

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

В.С.Тутыгин

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Лабораторный практикум

Санкт-Петербург
Издательство Политехнического университета
2015

УДК 001.891:681.3 (076.5)

Тутыгин В.С. Цифровая обработка сигналов.: лаб.практикум/ В.С.Тутыгин.- СПб.: Изд-во Политехн. ун-та, 2015. - 116 с.

Лабораторный практикум соответствует основным разделам курса дисциплины «Цифровая обработка сигналов» направления бакалаврской подготовки 09.03.01 и 09.03.04.

Рассмотрены основы программирования в среде MATLAB, приведены теоретические основы и описания базовых алгоритмов и программ цифровой обработки: быстрого преобразования Фурье, разложения сигналов в действительный и комплексный ряд Фурье, цифровой фильтрации, корреляции, линейной свертки. Описаны приемы решения сложных задач путем комбинации нескольких базовых алгоритмов цифровой обработки сигналов. Рассмотрены основы анализа нестационарных сигналов и очистки сигналов от шумов на базе преобразования Гильберта-Хуанга.

Описаны теоретические основы, описания алгоритмов и примеры программ цифровой обработки изображений: средства цифровой фильтрации шумов на изображениях, алгоритмы и программы идентификации объектов на изображениях.

Приведены расчетные задания для упражнений.

Предназначено для студентов высших технических учебных заведений, обучающихся по направлению подготовки 09.03.01 – «Информатика и вычислительная техника», 09.03.04 – «Программная инженерия», студентам других специальностей, изучающих дисциплину «Цифровая обработка сигналов».

Табл. 3. Ил. 25. , Библиогр.: 11 назв.

Печатается по решению редакционно – издательского совета Санкт – Петербургского государственного политехнического университета

© Тутыгин В.С., 2015

© Санкт – Петербургский государственный
политехнический университет им. Петра Великого, 2015

Содержание

Введение.....	4
Работа 1. Основы программирования в MATLAB	6
Работа 2.Разложение дискретизированных сигналов в действительный и комплексный ряд Фурье	21
Работа 3. Цифровая фильтрация шумов в среде MATLAB.....	37
Работа 4. Комбинированное использование ключевых операций ЦОС при определении частоты короткого сигнала	56
Работа 5. Комбинированное использование ключевых операций ЦОС при определении временных интервалов между радиоимпульсами.	65
Работа 6. Подгонка кривых	76
Работа 7. Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга.	83
Работа 8. Базовые средства фильтрации шумов на изображениях.....	95
Работа 9. Средства идентификации объектов на изображениях.....	101
Расчетные задания	108
Литература	116

Введение

Сигнал – это физическая величина, которая несет определенную информацию. Примером может быть сигнал датчика, несущего информацию о температуре, давлении или других параметрах объекта. Обычно сигналы являются аналоговыми, т.е. непрерывными во времени и принимающими в каждый момент времени одно из бесконечного множества значений. Информативным параметром сигнала может быть частота сигнала, центральная частота и ширина полосы сигнала (если сигнал широкополосный), период повторения сигналов и т.д. Целью обработки при проведении физического эксперимента может быть получение аналитического описания физического процесса, т.е. получение математической модели.

Сигналы могут быть детерминированными или случайными. Часто сигналы могут быть искажены шумами и помехами.

Обработка зашумленного сигнала имеет целью обнаружение сигнала и определение информативных параметров. Обработка может производиться в аналоговой или цифровой формах.

Для выполнения обработки в цифровой форме исходный аналоговый сигнал должен быть вначале преобразован в цифровую форму, т.е. представлен в виде серии отсчетов (мгновенных значений сигнала), взятых с некоторой частотой. Операция преобразования сигнала в цифровую форму производится с помощью АЦП. Для того, чтобы по набору цифровых отсчетов можно было восстановить значения измеряемого аналогового сигнала в любой момент времени с требуемой точностью, необходимо правильно выбрать разрядность АЦП и интервал дискретности измерений во времени. Обработку сигналов, представленных в цифровой форме, называют цифровой обработкой сигналов (ЦОС).

Методами ЦОС являются математические соотношения или алгоритмы, в соответствии с которыми выполняются вычислительные операции над цифровыми сигналами. К ним относятся алгоритмы цифровой фильтрации, спектрально-корреляционного анализа, модуляции и демодуляции сигналов, адаптивной обработки и др. Алгоритмы ЦОС, в отличие от других вычислений на ЭВМ, предусматривают, как правило, их выполнение *в реальном масштабе времени*.

Средствами реализации ЦОС являются жесткая логика, программируемые логические интегральные схемы (ПЛИС), микропроцессоры

общего назначения, микроконтроллеры, персональные компьютеры, одноплатные компьютеры, цифровые сигнальные процессоры (ЦСП). Последние аппаратно и программно оптимизированы на задачи ЦОС и образуют ее специализированную элементную базу. Совокупность аппаратных средств, осуществляющих цифровую обработку сигналов, называют *процессором ЦОС*. В режиме off line ЦОС может выполняться с помощью готовых компьютерных программ, представленных наиболее полно в MATLAB.

Реализация алгоритмов ЦОС включает:

1. Высокоуровневую программную (MATLAB, LabView).
2. Низкоуровневую программную (C/C++).
3. Аппаратно-программную (C/C++, LabWindows/CVI).
4. Аппаратную (VHDL, C/C++).

Примером задачи ЦОС и средств ее решения может быть задача обнаружения и определения параметров радиолокационного линейно частотно модулированного (ЛЧМ) спутникового сигнала в реальном времени

Лабораторный практикум знакомит с приемами решения практических задач цифровой обработки сигналов средствами программы MATLAB. Приложением к приведенному ниже описанию является базовое программное обеспечение для работы с аппаратурой и демонстрационные программы.

Работа 1. Основы программирования в MATLAB

Целью лабораторной работы является изучение методики проектирования программ цифровой обработки сигналов (ЦОС): моделирования сигналов и нормально распределенного и белого шума, программирования операций математической обработки, операций графического вывода данных, проектирования средств графического пользовательского интерфейса.

Теоретические основы

Программные средства систем ЦОС обычно создаются на языке C/C++, так как в этом языке предусмотрены средства, обычно необходимые при аппаратной реализации на базе программируемых логических интегральных схем (ПЛИС) и сигнальных процессоров: удобные средства работы с битами, логические операции, средства работы с аппаратными прерываниями и др. Кроме того, компилятор C обеспечивает формирование исполняемой программы, близкой по скорости исполнения к программе на ассемблере.

При разработке систем жесткого реального времени на базе ПЛИС и быстродействующих сигнальных процессоров необходимо производить отработку алгоритмов ЦОС на универсальных средствах (с помощью компьютерных программ) используя низкоуровневое программирование без библиотечных функций (на C/C++) с тем, чтобы на следующем этапе можно было реализовать разработанные алгоритмы ЦОС на ПЛИС и сигнальных процессорах. Кроме того, как известно, ЦОС должна быть выполнена в течение критического срока обслуживания и это так же важно, как и корректность алгоритма ЦОС. Невыполнение задачи реального времени в течение критического срока обслуживания равносильно невыполнению задачи в целом. Поэтому очень важно, чтобы на втором этапе разработки отработка алгоритмов ЦОС производилась с использованием соответствующей аппаратуры и скорость выполнения операций ЦОС контролировалась, например, с использованием системного таймера компьютера.

Конечной целью разработки и исследования различных алгоритмов ЦОС является их практическая реализация в устройствах на базе ПЛИС и сигнальных процессоров, таких, например, как модуль ЦОС SAMC-401, содержащий ПЛИС серии Virtex-4, в которой интегрированы два процессора

Power PC с тактовой частотой до 450 МГц, и сигнальный процессор Texas Instruments TMS320C6455, работающий на частоте 1.2ГГц с submodule АЦП SAMC-ADC, реализующий аналого-цифровое преобразование 12/14 бит с предельной частотой тактирования до 210 МГц.

Основные стадии разработки алгоритмов ЦОС:

1. Высокоуровневая программная (MATLAB, LabView).
2. Низкоуровневая программная (C/C++).
3. Аппаратно-программная (C/C++, LabWindows/CVI).
4. Аппаратная (VHDL, C/C++).

На первом этапе разработки наиболее совершенным средством является MATLAB, так как он содержит библиотеки функций для сложных видов математической обработки, таких как быстрое преобразование Фурье, цифровая фильтрация, корреляционная обработка, преобразование Гильберта и др., библиотеки программ для создания объектов графического пользовательского интерфейса (панелей, кнопок управления, окон цифрового ввода/вывода, окон графического вывода и др.).

Программные средства, создаваемые для работы в среде MATLAB, содержат две составляющие:

1. Собственно программу (mat-файл) на языке C¹, в которой содержатся необходимые функции математической обработки, отображения таблиц и графиков. При ее создании следует пользоваться описаниями функций библиотек математической обработки и графического пользовательского интерфейса.
2. Файлы ресурсов графического пользовательского интерфейса <имя программы>.fig.

Создаваемая пользователем прикладная программа представляет проект, содержащий два файла:

- файл основной программы <имя>.mat;
- файл макета <имя>.fig.

Программа на языке C может быть написана и редактироваться пользователем. Файлы макета <имя>.fig создаются автоматически при создании и редактировании пользователем графических панелей. Эти файлы нельзя редактировать!

¹ Синтаксис языка C в MATLAB имеет ряд особенностей, в частности, связанных с тем, что объектами при вычислениях являются матрицы.

Создание средств графического пользовательского интерфейса в традиционных системах программирования, таких как Visual C++ возможно, но достаточно сложно, так как для этого необходимо создание большого количества нестандартных графических объектов.

Создание прикладной программы ЦОС в среде MATLAB

1. Создайте панель интерфейса пользователя: **File/New/GUI/Blank GUI**. В результате появится всплывающее диалоговое окно `untitled.fig`
2. Разместите элементы управления (кнопки управления `PushButton`, окна ввода/вывода текста `EditText`, окна графического вывода `Axes` и др.), например, так, как показано на рис. 1.1.

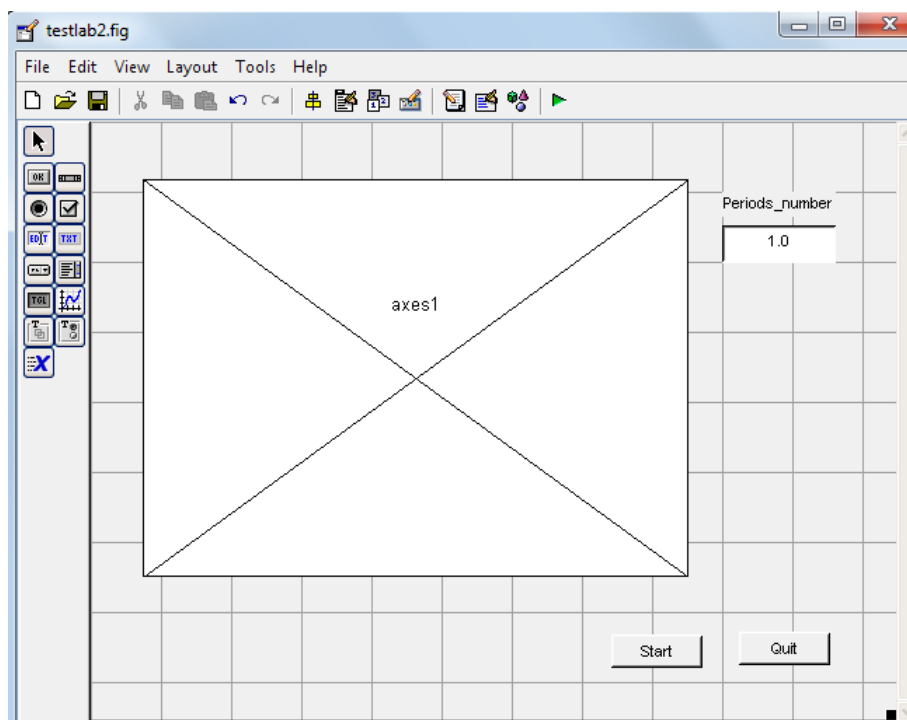


Рис. 1.1

и задать их атрибуты. Панель атрибутов элемента управления вызывается двойным щелчком левой кнопки мыши на изображении элемента. Основные атрибуты кнопки управления и окна ввода/вывода текста – `Callback`, `CreateFcn` и `tag`. Например, при задании атрибуту `tag` окна ввода/вывода текста значения `freq` имя окна будет `freq`. Такое же имя может быть дано и переменной, значение которой читается из этого окна.

Атрибуты Callback, CreateFcn создаются автоматически после задания *tag*.

3. Сохраните untitled.fig в файле с именем <имя>.fig. В результате будет не только сохранен файл интерфейса пользователя, но и создан и сохранен в файле <имя>.m «скелет» программы.

4. Создайте содержательную часть программы в блоке функции, вызываемой, например, нажатием кнопки управления (см. Приложение 2).

В итоге будут созданы два файла: файл программы с расширением .m и файл GUI с расширением .fig. Для внесения изменений в интерфейс пользователя нужно вызвать файл командой >>guide ('<полное имя файла, включая путь>.fig').

Загрузка и запуск MATLAB

Для загрузки и запуска MATLAB в среде ОС Windows XP/7 нужно вызвать из меню «Пуск»:

Programs/MATLAB

Создание программы

Создайте файл макета.

а) **File/New/GUI/Blank GUI ...**

б) сохраните созданный файл макета

File/Save Untitled.fig As.../{имя файла}

В результате на экране дисплея появится изображение панели GUI. Размеры и место расположения панели на экране можно установить с помощью мыши. Далее нужно установить атрибуты панели. Для этого перейти в режим редактирования панели двойным щелчком мыши и во всплывающей панели задать:

- Name - идентификатор панели, который определяет имя m-файла и fig-файла.
- и другие.

4. Создайте элементы GUI (графические окна вывода, текстовые окна ввода/вывода, командные кнопки и т.д.) из библиотеки элементов и разместить их на созданной панели.

Объекты GUI выбираются из меню и размещаются на экране методом "drag and drop" с помощью "мыши".

MATLAB предоставляет широкий выбор объектов:

- Edit text - окно ввода/вывода текста;
- Static Text – окно вывода текстового сообщения;
- Push Button – кнопка с двумя состояниями;
- Toggle Button – кнопка с двумя состояниями (включено/выключено);
- Check Box – флажок;
- Radio Button – аналог Option Button;
- Pop-up-menu – меню;
- Slider – линейка прокрутки;
- List box – поле со списком;
- Axis1 - окно вывода массивов данных в графической форме.

При выполнении лабораторной работы рекомендуется использовать следующие элементы управления и ввода/вывода: Edit text, Axis1, Pushbutton.

4.1 Задание атрибутов элементов GUI.

Сделайте двойной щелчок левой клавишей мыши на объекте GUI, далее во всплывающей панели задать его атрибуты, в первую очередь tag.

Далее нужно сохранить сделанные установки.

File/Save

/Save As.../{имя файла}

5. Создайте программный код. Создание fig-файла производится автоматически при сохранении GUI-образа. Одновременно создается m-файл.

В созданном программном коде m-файла объектам GUI будет соответствовать "программная оболочка" функций на языке C++. Эти функции (за исключением функции работы таймера) будут запускаться на исполнение пользователем с панели GUI. В приложении 1 приведен пример

созданной таким путем «программной оболочки».

Далее в эту оболочку вносится программный код прикладной программы на языке C++.

Если производится редактирование ранее уже созданного программного кода, например, при добавлении какого-либо объекта GUI, то для добавления в ранее созданный программный код изменений, связанных с добавленными элементами GUI нужно открыть файл <>.fig для редактирования командой:

>>guide ('<полное имя файла, включая путь>.fig')

в окне Command Window. Путь к файлу можно не указывать, если поместить его в папку MATLAB.

6. Добавьте в программу функцию выхода.

Для этого в уже сгенерированную по п.5 функцию выхода программы необходимо внести программный код:

```
fclose('all');  
close ('all');
```

7. Добавьте в функцию, вызываемую кнопкой на панели GUI (в приведенном примере - Start), строки обращения к элементам ввода/вывода GUI, текст программы. Ниже приведен пример программирования ввода текста из окна ввода/вывода (TextBox) с преобразованием в число типа double, вывода значения с преобразованием в текст, чтения номера элемента из поля со списком (ComboBox), вывода числового значения в TextBox:

```
periods_number=str2double(get(handles.periods_number,'string'));  
set(handles.chastotan,'String',fr_int);  
regim=get(handles.regim,'Value');  
set(handles.lampa,'Value',1);
```

Примечания:

1. Элементы поля со списком нумеруются, начиная с 1. Так же в MATLAB нумеруются и элементы массивов.

2. В атрибутах окна, предназначенного для вывода текстовых сообщений, задайте атрибут Style –Edit.
3. Для имитации сигнального индикатора можно использовать элемент radiobutton.

