

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт высокоточных систем им. В.П. Грязева
Кафедра «Системы автоматического управления»

Утверждено на заседании кафедры
«Системы автоматического управления»
«13» января 2021 г., протокол № 8

Заведующий кафедрой



О.В.Горячев

**СБОРНИК МЕТОДИЧЕСКИХ УКАЗАНИЙ
К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ
по дисциплине**

**«Цифровая фильтрация и цифровая обработка сигналов»
основной профессиональной образовательной программы
высшего образования – программы специалитета**

по специальности

24.05.06 Системы управления летательными аппаратами

со специализацией

Системы управления беспилотными летательными аппаратами

Форма обучения: очная

Идентификационный номер образовательной программы: 240506-01-21

Тула 2021 год

ЛИСТ СОГЛАСОВАНИЯ

Разработчик:

Ефромеев Андрей Геннадьевич, доц. каф. САУ, к.т.н.

(ФИО, должность, ученая степень, ученое звание)

(подпись)

Практическое занятие № 1

Генерация цифрового шума

I. ЦЕЛЬ И ЗАДАЧИ

Изучение способов генерации цифрового шума с заданными параметрами.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Случайные *шумы* являются очень важным типом сигналов как для сферы аналоговой электроники, так и для ЦОС. В частности, именно шум обычно определяет требования к минимальному уровню сигнала в системе, расстояние, на которое может быть осуществлена передача данных, или количество радиации, требуемое для получения рентгеновского снимка, и т. д. В цифровой обработке сигналов существует потребность в генерации шумов различных типов. Такая необходимость возникает при тестировании алгоритмов, которые должны оставаться работоспособными в присутствии шума заданного уровня.

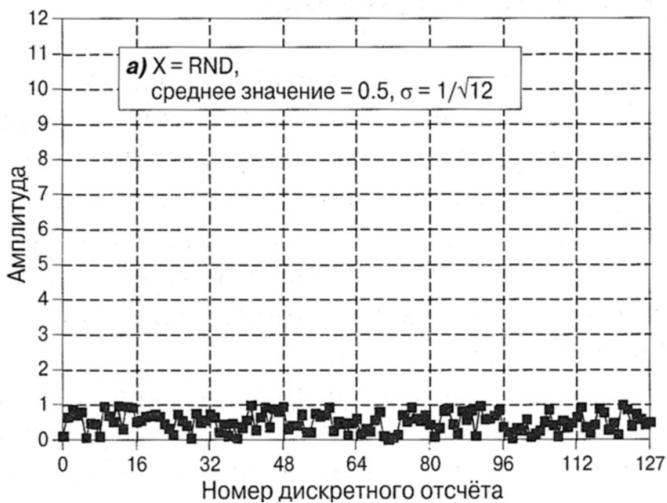
Генерация цифрового шума строится на базе *генератора случайных чисел*. В большинстве языков программирования для выполнения этой задачи предусмотрена специальная функция. Так, в языках программирования выражение вида $X = \text{random}$ производит присвоение переменной X случайного значения при каждом новом выполнении такой команды. Это случайное значение будет находиться в диапазоне от нуля до единицы и иметь равную вероятность появления в пределах этих границ. На рис. 1а показан сигнал, сформированный с помощью такого генератора случайных чисел для случая 128 отсчётов. Среднее значение исходного процесса при генерации данного сигнала составляет 0,5, а СКО равно $\frac{1}{\sqrt{12}} \approx 0,29$. Распределение является равномерным в диапазоне от нуля до единицы.

В процессе тестирования алгоритмов обычно используют сигналы того же типа, что будут встречаться в процессе реальной работы. Поэтому требуется генерировать цифровой шум не с равномерной, а с гауссовой функцией плотности вероятности. Существуют два метода формирования подобных сигналов с использованием описанного генератора случайных чисел. На Рис.1 проиллюстрирован первый метод. На (б) изображён сигнал, полученный с помощью сложения двух случайных чисел для формирования каждого отсчёта, т.е. $X = \text{random} + \dots + \text{random}$. Так как каждое случайное число может принимать значение от нуля до единицы, то их сумма уже может находиться в пределах от нуля до двух. Среднее значение в данном случае будет равно единице, а СКО составит $\frac{1}{\sqrt{6}}$ (запомните, что при сложении независимых случайных сигналов их *дисперсии*

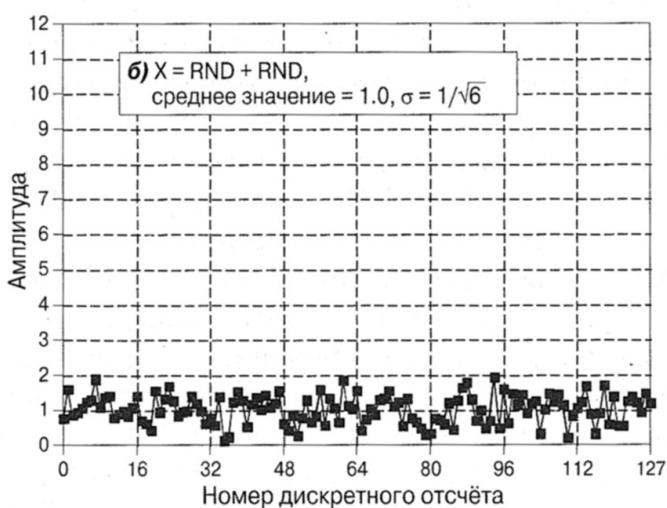
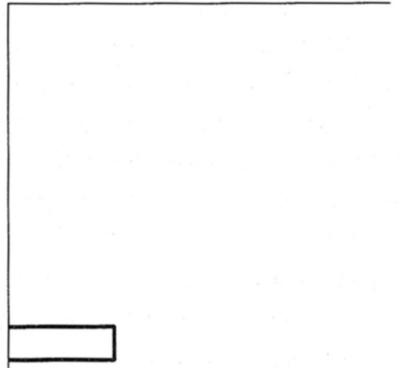
также складываются). При этом, как видно из графика функции плотности вероятности, равномерное распределение изменилось на треугольное. Объяснить это явление можно тем, что большинство значений сигнала теперь находится в области единицы, а меньшая часть отсчётов приближается к нулю и двум.

Далее перейдём к формированию каждого отсчёта сигнала как суммы 12 случайных чисел. В результате получим ещё большие изменения (в). Среднее значение стало равным *шести*, а СКО – *единице*. При этом самое главное то, что функция плотности вероятности стала практически гауссовой. Описанная процедура может использоваться для создания нормально распределённого шумового сигнала с произвольным требуемым средним значением и СКО. Для этого, рассчитывая каждый отсчёт сигнала, необходимо: 1) сложить 12 случайных чисел; 2) из полученной суммы вычесть шесть, делая среднее значение равным нулю; 3) умножить значение отсчёта на величину желаемого СКО; 4) добавить требуемое среднее значение.

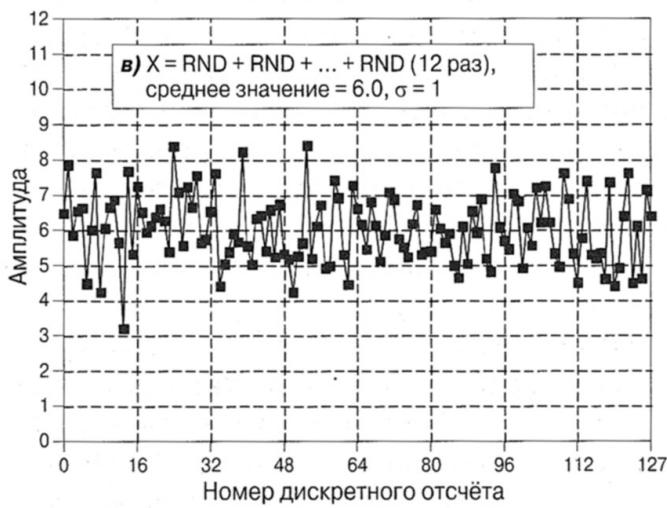
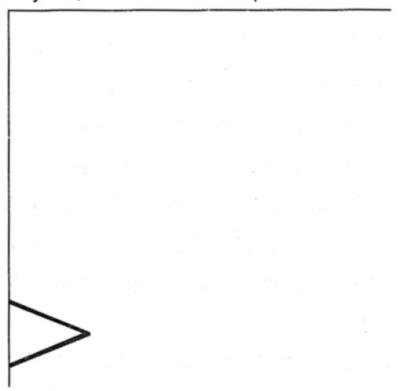
Математическое подтверждение справедливости описанной процедуры основывается на *центральной предельной теореме*, которая является одной из важнейших в теории вероятностей. В самом простом изложении теорема гласит, что *сумма* случайных чисел становится *нормально распределенной* с увеличением количества суммируемых элементов. При этом нет необходимости в том, чтобы отдельные случайные элементы были распределены по какому-то определённому закону или даже чтобы эти случайные числа имели один и тот же закон распределения. Центральная предельная теорема даёт объяснение факту широкого распространения нормально распределённых сигналов, в том числе в цифровой обработке сигналов. В реальном окружающем нас мире существует огромное множество различных сил, действующих на систему, в результате чего итоговая функция плотности вероятности такого совокупного влияния оказывается гауссовой.



Функция плотности вероятности



Функция плотности вероятности



Функция плотности вероятности

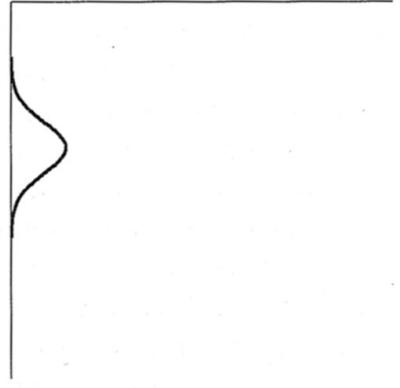


Рис. 1.

Далее перейдём к формированию каждого отсчёта сигнала как суммы 12 случайных чисел. В результате получим ещё большие изменения (в). Среднее значение стало равным **шести**, а СКО — **единице**. При этом самое главное то, что функция плотности вероятности стала практически гауссовой. Описанная процедура может использоваться для создания нормально распределённого

шумового сигнала с произвольным требуемым средним значением и СКО. Для этого, рассчитывая каждый отсчёт сигнала, необходимо: 1) сложить 12 случайных чисел; 2) из полученной суммы вычесть шесть, делая среднее значение равным нулю; 3) умножить значение отсчёта на величину желаемого СКО; 4) добавить требуемое среднее значение.

Математическое подтверждение справедливости описанной процедуры основывается на *центральной предельной теореме*, которая является одной из важнейших в теории вероятностей. В самом простом изложении теорема гласит, что *сумма* случайных чисел становится *нормально распределенной* с увеличением количества суммируемых элементов. При этом нет необходимости в том, чтобы отдельные случайные элементы были распределены по какому-то определённому закону или даже чтобы эти случайные числа имели один и тот же закон распределения. Центральная предельная теорема даёт объяснение факту широкого распространения нормально распределённых сигналов, в том числе в цифровой обработке сигналов. В реальном окружающем нас мире существует огромное множество различных сил, действующих на систему, в результате чего итоговая функция плотности вероятности такого совокупного влияния оказывается гауссовой.

Второй метод генерации нормально распределённого сигнала – метод Бокса-Мюллера – предлагает дважды запускать генератор случайных чисел, формируя два случайных числа R_1 и R_2 в интервале от 0 до 1. После этого может быть найдено нормально распределённое случайное число согласно выражениям

$$X_1 = \sqrt{-2 \ln R_1} \cos(2\pi R_2) \text{ или } X_2 = \sqrt{-2 \ln R_1} \sin(2\pi R_2)$$

В результате вычислений числа X_1 и X_2 окажутся нормально распределёнными и будут иметь среднее значение, равное нулю, а СКО, равное единице.

Как и ранее, с помощью данного подхода можно генерировать нормально распределённые случайные сигналы с произвольным средним значением и произвольным СКО. Для этого возьмите каждое число, полученное с помощью данного выражения, умножьте на желаемое значение СКО и добавьте требуемое среднее значение.

Работа генератора случайных чисел начинается с некоторого *начального значения*, лежащего в диапазоне от нуля до единицы. Это число по определённому фиксированному алгоритму преобразуется в новое число, лежащее в том же диапазоне и рассматриваемое как случайное. Это новое *число* сохраняется и при следующем вызове генератора случайных чисел используется в качестве начального значения. Алгоритм формирования случайного числа на основе начального значения выглядит следующим образом:

$$R = (aS + b) \bmod c$$

где S – начальное значение; R – генерируемое случайное число; a , b и c – выбранные некоторым образом константы.

Таким образом, на основе одного начального значения формируется последовательность случайных чисел. Нетрудно видеть, что если при генерации новой последовательности случайных чисел начальное значение оставить прежним, то она будет точно совпадать с первой сформированной последовательностью. Для того чтобы заставить генератор формировать неповторяющиеся случайные последовательности, большинство языков программирования включают функции переустановки начального значения генератора случайных чисел так, чтобы оно каждый раз оказывалось новым. Общепринятой практикой является использование системных часов в качестве начального значения показаний. Это при каждом новом запуске программы гарантирует выдачу новой случайной последовательности.

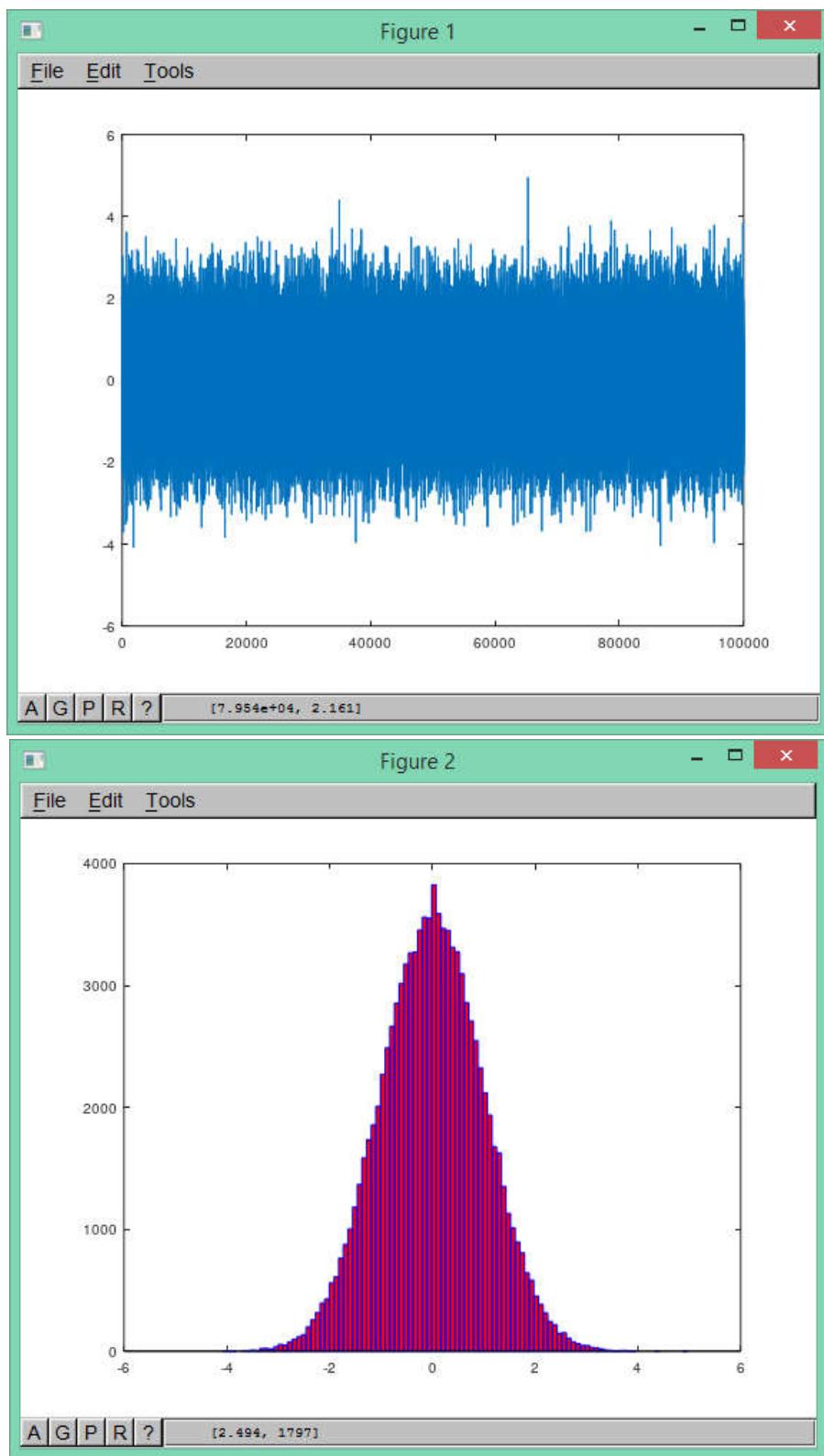
С математической точки зрения, числа, сгенерированные описанным образом, не являются абсолютно случайными, так как каждое полученное значение полностью определяется *предыдущим* числом. Поэтому для таких процессов принято использовать термин *псевдослучайные*. Однако эта особенность выходит за рамки рассматриваемой нами темы. С достаточно высокой степенью достоверности можно утверждать, что последовательности, формируемые с помощью генераторов случайных чисел, являются случайными. Вероятность того, что вам встретится ситуация, в которой такие последовательности нельзя будет считать случайными, крайне мала.

III. ПРАКТИЧЕСКАЯ ЧАСТЬ

При исследовании случайных сигналов в системах математических расчетов Matlab или Octave удобно пользоваться функциями формирования случайных последовательностей с нормальным распределением `randn` и функцией построения гистограммы сигнала `hist`:

```
y = randn (1, 100000);
h = hist (y, 1000);
figure(1);
plot(y)
figure(2);
hist (y, 100, "facecolor", "r", "edgecolor", "b")
```

В результате работы представленного скрипта получим случайный сигнал из 100000 отсчетов и его гистограмму:



IV. ЗАДАНИЕ

Изучить методы генерации случайных сигналов с заданными свойствами.

V. ПОРЯДОК ВЫПОЛНЕНИЯ

1. Реализуйте генерацию случайного сигнала различными способами.

2. Протестируйте работу вашей функции в различных условиях и с учетом сложения добейтесь нормального распределения.
3. Сравните работу вашего алгоритма со стандартной функцией генерации случайных последовательностей.
4. Сделайте выводы.

VI. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- алгоритмы и листинги разработанных программ;
- графики и иные результаты численного моделирования;
- выводы по проведённой работе.

VII. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит; пер. с англ. А.Ю. Линовича, С.В. Витязева, И.С. Гусинского. – М.: Додэка-XXI, 2012. – 720 с.
2. Умняшкин С.В. Истории цифровой обработки сигналов: Учебное пособие // М.: ТЕХНОСФЕРА, 2016. – 528 с.
3. Лайонс Р. Цифровая обработка сигналов. Пер. С англ. – М.: ООО «Бином-Пресс», 2015. – 656 с.
4. Основы цифровой обработки сигналов: Курс лекций / Авторы: А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б. Соловьёва, И.И. Гук. – Спб.: БХВ-Петербург, 2003. – 608 с.
5. Гадзиковский В.И. Цифровая обработка сигналов. – М.: СОЛОН-ПРЕСС, 2015. – 766 с.

Практическое занятие № 2

Спектральный анализ и дискретное преобразование Фурье.

I. ЦЕЛЬ И ЗАДАЧИ

Изучение спектрального анализа дискретных сигналов. Расчёт спектра дискретного сигнала в Matlab с применением дискретного преобразования Фурье.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Спектр дискретного сигнала

Разработка преобразований Фурье сыграла огромную роль в появлении и развитии ряда новых областей науки и техники. Достаточно отметить, что электрическая связь и радиосвязь базируются на спектральном представлении сигналов. Ряды Фурье также можно рассматривать как приближение произвольных функций (определенные ограничения в этом известны) тригонометрическими рядами бесконечной длины. При конечной длине рядов получаются наилучшие средне-квадратические приближения.

MATLAB содержит функции для выполнения быстрого дискретного преобразования Фурье.

Преобразование Фурье позволяет вычислить спектральную плотность сигнала, представляющего собой *функцию* (как правило, времени либо пространственных координат). Дискретный же сигнал является *последовательностью чисел*, поэтому для анализа его спектра обычными (аналоговыми) средствами необходимо сопоставить этой последовательности некоторую функцию.

Традиционным способом такого сопоставления является представление отсчетов в виде дельта-функций с соответствующими множителями и задержками. Для последовательности отсчетов $\{x(k)\}$ получится следующий сигнал:

$$s(t) = \sum_{k=-\infty}^{\infty} x(k)\delta(t-k). \quad (1)$$

Преобразование Фурье линейно, спектр дельта-функции равен единице, а задержка сигнала во времени приводит к умножению спектра на комплексную экспоненту. Все это позволяет сразу же записать спектр дискретного сигнала:

$$\dot{S}(\omega) = \sum_{k=-\infty}^{\infty} x(k)e^{-j\omega k}. \quad (2)$$

Из этой формулы видно главное свойство спектра любого дискретного сигнала: спектр является периодическим, и его период в данном случае равен 2π .

