

**ГОСУДАРСТВЕННЫЙ КОМИТЕТ РОССИЙСКОЙ  
ФЕДЕРАЦИИ ПО ВЫСШЕМУ ОБРАЗОВАНИЮ**

**НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Физический факультет**

**Кафедра радиофизики**

## **ЦИФРОВЫЕ ФИЛЬТРЫ**

**Методические указания к лабораторной работе практикума  
"Технические Средства Автоматизации Научных Исследований"**

**Новосибирск 1995**

Лабораторная работа посвящена изучению аппаратных и программных средств, применяемых в системах автоматизации научных исследований для сбора и обработки данных, в частности, цифровой фильтрации сигналов на фоне аддитивного шума.

Цель работы – изучение компонентов программно-аппаратных средств для регистрации и последующей обработки сигнала: аналого-цифровых преобразователей, дискретного преобразования Фурье, различных методов цифровой фильтрации.

Задача студента – написать и отладить компьютерную программу для регистрации периодического сигнала на фоне аддитивного шума, реализовать в программе различные методы цифровой фильтрации – метод вычисления, гребенчатый фильтр, согласованный фильтр, фильтр Винера, взаимнокорреляционный фильтр и другие.

Составитель Г.И. Кузин

Печатается по решению кафедры радиофизики.

С. ответственный за выпуск Г.И. Кузин

© Новосибирский государственный университет, 1995

## ВВЕДЕНИЕ

Проблема "очистки" полезного сигнала от шума возникает не только в технике связи, где она особенно актуальна, но и в экспериментальных научных исследованиях, когда шумы и помехи не позволяют провести измерения с достаточной точностью, либо сигнал скрыт под шумами.

Методы цифровой обработки сигналов, получившие быстрое распространение с появлением компактных, достаточно мощных компьютеров и специализированных процессоров, широко используются в научном эксперименте. Преимущества цифровой обработки сигналов заключаются в большой точности и воспроизводимости результатов, возможности быстрой реализации различных сложных алгоритмов, что особенно ценно в научном эксперименте. С помощью компьютера легко синтезировать самые различные фильтры, в том числе и такие, которые очень трудно или вообще невозможно реализовать с помощью аналоговой техники.

Вместе с тем методы цифровой обработки сигналов имеют ряд особенностей и ограничений, которые необходимо учитывать в практической работе.

В общем случае блок-схему цифрового фильтра можно представить в следующем виде.

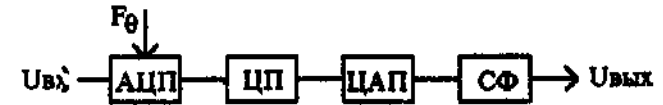


Рис. 1

где АЦП – аналого-цифровой преобразователь;

ЦП – цифровой процессор (компьютер);

ЦАП – цифро-аналоговый преобразователь;

СФ – синтезирующий фильтр;

$F_0$  – тактовая частота дискретизации сигнала.

Синтезирующий фильтр предназначен для получения на выходе непрерывного аналогового сигнала. Для этой цели обычно используют фильтр нижних частот – ФНЧ. При такой работе восстанавливать аналоговый сигнал не требуется, достаточно получить его график на экране дисплея. В этом случае

функцию синтезирующего фильтра выполняет линейная интерполация между дискретными точками.

### Дискретизация сигнала

Как правило, исходный сигнал, непрерывен во времени или пространстве. Это может быть зависимость от времени какого либо параметра, например, тока или зависимость интенсивности засветки кадра изображения от координат. Во всех случаях для того, чтобы сделать сигнал пригодным для обработки на компьютере, его нужно "оцифровать", то есть получить значения сигнала (отсчеты) в цифровом виде в дискретные моменты времени. Эту задачу, обычно выполняет АЦП, принцип работы которого подробно описан в методическом описании к лабораторной работе N5 практикума ТСАНИ.

Так как устройство выборки и хранения аналого-цифрового преобразователя запоминает входной сигнал за время много меньшее интервала между отсчетами, дискретный сигнал математически можно представить в виде:

$$U_{\theta}(t) = \sum_k U(k\theta) \cdot \delta(t - k\theta), \quad (1)$$

где  $\theta$  – период тактовой частоты запуска АЦП,  $k$  – номер отсчета,  $U(k\theta)$  – значение сигнала в момент времени  $k\theta$ ,  $\delta(t - k\theta)$  – дельта-функция. Формула (1) не означает, конечно, что сигнал может иметь бесконечную величину, она приобретает смысл лишь после интегрирования по времени и отражает тот факт, что отсчеты берутся в дискретных точках. Информация о сигнале в промежуточные моменты времени в общем случае теряется. Тем не менее, когда частота дискретизации достаточно велика, а спектр сигнала ограничен, исходный сигнал может быть полностью восстановлен. Теорема Котельникова, которую иногда называют теоремой Шеннона или просто теоремой отсчетов, утверждает, что если частота дискретизации больше удвоенной максимальной частоты  $F_m$  в спектре сигнала, то последовательность дискретных отсчетов полностью определяет исходный сигнал и функция  $U(t)$  может быть точно восстановлена:

$$U(t) = \sum_k U(k\theta) \frac{\sin[2\pi f \cdot (t - k\theta)]}{2\pi f \cdot (t - k\theta)} \quad (2)$$

Однако реальные сигналы обычно не имеют четко ограниченного спектра, так как такой сигнал должен быть определен в области  $-\infty < t < +\infty$ . Таким образом, по дискретной последовательности исходный сигнал можно восстановить только некоторой ошибкой. Эта ошибка заключается в том, что часть спектра с частотами  $f > F_{\theta}/2$  переносится в низкочастотную область. Этот эффект нужно обязательно иметь в виду при цифровой обработке сигнала. Пусть, например, входное воздействие состоит из смеси полезного сигнала и шума, т.е. по каким либо основаниям известно, что спектр сигнала сосредоточен в области частот  $f < F_m$ , а нижняя граничная частота в спектре шума  $F_{\xi} > F_m$ , то есть, спектры сигнала и шума не перекрываются. Если попытаться оцифровать входное воздействие с тактовой частотой  $F_{\theta} > 2F_m$ , затем найти его спектр, предполагая в дальнейшем занулить все частотные составляющие в области  $|f| > F_m$  и таким способом избавиться от шумов, то в полученном спектре обнаружим, что шум накладывается на спектр сигнала. Этот эффект наложения связан с тем, что спектр дискретного сигнала получается из спектра исходного сигнала сдвигом на все возможные значения, кратные  $F_{\theta}$  (свертка спектров) и ослабить его можно только предварительно пропустив сигнал через фильтр нижних частот.

Рисунок 2 иллюстрирует перенос спектра шума в спектральную область сигнала.

