

С.С. Борисов, А.А. Шаронов, И.И. Кузнецов

Пермский национальный исследовательский
политехнический университет

МОДЕЛИРОВАНИЕ ОСНОВНЫХ ПРОЦЕССОВ ЦИФРОВОЙ ОБРАБОТКИ ИНФОРМАЦИИ

Приводится краткое описание теории цифровой обработки сигналов (ЦОС). Модель последовательности цифровой обработки сигнала, а также пример работы программы «Basic DSP» версии 1.0 на языке Delphi в среде Borland Delphi.

В современном мире весьма широко используется теория цифровой обработки сигналов (ЦОС). Важнейшими алгоритмами данной теории являются быстрое преобразование Фурье (чаще всего с основанием 2) и цифровая фильтрация.

Алгоритмы БПФ с основанием 2. Алгоритмы БПФ хорошо известны и широко используются, но при первом ознакомлении с ними они по ряду причин достаточно трудны для понимания. В литературе [1, 3–6], встречается несколько различных подходов к описанию алгоритмов БПФ. Например, ввиду сложности операции перестановки данных ее проще всего понять на конкретных примерах.

Дискретное преобразование Фурье (ДПФ) конечной последовательности $\{x(n)\}$, где $0 \leq n \leq N - 1$:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)nk}, \quad (1)$$

где $k = 0, 1, \dots, N - 1$ или в более удобной форме

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}, \quad (2)$$

где $W = e^{-j(2\pi)/N}$. Легко показать, что W^{nk} является периодической последовательностью с периодом N , т.е.

$$W^{(n+mN)(k+lN)} = W^{nk}, \quad (3)$$

где $m, l = 0, \pm 1, \dots$

Периодичность W^{nk} является одним из ключевых моментов БПФ. Часто периодичность W^{nk} подчеркивают тем, что вместо W записывают W_N .

Из соотношения (1) следует, что в случае, когда последовательность $x(n)$ является комплексной, при прямом вычислении N -точечного ДПФ нужно выполнить $(N-1)^2$ комплексных умножений и $N(N-1)$ комплексных сложений. Таким образом, для достаточно больших N (порядка 1000) прямое вычисление ДПФ требует выполнения чрезмерного количества вычислительных операций. Основная идея БПФ состоит в том, чтобы разбить исходную N -точечную последовательность на две более короткие последовательности, ДПФ которых могут быть скомбинированы таким образом, чтобы получилось ДПФ исходной N -точечной последовательности. Так, например, если N четное, а исходная N -точечная последовательность разбита на две $(N/2)$ -точечные последовательности, то для вычисления искомого N -точечного ДПФ потребуется порядка $(N/2)^2 \cdot 2 = N^2/2$ комплексных умножений, т.е. вдвое меньше по сравнению с прямым вычислением. Здесь множитель $(N/2)^2$ дает число умножений, необходимое для прямого вычисления $(N/2)$ -точечного ДПФ, а множитель 2 соответствует двум ДПФ, которые должны быть вычислены. Эту операцию можно повторить, вычисляя вместо $(N/2)$ -точечного ДПФ два $(N/4)$ -точечных ДПФ (предполагая, что $N/2$ четное) и сокращая тем самым объем вычислений еще в два раза. Выигрыш в два раза является приближенным, поскольку не учитывается, каким образом из ДПФ меньшего размера образуется искомое N -точечное ДПФ.

Проиллюстрируем описанную методику для N -точечной последовательности $\{x(n)\}$, считая, что N равно степени 2. Введем две $(N/2)$ -точечные последовательности $\{x_1(n)\}$ и $\{x_2(n)\}$ из четных и нечетных членов $\{x(n)\}$ соответственно, т. е.

$$x_1(n) = x(2n), n = 0, 1, \dots, N/2 - 1, \quad (4)$$

$$x_2(n) = x(2n + 1), n = 0, 1, \dots, N/2 - 1. \quad (5)$$

N -точечное ДПФ последовательности $\{x(n)\}$ можно записать как

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} + \sum_{n=0}^{N-1} x(n)W_N^{nk} = \\ &= \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k}. \end{aligned} \quad (6)$$

С учетом того, что

$$W_N^2 = [e^{j(2\pi/N)}]^2 = e^{j[2\pi/(N/2)]} = W_{N/2}, \quad (7)$$

перепишем выражение (6) в виде

$$X(k) = \sum_{n=0}^{N/2-1} x_1(n)W_{N/2}^{nk} + W_{N/2}^{nk} \sum_{n=0}^{N/2-1} x_2(n)W_{N/2}^{nk}, \quad (8)$$

$$X(k) = X_1(k) + W_N^k X_2(k), \quad (9)$$

где $X_1(k)$ и $X_2(k)$ равны $(N/2)$ -точечным ДПФ последовательностей $x_1(n)$ и $x_2(n)$.