Описания типов переменных не обязательны, но возможны, если требуется. В случае необходимости передавать значения параметров из одной функции в другую нужно использовать описания глобальных переменных, например:

```
global N
```

В результате будет получен полный программный код прикладной программы. Простой пример такой программы генерации и графического отображения массива случайных чисел приведен ниже. Количество периодов сигнала задается переменной freq, получаемой с элемента GUI:

```
freq=str2double(get(handles.freq,'string'));  
for i=1:1000  
    y(i)=sin((2*pi*i*freq)/1000.0);  
end  
i=1:1000;  
plot(i,y);
```

Редактирование программного кода, в том числе строк обращения к объектам GUI производится точно так же, как в любом C/C++.

10. Запустите программу.

Debug/Run

Если компилятор найдет ошибки, то устранить ошибки и повторить предыдущее действие.

Программа работы

1. Изучите состав и функции библиотек программ MATLAB для работы с аппаратурой, математической обработки, создания средств графического пользовательского интерфейса.
2. Разработайте программу генерации гармонического сигнала.

3. Разработайте программу генерации стандартных сигналов: гармонического, пилообразного, треугольного, прямоугольного с нормально распределенным и белым шумом. Предусмотрите средства выбора вида сигнала, амплитуды сигнала, вида и уровня шума.

Указания к выполнению лабораторной работы

1. При выполнении п. 1 Программы изучите основы программирования в MATLAB по описанию к лабораторной работе и пользуясь справочной системой.
2. При выполнении п.2 Программы создайте графический пользовательский интерфейс и программу генерации синусоидального сигнала. Количество периодов сигнала сделайте регулируемым.
3. При выполнении п.3 Программы разработайте прикладную программу генерации различных стандартных сигналов:

- ♦ гармонического (синусоидального);
- ♦ пилообразного;
- ♦ треугольного;
- ♦ прямоугольных импульсов.

без шума и с нормально распределенным и белым шумом. Выбор вида сигнала, амплитуду, частоту сигналов и среднеквадратическую величину шума сделать регулируемыми.

- 3.1. Для генерации нормально распределенного и белого шума используйте функции `randn` и `wgn`. Пример фрагмента программы генерации модельного сигнала с шумом приведен ниже.

```
%Генерация нормально распределенного и белого шума
%noise=randn(points_number);%нормально распределенный
noise=wgn(points_number,1,0);%белый Гауссов шум
for i=1:points_number %генерация модельного сигнала с шумом
    x(i)=sin(2*3.14*kp*i/kt)+noise_sko*noise(i);
end
```

Здесь `points_number` – количество элементов массива (количество значений сигнала), `noise_sko` – среднеквадратическое значение шума.

3.2. На панели графического пользовательского интерфейса предусмотрите (см. рис. 2):

- ◆ переключатель выбора вида сигнала;
- ◆ элемент ввода уровня шума;
- ◆ элемент ввода количества точек за период генерируемого сигнала;
- ◆ окно графического вывода генерируемого сигнала.

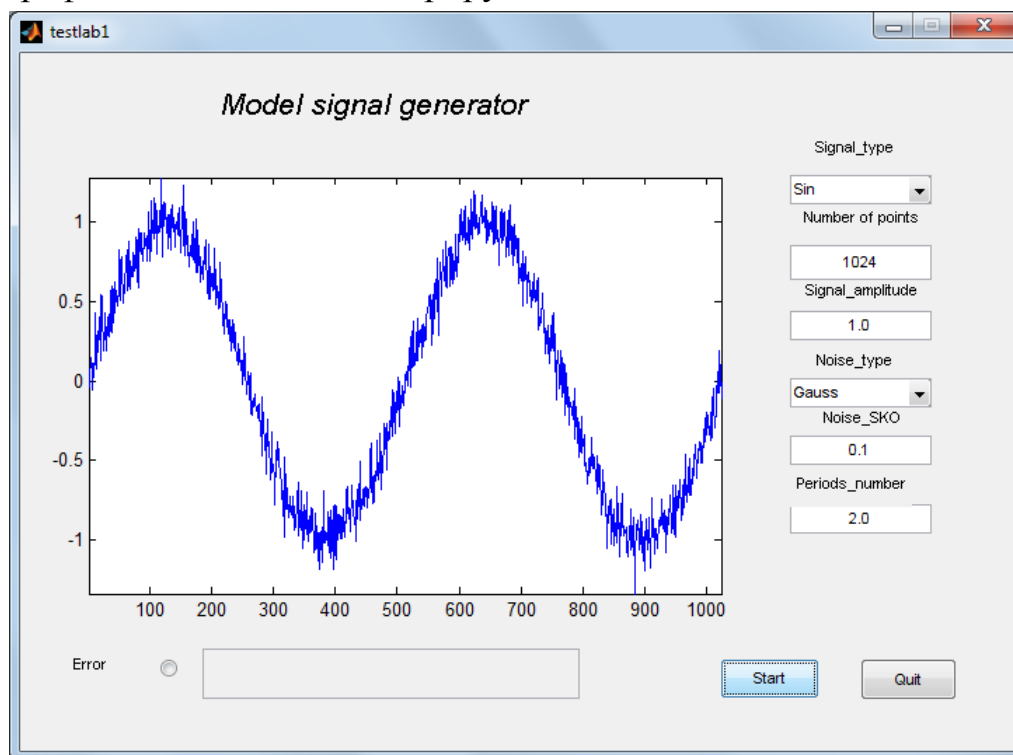


Рис. 1.2

3.3. Дополните программу, предусмотрев в ней возможность накопления (суммирования с усреднением) генерируемых сигналов.

На панели графического пользовательского интерфейса дополнительно предусмотрите переключатель задания количества накоплений.

Содержание отчета

1. Задание к работе.
2. Текст программы с комментариями.
3. Вид панели интерфейса пользователя при генерации различных сигналов с различным шумом.

Приложение 1

Пример программного кода, генерируемого автоматически после создания панели и объектов GUI в среде MATLAB

```
%Служебная часть М-файла, создаваемая автоматически после сохранения в
файле
%панели интерфейса пользователя <имя>.fig
%В данном примере панель интерфейса пользователя содержит кнопку
управления
%PushButton, окно ввода/вывода текста EditText (tag – КР), окно графического
%вывода Axes (tag – axis1)

function varargout = testlab2(varargin)
% TESTLAB2 M-file for testlab2.fig
%   TESTLAB2, by itself, creates a new TESTLAB2 or raises the existing
%   singleton*.
%
%   H = TESTLAB2 returns the handle to a new TESTLAB2 or the handle to
%   the existing singleton*.
%
%   TESTLAB2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in TESTLAB2.M with the given input
%   arguments.
%
%   TESTLAB2('Property','Value',...) creates a new TESTLAB2 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before testlab2_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to testlab2_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```

% Edit the above text to modify the response to help testlab2

% Last Modified by GUIDE v2.5 03-Feb-2012 21:54:41

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @testlab2_OpeningFcn, ...
    'gui_OutputFcn', @testlab2_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before testlab2 is made visible.
function testlab2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to testlab2 (see VARARGIN)

% Choose default command line output for testlab2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```



```

% UIWAIT makes testlab2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = testlab2_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function KP_Callback(hObject, eventdata, handles)
% hObject handle to KP (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP as text
% str2double(get(hObject,'String')) returns contents of KP as a double

% --- Executes during object creation, after setting all properties.
function KP_CreateFcn(hObject, eventdata, handles)
% hObject handle to KP (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
KP=str2double(get(handles.KP,'string'));
for i=1:1000
    y(i)=sin((2*pi*i*KP)/1000.0);
end
i=1:1000;
plot(i,y);

% --- Executes on button press in Quit.
function Quit_Callback(hObject, eventdata, handles)
% hObject    handle to Quit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
fclose('all');
close ('all');

```

Пример программного кода в функции Start

```
% --- Executes on button press in Start.
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
KP=str2double(get(handles.KP,'string'));
for i=1:1000
    y(i)=sin(2*pi*i*KP./1000.0);%KP – кол-во периодов
end
i=1:1000
plot(i,y);
```

После запуска программы Debug/Run диалоговое окно программы с результатами выполнения будет выглядеть, как показано на рис.1.3.

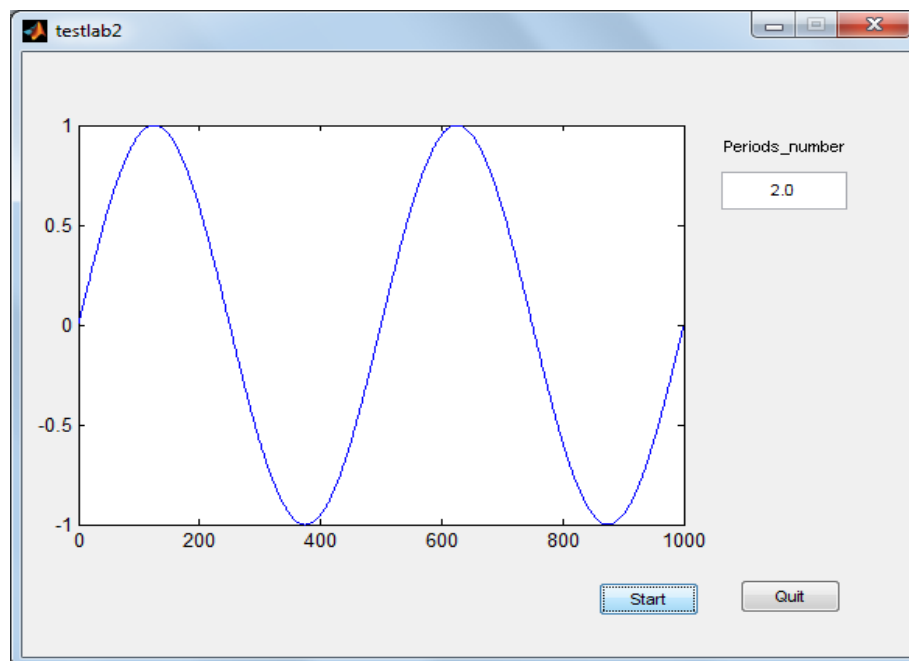


Рис. 1.3

Приложение 3. Построение и оформление графиков в MATLAB