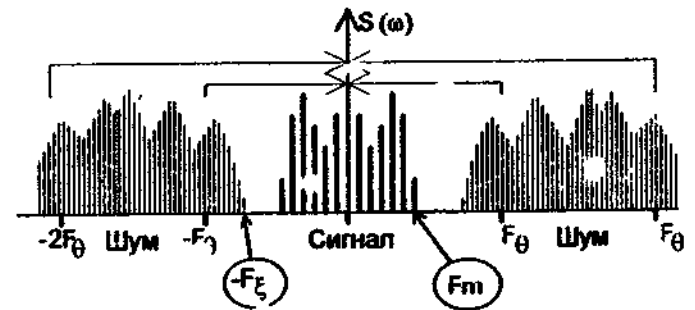


Рис. 2

### Дискретные спектры

Спектральная плотность  $S(\omega)$  произвольной, кусочно-непрерывной функции  $U(t)$  находится по формуле:

$$S(\omega) = \int_{-\infty}^{\infty} U(t) \cdot e^{-i\omega t} dt. \quad (3)$$

Применив это преобразование к выражению (1), при  $k=0, \dots, \infty$  получаем:

$$S_0(\omega) = \int_0^{\infty} e^{-i\omega t} \sum_{k=0}^{\infty} U(k\theta) \cdot \delta(t - k\theta) dt = \sum_{k=0}^{\infty} U(k\theta) \cdot e^{-i\omega k\theta}. \quad (4)$$

Этот спектр имеет периодическую структуру с периодом  $2\pi/\theta$ . Пусть дискретная выборка  $U(k\theta)$  содержит  $N$  отсчетов. Продолжим ее периодически с периодом  $N\theta$ . Эту функцию можно разложить в ряд Фурье с интервалом между гармониками  $\Delta\omega = 2\pi/N\theta$ . При четном  $N$  имеем:

$$S_0(n\Delta\omega) = \sum_{k=0}^{N-1} U(k\theta) \cdot e^{-i\Delta\omega n k \theta} = \sum_{k=0}^{N-1} U(k\theta) \cdot e^{-i\frac{2\pi}{N} nk}, \quad (5)$$

$$n = 0, \pm 1, \pm 2, \dots, \pm N/2.$$

Заменяя  $n\Delta\omega$  на  $n$ , а  $k\theta$  на  $k$ , получим дискретное преобразование Фурье (ДПФ):

$$S(n) = \sum_{k=0}^{N-1} U(k) \cdot e^{-i\frac{2\pi}{N} nk}, \quad n = 0, \pm 1, \pm 2, \dots, \pm N/2 \quad (6)$$

ДПФ (6) обладает одним очень важным свойством. Как можно показать, при четном  $N$  и действительном  $U(t)$  справедливо равенство:

$$S(N/2+j) = S^*(N/2-j), \quad j = 0, 1, \dots, N/2, \quad (7)$$

где  $S^*(n)$  — величина, комплексно сопряженная  $S(n)$ .

Так как полученный спектр можно периодически продолжить, то очевидно, что в пределах одного периода формулу (6) можно переписать в виде

$$S(n) = \sum_{k=0}^{N-1} U(k) e^{-i\frac{2\pi}{N} nk}, \quad n = 0, 1, 2, \dots, N-1, \quad (8)$$

что намного удобнее, поскольку переменные  $n$  и  $k$  при вычислениях будут индексами массивов.

Преобразование (8) соответствует обратное преобразование Фурье:

$$U(k) = \frac{1}{N} \sum_{n=0}^{N-1} S(n) \cdot e^{+i\frac{2\pi}{N} nk}, \quad k = 0, 1, 2, \dots, N-1. \quad (9)$$

Таким образом, дискретному сигналу  $U(k)$  длительностью  $N$  отсчетов, периодически продолженному с периодом  $N$ , соответствует дискретный спектр  $S(n)$ , также имеющий периодическую структуру с тем же периодом  $N$ .

### Быстрое преобразование Фурье

Как видно из формулы (8), для того чтобы найти частотный спектр сигнала, представленного  $N$  отсчетами, нужно сделать  $N^2$  операций умножения. Для выполнения вычислительной работы такого объема требуется значительное время. Например, при  $N=4096$  компьютер типа IBM PC AT затратит на вычисление спектра около 30 минут. Это обстоятельство долгое время препятствовало широкому применению преобразования Фурье для обработки данных. Ситуация изменилась в 1965 году, когда Джеймс Кули (фирма ИБМ) и Джон Тьюки (Bell Laboratories) разработали очень эффективный алгоритм вычисления, получивший известность как быстрое преобразование Фурье. Метод основан на симметрии и периодичности тригонометрических функций в формуле (8). В самом деле, если в качестве  $N$  выбрать степень двойки, то в сумме (8) будет много одинаковых слагаемых, которые не имеет смысла каждый раз вычислять. Как оказывается, можно обойтись всего  $N \cdot \log_2 N$  операциями умножения. При  $N=4096$  (размер буферной памяти АЦП-101SK, АЦП ФК-4225) алгоритм БПФ работает примерно в 300 раз быстрее чем решение задачи "в лоб".

В составе прикладного программного обеспечения (TSANI.LIB) функция `fft(...)` использует этот алгоритм для вычисления спектра.

Желающих разобраться с этим алгоритмом отсылаем к [3] и учебнику [2]. В последнем алгоритм описан весьма подробно. Там же приведен текст программы на ФОРТРАНе, правда, без комментариев.

### Функция корреляции

Корреляционный анализ является очень эффективным методом обнаружения периодического сигнала в шуме, определения меры соответствия между массивами данных и тесно связан с преобразованием Фурье.

Функция автокорреляции  $R(\tau)$  и функция взаимной корреляции  $R_{12}(\tau)$  определяются для действительных  $U(t)$ ,  $U_1(t)$  и  $U_2(t)$  следующим образом:

$$R(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T U(t) \cdot U(t-\tau) \cdot dt, \quad (10)$$

$$R_{12}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T U_1(t) \cdot U_2(t-\tau) \cdot dt.$$

Корреляционная функция характеризует степень связи между сигналом и его сдвинутой на  $\tau$  копией —  $R(\tau)$  или между двумя различными сигналами —  $R_{12}(\tau)$ .

Некоторые свойства корреляционной функции:

- а) автокорреляционная функция периодического процесса есть периодическая функция с тем же периодом;
  - б)  $R(t)$  — действительная, четная функция и имеет максимум при  $t=0$ , равный мощности сигнала;
  - в) если скрытая в шумах периодическая функция отсутствует, то  $R(t)$  асимптотически стремится к квадрату среднего значения (к нулю, если сигнал не имеет постоянной составляющей);
- Функция автокорреляции и спектр мощности связаны преобразованием Фурье:

$$|S(\omega)|^2 = \int_{-\infty}^{\infty} R(\tau) \cdot e^{-i\omega\tau} d\tau; \quad R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(\omega)|^2 \cdot e^{+i\omega\tau} d\omega, \quad (11)$$

$$|S(n)|^2 = \sum_{k=0}^{N-1} R(k) \cdot e^{-i\frac{2\pi}{N}kn}; \quad R(k) = \frac{1}{N} \sum_{n=0}^{N-1} |S(n)|^2 \cdot e^{+i\frac{2\pi}{N}kn} \quad (12)$$

Поскольку корреляционная функция действительная и четная, ее преобразование Фурье (спектр мощности) является действительной, неотрицательной, четной функцией частоты.