Из формулы (8) следует, что N -точечное ДПФ $X(k)$ может быть разложено на два $(N/2)$ -точечных ДПФ, результаты которых объединяются согласно (9). Если бы $(N/2)$ -точечные ДПФ вычислялись обычным способом, то для вычисления N -точечного ДПФ потребовалось бы очевидно $(N^2/2 + N)$ комплексных умножений. При больших N (когда $(N/2) \gg N$) это позволяет сократить время вычисления на 50 %.

Поскольку $X(k)$ определено при $0 \leq k \leq N - 1$, а $X_1(k)$ и $X_2(k)$ определены при $0 \leq k \leq N/2 - 1$, необходимо доопределить формулу (9) для $k \gg N/2$. Это определение достаточно очевидно и может быть описано следующим образом:

$$X(k) = \begin{cases} X_1(k) + W_N^k X_2(k), & 0 \leq k \leq N/2 - 1, \\ X_1(k - N/2) + W_N^k X_2(k - N/2), & N/2 \leq k \leq N - 1. \end{cases} \quad (10)$$

На рис. 1 с помощью направленного графа представлена последовательность операций при выполнении восьмиточечного ДПФ с использованием двух четырехточечных преобразований.

Входная последовательность $x(n)$ сначала разбивается на две последовательности $x_1(n)$ и $x_2(n)$ из четных и нечетных членов $x(n)$, после чего рассчитываются их преобразования $X_1(k)$ и $X_2(k)$.

Затем в соответствии с формулой (10) получают $X(k)$.

Рассмотренная схема вычислений может быть использована для расчета $N/2$ -точечных ДПФ в соответствии с формулами (9) и (10). Каждая из последовательностей $x_1(n)$ и $x_2(n)$ разбивается на две последовательности, состоящие из четных и нечетных, членов. Анало-

гично $N/2$ -точечные ДПФ могут быть записаны как комбинации двух $N/4$ -точечных ДПФ, т.е.

$$X_1(k) = A(k) + W_{N/2}^k B(k) \quad (11)$$

или

$$X_2(k) = A(k) - W_{N/2}^k B(k), \quad (12)$$

где $0 \leq k \leq N/2 - 1$, $A(k)$ и $B(k)$ – $N/4$ -точечные ДПФ соответственно четных и нечетных членов $x_1(n)$. На рис. 2 показан результирующий направленный граф, в котором четырехточечные ДПФ из рис. 1 рассчитываются согласно (12).