```
i=1:N;  
y=A*sin(6.28*KP*i/N); %создание N значений KP периодов модельного sin  
%сигнала,  
figure %создание окна графического вывода  
plot(i,y(1:50));%вывод графика y, кол-во точек – 50.  
plot(i,y) ;% вывод графика y (все N точек)  
plot (i,2*y(1:N),'r-');%отображение графика y линией красного цвета  
semilogx(i,abs(y(1:200))),grid; %то же, что и plot, но в логарифмическом  
% масштабе по X  
loglog(i,abs(y(1:200))), % в логарифмическом масштабе по X и Y  
stem(A(1:K)); %вывод графика дискретной последовательности данных,  
%например, коэффициентов ряда Фурье  
[C,h] = contour(X,Y,Z);%построение контурной карты (сетки изолиний)  
hist(<имя массива данных>,<кол-во столбцов>);%построение гистограммы  
grid on; %отображение линий координатной сетки  
title('сигнал до фильтра');%заголовок графика  
xlabel('номер отсчета'); % подпись по оси X  
ylabel('амплитуда'); % подпись по оси Y  
legend('до фильтра');%подпись легенды  
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax  
%(автомасштабирование)  
hold on; % «удержание» окна вывода для следующего графика  
hold off; % отмена «удержания»  
close all; % закрытие всех открытых окон графического вывода  
clear; %очистка Workspace  
clc; %очистка Command Window
```

Работа 2.Разложение дискретизированных сигналов в действительный и комплексный ряд Фурье

Целью лабораторной работы является изучение методики разработки программ разложения дискретизированных сигналов в действительный и комплексный ряд Фурье на интервалах $[-\pi, \pi]$ и $[-\frac{T}{2}, \frac{T}{2}]$ с конечным числом членов.

Задание к работе

Имеется дискретизированная функция в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую разложение этого сигнала в ряд Фурье с конечным числом членов и исследовать зависимость точности представления этой функции с помощью ряда Фурье от числа членов разложения и от шага дискретизации исходного непрерывного сигнала.

Теоретические основы

Известно, что любую функцию $f(t)$ можно представить в виде:

$$f(t) = c_0\varphi_0(t) + c_1\varphi_1(t) + c_2\varphi_2(t) + \dots + c_k\varphi_k(t) + \dots = \sum_{k=0}^{\infty} c_k\varphi_k(t)$$

где φ_k - ортонормированные функции.

Коэффициенты c_k вычисляются по формуле:

$$c_k = \langle f(t), \varphi_k(t) \rangle = \frac{1}{b-a} \int_a^b f(t) * \varphi_k(t) dt$$

Условие ортонормированности выполняется, если скалярное произведение любых двух функций, входящих в набор, равно нулю, а норма любой функции равна единице:

$$\langle \varphi_m(t), \varphi_n(t) \rangle = \frac{1}{b-a} \int_a^b \varphi_m(t) * \varphi_n(t) dt = 0, \quad m \neq n$$

$$\|\varphi_m(t)\| = \langle \varphi_m(t), \varphi_m(t) \rangle = \frac{1}{b-a} \int_a^b \varphi_m^2(t) dt = 1$$

Разложение функции на интервале $[-\pi, \pi]$

Функцию $f(t)$ можно разложить по системе тригонометрических функций на отрезке $[-\pi, \pi]$ следующим образом:

$$f(t) = \alpha_0 + \alpha_1 \sqrt{2} \cos t + \alpha_2 \sqrt{2} \cos 2t + \alpha_3 \sqrt{2} \cos 3t + \dots + \\ + \beta_1 \sqrt{2} \sin t + \beta_2 \sqrt{2} \sin 2t + \beta_3 \sqrt{2} \sin 3t + \dots$$

Коэффициенты α_k, β_k , как было показано ранее, можно выразить через скалярные произведения:

$$\alpha_0 = \langle f(t), 1 \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) dt$$

$$\alpha_k = \langle f(t), \sqrt{2} \cos kt \rangle = \frac{\sqrt{2}}{2\pi} \int_{-\pi}^{\pi} f(t) \cos kt dt$$

$$\beta_k = \langle f(t), \sqrt{2} \sin kt \rangle = \frac{\sqrt{2}}{2\pi} \int_{-\pi}^{\pi} f(t) \sin kt dt$$

В общем виде функцию $f(t)$ можно представить следующим образом:

$$f(t) = \frac{a_0}{2} + a_1 \cos t + a_2 \cos 2t + a_3 \cos 3t + \dots + \\ + b_1 \sin t + b_2 \sin 2t + b_3 \sin 3t + \dots = \\ = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt).$$

Коэффициенты a_0, a_k, b_k называют коэффициентами Фурье, а подобное представление функции называется разложением в ряд Фурье. Иногда такое представление называют действительным разложением в ряд Фурье, а коэффициенты — действительными коэффициентами Фурье. Термин «действительный» вводится для того, чтобы отличить представленное разложение от разложения в ряд Фурье в комплексной форме, о котором будет сказано далее. Как уже было сказано раньше, произвольную функцию можно разложить по системе ортогональных функций, даже если функции из этой системы не представляются в виде тригонометрического ряда. Обычно под разложением в ряд Фурье подразумевается разложение в тригонометрический ряд. Если коэффициенты Фурье выразить через a_0, a_k, b_k , получим:

$$a_0 = 2\alpha_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) dt \quad (2.1)$$

$$a_k = \sqrt{2}\alpha_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos kt dt$$

$$b_k = \sqrt{2}\beta_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin kt dt$$

Поскольку при $k = 0$ $\cos(kt) = 1$, то константа $a_0/2$ выражает общий вид коэффициента a_k при $k = 0$.

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt \quad (k = 0, 1, 2, \dots)$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt) dt \quad (k = 0, 1, 2, \dots)$$

В последнем соотношении колебание самого большого периода, представленное суммой $\cos(t)$ и $\sin(t)$, называют колебанием основной частоты или *первой гармоникой*. Колебание с периодом, равным половине основного периода, называют *второй гармоникой*. Колебание с периодом, равным $1/3$ основного периода, называют *третьей гармоникой* и т.д. Как видно из соотношения (2.1) a_0 является постоянной величиной, выражающей среднее значение функции $f(t)$. Если функция $f(t)$ представляет собой электрический сигнал, то a_0 представляет его постоянную составляющую. Следовательно, все остальные коэффициенты Фурье выражают его переменные составляющие.

Первая гармоника является периодической функцией с периодом 2π . Прочие гармоники также имеют период, кратный 2π . Исходя из этого, при формировании сигнала из составляющих ряда Фурье мы, естественно, получим периодическую функцию с периодом 2π . А если это так, то разложение в ряд Фурье — это, собственно говоря, способ представления периодических функций.

Итак, каждая из гармоник, представленных рядом Фурье, представляет собой сумму вида $a \cdot \cos(x) + b \cdot \sin(x)$. Эта сумма может быть преобразована, как будет показано ниже, к виду $\sqrt{a^2 + b^2} \cdot \sin(x + \varphi)$.

Лемма. Если сумма квадратов двух действительных чисел равна единице, то одно из этих чисел можно рассматривать как косинус, а другое как синус некоторого угла.

Другими словами, если $a^2 + b^2 = 1$, то существует угол φ , такой, что

$$a = \cos \varphi; \quad b = \sin \varphi.$$

Прежде чем доказывать эту лемму, поясним ее на следующем примере:

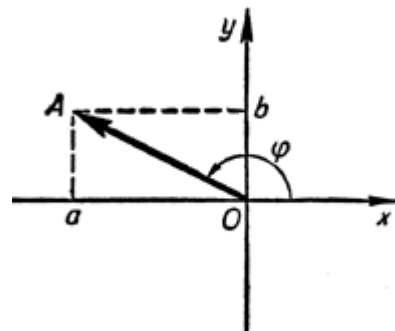
$$(\sqrt{3}/2)^2 + (1/2)^2 = 3/4 + 1/4 = 1$$

Поэтому существует угол φ , такой, что $\sqrt{3}/2 = \cos \varphi$; $1/2 = \sin \varphi$.

В качестве φ в данном случае можно выбрать любой из углов 30° , $30^\circ \pm 360^\circ$, $30^\circ \pm 2 \cdot 360^\circ$ и т. д.

Доказательство леммы.

Рассмотрим вектор \vec{OA} с координатами (a, b) . Поскольку $a^2 + b^2 = 1$, длина этого вектора равна 1. Но в таком случае его координаты должны быть равны $\cos \varphi$ и $\sin \varphi$, где φ — угол, который образует данный вектор с осью абсцисс.



Итак, $a = \cos \varphi$; $b = \sin \varphi$, что и требовалось доказать.

Доказанная лемма позволяет преобразовать выражение $a \sin x + b \cos x$ к более удобному для изучения виду.

Прежде всего вынесем за скобки выражение $\sqrt{a^2 + b^2}$

$$a \sin x + b \cos x = \sqrt{a^2 + b^2} \left(\frac{a}{\sqrt{a^2 + b^2}} \sin x + \frac{b}{\sqrt{a^2 + b^2}} \cos x \right)$$

Поскольку

$$\left(\frac{a}{\sqrt{a^2+b^2}}\right)^2 + \left(\frac{b}{\sqrt{a^2+b^2}}\right)^2 = 1,$$

первое из чисел $\frac{a}{\sqrt{a^2+b^2}}$ и $\frac{b}{\sqrt{a^2+b^2}}$ можно рассматривать как косинус некоторого угла φ , а второе как синус того же угла φ :

$$\frac{a}{\sqrt{a^2+b^2}} = \cos \varphi, \quad \frac{b}{\sqrt{a^2+b^2}} = \sin \varphi.$$

Но в таком случае

$$a \sin x + b \cos x = \sqrt{a^2+b^2} (\cos \varphi \sin x + \sin \varphi \cos x) = \sqrt{a^2+b^2} \sin (x + \varphi)$$

Итак,

$$a \sin x + b \cos x = \sqrt{a^2+b^2} \sin (x + \varphi), \text{ где угол } \varphi \text{ определяется из условий}$$

$$\sin \varphi = \frac{b}{\sqrt{a^2+b^2}} \quad \cos \varphi = \frac{a}{\sqrt{a^2+b^2}}$$

Разложение функции, представленной в дискретизированном виде, в ряд Фурье на интервале $[-\pi, +\pi]$

В дискретизированном виде (т.е. в виде набора дискретных значений или, что то же, в виде числового массива, содержащего N значений) функция $f(t)$ на интервале $[-\pi, +\pi]$ будет иметь вид:

$$f(t) \rightarrow f\left(\frac{2\pi i}{N}\right), \quad i = \left[-\frac{N}{2}, \frac{N}{2}\right]$$

где N – количество дискретных значений сигнала. При $n \rightarrow \infty$ дискретизированная функция будет приближаться к непрерывной $f(t)$.

Разложение в ряд Фурье будет иметь вид:

$$f(t) = \alpha_0 + \sum_{k=1}^{\infty} (\alpha_k \sqrt{2} \cos kt + \beta_k \sqrt{2} \sin kt)$$

а коэффициенты α_0 , α_k и β_k :

$$\alpha_0 = \left\langle f\left(\frac{2\pi i}{N}\right), 1 \right\rangle = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right);$$

$$\alpha_k = \left\langle f\left(\frac{2\pi k}{N}\right), \sqrt{2} \cos\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right);$$

$$\beta_k = \left\langle f\left(\frac{2\pi i}{N}\right), \sqrt{2} \sin\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right).$$

и коэффициенты a_0 , a_k и b_k :

$$a_0 = 2\alpha_0 = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right)$$

$$a_k = \sqrt{2}\alpha_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right)$$

$$b_k = \sqrt{2}\beta_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right)$$

а общий вид разложения:

$$\begin{aligned} f\left(\frac{2\pi i}{N}\right) &= \frac{a_0}{2} + a_1 \cos\left(\frac{2\pi i}{N}\right) + a_2 \cos\left(\frac{4\pi i}{N}\right) + a_3 \cos\left(\frac{6\pi i}{N}\right) + \dots \\ &+ b_1 \sin\left(\frac{2\pi i}{N}\right) + b_2 \sin\left(\frac{4\pi i}{N}\right) + b_3 \sin\left(\frac{6\pi i}{N}\right) + \dots \end{aligned}$$

или

$$\begin{aligned} f\left(\frac{2\pi i}{N}\right) &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos\left(\frac{k * 2\pi i}{N}\right) + b_k \sin\left(\frac{k * 2\pi i}{N}\right)) = \\ &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (\sqrt{a_k^2 + b_k^2} \sin\left(\frac{k * 2\pi i}{N} + \varphi_k\right)), \text{ где } \varphi_k = \arctg \frac{b_k}{a_k} \end{aligned}$$

Массив $C_k = \sqrt{a_k^2 + b_k^2}$ представляет спектр амплитуд, а φ_k - спектр фаз.

Разложение непрерывной функции на интервале $[-T/2, T/2]$ в действительный ряд Фурье

До этого момента мы рассматривали функцию переменной t на отрезке $[-\pi, \pi]$. В случае периодического сигнала с периодом 2π мы брали этот интервал за основной. В общем случае периодического сигнала с периодом T при разложении в ряд Фурье мы должны использовать интервал $[-T/2, T/2]$. Если интервал $[-\pi, \pi]$ расширить (или сократить) до интервала $[-T/2, T/2]$, то и период первой гармоники увеличится (или уменьшится) от 2π до T . Поскольку кратность этого преобразования равна $(T/2) * \pi$, то составляющие первой гармоники примут вид:

$$\cos \frac{2\pi}{T} t \quad \sin \frac{2\pi}{T} t.$$

Для составляющих k -й гармоники можно записать:

$$\cos \frac{2\pi}{T} kt \quad \sin \frac{2\pi}{T} kt.$$

Следовательно, если функцию $f(t)$ разложить в ряд Фурье на интервале $[-T/2, T/2]$, получим:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left\{ a_k \cos \left(\frac{2\pi}{T} kt \right) + b_k \sin \left(\frac{2\pi}{T} kt \right) \right\}.$$

Если обозначить угловую частоту через ω_0 , то поскольку $\omega_0 = \frac{2\pi}{T}$, последнее выражение можно записать и в таком виде:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \{ a_k \cos \omega_0 kt + b_k \sin \omega_0 kt \}$$

В соотношении, определяющем коэффициенты Фурье на отрезке $[-\pi, \pi]$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos kt dt.$$

произведем замену переменной

$$t \rightarrow \omega_0 t,$$

а также замену отрезка, на котором берется интеграл

$$[-\pi, \pi] \rightarrow [-T/2, T/2].$$

Оставив функцию $f(t)$ без изменения, получим

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos \omega_0 k t dt \quad (k = 0, 1, 2, \dots)$$

Аналогичным образом выводится следующее соотношение:

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin \omega_0 k t dt \quad (k = 1, 2, \dots).$$

Разложение функции, представленной в дискретизированном виде, в действительный ряд Фурье на интервале $[-T, T]$

В дискретизированном виде (т.е. в виде набора дискретных значений или, что то же, в виде числового массива, содержащего N значений) функция $f(t)$ на интервале $[-T, T]$ будет иметь вид:

$$f(t) \rightarrow f\left(\frac{2Ti}{N}\right), \quad i = \left[-\frac{N}{2}, \frac{N}{2}\right]$$

где N – количество дискретных значений сигнала. При $n \rightarrow \infty$ дискретизированная функция будет приближаться к непрерывной $f(t)$.

Разложение в ряд Фурье будет иметь приведенный ранее вид:

а коэффициенты α_0 , α_k и β_k :

$$\alpha_0 = \left\langle f\left(\frac{2Ti}{N}\right), 1 \right\rangle = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right);$$

$$\alpha_k = \left\langle f\left(\frac{2Ti}{N}\right), \sqrt{2} \cos\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right);$$

$$\beta_k = \left\langle f\left(\frac{2Ti}{N}\right), \sqrt{2} \sin\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right).$$

и коэффициенты a_0 , a_k и b_k :

$$a_0 = 2\alpha_0 = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right)$$

$$a_k = \sqrt{2}\alpha_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right)$$

$$b_k = \sqrt{2}\beta_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right)$$

а общий вид разложения:

$$f\left(\frac{2Ti}{N}\right) = \frac{a_0}{2} + a_1 \cos\left(\frac{2\pi i}{N}\right) + a_2 \cos\left(\frac{4\pi i}{N}\right) + a_3 \cos\left(\frac{6\pi i}{N}\right) + \dots \\ + b_1 \sin\left(\frac{2\pi i}{N}\right) + b_2 \sin\left(\frac{4\pi i}{N}\right) + b_3 \sin\left(\frac{6\pi i}{N}\right) + \dots$$

Разложение функции в комплексный ряд Фурье

Система функций $\{e^{jk\tau}, k = 0, \pm 1, \pm 2, \dots\}$ на отрезке $[-\pi, \pi]$ образует ортонормированную систему функций. Значит, произвольная функция $f(t)$ может быть представлена по этой системе следующим образом:

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{jk\tau}$$

Это и есть разложение в комплексный ряд Фурье. Коэффициенты C_k называются комплексными коэффициентами Фурье и, подобно действительным коэффициентам Фурье, вычисляются как скалярные произведения $f(t)$ и e^{jkt} :

$$C_k = \langle f(t), e^{jk\tau} \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-jk\tau} dt$$

Если период функции не равен 2π , а, например, равен T , то получим следующее общее выражение для комплексных коэффициентов:

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\omega_0 k t} \\ C_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega_0 k t} dt \quad (\omega_0 = 2\pi / T)$$

Содержание работы

1. Разработайте программы разложения дискретизированного сигнала $f(x)=x, x^2, |x|, |\sin(x)|, x*\sin(x), x*\cos(x), \exp(ax), \ln[2\cos(x/2)], -\ln[\operatorname{tg}(x/2)]/2, \operatorname{sign}[\sin(x)]$ в дискретизированный и комплексный ряд Фурье с конечным числом членов.
2. Исследуйте зависимость точности представления заданной функции с помощью ряда Фурье от числа членов разложения и от шага дискретизации функции.

Указания к выполнению

Используйте в качестве основы программы, приведенные в приложении.

Содержание отчета

1. Задание к работе.
2. Тексты разработанных программ с комментариями.
3. Графики зависимости погрешности представления заданной функции в действительный и комплексный ряд Фурье на интервале $[-\pi, \pi]$ и $[-\frac{T}{2}, \frac{T}{2}]$ от количества членов разложения в ряд Фурье и от шага дискретизации.
4. Выводы.

Приложение. Тексты базовых программ

Приложение 1. Программа вычисления коэффициентов разложения в

действительный ряд Фурье для функции $y(t)=t \rightarrow \frac{2\pi i}{N}, i = [-\frac{N}{2}, \frac{N}{2}]$.

% Разложение функции t в ряд Фурье

% в дискретизированном виде на интервале $[-T, T]$, например, $[-\pi, \pi]$

N=255; %Количество отсчетов (элементов массива y(t))

K=16; %Количество членов ряда Фурье

T=pi; %диапазон изменения функции f(i) равен +/-T

kp=2.4; %количество периодов гармонической функции

y=zeros(1,N+1);

Sa = zeros(1,K);

Sb = zeros(1,K);

p=1; % показатель степени функции t^p

f=zeros(1,N+1);

Sa0=0;

for i=1:N+1

 f(i)=sin(2*pi*kp*(i-1)/N); % гармоническая функция

 % f(i)= (2*T*((i-1-N/2))/N))^p; % функция t^p

 Sa0=Sa0+f(i);

end

Sa0=Sa0/N

for i=1:N+1

 for j=1:K

 Sa(j) = (Sa(j)+f(i)*cos((j)*2*pi*(i-1-N/2)/N));

 Sb(j) = (Sb(j)+f(i)*sin((j)*2*pi*(i-1-N/2)/N));

 end

end

for j=1:K

 Sa(j)=Sa(j)*(1/(N/2));

 Sb(j)=Sb(j)*(1/(N/2));

end

%Вычисление и отображение спектра амплитуд (начало)

```

for j=1:K
Sab(j)=sqrt(Sa(j)^2+Sb(j)^2);
end
i=1:K;
figure
plot(i,Sab);
stem(Sab(1:K)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]);%задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
y=zeros(1,N+1);
for i=1:N+1
    for j=1:K
        y(i)= y(i)+Sa(j)*cos(j*2*pi*(i-1-N/2)/N)+Sb(j)*sin(j*2*pi*(i-1-N/2)/N);
    end
    y(i)=Sa0+y(i);
end
i=1:N+1;
figure
plot(i,f);
axis tight;
hold on;
plot(i,y,'r-')
hold off;
pause;
close all;
clear; %очистка Workspace

```

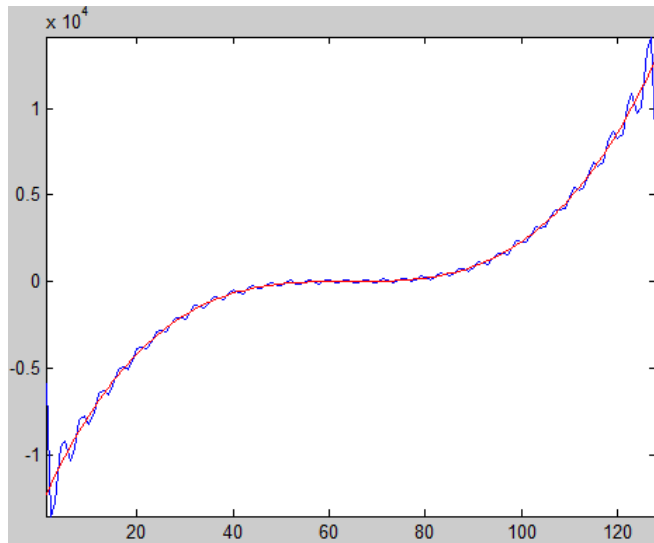


Рис. 2.1. Исходная и восстановленная функция t^3 при $N=128$, $K=32$, $p=3$.

Примечание. Для разложения четной функции из ряда можно исключить члены, содержащие синусы, для разложения нечетной – косинусы. Но можно оставить ряд разложения полностью.

Приложение 2. Программа вычисления коэффициентов разложения в действительный ряд Фурье для функции $y(t) = t \rightarrow \frac{2Ti}{N}$, $i = [-\frac{N}{2}, \frac{N}{2}]$.