Как видно из (11) и (12), автокорреляционная функция зависит только от квадратов модулей комплексных амплитуд спектрального разложения сигнала, следовательно, автокорреляционная функция не содержит информации о фазовом спектре сигнала. Таким образом, одной и той же автокорреляционной функции соответствует множество сигналов с различными фазовыми спектрами. Поэтому по автокорреляционной функции восстановить форму сигнала невозможно, хотя некоторые параметры сигнала можно оценить, например, задержку одного импульса относительно другого, соотношение амплитуд, длительность.

Взаимнокорреляционная функция отличается от автокорреляционной прежде всего тем, что частично сохраняет информацию о фазовом спектре сигналов в виде разности фаз соответствующих гармоник, хотя взаимной однозначности между сигналами и их функцией взаимной корреляции и в этом случае нет.

По аналогии с (11), (12) функция взаимной корреляции имеет своим Фурье-изображением так называемый взаимный энергетический спектр мощности:

$$W_{12}(\omega) = S_1^*(\omega) \cdot S_2(\omega) \Leftrightarrow R_{12}(\tau), \quad (13)$$

где  $\Leftrightarrow$  означает соответствие относительно преобразования Фурье.

Как видно из (10), время, необходимое для численного расчета корреляционной функции дискретизированного сигнала, как и в случае преобразования Фурье, примерно равно  $N^2$ . Поэтому корреляционные функции удобнее вычислять "в обход" с помощью алгоритма БПФ, пользуясь соотношениями (12), (13). Сначала нужно вычислить спектр (или спектры для  $R_{12}(\tau)$ ), затем найти его модуль (произведение) и сделать обратное преобразование Фурье.

### ФИЛЬТРЫ

Задачи, связанные с фильтрацией сигналов, можно условно разделить на две группы: 1) выделение сигнала из шума с минимальной ошибкой; 2) обнаружение сигнала, когда максимизируется отношение сигнал/шум даже ценой

значительного искажения его формы. Эти задачи можно решать разными способами в зависимости от целей и характеристик сигнала. Ниже кратко описаны цифровые фильтры, которые вам нужно реализовать в данной работе.

### Метод накопления

Метод накопления является, пожалуй, самым распространенным способом борьбы с помехами и ошибками из рений благодаря простоте аппаратной реализации. Однако, этот метод эффективен лишь для периодического сигнала, причем процесс регистрации должен быть синхронизирован с принимаемым сигналом.

Пусть входное воздействие представляет собой смесь периодического сигнала и шума  $X(t) = U(t) + \xi(t)$ , причем интервал регистрации содержит  $n$  периодов. Усредняя  $X(t)$  по периодам сигнала  $U(t)$ , получаем:

$$X(t) = \frac{1}{n} \sum_{k=1}^n X_k(t) = U(t) + \frac{1}{n} \sum_{k=1}^n \xi_k(t), \quad (14)$$

где  $k$  – номер периода. Отсюда видно, что отношение сигнал/шум увеличивается в  $n$  раз по мощности и, соответственно, в  $\sqrt{n}$  по амплитуде. Например, при длительности реализации  $N=4096$  для периода сигнала 128 отсчетов метод накопления дает выигрыш 15 децибел, то есть отношение сигнал/шум увеличивается в 32 раза по мощности.

Тот же самый результат можно получить, если в спектре в результате все частотные составляющие кроме тех, в которых содержится сигнал, то есть оставить только гармоники, кратные периоду сигнала (дискретный гребенчатый фильтр).

### Согласованный фильтр

Для обнаружения сигнала на фоне шума или искусственных помех применяются согласованные фильтры. Согласованный фильтр минимизирует отношение сигнал/шум для решения вопроса о том, есть сигнал в данной смеси или нет. Поскольку форма сигнала считается известной, сам сигнал не несет никакой информации, следовательно, можно допускать любые искажения формы сигнала, лишь бы на входе решающего устройства в определенный момент времени отношение сигнал/шум было

максимально. Передаточная функция согласованного фильтра имеет вид:

$$K(\omega) = \frac{b \cdot S_{\text{вх}}^*(\omega)}{W_{\xi}(\omega)} \cdot e^{-i\omega t_0}, \quad (15)$$

где  $S_{\text{вх}}(\omega)$  – спектр обнаруживаемого сигнала,  $W_{\xi}(\omega)$  – спектральная мощность шума на входе фильтра,  $b$  – константа,  $t_0$  выбирается из условия физической осуществимости (к моменту  $t_0$  сигнал должен закончиться).

Для нормального белого шума  $W_{\xi}(\omega) = \text{const.}$  и его можно положить равным  $b$ . Тогда формула (15) упрощается:

$$K(\omega) = S_{\text{вх}}^*(\omega) \cdot e^{-i\omega t_0}. \quad (15a)$$

Форму сигнала на выходе фильтра при отсутствии шумов можно получить с помощью обратного преобразования Фурье:

$$\begin{aligned} U_{\text{вых}}(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} K^*(\omega) \cdot S_{\text{вх}}(\omega) \cdot e^{i\omega t} d\omega = \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |S_{\text{вх}}(\omega)|^2 \cdot e^{i\omega(t-t_0)} d\omega. \end{aligned} \quad (16)$$

Сравнивая полученное выражение с (11), видим, что на выходе согласованного фильтра мы получили автокорреляционную функцию сигнала, сдвинутую на время  $t_0$ . Другими словами, согласованный фильтр определяет функцию взаимной корреляции между входной реализацией и образцом сигнала, который нужно обнаружить.

Среднеквадратичное значение помехи в момент  $t_0$ , когда на выходе фильтра наблюдается пик сигнала, можно найти по формуле:

$$\sigma_{\text{вых}} = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\omega) \cdot K^2(\omega) \cdot d\omega = \frac{W_0}{2\pi} \int_{-\infty}^{\infty} K^2(\omega) \cdot d\omega. \quad (16a)$$

Передаточную функцию (15) легко истолковать физически. Фильтр пропускает частотные составляющие неравномерно, ослабляя те участки спектра, где шум доминирует. Кроме того, фаза выходного сигнала для всех частотных составляющих имеет

вид  $\varphi(t) = \omega(t - t_0)$ , и при  $t = t_0$  все частотные составляющие сигнала имеют нулевую фазу, в результате на выходе имеем пик сигнала. Шумовые же спектральные составляющие складываются со случайными фазами и компенсируют друг друга. Благодаря этому, согласованный фильтр позволяет получить в определенный момент времени  $t_0$  отношение сигнал/шум намного большее, чем метод накопления. На выходе согласованного фильтра обычно ставится компаратор, который и принимает решение о наличии или отсутствия сигнала.

