

СОДЕРЖАНИЕ

Цель проекта	4
1. Структура АСНИ	4
1.1. Описание АСНИ	4
1.2. Методическое обеспечение	5
2. Техническое обеспечение	7
3. Разработка по АСНИ для подсистемы «Спектральный анализ». Среда разработки C++ Builder	8
3.1. Краткие сведения о среде разработки	8
3.2. Программирование процессов работы с файлами данных	13
4. Проектирование алгоритмов дискретного преобразования Фурье и оценивание спектральной плотности мощности	15
4.1. Определение дискретного преобразования Фурье	15
4.2. Определение спектральной плотности мощности	16
5. Задание на курсовое проектирование	18
5.1. Разработка технического обеспечения	18
5.2. Разработка программного обеспечения	18
6. Документация проекта	19
Библиографический список	20
Приложение А	21
Приложение Б	22
Приложение В	23

ЦЕЛЬ ПРОЕКТА

Целью курсового проекта является разработка технического и программного обеспечений автоматизированной системы научных исследований (АСНИ).

1. СТРУКТУРА АСНИ

1.1. Описание АСНИ

АСНИ предназначена для спектрального анализа данных, поступающих от первичных преобразователей физических величин, характеризующих некоторый технологический процесс. В состав АСНИ входят следующие подсистемы:

- подсистема измерений – информационно-измерительный канал (ИИК);
- подсистема передачи данных;
- подсистема обработки данных;
- подсистема визуализации и документирования результатов.

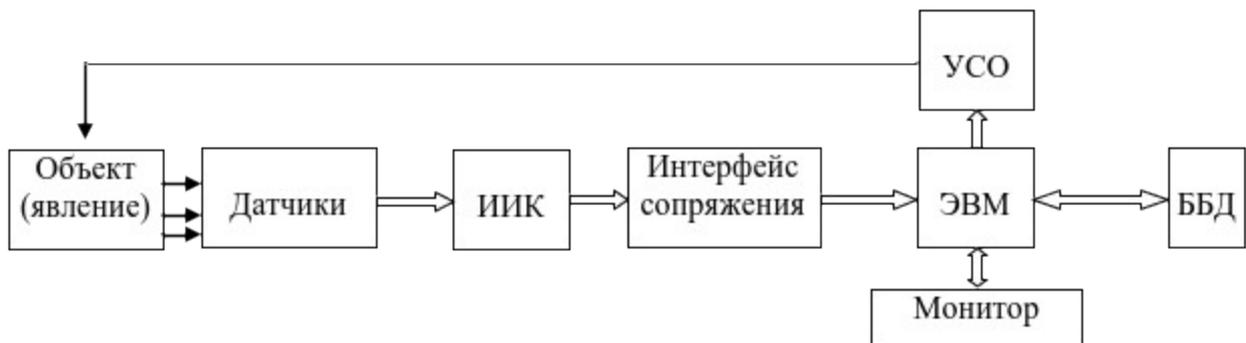


Рисунок 1.1 - Обобщенная структура АСНИ

На рисунке 1.1 приняты следующие обозначения:

ИИК - информационно-измерительный канал;

УСО – устройство связи с объектом;

БД - база данных.

Состав и структура ИИК приведены на рисунке 1.2.

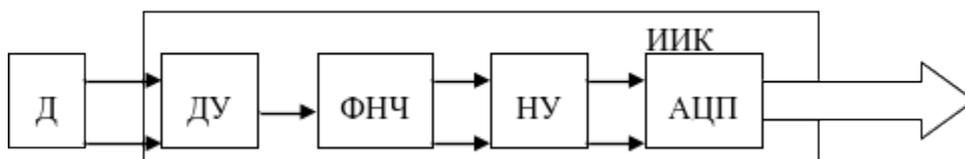


Рисунок 1.2 - Структура информационно-измерительного канала

На рисунке 1.2 приняты следующие обозначения:

Д - датчик;

ДУ – дифференциальный электронный усилитель;

ФНЧ - фильтр нижних частот;

НУ – нормирующий усилитель;

АЦП - аналого-цифровой преобразователь.

ДУ предназначен для усиления сигналов, поступающих от датчиков и подавления синфазных помех.

ФНЧ выполняет функции противомаскировочного фильтра и служит для подавления высокочастотных составляющих сигнала в целях исключения ошибок аналого-цифрового преобразования.

НУ предназначен для согласования входного напряжения АЦП с динамическим диапазоном изменения преобразуемого аналогового сигнала.

1.2. Методическое обеспечение

Одной из важнейших задач, решаемых автоматизированными системами, является сбор и обработка данных, поступающих от первичных преобразователей (датчиков), установленных на объектах автоматизации. Эти данные рассматривают как временные ряды. Временной ряд - это множество наблюдений, генерируемых последовательно во времени. В зависимости от того, как изменяется время: непрерывно или дискретно, различают временные ряды непрерывные и дискретные.

Современные автоматизированные системы обрабатывают данные с помощью компьютеров, поэтому все данные, которые поступают в виде аналоговых сигналов преобразуются в цифровую форму.

При исследовании процесса аналого-цифрового преобразования будут рассматриваться следующие временные ряды:

$X(t)$ – исходная физическая величина (непрерывный ряд);

$x(t)$ – выходной сигнал датчика (в вольтах), соответствующий функции $X(t)$ (непрерывный временной ряд);

$c(t)$ – выходной сигнал датчика $x(t)$, переведенный в непрерывные отсчеты (фиктивный непрерывный временной ряд);

$c(iT)$ – выходной сигнал датчика $x(t)$, переведенный в непрерывные отсчеты - выборки, выполненные в дискретные моменты времени с периодом T (дискретный временной ряд);

$c(i)$ – выходной сигнал датчика $x(t)$, переведенный в квантованные отсчеты, полученные после операции квантования (дискретный временной ряд);

$e(i)$ – ошибка, равная $c(iT) - c(i)$.

