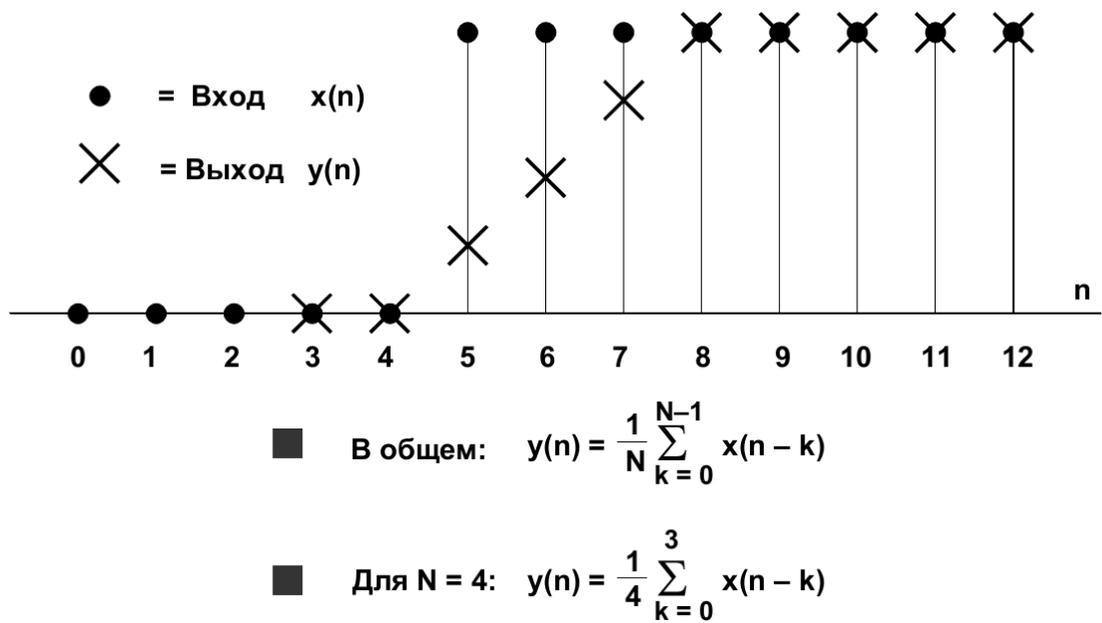


Рисунок 10. Реакция 4-точечного фильтра скользящего среднего на ступенчатое воздействие