#### Корреляционный метод обнаружения периодических сигналов

Пусть исходная реализация  $X(t)$  состоит, как и в (14), из смеси периодического сигнала  $U(t)$  и шума  $\xi(t)$ :  $X(t) = U(t) + \xi(t)$ . Функция автокорреляции для  $X(t)$  имеет вид:

$$R_{XX}(\tau) = \frac{1}{2T} \int_{-T}^T X(t) \cdot X(t - \tau) dt = \quad (17)$$

$$= R_{UU}(\tau) + R_{\xi U}(\tau) + R_{U\xi}(\tau) + R_{\xi\xi}(\tau).$$

Поскольку шум и сигнал статистически независимы, функции взаимной корреляции  $R_{\xi U}(\tau)$  и  $R_{U\xi}(\tau)$  стремятся к нулю при  $T \rightarrow \infty$ . Функция автокорреляции шума определяется его свойствами, в частности, полосой частот. При этом длина корреляции  $\tau_0$  (характерное время спада функции автокорреляции) обратно пропорциональна полосе частот шума. Белый шум по определению имеет бесконечную полосу частот, следовательно, для него  $\tau_0 = 0$ . Так как автокорреляционная функция периодического сигнала также периодическая, то при  $\tau \gg \tau_0$  и  $T \rightarrow \infty$   $R_{XX}(\tau) = R_{UU}(\tau)$ .

На конечных выборках функции  $R_{\xi U}(\tau)$  и  $R_{U\xi}(\tau)$  не равны нулю, так что возможность обнаружить сигнал определяется отношением сигнал/шум и временем интегрирования.

Отметим одну важную особенность корреляционного метода обнаружения на реализациях конечной длины при использовании алгоритма БПФ. Если число отсчетов  $N$  не кратно периоду, то функция автокорреляции периодического сигнала представляет собой как бы две "треугольные гребенки", вложенные друг в друга и имеющие максимумы на разных концах. При этом сдвиг

"гребенок" относительно друг друга определяется остатком периода. Это обстоятельство затрудняет точное измерение периода. При выполнении соответствующего задания обратите на это внимание.

#### Фильтр Винера

Согласованный фильтр и метод накопления являются крайними случаями задачи фильтрации. Первый дает максимальный выигрыш в отношении сигнал/шум, сильно искажая при этом сигнал, второй асимптотически идеально сохраняет форму, но для конечных выборок отношение сигнал/шум оставляет желать лучшего. Если рассмотреть спектры различных сигналов, то можно увидеть, что многими спектральными составляющими можно пожертвовать без особого ущерба для амплитуды и формы сигнала, значительно улучшив при этом качество фильтрации. Например, колоколообразный импульс имеет быстро спадающий спектр и нет никакого смысла одинаково передавать все частоты, кратные периоду сигнала, как в гребенчатом фильтре.

Очевидно, должен существовать фильтр с оптимальной частотной характеристикой  $K(\omega)$ , который минимизирует среднеквадратичную ошибку измерения сигнала. Эта задача впервые решена Норбертом Винером в довольно общем виде, но здесь мы приведем результат для частного случая. Отвлекаясь от требования физической осуществимости и считая сигнал и шум статистически независимыми, передаточную функцию фильтра Винера можно записать в виде:

$$K(\omega) = \frac{b \cdot W_u(\omega)}{W_u(\omega) + W_\xi(\omega)} \cdot e^{-i\omega t_0}, \quad (18)$$

где  $W_u(\omega)$  и  $W_\xi(\omega)$  спектры мощности сигнала и шума соответственно и  $t_0$  — константы.

Физический смысл формулы (18) можно пояснить следующим образом. При  $W_u(\omega) \gg W_\xi(\omega)$  фильтр одинаково передает все частоты, следовательно, сигнал совсем не искажается (напомним, что множитель  $e^{-i\omega t_0}$  приводит лишь к сдвигу сигнала на время  $t_0$ ). В другом предельном случае  $W_u(\omega) \ll W_\xi(\omega)$  фильтр эффективно подавляет частотные составляющие, не содержащие

сигнала, либо вносящие малый вклад. Форма сигнала при этом искажается, но благодаря тому, что фазовый спектр сохраняется, сигнал остается "узнаваемым".

#### Взаимнокорреляционный фильтр

Как вы, возможно, уже успели заметить, согласованный фильтр и фильтр Винера требуют знания спектра сигнала. В тех случаях, когда точный вид спектра неизвестен, но ограничен неоторой полосой, неплохой результат можно получить вычисляя функцию взаимной корреляции входного сигнала с импульсом нужной длительности. Длительность импульса подбирается в зависимости от ширины спектра сигнала и требований к допустимым искажениям. Этот фильтр сочетает в себе метод накопления и сглаживающий фильтр (ФНЧ).

#### Оценка эффективности фильтров

Качество работы фильтра можно оценивать по различным критериям. В частности, по смыслу и содержанию данной работы оценку можно провести по двум – отношению сигнал/шум  $\gamma = W_u / W_\xi$  и относительной среднеквадратичной ошибке  $\epsilon = (1/N \cdot \sum (U_i - U_{oi})^2 / W_u)^{1/2}$ , где  $W_u$ ,  $W_\xi$  – мощность сигнала и шума соответственно,  $U_{oi}$  – "чистый" сигнал,  $U_i$  – сигнал на выходе фильтра.

По первому критерию можно оценивать практически все фильтры, по второму – только неискажающие или слабоискажающие сигнал. Например, согласованный фильтр настолько искажает форму сигнала, что применение к нему оценки среднеквадратичного отклонения не имеет смысла.

Для определения эффективности фильтра по первому критерию нужно вычислить  $\gamma_{\text{вых}}$  и  $\gamma_{\text{вх}}$  в децибелах и взять их разность. Поскольку фильтр может искажать сигнал, следовательно, и его мощность, для измерения  $\gamma$  нужно применить фильтр сначала к чистому сигналу, а затем к смеси.

Для измерения  $\epsilon$  нужно сначала получить образец чистого сигнала  $U_0$  и записать его в специальный массив, а затем вычислить  $\epsilon$  по вышеприведенной формуле.

Установка состоит из крейта КАМАК, связанного посредством крейт-контроллера с компьютером типа IBM PC, аналого-цифровых преобразователей "Коммутатор и АЦП-10", "АЦП 101SK" и блока генератора "Сигнал+шум". Прохождение сигналов по магистрали КАМАК можно контролировать по индикатору магистрали.

Генератор "Сигнал+шум" выдает на "ВЫХ" смесь периодического сигнала и шума, уровни которых регулируются независимо. Форма сигнала выбирается комбинацией четырех переключателей "Форма сигнала". Блок может генерировать 16 сигналов различной формы. Разъемы "Синхронизация" предназначены для синхронизации запуска АЦП. Блок имеет выходы для синхронизирующих импульсов на каждый такт и на период генератора сигнала устроен следующим образом. В постоянное запоминающее устройство (ПЗУ) в цифровом виде записано 16 различных сигналов. Емкость ПЗУ – 2048 байт. На каждый сигнал отводится 128 байт. Счетчик циклически перебирает адреса соответствующего 128-байтного раздела, который выбирается переключателями. Данные ПЗУ при помощи цифро-аналогового преобразователя преобразуются в аналоговый сигнал, который аддитивно смешивается с шумом. Таким образом, выходной сигнал имеет период, равный 128 внутренним тактам.