Фиктивный временной ряд $c(t)$ введен здесь для удобства формализации описания процесса аналого-цифрового преобразования и последующего программирования соответствующего алгоритма работы АЦП. Как временной ряд $c(t)$, так и ряд $c(i)$ измеряются в одних единицах - отсчетах. При этом элементы ряда $c(t)$ могут принимать как целые, так и дробные значения, а элементы ряда $c(i)$ принимают только целые значения.

В роли объектов исследования в данной работе выступают детерминированные полигармонические сигналы и сигналы случайной

природы. Структура полигармонического сигнала $x(t)$ (непрерывного временного ряда) определена следующим соотношением

$$x(t) = \sum_{k=1}^M a_k \sin 2\pi f_k t, \quad (1.1)$$

где a_k – амплитуда и f_k – частота k -ой гармоники; M – количество гармонических составляющих.

Непрерывный временной ряд $C(t)$ можно представить как результат линейного преобразования функции $x(t)$ в виде

$$C(t) = Ax(t) + B. \quad (1.2)$$

Значения дискретного временного ряда $C(iT)$ и соответствующие значения дискретного временного ряда $c(i)$, которые получаются на выходе АЦП, можно связать с помощью следующего соотношения

$$c(i) = \text{int}[C(iT) + 0,5]. \quad (1.3)$$

В соотношении (1.3) выражение в квадратных скобках принимает целое значение в результате округления, выполняемого всегда в меньшую сторону.

Добавление к значениям временного ряда $C(iT)$ величины 0,5 обеспечивает повышение точности округления.

Ошибка квантования $e(i)$ определяется следующим соотношением

$$e(i) = C(iT) - c(i). \quad (1.4)$$

Если модель АЦП работает правильно, то значения $e(i)$, будут ограничены интервалом $(-0,5; 0,5)$.

Последнюю формулу (1.4) можно переписать в виде

$$c(i) = C(iT) + e(i). \quad (1.5)$$

В этом случае можно считать, что временной ряд $c(i)$ формируется как сумма соответствующих элементов временного ряда $C(iT)$ и ошибки квантования, которую часто называют шумом квантования.

При идеальном преобразовании ошибка квантования распределена равномерно со стандартным отклонением $\sim 0,29 \Delta x$, где Δx - шаг квантования.

Основными формулами статической обработки временного ряда ошибки квантования являются

$$m_e = \frac{1}{N} \sum_{i=0}^{N-1} e(i),$$

$$\sigma_e^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (e(i) - m_e)^2,$$

$$\sigma_e = \sqrt{\sigma_e^2},$$

где m_e - математическое ожидание; σ_e^2 - дисперсия; σ_e - среднеквадратическое отклонение; N - разрядность временного ряда.

2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Информационно-измерительный канал системы автоматического управления (рисунок 1.2) состоит из следующих звеньев: датчик, предварительный усилитель, фильтр нижних частот, нормирующий усилитель и АЦП.

Пример расчета информационно-измерительного канала.

Исходные данные для расчета:

- внутреннее сопротивление датчика $R_g = 220 \text{ Ом}$;
- выходное напряжение датчика $U_g = 2,0 \text{ мВ}$;
- эффективное значение синфазной помехи, наводимой в витой паре проводов, соединяющих датчик с усилителем, $U_{сф} = 1,2 \text{ В}$;
- максимальная погрешность от синфазной помехи $\delta = 3\%$;
- АЦП работает с частотой $f_{\text{АЦП}} = 2000 \text{ Гц}$.

Нужно найти:

- схемотехническое решение;
- необходимый КОСС (коэффициент ослабления синфазного сигнала);
- минимальный коэффициент усиления дифференциального сигнала $K_{\text{диф}}$;
- величины сопротивлений резисторов;
- подходящий тип операционного усилителя.

Расчет предварительного усилителя

Так как датчик подключается к ПУ посредством двухпроводной линии, в которой наводится синфазная помеха, то ПУ нужно выполнить по схеме дифференциального усилителя, обеспечивающего необходимое подавление синфазной помехи. Эти схемы более удобны для использования в измерительных системах. Коэффициенты ослабления синфазного сигнала, которые можно получить в этих схемах, в основном зависят от коэффициентов ослабления синфазного сигнала операционных усилителей, а также от $K_{\text{диф}}$ и $K_{\text{синф}}$. На коэффициент усиления синфазного сигнала влияет главным образом точность изготовления употребляемых в схемах резисторов, определяющие действие усилителя. На рис 2.1 приведена схема однокаскадного дифференциального усилителя.

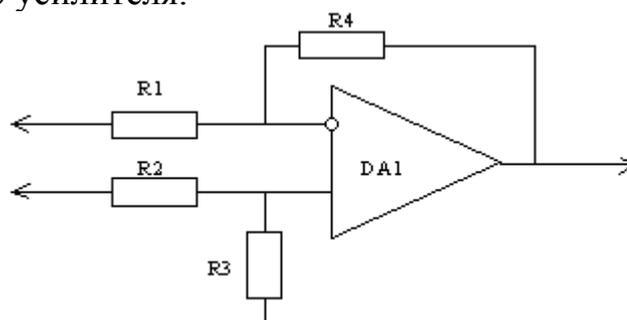


Рисунок 2.1 – Однокаскадный дифференциальный усилитель

Требуемый коэффициент ослабления синфазной помехи вычисляется по формуле

$$\text{КОСС} = \frac{U_{сф}}{U_g} \cdot \frac{1}{\delta} = \frac{1,2}{0,002} \cdot \frac{1}{0,03} = 20000 \text{ (86дБ)}.$$

Минимально допустимый коэффициент усиления дифференциального сигнала находится в соответствии с методикой [1] по формуле

$$K_{диф} = КОСС \cdot K_{сиф}.$$

Расчет активного фильтра нижних частот (НЧ)

Считаем, что рабочая полоса ОУ достаточно широкая, поэтому частоту среза находим в соответствии с теоремой Шеннона - Котельникова

$$f_c \leq \frac{1}{2} f_{Ашн} = 1000 \text{ Гц}.$$

ФНЧ и УН можно выполнить в виде активного фильтра, собранного на операционном усилителе. В качестве активного фильтра выбираем схему фильтра Саллена-Ки, изображенную на рис 2.2.