```
% Разложение функции y(t)=t в ряд Фурье
% в дискретизированном виде на интервале [0,T], например, [0,  $\pi$ ]

N=255; %Количество отсчетов (элементов массива y(t)=t)
K=64;%Количество членов ряда Фурье
T=pi;%диапазон изменения функции f(i)+/-T
kp=2.0
y=zeros(1,N+1);
Sa = zeros(1,K);
Sb = zeros(1,K);
p=3;%показатель степени функции t^p
f=zeros(1,N+1);
Sa0=0;
for i=1:N+1
    f(i)=sin(2*pi*kp*(i-1)/N); % гармоническая функция
    % f(i)= (T*(((i-1))/N))^p; % функция t^p, i-1, если p>0, i, если p<0
    Sa0=Sa0+f(i);
end
Sa0=Sa0/N
for i=1:N+1
    for j=1:K
        Sa(j) = (Sa(j)+f(i)*cos((j)*2*pi*(i-1)/N));
        Sb(j) = (Sb(j)+f(i)*sin((j)*2*pi*(i-1)/N));
    end
end
for j=1:K
    Sa(j)=Sa(j)*(1/(N/2));
    Sb(j)=Sb(j)*(1/(N/2));
end
%Вычисление и отображение спектра амплитуд (начало)
for j=1:K
    Sab(j)=sqrt(Sa(j)^2+Sb(j)^2);
```

```

end
i=1:K;
figure
plot(i,Sab);
stem(Sab(1:K)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]);%задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
y=zeros(1,N+1);
for i=1:N+1
    for j=1:K
        y(i)= y(i)+Sa(j)*cos(j*2*pi*(i-1-N)/N)+Sb(j)*sin(j*2*pi*(i-1-N)/N);
    end
    y(i)=Sa0+y(i);
end

i=1:N+1;
figure
plot(i,f);
axis tight;
hold on;
plot(i,y,'r-')
hold off;
pause;
close all;
clear; %очистка Workspace

```

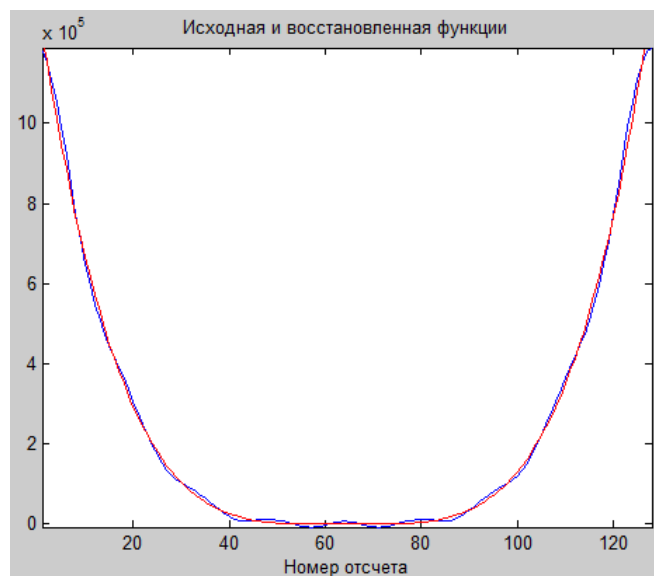


Рис. 2.2. Исходная и восстановленная функция $y=t^p$ после разложения в ряд Фурье, $N=128$, $T=5$, $p=4$, $K=16$.

Приложение 3. Программа вычисления коэффициентов разложения в

комплексный ряд Фурье для функции $y(t)=t^2 \rightarrow (\frac{2Ti}{N})^2, \quad i = [-\frac{N}{2}, \frac{N}{2}]$.

%Разложение функции t^p в комплексный ряд Фурье

%в дискретизированном виде на интервале $[0,T]$

%Восстановление функции производится по формуле

% $f_b(i)=y(i)=\sum(c(k)*\exp(j*2*\pi*i*k/N)), k=[1,M], i=[0,N-1]$

%Чем больше M, тем точнее восстановление

T=pi;%Значение T (произвольное)

N=128;%количество значений функции на интервале $[0,T]$

M=6;%количество членов ряда Фурье

p=1;%показатель степени функции x^p

kp=2.4;%количество периодов гармонического сигнала

C0=0;

for i=1:N+1 %генерация модельной функции

 f(i)=sin(2*pi*kp*(i-1)/N); % гармоническая функция

 % f(i)= (T*((i-1)/N))^p; %функция t^p

 C0=C0+f(i);

end

C0=C0*(2/N);

for k=1:M

 C(k)=0;

end

for i=1:N+1

 for k=1:M

 C(k)=C(k)+f(i)*exp(-j*2*pi*k*(i-1)/N);

 end

end

for k=1:M

 C(k)=C(k)*(2/N);

end

%Вычисление и отображение спектра амплитуд (начало)

for k=1:M

 Cab(k)=abs(C(k));%коэффициенты Cab(k)- комплексные числа вида $a+jb$,

 %функция abs вычисляет $\sqrt{a^2+b^2}$

```

end
k=1:M;
figure
plot(k,Cab);
stem(Cab(1:M)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]);%задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
for i=1:N+1
    y(i)=0;
    %f3(i)=0;
    for k=1:M
        y(i)=y(i)+C(k)*exp(j*2*pi*k*(i-1)/N);
    end
    y(i)=C0/2+y(i);
end
i=1:N+1;
figure
plot(i,f);
axis tight;
title('Исходная и
восстановленная функция')
xlabel('Номер элемента массива')
hold on;
plot(i,real(y),'r-');
axis tight;
hold off;
pause;
close all;
clear;

```

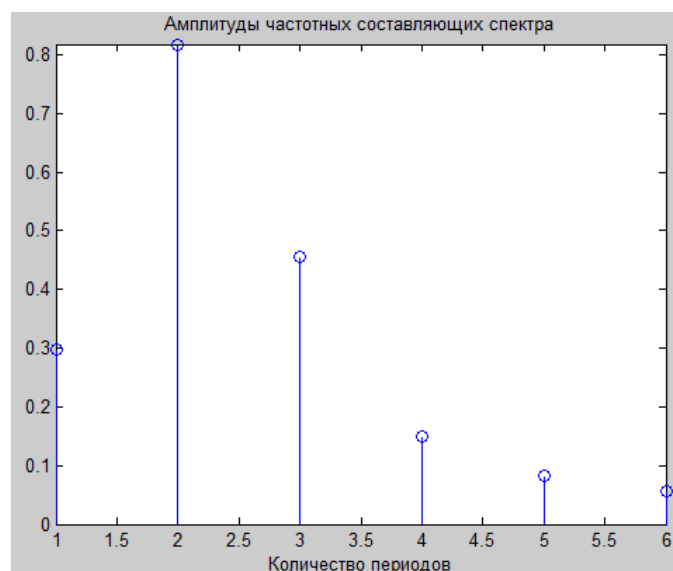


Рис. 2.3. Спектр амплитуд гармонического сигнала с количеством периодов $k_p=2.4$

Работа 3. Цифровая фильтрация шумов в среде MATLAB

Целью лабораторной работы является изучение методики разработки программ цифровой обработки сигналов, включающей различные способы улучшения отношения сигнал/шум (накопление, использование НЧ и ВЧ-фильтров, оптимального фильтра Колмогорова-Винера, прямого и обратного БПФ).

Задание к работе

Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую различные способы улучшения отношения сигнал/шум:

1. Накопление.
2. НЧ-фильтр.
3. Фильтр Баттерворта.
4. Фильтр скользящего среднего.
5. Фильтр БПФ - ОБПФ.
6. Оптимальный фильтр Колмогорова-Винера.

и оценить сравнительный эффект улучшения отношения сигнал/шум и степень искажения сигнала в результате обработки.

Теоретические основы

Цифровая обработка сигналов решает задачи обнаружения и определения параметров информативных сигналов и изображений, искаженных шумами и помехами. Для этой цели используются различные средства:

- накопление (временная фильтрация);
- цифровые частотные фильтры (высокой частоты, низкой частоты, полосовые фильтры, фильтр Баттерворта);
- сглаживающие фильтры (скользящего среднего, медианный);
- оптимальные фильтры (фильтр Колмогорова-Винера, LMS и RLS-фильтры);
- адаптивные фильтры;

Выбор способа борьбы с шумами должен производиться с учетом свойств и особенностей информативного сигнала и помехи. Чем в большей степени свойства сигнала и шума априори известны, тем может быть получен больший эффект от цифровой обработки. Кроме того, несмотря на наличие стандартных программ цифровой обработки, с учетом конкретных априори известных свойств информативного сигнала и шума может оказаться полезным разработка новых методов борьбы с шумами.

ЦИФРОВОЙ НИЗКОЧАСТОТНЫЙ ФИЛЬТР

Для фильтрации высокочастотного шума может быть применен фильтр низких частот (ФНЧ). Частотная характеристика ФНЧ выражается как

$$K(\omega) = \frac{1}{1 + j\omega}, \quad \text{где } \omega - \text{частота среза НЧ - фильтра}$$

Для фильтрации сигнала нужно вычислить частотный спектр сигнала с помощью преобразования Фурье, затем перемножить частотный спектр сигнала и частотную функцию фильтра и выполнить обратное преобразование Фурье. Второй способ – вычислить импульсную переходную характеристику фильтра (реакцию на единичный импульс), затем выполнить операцию свертки входного сигнала с импульсной переходной характеристикой фильтра.

В программах обработки дискретизированных сигналов, представленных в форме числовых массивов, цикл

```
for k=1:N
```

```
x(k) = A*sin(2*pi*KP*k/N);
```

```
end
```

создает KP периодов дискретизированного синусоидального сигнала в числовом массиве, содержащем N значений, понятие частоты в этом случае отсутствует и появляется только в том случае, если задать шаг дискретности по времени. Аналогично этому, в результате выполнения быстрого преобразования Фурье

```
I=1:N;
```

```
Y=fft(x,N);
```

мы получаем числовой массив, который может содержать N элементов, причем информативной будет только первая половина массива, вторая будет зеркально отображать первую половину. В первой половине массива, содержащей N/2

элементов, будет представлен «частотный» спектр. Спектр будет отражать не частоту сигнала, установить которую по числовому массиву, представляющему сигнал во временной области, невозможно, а количество периодов. Т.е. в приведенном выше примере пик в массиве частотного спектра будет в элементе с номером КР. Таким образом, положение пика (номер элемента массива частотного спектра) укажет на количество периодов сигнала во временной области.

Частотная характеристика линейного фильтра низких частот может быть вычислена следующим образом:

```
for i=1:N
H(i)=1/((1+j*i/NC));
end
```

Здесь NC - полоса пропускания фильтра по уровню 0,7 амплитуды выражена в количестве отчетов спектра БПФ, пропускаемых фильтром. Остальные отсчеты в массиве частотного спектра будут ослабляться по амплитуде. Таким образом, понятие постоянной времени фильтра, равно как и полосы пропускания, при дискретизированном представлении линейного фильтра отсутствует.

Ниже приведены программы фильтрации сигналов и временные диаграммы.

%Низкочастотный фильтр

A=1; %амплитуда сигнала

Q=0.05; %амплитуда шума

KP1=5;% - количество периодов первого сигнала

KP2=5;% - количество периодов второго сигнала

N=1024;%количество отсчетов

NC=5;

%NC - полоса пропускания фильтра по уровню 0,7 амплитуды

% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром

% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде

for k=1:N % генерация сигнала и шума

s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);

q(k)=Q*(randn(size(N))); %СКО шума равно Q

x(k)=s(k)+q(k); % суммирование сигнала и шума

```

end
figure
plot(x);
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Kmax
title('Зашумленный сигнал до фильтра');
Y=fft(x,N); %БПФ сигнала с шумом
i=1:N/2;
figure
% semilogy(i(1:200),2*abs(SS1(1:200)));
%plot(i(1:100),2*abs(SS1(1:100)));
plot(i(1:N/2),abs(Y(1:N/2)));
title('Частотный спектр сигнала с шумом');
for i=1:N;
H(i)=1/((1+j*i/NC)); %передаточная функция фильтра НЧ 1-го порядка
%в частотной области
end
h=ifft(H);
% HH=fft(h,N);
i=1:N;
plot(i(1:20),abs(h(1:20))); %импульсная характеристика фильтра
title('Импульсная характеристика фильтра');
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));%то же, что и plot, но в логарифмическом
%масштабе по X
grid on;
title('Частотная хар-ка НЧ-фильтра');
i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала
Z2=ifft(XX2.*H);%свертка незашумленного сигнала с частотной хар-кой
фильтра
DZ(i)=(2*real(Z(i))-2*real(Z2(i)))*100./max(real(Z2));%случайная погрешность
DZ1(i)=(2*real(Z(i))-x(i))*100/max(x);%полная погрешность

```



```

SKO=std(DZ)
SKO1=std(DZ1)
i=1:N;
yy=A*sin((6.28*KP1*i/N));
figure
plot(i,x); %вывод сигнала до фильтра
title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика
i=1:N;
plot(i,2*real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
%представление графика линией красного цвета, отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;
i=1:N;
figure
plot(i,DZ(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Случайная погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
axis tight;
i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Полная погрешность отфильтрованного сигнала');
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all;%закрытие окон графического вывода
clear;%очистка Workspace

```

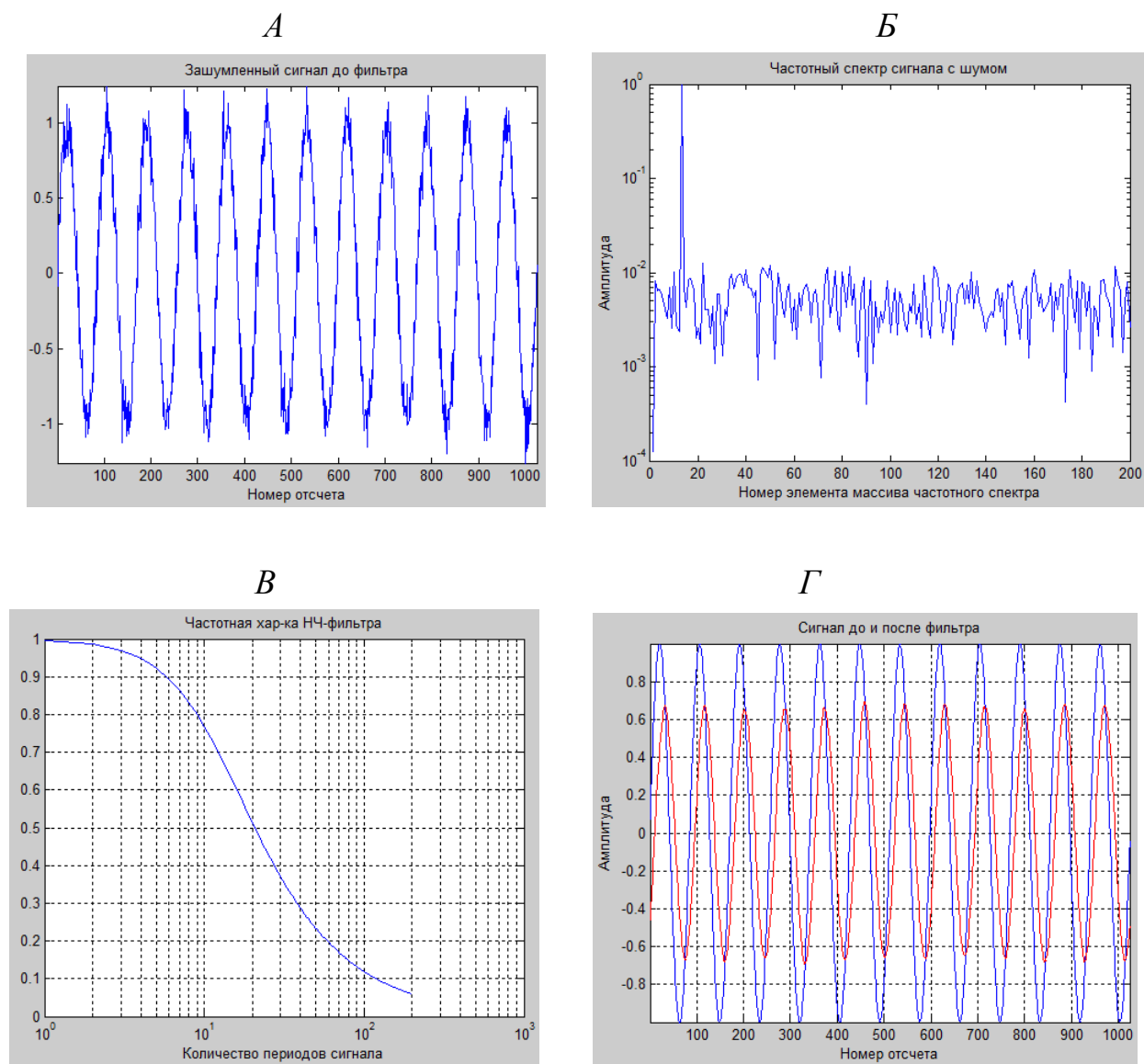


Рис.3.1. Исходный сигнал с шумом (А), его частотный спектр (Б), частотная характеристика фильтра в полулогарифмическом масштабе (В) и сигнал до и после фильтра (Г).

ФИЛЬТР БАТТЕРВОРТА

Недостатком линейного цифрового НЧ-фильтра первого порядка является существенная неравномерность амплитудно-частотной характеристики в полосе пропускания, малая крутизна ее спада. Эти недостатки могут быть в некоторой степени преодолены при использовании фильтров более высоких порядков, в значительной мере указанные недостатки преодолены в фильтрах Баттерворта.

Ниже приведена программа фильтрации сигналов с помощью фильтра Баттерворта 4-го порядка и временные диаграммы.

%Фильтр НЧ Баттерворта 4-го порядка

A=1; %амплитуда сигнала

Q=0.3; %СКО шума

KP1=5;% - количество периодов первого сигнала

KP2=5;% - количество периодов второго сигнала

N=1024;%количество точек расчета

NC=15;

%NC - полоса пропускания фильтра по уровню 0,7 амплитуды

% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром

% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде

for k=1:N % генерация сигнала и шума

s(k) = A*sin(2*pi*KP1*k/N); %+ A*sin(2*pi*KP2*k/1000);

q(k)=Q*(randn(size(N))); %СКО шума, амплитуда равна 3Q

x(k)=s(k)+q(k); % суммирование сигнала и шума

end

figure

plot(x);

title('Зашумленный сигнал до фильтра');

Y=fft(x,N)/N; %БПФ сигнала с шумом

i=1:N/2;

figure

% semilogy(i(1:200),2*abs(SS1(1:200)));d

%plot(i(1:100),2*abs(SS1(1:100)));

plot(i(1:N/2),real(Y(1:N/2)));

title('Частотный спектр сигнала с шумом');

for i=1:N;

H(i)=1/(sqrt(1+(j*i/NC).^4)); %передаточная функция фильтра Баттерворта 4-го порядка

end

i=1:200;

figure

%plot(i,abs(H(1:200))); %вывод частотной хар-ки фильтра