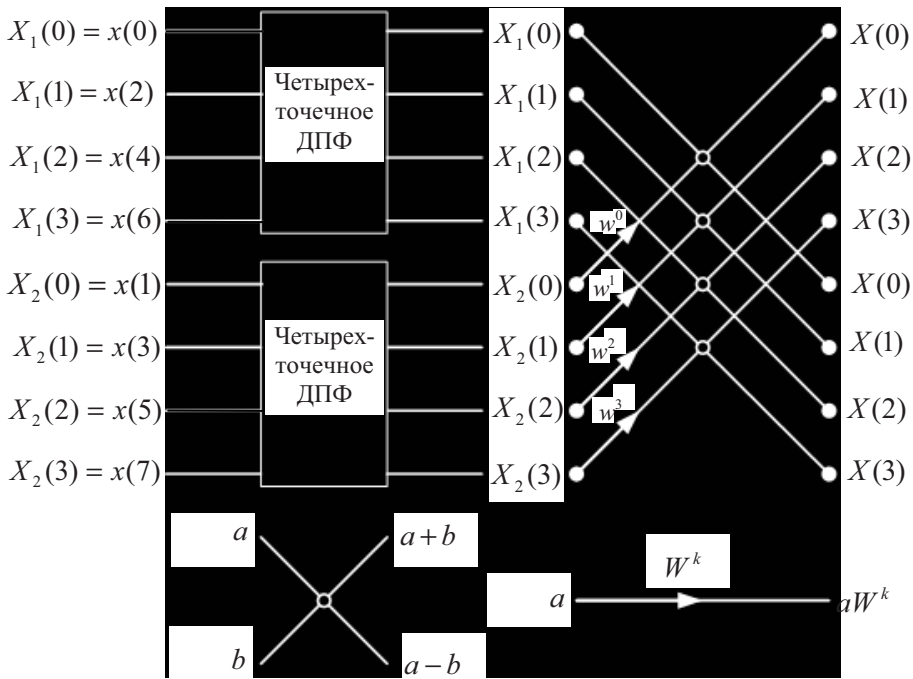


Рис. 1. Вычисление восьмиточечного ДПФ через два четырехточечных ДПФ

Процесс уменьшения размера ДПФ от L до $L/2$, где L равно степени 2, может быть продолжен до тех пор, пока не останутся только двухточечные ДПФ.

Двухточечное ДПФ $F(k)$, $k = 0, 1$, может быть рассчитано без использования умножений по формулам:

$$F(0) = f(0) + f(1)W_8^0, \quad (13)$$

$$F(1) = f(0) - f(1)W_8^4. \quad (14)$$

Здесь $f(n)$, $n=0,1$, – преобразуемая двухточечная последовательность. Поскольку $W_8^0 = 1$ и $W_8^4 = -1$, для вычислений по формулам (13) действительно не нужны операции умножения.

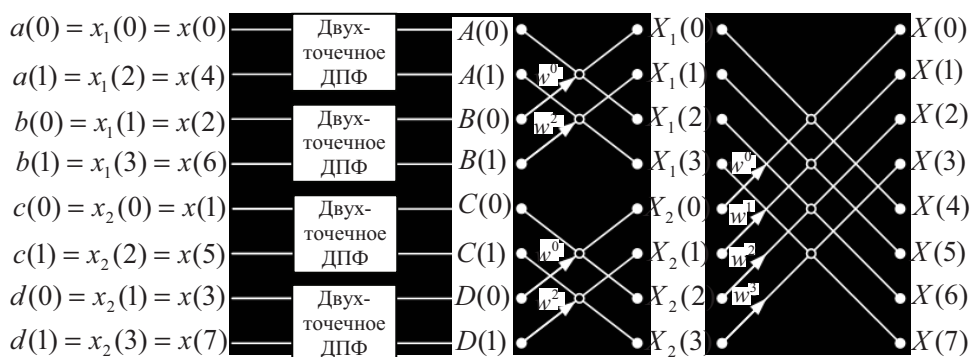


Рис. 2. Вычисление восьмиточечного ДПФ через два четырехточечных ДПФ, которые в свою очередь вычисляются через четыре двухточечных ДПФ

Таким образом, восьмиточечное ДПФ (рис. 1, 2) в итоге сводится к алгоритму, описываемому направленным графом, представленным на рис. 3.