(то есть круговой частоте дискретизации, поскольку, составляя сигнал из дельта-функций, мы выбрали единичный интервал между ними, что дает $\omega_D = 2\pi$).

Размерность спектральной функции дискретного сигнала совпадает с размерностью отсчетов. Это связано с тем, что дельта-функции времени, из которых был составлен сигнал (1), имеют размерность частоты.

Формула (2) позволяет вычислить спектральную функцию по известным отсчетам $x(k)$. При конечном числе ненулевых отсчетов этот расчет несложен; он может быть выполнен с помощью функции MATLAB freqz.

Теперь рассмотрим несколько иную задачу. Пусть значения $x(k)$ являются отсчетами аналогового сигнала $s(t)$ взятыми с периодом T :

$$x(k) = s(kT).$$

Выясним, как в этом случае спектр дискретного сигнала (2) связан со спектром аналогового сигнала $s(t)$.

Итак, мы рассматриваем *дискретизированный* сигнал в виде последовательности дельта-функций, «взвешенной» значениями отсчетов $s(kT)$ аналогового сигнала $s(t)$ (рис. 1):

$$s_{\tilde{A}}(t) = \sum_{k=-\infty}^{\infty} s(kT) \delta(t - kT). \quad (3)$$

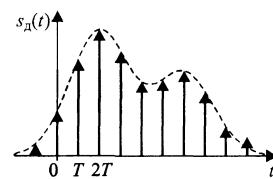


Рис. 1. Дискретизированный сигнал в виде последовательности дельта-функций

Так как функция $\delta(t - kT)$ равна нулю всюду, кроме момента $t = kT$, можно заменить в выражении (3) константы $s(kT)$ на исходный непрерывный сигнал $s(t)$:

$$s_{\tilde{A}}(t) = s(t) \sum_{k=-\infty}^{\infty} \delta(t - kT). \quad (4)$$

С учётом того, что в интервал интегрирования $(-T/2, T/2)$ попадает только одна дельта-функция, соответствующая $k = 0$, периодическая последовательность дельта-функций может быть представлена в виде комплексного ряда Фурье:

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j\omega_n t}, \quad (5)$$

где $\omega_n = 2\pi n/T$. Подставив (5) в (4), получим $s_{\tilde{A}}(t) = \frac{s(t)}{T} \sum_{n=-\infty}^{\infty} e^{j\omega_n t} = \frac{1}{T} \sum_{n=-\infty}^{\infty} s(t) e^{j\omega_n t}$.

Умножение сигнала на $\exp(j\omega_n t)$ соответствует сдвигу спектральной функции на ω_n , поэтому спектр дискретизированного сигнала можно записать следующим образом:

$$\dot{S}_{\hat{A}}(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \dot{S}\left(\omega - \frac{2\pi n}{T}\right). \quad (6)$$

Таким образом, спектр дискретизированного сигнала представляет собой бесконечный ряд сдвинутых копий спектра исходного непрерывного сигнала $s(t)$ (рис. 2). Расстояние по частоте между соседними копиями спектра равно частоте дискретизации $\omega_d = 2\pi/T$.

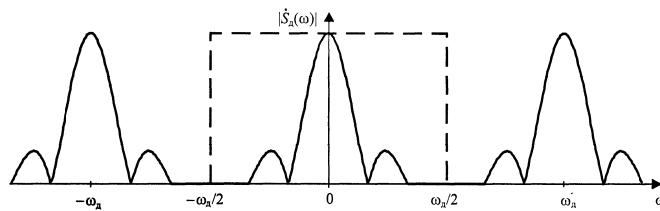


Рис. 2. Спектр дискретизированного сигнала

Характер спектра дискретизированного сигнала еще раз демонстрирует частотно-временную дуальность преобразования Фурье:

периодический сигнал \rightarrow дискретный спектр;

периодический спектр \rightarrow дискретный сигнал.

Рисунок 2 наглядно демонстрирует способ восстановления непрерывного сигнала по дискретным отсчетам. Для этого необходимо пропустить дискретный сигнал через идеальный фильтр нижних частот (ФНЧ) с частотой среза, равной половине частоты дискретизации. АЧХ такого фильтра показана на рис. 2 пунктиром.

Очевидно, что точное восстановление сигнала возможно, если сдвинутые копии спектра не перекрываются. Из рис. 2 видно, что для этого необходимо, чтобы частота дискретизации как минимум в два раза превышала верхнюю граничную частоту в спектре сигнала:

$$\omega_{\hat{A}} > 2\omega_A \quad (7)$$

В случае произвольного сигнала, если условие (7) не выполняется, сдвинутые копии спектра будут накладываться друг на друга, что приведет к неизбежным искажениям при восстановлении непрерывного сигнала (рис. 3).

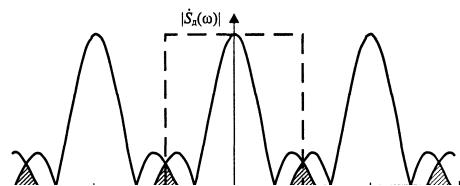


Рис. 3. Перекрытие сдвинутых копий спектра при недостаточно высокой частоте дискретизации

Эти искажения вызваны тем, что спектральные составляющие сигнала с частотами, превышающими частоту Найквиста, равную $\omega_d / 2$, не могут быть восстановлены правильно - вместо этого они вызывают наложение «хвостов» соседних сдвинутых копий спектра и появление ложных частот.

Если подлежащий дискретизации сигнал может содержать спектральные составляющие с частотами, превышающими частоту Найквиста, полезно предварительно пропустить его через ФНЧ с частотой среза, равной частоте Найквиста (рис. 4). При этом все равно будут потеряны высокочастотные составляющие — сохранить их можно лишь путем повышения частоты дискретизации. Однако в этом случае благодаря отсутствию наложения «хвостов» не произойдет появления ложных частот и диапазон частот $0 \dots \omega_d / 2$ будет представлен в дискретном сигнале без искажений.

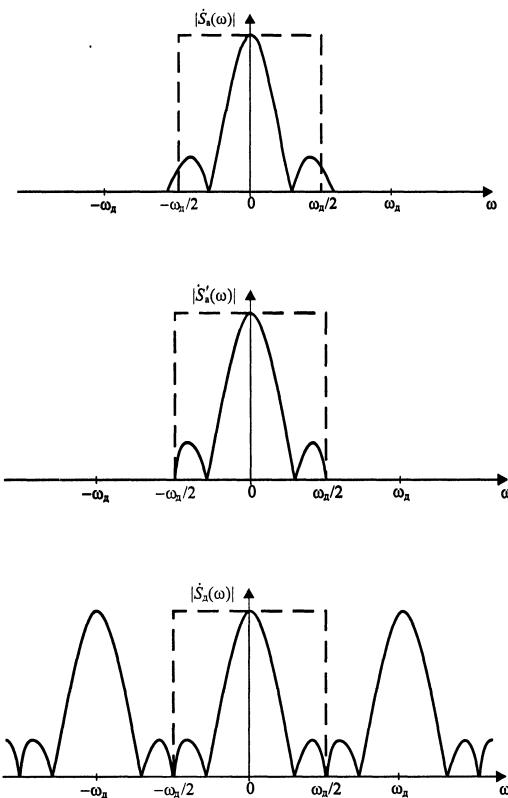


Рис. 4. При дискретизации сигнала, содержащего высокочастотные составляющие (а), полезно пропустить его через ФНЧ (б), чтобы избежать появления ложных частот (в)

Теорема Котельникова

Выше было сформулировано утверждение о том, что некоторые сигналы могут быть без потерь информации представлены своими дискретными отсчетами, а полученное выражение для спектра дискретизированного сигнала позволяет выделить тот класс сигналов, для которых это возможно, и описать способ такого восстановления.

Согласно (6), спектр дискретизированного сигнала представляет собой сумму сдвинутых копий спектра исходного сигнала, при этом шаг сдвига равен частоте дискретизации ω_d . Из рис. 1 видно, что если в спектре аналогового сигнала не содержится составляющих с частотами, превышающими частоту Найквиста ($\omega_d/2$), то сдвинутые копии спектра не будут перекрываться. В этом случае использование идеального ФНЧ с прямоугольной АЧХ позволит выделить исходную (несдвинутую) копию спектра, сосредоточенную в окрестностях нулевой частоты, и, таким образом, в точности восстановить исходный аналоговый сигнал.

АЧХ идеального ФНЧ, восстанавливающего аналоговый сигнал, приведена на рис. 5 слева. Коэффициент передачи в полосе пропускания равен T , а не единице, чтобы компенсировать множитель $1/T$ в формуле (6). С помощью обратного преобразования Фурье найдем импульсную характеристику фильтра (рис. 5 справа):

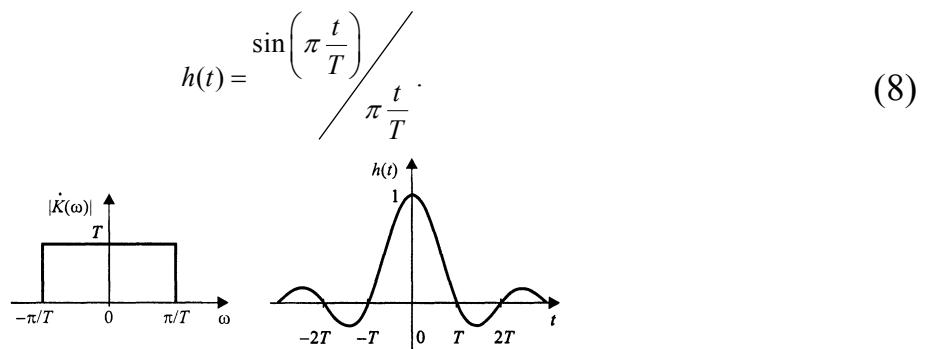


Рис. 5. Амплитудно-частотная (слева) и импульсная (справа) характеристики идеального восстанавливающего фильтра

Дискретизированный сигнал (3) представляет собой сумму дельта-функций. При прохождении такого сигнала через восстанавливающий ФНЧ каждая дельта-функция породит на выходе соответствующим образом сдвинутую и масштабированную копию импульсной характеристики фильтра. Выходной сигнал (в точности соответствующий исходному аналоговому сигналу), таким образом, будет представлять собой сумму сдвинутых и умноженных на отсчеты сигнала копий импульсных характеристик идеального ФНЧ (8):

$$s(t) = \sum_{k=-\infty}^{\infty} s(kT) \frac{\sin\left(\pi \frac{t-kT}{T}\right)}{\pi \frac{t-kT}{T}}. \quad (9)$$

Подводя итог всему сказанному, сформулируем *теорему Котельникова*: любой сигнал $s(t)$, спектр которого не содержит составляющих с частотами выше некоторого значения $\omega_A = 2\pi f_A$, может быть без потерь информации представлен своими дискретными отсчетами $\{s(kT)\}$, взятыми с интервалом T , удовлетворяющим следующему неравенству:

$$T \leq \frac{1}{2f_A} = \frac{\pi}{\omega_A}. \quad (10)$$

Восстановление исходного непрерывного сигнала $s(t)$ по набору его дискретных отсчетов $\{s(kT)\}$ производится по формуле (9).

Формирование непрерывного сигнала по его дискретным отсчетам поясняет рис. 6. Пунктиром показаны графики отдельных слагаемых формулы (9), сплошной линией - восстановленный сигнал. Ниже приводится код MATLAB, использованный при построении рисунка:

```
t = -2:0.01:6; % время для восстановленного сигнала
td = -2:6; % номера отсчетов
s= [0 0 4 3 2 1 0 0 0]; % дискретный сигнал
d = [td' s'];
y = pulstran(t, d, 'sinc'); % восстановленный сигнал
plot(td, s, 'o', t, y) % график восстановленного
% сигнала
hold on % выводим графики отдельных sinc-импульсов
for k=1:length(s), plot(t, s(k)*sinc(t-td(k)), ':'), end
hold off
```

Рисунок 6 наглядно демонстрирует главное свойство сигнала с ограниченным спектром - его бесконечность во времени. Хотя отличны от нуля лишь несколько отсчетов показанного сигнала, аналоговый сигнал оказывается бесконечно колеблющимся - между нулевыми отсчетами (на рисунке это отсчеты с номерами -2, -1, 4, 5, 6) его значения отличны от нуля. Эти колебания нигде не заканчиваются, хотя их амплитуда стремится к нулю.

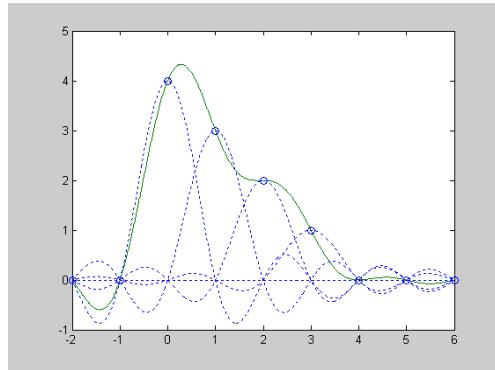


Рис. 6. Восстановление непрерывного сигнала по его дискретным отсчетам

Иногда можно встретить примерно следующее «объяснение» сущности теоремы Котельникова: «если брать отсчеты достаточно часто, в промежутках между ними сигнал с ограниченными спектром не успеет сильно измениться, и мы сможем точно восстановить его». Такая трактовка является принципиально неправильной. Когда мы говорим об ограниченной полосе частот сигнала, имеется в виду спектральная функция *всего сигнала, имеющего бесконечную длительность*. При этом *мгновенные спектры отдельных фрагментов* сигнала могут содержать сколь угодно высокие частоты. В частности, в отдельном промежутке между соседними отсчетами сигнал с ограниченным спектром может иметь сколь угодно сложную форму, например произвольное число раз менять знак.

Дискретное преобразование Фурье

Прямое преобразование Фурье переводит описание сигнала (функции времени) из временной области в частотную, а обратное преобразование Фурье — из частотной области во временную. На этом основаны многочисленные методы фильтрации сигналов.

В разделе «Спектр дискретного сигнала» мы проанализировали явления, происходящие со спектром при дискретизации сигнала. Рассмотрим теперь, что представляет собой спектр дискретного *периодического* сигнала.

Итак, пусть последовательность отсчетов $\{x(k)\}$ является периодической с периодом N :

$$x(k + N) = x(k) \text{ для любого } k.$$

Такая последовательность полностью описывается *конечным* набором чисел, в качестве которого можно взять произвольный фрагмент длиной N , например $\{x(k), k = 0, 1, \dots, N - 1\}$. Поставленный в соответствие этой последовательности сигнал из смешенных по времени дельта-функций:

$$s(t) = \sum_{k=-\infty}^{\infty} x(k) \delta(t - kT) \quad (11)$$

также, разумеется, будет периодическим с минимальным периодом $N T$.

Так как сигнал (11) является дискретным, его спектр должен быть *периодическим* с периодом $2\pi/T$. Так как этот сигнал является также и периодическим, его спектр должен быть *дискретным* с расстоянием между гармониками, равным $2\pi/(NT)$.

Итак, периодический дискретный сигнал имеет периодический дискретный спектр, который также описывается конечным набором из N чисел (один период спектра содержит $\frac{2\pi}{T}/\frac{2\pi}{NT} = N$ гармоник).

Таким образом, формула для вычисления комплексных амплитуд гармоник представляет собой линейную комбинацию отсчетов сигнала.

В выражении (12) реальный масштаб времени фигурирует только в множителе $1/T$ перед оператором суммирования. При рассмотрении дискретных последовательностей обычно оперируют номерами отсчетов и спектральных гармоник без привязки к действительному масштабу времени и частоты. Поэтому множитель $1/T$ из (12) удаляют, то есть считают частоту дискретизации равной единице. Удаляют обычно и множитель $1/N$. Получившееся выражение называется *дискретным преобразованием Фурье* (ДПФ; английский термин — Discrete Fourier Transform, DFT):

$$\dot{X}(n) = \sum_{k=0}^{N-1} x(k) \exp\left(-j \frac{2\pi n k}{N}\right). \quad (13)$$

Существует и *обратное* дискретное преобразование Фурье. Переход от дискретного спектра к временным отсчетам сигнала выражается следующей формулой:

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} \dot{X}(n) \exp\left(j \frac{2\pi n k}{N}\right). \quad (14)$$

Матрица ДПФ

ДПФ является линейным преобразованием, трансформирующим вектор временных отсчетов в вектор такой же длины, содержащий отсчеты спектральные. Такие преобразование может быть реализовано как умножение некоторой квадратной матрицы на входной вектор-столбец:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

где \mathbf{A} — *матрица преобразования*. В случае ДПФ эта матрица имеет вид

$$A_{DFT} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & e^{-j\frac{6\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}(N-1)} \\ 1 & e^{-j\frac{4\pi}{N}} & e^{-j\frac{8\pi}{N}} & e^{-j\frac{12\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}2(N-1)} \\ 1 & e^{-j\frac{6\pi}{N}} & e^{-j\frac{12\pi}{N}} & e^{-j\frac{18\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\frac{2\pi}{N}(N-1)} & e^{-j\frac{2\pi}{N}2(N-1)} & e^{-j\frac{2\pi}{N}3(N-1)} & \dots & e^{-j\frac{2\pi}{N}(N-1)^2} \end{bmatrix}.$$

Общая формула для элемента матрицы, расположенного в n -м столбце m -й строки, выглядит так:

$$A_{DFT}(m, n) = \exp\left(-j2\pi \frac{(m-1)(n-1)}{N}\right), 1 \leq m \leq N, 1 \leq n \leq N.$$

Вычисление ДПФ путем умножения матрицы на вектор полностью соответствует формуле (13). Этот метод требует большого количества вычислительных операций, поэтому на практике вместо него применяются быстрые алгоритмы, рассматриваемые далее.

Связь ДПФ и спектра дискретного сигнала

Имея один и тот же конечный набор чисел, можно рассчитать либо *спектральную функцию* этого дискретного сигнала по формуле (2), либо его ДПФ по формуле (13). Разумеется, возникает вопрос о том, как связаны друг с другом эти два спектральных представления, полученные на основе одних и тех же отсчетов сигнала.

Сравнение формул (2) и (13) показывает, что ДПФ представляет собой просто *дискретные отсчеты* спектральной функции дискретного сигнала, соответствующие частотам $\omega_n = \frac{\omega_A n}{N}$:

$$\dot{X}(n) = \dot{S}\left(\frac{2\pi n}{NT}\right) = \dot{S}\left(\omega_A \frac{n}{N}\right). \quad (16)$$

По этой причине значения ДПФ иногда называют *спектральными отсчетами*.

Из соотношения (16) следует еще один важный вывод: если добавить к конечному набору отсчетов некоторое количество нулей, спектральная функция дискретного сигнала, естественно, не изменится, но ДПФ даст большее число спек-

тральных отсчетов, соответствующих частотам, более тесно расположенным в интервале от нуля до частоты дискретизации.

Поясним сказанное на простом примере, вычислив ДПФ для отсчетов прямоугольного импульса при разном количестве концевых нулей (рис. 7):

```

x1 = [ones(8,1); zeros(8,1)];           % 16 отсчетов
y1 = fft(x1);                         % ДПФ сигнала x1
x2 = [x1; zeros(16,1)];                % добавляем 16 нулей
stem((0:31)/32, abs(y2))              % модуль ДПФ сигнала x2
subplot(2, 2, 1), stem(0:15, x1)       % график сигнала x1
xlim([0 31])
subplot(2, 2, 2)
stem((0:15)/16, abs(y1))              % модуль ДПФ сигнала x1
subplot(2, 2, 3), stem(0:31, x2)       % график сигнала x2
xlim([0 31])
subplot(2, 2, 4)
y2 = fft(x2);                         % ДПФ сигнала x2

```

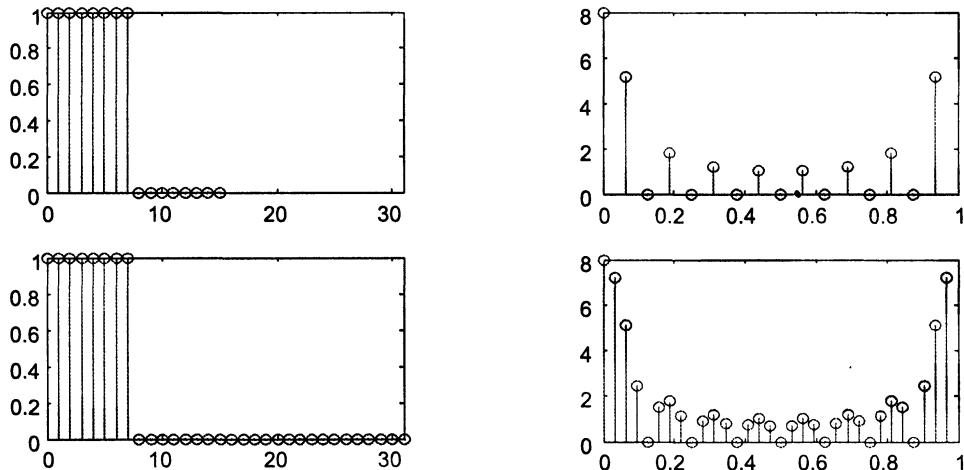


Рис. 7. Повышение спектрального разрешения ДПФ при дополнении сигнала нулями: сверху - исходный сигнал и модуль его ДПФ, снизу - сигнал, дополненный 16 нулями, и модуль его ДПФ

Горизонтальные оси на спектральных графиках проградуированы в долях частоты дискретизации. Из рис. 5.1 видно, что после увеличения длины сигнала вдвое за счет добавления нулевых отсчетов результат ДПФ стал содержать вдвое больше значений, соответствующих в два раза чаще расположенным частотам. Таким образом, дополнение сигнала нулями позволяет повысить спектральное разрешение при вычислении ДПФ.