На магистраль КАМАК блок отдает сигнал X. Субадрес A и функция F не дешифрируются. Кроме того, блок выдает на шину R1 сигнал N & M(t), где M(t) – меандр, равный периоду выходного сигнала. Читая бит R1, можно синхронизировать момент начала регистрации с началом или серединой периода.

"Коммутатор и АЦП-10" предназначен для регистрации аналогового сигнала в цифровом виде. Входной коммутатор позволяет последовательно измерять сигнал от 15 различных источников (в данной работе не используется).

#### Технические данные

- разрядность АЦП ..... 10 дв. разряд;
- количество каналов ..... 16;
- (15 информационных, канал 0 – контроль)



- величина кванта преобразования.....10 мВ;
- время преобразования ..... 35 мкс;
- входное сопротивление..... 10 МОм;
- диапазон измеряемых напряжений .....±5.12 В.

Блок имеет два режима работы.

1. Режим преобразования по одному выбранному каналу. A1-F16 W1-W4 устанавливает номер выбранного канала. Нулевой канал предназначен для контроля смещения и постоянно присоединен к 0 вольт. Запуск по A0-F25, A0-F2 или внешним сигналом "START" с передней панели. По концу цикла преобразования модуль формирует сигнал "END" на переднюю панель и запрос, который можно проверить командой F8.

2. Режим циклического опроса каналов (в дочерней работе не используются). По команде A1-F17 (W1-W15) заносятся номера спрашиваемых каналов. Поиск очередного выбранного канала и последующий запуск производится командой A1-F25 или "SEARCH" с передней панели. По концу преобразования выдается LAM-запрос и сигнал "END" на переднюю панель.

**Формат данных.** Результат представлен в дополнительном коде: R10 – знаковый разряд, R1 – младший, если R10 = 0, то результат положительный. Соответствие кода: #777 = 5.11V; #1 = 10mV; #1777 = -10mV; #1000 = -5.12V. Символ # означает восьмеричную запись числа.

#### КАМАК-функции блока "Коммутатор и АЦП"

- A0-F0 чтение данных: в R1-R10 результат, в R17-R20 содержимое регистра кода номера канала, сброс LAM-запроса;
- A0-F2 аналогично A0-F0, но с последующим запуском АЦП;
- A0-F8 проверка LAM-запроса, Q=1, если есть запрос и он разрешен командой A0-F26;
- A0-F10 сброс запроса;
- A1-F16 запись номера канала (W1-W4);
- A1-F17 запись регистра масок каналов (W1-W15) (используется при циклическом переборе каналов);
- A0-F24 запрет LAM-запроса;

- A0-F25 запуск цикла преобразования по установленному каналу 'v';
- A1-F25 поиск следующего канала, разрешенного регистром масок и запуск цикла преобразования;
- A0-F26 разрешение LAM-запроса;
- LAM=1, если цикл преобразования закончен и запрос разрешен;
- Zero установка всех регистров, кроме регистра масок каналов, в состояние 0;
- Clear то же, что и Zero;
- Inhibit не используется.

#### Работа с блоком "Коммутатор и АЦП-10"

Подать на вход блока сигнал с выхода генератора "Сигнал+шум". На вход "START" подать синхронные импульсы высокой частоты (это будет частота дискретизации при синхронном режиме, равная  $1/28/T$ , T – период сигнала)

Для одноканального режима работы в программе возможна следующая последовательность действий:

- снять запрет LAM (A0-F26);
  - установить режим преобразования по одному каналу и указать номер канала, например, первый: (A1-F16 W=1);
  - читать бит R1 по адресу генератора "Сигнал+шум" подождать изменения его состояния для привязки начала регистрации к определенной фазе сигнала;
  - по флагу готовности LAM (A0-F8) читать регистр данных АЦП (A0-F0) столько раз, сколько нужно для заполнения массива.
- Заметим, что при несинхронной записи, когда частота дискретизации задается, например, программной задержкой, следует один раз запустить АЦП командой (A0-F25), а потом читать по (A0-F2) с предварительной проверкой LAM (A0-F8);
- нормализовать формат данных.

Блок "АЦП-10ISK" предназначен для регистрации аналогового сигнала в цифровом виде. Входной коммутатор позволяет последовательно измерять сигнал по четырём каналам (в данной работе не используется).

#### Технические данные

- разрядность АЦП ..... 10 дв.разр.:

- количество каналов ..... 4
- пределы измерения : .....  $\pm 1.28, \pm 2.56, \pm 5.12, \pm 10.24$  В,
- время преобразования ..... 1 мксек;
- входное сопротивление ..... 100 кОм;
- емкость памяти ..... 4096 слов;
- собственные шумы (макс.) .....  $\pm 2$  кванта;
- габариты ..... 2М.

### Статусный регистр (A1·F16)

Режим работы задается статусным регистром, состоящим из 5 разрядов (5 младших разрядов в слове данных). Биты имеют следующие значения:

- 1 - "Обращение ЭВМ к ОЗУ" - сброс триггера "SINGLE", прерывание записи от АЦП, запрет запусков.
- 2 - "Запрет LAM".
- 3 - "SINGLE" - режим однократного запуска с автоблокировкой повторных пусков. Снимается установкой и сбросом бита 1.
- 4 - "Страничная запись".
- 5 - "Самописец" - после запуска данные циклически пишутся в ОЗУ. Запись прерывается импульсом "СТОП" или по N.A5.F16 W=1.

### Регистр пределов (A3·F16)

12	11	10	9	8	7	6	5	4	3	2	1
0	*	*	1	0	0	*	Период дискретизации				

Регистр пределов состоит из 12 разрядов. Младшие 5 разрядов задают период дискретизации. Разряды 6, 10, 11 - резервные. Разряды 9,8,7 необходимо загрузить 100. Таким образом, в регистр пределов нужно писать #4XX, где XX - восьмеричное число в соответствии с таблицей:

#04 - 1 мкс	#05 - 10 мкс	#06 - 100 мкс	#07 - 1 мс
#14 - 2 мкс	#15 - 20 мкс	#16 - 200 мкс	#17 - 2 мс
#24 - 4 мкс	#25 - 40 мкс	#26 - 400 мкс	#27 - такт от ЭМ
#34 - 5 мкс	#35 - 50 мкс	#36 - 500 мкс	#37 гаймер.

Бит 12 равен нулю, если пределы выбраны правильно.

### Регистр коммутатора (A7·F16)

12	11	10	9	8	7	6	5	4	3	2	1
*	Режим			Канал 3		Канал 2		Канал 1		Канал 0	
0	№ канала			По всем каналам:							
1	Сканир.			0 - 1.28В; 1 - 2.56В; 2 - 5.12В; 3 - 10.24В							

Регистр коммутатора состоит из 12 разрядов. Младшие 8 разрядов задают диапазон измерения АЦП по амплитуде (по два бита на канал). Разряды 9-11 определяют режим работы коммутатора. Разряд 12 - резервный.