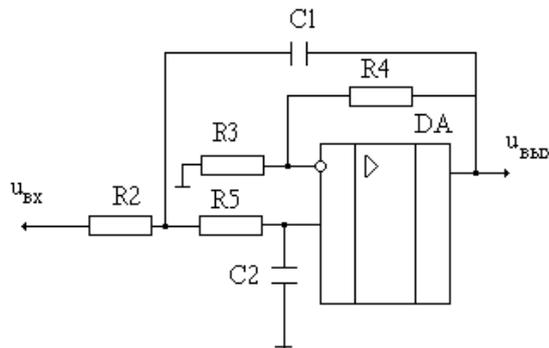


Рисунок 2.2 – Схема активного НЧ фильтра

В соответствии с методикой [2] сначала находится величина емкости конденсатора C_1 :

$$C_1 = \frac{10}{f_c} = \frac{10}{1} = 10 \text{ нФ} = 0,01 \text{ мкФ}.$$

3. РАЗРАБОТКА ПО АСНИ ПОДСИСТЕМЫ «СПЕКТРАЛЬНЫЙ АНАЛИЗ». СРЕДА РАЗРАБОТКИ MatLab

3.1. Краткие сведения о среде разработки

Программа MATLAB представляет собой высокоуровневый технический вычислительный язык и интерактивную среду для разработки алгоритмов, визуализации и анализа данных, числовых расчетов.

MATLAB сокращенно от Matrix Laboratory, все данные в программе представлены в виде матриц. После запуска MATLAB открываются несколько окон в зависимости от их раскладки и версии MATLAB, основные из них это:

- **Workspace**: рабочее пространство. В нем отображаются переменные, с которыми вы работаете в данный момент.

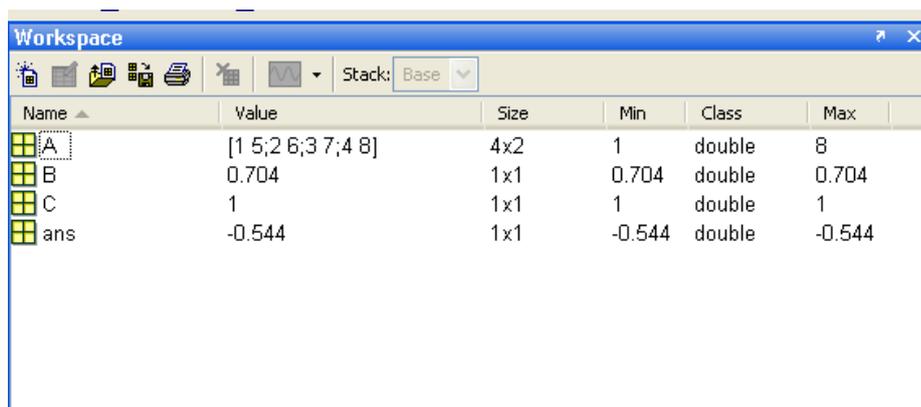


Рисунок 3.1 – Окно рабочего пространства

- **Current Directory:** рабочая папка, MATLAB работает с файлами из его папки а также с файлами из рабочей папки, можно добавить свои папки из меню File->Set Path.

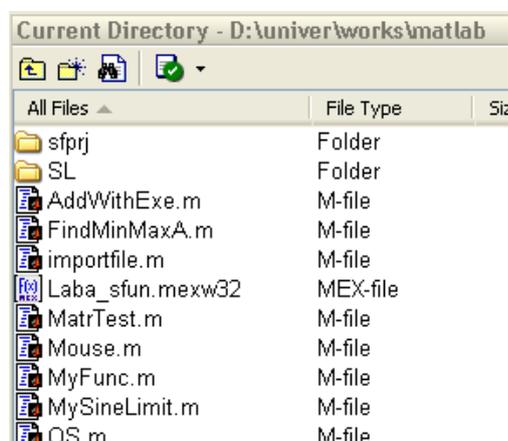


Рисунок 3.2 – Окно рабочей папки

- **Command History:** история использованных команд. Любую команду можно вызвать еще раз двойным щелчком мыши на ней.

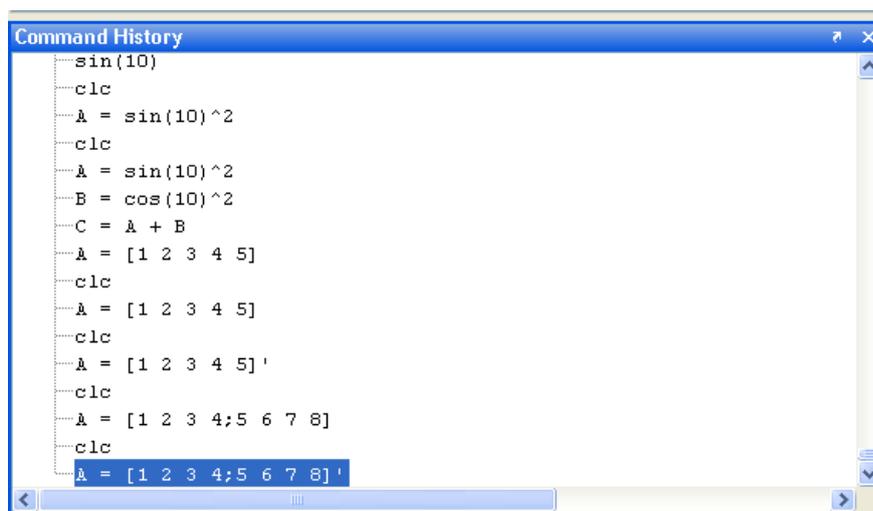


Рисунок 3.3 – Окно истории использования команд

- **Command Window:** окно команд, является основным для работы с MATLAB. Команды вводятся после знака “>>”



Рисунок 3.4 – Командное окно

Для примера введем $\sin(10)$

>> $\sin(10)$

Результат:

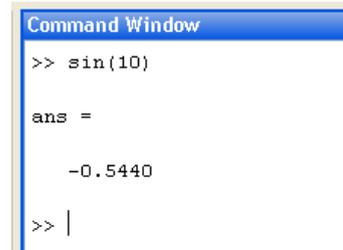


Рисунок 3.5 – Синтаксис командного окна

ans – сокр. от Answer (ответ) автоматически создаваемая переменная, которая содержит результат вычислений. Переменная ans также появляется в окне Workspace, где можно увидеть ее значение, размерность и т.д. очистить Workspace можно командой clear. Для удаления одной или нескольких переменных надо указать их имена после команды clear (например clear a b c). Для очистки командного окна используется команда clc (не влияет на результаты). Для присвоения переменной определенного значения используется обычный знак равенства “=”. Пример.