Функции спектрального анализа в MATLAB

Функция одномерного прямого преобразования Фурье

В описанных ниже функциях реализован особый метод *быстрого преобразования Фурье* — Fast Fourier Transform (FFT, или БПФ), позволяющий

резко уменьшить число арифметических операций в ходе приведенных выше преобразований.

Для одномерного преобразования используется следующая функция:

- ✓ `fft(X)` возвращает для вектора X дискретное преобразование Фурье, по возможности используя алгоритм быстрого преобразования Фурье. Если X — матрица, функция `fft` возвращает преобразование Фурье для каждого столбца матрицы;
- ✓ `fft(X,n)` возвращает n-точечное преобразование Фурье. Если длина вектора X меньше n, то недостающие элементы заполняются нулями. Если длина X больше n, то лишние элементы удаляются. Когда X — матрица, длина столбцов корректируется аналогично;
- ✓ `fft(X,[],dim)` и `fft(X,n,dim)` применяют преобразование Фурье к одной из размерностей массива в зависимости от значения параметра dim

Функции обратного преобразования Фурье

В одномерном случае возможно обратное преобразование Фурье, реализуемое следующей функцией:

- ✓ `ifft(F)` возвращает результат дискретного обратного преобразования Фурье вектора F. Если F - матрица, то `ifft` возвращает обратное преобразование Фурье для каждого столбца этой матрицы;
- ✓ `ifft(F,n)` возвращает результат n-точечного дискретного обратного преобразования Фурье вектора F;
- ✓ `ifft(F,[],dim)` и `y = ifft(X,n,dim)` возвращают результат обратного дискретного преобразования Фурье массива F по строкам или по столбцам в зависимости от значения скаляра dim.

Для любого X результат последовательного выполнения прямого и обратного преобразований Фурье `ifft(fft(x))` равен X с точностью до погрешности округления. Если X — вещественный массив, `ifft(fft(x))` может иметь малые мнимые части.

III. ПРАКТИЧЕСКАЯ ЧАСТЬ

Для иллюстрации применения преобразования Фурье создадим трехчастотный сигнал на фоне сильного шума, создаваемого генератором случайных чисел:

```
t=0:0.0005:1;  
x=sin(2*pi*200*t)+0.6*sin(2*pi*150*t)+0.8*sin(2*pi*250*t);  
y=x+2*randn(size(t));  
plot(y(1:100))
```

Этот сигнал имеет среднюю частоту 200 Гц и два боковых сигнала с частотами 150 и 250 Гц, что соответствует амплитудно-модулированному сигналу с частотой модуляции 50 Гц и глубиной модуляции 0,8 (амплитуда боковых частот составляет 0,4 от амплитуды центрального сигнала). На рис. 8 показан график этого сигнала (по первым 100 отсчетам из 2000). Нетрудно заметить, что из него никаким образом не видно, что полезный сигнал — амплитудно-модулированное колебание, настолько оно забито шумами.

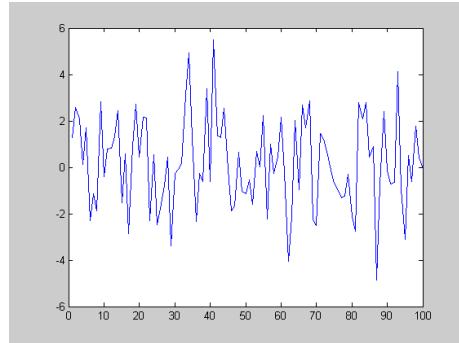


Рис.8 Форма зашумленного сигнала

Теперь построим график спектральной плотности полученного сигнала с помощью прямого преобразования Фурье, по существу переводящего временное представление сигнала в частотное. Этот график (см. рис. 8) в области частот до 300 Гц строится следующими командами:

```
Y=fft(y,1024);
f=2000*(0:150)/1024;
figure;
plot(f,abs(Y(1:151))); grid
```

График спектральной плотности сигнала, построенный в этом примере, представлен на рис. 9.

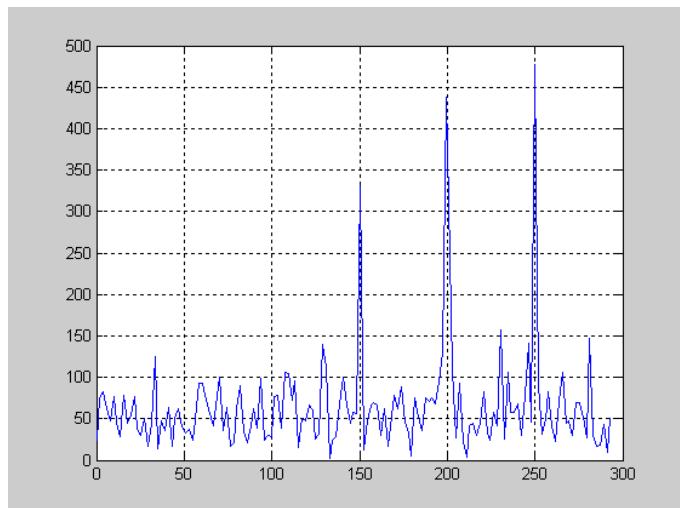


Рис 9. График спектральной плотности приведенного на рис. 8 сигнала

Даже беглого взгляда на рис. 9 достаточно, чтобы убедиться в том, что спектограмма сигнала имеет явный пик на средней частоте амплитудно-модулированного сигнала и два боковых пика. Все эти три частотные составляющие сигнала явно выделяются на общем шумовом фоне. Таким образом, данный пример наглядно иллюстрирует технику обнаружения слабых сигналов на фоне шумов лежащую в основе работы радиоприемных устройств.

IV. ЗАДАНИЕ

Изучить основные положения теории спектрального анализа дискретных сигналов.

Изучить функции MATLAB дискретного преобразования Фурье, необходимые для спектрального анализа дискретных сигналов.

Рассчитать спектр сигнала.

V. ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить основные положения теории спектрального анализа дискретных сигналов.

2. Изучить основные положения теории дискретного преобразования Фурье.

Изучить основные функции MATLAB, необходимые для спектрального анализа дискретных сигналов.

3. В соответствии с планом предложенным в параграфе IV рассчитать спектр сигнала с заданными преподавателем параметрами.

VI. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- краткие теоретические сведения по выполняемой работе;
- алгоритмы и листинги программ расчёта спектра;
- выводы по проведённой работе.

VII. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Конспект лекций по курсу.
2. Дьяконов В. MATLAB: учебный курс. – СПб: Питер, 2001. – 560 с.: ил.
3. Цифровая обработка сигналов / А.Б. Сергиенко – СПб.: Питер, 2003. – 604 с.: ил.

Практическое занятие № 3

Спектральный анализ непрерывных и дискретных сигналов. Быстрое преобразование Фурье.

I. ЦЕЛЬ И ЗАДАЧИ

Изучение алгоритма быстрого преобразования Фурье дискретных сигналов. Вычисление 8-точечного ДПФ с помощью алгоритма БПФ с прореживанием по времени. Проверка полученных вычислений в Matlab.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Алгоритм быстрого преобразования Фурье

Для вычисления одного коэффициента ДПФ по формуле (13) (см. л.р. №2) необходимо выполнить N комплексных умножений и сложений. Таким образом, расчет всего ДПФ, содержащего N коэффициентов, потребует N^2 пар операций «умножение-сложение». Число операций возрастает пропорционально квадрату размерности ДПФ. Однако, если N не является простым числом и может быть разложено на множители, процесс вычислений можно ускорить, разделив анализируемый набор отсчетов на части, вычислив их ДПФ и объединив результаты. Такие способы вычисления ДПФ называются *быстрым преобразованием Фурье* (БПФ; английский термин — Fast Fourier Transform, FFT) и повсеместно используются на практике.

При реализации БПФ возможно несколько вариантов организации вычислений в зависимости от способа деления последовательности отсчетов на части (*прореживание по времени либо по частоте*) и от того, на сколько фрагментов производится разбиение последовательности на каждом шаге (*основание БПФ*).

БПФ с прореживанием по времени

Рассмотрим идею БПФ с *прореживанием по времени* (decimation in time, DIT) на примере деления набора отсчетов пополам.

Итак, пусть N - четное число. Выделим в формуле (13) два слагаемых, соответствующих элементам исходной последовательности с четными и нечетными номерами:

$$\dot{X}(n) = \sum_{m=0}^{N/2-1} x(2m)e^{-j\frac{2\pi 2mn}{N}} + \sum_{m=0}^{N/2-1} x(2m+1)e^{-j\frac{2\pi(2m+1)n}{N}}.$$

Введем обозначения $y(m) = x(2m)$ и $z(m) = x(2m+1)$, а также вынесем из второй суммы общий множитель $e^{-j\frac{2\pi mn}{N}}$:

$$\dot{X}(n) = \sum_{m=0}^{N/2-1} y(m)e^{-j\frac{2\pi mn}{N/2}} + e^{-j\frac{2\pi n}{N}} \sum_{m=0}^{N/2-1} z(m)e^{-j\frac{2\pi mn}{N/2}}. \quad (17)$$

Две суммы в (17) представляют собой ДПФ последовательностей $\{y(m)\}$ (отсчеты с четными номерами) и $\{z(m)\}$ (отсчеты с нечетными номерами). Каждое из этих ДПФ имеет размерность $N/2$. Таким образом,

$$\dot{X}(n) = \dot{Y}(n) + e^{-j\frac{2\pi n}{N}} \dot{Z}(n), \quad (18)$$

где $\dot{Y}(n)$ и $\dot{Z}(n)$ - ДПФ соответственно последовательностей отсчетов с четными и нечетными номерами:

$$\dot{Y}(n) = \sum_{m=0}^{N/2-1} y(m)e^{-j\frac{2\pi mn}{N/2}},$$

$$\dot{Z}(n) = \sum_{m=0}^{N/2-1} z(m)e^{-j\frac{2\pi mn}{N/2}}.$$

Так как ДПФ размерности $N/2$ дает лишь $N/2$ спектральных коэффициентов, непосредственно использовать формулу (18) можно только при $0 \leq n < N/2$. Для остальных n ($N/2 \leq n < N$) следует воспользоваться периодичностью спектра дискретного сигнала (и, соответственно, периодичностью результатов ДПФ):

$$\dot{Y}\left(n + \frac{N}{2}\right) = \dot{Y}(n), \dot{Z}\left(n + \frac{N}{2}\right) = \dot{Z}(n).$$

С учетом этого при $n \geq N/2$ формула (18) представляется в виде

$$\dot{X}(n) = \dot{Y}\left(n - \frac{N}{2}\right) + e^{-j\frac{2\pi n}{N}} \dot{Z}\left(n - \frac{N}{2}\right) = \dot{Y}\left(n - \frac{N}{2}\right) + e^{-j\frac{2\pi}{N}\left(n - \frac{N}{2}\right)} \dot{Z}\left(n - \frac{N}{2}\right). \quad (19)$$

Процесс вычисления 8-точечного ДПФ путем разбиения его на два 4-точечных ДПФ иллюстрируется на рис. 8.

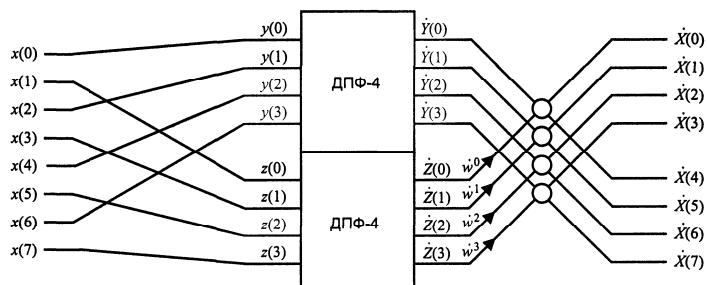


Рис. 8. Вычисление 8-точечного ДПФ с помощью двух 4-точечных ДПФ

Блоки, выполняющие на рис. 8 объединение результатов двух ДПФ, требуют дополнительных комментариев. Каждый такой блок имеет два входных и два выходных сигнала. Один из входных сигналов умножается на комплексную экспоненту w^k , после чего суммируется со вторым входным сигналом и вычитается из него, формируя таким образом два выходных сигнала. Это соответствует реализации формул (18) и (19). Данная операция получила название «бабочки» (butterfly). Расшифровка ее структуры представлена на рис. 9.

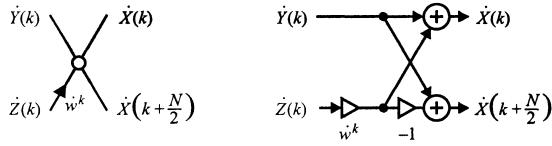


Рис. 9. Условное обозначение «бабочки» БПФ

с прореживанием по времени (слева) и ее структурная схема (справа)

Оценим количество операций, необходимое для вычисления ДПФ указанным способом. Каждое из двух ДПФ половинной размерности требует $\frac{N^2}{4}$ операций. Кроме того, при вычислении окончательных результатов каждый спектральный коэффициент $\dot{z}_{(n)}$ умножается на экспоненциальный комплексный множитель. Это добавляет еще $N/2$ операций. Итого получается $\frac{2N^2}{4} + \frac{N}{2} = \frac{N(N+1)}{2}$, что почти вдвое меньше, чем при вычислении ДПФ прямым способом.

Если $N/2$ тоже является четным числом (то есть если N делится на 4), можно продолжить описанную процедуру, выразив результат через четыре ДПФ размерности $N/4$. Это позволяет еще больше сократить число требуемых вычислительных операций.

Наибольшая степень ускорения вычислений может быть достигнута при $N=2^k$ в этом случае деление последовательностей на две части можно продолжать до тех пор, пока не получатся двухэлементные последовательности, ДПФ которых рассчитывается вообще без использования операций умножения (достаточно вычислить сумму и разность двух отсчетов). Число требуемых при этом пар операций «умножение - сложение» можно оценить как $N \log_2(N)$. Таким образом, вычислительные затраты по сравнению с непосредственным использованием формулы (18) уменьшаются в $N \log_2(N)$ раз. При больших N это отношение становится весьма велико (например, $1024/\log_2(1024)=102,4$, то есть при $N = 1024$ достигается более чем 100-кратное ускорение).

III. ПРАКТИЧЕСКАЯ ЧАСТЬ

Пример вычисления 8-точечного ДПФ с помощью алгоритма БПФ с прореживанием по времени

Алгоритм БПФ для 8-точечного ДПФ представляет собой трёхэтапную процедуру вычисления ДПФ по общей формуле (19) при $i=1, 2, 3$. Формирование начальных условий перед выполнением первого этапа заключается в трёхкратном разбиении исходной 8-точечной последовательности на группы чётных и нечётных отсчётов, а именно:

- 1) 8-точечная последовательность разбивается на две группы 4 точечных с чётными и нечётными отсчёты (см. рис. 8):

$[x(0), x(2), x(4), x(6)]$ - чётные отсчёты,
 $[x(1), x(3), x(5), x(7)]$ - нечётные отсчёты;

2) каждая из двух 4-точечных последовательностей разбивается на две группы 2-точечных с чётными и нечётными отсчётами по порядку следования, считая от нуля, а именно:

последовательность $[x(0), x(2), x(4), x(6)]$:

$[x(0), x(4)]$ - чётные отсчёты,
 $[x(2), x(6)]$ - нечётные отсчёты;
последовательность $[x(1), x(3), x(5), x(7)]$:
 $[x(1), x(5)]$ - чётные отсчёты,
 $[x(3), x(7)]$ - нечётные отсчёты;

3) каждая из четырёх 2-точечных последовательностей разбивается на два отсчёта - чётный и нечётный отсчётами по порядку следования, считая от нуля, т.е. на нулевой (чётный) и первый (нечётный), а именно:

последовательность $[x(0), x(4)]$:

$[x(0)]$ - чётные отсчёты,
 $[x(4)]$ - нечётные отсчёты;

последовательность $[x(2), x(6)]$:

$[x(2)]$ - чётные отсчёты,

$[x(6)]$ - нечётные отсчёты;

последовательность $[x(1), x(5)]$:

$[x(1)]$ - чётные отсчёты,

$[x(5)]$ - нечётные отсчёты;

последовательность $[x(3), x(7)]$:

$[x(3)]$ - чётные отсчёты,

$[x(7)]$ - нечётные отсчёты.

После этого каждой паре отсчётов ДПФ $x_{2m}^0(0)$ и $x_{2m+1}^0(0)$, $m=0, 1, 2, 3$ последовательно присваиваются значения пары отсчётов прореженной 8-точечной последовательности, а именно (см. рис. 10)

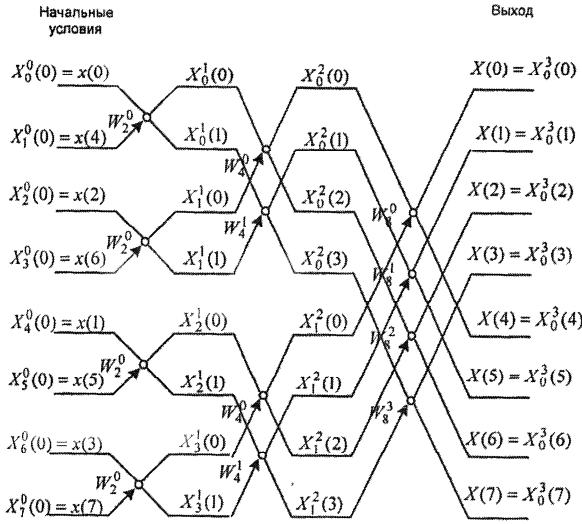


Рис. 10. Направленный граф алгоритма БПФ с прореживанием по времени для 8-точечного ДПФ

Далее выполняется трёхэтапный алгоритм БПФ, направленный граф которого приводится на рис. 10:

1) $i=1$

На первом этапе выполняется четыре операции «бабочка» при $m=0, 1, 2, 3$ с учётом начальных условий

$$\begin{cases} X_0^1(0) = x(0) + W_2^0 x(4), \\ X_0^1(1) = x(0) - W_2^0 x(4); \end{cases} \quad \begin{cases} X_1^1(0) = x(2) + W_2^0 x(6), \\ X_1^1(1) = x(2) - W_2^0 x(6); \end{cases}$$

$$\begin{cases} X_2^1(0) = x(1) + W_2^0 x(5), \\ X_2^1(1) = x(1) - W_2^0 x(5); \end{cases} \quad \begin{cases} X_3^1(0) = x(3) + W_2^0 x(7), \\ X_3^1(1) = x(3) - W_2^0 x(7); \end{cases}$$

Здесь поворачивающий множитель оставлен исключительно для единства «бабочек»; в действительности $W_2^0 = 1$.

2) $i=2$

На втором этапе выполняется четыре операции «бабочка» при $m=0, 1$

$$\begin{cases} X_0^2(k) = X_0^1(k) + W_4^k X_1^1(k); \\ X_0^2(k+2) = X_0^1(k) - W_4^k X_1^1(k); \\ k = 0, 1, \end{cases}$$

$$\begin{cases} X_1^2(k) = X_2^1(k) + W_4^k X_3^1(k); \\ X_1^2(k+2) = X_2^1(k) - W_4^k X_3^1(k); \\ k = 0, 1. \end{cases}$$

3) $i=3$

На третьем этапе выполняются четыре операции «бабочка» при $m=0$

$$\begin{cases} X_0^3(k) = X_0^2(k) + W_8^k X_1^2(k); \\ X_0^2(k+2) = X_0^2(k) - W_8^k X_1^2(k); \\ k = 0, 1, 2, 3. \end{cases}$$

На выходе имеем 8-точечное ДПФ, отсчёты которого следуют в естественном порядке

$$\begin{cases} X(k) = X_0^3(k); \\ k = 0, 1, \dots, N-1. \end{cases}$$

III ЗАДАНИЕ НА

Изучить основные положения теории вычисления ДПФ с помощью алгоритма БПФ с прореживанием по времени.

Изучить функции MATLAB дискретного преобразования Фурье.

Провести вычисления 8-точечного ДПФ с помощью алгоритма БПФ с прореживанием по времени.