```

semilogx(i,abs(H(1:200))),grid;%то же, что и plot, но в логарифмическом
%масштабе по X
%loglog(i,abs(H(1:200))),grid;% в логарифм. масштабе по X и Y
axis tight;
title('Частотная хар-ка фильтра Баттерворта');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала без шума
Z2=ifft(XX2.*H);%свертка сигнала с частотной хар-кой фильтра
DZ=(2*real(Z(i))-2*real(Z2(i)))/A*100;%вычисление случайной составляющей
погрешности
DZ1=(2*real(Z(i))-s(i))/A*100;%вычисление полной погрешности
SKO=std(DZ)
SKO1=std(DZ1)
figure
plot(i,x); %вывод сигнала до фильтра
%title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
legend('до фильтра');
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика
plot(i,2*real(Z(1:N)),'r-'); %вывод отфильтрованного сигнала
%представление графика линией красного цвета
grid on; % отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;

i=1:N;
figure
plot(i,real(DZ(1:N))); %вывод случайной погрешности
title('Случайная погрешность отфильтрованного сигнала, %');
i=1:N;
figure

```

```
plot(i,real(DZ1(1:N))); %вывод полной погрешности
title('Полная погрешность отфильтрованного сигнала, %');
```

```
pause;
close all; %заккрытие окон графического вывода
clear; %очистка Workspace
```

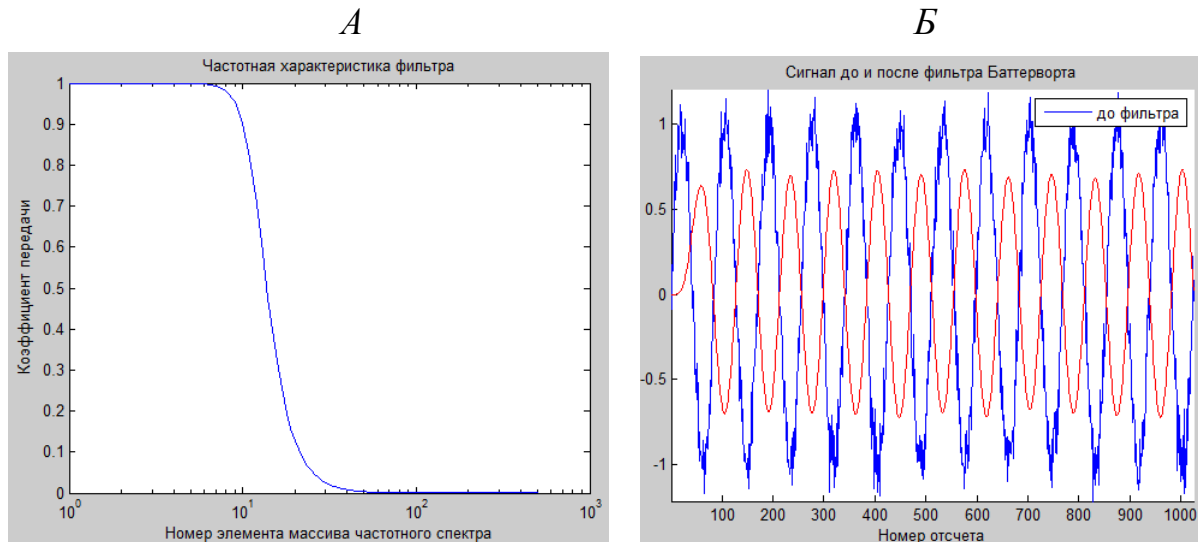


Рис.3.2. Частотная характеристика фильтра в полулогарифмическом масштабе (А) и сигнал до и после фильтра (Б).

ОПТИМАЛЬНЫЙ ФИЛЬТР КОЛМОГорова-ВИНЕРА

Фильтры низкой частоты, высокой частоты и полосовые фильтры эффективны в том случае, когда частотные спектры сигнала и шума не перекрываются.

Наилучшее разделение сигнала и шума цифровыми методами обеспечивает оптимальный фильтр Колмогорова-Винера.

Частотная характеристика фильтра Колмогорова-Винера:

$$H(w) = W_s(w) / [W_s(w) + W_q(w)] ,$$

где $W_s(w)$ и $W_q(w)$ - энергетические спектры (плотности мощности) сигнала и помех.

Спектр плотности мощности W в MATLAB рассчитывается по формуле:

$$W(k) = \frac{1}{N * F_s} X(k) .* conj(X(k))$$

где $X(k)$ – N-точечное ДПФ N-точечной последовательности $x(n)$, N-размерность ДПФ, F_s – частота дискретизации.

$W(k)$ можно вычислять по более простой формуле:

$$W(k) = X(k) \cdot \text{conj}(X(k))$$

т.к. множитель $\frac{1}{N * F_s}$ в выражении $H(w)$ равен единице.

Программа, реализующая оптимальный фильтр Колмогорова – Винера в среде MATLAB для трех часто встречающихся видов сигнала: гармонического, колоколообразного и прямоугольного, приведена ниже.

%Фильтр Колмогорова-Винера

A=1; %амплитуда сигнала

Q=0.1; %амплитуда шума (в долях СКО)

N=1024;%количество точек расчета

kp1=12;

kp2=24;

for k=1:N %цикл вычисления сигнала и шума

%s1(k)=A*exp(-0.0003*(k-200)^2.0); %колоколообразный сигнал

s1(k)=A*sin(2*pi*kp1*k/N);%+A*sin(2*pi*kp2*k/N);%гармонический сигнал

% s1(k)=0; % сигнал прямоугольной формы

% if (k>100)&(k<300) % сигнал прямоугольной формы

% s1(k)=A;

% end

q(k)=Q*(randn(size(N))); %шум

x1(k)=s1(k)+q(k); % суммирование сигнала и шума

end

figure

plot(x1(1:N));

title('Зашумленный сигнал до фильтра');

axis tight;

Y=fft(x1,N)/N; %БПФ сигнала с шумом

SS1=Y.*conj(Y)/N; %спектр мощности

i=1:200;

figure

%plot(i,SS1(1:200));

semilogy(i,SS1(1:200)); %вывод спектра мощности сигнала с шумом

title('Частотный спектр сигнала с шумом');

```

Y=fft(s1,N)/N; %БПФ сигнала без шума
f = N/2*linspace(0,1,N/2);%вычисление шкалы для частотной области
SS1=Y.*conj(Y)/N; %спектр мощности сигнала без шума
Y1=fft(q,N)/N; %БПФ шума
f = N/2*linspace(0,1,N/2);
SS2=Y1.*conj(Y1)/N; %спектр мощности шума
for i=1:N
H(i)=SS1(i)/(SS1(i)+SS2(i));%частотная характеристика оптимального фильтра
end
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));
%hold on
title('Частотная характеристика оптимального фильтра');
i=1:N;
XX1=fft(x1,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
axis tight;
figure
plot(i,s1(1:N)); %вывод незашумленного сигнала до фильтра сигнала
title('Незашумленный сигнал до фильтра');
axis tight;
figure
plot(i,Z(1:N)); %вывод отфильтрованного сигнала
title('Сигнал после свертки с част. хар-кой оптимального фильтра');
axis tight;
i=1:N;
DZ(i)=Z(i)-s1(i);
DZ1=DZ*100/max(s1);
SKO=std(DZ1)

i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');

```

```

ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all; %заккрытие окон графического вывода
clear; %очиска Workspace

```

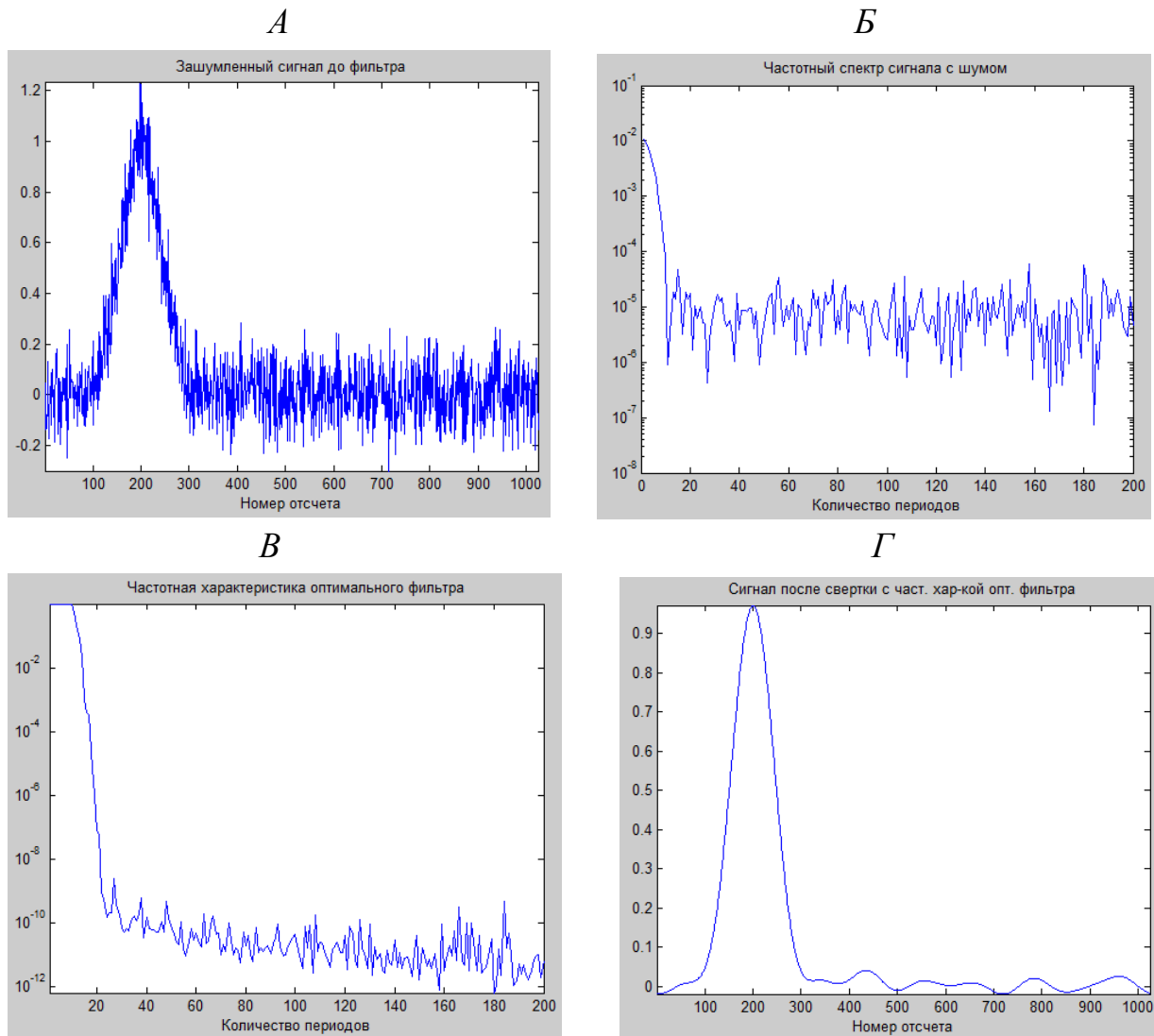


Рис. 3.3. Исходный колоколообразный сигнал с шумом (А), его частотный спектр (Б), частотная характеристика оптимального фильтра Колмогорова-Винера (В) и сигнал после фильтра (Г).

ФИЛЬТРАЦИЯ ШУМОВ С ИСПОЛЬЗОВАНИЕМ ПРЯМОГО И ОБРАТНОГО БПФ

В том случае, когда частотные спектры сигнала и шума не перекрываются, фильтрация шума может быть произведена путем выполнения БПФ, обнуления спектральных линий шума и последующего обратного БПФ.

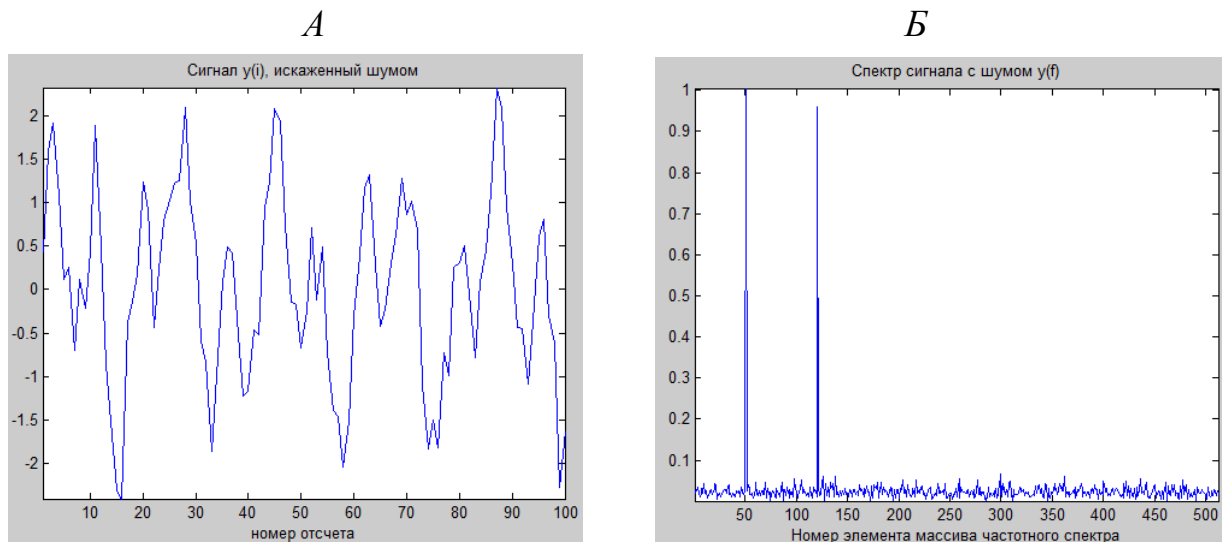


Рис. 3.4. Зашумленный сигнал (А), представляющий сумму двух синусоидальных сигналов разных частот (количество периодов во временном окне 50 и 120), и частотный спектр, полученный с помощью БПФ (Б).

Если теперь обнулить участки спектра от 0 до 40, от 55 до 110 и от 125 до 500, а затем выполнить обратное преобразование БПФ (ifft), то получим спектр и сигнал, представленные на рис. 3.5.

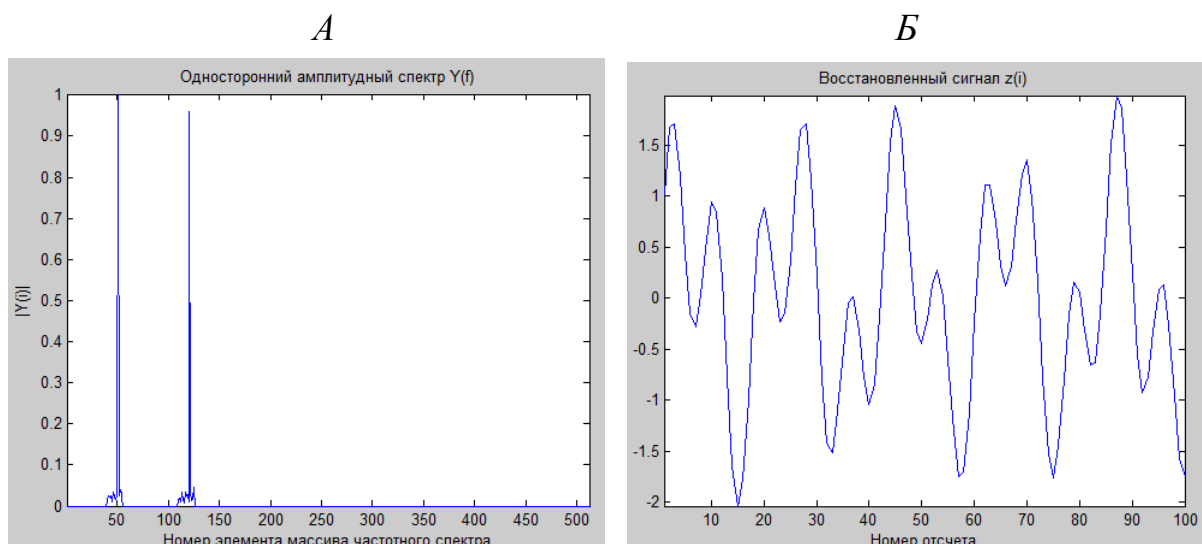


Рис. 3.5. Частотный спектр после удаления спектральных составляющих шума (А) и восстановленный с помощью обратного преобразования Фурье сигнал (Б)

Текст программы фильтрации помех с помощью прямого и обратного преобразования Фурье приведен ниже.