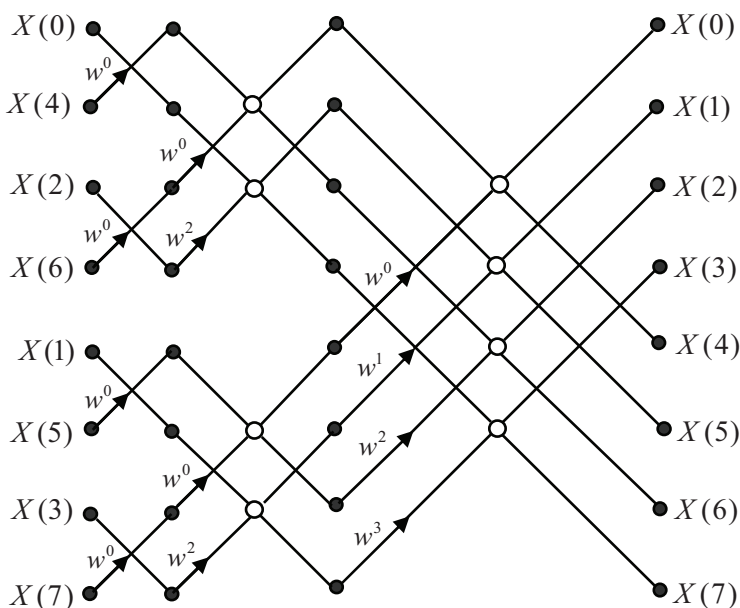


Рис. 3. Восьмиточечное ДПФ, полученное последовательным прореживанием в 2 раза

Перестановка данных и двоичная инверсия. Еще одной особенностью алгоритма с прореживанием по времени (как, впрочем, и большинства других алгоритмов БПФ) является необходимость такой перестановки элементов входной последовательности, чтобы выходная последовательность $X(k)$ имела естественный (прямой) порядок расположения, т. е. $k = 0, 1, \dots$. В примере на рис. 3 для этого требовался следующий порядок размещения входной последовательности: $x(0)$, $x(4)$, $x(2)$, $x(6)$, $x(1)$, $x(5)$, $x(3)$ и $x(7)$. Характер перестановки элементов входной последовательности может быть описан сравнительно просто. Ниже будет показано, что в случае, когда N является степенью 2, входная последовательность должна быть расположена в памяти в двоично-инверсном порядке для того, чтобы выходная последовательность получалась в прямом порядке. Двоично-инверсный порядок определяется следующим образом.

Если записать порядковые номера элементов входной последовательности в двоичном коде, используя L двоичных разрядов, причем $N = 2^L$, а затем инвертировать порядок следования разрядов, то получаемые при этом числа и будут номерами элементов входной последовательности после их перестановки. Так, для случая $N = 8 = 2^3$ прямой порядок номеров приведен в таблице слева, а двоично-инверсный порядок – справа.

Порядковые номера элементов входной последовательности

Номер	Двоичное представление	Двоичная инверсия	Двоично-инверсный код
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Цифровые фильтры

Рассмотрим наиболее простые из цифровых фильтров – фильтры с постоянными параметрами.

На вход цифрового фильтра подается входной сигнал $x(kT)$ в виде последовательности числовых значений, следующих с интервалом T (рис. 4, а). При поступлении каждого очередного значения сигнала $x(kT)$ в цифровом фильтре производится расчет очередного значения выходного сигнала $y(kT)$.

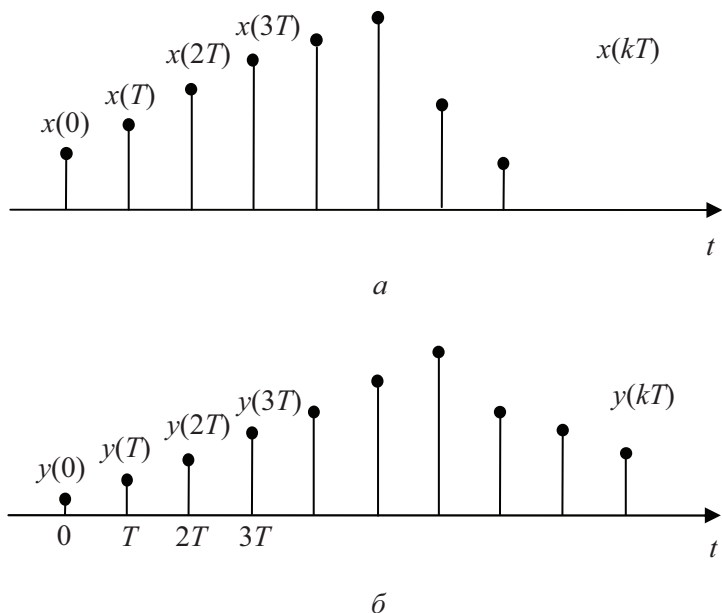


Рис. 4. Сигнал на входе (а) и на выходе (б) цифрового фильтра

Алгоритмы расчета могут быть самыми разнообразными; в процессе расчета кроме последнего значения входного сигнала $x(kT)$ могут использоваться предыдущие значения входного и выходного сигналов: $x(kT - T)$, $x(kT - 2T)$, ..., $y(kT - T)$, $y(kT - 2T)$, ... Сигнал на выходе цифрового фильтра $y(kT)$ (рис. 4, б) также представляет собой последовательность числовых значений, следующих с интервалом T . Этот интервал является единым для всего устройства цифровой обработки сигналов.