IV ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить основные положения теории вычисления ДПФ с помощью алгоритма БПФ с прореживанием по времени.
2. Изучить основные функции MATLAB, необходимые для расчёта ДПФ.
3. В соответствии с планом предложенным в параграфе III провести вычисления 8-точечного ДПФ с помощью алгоритма БПФ с прореживанием по времени с заданными преподавателем параметрами.
4. Проверить в Matlab результаты вычисления ДПФ.

V. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- краткие теоретические сведения по выполняемой работе;
- алгоритмы и листинги программ расчёта ДПФ;
- выводы по проведённой работе.

VI. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Конспект лекций по курсу.
2. Дьяконов В. MATLAB: учебный курс. – СПб: Питер, 2001. – 560 с.: ил.
3. Цифровая обработка сигналов / А.Б. Сергиенко – СПб.: Питер, 2003. – 604 с.: ил.

Практическое занятие № 4

Синтез цифровых фильтров по аналоговому прототипу

I. ЦЕЛЬ И ЗАДАЧИ

Изучение фильтров. Расчёт аналоговых фильтров в среде Matlab. Переход от непрерывной к z – передаточной функции. Составление разностных уравнений. Разработка программного обеспечения, реализующего фильтр с заданными преподавателем параметрами.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Аналоговыми фильтрами называется частотно-избирательная цепь, обеспечивающая пропускание сигналов в определенных полосах частот и подавление в других. Область частот, в которой фильтр пропускает сигнал, называют полосой пропускания, а область частот, в которой ослабление входного сигнала велико - полосой задерживания.

Фильтры применяют для выделения требуемого сигнала из смеси полезных и нежелательных сигналов. В зависимости от взаимного расположения полосы пропускания и полосы задерживания различают следующие виды фильтров:

- фильтры нижних частот (ФНЧ; английский термин — low-pass filter), пропускающие частоты, меньшие некоторой частоты среза ω_0 ;
- фильтры верхних частот (ФВЧ; английский термин — high-pass filter), пропускающие частоты, большие некоторой частоты среза ω_0 ;
- полосовые фильтры (ПФ; английский термин — band-pass filter), пропускающие частоты в некотором диапазоне $\omega_1 \dots \omega_2$ (они могут также характеризоваться средней частотой $\omega_0 = \frac{\omega_1 + \omega_2}{2}$ и шириной полосы пропускания $\omega = \omega_2 - \omega_1$).

- режекторные фильтры (другие возможные названия — заграждающий фильтр, фильтр-пробка, полосно-задерживающий фильтр; английский термин — band-stop filter), пропускающие на выход все частоты, кроме лежащих в некотором диапазоне $\omega_1 \dots \omega_2$ (они тоже могут характеризоваться средней частотой $\omega_0 = \frac{\omega_1 + \omega_2}{2}$ и шириной полосы задерживания $\omega = \omega_2 - \omega_1$)

Идеальная форма АЧХ фильтров этих четырех типов показана на рисунке 1. Однако такая идеальная форма АЧХ не может быть физически реализована. Поэтому в теории аналоговых фильтров разработан ряд методов аппроксимации прямоугольных АЧХ.

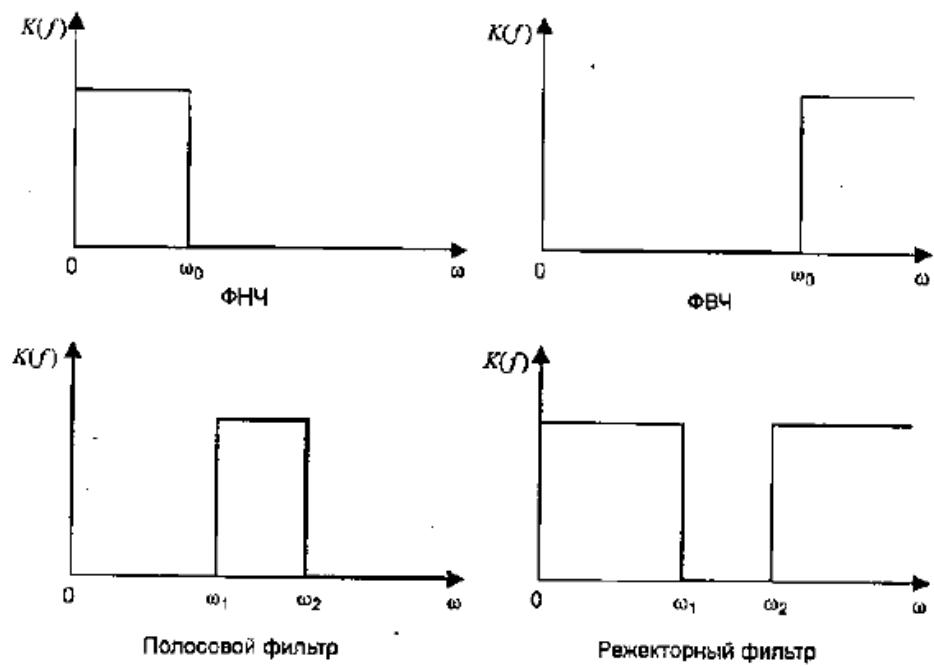


Рисунок 1 – АЧХ фильтров различного типа

2.1. Фильтр Баттерворт

Передаточная функция фильтра Баттервортса (Butterworth filter) не имеет нулей, а ее полюсы равномерно расположены на окружности единичного радиуса.

Благодаря такому размещению полюсов формула для АЧХ фильтра Баттервортса оказывается очень простой:

$$K(\omega) = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}}}$$

где ω_0 частота среза (для фильтра прототипа она равна 1рад/с); n – порядок фильтра.

2.2. Фильтр Чебышева первого рода

Передаточная функция фильтра Чебышева первого рода также не имеет нулей, а ее полюсы расположены в левой половине эллипса на s - плоскости. АЧХ фильтра Чебышева первого рода описывается следующим образом:

$$K(\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

где ω_0 частота среза; ε параметр, определяющий величину пульсаций АЧХ в полосе пропускания; $T_n(x)$ полином Чебышева n -го порядка.

По сравнению с фильтром Баттервортса того же порядка фильтр Чебышева обеспечивает более крутой спад АЧХ в области перехода от полосы пропускания к полосе задерживания.

2.3. Фильтр Чебышева второго рода

Передаточная функция фильтра Чебышева второго рода в отличие от

предыдущих случаев, имеет и нули, и полюсы. Она связана с функцией передачи фильтра Чебышева первого рода следующим образом:

$$H_2(s) = 1 - H_1(1/s)$$

Здесь $H_1(s)$ и $H_2(s)$ функции передачи фильтров Чебышева первого и второго рода соответственно.

АЧХ фильтра Чебышева второго рода описывается следующим образом:

$$K(\omega) = \frac{1}{\sqrt{1 + \frac{\varepsilon^2}{T_n^2\left(\frac{\omega}{\omega_0}\right)}}}$$

где ω_0 частота среза; ε параметр, определяющий величину пульсаций АЧХ в полосе пропускания; $T_n(x)$ полином Чебышева n-го порядка.

2.4. Фильтр Бесселя

В отличие от фильтров предыдущих типов, фильтры Бесселя (Bessel filter) не аппроксимируют прямоугольную АЧХ – их АЧХ по форме близка к гауссовой кривой (точнее, стремится к ней с ростом порядка фильтра). Практическая ценность фильтров Бесселя определяется тем, что для них зависимость группового времени задержки от частоты является максимально гладкой в точке $\omega_0 = 0$ и групповая задержка очень мало меняется в полосе пропускания.

Передаточная функция фильтра Бесселя имеет только полюсы, лежащие на окружности с центром в положительной области вещественной оси. Сама передаточная функция имеет следующий вид:

$$H(s) = d_0 / \sum_{k=0}^n d_k s^k$$

2.5. Расчёт фильтров в MATLAB

Для расчета аналогового фильтра необходимо выполнить две основные операции: рассчитать ФНЧ - прототип и преобразовать его к нужному типу фильтра с заданными частотами среза.

Действия, выполняемые функциями расчета аналоговых фильтров:

1. Производится расчет фильтра-прототипа с заданными параметрами АЧХ.
2. Полученные нули и полюсы преобразуются в параметры пространства состояний.
3. Производится преобразование фильтра-прототипа к требуемому типу с заданными частотами среза.

2.6. Расчёт ФНЧ - прототипа

В MATLAB расчёт аналоговых фильтров-прототипов для рассмотренных фильтров производится с помощью следующих функций:

– $[z,p,k]=buttap(n)$ – для расчёта аналогового фильтра-прототипа Баттервортса;

– $[z,p,k]=cheb1ap(n,Rp)$ – для расчёта аналогового фильтра-прототипа Чебышева I-го рода;

– $[z,p,k]=cheb2ap(n,Rs)$ – для расчёта аналогового фильтра-прототипа Чебышева II-го рода;

– $[z,p,k]=besselap(n)$ – для расчёта аналогового фильтра-прототипа Бесселя.

Входные параметры для соответствующих фильтров-прототипов:

n – порядок фильтра (целочисленный);

Rp – уровень пульсаций в полосе пропускания (в дБ);

Rs – уровень пульсаций в полосе задерживания (в дБ).

Все функции MATLAB для расчета аналоговых прототипов возвращают векторы-столбцы нулей и полюсов функции передачи, а также значение коэффициента усиления.

2.7. Преобразование способов описания линейных цепей

Необходимость в преобразовании описаний часто возникает из-за того, что функции расчета цепей (такие как рассматриваемые функции расчета фильтров-прототипов) дают результат в одной форме, а функция, например построения частотной характеристики, требует задания входных параметров в другой форме.

Формы описания цепей в именах функций обозначаются следующим образом:

tf - коэффициенты полиномов числителя и знаменателя функции передачи (transfer function);

zp - нули и полюсы (zeros and poles);

ss - описание в пространстве состояний (state-space).

Далее кратко рассматриваются конкретные функции преобразования описаний цепей.

Для входных и выходных параметров используются следующие обозначения:

– функция передачи:

b - вектор-строка коэффициентов (в порядке убывания степеней) числителя функции передачи;

a - вектор-строка коэффициентов (в порядке убывания степеней) знаменателя функции передачи;

– нули и полюсы:

z - вектор нулей (столбец);

p - вектор полюсов (столбец);

k - коэффициент усиления (скаляр);

– пространство состояний:

А - квадратная матрица связи вектора состояния и его производной;
В - вектор-столбец связи входного сигнала и производной вектора состояния;

С - вектор-строка связи выходного сигнала и вектора состояния;

Д - скалярный коэффициент связи выходного и входного сигналов.

Функция *tf2zp* преобразует наборы коэффициентов полиномов числителя и знаменателя функции передачи в векторы нулей и полюсов, рассчитывая также значение общего коэффициента усиления:

$[z, p, k] = \text{tf2zp}(b, a)$

Функция *zp2tf* является обратной по отношению к функции *tf2zp*: она осуществляет преобразование коэффициента усиления, а также векторов нулей и полюсов функции передачи в коэффициенты полиномов ее числителя и знаменателя:

$[b, a] = \text{zp2tf}(z, p, k)$

Функция *tf2ss* преобразует наборы коэффициентов полиномов числителя и знаменателя функции передачи в параметры представления цепи в пространстве состояний:

$[A, B, C, D] = \text{tf2ss}(b, a)$

Функция *ss2tf* является обратной по отношению к функции *tf2ss*: она преобразует параметры пространства состояний в коэффициенты полиномов функции передачи цепи:

$[b, a] = \text{ss2tf}(A, B, C, D);$

Функция *zp2ss* преобразует нули, полюсы и коэффициент усиления цепи в ее параметры пространства состояний:

$[A, B, C, D] = \text{zp2ss}(z, p, k)$

Функция *ss2zp* является обратной по отношению к функции *zp2ss*, преобразуя параметры пространства состояний в нули, полюсы и коэффициент усиления цепи:

$[z, p, k] = \text{ss2zp}(A, B, C, D)$

Функция *residue*.

Идентификатор последней функции, позволяющей преобразовывать описания линейных цепей, выпадает из общего ряда. Это связано с тем, что данное преобразование не является специфическим для обработки сигналов - оно сводится к разложению дробно-рациональной функции на простейшие дроби и часто применяется в математике. По этой же причине данная функция относится не к пакету *Signal Processing*, а к базовой библиотеке MATLAB.

Преобразование функции передачи, заданной в виде коэффициентов полиномов числителя и знаменателя, в сумму простых дробей производится с помощью функции *residue*. Она же осуществляет и обратное преобразование;

нужное направление преобразования выбирается в зависимости от числа входных параметров.

При двух входных параметрах производится разложение функции передачи на простые дроби:

$$[r, p, k] = \text{residue}(b, a)$$

Здесь b и a — коэффициенты полиномов числителя и знаменателя функции передачи соответственно. Выходные параметры — векторы-столбцы полюсов (p) и соответствующих им вычетов (r), а также строка коэффициентов целой части k .

2.8. Преобразования фильтров-прототипов

Следующим этапом после расчета фильтра-прототипа является его преобразование с целью получения фильтра заданного вида с требуемыми частотами среза. Для этого используются приведенные ниже четыре функции MATLAB. Принцип составления их имен следующий: сначала идет сокращение lp , означающее, что исходным фильтром является ФНЧ (low-pass), потом следует символ преобразования 2 и в конце стоит обозначение типа результирующего фильтра:

– Изменение частоты среза ФНЧ (low-pass) производится функцией $lp2lp$:

$$[bl, al] = lp2lp(b, a, w0)$$

$$[Al, Bl, Cl, Dl] = lp2lp(A, B, C, D, w0)$$

– Преобразование ФНЧ в ФВЧ (high-pass) производится функцией $lp2hp$:

$$[bl, al] = lp2hp(b, a, w0)$$

$$[A1, B1, C1, D1] = lp2hp(A, B, C, D, w0)$$

– преобразование ФНЧ в полосовой фильтр (band-pass) производится функцией $lp2bp$:

$$[bl, al] = lp2bp(b, a, w0, Bw)$$

$$[A1, B1, C1, D1] = lp2bp(A, B, C, D, w0, Bw)$$

– преобразование ФНЧ в режекторный фильтр (band-stop) выполняется функцией $lp2bs$:

$$[bl, al] = lp2bs(b, a, w0, Bw)$$

$$[A1, B1, C1, D1] = lp2bs(A, B, C, D, w0, Bw)$$

Входными параметрами функций являются описание фильтра (в виде коэффициентов полиномов числителя и знаменателя функции передачи - b , a или в пространстве состояний - A , B , C , D), средняя частота $w0$ и ширина Bw полосы задерживания фильтра (в радианах в секунду). Возвращаемый результат - пересчитанные параметры фильтра.

2.9. Выбор порядка фильтра

Рассмотренные выше функции расчета фильтров требуют задания в качестве входных параметров порядка фильтра и его частоты среза. При этом понятие частоты среза для фильтров разных типов определяется по-разному.

Однако исходными данными при разработке фильтров, как правило, являются другие параметры: частотные границы полос пропускания (ω_p) и задерживания (ω_s), а также допустимая неравномерность АЧХ в полосе пропускания (R_p) и минимально необходимое затухание в полосе задерживания (R_s). Серые области на рисунке 2 демонстрируют допуски, в которые должна укладываться АЧХ фильтра в полосах пропускания и задерживания.

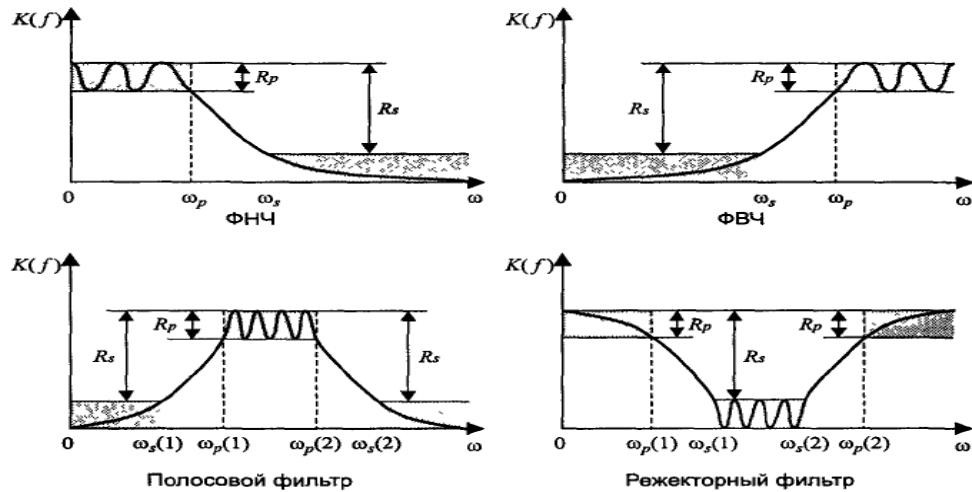


Рисунок 2 – Задание исходных параметров для расчета ФНЧ, ФВЧ, полосовых и режекторных фильтров

Выбрать минимально необходимый порядок фильтра позволяют следующие однотипные функции пакета Signal Processing:

$[n, Wn] = \text{buttord} (Wp, Ws, Rp, Rs, 's')$
 $[n, Wn] = \text{cheblord} (Wp, Ws, Rp, Rs, 's')$
 $[n, Wn] = \text{cheb2ord} (Wp, Ws, Rp, Rs, 's')$

Входной параметр R_p - допустимый уровень пульсаций в полосе пропускания (в децибелах), R_s - минимально необходимое затухание в полосе задерживания (в децибелах). Параметры Wp и Ms задают границы полос пропускания и задерживания, способ задания этих параметров зависит от типа проектируемого фильтра:

- ФНЧ: Wp и Ws — числа, при этом должно выполняться неравенство $Wp < Ws$;
- ФВЧ: Wp и Ms — числа, при этом должно выполняться неравенство $Wp > Ms$;
- полосовой фильтр: Wp и Ws — двухэлементные векторы, при этом должны выполняться неравенства $Ws(1) < Wp(1) < Wp(2) < Ws(2)$;
- режекторный фильтр: Wp и Ws — двухэлементные векторы, при этом должны выполняться неравенства $Wp(1) < Ws(1) < Ws(2) < Wp(2)$.

Значения параметров Wp и Ws были обозначены на рис. 2 как ω_p и ω_s соответственно. Выходными параметрами являются минимально необходимый

для выполнения заданных требований порядок фильтра n и частота среза фильтра W_n . Эти параметры должны затем использоваться при вызове функции расчета фильтра. Возврат значения W_n избавляет пользователя от забот, связанных с тем, что при расчете разных фильтров понятие частоты среза имеет разный смысл.

Поскольку порядок фильтра – величина целочисленная, то обычно оказывается, что фильтр минимально необходимого порядка обеспечивает некоторый запас по исходным параметрам. Этот запас можно использовать по разному – либо сделать пульсации в полосе пропускания точно равными заданным, но увеличить затухание в полосе задерживания, либо точно выдержать заданное затухание в полосе задерживания, уменьшив при этом пульсации в полосе пропускания. Поведение функций выбора порядка фильтра в этом аспекте определяется тем, что при расчете фильтра должны будут использоваться те же параметры пульсаций R_p и R_s , что и при выборе порядка фильтра. Поэтому для фильтров Баттервортса и Чебышева первого рода будет увеличиваться затухание в полосе задерживания, для фильтров Чебышева второго рода - уменьшаться пульсации в полосе пропускания.

III. ЗАДАНИЕ

Изучить аналоговые фильтры.

Изучить основные функции MATLAB, необходимые для расчёта аналоговых фильтров.

Рассчитать аналоговый фильтр.

Синтезируйте на его основе цифровой фильтр и следуйте его работе.

IV. ПОРЯДОК ВЫПОЛНЕНИЯ

- Изучить аналоговые фильтры.
- Изучить основные функции MATLAB, необходимые для расчёта аналоговых фильтров.
- Рассчитать аналоговый фильтр с заданными преподавателем параметрами.
- Сравнить результаты проведённого расчёта фильтра с результатами расчёта при использовании специальных функций MATLAB непосредственного расчёта фильтров.
- Произвести выбор порядка фильтра по заданным преподавателем параметрам.

V. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- краткие теоретические сведения по выполняемой работе;
- алгоритмы и листинги программ расчёта фильтра;
- выводы по проведённой работе.

VI. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определения фильтрам НЧ, ВЧ, ...
2. Основные характеристики фильтров Баттервортса, Чебышева, Бесселя ...
3. Какой из перечисленных типов фильтров имеет наиболее крутой спад АЧХ в области заграждения.
4. Какой из перечисленных типов фильтра имеет минимальный фазовый сдвиг в рабочей полосе частот.
5. Перечислите параметры, которые необходимо задать при расчете фильтров НЧ, ВЧ, режекторных ...

VIII. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Конспект лекций по курсу.
2. Дьяконов В. MATLAB: учебный курс. – СПб: Питер, 2001. – 560 с.: ил.
3. Цифровая обработка сигналов / А.Б. Сергиенко – СПб.: Питер, 2003. – 604 с.: ил.