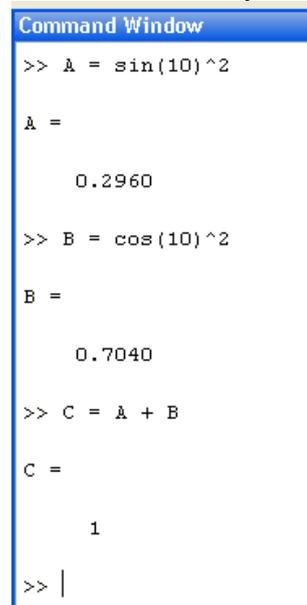


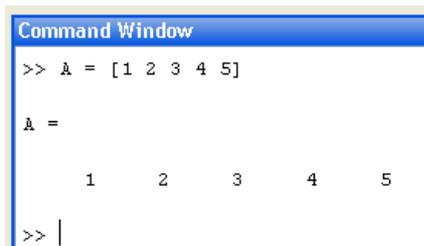
Рисунок 3.6 – Командное окно

Можно заметить что переменная **ans** не создалась поскольку результаты вычислений сохранились в переменных A B C

Для создания вектора строки используется квадратные скобки [] с указанием значений через пробел.

>> A = [1 2 3 4 5]

Результат:



```
Command Window
>> A = [1 2 3 4 5]

A =

     1     2     3     4     5

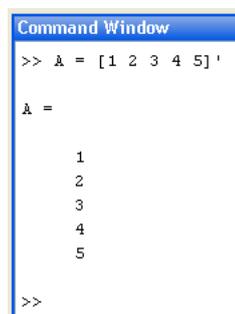
>> |
```

Рисунок 3.7 – Командное окно

Для создания вектора столбца используется квадратные скобки [] с указанием значений через пробел а в конце добавляется

'>> A = [1 2 3 4 5]'

Результат:



```
Command Window
>> A = [1 2 3 4 5]

A =

     1
     2
     3
     4
     5

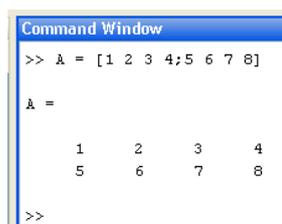
>>
```

Рисунок 3.8 – Командное окно

Для создания матрицы используется квадратные скобки [] с указанием значений строк через пробел, а разделителем строка является точка с запятой“;”

'>> A = [1 2 3 4; 5 6 7 8]'

Результат:



```
Command Window
>> A = [1 2 3 4; 5 6 7 8]

A =

     1     2     3     4
     5     6     7     8

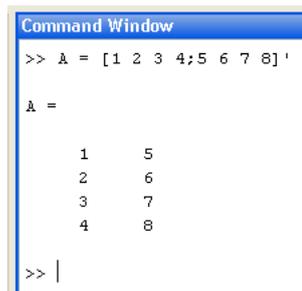
>>
```

Рисунок 3.9 – Командное окно

Для транспонирования матрицы так же в конце надо добавить ‘

'>> A = [1 2 3 4; 5 6 7 8]'

Результат:



```
Command Window
>> A = [1 2 3 4; 5 6 7 8]'

A =

     1     5
     2     6
     3     7
     4     8

>> |
```

Рисунок 3.10 – Командное окно

Для того чтобы MATLAB не выводил каждый раз значение переменной после ее ввода, надо завершать каждую команду “;”.

Можно писать несколько команд в одной строке разделяя их “;”.

```
>> A = 5;B = 6;C = 7;
```

Для создания массива чисел с фиксированным шагом используется двоеточие

```
>> A = 1:0.1:5;
```

Результатом будет массив от 1 до 5 с шагом 0,1

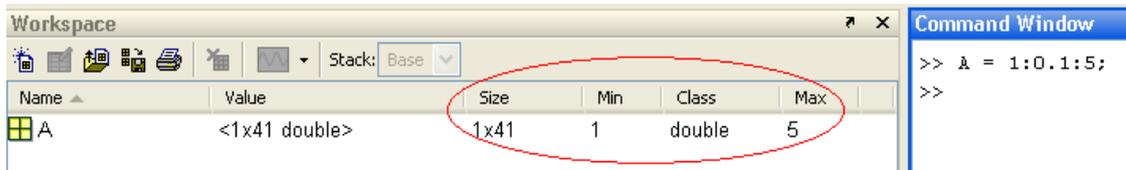


Рисунок 3.11 – Окно рабочего пространства

M-Файлы это обычные текстовые файлы с расширением *.m содержащие команды на языке MATLAB. M-Файлы бывают двух типов M-Файлы сценарии и M-Файлы функции. M-Файлы сценарии не имеют входных и выходных параметров. Работают с переменными из Workspace.

Также переменные, создаваемые в M-Файлах сценариях после выполнения M-Файла остаются в WorkSpace. Удобны для сохранения часто повторяющихся последовательных команд.