Поэтому если на вход цифрового фильтра подать простейший сигнал в виде единичного импульса (рис. 5, а)

$$x_0 = \begin{cases} 1, k = 0, \\ 0, k \neq 0, \end{cases} \quad (15)$$

то на выходе получим сигнал в виде дискретной последовательности числовых значений, следующих с интервалом T . По аналогии с обыч-

ными аналоговыми цепями назовем этот ответный сигнал импульсной характеристикой фильтра $g(kT)$ (рис. 5, б). В отличие от импульсной характеристики аналоговой цепи функция $g(kT)$ является безразмерной.

При подаче на вход фильтра произвольного дискретного сигнала $x(kT)$ (см. рис. 4, а), представляющего собой набор дискретных значений $x(0), x(T), x(2T), \dots$, под действием первого элемента $x(0)$ на выходе фильтра формируется последовательность $g(kT)$, умноженная на $x(0)$; при действии $x(T)$ – последовательность $g(kT)$, умноженная на $x(T)$ и сдвинутая вправо на величину T , и т.д.

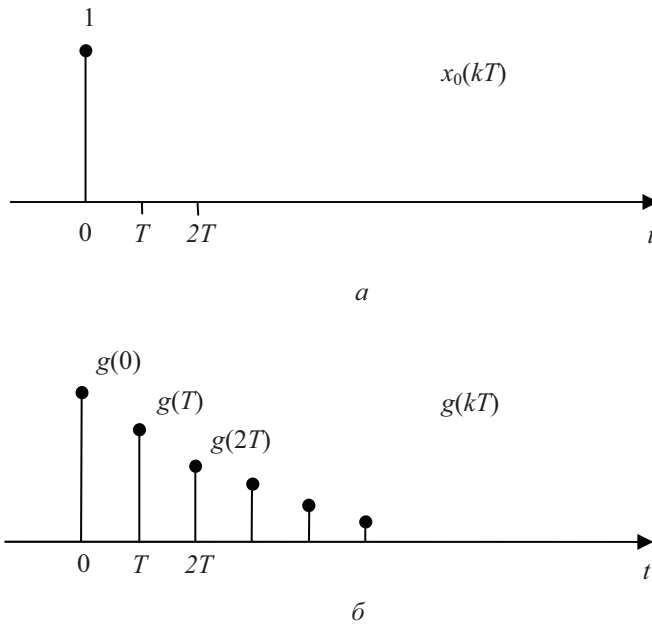


Рис. 5. Единичный импульс (а) и импульсная характеристика цифрового фильтра (б)

В результате на выходе получим последовательность $y(\pi T)$, причем

$$y(0) = g(0)x(0), \quad (16)$$

$$y(T) = g(T)x(0) + g(0)x(T), \quad (17)$$

$$y(2T) = g(2T)x(0) + g(T)x(T) + g(0)x(2T), \quad (18)$$

$$x(nT) = \sum_{k=0}^n x(kT)g(nT - kT) = \sum_{k=0}^n x(nT - kT)g(kT). \quad (19)$$

Таким образом, выходной сигнал определяется как дискретная свертка входного сигнала и импульсной характеристики. В этом отношении цифровые фильтры аналогичны обычным цепям, где выходной сигнал равен свертке входного сигнала и импульсной характеристики.

Формула (18) представляет собой алгоритм цифровой фильтрации. Если импульсная характеристика фильтра описывается последовательностью с конечным числом членов, то фильтр может быть реализован в виде схемы, изображенной на рис. 6. Здесь буквой T обозначены элементы задержки сигнала на время T (на одну ячейку), $g(0), g(T), \dots, g(NT)$ – элементы, умножающие сигнал на соответствующий коэффициент.

Схема, изображенная на рис. 6, не является электрической схемой цифрового фильтра – эта схема представляет собой графическое изображение алгоритма цифровой фильтрации и показывает последовательность арифметических операций, выполняемых при обработке сигнала.