Практическое занятие № 5

Свёртка и её математические свойства

I. ЦЕЛЬ И ЗАДАЧИ

Изучить операцию свёртки и на практике познакомиться с её свойствами.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Свёртка — это математическая процедура, которая из двух исходных сигналов формирует по определённому правилу третий сигнал. В цифровой обработке она играет важнейшую роль. Дело в том, что цифровые системы можно описывать их импульсными характеристиками, благодаря использованию *импульсной декомпозиции*. Свёртка устанавливает взаимосвязь сигнала на входе и на выходе системы через её импульсную характеристику. Свёртка лежит в основе всей цифровой обработки сигналов.

Дельта-функция и импульсная характеристика

Огромное значение для ЦОС имеют следующие два понятия. Первое из них — это дискретная дельта-функция, обычно обозначаемая символом $\delta[n]$ (Рис. 1). Для дискретных систем она называется *единичным импульсом*, или *единичным отсчётом*, и представляет собой сигнал, в котором нулевой отсчёт имеет значение единицы, а все остальные отсчёты равны нулю.

Второе понятие — это *импульсная характеристика* системы (Рис. 1). Как можно догадаться по самому названию, импульсная характеристика — это реакция системы на единичный импульс. Если две любые системы имеют хотя бы какие-то различия, их импульсные характеристики будут обязательно отличаться. Если сигналы на входе и выходе системы традиционно обозначают $x[n]$ и $y[n]$ соответственно, то импульсную характеристику системы принято обозначать $h[n]$. Конечно, это не является обязательным, и можно использовать другие обозначения, если они кажутся вам более удобными: например, импульсную характеристику фильтра можно обозначить $f[n]$.

Любой импульсный сигнал, являющийся компонентом разложения при импульсной декомпозиции, может быть представлен как единичный импульс, сдвинутый на соответствующую величину по временной оси и умноженный на значение отсчёта исходного сигнала в этот же момент времени. Для примера рассмотрим сигнал $a[n]$, все значения которого равны нулю, кроме 8-го отсчёта, принимающего значение – 3. Этот сигнал может рассматриваться как единичный импульс, сдвинутый (задержанный) на 8 периодов дискретизации и умноженный на – 3, или в форме математического выражения:

$$a[n] = -3\delta[n - 8].$$

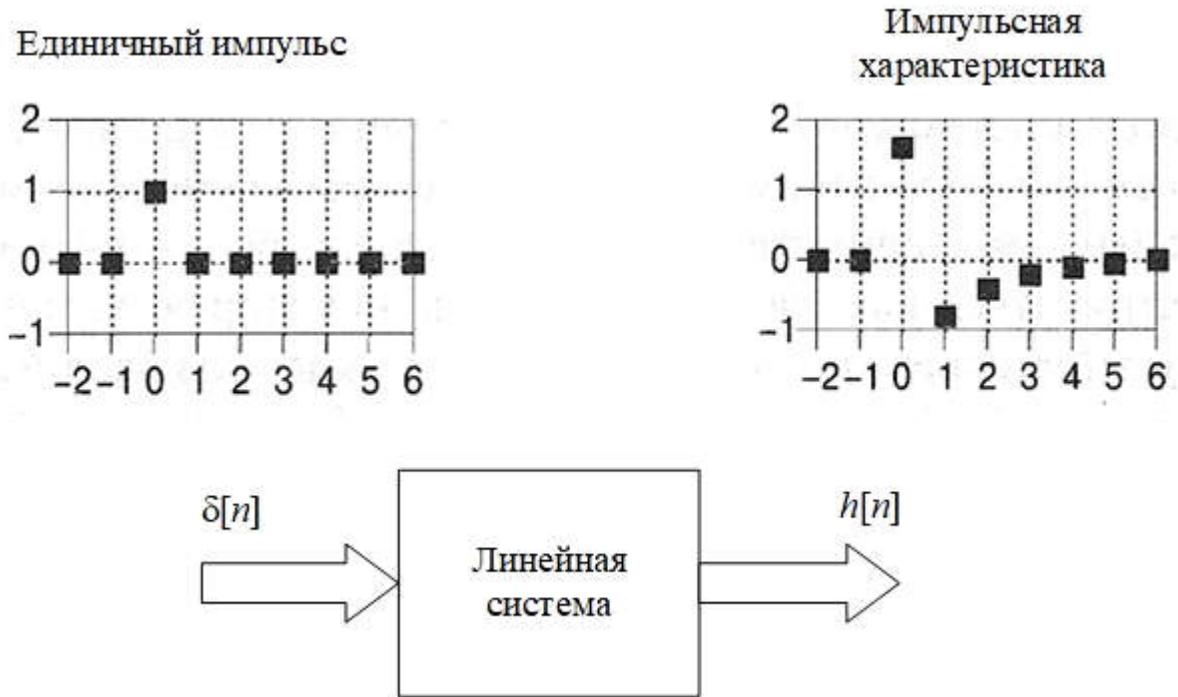


Рис. 1. Иллюстрация понятий единичного импульса и импульсной характеристики. Единичный импульс – это дискретный сигнал, все отсчёты которого, кроме нулевого, равны нулю. Значение нулевого отсчёта равно единице. Единичный импульс является дискретным аналогом дельта-функции, которую принято обозначать символом $\delta(t)$. Импульсная характеристика линейной системы – это реакция системы на единичный импульс. Импульсную характеристику обычно обозначают $h[n]$.

Допустим, на вход системы подан сигнал $a[n] = -3\delta[n - 8]$. Каким окажется её выход? Чтобы ответить на этот вопрос, придётся вспомнить ранее рассмотренные свойства однородности и инвариантности к сдвигу. Умножение и задержка сигнала на входе приводят к аналогичным преобразованиям выходного сигнала. Таким образом, если входному воздействию $\delta[n]$ соответствует реакция на выходе $h[n]$ (импульсная характеристика системы), то сигналу $-3\delta[n - 8]$ на входе системы будет соответствовать выходной сигнал $-3h[n - 8]$. Другими словами, сигналом на выходе окажется импульсная характеристика системы, сдвинутая во времени и усиленная по амплитуде, так же, как сдвинут и усилен единичный импульс на входе. Итак, если вам известна импульсная характеристика системы, то вычислить её реакцию на произвольный импульсный сигнал с единственным ненулевым отсчётом не составляет труда.

2. Свёртка

Нам осталось подвести итог материалу, посвящённому определению реакции на выходе системы при произвольных входных воздействиях. Во-первых, входной сигнал путём декомпозиции представляется в виде совокупности импульсных сигналов с единственным ненулевым отсчётом, каждый из которых может рассматриваться как *единичный импульс*, умноженный на некоторую величину и сдвинутый во времени. Во-вторых, реакция системы на каждый из входных импульсных сигналов может быть получена путём соответствующего усиления и сдвига *импульсной характеристики системы*. И наконец, в-третьих, реакция системы на исходный входной сигнал рассчитывается суммированием реакций системы на все компоненты декомпозиции в отдельности. Таким образом, если нам известна импульсная характеристика системы, то мы можем определить её реакцию на любое входное воздействие. Фактически это означает, что нам известно всё о системе. При описании *линейной системы* импульсная характеристика является исчерпывающей (заметим, однако, что в последующих главах мы рассмотрим и другие характеристики систем, которые не несут новой информации, а просто представляют её в другой форме).

Импульсная характеристика может обозначаться и другими терминами в зависимости от сферы применения. Например, при описании фильтров возможно использование понятия *ядра фильтра*, *ядра свёртки* или просто *ядра*. В приложениях цифровой обработки изображений вместо импульсной характеристики используется термин *функция рассеяния точки*. Несмотря на небольшие различия в использовании указанных терминов, все они обозначают одно и то же: реакцию системы на воздействие в виде дельта-функции или единичного импульса.

Операция свёртки является обычной математической операцией, такой, как, например, умножение, сложение или интегрирование. Аналогично любой другой операции при свёртке двух сигналов (дискретных последовательностей) получают единственный третий сигнал. Процедура свёртки применяется в различных разделах математики, в частности в теории вероятностей и математической статистике. Теория линейных систем использует свёртку для установления взаимосвязи между входом системы, её импульсной характеристикой и выходным сигналом.

Рис. 2 иллюстрирует понятие свёртки применительно к линейным системам. Сигнал $x[n]$ поступает на вход линейной системы, описываемой импульсной характеристикой $h[n]$, и вызывает на выходе системы реакцию $y[n]$. В форме уравнения это записывается так: $x[n] * h[n] = y[n]$. То есть сигнал на выходе

линейной системы равен свёртке сигнала на входе системы и её импульсной характеристики. Подобно тому как сложение обозначается плюсом (+), а умножение – крестиком (\times), свёртка обозначается звёздочкой (*). К сожалению, в большинстве языков программирования символ * принято использовать для обозначения умножения. Таким образом, звёздочка в тексте программы означает умножение, а та же звёздочка в математических выражениях означает свёртку.

На Рис. 3 проиллюстрировано использование свёртки для реализации низкочастотной и высокочастотной фильтрации. В этом примере входным сигналом является сумма медленно возрастающего сигнала (спектр лежит в области низких частот) и синусоиды, у которой на интервале наблюдения укладывается несколько периодов повторения (спектр относительно высокочастотный). В случае низкочастотного фильтра (а) импульсная характеристика в форме дуги позволяет проходить на выход системы только медленно меняющейся составляющей. В случае высокочастотного фильтра (б), напротив, импульсная характеристика системы такова, что на выходе системы присутствует только относительно быстро меняющийся компонент входного сигнала.

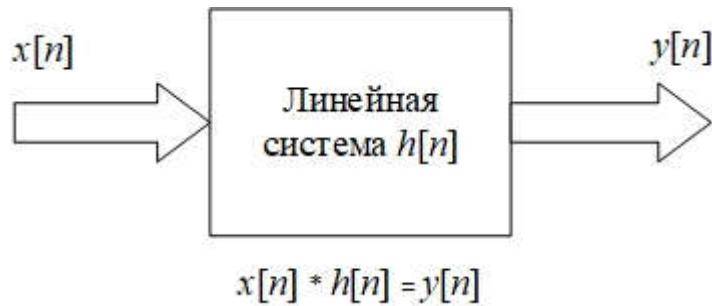


Рис. 2. Использование операции свёртки в цифровой обработке сигналов. Сигнал на выходе линейной системы равен свёртке входного сигнала и импульсной характеристики системы. В математических выражениях операция свёртки обозначается звёздочкой ().*

Рис. 4 демонстрирует другие два примера использования свёртки для описания процесса обработки сигналов. Инвертирующий аттенюатор (а) «переворачивает» сигнал «вверх ногами» и уменьшает его амплитуду. Взятие дискретной производной (вычисление первой разности) даёт на выходе системы сигнал, пропорциональный скорости изменения входного сигнала (б).

Обратите внимание на длины сигналов, приведённых на Рис. 3 и 4. Входные сигналы включают 81 дискретный отсчёт, в то время как импульсные характеристики состоят только из 31 отсчёта. Дело в том, что в большинстве приложений цифровой обработки входные сигналы рассматриваются на

интервалах в сотни, тысячи и даже миллионы отсчётов. Импульсная же характеристика значительно короче, скажем, в сотни раз. Математическое определение операции свёртки не накладывает никаких ограничений на длительность этих сигналов. Оно, однако, позволяет определить длину сигнала на выходе, равную сумме длины входного сигнала и длины импульсной характеристики минус один. Так для примеров, показанных на Рис. 6.3 и 6.4, длительность выходных сигналов составляет $81 + 31 - 1 = 111$. Отсчёты входных сигналов нумеруются с 0-го по 80-й, отсчёты импульсных характеристик — с 0-го по 30-й, а отсчёты выходных сигналов — с 0-го по 110-й.

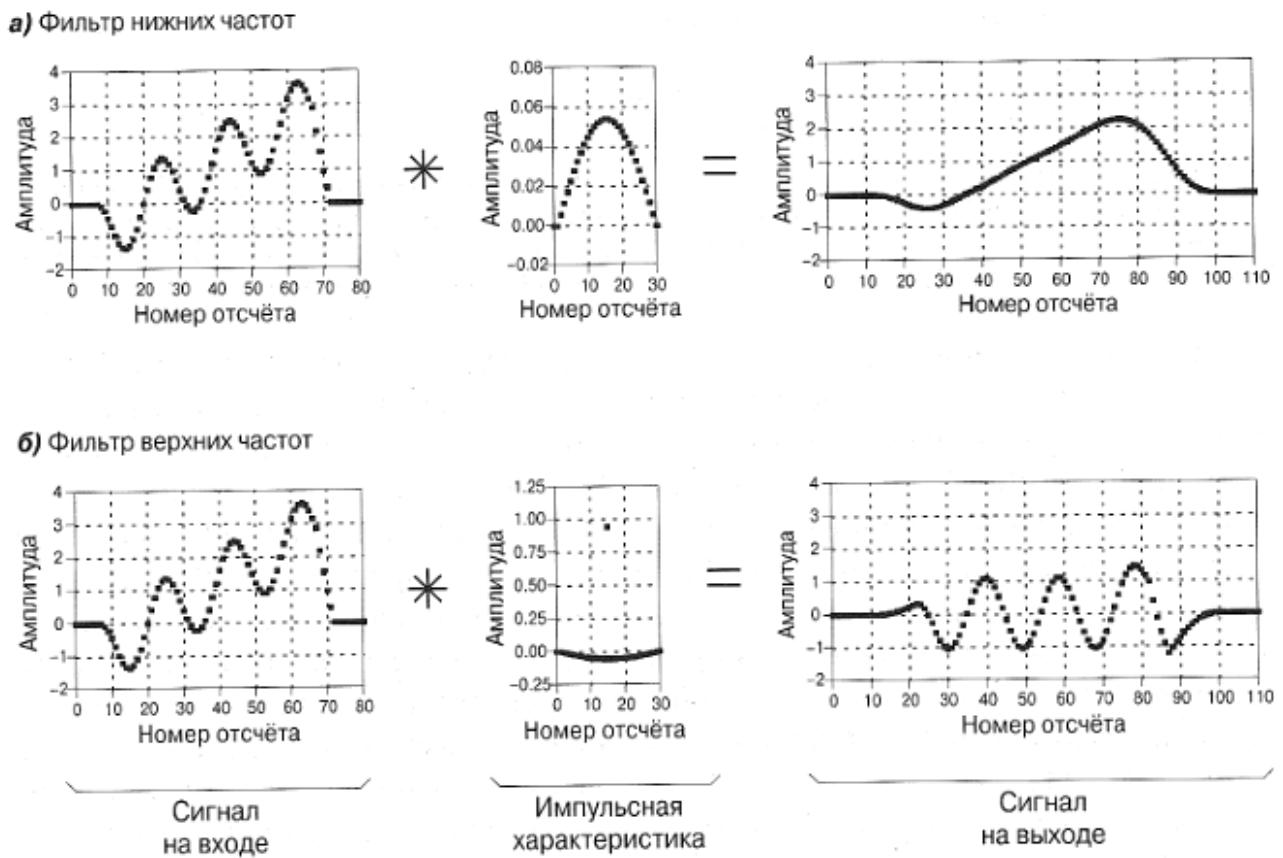
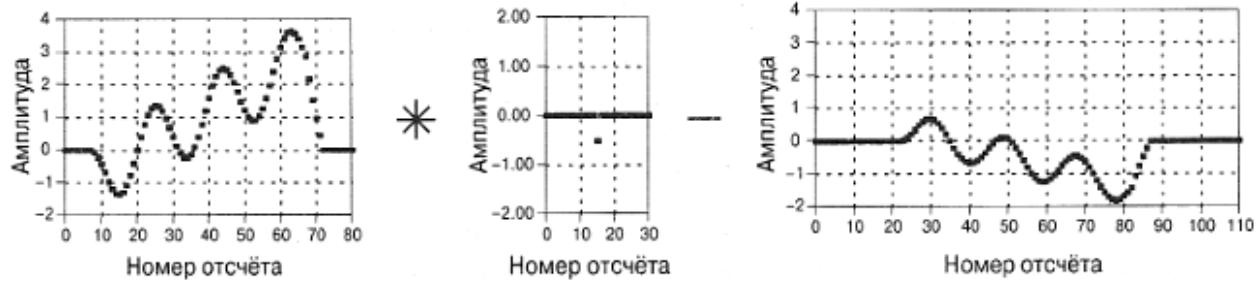


Рис. 3. Пример использования свёртки для описания систем низкочастотной и высокочастотной фильтрации. В приведённом примере на вход системы поступает сумма медленно возрастающего сигнала синусоиды, у которой на интервале наблюдения укладывается несколько периодов повторения. Эти два компонента входного сигнала удаётся разделить за счёт использования выбранных соответствующим образом импульсных характеристик.

Приступим теперь к подробному рассмотрению математического описания свёртки. В том качестве, в котором свёртка используется в ЦОС, её следует рассматривать с двух позиций: по отношению к входному сигналу и по отношению к сигналу на выходе. Представление операции свёртки со стороны входа системы подразумевает анализ того, как каждый отсчёт входного сигнала

влияет на общую реакцию системы на её выходе. Представление операции свёртки со стороны выхода, напротив, означает понимание того, как каждый отсчёт выходного сигнала зависит от совокупности входных отсчётов.

а) Инвертирующий аттенюатор



б) Дискретная производная

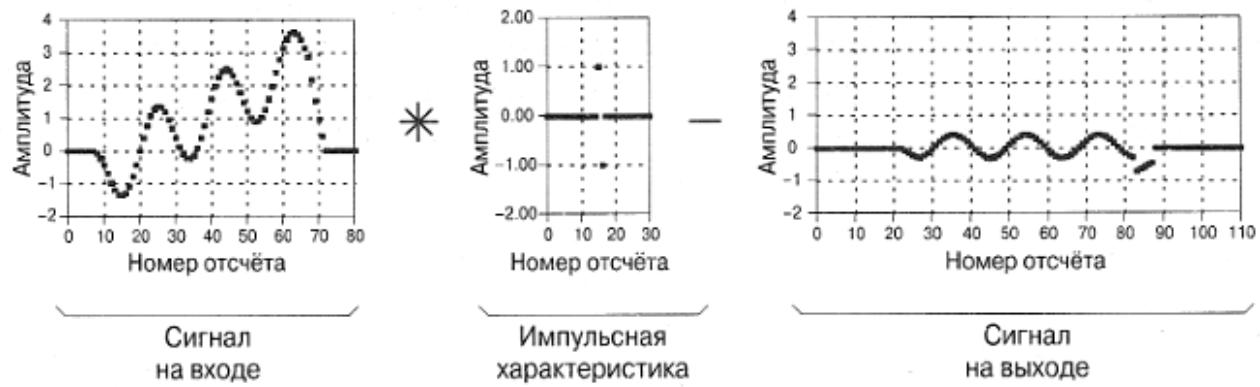


Рис. 4. Использование свёртки для описания процесса обработки сигналов. Большинство систем ЦОС имеют достаточно простые импульсные характеристики. Приведённые примеры показывают, что даже система с импульсной характеристикой, имеющей лишь несколько ненулевых отсчётов, может осуществлять значительные преобразования сигналов.

Следует помнить, что оба указанных способа анализа в действительности являются двумя сторонами одного и того же процесса — математической операции свёртки. Представление свёртки со стороны входного сигнала позволяет понять огромную роль этой процедуры в теории ЦОС. В то же время представление свёртки со стороны выходного сигнала даёт детальное понимание её математической сути. В этом проявляется одна из типичных для ЦОС проблем: интуитивно понятные принципы цифровой обработки должны быть описаны сложным математическим языком, чтобы ваши идеи могли стать доступны другим.

III. ПРАКТИЧЕСКАЯ ЧАСТЬ

В системах математических вычислений Matlab или Octave операция свертки может быть реализована с помощью функции `conv`.