Пример M-Файла сценария “Sample.m”

File->New->M-File

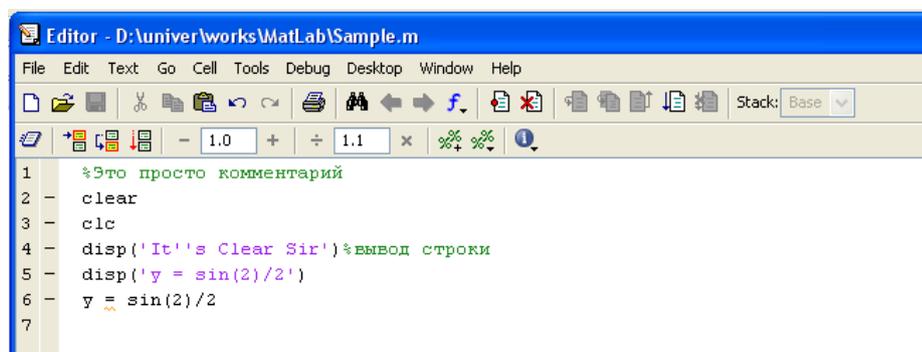


Рисунок 3.12 – Окно M-файла

Код сценария (или скрипта) надо сохранить в папку Current Directory с именем “Sample.m”.

Для выполнения сценария введем команду Sample (MATLAB чувствителен к регистру). Результат:

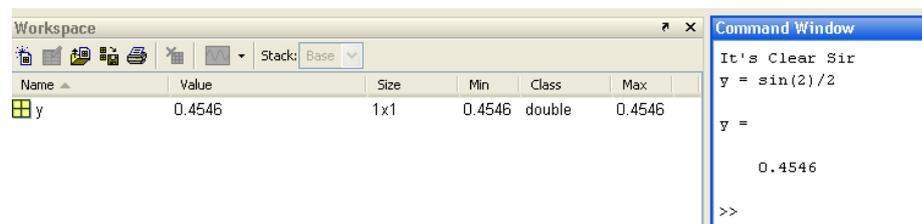


Рисунок 3.13 – Окно рабочего пространства

Обратите внимание, что в Workspace осталась переменная “y”.

M-Файлы функции имеют входные и выходные параметры. Не могут работать с переменными из Workspace. Переменные создаваемые внутри M-Файла функции уничтожаются после исполнения. Создаются с помощью ключевого слова function.

Необходимо загрузить функцию с файлом в Current Directory.

ВАЖНО: Имя M-Файла функции должно совпадать с именем функции.

Пример M-Файла функции “Sample2.m”

```

1 function [Y, Levaia_Peremennaia] = Sample2( x, y )
2 - Y(1) = x + y;
3 - Y(2) = x - y;
4 - Y(3) = sin(x);
5 - Y(4) = sin(y);
6 - Levaia_Peremennaia = 123123;
7 %тоже самое можно записать так
8 %Y = [ (x + y) (x - y) sin(x) sin(y) ];
9
10
11

```

Рисунок 3.14 – Окно M-файла

В строке команд введем

[A B] = Sample2(1, 2)

Результат:

```

>> [A B] = Sample2(1, 2)

A =

    3.0000   -1.0000    0.8415    0.9093

B =

    123123

>>

```

Рисунок 3.15 – Командное окно

Окно Workspace после выполнения функции Sample2

Name	Value	Size	Min	Class	Max
A	[3 -1 0.8415 0.9093]	1x4	-1	double	3
B	123123	1x1	123123	double	123123

Рисунок 3.16 – Окно рабочего пространства

3.2. Программирование процессов работы с файлами данных

M-файл-функция является типичным объектом языка программирования системы MATLAB. Одновременно он является полноценным модулем с точки

зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. Структура такого модуля с одним выходным параметром выглядит следующим образом:

```
function var=f_name(Список_параметров)
%Основной комментарий
%Dополнительный комментарий
Тело файла с любыми выражениями
var=выражение
```

М-файл-функция имеет следующие свойства: он начинается с объявления `function`, после которого указывается имя переменной `var` — выходного параметра, имя самой функции и список ее входных параметров; функция возвращает свое значение и может использоваться в виде `name(Список_параметров)` в математических выражениях; все переменные, имеющиеся в теле файла-функции, являются локальными, т.е. действуют только в пределах тела функции; файл-функция является самостоятельным программным модулем, который общается с другими модулями через свои входные и выходные параметры; правила вывода комментариев те же, что у файлов-сценариев; файл-функция служит средством расширения системы MATLAB; при обнаружении файла-функции он компилируется и затем исполняется, а созданные машинные коды хранятся в рабочей области системы MATLAB.

Последняя конструкция `var=выражение` вводится, если требуется, чтобы функция возвращала результат вычислений.

Приведенная форма файла-функции характерна для функции с одним выходным параметром.

В пакете MATLAB используются следующие команды для работы с файлами:

- `fopen(filename, permission)` - открывает файл с именем `filename` и параметром, определенным в `permission` и возвращает идентификатор `fid` со значением: 0 - чтение с клавиатуры (`permission` установлено в 'r'); 1 - вывод на дисплей (`permission` установлено в 'a'); 2 - вывод сообщения об ошибке (`permission` установлен в 'a'); -1 - неудача в открытии файла с выводом сообщения `message` о типе ошибки. Идентификатор `fid` часто используется в качестве аргумента другими функциями и программами ввода-вывода. Имя файла `filename` может содержать путь к файлу. Если открываемый для чтения файл не найден в текущем каталоге, то функция `fopen` осуществляет поиск файла по пути, указанном в MATLAB;

- `fids = fopen('C:\all')` - возвращает вектор-строку, содержащую идентификаторы всех открытых файлов, не включая стандартные потоки 0, 1 и 2. Число элементов вектора равно числу открытых пользователем файлов;

Команда `fclose` закрывает файл. Она имеет следующие варианты:

- `status = fclose(fid)` - закрывает файл, если он открыт. Возвращает статус файла `status`, равный 0, если закрытие завершилось успешно, и -1 в противном

случае. Аргумент `fid` - это идентификатор, связанный с открытым файлом (см. функцию `foren` для более подробного описания).

- `status = fclose('all')` закрывает все открытые файлы. Возвращает 0 в случае успешного завершения и -1 - в противном случае.

4. ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ И ОЦЕНИВАНИЕ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ МОЩНОСТИ

4.1. Определение дискретного преобразования Фурье

Дискретное преобразование Фурье (ДПФ) определяется следующим соотношением

$$X(k) = T \sum_{i=0}^{N-1} x(i) \exp\left(-j2\pi \frac{ik}{N}\right), \quad (4.1)$$

где $X(k)$ - значение (комплексное) дискретного преобразования Фурье, определенное в частоте с номером k ; $x(i)$ - значение (вещественное) исходного временного ряда, определенное в момент времени с номером i ; T - период дискретизации; N - количество отсчетов (длина) временного ряда.

Дискретное преобразование Фурье связывает спектральную характеристику (комплексный спектр) $X(k)$, определенную в дискретных значениях частоты (с номером k), с дискретными значениями временного ряда (сигнала) $x(i)$, определенными в дискретные моменты времени (с номером i).

Масштаб представления спектральной характеристики определяется разрешением по частоте

$$b = 1/NT. \quad (4.2)$$

По номеру k частоты f_k можно найти ее значение в герцах

$$f_k = kb \text{ (Гц)}. \quad (4.3)$$

Аналогично по номеру i времени t_i можно найти его значение в секундах

$$t_i = iT. \quad (4.4)$$

Обратное дискретное преобразование Фурье определяется соотношением

$$x(i) = 1/NT \sum_{k=0}^{N-1} X(k) \exp\left(j2\pi \frac{ik}{N}\right). \quad (4.5)$$

Из сравнения формул (4.1) и (4.5) следует, что они отличаются знаком показателя экспоненты, множителем перед знаком суммы, а также переменной суммирования. Это позволяет строить единые программы для прямого и обратного дискретного преобразования Фурье.

Применяя формулу Эйлера, выражение (4.1) можно привести к виду

$$X(k) = C(k) - jS(k), \quad (4.6)$$

где

$$\left. \begin{aligned} C(k) &= T \sum_{i=0}^{N-1} x(i) \cos(2\pi ik/N) \\ S(k) &= T \sum_{i=0}^{N-1} x(i) \sin(2\pi ik/N) \end{aligned} \right\}. \quad (4.7)$$

Принимая во внимание (4.7), выражение (4.5) для обратного ДПФ можно привести к виду

$$x(i) = c(i) + js(i),$$

где

$$\left. \begin{aligned} c(i) &= \frac{1}{NT} \sum_{k=0}^{N-1} C(k) \cos(2\pi ik/N) + \frac{1}{NT} \sum_{k=0}^{N-1} S(k) \sin(2\pi ik/N), \\ s(i) &= -\frac{1}{NT} \sum_{k=0}^{N-1} S(k) \cos(2\pi ik/N) + \frac{1}{NT} \sum_{k=0}^{N-1} C(k) \sin(2\pi ik/N) \end{aligned} \right\}, \quad (4.8)$$

Выражения (4.7) и (4.8) могут быть положены в основу алгоритмов вычисления прямого и обратного ДПФ.

4.2. Определение спектральной плотности мощности

Оценивание спектральной плотности мощности (СПМ) с помощью дискретного преобразования Фурье осуществляется по формуле:

$$\hat{S}_{xx}(k) = \frac{T}{N} |X(k)|^2, \quad (4.9)$$

где $X(k)$ - дискретное преобразование Фурье (спектральная характеристика) временного ряда $\{x(i)\}_N$, соответствующего процессу $x(t)$, обладающего свойством эргодичности.

Черта в правой части формулы (4.9) означает операцию осреднения. Применение формулы (4.9) без операции осреднения приводит к получению "грубой" оценки СПМ. Формула (4.7) позволяет вычислить оценку СПМ посредством статистического осреднения модуля спектральной характеристики совокупности данных, поделенного на длину записи данных.

В спектральном анализе случайных временных рядов на статистическую устойчивость влияют два параметра - разрешение по частоте $b = \frac{1}{NT}$ и длина записи $T_p = NT$.

Можно показать, что оценки СПМ приближенно имеют распределение x_n^2 с n степенями свободы, где $n = 2bT_p$. Более того, для достаточно больших n , например, $n \geq 30$, распределение x_n^2 аппроксимируется гауссовским (нормальным) распределением. В этом случае нормированное стандартное отклонение (стандартное отклонение, связанное с оцениваемой величиной, т.е. процентная ошибка, или, в статистической терминологии, "коэффициент разброса") определяется соотношением:

$$\varepsilon_0 = \frac{1}{\sqrt{bT_p}}. \quad (4.10)$$

Величину ε_0 называют стандартной ошибкой. Если $B = \frac{1}{T_p}$, то $\varepsilon_0 = \frac{1}{\sqrt{(1/T_p) \cdot T_p}} = 1$. Результат означает, что вычисление оценки СПМ с использованием полной длины временного ряда имеет стандартную ошибку, равную 100 %.

Если отрезок T_p поделить на m участков, то в этом случае:

$$b = \frac{1}{T_p/m} = \frac{m}{T_p}. \quad (4.11)$$

Подставляя полученный результат в (3.9), найдем:

$$\varepsilon_0 = \frac{1}{\sqrt{(m/T_p)T_p}} = \frac{1}{\sqrt{m}}. \quad (4.12)$$

Таким образом, для повышения точности оценивания СПМ необходимо исходный временной ряд длины N разбить на m участков длины N_y , вычислить для каждого i -го участка $S_{xx}^i(k)$ по формуле (4.9), а затем найти осредненную оценку по формуле:

$$\hat{S}_{xx}(k) = \frac{1}{m} \sum_{i=1}^m S_{xx}^i(k). \quad (4.13)$$

Число степеней свободы для найденной оценки СПМ можно найти следующим образом:

$$n = 2 \frac{m}{T_p} T_p = 2m. \quad (4.14)$$

Следовательно, для повышения степеней свободы и, соответственно, статистической устойчивости оценок СПМ необходимо увеличивать число участков для осреднения.