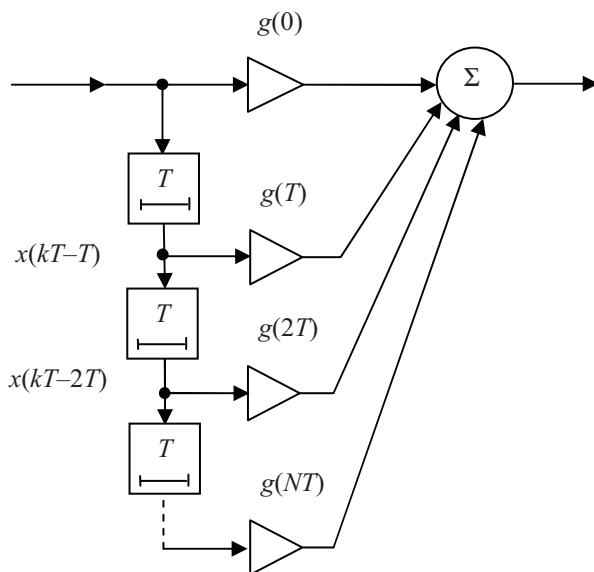


Рис. 6. Схема нерекурсивного цифрового фильтра

Для цифровых фильтров, обрабатывающих сигналы в виде абстрактных числовых последовательностей, понятие «задержка на время T » является не совсем корректным. Поэтому элементы, задерживающие сигнал на одну ячейку, на схемах цифровых фильтров обычно отмечают символом z^{-1} , обозначающим задержку сигнала на языке z -преобразований.

Вернемся к схеме цифрового фильтра, изображенной на рис. 6. Такие фильтры, где для расчета используются лишь значения входного сигнала, называют простыми или нерекурсивными. Алгоритм нерекурсивного фильтра легко записать, если известна импульсная характеристика фильтра. Для практической реализации алгоритма [7] необходимо, чтобы импульсная характеристика содержала конечное число членов. Если импульсная характеристика содержит бесконечное число членов, но они быстро убывают по величине, то можно ограничиться конечным числом членов, отбросив те, значения которых малы. В случае, если элементы импульсной характеристики не убывают по величине, алгоритм нерекурсивного фильтра оказывается нереализуемым.

В качестве примера рассмотрим простейший цифровой фильтр, аналогичный RC-цепи (рис. 7). Импульсная характеристика RC-цепи имеет вид:

$$g(t) = \frac{1}{\tau} e^{-t/\tau}. \quad (20)$$

Чтобы записать импульсную характеристику соответствующего цифрового фильтра, в выражении (20) t следует заменить на kT . Однако импульсная характеристика RC-цепи имеет размерность $1/c$, а импульсная характеристика цифрового фильтра должна быть безразмерной. Поэтому опустим множитель $1/\tau$ в выражении (20) и запишем импульсную характеристику цифрового фильтра в виде

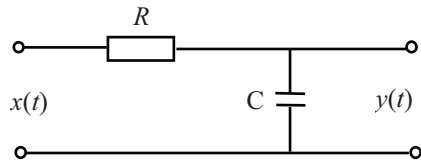


Рис. 7. RC-цепь

$$g(kT) = e^{-kT/\tau}. \quad (21)$$

Такая импульсная характеристика содержит бесконечно много членов, но их величина убывает по экспоненциальному закону, и можно ограничиться N членами, выбирая N таким, чтобы

$$e^{-NT/\tau} \ll 1. \quad (22)$$

Теперь можно записать выражение для сигнала на выходе фильтра:

$$y(nT) = x(nT) + e^{-T/\tau} x(nT + T) + e^{-2T/\tau} x(nT - 2T) + \dots + e^{-NT/\tau} x(nT - NT). \quad (23)$$

Это выражение является одновременно алгоритмом цифрового фильтра. Схема этого фильтра представлена на рис. 8.

Как видно из вышеизложенного, при обучении основам цифровой обработки сигналов приходится изучать весьма сложный математический аппарат, и зачастую физический смысл происходящих процессов теряется за математическими формулами.