Например, фильтры ВН и ВН могут быть реализованы как свёртка соответствующих импульсных функций с тестовым сигналом:

```
graphics_toolkit ("fltk");
t = 1:1:256;
x = 256 * sin(t * pi / 32) + 25 * sin (2 * t);
h_lp = sinc(-2:0.1:2) / 10;

imp = zeros(1, length(h_lp));
imp(round(length(imp) / 2)) = 1;
h_hp = imp - h_lp;

subplot (3, 1, 1);
plot(x);
grid on;
xlabel ("n");
title ("x");

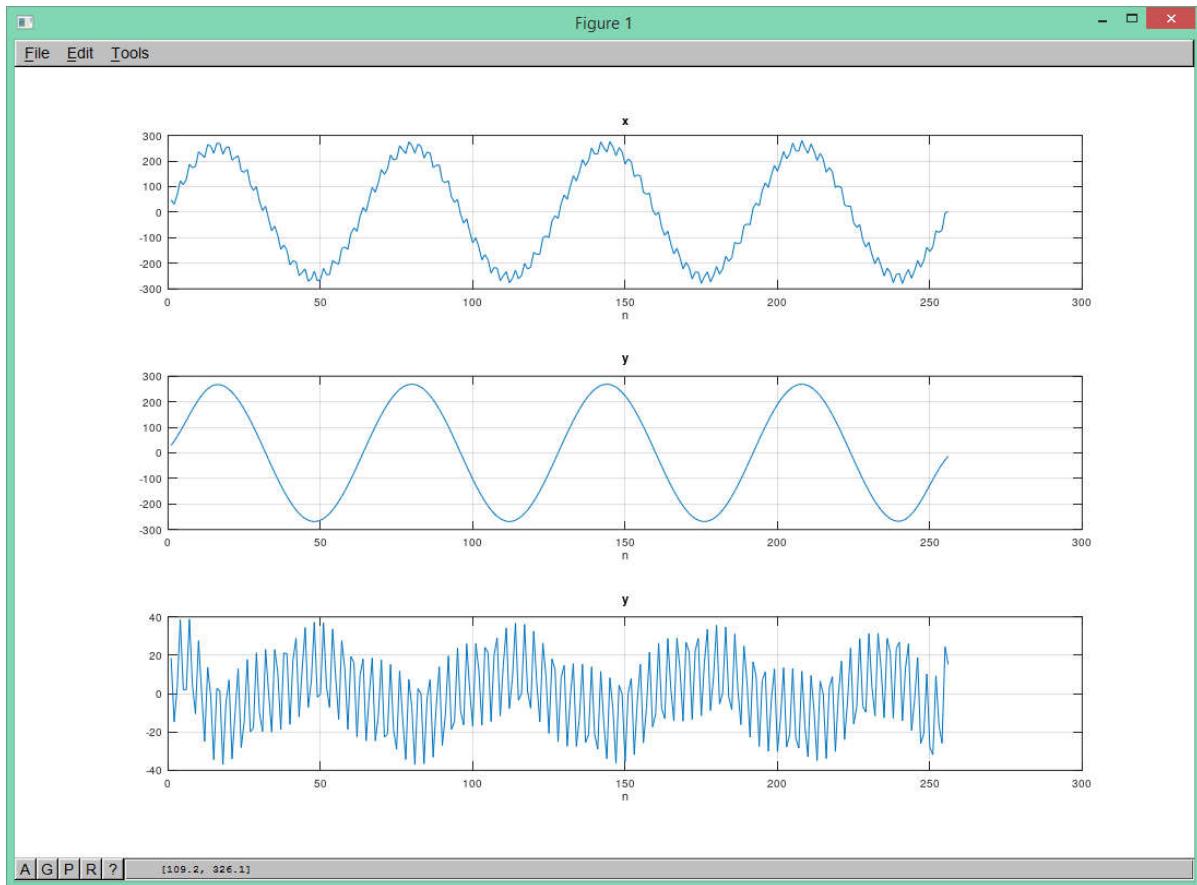
y = conv(x, h_lp, "same");
subplot (3, 1, 2);
plot(y);
grid on;
xlabel ("n");
title ("y");

y = conv(x, h_hp, "same");
subplot (3, 1, 3);
plot(y);
grid on;
xlabel ("n");
title ("y");

figure;
plot(h_lp);
grid on;
xlabel ("n");
title ("h - ФНЧ");

figure;
plot(h_hp);
grid on;
xlabel ("n");
title ("h - ФВЧ");
```

В результате работы представленного кода получим следующие графики:



На верхней диаграмме показан исходный сигнал, ниже результат его свёртки с импульсной характеристикой ФНЧ, а ещё ниже – ФВЧ.

IV. ЗАДАНИЕ

Изучите операцию свертки и экспериментально проверьте её свойства.

V. ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить основные положения теории свёртки дискретных сигналов.
2. Выполните свёртку тестового сигнала с различными импульсными функциями.
3. Выполните свёрстку тестового сигнала с единичным импульсом. Добейтесь усиления и сдвига, за счет изменения единичного сигнала.
4. С помощью свёртки быть выполнение преобразования, аналогичные дифференцированию и интегрированию.
5. Проверьте свойства коммутативности, ассоциативности, дистрибутивности свёрстки, а также центральную предельную теорему.
6. Сделайте выводы о проделанной работе.

VI. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- алгоритмы и листинги разработанных программ;
- графики и иные результаты численного моделирования;
- выводы по проведённой работе.

VII. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит; пер. с англ. А.Ю. Линовича, С.В. Витязева, И.С. Гусинского. – М.: Додэка-XXI, 2012. – 720 с.
2. Умняшкин С.В. Истории цифровой обработки сигналов: Учебное пособие // М.: ТЕХНОСФЕРА, 2016. – 528 с.
3. Лайонс Р. Цифровая обработка сигналов. Пер. С англ. – М.: ООО «Бином-Пресс», 2015. – 656 с.
4. Основы цифровой обработки сигналов: Курс лекций / Авторы: А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б. Соловьёва, И.И. Гук. – Спб.: БХВ-Петербург, 2003. – 608 с.
5. Гадзиковский В.И. Цифровая обработка сигналов. – М.: СОЛОН-ПРЕСС, 2015. – 766 с.

Практическое занятие № 6

Однородные фильтры

I. ЦЕЛЬ И ЗАДАЧИ

Изучение спектрального анализа дискретных сигналов. Расчёт спектра дискретного сигнала в Matlab с применением дискретного преобразования Фурье.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Однородные фильтры, или *фильтры скользящего среднего*, получили наиболее широкое распространение в ЦОС, благодаря простоте их реализации. Кроме того, они являются оптимальными при решении такой классической задачи ЦОС, как подавление аддитивного шума при одновременном сохранении скорости нарастания переходной характеристики. Другими словами, однородные фильтры превосходят все остальные виды цифровых фильтров при обработке информации, представленной во временной области. В то же время эти фильтры оказываются малоэффективны при обработке информации в частотной области, так как не способны разделять между собой сигналы, лежащие в разных диапазонах частот. Имеется несколько модификаций однородных фильтров, в которых частотная избирательность повышается за счёт увеличения времени обработки.

Однородные нерекурсивные фильтры

Однородные фильтры часто называют *фильтрами скользящего среднего*, потому что они основаны на усреднении некоторого множества отсчётов входного сигнала, что выражается следующим разностным уравнением:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \quad (1)$$

Здесь x – входной сигнал, y – выходной сигнал, M – число усредняемых отсчётов. В данном уравнении все используемые отсчёты расположены по одну сторону от результирующего выходного отсчёта.

Например, для однородного фильтра 5-го порядка 10-й отсчёт выходного сигнала определяется следующим уравнением:

$$y[10] = \frac{x[10] + x[11] + x[12] + x[13] + x[14]}{5}$$

Возможен другой вариант построения однородного нерекурсивного фильтра, при котором выбирается равное число отсчётов входного сигнала по обе стороны от текущего выходного отсчёта:

$$y[10] = \frac{x[8] + x[9] + x[10] + x[11] + x[12]}{5}$$

Такой вариант усреднения связан с изменением пределов вычисления суммы в исходном выражении фильтра:

$$y[i] = \frac{1}{M} \sum_{j=-\frac{M-2}{2}}^{\frac{M-2}{2}} x[i+j]$$

Несмотря на то что программа для алгоритма одностороннего усреднения несомненно проще, данный алгоритм приводит к возникновению относительного сдвига между входным и выходным сигналами.

Однородные фильтры – это фильтры с самой простой *импульсной характеристикой*. Например, однородный нерекурсивны фильтр 5-го порядка имеет импульсную характеристику вида $\{..., 0, 0, 1/5, 1/5, 1/5, 1/5, 1/5, 0, 0, ...\}$. То есть в основе однородных нерекурсивных фильтров лежит алгоритм свёртки входного сигнала с прямоугольным импульсом единичной площади.

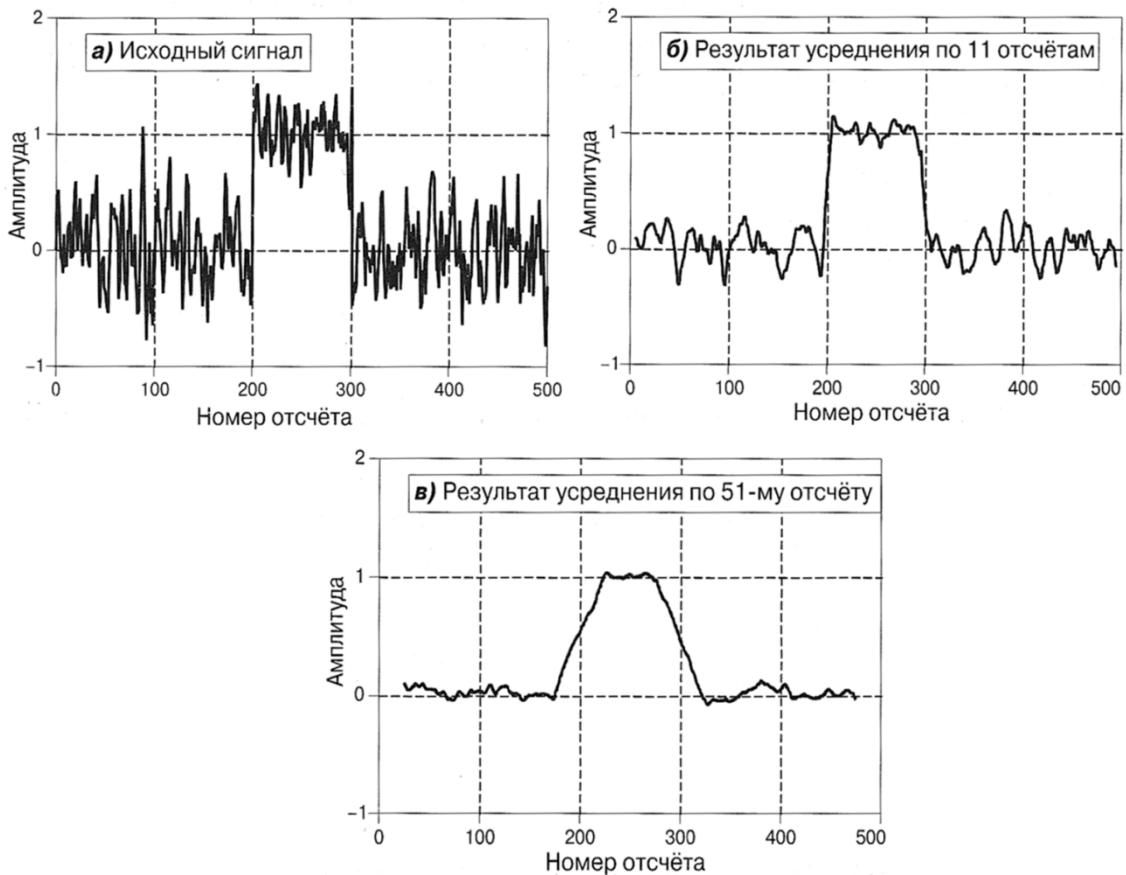


Рис. 1. Функционирование однородного фильтра.

Функционирование однородного фильтра иллюстрирует Рис. 1. На вход фильтра поступает сигнал, представляющий собой прямоугольный импульс, искажённый аддитивной шумовой помехой (а). В результате фильтрации (б и в) мощность шума заметно уменьшается (положительный эффект), но вместе с тем снижается крутизна фронтов импульса (отрицательный эффект). Среди всех линейных фильтров однородные фильтры характеризуются наиболее сильным по-

давлением шума при заданной крутизне фронтов импульса. Коэффициент подавления шума равен квадратному корню из числа отсчётов, участвующих в усреднении. Так, фильтр 100-го порядка позволяет уменьшить шум в 10 раз.

Частотная характеристика однородного фильтра

На Рис. 2 показана АЧХ однородного фильтра. Математически она выражается преобразованием Фурье от прямоугольного импульса:

$$H(f) = \frac{\sin(\pi f M)}{M \sin(\pi f)} \quad (2)$$

Судя по Рис. 2, спад АЧХ очень медленный, а затухание в зоне непрозрачности слишком далеко от идеала. Совершенно очевидно, что однородный фильтр не может быть использован для отделения одного частотного диапазона от другого. Не забывайте, что хорошее качество работы во временной области обычно сопровождается слабым разрешением в частотной области и наоборот. Проще говоря, однородный фильтр является самым лучшим *сглаживающим фильтром* (во временной области), но вместе с тем и самым худшим *НЧ-фильтром* (в частотной области).

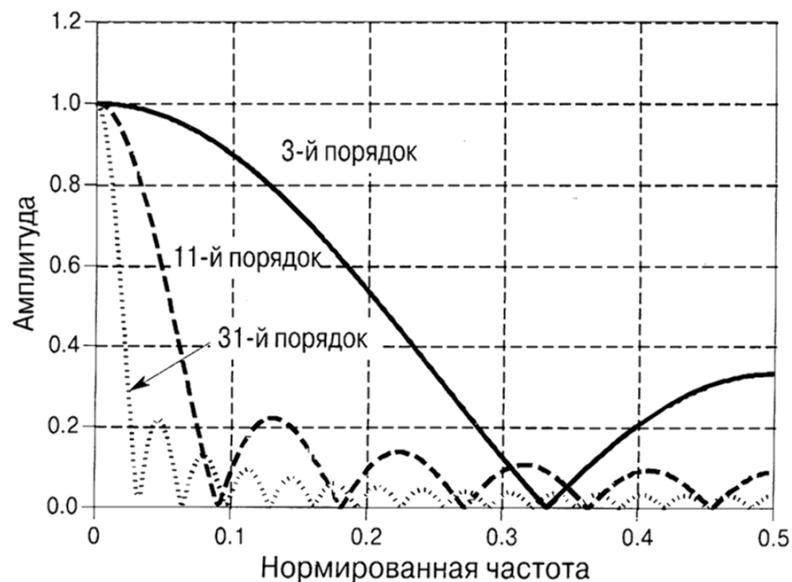


Рис. 2. АЧХ однородного фильтра. Однородный фильтр является очень слабым НЧ-фильтром: АЧХ спадает медленно, затухание в зоне непрозрачности очень мало. При построении графиков использовалось выражение (15.2).

Модифицированные однородные фильтры

В своей профессиональной деятельности разработчикам цифровых фильтров приходится иметь дело с информацией, заложенной либо во временной, либо в частотной области (*кодирование во временной и частотной областях*), и никогда информация не размещается в этих двух областях одновременно. Однако встречаются практические приложения, в которых сигнал все же необходимо рассматривать сразу в двух областях. Одним из таких «неприятных» примеров служит телевизионный сигнал. Видеоинформация кодируется во временной области, так как распределение яркости по элементам изображения закладывается в форму передаваемого сигнала. Однако в процессе передачи видеосигнала огромное значение имеет его частотная структура: диапазон занимаемых частот, разделение частотных диапазонов для изображений и звука, подавление и восстановление постоянной составляющей и т. п. Другим примером является учёт электромагнитных помех, которые удобнее анализировать в частотной, а не во временной области. Типовыми помехами, например, для обычных научных и инженерных измерений, являются: на частоте 50/60 Гц – помеха от сетей электропитания; на частоте 30 кГц – от импульсных источников питания; на частоте 1320 кГц – со стороны радиостанций FM-диапазона. *Модифицированные однородные фильтры* отличаются от обычных более чёткой частотной локализацией и поэтому могут более эффективно использоваться в подобных приложениях.

Фильтры с многократной обработкой представляют собой последовательное соединение нескольких однородных фильтров, т.е. одна и та же процедура однородной фильтрации повторяется в них последовательно несколько раз. На Рис. 3 показаны *импульсные характеристики*, построенные для фильтров с одной, двумя и четырьмя ступенями обработки (каждая ступень – однородный фильтр 7-го порядка). При наличии двух ступеней фильтр имеет треугольную импульсную характеристику, поскольку треугольная характеристика может быть представлена как результат свёртки двух прямоугольных характеристик. При увеличении количества ступеней до четырёх и более форма импульсной характеристики приближается к форме *кривой Гаусса* (согласно *центральной предельной теореме*). По *переходным характеристикам* (б) видно, что для фильтров с многократной обработкой характерен *s*-образный переходный процесс в отличие от однородного фильтра, которому соответствует линейно нарастающая реакция. Частотные характеристики, приведённые на рисунке (в и г), получены в результате многократного умножения АЧХ выражения (2) на саму себя. Умножение производится столько раз, сколько последовательных звеньев содержится в фильтре. Такой подход оказывается возможным в силу того, что каждой операции свёртки во временной области соответствует операция умножения в частотной области.

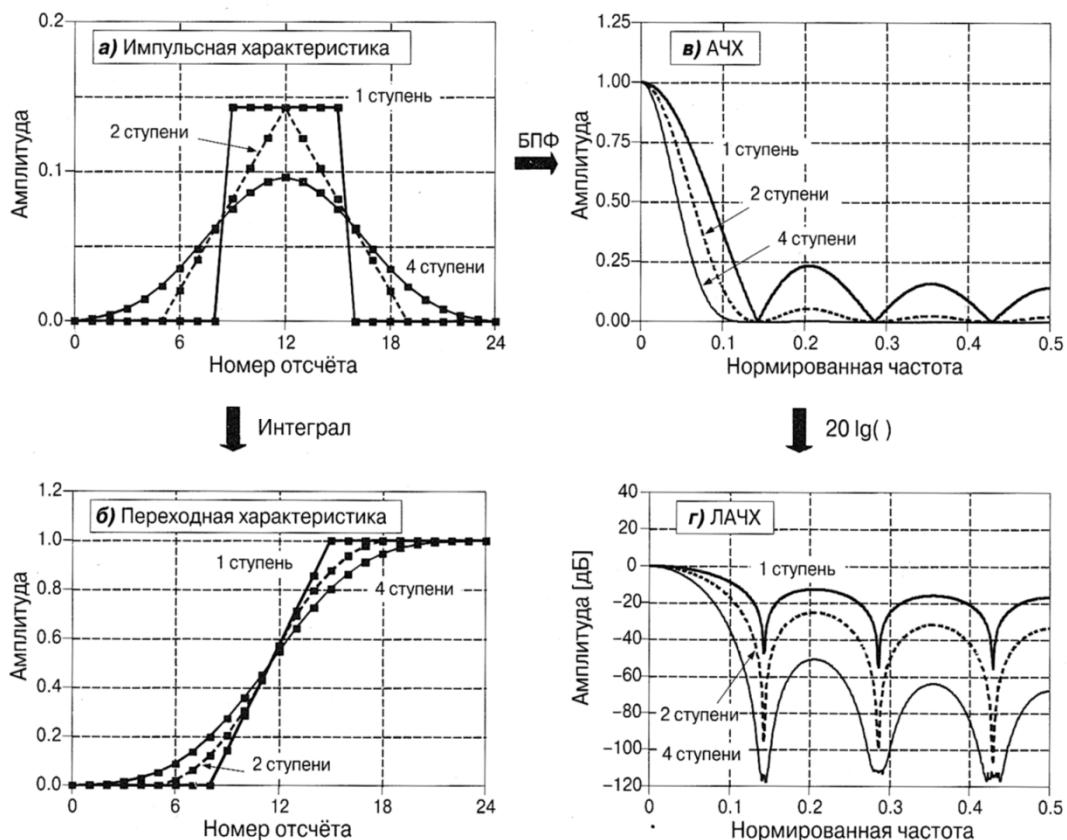


Рис. 15.3. Характеристики однородных фильтров с многократной обработкой.

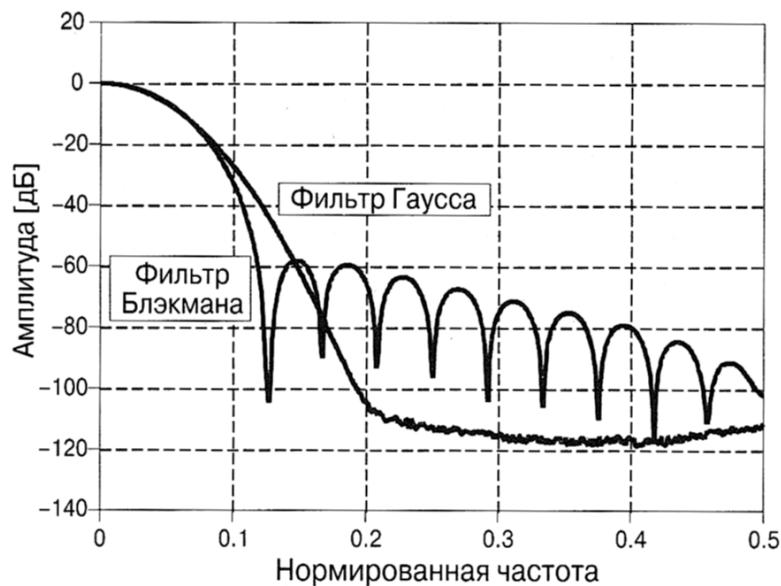


Рис. 4. ЛАЧХ фильтров Блэкмана и Гаусса.