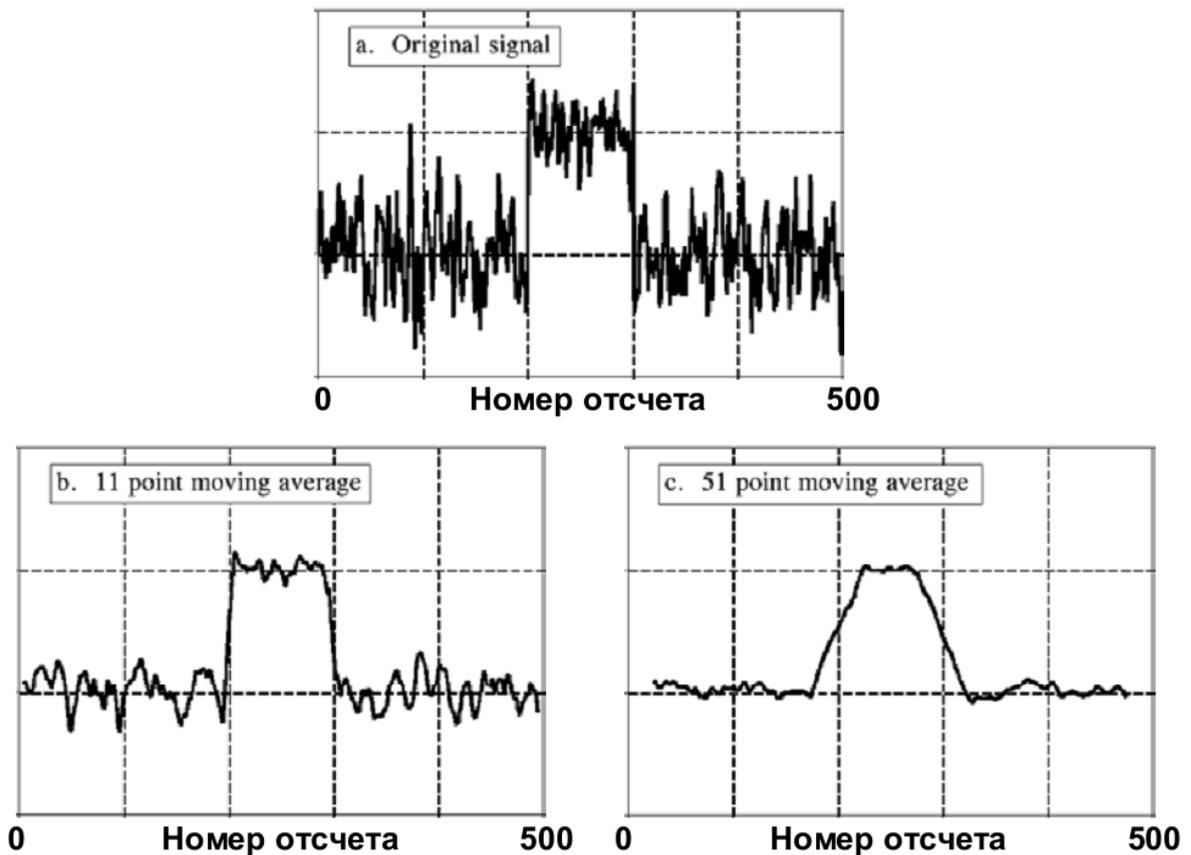


Рисунок 11. Реакция фильтра скользящего среднего на воздействие в виде смеси шума и ступенчатого сигнала

Частотная характеристика простого фильтра скользящего среднего выражается функцией $\sin(x)/x$. Она представлена в линейном масштабе на рисунке 12. Увеличение числа точек при реализации фильтра сужает основной лепесток, но существенно не уменьшает амплитуду боковых лепестков частотной характеристики, которая равна приблизительно -14 дБ для фильтра с 11 и с 31 отводами (длиной буфера). Естественно, эти фильтры не подходят в том случае, где требуется большое ослабление в полосе задержания.