Диапазоны по амплитуде:      00 - 1.28В      01 - 2.56В  
[биты (1,2); (3,4); ...]      10 - 5.12В      11 - 10.24В

При W11=0 режим работы одноканальный. Номер канала определяется битами 10, 9:

00 - канал 0	01 - канал 1
10 - канал 2	11 - канал 3

При W11=1 многоканальный режим. Номера и количество рабочих каналов определяется теми же битами 10, 9:

- 00 - сканирование по каналам 0-3 (4 канала);
- 01 - сканирование по каналам 0-1 (2 канала);
- 10 - сканирование по каналам 2-3 (2 канала);
- 11 - сканирование по каналам 0-3 (4 канала);

Данные укладываются в память следующим образом:

4 канала	2 канала
0 кан. → 0, 4, 8, ...	0 кан. → 0, 2, 4, ...
1 кан. → 1, 5, 9, ...	1 кан. → 1, 3, 5, ...
2 кан. → 2, 6, 10, ...	2 кан. → 0, 2, 4, ...
3 кан. → 3, 7, 11, ...	3 кан. → 1, 3, 5, ...

Формат данных. Прибор выдает 12-разрядный код, из которых значащими являются 10 разрядов R3...R12. Код 0 соответствует  $-U_{\max}$ , код 4095 соответствует  $+U_{\max}$ . Таким образом, входное напряжение в вольтах можно вычислять по формуле:  $U = K \cdot (D - 2048)$ , где  $K = (\text{Предел изм.} / 2048)$ , D - прочитанный код. Заметим, что разрешающая способность

прибора (квант преобразования) равна 4. Такой формат данных сделан для совместимости с 12-разрядным прибором.

### КАМАК-функции блока "АЦП-101SK"

- / 0·F0 – чтение памяти;
  - A1·F0 – чтение статусного регистра;
  - A2·F0 – чтение текущего адреса памяти;
  - A3·F0 – чтение регистра пределов;
  - A4·F0 – чтение дополнит. информации и типа модуля;
  - A7·F0 – чтение регистра коммутатора;
  - A0·F8 – проверка LAM;
  - A0·F10 – сброс LAM;
  - A0·F16 – запись в память;
  - A1·F16 – запись в статусный регистр;
  - A2·F16 – запись в адресный счетчик;
  - A3·F16 – запись в регистр пределов;
  - A5·F16 – запуск прибора;
  - A6·F16 – запись 1 слова в ОЗУ от АЦП по команде ЭВМ;
  - A7·F16 – запись в регистр коммутатора.
- При F8, F10 Q=L; для всех остальных функций Q=0.

### Работа с прибором "АЦП-101SK"

Подать на один из входов блока сигнал с выхода генератора "Сигнал+шум". На вход "ТАЙМЕР" подать синхронимпульсы высокой частоты, если АЦП будет использоваться в синхронном режиме.

Последовательность команд в подпрограмме записи реализации может быть, например, такой:

- сбросить LAM (A0·F10);
- установить период дискретизации (A3·F16) с проверкой правильности установки (бит 12);
- загрузить регистр коммутатора (A7·F16);
- установить нулевой начальный адрес памяти (A2·F16 W=0);
- загрузить в статусный регистр #1 (A1·F16 W=1) – режим ОЗУ; команда нужна для установки и сброса бита 1, что разрешает последующий запуск в режиме "SINGLE";

- установить статусном регистре режим "SINGLE" (A1·F16 W=4);
- читать бит R1 по адресу генератора "Сигнал+шум", подождать начала периода сигнала;
- стартовать АЦП (A5·F16);
- подождать запрос LAM;
- установить статусном регистре режим "Обращение ЭВМ к ОЗУ" (A1·F16 W=1);
- установить начальный адрес памяти (A2·F16 W=0);
- читать память (A0·F0) столько раз, каков размер буфера для данных, но не более 4096;
- нормализовать формат данных.

### ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

При выполнении заданий следует использовать подпрограммы-функции, специально разработанные для данной лабораторной работы. Эти подпрограммы включены в библиотеку TSANLIB.

Подпрограмма быстрого преобразования Фурье

```
void fft(int m, float *ar, float *ai);
```

m – логарифм по основанию 2 размера обрабатываемого массива ( $2^m$  – количество отсчетов);

\*ar – указатель на буфер (массив) для реальной части спектра;

\*ai – указатель на буфер для мнимой части спектра;

Подпрограмма fft использует алгоритм быстрого преобразования Фурье. На входе подпрограммы в массиве ar должен быть оригинал реализации, в ai – нули, на выходе в ar и ai соответственно реальная и мнимая часть спектра. Таким образом, исходную реализацию, если не хотите ее терять, следует хранить в другом массиве, а при вызове fft переписывать данные в ar.

Подпрограмма обратного преобразования Фурье

```
void inv_fft(int m, float *ar, float *ai)
```

Параметры те же, что и в fft. Оригинал возвращает и через ar.

Подпрограмма визуализации

```
char show_diagram(float *buff, int N, char *mes)
```

\*buff – указатель на массив данных;

N – число элементов в массиве;

\*mes – текст к картинке.

Просмотр картинки завершается по "Esc" или "Enter". Подпрограмма возвращает символ завершения, который можно использовать по усмотрению, например, так:

```
if(show_diagram(am, N, "Спектр") == ' ') exit(0);
```

Подпрограмма сама масштабирует картинку. По оси X – номер элемента массива, по оси Y – значение. Правым и левым курсором можно выбирать масштаб по X. "PgUp", "PgDn" смещают картинку вправо или влево на 1/8 видимой части изображения. "Home", "End" сдвигают указатели соответственно в начало и конец, не изменяя масштаба. <F1> – подсказка.

Подпрограмма измерения периода

```
float getper(float period, float *buff, int N)
```

period – значение периода на 1 моде;

\*buff – указатель на массив данных;

N – число элементов массива.

Подпрограмма предназначена для измерения периода сигнала. Управление метками периода производится с помощью "Left", "Right". Клавиши "Up", "Down" изменяют дискретность перемещения меток в 10 раз. "PgUp", "PgDn" смещают картинку вправо или влево на 1/8 видимой части изображения. "Home" "End" перемещают окно в начало и конец, "Space" – обновление картинка, "Enter" – выход. Подпрограмма возвращает значение периода.