На Рис. 4 изображены ЛАЧХ для ещё двух модификаций однородных фильтров. Если импульсная характеристика фильтра имеет вид кривой Гаусса, то и его АЧХ является также гауссовой. Импульсные характеристики систем как

естественного, так и искусственного происхождения очень часто описываются кривой Гаусса.

Какие же преимущества характерны для рассмотренных выше модификаций однородных фильтров? Основных преимуществ – три. Во-первых, модифицированные однородные фильтры позволяют достичь большего подавления в зоне непрозрачности, чем обычные однородные фильтры. Во-вторых, их импульсная характеристика на краях плавно сходится к нулю. Как вы помните, каждый отсчёт выходного сигнала определяется взвешенной суммой группы входных отсчётов. Поэтому сходимость к нулю импульсной характеристики означает, что отсчёты, близкие к центру такой группы, оказывают более сильное влияние на формирование выходного отсчёта. В-третьих, переходные процессы модифицированных фильтров оказываются более плавными и не имеют резких изгибов. Последние два преимущества менее важны, хотя, надо заметить, существует ряд приложений, в которых они оказываются очень полезными.

Модифицированные однородные фильтры и обычные однородные фильтры проявляют приблизительно равные возможности при решении задачи подавления шума при одновременном сохранении скорости нарастания переходной характеристики. Точное решение вопроса, какой из фильтров лучше, связано с тем, каким образом мы измеряем длительность *переходного процесса*. Если считать, что переходный процесс – это время нарастания переходной характеристики от 0 до 100% её установившегося значения, то преимущество оказывается на стороне обычного однородного фильтра, как было показано выше. Если длительность переходного процесса измерять интервалом времени, в течение которого переходная характеристика нарастает от 10 до 90% своей установившейся величины, то фильтр Блэкмана несомненно превосходит однородный фильтр. Так что спор по этому вопросу носит исключительно теоретический характер.

Наиболее заметным является различие в объёмах *вычислительных затрат*. Рекурсивные однородные фильтры, речь о которых пойдет далее, могут с лёгкостью быть реализованы на обычном персональном компьютере. Это самые экономные по вычислительным затратам фильтры. Затраты на реализацию фильтров с многократной обработкой несколько больше, но всё же невелики. А вот фильтры Гаусса и Блэкмана требуют очень большого объёма вычислений, так как основаны на операции свёртки.

Однородные рекурсивные фильтры

Однородные фильтры имеют одно важное преимущество перед всеми другими: для них существует самый простой алгоритм реализации, какой только можно придумать. Чтобы понять этот алгоритм, рассмотрим следующий пример.

Пусть сигнал x поступает на вход однородного нерекурсивного фильтра пятого порядка, y – реакция этого фильтра. Тогда отсчёты $y[10]$ и $y[11]$ выражаются соотношениями

$$y[10] = \frac{x[8] + x[9] + x[10] + x[11] + x[12]}{5}$$

и

$$y[11] = \frac{x[9] + x[10] + x[11] + x[12] + x[13]}{5}.$$

Эти выражения во многом повторяют друг друга, поскольку отсчёты 9...12 используются для вычисления и $y[10]$, и $y[11]$. Если $y[10]$ уже найден, то существует более эффективный способ вычислить $y[11]$:

$$y[11] = y[10] - \frac{x[8]}{5} + \frac{x[13]}{5}$$

Если найдено значение $y[10]$, то $y[11]$ проще всего выразить через него. То же самое относится и к последующим отсчётам выходной последовательности. Если найден один отсчёт выходного сигнала каждый последующий может быть найден в результате одной операции сложения и одной операции вычитания:

Как видим, этот алгоритм использует два источника данных: отсчёты входной последовательности и найденные ранее отсчёты выходной последовательности. Алгоритм, в котором найденные на предшествующих итерациях отсчёты выходной последовательности также используются для вычисления последующих отсчётов выходного сигнала, называется *рекурсивным*. Однако однородные рекурсивные фильтры существенно отличаются от остальных рекурсивных фильтров. В частности, *импульсная характеристика* однородных фильтров имеет вид прямоугольного импульса, т. е. это *фильтры с конечной импульсной характеристикой*, или *КИХ-фильтры*, в то время как *рекурсивные фильтры* образуют класс *БИХ-фильтров*, а их импульсная характеристика имеет бесконечную длину и представляется набором синусоид и экспонент.

III. ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить основные теоретические сведения о однородных фильтрах.
2. Реализовать нерекурсивный фильтр скользящего среднего по 5-ого, 10-ого и 50-ого порядка в системе Matlab или Octave.
4. Проверить работу фильтра на тестовом сигнале.
5. Оценить быстродействие и эффективность фильтра.

6. Реализовать модифицированный вариант однородного фильтра.
7. Реализовать рекурсивный вариант фильтра скользящего среднего.
8. Сравнить различные варианты реализации однородных фильтров.
9. Сделать выводы.

IV. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- алгоритмы и листинги программ реализации фильтров;
- графики, полученные в результате численного моделирования работы синтезированных фильтров;
- выводы по проведённой работе.

V. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит; пер. с англ. А.Ю. Линовича, С.В. Витязева, И.С. Гусинского. – М.: Додэка-XXI, 2012. – 720 с.
2. Умняшкин С.В. Истории цифровой обработки сигналов: Учебное пособие // М.: ТЕХНОСФЕРА, 2016. – 528 с.
3. Лайонс Р. Цифровая обработка сигналов. Пер. С англ. – М.: ООО «Бином-Пресс», 2015. – 656 с.
4. Основы цифровой обработки сигналов: Курс лекций / Авторы: А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б. Соловьёва, И.И. Гук. – Спб.: БХВ-Петербург, 2003. – 608 с.
5. Гадзиковский В.И. Цифровая обработка сигналов. – М.: СОЛОН-ПРЕСС, 2015. – 766 с.

Практическое занятие № 7

Рекурсивные фильтры

I. ЦЕЛЬ И ЗАДАЧИ

Синтез и исследование однополюсных рекурсивных фильтров.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Рекурсивные фильтры, в отличие от нерекурсивных, не используют операцию *свёртки* временных последовательностей, требующую значительных вычислительных затрат. Поэтому их применение повышает вычислительную эффективность алгоритмов фильтрации, но при этом могут возникать значительные ограничения по реально достижимой точности воспроизведения желаемых частотных характеристик. Рекурсивные фильтры относятся к классу цепей с *бесконечной импульсной характеристикой (БИХ)*. Импульсная характеристика БИХ-фильтров складывается из множества синусоид, амплитуда которых убывает по экспоненциальному закону. Форма импульсной характеристики является основным отличием рекурсивных фильтров от фильтров, построенных на основе свёртки, которые образуют, в свою очередь, класс цепей с *конечной импульсной характеристикой (КИХ)*.

Предположим, вам необходимо выделить информационную составляющую, содержащуюся в сигнале x .

Самым очевидным источником данных являются отсчеты входной последовательности: $x[n], x[n-1], x[n-2], \dots$ Именно так и происходит при расчете однородных фильтров, но можно пользоваться и другим источником данных – предыдущими отсчётами выходного сигнала: $y[n-1], y[n-2], y[n-3], \dots$ В результате учёта дополнительной информации алгоритм приобретает новый вид:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \dots + b_1y[n-1] + b_2y[n-2] + b_3y[n-3] + \dots$$

В общем случае рекурсивные фильтры во временной области описываются линейными разностными уравнениями вида:

$$y[n] = \sum_{i=0}^{N-1} b_i x[n-i] - \sum_{k=1}^{M-1} a_k y[n-k],$$

где $x[n]$ – отсчеты входного воздействия; $y[n]$ – отсчеты реакции; $\{b_i, a_k\}$ – вещественные коэффициенты, полностью определяющие свойства ЦФ; M и N – константы, задающие сложность (порядок) ЦФ; $x[n-i]$ и $y[n-k]$ –

отсчеты воздействия и реакции, задержанные на i и k периодов дискретизации T соответственно.

Теперь значение каждого очередного отсчёта выходной последовательности вычисляется путём умножения отсчётов входного сигнала на коэффициенты a_i и умножения полученных на предыдущих итерациях отсчётов выходного сигнала на коэффициенты b_i с последующим суммированием всех произведений. Заметьте, что коэффициент b_0 не входит в выражение, так как соответствует текущему отсчёту выходного сигнала. Такие фильтры называют рекурсивными фильтрами. Коэффициенты a_i и b_i являются его весовыми коэффициентами. Порядок рекурсивных фильтров, как правило, не превышает 12, что связано с проблемой устойчивости (на выходе фильтра могут возникнуть неконтролируемый рост сигнала или свободные колебания).

Главное достоинство рекурсивных фильтров заключается в том, что при их реализации удаётся избежать использования операции свёртки, требующей большого количества арифметических операций. Для примера подадим на вход рекурсивного фильтра дискретную дельта-функцию (единичный импульс). Реакцией на такое воздействие будет импульсная характеристика фильтра, которая представляет собой синусоидальные колебания с убывающей по экспоненциальному закону амплитудой. Поскольку такая импульсная характеристика бесконечна по времени, рекурсивные фильтры называют фильтрами с бесконечной импульсной характеристикой (БИХ). В действительности происходит свёртка входного сигнала с бесконечной импульсной характеристикой при конечном числе весовых коэффициентов.

Взаимосвязь между импульсной характеристикой фильтра и его весовыми коэффициентами определяется при помощи Z-преобразования. Z-преобразование используется для решения следующих задач: определение частотной характеристики по весовым коэффициентам рекурсивного фильтра, объединение последовательных и параллельных звеньев в единый фильтр, проектирование рекурсивных систем, повторяющих по своим свойствам аналоговые фильтры, и т. д. К сожалению, Z-преобразование требует специальных знаний из области высшей математики и оказывается для многих слишком сложным. Данное преобразование создано для специалистов в области ЦОС и цифровых систем.

Однополюсный рекурсивный фильтр

Пример однополюсного НЧ-фильтра показан на рис. 2. Этот рекурсивный фильтр имеет всего два весовых коэффициента, принимающих

значения $a_0 = 0,15$ и $b_1 = 0,85$. Подадим ступенчатое входное воздействие. Как и следовало ожидать, выходная реакция НЧ-фильтра плавно нарастает до некоторого установившегося значения.

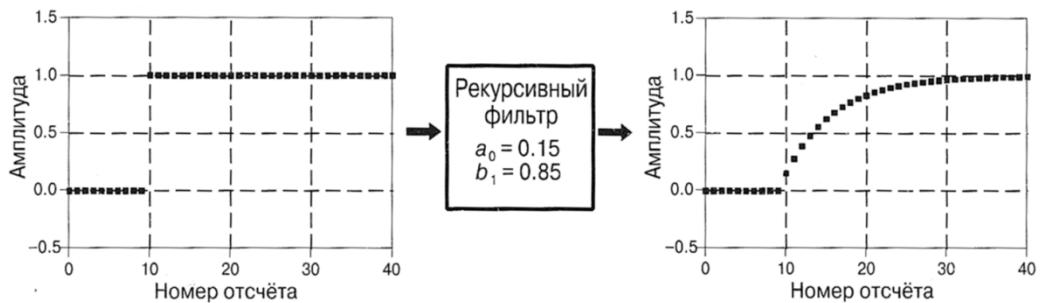


Рис. 2. Однополюсный НЧ-фильтр.

Достоинство рекурсивного метода состоит в том, что, меняя всего несколько параметров, можно создавать самые разные импульсные характеристики. Фильтр, характеристики которого приведены на рис. 3, имеет три коэффициента: $a_0 = 0,93$, $a_1 = -0,93$ и $b_1 = 0,86$.

Разумеется, такие фильтры вам очень хотелось бы иметь в наборе доступных инструментов ЦОС. Их можно использовать при обработке цифровых сигналов так же, как RC -цепи используются в аналоговой обработке. Решаемые задачи: устранение постоянной составляющей, подавление высокочастотного шума, формирование огибающей, сглаживание и т.д.

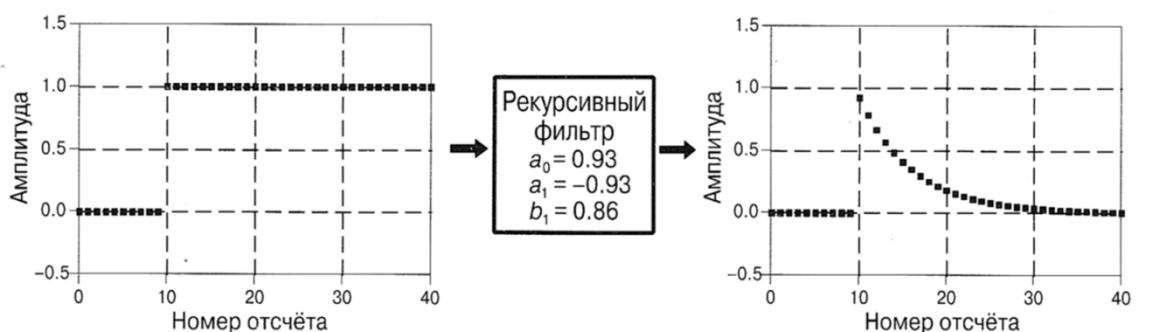


Рис. 3. Однополюсный ВЧ-фильтр.

Для таких фильтров легко написать программу, они не требуют больших вычислительных затрат и в процессе использования практически не преподносят неприятных сюрпризов. Для расчёта весовых коэффициентов применяются следующие простые уравнения:

$$a_0 = 1 - x,$$

$$b_1 = x. \quad (2)$$

Однополюсный НЧ-фильтр. Форма импульсной характеристики определяется параметром $x = 0...1$.

$$\begin{aligned} a_0 &= (1 + x)/2, \\ a_1 &= -(1 + x)/2, \\ b_1 &= x. \end{aligned} \quad (3)$$

Характеристики таких фильтров определяются параметром $x = 0...1$. Физический смысл параметра x – относительное затухание на интервале между соседними отсчётами. Например, у фильтра на рис. 19.3 $x = 0,86$, вследствие чего каждый очередной отсчёт импульсной характеристики равен предыдущему, умноженному на 0,86. Чем больше x , тем медленнее спад. Если x больше единицы, фильтр оказывается неустойчивым. В этом случае появление любого ненулевого отсчёта на входе фильтра приведёт к переполнению (появлению слишком больших значений) на выходе.

Параметр x может быть задан непосредственным образом, а также может быть выражен через постоянную времени фильтра. Величина $R \cdot C$ имеет размерность времени, и в простейших RC-цепях она соответствует интервалу времени, через который отклонение амплитуды выходного сигнала от установившегося значения сокращается в ходе переходного процесса до 36,8% от первоначальной величины. Если d – постоянная времени, измеряемая числом отсчётов дискретного сигнала, то имеет место следующее равенство:

$$x = e^{-1/d} \quad (19.4)$$

Относительному затуханию $x = 0,86$ соответствует постоянная времени $d = 6.63$ дискретных отсчётов (рис. 19.3). Кроме того, между x и частотой среза, определяемой на уровне – 3 дБ, существует взаимосвязь:

$$x = e^{-2\pi f_c} \quad (19.5)$$

Из сказанного следует, что коэффициенты a_i и b_i могут быть выражены через следующие три параметра: постоянную времени, частоту среза и непосредственно через относительное затухание x .

Пример работы рекурсивных однополюсных фильтров иллюстрируется на рис. 4. Входной сигнал (а) представляет собой гладкую кривую, на которую на протяжении короткого интервала времени накладывается всплеск более высокой частоты. С помощью НЧ- и ВЧ-фильтров можно разделить

компоненты входного сигнала (б), но всё же качество фильтрации оказывается низким, как и при использовании аналоговых RC-цепей.

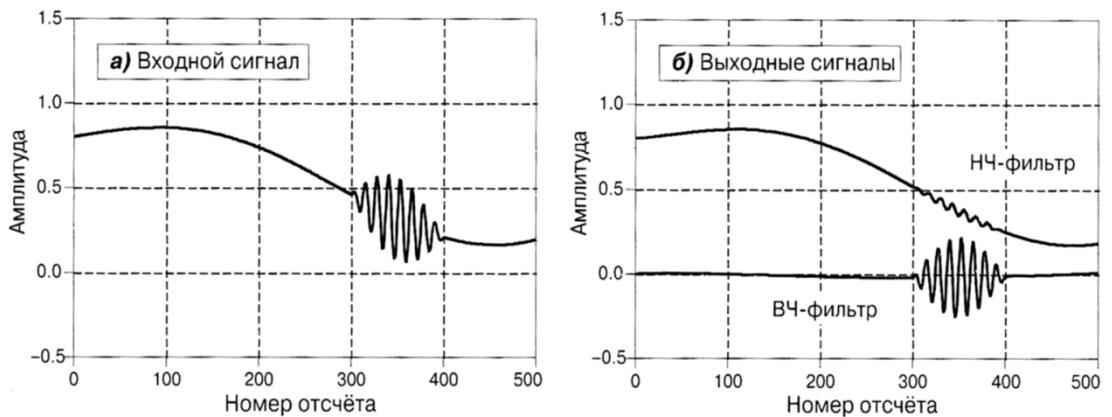


Рис. 4. Пример работы однополюсных рекурсивных фильтров.

На рис. 5 показаны АЧХ однополюсных рекурсивных фильтров. Чтобы получить частотную характеристику рекурсивного фильтра, сначала находят его импульсную характеристику как реакцию на единичное импульсное воздействие. Затем применяют БПФ, чтобы перейти от импульсной характеристики к частотной. Теоретически импульсная характеристика имеет бесконечную длину, но в реальности через интервал, в 15...20 раз превышающий постоянную времени, она затухает ниже уровня шумов округления. Например, для $d = 6.63$ отсчёта в импульсной характеристике значащими остаются только около 128 отсчётов.

На рис. 5 хорошо заметна важная особенность однополюсных рекурсивных фильтров, заключающаяся в слабых возможностях разделения частотных диапазонов. Эти фильтры отлично проявляют себя при обработке сигналов во временной области, но оказываются недостаточно мощными в задачах частотной селекции. Некоторого улучшения частотных характеристик позволяет достичь использование последовательного (каскадного) соединения нескольких однополюсных фильтров. Здесь возможны два способа. Во-первых, сигнал можно пропускать через один и тот же фильтр несколько раз. Во-вторых, воспользовавшись Z-преобразованием, можно найти весовые коэффициенты цифрового рекурсивного фильтра, эквивалентного по своим свойствам последовательному соединению нескольких звеньев. Оба метода хорошо работают и находят широкое применение. На рис. 5в показаны АЧХ 4-каскадных рекурсивных НЧ-фильтров. Хотя затухание в зоне подавления при переходе к каскадной форме построения значительно увеличивается, переходная зона по-прежнему оставляет желать лучшего. Если вам

необходимо повысить качество в частотной области, ориентируйтесь на использование фильтров Чебышева.

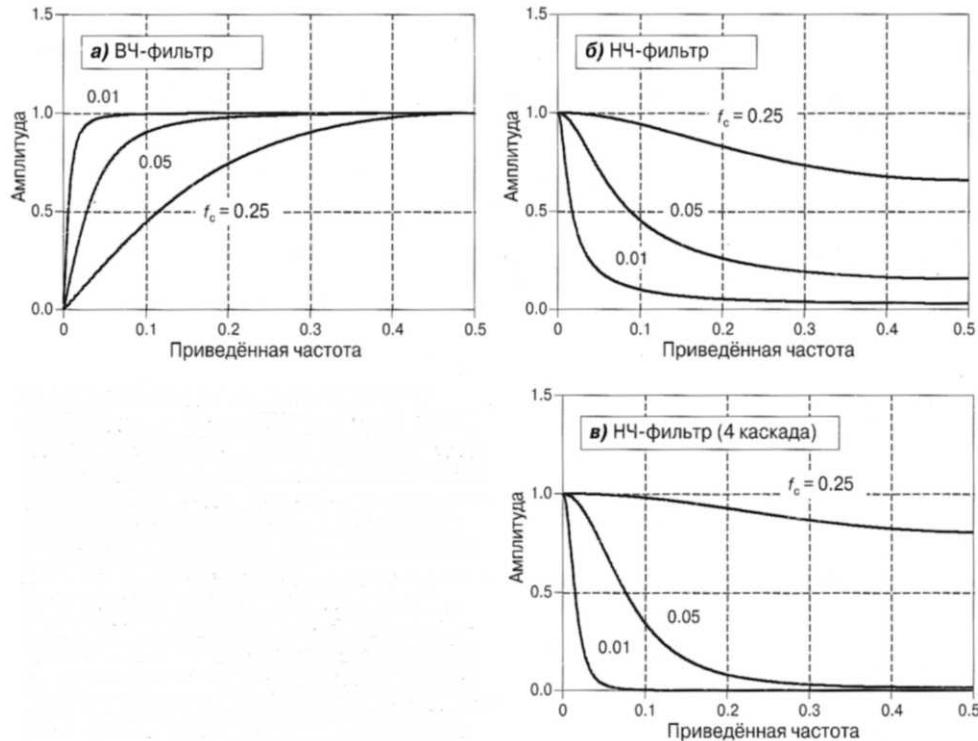


Рис. 5. АЧХ однополюсных фильтров.