```
%Программа фильтрации помех с использованием БПФ-ОБПФ
```

```
A=1;%амплитуда первого сигнала
```

```
B=1; %амплитуда второго сигнала
```

```
Q=0.4;%уровень шума, выраженный в долях СКО. СКО шума равно 1.
```

```
N=1024;%количество отсчетов сигнала
```

```
kp1=50;%количество периодов 1-го сигнала
```

```
kp2=120;%количество периодов 2-го сигнала
```

```
% Сумма сигналов 1 и 2
```

```
for i=1:N
```

```
x(i) = A*sin(2*pi*kp1*i/N) + B*sin(2*pi*kp2*i/N);
```

```
z(i)=Q*randn(size(N));%
```

```
y(i) = x(i) + z(i); % Суммирование сигнала и шума
```

```
end
```

```
Y = fft(y,N)/N;%вычисление спектра сигнала с шумом
```

```
i=1:N/2;
```

```
figure
```

```
plot(i,2*abs(Y(1:N/2)));
```

```
title('Спектр сигнала с шумом y(f)')
```

```
xlabel('Номер элемента массива частотного спектра')
```

```
axis tight;
```

```
for i=1:N %"вырезание" спектральных составляющих шума
```

```
if (i>125)|(i<40)|((i>55)&(i<110))
```

```
Y(i)=0;
```

```
end
```

```
end
```

```
i=1:N/2;
```

```
figure
```

```
plot(i,2*abs(Y(1:N/2)));
```

```
title('Односторонний амплитудный спектр Y(f)')
```

```
xlabel('Номер элемента массива частотного спектра')
```

```
ylabel('|Y(i)|')
```

```

axis tight;
%восстановление исходного сигнала после "вырезания"
%спектральных составляющих шума
z=ifft(Y)*2*N;
i=1:N;
figure
plot(i(1:100),x(1:100));
title('Исходный сигнал x(i)')
xlabel('Номер отсчета');
axis tight;
figure
plot(i(1:100),y(1:100));
title('Сигнал y(i), искаженный шумом')
xlabel('номер отсчета')
axis tight;
figure
plot(i(1:100),z(1:100));
title('Восстановленный сигнал z(i)')
xlabel('Номер отсчета');
axis tight;
for i=1:N
DZ(i)=x(i)-real(z(i));%уровень зашумления в сигнале после фильтра
%результат обратного преобразования Фурье - массив комплексных чисел
%Восстановленный сигнал - действительные части комплексных чисел
%Вычисляются с помощью функции real
end
DZ1=DZ*100/max(x);
SKO=std(DZ1)
figure
i=1:N;
plot(i,DZ1(i:N)); %вывод погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
xlabel('Номер отсчета');
axis tight;

```

```

pause;
close all; %заккрытие всех окон графического вывода
clear;%очистка Workspace

```

ФИЛЬТР СКОЛЬЗЯЩЕГО СРЕДНЕГО

Пусть мы имеем массив N значений измеренного сигнала, представленный в цифровой форме:

$$\{f_1, f_2, \dots, f_N\}, i=1, 2, 3, \dots, N$$

Для нахождения скользящего среднего в окрестности точки f_i берем среднее арифметическое от K предыдущих и K последующих точек, включая и f_i . Таким же образом производим обработку для всех значений i . В результате вычисляем новый массив g_i :

$$g_i = \frac{1}{2K+1} (f_{i-K} + f_{i-K-1} + \dots + f_i + \dots + f_{i+K})$$

или

$$g_i = \frac{1}{2K+1} \sum_{j=-K}^K f_{i+j}$$

Текст программы фильтрации помех с помощью фильтра скользящего среднего приведен ниже.

```

%Фильтр скользящего среднего

A=1; %амплитуда сигнала
Q=0.05; %СКО шума
KP1=10;% - количество периодов первого сигнала
KP2=5;% - количество периодов второго сигнала
N=1024;%количество точек расчета

for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
q(k)=Q*(randn(size(N))); %СКО шума равно Q
x(k)=s(k)+q(k); % суммирование сигнала и шума

```

```

end

S=x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7);
y(4)=S/7;
S1=s(1)+s(2)+s(3)+s(4)+s(5)+s(6)+s(7);
y1(4)=S1/7;
for i=1:N-7 %сглаживание зашумленного сигнала
    S=S-x(i)+x(i+7);
    y(i+4)=S/7;
end

for i=4:N-4
    DZ(i)=s(i)-y(i);%уровень зашумления в сигнале после фильтра
end
DZ=DZ*100/max(s);%остаточная погрешность после фильтрации
SKO=std(DZ)

i=4:N-4;
figure;
plot(x);
hold on;
title('Зашумленный сигнал до фильтра');
plot(i,y(4:N-4),'r-');
title('Сигнал после фильтра');
xlabel('Номер отсчета'); % подпись по оси X
axis tight;
hold off;
figure
plot(i,DZ(4:N-4)); %вывод погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all; %закрытие окон графического вывода
clear; %очистка Workspace

```

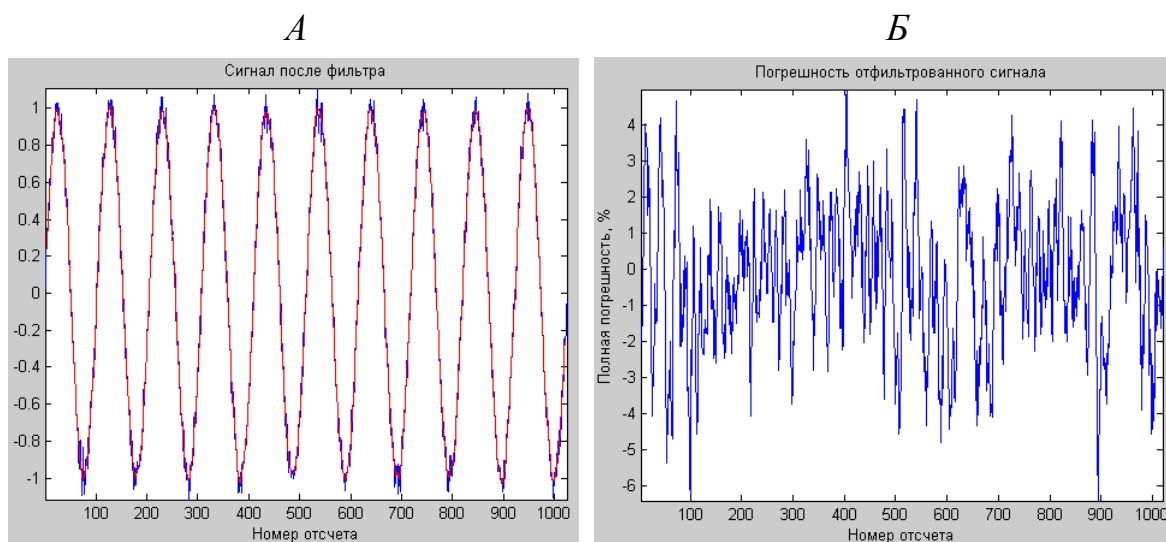


Рис. 3.6. Сигнал до и после сглаживания (А) и полная погрешность после сглаживания (Б).

Указания к выполнению работы

1. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря накоплению постройте зависимость отношения сигнал/шум и разности исходного и обработанного сигналов от количества накоплений.

2. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтров (НЧ, Баттерворта, БПФ-ОБПФ и др.) найдите оптимальную полосу пропускания фильтра, при которой эффект будет наибольшим с точки зрения улучшения отношения сигнал/шум и уменьшения полной погрешности обработанного сигнала отдельно.

3. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтра скользящего среднего (Smoothing) определите зависимость степени подавления шумов от ширины окна (при рамере окна 3, 5, 7 элементов) при различном уровне шумов и влияние параметров фильтра на изменения амплитуды и фазы сигнала на выходе фильтра.

4. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала произведите сравнение эффективности исследуемых цифровых фильтров с оптимальным фильтром Колмогорова-Винера при обработке сигналов различной формы:

гармонических, импульсных, колоколообразных при различных параметрах частоты и длительности.

Содержание отчета

1. Задание к работе.
2. Таблица исходных данных.
3. Тексты программ фильтрации с комментариями
4. Графики исходного и обработанного сигналов, график погрешности при различных параметрах обработки.
5. Выводы.

Работа 4. Комбинированное использование ключевых операций ЦОС при определении частоты короткого сигнала

Целью лабораторной работы является изучение методики разработки программ сложных видов цифровой обработки сигналов, включающей комбинацию ключевых операций (БПФ, корреляции, сплайн-аппроксимации и передискретизации).

Задание к работе

Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую точное определение количества периодов сигнала и частоту во временной области с использованием нескольких ключевых операций ЦОС: БПФ, корреляции, сплайн-аппроксимации и передискретизации.

Теоретические основы

Одной из важнейших задач цифровой обработки зашумленных сигналов является обнаружение информативного сигнала в потоке данных, искаженных шумами и помехами, и определение его параметров. Для этого применяются различные методы, такие как временная фильтрация (накопление), оптимальная частотная фильтрация, прямое и обратное преобразование Фурье, корреляционная обработка. Каждая из этих операций позволяет выполнять преобразования исходного сигнала, например, переход сигнала из временной области в частотную или наоборот, причем при этом производится уменьшение уровня шумов в обработанном сигнале. В задачах обнаружения и определения параметров зашумленных сигналов усиление эффекта подавления шумов и увеличения точности определения параметров сигнала можно достичь, используя несколько методов цифровой обработки в комплексе. Примером может быть задача обработки эхо-сигнала спектрометра ЯМР

Результатом БПФ дискретизированного эхо-сигнала спектрометра ЯМР определенной частоты является количество периодов сигнала во временном окне. Если частота отсчетов или интервал дискретности по времени при измерении сигнала известен, то по количеству периодов во временном окне

можно установить и частоту измеряемого сигнала. Точность определения частоты в спектре входного сигнала вполне определена и зависит от количества периодов p сигнала. Если количество периодов целое, то частота с помощью БПФ находится абсолютно точно (при отсутствии зашумленности сигнала). Если же количество периодов не является целым, то появляется погрешность определения частоты. Максимальное значение погрешности равняется $1/(2p)$. В некоторых, практически важных случаях, например, при обработке эхо-сигналов (рис. 1) импульсных спектрометров ЯМР, количество периодов анализируемого сигнала во временном окне принципиально ограничено величиной около 10. В этом случае погрешность определения частоты с помощью БПФ достигает $1/5$, т. е. 5%.

Влияние шума в регистрируемом сигнале во временной области можно значительно уменьшать за счет многократного повторения эксперимента и синхронного накопления эхо-сигналов (см. рис. 4.1).

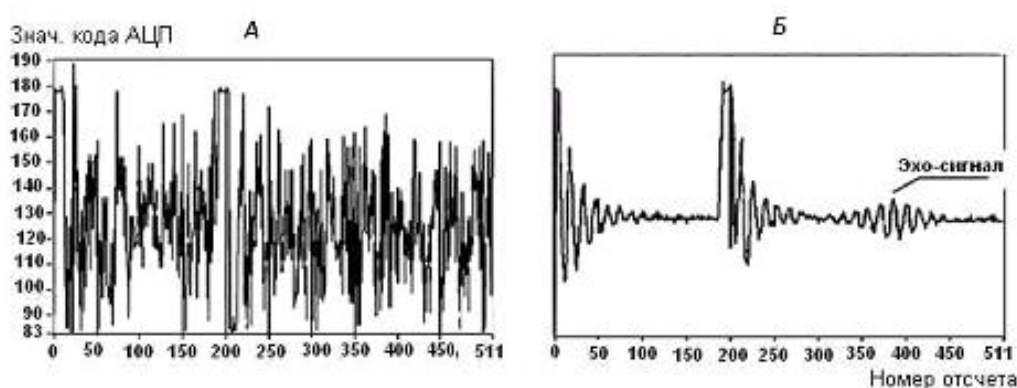


Рис. 4.1. Исходный (А) и накопленный сигнал (Б) в борате железа.

Однако увеличение количества накоплений позволяет улучшать отношение сигнал/шум без искажения формы и уменьшения амплитуды накопленного эхо-сигнала лишь до некоторого предела. После этого предела накопление уже не приносит ощутимого улучшения качества. В частности, при большом времени накопления на эхо-сигнал начинают влиять постепенные изменения параметров приборов, входящих в состав импульсного спектрометра ЯМР. При ограничении времени проведения анализа веществ количество возможных накоплений сигнала должно быть ограничено или вообще должно отсутствовать. Проблема обработки эхо-сигналов в условиях значительных шумов возникает еще и в том случае, когда имеется необходимость исследования веществ малой концентрации. Поэтому задача выполнения частотного анализа зашумленных эхо-сигналов актуальна.

Непосредственное использование БПФ для зашумленного сигнала не позволяет получить точное значение количества периодов во временном окне и частоты измеряемого сигнала в том случае, когда количество периодов не является целым, когда анализируемый сигнал занимает только часть временной области, модулирован по амплитуде и зашумлен. В практически важных случаях частотного анализа сигналов спектрометров ЯМР форма и начальная фаза эхо-сигналов известны. Это дает возможность создать эталонные сигналы, соответствующие ожидаемому эхо-сигналу по форме и начальной фазе, и произвести их корреляционное сравнение. Коэффициент корреляции эхо-сигнала с эталонным сигналом будет равен единице, если частоты эхо-сигнала и эталонного сигнала равны и эхо-сигнал не зашумлен. Поэтому при отсутствии шумов найти частоту эхо-сигнала можно производя корреляционное сравнение с эталонными сигналами, частоту эталонных сигналов подбирать до выполнения условия, когда коэффициент корреляции будет равен единице. Однако коэффициент корреляции уменьшается как при разнице частот эхо и эталонного сигналов, так и при совпадении частот, но из-за наличия шума. Поэтому таким способом определить частоту зашумленного сигнала невозможно.

Усовершенствование алгоритма определения частоты сигнала достигается за счет сочетания положительных качеств корреляционного подхода и БПФ с целью повышения точности определения частоты дискретизированного сигнала в условиях, когда обрабатываемый сигнал во временной области зашумлен, количество периодов не является целым, сигнал модулирован по амплитуде, количество периодов сигнала мало.

Идея предлагаемого алгоритма², описанного в [2], заключается в том, что в небольшой окрестности от предполагаемой частоты сигнала или количества периодов (приближенное значение частоты сигнала может быть найдено с помощью быстрого преобразования Фурье) вычисляются коэффициенты корреляции с несколькими эталонными сигналами в некоторой окрестности от приближенного значения, затем с помощью сплайн-интерполяции и передискретизации строится функция, выражающая зависимость коэффициента корреляции от частоты эталонов и находится максимум этой функции, по положению максимума определяется частота эталонного сигнала. Функция, построенная таким образом, имеет вид параболы с явно выраженным максимумом (см. рис. 4.2) как в случае незашумленного так и зашумленного

² Тутыгин В.С. Способ измерения частоты сигнала. Патент РФ №2478213.

сигнала, что и позволяет определить частоту эхо-сигнала более точно, чем это позволяет сделать БПФ. При наличии шума форма функции сохраняется, уменьшается лишь абсолютное значение максимума.

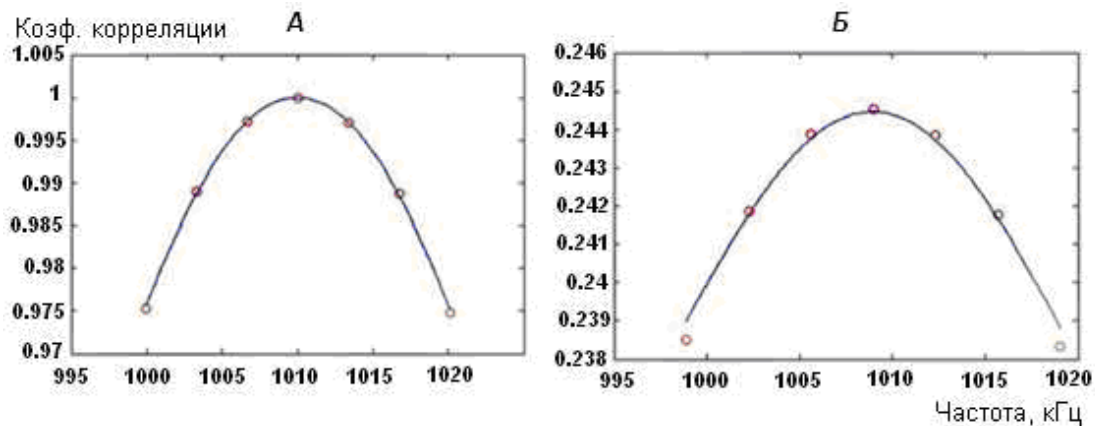


Рис. 4.2. Зависимость коэффициента корреляции от частоты при отсутствии шума (А) и при отношении сигнал/шум 1/3 (Б). Точное значение частоты равно 1010. Аппроксимирующие кривые построены с помощью функции сплайн-аппроксимации `splaps` в MATLAB.

Точность определения частоты сигнала тем выше, чем ближе начальное приближение к истинной частоте. Поэтому применяется итерационное вычисление, на каждом этапе итерации в качестве начального приближения используется уточненное значение частоты, полученное на предыдущем этапе. В качестве первого приближения берется частота, определенная с помощью БПФ.

Количество итераций для расчета частоты с заданной точностью с помощью описанного алгоритма зависит от того, насколько близко к искомой частоте будет находиться начальное приближение.

Содержание работы

Исследуйте эффективность использования комбинированного использования нескольких операций цифровой обработки и итерационного процесса для определения количества периодов и частоты сигнала по сравнению с использованием для решения этой задачи быстрого преобразования Фурье. Исследование провести для различных условий:

1. Для различного количества итераций.
2. При целом и нецелом количестве периодов сигнала, меньшем 10.
3. При различном уровне зашумленности сигнала

Указания к выполнению

1. Используйте в качестве основы программу, приведенную в приложении.

2. При исследовании эффекта увеличения точности определения количества периодов и частоты сигнала по сравнению с БПФ за счет дополнительных операций цифровой обработки произведите измерения количества периодов сигнала в диапазоне количества периодов от K до $K+1$ с шагом 0.2 и постройте графики погрешности определения количества периодов сигнала на этом интервале с помощью БПФ и комбинированного способа. Результаты измерений занесите в таблицу:

Табл. 4.1

СКО шума	Кол-во периодов сигнала k_p	1-я итерация				2-я итерация				3-я итерация			
		k_{p_int}	Отн. погрешность k_{p_int}	k_{p_fft}	Отн. погрешность k_{p_fft}	k_{p_int}	Отн. погрешность k_{p_int}	k_{p_fft}	Отн. погрешность k_{p_fft}	k_{p_int}	Отн. погрешность k_{p_int}	k_{p_fft}	Отн. погрешность k_{p_fft}
0	4												
	4,2												

3. При исследовании достижимой точности определения количества периодов и частоты сигнала комбинированным способом при различных уровнях зашумленности сигнала произведите оценку точности определения количества периодов и частоты сигнала в диапазоне количества периодов от K до $K+1$ с шагом 0.2 при значениях СКО шума от 0 до 0.3 с шагом 0.1. Для этого произведите статистические испытания, постройте гистограммы распределения вычисленных значений количества периодов и вычислите максимальную и среднеквадратическую погрешности определения количества периодов и частоты сигнала для каждого уровня шума.

4. Исследуйте влияние способа сравнения эхо-сигнала с эталонными (с помощью ковариации, корреляции, нормы Минковского, нормы Поддорогина) на точность определения количества периодов и частоты сигнала.

Содержание отчета

1. Задание к работе.
2. Текст разработанной программы.

3. Таблица погрешностей определения количества периодов сигнала с помощью БПФ и исследуемого способа комбинированной обработки.
4. Гистограммы распределения вычисленных значений количества периодов для исследуемого способа комбинированной обработки при различных уровнях шума.
5. Графики зависимости среднеквадратической погрешности определения количества периодов и частоты сигнала от уровня зашумленности исходного сигнала.
6. Выводы.

Приложение. Текст базовой программы