Осреднение по частотам:

$$\hat{S}_{xx}(k) = \frac{1}{l} (S_{xx}(k) + S_{xx}(k+1) + \dots + S_{xx}(k+l-1)). \quad (4.15)$$

Следует отметить, что разрешение по частоте в данном случае определится из соотношения:

$$b = \frac{1}{T_p}. \quad (4.16)$$

Поскольку операция осреднения линейная, оценку СПМ можно найти, комбинируя осреднение по участкам с осреднением по частотам. При этом сначала выполняется осреднение по участкам, а затем - по частотам. При осреднении по m участкам с последующим осреднением l соседних спектральных оценок в итоге получаются оценки, число степеней свободы которых равно $n \approx 2lm$. Разрешение в этом случае равно $b \approx \frac{1}{N_y}$.

Листинг программы определения СПМ приведен в приложении Б.

5. ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

5.1. Разработка технического обеспечения

5.1.1. Выполнить расчет информационно-измерительного канала автоматизированной системы научных исследований, состоящего из дифференциального усилителя и активного фильтра по следующим исходным данным:

- внутреннее сопротивление датчика - R_i ;
- выходное напряжение датчика - U_c ;
- эффективное значение синфазной помехи, которая наводится в витой паре проводов, соединяющих датчик с усилителем, - $U_{сф}$;
- максимальная погрешность от синфазной помехи - δ .

При этом следует определить:

- необходимый КООС (коэффициент ослабления синфазного сигнала);

- минимально необходимый коэффициент усиления дифференциального сигнала $K_{диф}$;
- выбрать схемотехническое решение;
- найти подходящий тип операционного усилителя и рассчитать элементы схемы в соответствии с найденным $K_{диф}$.

При расчете активного фильтра учесть, что он подключается к АЦП, который работает с частотой F_0 . Порядок фильтра принять равным 2.

5.2. Разработка программного обеспечения

5.2.1. Разработать алгоритм и программу оценивания спектральной плотности с разрешением по частоте не хуже b . При этом рассчитать необходимую длину исходного временного ряда N для периода дискретизации T и ошибки оценивания ε .

Язык программирования - C++. Можно воспользоваться готовым программным модулем, содержащим функцию быстрого преобразования Фурье (БПФ).

5.2.2. Разработать интерфейс пользователя в стиле Windows, позволяющий осуществлять управление процессами моделирования временных рядов и их спектрального анализа, включая операции ввода данных, визуализации результатов в виде графиков и таблиц с цифровыми значениями.

Варианты заданий приведены в таблице 1.

Таблица 1 - Варианты заданий на курсовое проектирование

Вариант	Исходные данные							
	R_n (Ом)	U_c (мВ)	$U_{сф}$ (В)	δ (%)	F_d (Гц)	b (Гц)	T (с)	ε (%)
0	200	2,0	0,9	3	1000	2	0.005	30
1	100	2,5	1,1	5	500	5	0,002	20
2	300	3,0	1,2	3	2000	5	0,02	25
3	400	2,5	1,0	5	1500	1	0,001	40
4	150	3,0	0,8	3	500	2	0,0025	20

Продолжение таблицы 1

5	200	2,0	0,7	3	1000	5	0,001	25
6	300	2,5	1,1	5	2000	5	0,002	30
7	400	3,5	0,9	5	1500	2	0,001	40
8	150	4,0	1,2	3	500	1	0,001	20
9	200	2,0	1,0	5	1000	5	0,002	30

6. ДОКУМЕНТАЦИЯ ПРОЕКТА

6.1. Документация проекта включает техническое задание (ТЗ), пояснительную записку (ПЗ) и графическую часть (ГЧ).

6.2. Пояснительная записка имеет следующую структуру:

- Титульный лист;
- Лист задания;
- Содержание;

- Введение;
- Разработка технического обеспечения;
- Разработка программного обеспечения;
- Руководство пользователя;
- Решение контрольного примера (задача спектрального анализа);
- Заключение;
- Перечень ссылок;
- Приложения.

6.3. Графическая часть должна содержать следующие элементы:

- схему электрическую принципиальную информационно-измерительного канала;
- схему программы.

6.4. В приложениях следует привести листинги разработанных программных модулей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гарет П. Аналоговые устройства для микропроцессоров и мини-ЭВМ / П. Гарет. – М.: Мир, 1981. – 382 с.
2. Гутников В. С. Фильтрация измерительных сигналов / В.С. Гутников. - Л.: Энергоатомиздат, 1990. – 192 с.
3. Бобровский С.И. Самоучитель программирования на языке С++ в системе Borland C++ Builder 4.0 / С.И. Бобровский. - М.: ДЕСС, 1999. – 286 с.
4. Гальперин М.В. Практическая схемотехника в промышленной автоматике / М.В. Гальперин. - М.: Энергоатомиздат, 1987. – С. 262 - 269.
5. Гутников В.С. Фильтрация измерительных сигналов / В.С. Гутников -Л.: Энергоатомиздат, 1990. – С. 86 – 97.
6. ГОСТ 2.304-81 «ЕСКД. Шрифты».
7. Отнес Р. Прикладной анализ временных рядов. Основные методы/ Р. Отнес, Л. Эноксон. - М.: Мир, 1982. – 428 с.
8. Шамис В.А. С++ Builder 4.0. Техника визуального программирования / В. А. Шамис. - М.: Нолидж, 2000. – 656 с.

ПРИЛОЖЕНИЕ А

(рекомендуемое)

Ниже приведен листинг подпрограммы, предназначенной для генерации дискретного временного ряда с неквантованными элементами (без учета влияния АЦП) на языке MATLAB.

```
function tsg(szFileName)
%tsg.m - формирует и сохраняет в файле временной ряд
данных
%
if nargin == 0
szFileName = 'file.dat'
end
%задаем входные параметры
disp(['формируем временной ряд:']);
disp(['длина временного ряда ']), n = 2048
disp(['амплитуда ']), a = 1
disp(['вектор частот гармоник ']), fi = [10 40]
disp(['период дискретизации ']), dt = 0.01
% вектор временных отсчетов
t = 0:dt:dt*(n-1);
x = a*sin(2*pi*t*fi(1));
for i=2:length(fi)
x = x+ a*sin(2*pi*t*fi(i));
end
x = x/length(fi);
% открываем файл для записи
fid = fopen(szFileName,'w');
% сохраняем на диске размер временного ряда,
%период дискретизации, амплитуду и временной ряд
fprintf(fid,'%12.8f\n%12.8f\n%12.8f\n%12.8f\n
n',n,dt,a,x);
% закрываем файл
fclose(fid);
% конец функции
```