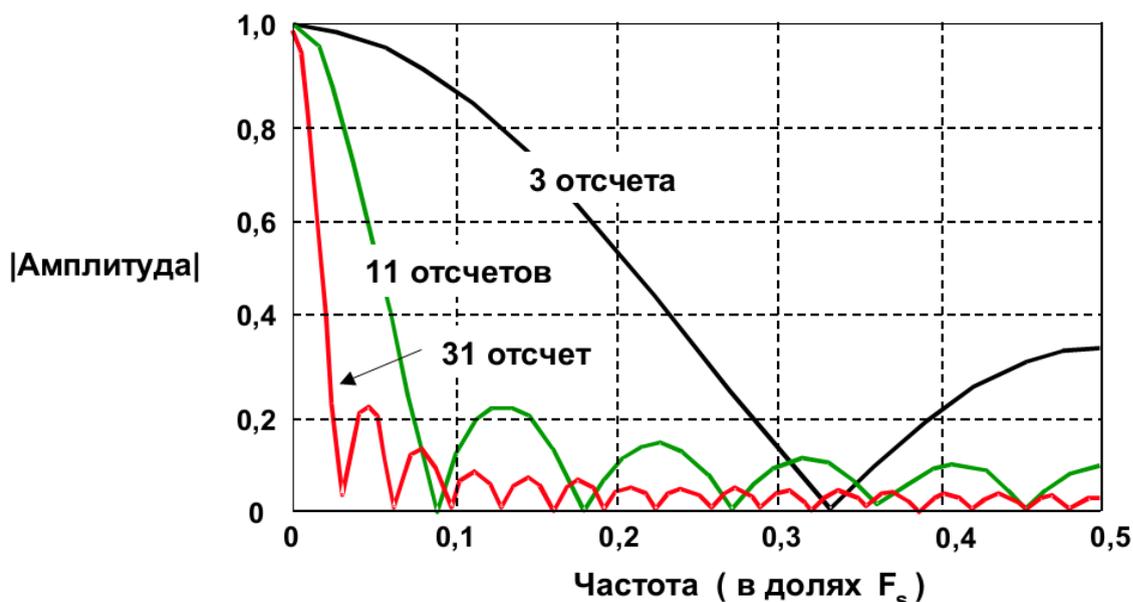


Рисунок 12. Частотная характеристика фильтра скользящего среднего

Можно существенно улучшить эффективность простого КИХ-фильтра скользящего среднего, выбирая разные веса или значения коэффициентов вместо равных значений. Крутизна спада может быть увеличена добавлением большего количества звеньев в фильтр, а характеристики полосы затухания улучшаются выбором надлежащих коэффициентов фильтра. Обратите внимание, что, в отличие от фильтра скользящего среднего, для реализации каждой ступени обобщенного КИХ-фильтра требуется цикл умножения с накоплением. Сущность проектирования КИХ-фильтра сводится к выбору соответствующих коэффициентов и необходимого числа звеньев при формировании желаемой частотной характеристики фильтра $H(f)$. Для включения необходимой частотной характеристики $H(f)$ в набор КИХ-коэффициентов имеются различные алгоритмы и программные пакеты. Большинство этого программного обеспечения разработано для персональных компьютеров и доступно на рынке. Ключевой теоремой проектирования КИХ-фильтра является

утверждение, что коэффициенты $h(n)$ КИХ-фильтра являются просто квантованными значениями импульсной характеристики этого фильтра. Соответственно, импульсная характеристика является дискретным преобразованием Фурье от $H(f)$.

Задание на работу:

Написать на языке Scilab программу цифровой фильтрации с помощью фильтра скользящего среднего. Для проверки фильтрации использовать сигнал с лабораторной работы №3. Подобрать количество коэффициентов фильтра N так, чтобы подавить гармонические сигналы всех присутствующих частот, кроме наименьшей. Изобразить спектр сигнала до фильтрации и после.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Белов В.М. Методические указания к лабораторным работам по системе SciLab / Белов В.М. - Томск : ТУСУР, 2012. – 8 с.
2. Алексеев Е. Р. Решение инженерных и математических задач / Е.Р. Алексеев, О.В.Чеснокова, Е. А.Рудченко. - М. : ALT Linux ; БИНОМ. Лаборатория знаний, 2008.- 260 с. : ил.; 8 с. цв. вклейки.- (Библиотека ALT Linux).
3. Кестер У. Аналого-цифровое преобразование. М.: Техносфера, 2007. С.1016.
4. Айфичер, Э. С. Цифровая обработка сигналов: практический подход : пер. с англ. 2-е изд. / Эммануил С. Айфичер, Барии У. Джервис. – М. : Вильямс, 2004. – 992 с.: ил.

ПРИЛОЖЕНИЕ

```

plot (y, <LineSpec>, <GlobalProperty>)
plot (x, y, <LineSpec>, <GlobalProperty>)
plot (x1, y1, <LineSpec1>, x2, y2, <LineSpec2>, ... xN, yN, <LineSpecN>, <GlobalProperty1>,
      <GlobalProperty2>, ... <GlobalPropertyM>)
plot (<axes_handle>, ...)
x a real matrice or vector
y a real matrice or vector

```

<LineSpec>

Color:

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

LineStyle:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dotted line

Marker type:

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or 's'	Square
'diamond' or 'd'	Diamond
^	Upward-pointing
>	Right-pointing
<	Left-pointing
'none'	No marker