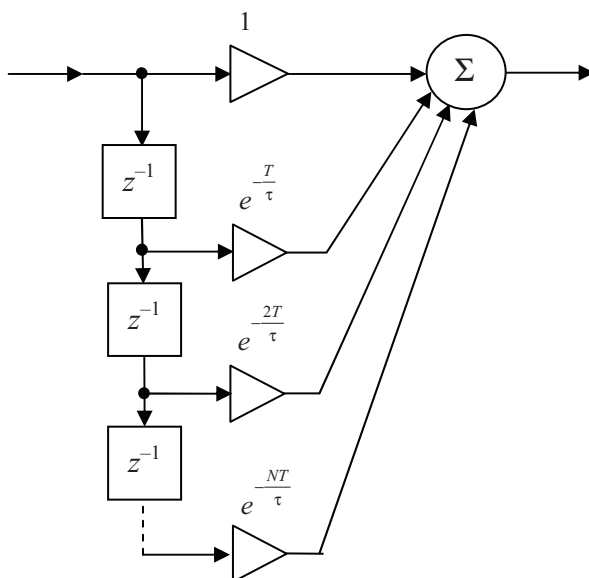


Рис. 8. Схема нерекурсивного цифрового фильтра, аналогичного RC-цепи «Basic DSP»

В связи с этим полезно было бы иметь некий программный комплекс, наглядно демонстрирующий физические аспекты ЦОС – построение стандартной модели, изучение поведения, просмотр реакций.

Примером модели последовательности цифровой обработки сигнала в современной аппаратуре может служить рис. 9.

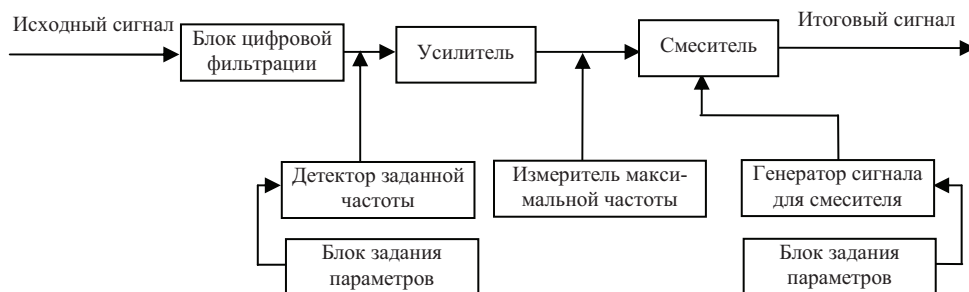


Рис. 9. Модель последовательности цифровой обработки сигнала

Для этих целей можно использовать программный пакет MatLab, но у данного решения есть недостатки:

- MatLab является платным программным продуктом, и для его использования необходимо приобретение достаточно дорогой лицензии;
- для построения модели требуется опыт работы в среде MatLab;
- для эффективной работы среды MatLab требуется мощный персональный компьютер.