ПРИЛОЖЕНИЕ Б

(рекомендуемое)

Ниже приведен листинг программы на языке MatLAB, предназначенной для грубого оценивания СПМ.

```

%sdprq.m - производит оценивание спектральной
плотности
clear; % очищаем рабочую область
% открываем файл для чтения
fid = fopen('file.dat',r);
% заполняем массив значениями из файла
x = fscanf(fid,'%f');
% закрываем файл
fclose(fid);
n = x(1); dt = x(2); a = x(3); x = x(4:length(x));
x = x - sum(x)/length(x); % центрируем временной ряд
t = 0:dt:dt*(n-1);
% формируем вектор отсчетов по частоте
f = (0:(n-1))/(dt*n);
tic; % включаем таймер
% производим преобразование Фурье
y = fft(x);
% выключаем таймер и отображаем полученное значение
disp(['Время прямого преобразования: ' num2str(toc) '
с.']);
% задаем параметры для окна с графиком
set(gcf, 'Name','*Оценка спектральной
плотности',...
'Position',[70 100 400 250], ...
'MenuBar', 'none',...
'NumberTitle','off',...
'Visible','on');
% задаем параметры для координатной сетки
set(gca, 'Units','normalized',...
'XGrid','on',...
'YGrid','on',...
'NextPlot','add');
%выводим график сплошной зеленой линией
plot(f,abs(y)/dt,'-g');
% производим обратное преобразование Фурье
x = ifft(y);
%выводим график исходного временного ряда
plot(t,x,'-g');

```

ПРИЛОЖЕНИЕ В

(справочное)

Ниже приведены графики исходного временного ряда и грубая оценка СПМ без сглаживания комплексного спектра, построенные в среде MatLAB.

Исходные данные:

- Количество гармоник – 5;
- Частоты гармоник: 10,20,30,40,50;
- Период дискретизации – 0,001 с;
- Количество элементов - 2048
- Разрядность АЦП – 12
- Амплитуда сигнала – 10 В
- Диапазон входных напряжений от -10 В до +10 В;
- Статическая ошибка – 25%;
- Разрешение по частоте – 5 Гц.

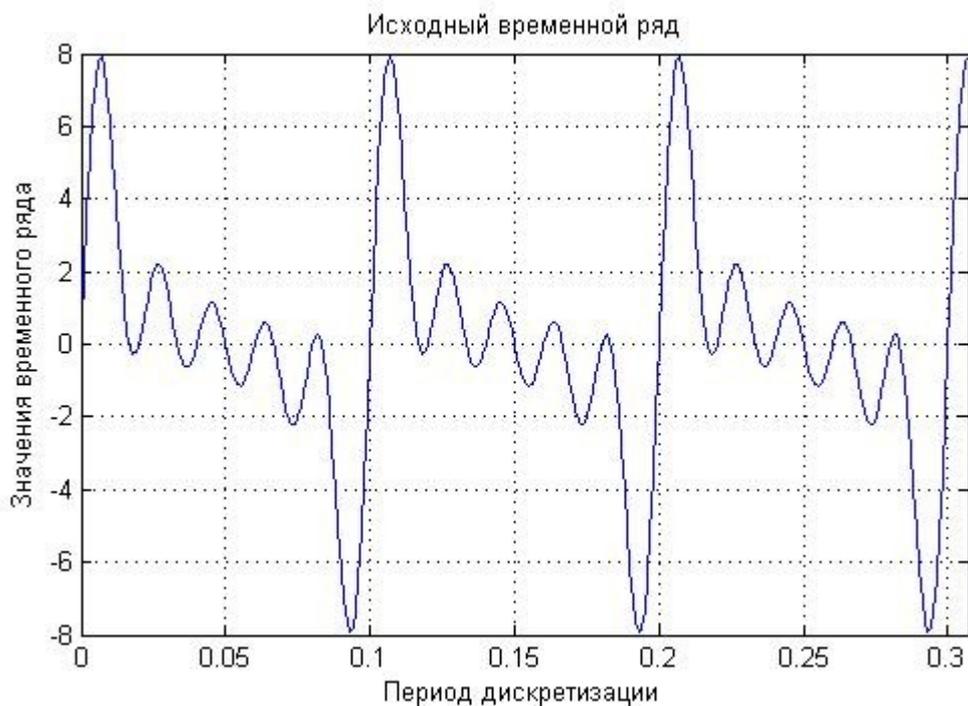


Рисунок 1 – График исходного временного ряда

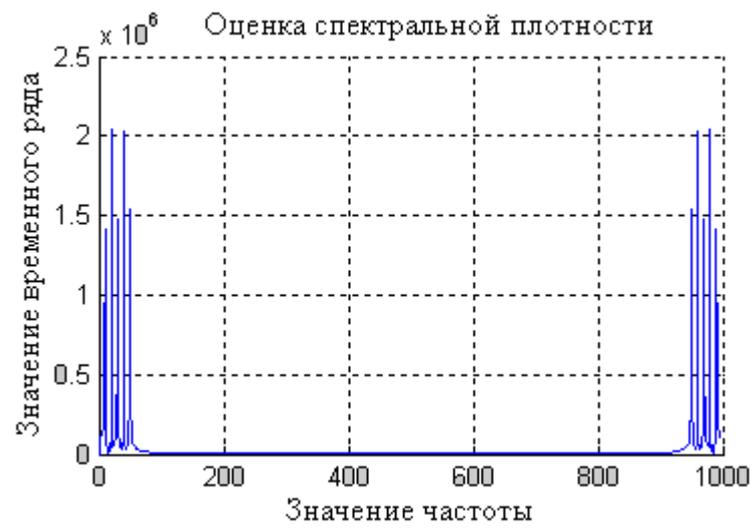


Рисунок 2 – График грубой оценки СПМ без сглаживания комплексного спектра