```
%Комбинированное использование ключевых операций ЦОС
%Для повышения точности определения частоты
%"короткого" сигнала используется комбинация
%БПФ, кросскорреляции, сплайн-аппроксимации, передискретизации
%В качестве показателя сравнения исходного и эталонных сигналов
%предусмотрена возможность использования коэффициента ковариации,
%коэффициента корреляции, суммы модулей разности (нормы Минковского),
%суммы модулей суммы (нормы Поддорогина)

clc;%очистка Command Window
kt=1024; % количество отсчетов
Q=0.1;%шум
kp=4.3;%количество периодов сигнала

%1. ГЕНЕРАЦИЯ МОДЕЛЬНОГО СИГНАЛА
for i=1:kt %обнуление массива сигнала
    y(i)=0;
end
noise=randn(kt);
%noise=wgn(kt,1,0);
for i=1:kt %генерация модельного сигнала с экспоненциальной модуляцией
    w(i)=exp(-20*((i-kt/2)/kt)^2);
```

```

    y(i)=sin(2*pi*kp*i/kt)*w(i);
    y(i)=y(i)+Q*noise(i);
end
i=1:kt; %отображение модельного сигнала во временной области
figure
plot(i,y);
axis tight;
title('Original signal')
xlabel('Sample number')
%2. ФУНКЦИОНАЛЬНОЕ ПРЕОБРАЗОВАНИЕ (БПФ)
bpfy=fft(y,kt);%БПФ
bpf=bpfy.*conj(bpfy)/kt;%БПФ
%нахождение макс. знач. функции БПФ для массива Y
C=max(bpf);
for i=1:kt %поиск количества периодов, соответствующих максимуму БПФ
    if (bpf(i)==C)
        kpbpf=(i-1);
        break
    end
end
kpbpf=kpbpf

%3. СОЗДАНИЕ ЭТАЛОНОВ И КРОССКОРРЕЛЯЦИЯ
kp1=kpbpf;
seach_area=0.8/kp1;%область поиска относит. kp_bpf
for ki=1:3 %количество итераций
shagkor=kp1*seach_area/3;%шаг поиска
    k=0;
    for j=kp1-kp1*seach_area:shagkor:kp1+kp1*seach_area %цикл для создания 6
эталонов в окрестности приближенного
        %значения количества периодов, определенных с помощью БПФ.
        k=k+1;
        xkor(k)=j;
        kor(k)=0;
        for i=1:kt
            x(i)=0;

```

```

end
%Вычисление массивов эталонных сигналов
for i=1:kt
    x(i)=sin(2*pi*j*i/kt)*w(i);
end
%вычисление средних значений модельного и эталонных сигналов
x_sr=mean(x);
y_sr=mean(y);
x_sko=0;
y_sko=0;
kor1(k)=0;%%начальное значение показателя сравнения
%вычисление показателя сравнения модельного и эталонных сигналов
for i=1:kt
    x_sko=x_sko+(x(i)-x_sr)*(x(i)-x_sr);
    y_sko=y_sko+(y(i)-y_sr)*(y(i)-y_sr);
    kor(k)=kor(k)+(x(i)-x_sr)*(y(i)-y_sr);%вычисление коэф. ковариации
    sxy(i)=abs(x(i)-y(i));%вычисление модуля разности
    kor1(k)=kor1(k)+sxy(i); %вычисление суммы модулей разности
    (нормы Минковского)
    sxy(i)=abs(x(i)+y(i));%вычисление модуля суммы
    kor1(k)=kor1(k)+sxy(i); %вычисление суммы модулей суммы
    (нормы Поддорогина)
end
    kor1(k)=kor(k)/(sqrt(x_sko*y_sko));%вычисление коэф. корреляции
end %конец цикла создания эталонов и вычисления массива коэф. корр.

%СПЛАЙН-АППРОКСИМАЦИЯ И ПЕРЕДИСКРЕТИЗАЦИЯ
xx=1:k;
xi=1:0.1:k;
r1=sin(xx); %только для тестирования сплайн-аппроксимации
yint=interp1(xx,kor1,xi,'spline');% сплайн-аппроксимация коэф корреляции
r1=kor1;
apr=spaps(xkor,kor1,0.000001);
figure
fnplt(apr)
hold on

```

```
plot(xkor,r1,'ro');  
hold off
```

%НАХОЖДЕНИЕ УТОЧНЕННОГО ЗНАЧЕНИЯ КОЛИЧЕСТВА ПЕРИОДОВ
СИГНАЛА

```
    cmax=max(yint); %нахождение максимума коэф. корр.  
    for i=1:round((k-1)/0.1+1)  
        if (yint(i)==cmax)  
            kp_int=kp1-kp1*seach_area+(i-1)*shagkor/10 %уточненное значение  
частоты по МАХ функции коэф. корр.  
        end  
    end  
    seach_area=seach_area/2;  
    kp1=kp_int;  
end
```

```
pause;  
close all;%закрытие всех окон графического вывода  
clear;%очистка Workspace
```


Работа 5. Комбинированное использование ключевых операций ЦОС при определении временных интервалов между радиоимпульсами.

Целью лабораторной работы является изучение методики разработки программ сложных видов цифровой обработки сигналов, включающей комбинацию ключевых операций (БПФ, корреляции, сплайн-аппроксимации и передискретизации).

Задание к работе

Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую точное определение временного интервала между радиоимпульсами с использованием нескольких ключевых операций ЦОС: БПФ, корреляции, сплайн-аппроксимации и передискретизации.

Теоретические основы

Одной из важнейших задач цифровой обработки зашумленных сигналов является обнаружение информативного сигнала в потоке данных, искаженных шумами и помехами, и определение его параметров. Для этого применяются различные методы, такие как временная фильтрация (накопление), оптимальная частотная фильтрация, прямое и обратное преобразование Фурье, корреляционная обработка. Каждая из этих операций позволяет выполнять преобразования исходного сигнала, например, переход сигнала из временной области в частотную или наоборот, причем при этом производится уменьшение уровня шумов в обработанном сигнале. В задачах обнаружения и определения параметров зашумленных сигналов усиление эффекта подавления шумов и увеличения точности определения параметров сигнала можно достичь, используя несколько методов цифровой обработки в комплексе. Примером может быть задача обработки эхо-сигнала электромагнитно-акустического толщиномера (ЭМА-толщиномера).

Известный способ измерения расстояния до объекта основан на измерении времени задержки отраженного радиолокационного эхо-сигнала от

возбуждающего радиоимпульса. Такой способ применяется, например, в электромагнитно-акустических (ЭМА) толщиномерах, использующих принцип измерения времени прохождения ультразвукового импульса, порождаемого радиоимпульсом, через металлическое изделие.

В этом случае источник радиоимпульса помещается у одной поверхности металлического изделия, и регистрируется эхо-сигнал, отраженный от другой поверхности. По времени задержки эхо-сигнала от возбуждающего радиоимпульса определяется толщина металла. При коррозии металлических изделий эхо-сигнал оказывается значительно зашумлен (см. рис. 5.1А), что приводит к уменьшению точности определения временной задержки и, следовательно, к уменьшению точности определения толщины металла.

Для улучшения отношения сигнал/шум в эхо-сигнале могут применяться различные методы, такие как временная фильтрация (накопление), оптимальная частотная фильтрация, прямое и обратное преобразование Фурье, корреляционный анализ.

Влияние шума в регистрируемом эхо-сигнале во временной области можно значительно уменьшать за счет многократного повторения эксперимента и синхронного накопления эхо-сигналов (см. рис. 5.1Б).

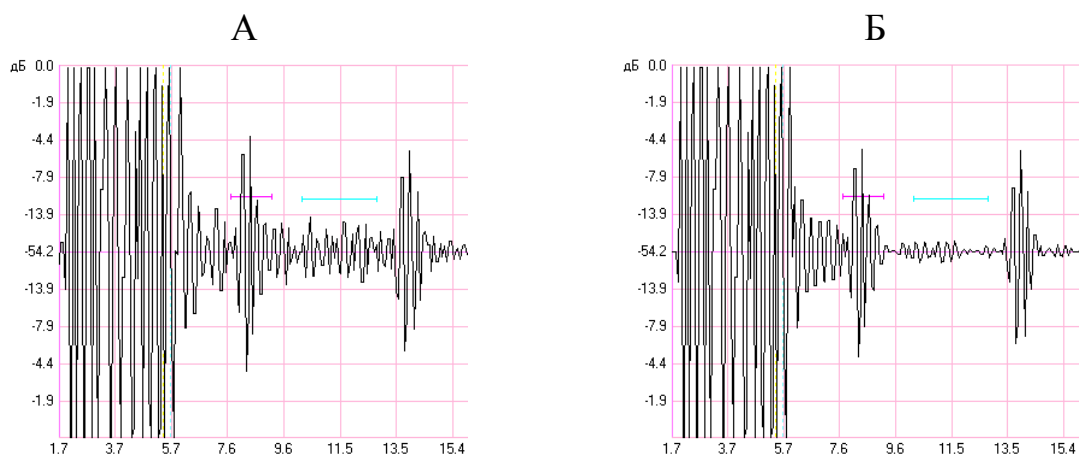


Рис. 5.1. Исходный (А) и накопленный (Б) эхо-сигнал ЭМА толщиномера.

Количество накоплений – 16.

Однако, увеличение количества накоплений позволяет улучшать отношение сигнал/шум без искажения формы и уменьшения амплитуды накопленного эхо-сигнала лишь до некоторого предела. После этого предела накопление уже не приносит ощутимого улучшения качества. В частности, при большом времени накопления на эхо-сигнал начинают влиять постепенные изменения параметров узлов, входящих в состав прибора для измерения

толщины металла. При ограничении времени проведения анализа количество возможных накоплений сигнала должно быть ограничено или вообще должно отсутствовать. Поэтому актуальна задача повышения точности измерения временного сдвига зашумленных эхо-сигналов.

Использование традиционного подхода определения временного сдвига по максимуму эхо-сигнала или автокорреляционной функции для зашумленного сигнала не позволяет получить точное значение временного сдвига. Но, поскольку форма и начальная фаза эхо-сигналов известны, это дает возможность создать эталонные сигналы, соответствующие ожидаемому эхо-сигналу по форме и начальной фазе, и произвести их корреляционное сравнение. Коэффициент корреляции эхо-сигнала с эталонным сигналом будет равен единице, если частоты эхо-сигнала и эталонного сигнала равны и эхо-сигнал не зашумлен. Поэтому при отсутствии шумов найти временной сдвиг эхо-сигнала можно, производя корреляционное сравнение с эталонными сигналами, временной сдвиг эталонных сигналов подбирать до выполнения условия, когда коэффициент корреляции будет равен единице. Однако коэффициент корреляции уменьшается как при разнице времени сдвига эхо и эталонного сигналов, так и при совпадении времени сдвига, но из-за наличия шума. Поэтому таким способом определить временной зашумленного эхо-сигнала невозможно.

Усовершенствование алгоритма определения времени задержки эхо-сигнала достигается за счет сочетания положительных качеств известного подхода, позволяющего приблизительно определить временную задержку, и дополнительной цифровой обработки, включающей механизмы генерации серий эталонных сигналов, корреляционное сравнение эхо-сигнала с эталонами, сплайн-интерполяцию, передискретизацию с целью повышения точности определения временного сдвига дискретизированного сигнала. Подобный подход был ранее применен (см. лаб. работу 4) для повышения точности определения частоты эхо-сигналов спектрометров ЯМР.

Идея предлагаемого алгоритма цифровой обработки³ в данном случае заключается в том, что в небольшой окрестности от предполагаемого временного сдвига сигнала (приближенное значение временного сдвига сигнала может быть найдено с помощью автокорреляции или по максимуму амплитуды) вычисляются коэффициенты корреляции с несколькими эталонными сигналами в некоторой окрестности от приближенного значения

³ Тутыгин В.С. Способ измерения временных интервалов между радиоимпульсами. Патент РФ №2456632.

временного сдвига, затем с помощью сплайн-интерполяции и передискретизации строится функция, выражающая зависимость коэффициента корреляции от временного сдвига эталонов и находится максимум этой функции, по положению максимума определяется уточненное значение временного сдвига эталонного сигнала. Функция, построенная таким образом, имеет вид параболы с явно выраженным максимумом (см. рис. 2А) как в случае незашумленного так и зашумленного сигнала (см. рис. 2Б), что и позволяет определить временной сдвиг радиоимпульса более точно. При наличии шума форма функции сохраняется, уменьшается лишь абсолютное значение максимума.

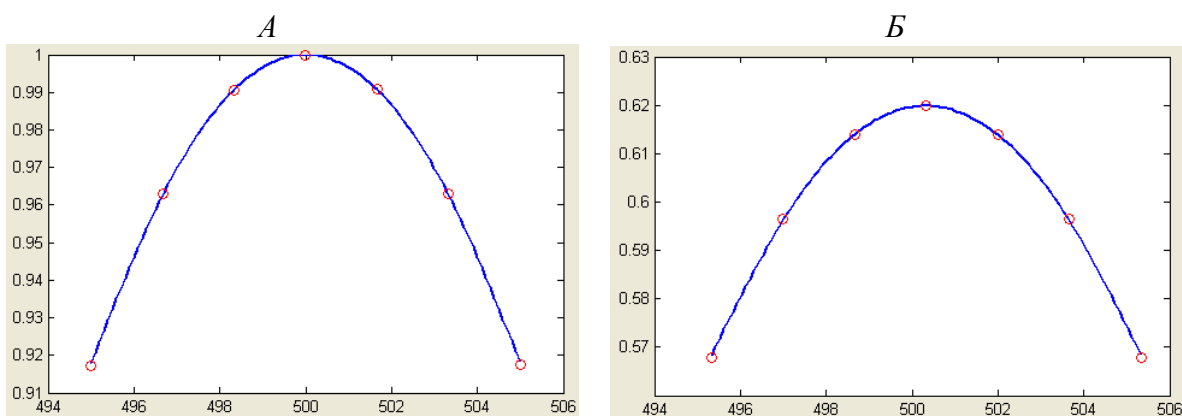


Рис. 5.2. Зависимость коэффициента корреляции от временного сдвига при отсутствии шума (А) и при отношении сигнал/шум 2/1 (Б). Точное значение времени сдвига равно 500. Вычисленное значение времени сдвига равно 499.998 (А) и 500.327 (Б). Заданная точность вычисления – 0.001. Для построения графиков использована функция сплайн-аппроксимации `splaps` в MATLAB.

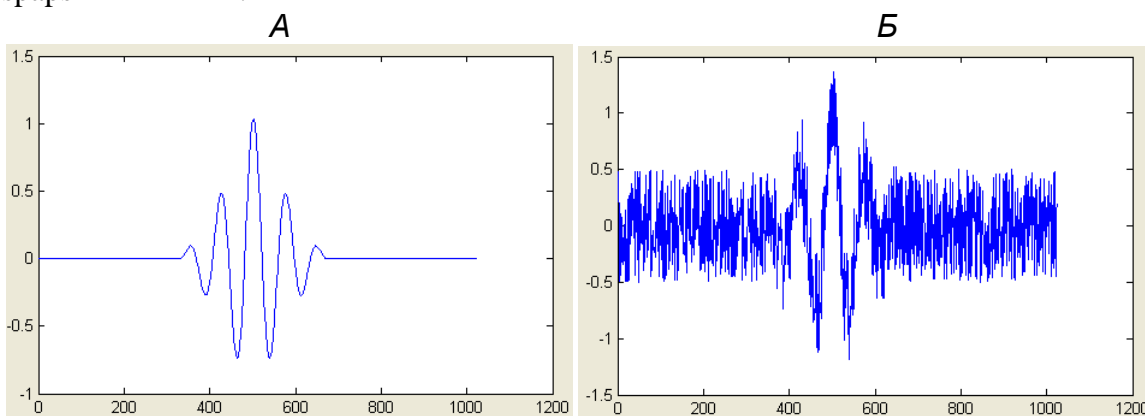


Рис. 5.3. Вид незашумленного (А) и зашумленного (Б) модельных сигналов, для которых приведены зависимости рис. 2. Частота сигналов - 2660 кГц, частота отсчетов – 200000 кГц. Величина сдвига – 500. Шум – нормально распределенный (функция `randn` в MATLAB).

Точность определения временного сдвига радиоимпульса тем выше, чем ближе начальное приближение к истинному значению. Поэтому предлагаемый

алгоритм применяет итерационное вычисление, на каждом этапе итерации в качестве начального приближения используется уточненное значение временного сдвига, полученное на предыдущем этапе. В качестве первого приближения берется временной сдвиг, определенный по максимуму амплитуды радиоимпульса.

Краткое описание алгоритма.

- 1) Получение рабочего числового массива исходных данных $X[i]$, $i = 0, \dots, n$;
- 2) Нахождение значения $j_{\text{макс}}$, соответствующего максимальному значению X . (найденная величина $j_{\text{макс}}$ соответствует приближенному значению временного сдвига в числовом массиве $X[i]$);
- 3) Создание $2k + 1$ эталонных числовых массивов исходных данных $Xs[i]$, с временным сдвигом $j_{\text{макс}} * (s/k - 1) * b$, где $s = 0, 1, 2, \dots, 2k$, $b = 0..1$, значение b задается с учетом возможной погрешности приближенного значения временного сдвига. Например, $b = 0.1$, если погрешность оценки приближенного значения временного сдвига не превышает 10%;
- 4) Вычисление коэффициентов корреляции рабочего числового массива со всеми эталонными и формирование числового массива коэффициентов корреляции $KK[m]$, где $m = j_{\text{макс}} + j_{\text{макс}} * (s/k - 1) * b$;
- 5) Выполнение сплайн-интерполяции для массива $KK[m]$ (Нахождение непрерывной функциональной зависимости $F(m)$, соответствующей массиву $KK[m]$);
- 6) Выполнение передискретизации на основе найденной функциональной зависимости $F(m)$ для массива $KK[m]$ с увеличением количества элементов массива в r раз, т. е. формирование массива $KK1[m_1]$, где $m_1 = m * r$, r – коэффициент передискретизации;
- 7) Нахождение значения $m_{1\text{макс}}$, соответствующего максимальному значению $KK1$ (значение $m_{1\text{макс}}/r$ будет представлять вещественное число, соответствующее уточненному (в общем случае нецелому) значению временной задержки в рабочем числовом массиве исходных данных $X[i]$).
- 8) Вычисление разности $\text{err} = m_{1\text{макс}}/r - j_{\text{макс}}$;
- 9) Если $\text{err} < \text{err}_{\text{доп}}$, то $j_{\text{макс}} = m_{1\text{макс}}/r$ и переход к п.3., иначе переход к п. 10;
- 10) Вывод найденного точного значения временной задержки (в безразмерной форме) $m_{1\text{макс}}/r$ (в общем случае не целого) и временной задержки

в единицах времени $T = (m_{\text{Iмакс}}/r) \cdot dt$, где dt — шаг дискретности по времени при измерении сигнала $X(t)$.

Программа, реализующая данный алгоритм, приведена в [8].

Точность определения основной частоты при использовании предложенного алгоритма зависит от значений k и r и тем выше, чем они больше, однако, если анализируемый сигнал зашумлен, существенное ее увеличение происходит при росте k и r лишь до некоторого предела. В частности, при соотношении сигнал/шум $>2/1$, $k = 3$ и $r = 10$, как показали проведенные исследования, оказывается наилучшим выбором по критерию точность/время анализа.

Количество итераций для расчета частоты с заданной точностью с помощью описанного алгоритма зависит от того, насколько близко к искомой частоте будет находиться начальное приближение.

Содержание работы

1. Исследуйте эффективность использования комбинированного использования нескольких операций цифровой обработки и итерационного процесса для определения временного сдвига радиоимпульсов по сравнению с использованием для решения этой задачи нахождения положения максимума амплитуды (МА) отраженного сигнала или аппроксимационно-корреляционным методом (АКМ) исходного задержанного и эталонных радиоимпульсов. Исследование проведите при различных способах формирования радиоимпульса и различном уровне зашумленности сигнала.

Указания к выполнению

1. Используйте в качестве основы программу, приведенную в приложении.

2. При исследовании эффекта увеличения точности определения временного сдвига методом АКМ по сравнению с методом МА произведите измерения временного сдвига в диапазоне от K до $K+100$ с шагом 20 и вычислите погрешности определения временного сдвига сигнала на этом интервале. Результаты занесите в таблицу:

Табл. 5.1

СКО шума	Точное значение задержки	1-я итерация				2-я итерация				3-я итерация			
		dt_acm	Отн. погрешность dt_acm	dt_ma	Отн. погрешность dt_ma	dt_acm	Отн. погрешность dt_acm	dt_ma	Отн. погрешность dt_ma	dt_acm	Отн. погрешность dt_acm	dt_ma	Отн. погрешность dt_ma
0	300												
	320												

3. При исследовании эффекта увеличения точности определения количества периодов и частоты сигнала методом АКМ по сравнению с методом МА при зашумленности сигнала произведите измерения временного сдвига в диапазоне количества периодов от K до $K+100$ с шагом 20 при значениях амплитуды шума от 0 до 0.3 с шагом 0.1. Для этого произведите статистические испытания, постройте гистограммы распределения вычисленных значений времени задержки и вычислите максимальную и среднеквадратическую погрешности определения времени задержки сигнала для каждого уровня шума.

4. Исследуйте влияние выбора вида оконной функции (Барлетта, Хемминга, Хеннинга, экспоненциальной), используемой для формирования радиоимпульса, способа сравнения задержанного радиоимпульса с эталонными (с помощью ковариации, корреляции, нормы Минковского, нормы Поддорогина) на точность определения временной задержки.

Содержание отчета

1. Задание к работе.
2. Текст разработанной программы.
2. Таблица погрешностей при различном уровне шума от величины задержки сигнала для методов МА и АКМ.
3. Гистограммы распределения определенных методами МА и АКМ значений времени задержки, полученные при статистических испытаниях.
4. Графики зависимостей максимальной и среднеквадратической погрешностей определения значений времени задержки методами МА и АКМ от уровня зашумленности радиоимпульса.
5. Выводы.

Приложение. Текст базовой программы.

%Программа определения времени задержки радиоимпульса
%Используется сочетание приближенного определения времени задержки
%по сдвигу максимума радиоимпульса с вычислением функции коэффициентов
%сходства радиоимпульса с серией эталонных сигналов,
%передискретизацией, нахождением максимума этой функции, итерационным
%процессом вычисления.

kt=1024;%количество отсчетов сигнала
shift=333;% сдвиг отраженного модельного сигнала
h=100;%полуширина радиоимпульса
Q=0.1;%уровень шума в долях СКО

clc; %очистка Command Window
for i=1:kt
 y(i)=0;
 sy(i)=0;
 w(i)=0;
 w1(i)=0;
end

%ПРОГРАММНАЯ ГЕНЕРАЦИЯ СИГНАЛА

noise=randn(kt);%генерация массива нормально распределенного шума
for i=1:kt %генерация оконной функции
 if(i>(shift-h)&(i<=shift+h))
 w(i)=1-abs(i-shift)/h;%окно Барлетта(треугольное)
 w(i)=0.5*(1+cos(pi*(i-shift)/h));%окно Хеннинга
 w(i)=0.54+0.46*cos(pi*(i-shift)/h);%окно Хемминга
 w(i)= exp(-0.0003*(i-shift)^2.0); %экспоненциальное окно
 end
end
for i=1:kt %генерация задержанного радиоимпульса
 y(i)=w(i)*cos(2*3.14*(i-(shift))/75);
 y(i)=y(i)+Q*noise(i);