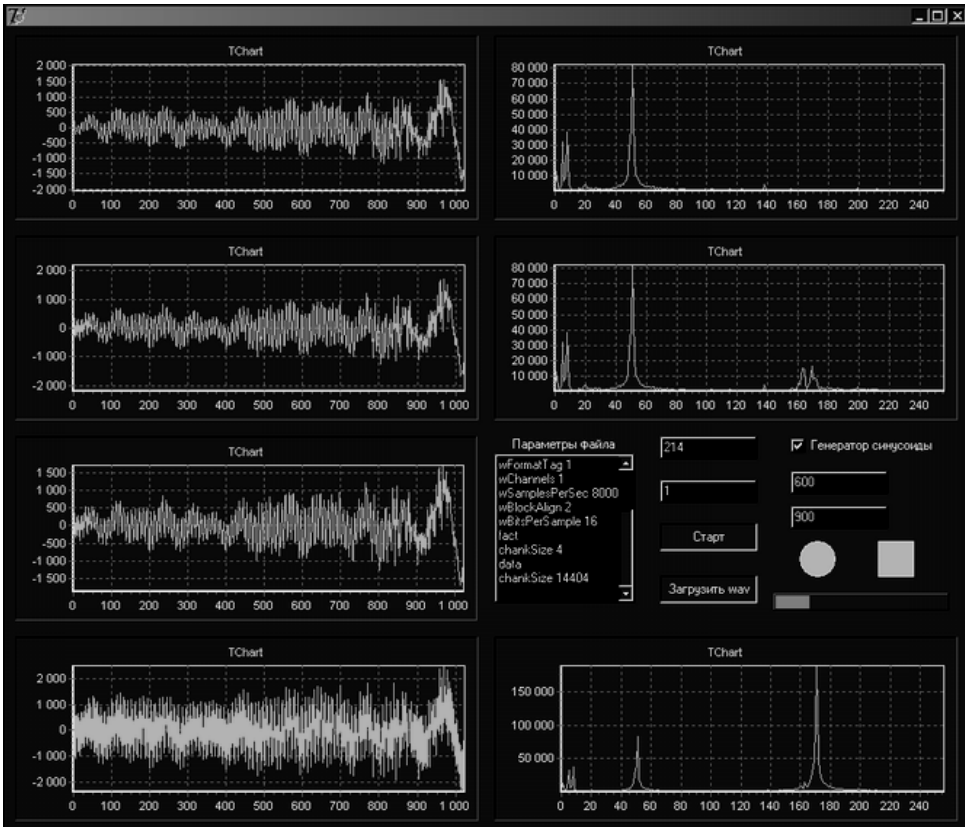


Рис. 10. Внешний вид работы программы

Поэтому был создан программный продукт, имеющий следующие свойства:

- возможность моделирования процессов ЦОС,
- простота пользовательского интерфейса,
- минимальность аппаратных требований,
- возможность работы без процедуры установки приложения,
- бесплатность.

В качестве исходных данных, для обработки информации принято решение использовать звуковой файл с расширением *.wav. Этот формат широко распространен, и результаты ЦОС (изменение спектрального состава звука, выделение и удаление заданных частотных полос, смешение двух и более звуковых сигналов, измерение максимальной частоты сигнала) можно не только увидеть на мониторе в виде графиков (как спектральных, так и амплитудных), но и услышать.

Программа «Basic DSP» версии 1.0 была написана на языке Delphi в среде Borland Delphi 7 [2] и в значительной мере отвечала вышеозначенным заданным свойствам.

Исходный сигнал моделируется путём загрузки звуковых файлов формата WAVE («Загрузить wav» «Старт») (рис. 10). Имеются индикатор продолжительности процесса симуляции (зависит от длины WAVE файла) и окно основных свойств загруженного файла (частота дискретизации, количество бит и т.д.).

Обучающийся имеет возможность визуального наблюдения за амплитудным и спектральным составом:

- а) исходного сигнала – верхняя пара индикаторов;
- б) сигнала после прохождения цифрового фильтра на 2600 МГц – вторая сверху пара индикаторов;
- в) сигнала после усиления в заданное число раз – третий индикатор сверху;
- г) сигнала после суммирования с сигналом синусоиды задаваемой частоты и амплитуды – нижняя пара индикаторов.

Также в программе реализованы цифровой обнаружитель сигнала с индикатором, детектор гармоникой частотой 2600 Гц с индикатором, измеритель частоты гармоникой с максимальной амплитудой, включение/отключение генератора синусоиды.

В последующих версиях программы планируются: добавление функций выбора параметров цифровых фильтров, их количества, возможность сохранения выходного сигнала в различных форматах, выбор вида сигнала, с которым пользователь может смешать свой исходный звуковой сигнал, и количества этих сигналов.

Также планируется переработка пользовательского интерфейса с целью повышения эстетических и эргономических качеств.

Библиографический список

1. Лайонс Р. Цифровая обработка сигналов. – М.: БИНОМ, 2003.
2. Архангельский А.Я. Программирование в Delphi 7. – М.: БИНОМ, 2000.
3. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов: пер. с англ. / под ред. Ю.Н. Александрова. – М.: Мир, 1978.
4. Цифровая обработка сигналов: учеб. пособие для вузов / Л.М. Гольденберг [и др.]. – 2 изд., перераб. и доп. – М.: Радио и связь, 1990.
5. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов: пер. с англ. / под ред. С.Я. Шаца. – М.: Связь, 1979.
6. Burrus C.S., Parks T.W. DFT/FFT and Convolution Algorithms, Wiley Interscience. – NY, 1985.
7. URL: <http://www.codenet.ru>.

Получено 09.09.2011