Пример: `period = getper(period, buff, NS);`

Рекомендации по составлению программы

В данной работе вам предлагается написать программу регистрации и цифровой обработки сигнала различными методами. Поскольку объем работы достаточно велик, с самого начала следует тщательно продумать структуру программы с той целью, чтобы при дополнениях максимально использовать уже написанную и отлаженную часть. Блок-схема программы может выглядеть, например, следующим образом:

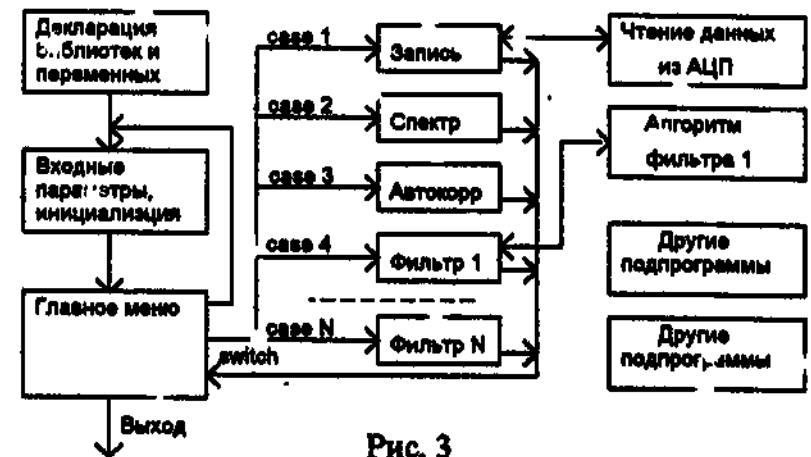


Рис. 3

Динамическое выделение памяти. Отлаживать программу удобнее при небольших массивах при N=256 или N 512, а результат работы фильтра тем эффективнее, чем больше размер массива. Поэтому вам понадобится динамическое отведение памяти под массивы. Это можно сделать, например, следующим образом:

```
static int *ibuf; /* Входной сигнал */
static float *srr; /* Реальная часть спектра */
static float *spi; /* Мнимая часть спектра */
static float *spa; /* Амплитудный спектр */
static float *spf; /* Фазовый спектр */
static float *abus; /* Буферный массив */
int lng, NS /* lng – двоичн. логарифм размера массива, NS=2^lng */
...; ...;
void main() { ...; ...; ...;
free(ibuf); free(srr); free(spi); free(spa); free(spf); free(abus);
printf("\n Free memory: %6lu bytes", sizeof(int));
if(inimem(lng)==0) { printf("\nМАЛО ПАМЯТИ!");
getch(); goto Begin; } /*Begin – метка в начале программы */
printf("\n Rest: %6lu bytes", sizeof(float));
NS = 1 << lng;
for(i=0; i<NS; i++) /* Чистим массивы */
```

```

ibuff[i]=spr[i]=spi[i]=spa[i]=spf[i]=0.0;
...; ...; ...} /* END MAIN */

```

```

int inmem(int lng)
{ int r; r = 1;
ibuf = malloc( (1 << lng) * 2); if(ibuf== 0) r = 0;
spr = malloc( (1 << lng) * 4); if(spr == 0) r = 0;
spi = malloc( (1 << lng) * 4); if(spi == 0) r = 0;
spa = malloc( (1 << lng) * 4); if(spa == 0) r = 0;
spf = malloc( (1 << lng) * 4); if(spf == 0) r = 0;
abuf = malloc( (1 << lng) * 4); if(abuf== 0) r = 0;
return r;
}/* END INMEM*/

```

Замечание: некорректное применение функции free(ptr) может привести к непредсказуемым последствиям – зависанию компьютера, неправильной работе системы и т.д., причем эти ошибки, как правило, трудно локализовать.

Работа с аппаратурой. Внимательно прочитайте описание прибор. В самом начале программы main() следует дать команды общей инициализации системы:

```

cam_crate(ncrate); cam_z(); delay(1);
cam_off_(); delay(1); cam_c(); delay(1);

```

Члены массива данных из АЦП лучше поместить в подпрограмму, куда передаются указатель типа int для приема данных, количество отчетов, период дискретизации и предел измерения (для АЦП-101SK). Чтобы не нагромождать много параметров, номер крейта и позиции модулей можно сделать глобальными.

Пример описания функции чтения данных:

```

int read_adc(int *bufi, int NS, int tim, int mod)
/* Считывает данные из АЦП-101SK */
/* bufi – Буфер данных */
/* NS – Количество точек */
/* tim – Период дискретизации */
/* mod – Предел измерения */

```

Для того, чтобы каждая картинка была повторяющейся, начав регистрации надо привязать к определенному моменту периода, читая первый бит по адресу генератора "Сигнал+шум". Первую половину периода бит R1 находится в состоянии "1", вторую – в "0". По моменту перепада R1 из одного состояния в другое можно начинать запись массива.

Организация меню. Основное меню программы можно устроить достаточно просто с помощью оператора switch(select) { case 1:{...} ... case n {...} }. Сначала распечатайте назначение ключей select, затем запросите ввод select. Стандартная функция форматного ввода scanf в данном случае весьма неудобна, главным образом, потому, что требует полного ввода числа. Советуем использовать для организации меню функции ini, ino, inx, inf, ing и другие, описанные в файле: D:\TC\INCLUDE\tansi.h и в кратких справках.

Пример меню:

```

M0: clrscr();
if(ini("Номер крейта ", &ncrate)=='q') exit(0);
if(ini("\nПозиция АЦП ", &adc)=='q') goto M0;
M1:
printf("\n 1 – Регистрация входного сигнала.");
printf("\n 2 – Входной сигнал на экран.");
printf("\n 3 – Вычисление спектра.");
printf("\n 4 – Вычисление функции автокорреляции.");
...
printf("\n 8 – Восстановление сигнала по спектру.");
printf("\n 9 – Восстановленный сигнал на экран.");
if(ini("\n Что делать? ", &select)=='q') goto M0;
switch (select)
{
case 1: {...; ...; ...; ...; break; }
...; ...; ...;
case 9: {...; ...; ...; ...; break; }
}
closegraph(); restorecrmode(); goto M1

```

Чтение файлов с диска. Для реализации согласованного фильтра и фильтра Винера вам в соответствии с (15) понадобится образец спектра чистого сигнала. Этот спектр можно получить двумя способами. 1) Записать сигнал при отключенном шуме, применить преобразование Фурье и отнормировать полученный спектр. 2) Можно также использовать готовые спектры чистых сигналов, которые содержатся в файлах S0, S1, ..., S15 в директории D:\LAB12. Если хотите научиться работать с файлами на диске, то используйте этот способ. При достаточном дисковом пространстве файлы, содержащие образцы спектров, лучше переписать в свою рабочую директорию В первой половине файла (5,2 байт) лежит амплитудный спектр, во второй – фазовый. Так как максимальный период сигнала при синхронном режиме записи равен 128 отсчетам, файлы содержат полную информацию о спектрах сигналов. Под образец спектра вам следует отвести два массива типа float размером по 128 элементов. Прочитать спектры можно, например, так:

```
char fn[16]; float asp[128], fsp[128]; int fid, err;
.....
case 12: { /* Согласованный фильтр */
printf("\n Name: "); gets(fn);
if((fn[0]!='q') && (fn[1]!='0')) break;

fid = open(fn,O_BINARY);
if(fid<0) { printf("ERR OPEN %s", fn), getch(); break; }
err = read(fid,asp,128);
if(err<0) { printf("ERR READ %s",fn), getch(); break; }
err = read(fid,isp,128*4);
if(err<0) { printf("ERR READ %s",fn), getch(); break; }
err = close(fid);
if(err<0) { printf("ERR CLOSE %s",fn), getch(); break; }
.....
} /* END CASE 12 */
```

Здесь две функции read последовательно считывают амплитудный и фазовый спектр в массивы asp и fsp.