Многокаскадный НЧ-фильтр сопоставим с фильтрами Блэкмана и Гаусса, являющимися модификациями однородных КИХ-фильтров, но при этом он требует существенно меньших вычислительных затрат. Для расчёта 4-каскадного НЧ-фильтра можно воспользоваться следующими уравнениями:

$$\begin{aligned} a_0 &= (1-x)^4, \\ b_1 &= 4x, \\ b_2 &= -6x^2, \\ b_3 &= 4x^3, \\ b_4 &= -x^4. \end{aligned} \quad (6)$$

Сравним однородный фильтр с усреднением по 9 отсчётам и однополюсный рекурсивный фильтр, построенный по двунаправленной технологии. Для получения сопоставимой частотной характеристики коэффициент однополюсного фильтра $x = 0,7$. АЧХ фильтров представлены на рис. 6. Частотные характеристики не отличаются высоким качеством, что вполне естественно для фильтров, работающих во временной области.

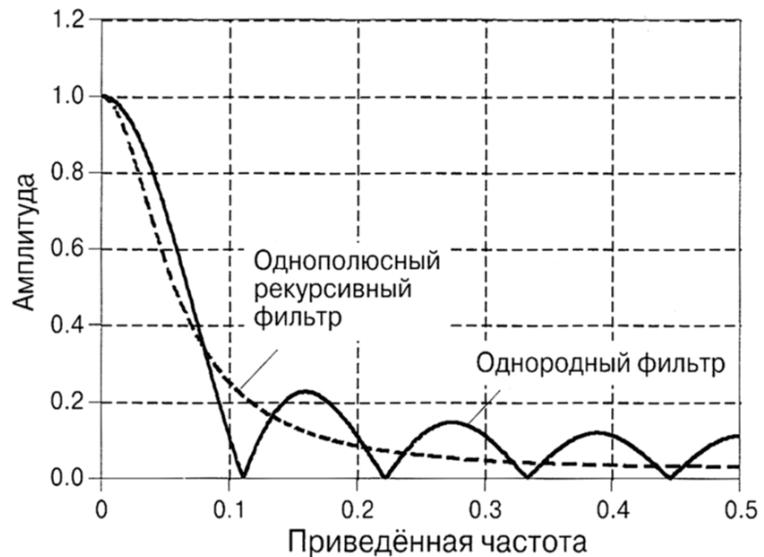


Рис. 6. АЧХ однородного и однополюсного рекурсивных фильтров.

Переходные характеристики фильтров изображены на рис. 7. Переходный процесс однородного фильтра (а) имеет вид отрезка прямой, что соответствует наиболее быстрому переходу с одного уровня на другой. Переходная характеристика рекурсивного фильтра (б) имеет гладкий переходный процесс, что в некоторых случаях обеспечивает преимущество. Таким образом, преимуществом однородного фильтра является более короткий переходный процесс; преимуществом однополюсного фильтра – сглаженный переходный процесс.

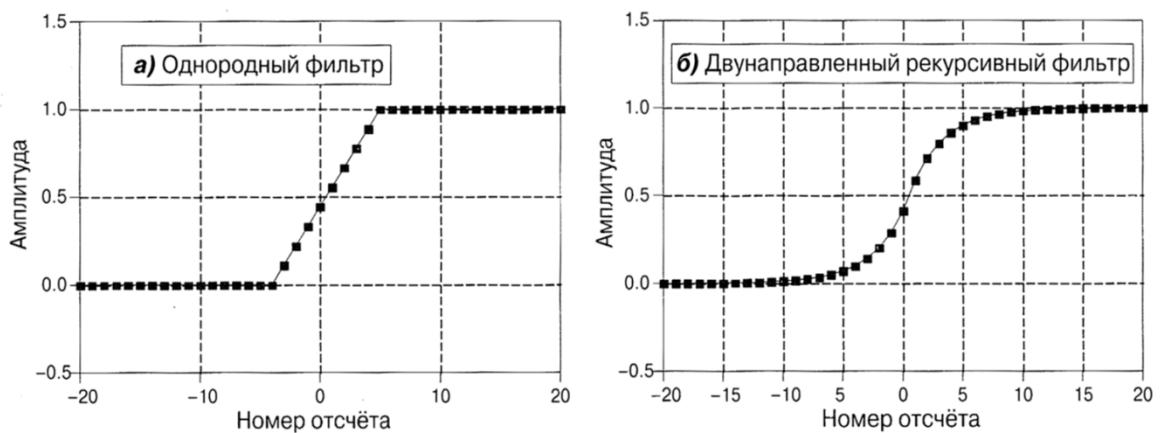


Рис. 7. Переходные характеристики однородного фильтра и двунаправленного однополюсного фильтра.

Фильтры мало отличаются по качеству работы, поэтому выбор часто зависит только от предпочтений разработчика. Тем не менее некоторое преимущество обнаруживается при анализе двух характеристик – времени разработки фильтра и времени обработки отсчётов входного сигнала. В

первом случае требуется сократить время разработки фильтра, пренебрегая увеличением вычислительных затрат. Например, необходимо всего один раз воспользоваться фильтром, порядок которого равен нескольким тысячам. Поскольку вся программа фильтрации выполняется всего за несколько секунд, бессмысленно тратить время на оптимизацию алгоритма. Ясно, что предпочтение следует отдать вычислениям в формате с плавающей точкой. Остаётся выбрать из двух вариантов: либо однородный фильтр, реализуемый на базе свёртки, либо однополосный рекурсивный фильтр. Здесь предпочтительнее рекурсивный фильтр, т.к. его проще программировать и настраивать, а его вычислительные затраты намного меньше.

Во втором случае всё наоборот: необходимо минимизировать вычислительные затраты, а времени на разработку достаточно. Например, фильтр может входить в состав серийно выпускаемого изделия, где будет использован миллионы раз. Для ускорения обработки, вероятно, нужно будет использовать целые числа. Выбор здесь производится между однородным фильтром на основе рекурсивного алгоритма и однополосным рекурсивным фильтром, использующим справочную таблицу (таблицу соответствия) или целочисленную математику. Победителем оказывается однородный фильтр. Он работает быстрее и не создаёт проблем, связанных с целочисленной математикой, на стадиях расчёта и обработки.

III. ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить основные теоретические сведения о рекурсивных фильтрах.
2. Рассчитать параметры НЧ, ВЧ и полосных рекурсивных фильтров по заданным преподавателем исходным данным.
3. Реализовать соответствующие фильтры в системе Matlab или Octave.
4. Провести численное моделирование работы фильтров.
5. Оценить быстродействие фильтров. Сделать выводы.

IV. ОФОРМЛЕНИЕ ОТЧЕТА

Отчёт должен содержать:

- алгоритмы и листинги программ реализации фильтров;
- графики, полученные в результате численного моделирования работы синтезированных фильтров;
- выводы по проведённой работе.

V. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит; пер. с

англ. А.Ю. Линовича, С.В. Витязева, И.С. Гусинского. – М.: Додэка-XXI, 2012. – 720 с.

2. Умняшкин С.В. Иновы цифровой обработки сигналов: Учебное пособие // М.: ТЕХНОСФЕРА, 2016. – 528 с.

3. Лайонс Р. Цифровая обработка сигналов. Пер. С англ. – М.: ООО «Бином-Пресс», 2015. – 656 с.

4. Основы цифровой обработки сигналов: Курс лекций / Авторы: А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б. Соловьёва, И.И. Гук. – Спб.: БХВ-Петербург, 2003. – 608 с.

5. Гадзиковский В.И. Цифровая обработка сигналов. – М.: СОЛОН-ПРЕСС, 2015. – 766 с.

Практическое занятие № 8

СИНТЕЗ ЦИФРОВОГО КОРРЕКТИРУЮЩЕГО УСТРОЙСТВА

I. ЦЕЛЬ И ЗАДАЧИ

1. Синтез Z-передаточной функции корректирующего устройства с помощью пакета Matlab.
2. Программирование Z-передаточной функции.
3. Получение основных характеристик скорректированного ЦЭСП.

II. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Классический способом синтеза является синтез методом частотных характеристик. Однако в системах с цифровым вычислителем непосредственно использовать этот простой метод не удается. Один из путей решения этой проблемы – использование Z-преобразования, далее выполнение билинейного преобразования и переход к псевдочастотным характеристикам, после чего получение передаточной функции фильтра и обратный переход к Z-передаточным функциям. Безусловно, этот путь является правильным, но, в то же время, достаточно трудоемким. Однако, если принять некоторые допущения, то можно значительно упростить эту процедуру.

На рисунке 1. представлена структурная схема привода с использованием аналогового фильтра. Передаточную функцию фильтра получим, используя исходную и желаемую ЛАЧХ разомкнутой системы.

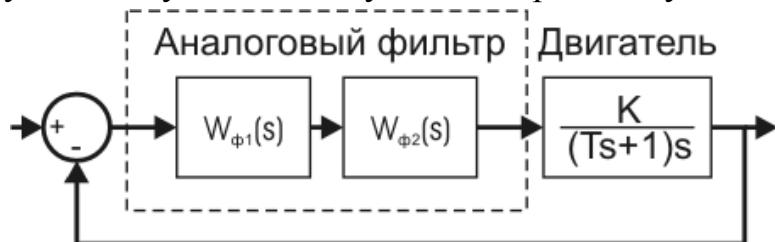


Рисунок 1. Упрощенная модель привода с аналоговым фильтром.

Исходная ЛАЧХ – это ЛАЧХ двигателя, который представлен апериодическим звеном и интегратором. На рисунке 2 представлена ЛАЧХ исходной (сплошная) и желаемой (штриховая) системы, а также показаны частоты всех изломов ($\omega_1, \omega_2 \dots$). Передаточная функция фильтра получается путем вычитания исходной ЛАЧХ из желаемой.

После того как получена передаточная функция фильтра, производим Z-преобразование и получим Z-функцию фильтра. Однако надо помнить об одном ограничении: тик квантования в системе должен быть хотя бы в 2

раза меньше самой маленькой постоянной времени системы. На рисунке 3 представлена эквивалентная схема системы, но уже с цифровым фильтром.

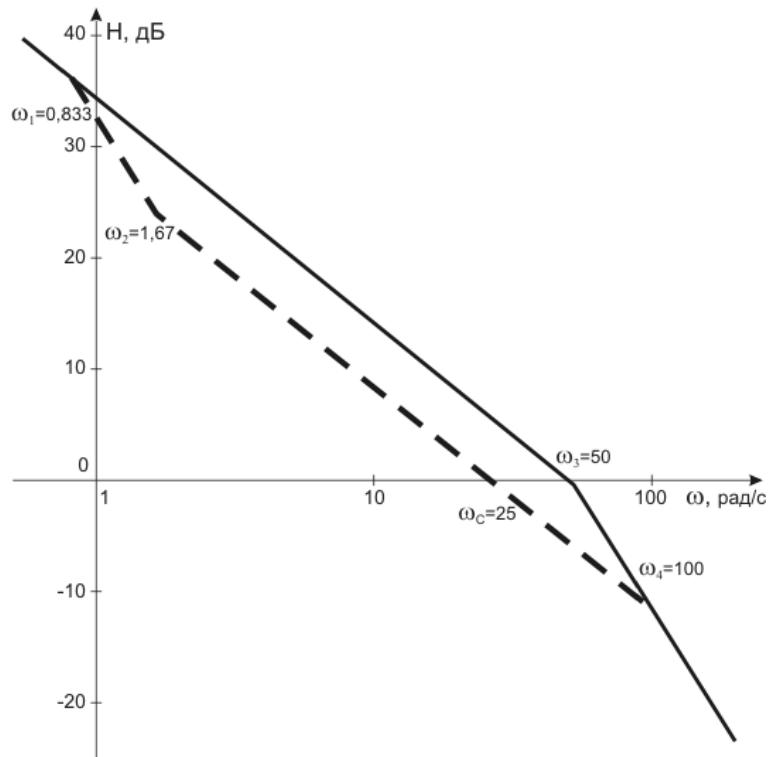


Рисунок 2. ЛАЧХ исходной (сплошная) и желаемой (штриховая) системы.

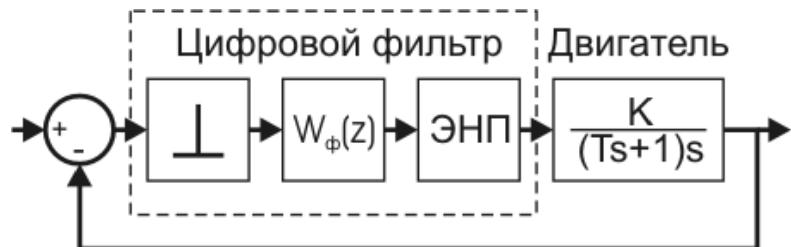


Рисунок 3. Упрощенная модель привода с цифровым фильтром.

Далее необходимо запрограммировать Z-функцию фильтра. Существует несколько способов выбора переменных состояния; будем использовать прямое, параллельное или последовательное программирование.

В результате получаем структурную схему, вводим переменные состояния и записываем в виде системы разностных уравнений. Далее необходимо выписать главную матрицу A, матрицу-столбец B, матрицу-строку C и матрицу D. Полученные матрицы вводятся в программу Scope.exe (смотри рисунок 4).



Рисунок 4. Окно ввода коэффициента передачи и матриц фильтра.

Приборы и оборудование

- Персональный компьютер с установленной системой Matlab;
- Приложение Scope.exe;
- Цифровой следящий привод.

III. ПОРЯДОК ВЫПОЛНЕНИЯ.

- Передаточная функция двигателя имеет вид:

$$W_{\ddot{a}d}(s) = \frac{7.14}{(19.2 \cdot 10^{-3} s + 1)s}$$

По рисунку 2 необходимо определить величину коэффициента передачи по контуру А затем получить коэффициент передачи фильтра К.

- После этого открываем модель Simulink и вводим полученный коэффициент передачи (смотри рисунок 5). Сохраняем переходный процесс в отчет.

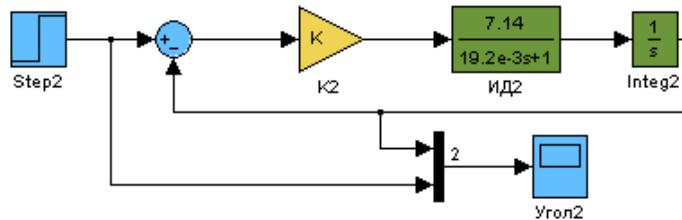


Рисунок 5. Упрощенная модель с П-регулятором.

- Необходимо получить передаточную функцию фильтра вычитанием исходной ЛАЧХ из желаемой (смотри рисунок 2). В результате должна получиться передаточная функция вида:

$$W(s) = \frac{(T2s + 1)(T3s + 1)}{(T1s + 1)(T4s + 1)} \quad (1)$$

- Ввести полученный аналоговый фильтр в модель Simulink (смотри рисунок 6). Сохранить переходный процесс в отчет.

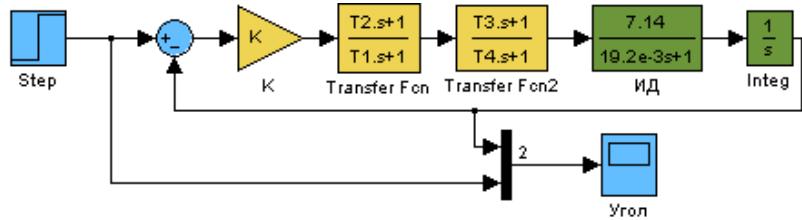


Рисунок 6. Модель с аналоговым фильтром.

- В командной строке Matlab создать переменную s – оператор дифференцирования. Для этого необходимо воспользоваться функцией tf (transfer function - передаточная функция): $s = \text{tf}('s')$
- Далее вводим передаточную функцию в виде: $W_s = (T2*s+1)*(T1*s+1)/(T3*s+1)*(T4*s+1)$
- где $T1, T2 \dots T4$ – конкретные числа из формулы (1).
- Далее производим Z-преобразование с помощью функции c2d (continues to discrete): $W_z = \text{c2d}(W_s, T)$

где W_s – введенная ранее переменная; T – такт квантования, с. Для совместимости со стендом примем $T = 5$ мс.

- Вводим полученный цифровой фильтр в модель Simulink (смотри рисунок 7). Сохраняем переходный процесс в отчет.

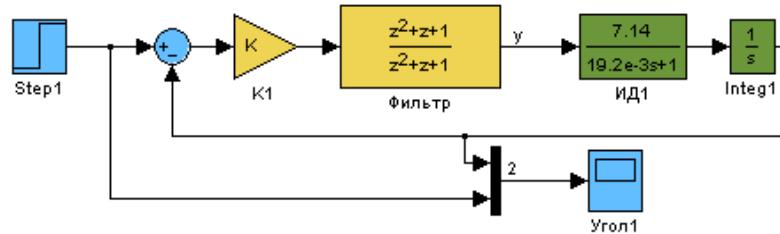


Рисунок 7. Модель с цифровым фильтром.

- После того как получена верная Z-передаточная функция, необходимо запрограммировать её одним из указанных преподавателем способов (прямое, параллельное или последовательное программирование). Записать в отчет полученную структурную схему.
- Записать систему разностных уравнений и выписать главную матрицу А, матрицу-столбец В, матрицу-строку С и матрицу D.
- Включить тумблер +27 В на панели блока «Усилитель, генератор, демодулятор».
- Вызвать окно «Настройки», выбрать закладку «Сигнал» и установить такт квантования 5 мс, амплитуду сигнала 50 мрад и 120 мрад. На закладке «Регулятор» выбрать тип регулятора «Цифровой фильтр», установить

полученный коэф. передачи и ввести матрицы A, B, C и D. Зафиксировать изменения и запустить управление. Занести результаты в отчет.

- Вызвать окно «Настройки», выбрать закладку «Сигнал» и установить амплитуду сигнала 50 мрад и 120 мрад. На закладке «Регулятор» выбрать тип регулятора «П-регулятор». Зафиксировать изменения и запустить управление. Занести результаты в отчет.
- Выключить двигатель. Закрыть программу. Выключить блок питания привода.

IV. СОДЕРЖАНИЕ ОТЧЕТА

- Название и номер работы, цель работы;
- ЛАЧХ исходной и скорректированной системы, а также ЛАЧХ фильтра.
- Коэффициент передачи фильтра, передаточная функция аналогового фильтра, Z-передаточная функция цифрового фильтра,
- Структурная схема Z-передаточной функции цифрового фильтра
- Система разностных уравнений и матрицы A, B, C, D.
- Все переходные процессы с показателями качества, указанные в порядке выполнения работы;
- Выводы.

V. КОНТРОЛЬНЫЕ ВОПРОСЫ

- Как по ЛАЧХ определить коэффициент передачи по контуру?
- Какой наклон должна иметь ЛАЧХ в районе частоты среза?
- Какой способ программирования наиболее выгоден? Почему?
- Почему матрица B в данной системе является матрицей столбцом, а не прямоугольной матрицей? Что такое матрица D?

VI. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Горячев О.В. Компьютерное управление мехатронными системами. Часть 2: учеб.пособие (практикум). Тула: Изд-во ТулГУ, 2014.152с.
2. "Буран". Основы проектирования интеллектуальной системы управления орбитальным кораблем на атмосферном участке полета: монография / В. П. Андреев [и др.] ; под ред. А. С. Сырова; Федер. космич. агентство, Моск. опытно-конструктор. бюро «Марс» .— Москва : Марс, 2013 .— 276 с. : ил. — Дар каф. САУ ТулГУ ТулГУ : 1348233-1348246 .— Библиогр. в конце кн. — ISBN 978-5-9903829-3-0 (в пер.) .
3. Макаров Н.Н., Семашкин В.Е. Численные методы в анализе и синтезе систем управления: учеб. Пособие. Тула: Изд-во ТулГУ, 2014. 121с.
4. Дьяконов В. MATLAB: учебный курс. – СПб: Питер, 2001. – 560 с.: ил.
5. Цифровая обработка сигналов / А.Б. Сергиенко – СПб.: Питер, 2003. – 604 с.: ил.
6. Руководство по системе Matlab.
7. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит; пер. с англ. А.Ю. Линовича, С.В. Витязева, И.С, Гусинского. – М.: Додэка-XXI, 2012. – 720 с.
8. Умняшкин С.В. Иосновы цифровой обработки сигналов: Учебное пособие // М.: ТЕХНОСФЕРА, 2016. – 528 с.
9. Лайонс Р. Цифровая обработка сигналов. Пер. С англ. – М.: ООО «Бином-Пресс», 2015. – 656 с.
10. Основы цифровой обработки сигналов: Курс лекций / Авторы: А.И. Солонина, Д.А, Улахович, С.М. Арбузов, Е.Б. Соловьёва, И.И. Гук. – Спб.: БХВ-Петербург, 2003. – 608 с.
11. Гадзиковский В.И. Цифровая обработка сигналов. – М.: СОЛОН-ПРЕСС, 2015. – 766 с.