При составлении алгоритма фильтрации нужно, разумеется, учитывать то обстоятельство, что для сигнала длительностью в  $2^N$

периодов  $i$ -й элемент образца спектра соответствует элементу номер  $i*2^N$  в спектре реализации.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Напишите основу программы – ввод параметров, динамическое выделение памяти, меню, подпрограмму работы с блоками КАМАК.

2. Регистрация сигнала. Напишите подпрограмму регистрации входного сигнала с помощью модуля "Коммутатор и АЦП-10" и получите картинку на экране. Запуск АЦП синхронизируйте с внутренней тактовой частотой генератора (синхронный режим). При таком режиме в реализации должно укладываться целое число периодов; равное  $2^{(m-1)}$ , где  $m = \log_2(N)$ .

Отключите шум. Переключая тумблеры выбора формы, просмотрите различные сигналы.

Включите шум. Меняя уровни сигнала и шума, получите входную реализацию при различных отношениях сигнал/шум.

Определите мощность сигнала и шума, найдите отношение сигнал/шум на входе для нескольких разных сигналов.

3. Научитесь получать спектры сигналов. При  $N = 256$ , когда алгоритм работает достаточно быстро, просмотрите спектры чистых сигналов. Включите шум. Сравните полученные спектры при различных  $N = 128, 512, 2048$ .

4. Фильтр нижних частот. Реализуйте идеальный фильтр нижних частот. Для этого достаточно в спектре обнулить все частотные составляющие с номерами  $K < i < N - K$ , где  $K$  – верхняя граничная частота фильтра. Напомним, что верхние доли в спектре сигнала фактически соответствуют низким частотам (см. переход от формулы (6) к (8)). Оцените результат фильтрации.

5. Метод накопления. Для реализации метода накопления отведите дополнительный массив типа float размером 128 элементов и сложите в него все периоды входного сигнала. Оцените увеличение отношения сигнал/шум. Объясните результат.

6. Гребенчатый фильтр. Получите спектр реализации. Оставьте в амплитудном спектре только гармоники, кратные  $N/T$ , где  $N$  – число отсчетов в реализации,  $T$  – период сигнала (для синхронного режима  $T=128$ ) с помощью обратного преобразования Фурье получите отфильтрованный сигнал

Сравните полученную картинку с методом накопления для одной и той же реализации. Выходные сигналы должны быть идентичны. Почему?

7. Автокорреляционная функция. Вычислите спектр мощности реализации. С помощью обратного преобразования Фурье получите автокорреляционную функцию. Сравните автокорреляционные функции чистых и "зашумленных" сигналов. Как по автокорреляционной функции вычислить отношение сигнал/шум?

8. Согласованный фильтр. Используя формулу (15а), напишите программу согласованного фильтра. После того, как согласованный фильтр заработает, сравните сигнал на его выходе с автокорреляционной функцией. Оцените увеличение отношения сигнал/шум.

Поскольку согласованные фильтры применяются обычно для обнаружения сигнала, постарайтесь оформить программу таким образом, чтобы она прямо сообщала, есть такой-то сигнал в смеси или нет. От чего зависит вероятность ошибочной ответа?

9. Фильтр Винера. Программная реализация этого фильтра подобна согласованному фильтру.

Сравните эффективность фильтра Винера с методом накопления по критерию среднеквадратичной ошибки при различных  $\gamma$  на входе.

10. Корреляционный фильтр. Реализуйте алгоритм вычисления функции взаимной корреляции между входной реализацией и периодическим импульсным сигналом. Форму импульса можно задать прямоугольной, хотя сглаженные фронты дают лучший результат... Программу составьте так, чтобы длительность импульса можно было изменять. Примените данный фильтр к разным сигналам. Оцените его эффективность.

#### Несинхронный сигнал.

11. Напишите подпрограмму регистрации сигнала с помощью аналого-цифрового преобразователя АЦП-101SK. Получите сигнал при разных частотах дискретизации.

12. Получите спектр несинхронного сигнала при отключенном шуме для различных частот дискретизации. Объясните структуру спектра.

13. Обнаружение периодического сигнала. Установите уровни сигнала и шума в максимальные положения. Получите функцию

автокорреляции для несинхронного режима записи. Объясните структуру функции автокорреляции. Измените период сигнала.

14. Метод накопления. Модифицируйте алгоритм метода накопления таким образом, чтобы его можно было применить к несинхронному сигналу.

15. Нелинейный фильтр. При сравнительно больших  $\gamma$ , когда шум невелик, к спектру сигнала можно применить следующее преобразование: затушить все частотные составляющие, амплитуда которых составляет, скажем, <5% от максимальной. Оцените искажение формы сигнала и среднеквадратичную ошибку при различных "уровнях обрезания".

16. Корреляционный фильтр. Реализуйте алгоритм вычисления функции взаимной корреляции для несинхронного режима записи сигнала подобно п.10.

**Примечание.** Жирным шрифтом отмечены обязательные задания.

#### ЛИТЕРАТУРА

1. Науман Г., Майлинг В., Щербина А. Стандартные интерфейсы для измерительной техники. М.: Мир, 1982.
2. Гоноровский И.С. Радиотехнические цепи и сигналы. М.: Радио и связь, 1986.
3. Мушер С.Л. Вычислительный эксперимент и обработка данных. Методические указания. Новосибирск: НГУ, 1991.
4. Бобровников Л.З. Радиотехника и электроника. М.: Недра, 1967.
5. Алиев Т.М., Тер-Хачатуров А.А. Измерительная техника. М.: Высш. шк., 1991.
6. Хемминг Р.В. Цифровые фильтры. М.: Недра, 1987.
7. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов. М.: Связь, 1979.
8. Введение в цифровую фильтрацию, под ред. Р. Баснера и А. Константидиса. М.: Мир, 1976.
9. Васильев Д.В., Витсень М.Р., Горшенков Ю.Н. и др. Радиотехнические цепи и сигналы. М.: Радио и связь, 1982.

## Содержание

Введение .....	3
Дискретизация сигнала .....	4
Дискретные спектры .....	6
Быстрое преобразование Фурье .....	7
Функция корреляции ... ..	8
Фильтры .....	9
Метод накопления .....	10
Согласованный фильтр .....	10
Корреляционный метод обнаружения периодических сигналов .....	12
Фильтр Винера .....	13
Взаимокорреляционный фильтр .....	14
Оценка эффективности фильтров .....	14
Описание установки и приборов .....	15
Программное обеспечение .....	21
Рекомендации по составлению программы .....	22
Практическая часть .....	27
Питерсгура .....	29

---

Подписано в печать 19 мая 1965 г.

Формат 60x84 1/16.

Печать офсетная. Уч.-изд. л. - 2

Тираж 100 экз.

Заказ № 252

---

Редакционно-издательский отдел Новосибирского университета.  
Участок оперативной полиграфии НГУ.  
630090, Новосибирск, 90, ул. Пирогова, 2.