```

    end
i=1:kt;
plot(i,y);
title('y');
%нахождение макс. знач. y[i] массива Y
C=max(y);
%нахождение номера элемента массива Y, соответствующего макс. знач. y[i]
for i=1:kt
    if (y(i)==C)
        shiftmax=i-1;
        break
    end
end
if(shiftmax<0)|(shiftmax>=550000)
    dt_int=shiftmax;
else
    dt=shiftmax; %оценка времени задержки по методу максимума амплитуды
    search_area=0.1; %область поиска относительно центра
    korpred=0;
    chast_intpred=0;
    for ki=1:3 %Цикл определяет количество итераций
        shagkor=dt*search_area/3;
        k=0;
        dt1=dt-dt*search_area;
        dt2=dt+dt*search_area;
        %диапазон сдвигов должен быть ограничен - от 0 до 800
        if (dt1<0) dt1=0;end;
        if (dt2>800) dt2=800; end;
        for j=dt1:shagkor:dt2 %цикл для создания 6 эталонов в окрестности
            %приближенного значения сдвига, определенного по MAX амплитуды.
            k=k+1;
            xkor(k)=j;
            shift1=j;
            kor(k)=0;
            for i=1:kt
                x(i)=0;

```

```

end
%Вычисление массива эталонного радиоимпульса X
for i=1:kt %генерация оконной функции для эталонного радиоимпульса
    if(i>(shift1-h)&(i<=shift1+h))
        w1(i)=1-abs(i-shift1)/h;%окно Барлетта(треугольное)
        w1(i)=0.5*(1+cos(pi*(i-shift1)/h));%окно Хеннинга
        w1(i)=0.54+0.46*cos(pi*(i-shift1)/h);%окно Хемминга
        w1(i)=exp(-0.001*(i-shift1)^2.0); %экспоненциальное окно
    end
end
for i=1:kt %генерация эталонного радиоимпульса
    x(i)=w1(i)*cos(2*3.14*(i-(shift1))/75);
end
%вычисление средних значений X и Y
    x_sr=mean(x);
    y_sr=mean(y);
    x_sko=0;
    y_sko=0;
    kor1(k)=0; % начальное значение суммы модулей суммы
    kor(k)=0; % начальное значение коэф. корреляции
%вычисление СКО и коэффициента корреляции X и Y
    for i=1:kt
        %    x_sko=x_sko+(x(i)-x_sr)*(x(i)-x_sr);
        %    y_sko=y_sko+(y(i)-y_sr)*(y(i)-y_sr);
        %    kor(k)=kor(k)+(x(i)-x_sr)*(y(i)-y_sr);

        sxy(i)=(abs(x(i)-y(i))); %вычисление модуля разности
    %    sxy(i)=(abs(x(i)+y(i))); %вычисление модуля суммы
        kor1(k)=kor1(k)+sxy(i);%вычисление нормы
                                Минковского/Поддорогина
    end
    % kor1(k)=kor(k)/(sqrt(x_sko*y_sko));%вычисление коэффициента
    % корреляции X и Y
    kor(k)=kor1(k);
end
xx=1:k;

```

```

xi=1:0.1:k;
yint=interp1(xx,kor,xi,'spline');% сплайн-интерполяция коэф корреляции
r1=kor;
%следующие 5 строк - отображение графика функции коэф. корреляции/
%коэф. Минковского/коэф. Поддорогина от сдвига эталонов
%график получен с помощью ф-и сплайн-аппроксимации spaps
apr=spaps(xkor,kor,0.000001);
figure
fnplt(apr)
hold on
plot(xkor,r1,'ro')
hold off

% cmax=max(yint); %нахождение максимума функции коэф. корр./ковар.
cmax=min(yint); %нахождение минимума функции коэф. Минковского
for i=1:round((k-1)/0.1+1)
    if (yint(i)==cmax)
        dt_int=dt-dt*search_area+(i-1)*shagkor/10; %уточненное значение
врем.задержки по МАХ функции коэф. корр.
    end
end
dt=dt_int;
search_area=search_area/2;
end
end
shift_result=dt %вывод конечного результата

pause;
close all; %закрытие окон графического вывода
clear; %очистка Workspace

```

Работа 6. Подгонка кривых

Целью работы является освоение методики получения адекватного аналитического описания зависимости $Y=f(X)$, по данным, содержащим случайные погрешности.

Теоретические основы

Обработка результатов измерений, содержащих случайные погрешности, включает 3 этапа:

1. Выбор вида аппроксимирующей зависимости.
2. Расчет коэффициентов аппроксимирующей зависимости.
3. Проверку адекватности полученной зависимости экспериментальным данным.

Наиболее совершенным программным средством для решения данной задачи является пакет программ MATLAB. MATLAB является средой разработки программ обработки данных и, одновременно, содержит большое количество готовых программ, в частности, программу подгонки кривых CurveFitting.

Программа подгонки кривых CurveFitting предоставляет следующие виды функций для аппроксимации и интерполяции одномерных массивов данных:

а) экспоненциальную;

$$y = ae^{bx}$$
$$y = ae^{bx} + ce^{dx}$$

б) гауссиан

$$y = \sum_{i=1}^n a_i e^{\left[-\left(\frac{x-b_i}{c_i}\right)^2\right]}$$

$$n=1,2,\dots,10$$

в)Фурье

$$y = a_0 + \sum_{i=1}^n a_i \cos(nwx) + b_i \sin(nwx)$$
$$n=1,2,\dots,10$$

г)полиномиальную

$$y = \sum_{i=1}^{n+1} p_i x^{n+1-i}$$
$$n=1,2,\dots,10$$

д)показательную

$$y = ax^b$$
$$y = a + bx^c$$

е)рациональную

$$y = \frac{\sum_{i=1}^{n+1} p_i x^{n+1-i}}{x^m + \sum_{i=1}^m q_i x^{m-i}}$$
$$n, m=1,2,\dots,10$$

Критерием качества подгонки являются вычисляемые программой коэффициенты множественной детерминации (R-square и Adjusted R-square). Коэффициент R-square показывает насколько разброс данных относительно аппроксимирующей кривой может быть объяснен наличием случайных погрешностей в данных. Максимально возможное значение R-square равно 1.

Коэффициент Adjusted R-square характеризует то же самое, но учитывает количество степеней свободы дисперсий воспроизводимости и адекватности. Поэтому Adjusted R-square лучше характеризует степень соответствия экспериментальных данных аппроксимирующей кривой в том случае, если мы увеличиваем степень аппроксимирующего полинома и хотим проверить, происходит ли при этом улучшение качества подгонки. Максимально возможное значение Adjusted R-square также равно 1.

$$SSE = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2 \quad \text{SSE – sum square error}$$

$$SST = \sum_{i=1}^n w_i (y_i - \bar{y})^2 \quad \text{SST – sum square total,}$$

$$w_i = \frac{1}{\sigma_i^2} \quad \text{- коэффициент, выравнивающий степень разброса данных в опытах}$$

\hat{y} – значение полученное по аппроксимирующей зависимости

$$R - square = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$Adjusted R - square = 1 - \frac{SSE(n-1)}{SST(n-m)}$$

n- количество результатов измерений;

m – количество членов в уравнении регрессии;

n-1 – количество степеней свободы дисперсии SSE;

n-m - количество степеней свободы дисперсии SST.

Программа работы

6. Освоить технологию работы с программой CurveFitting пакета . MATLAB.
7. Научиться производить обоснованный выбор наилучшей аппроксимирующей зависимости из возможных.

Указания к выполнению

1. При выполнении п.1. Программы изучите теоретические основы подгонки кривых, положенные в основу программы Curve Fitting и описание библиотеки аппроксимирующих функций Fitting Library в MATLAB.
2. При выполнении п.2 выполните подгонку кривых, выберите наилучшую функциональную зависимость и обоснуйте сделанный выбор для зашумленного массива данных: файл censuN.mat
3. При работе с программой CurveFitting подгонку кривых производите в следующей последовательности:

а) Запустите MATLAB;

б) откройте окно Workspace (View/Workspace) и загрузите в это окно mat-файл данных по заданию преподавателя;

в)откройте панель CurveFitting для чего выберите Start/ Toolboxes/ CurveFitting в окне MATLAB;

г) на панели CurveFittingTool активизируйте кнопку Data. В окнах Xdata, Ydata станут доступными составляющие cdate (X) и pop (Y) из файла census.mat;

д) произведите выбор cdate и pop, затем активизируйте кнопку CreateDataSet;

е) активизируйте кнопку Fitting;

ж) выберите вид аппроксимирующей функции и активизируйте кнопку Apply;

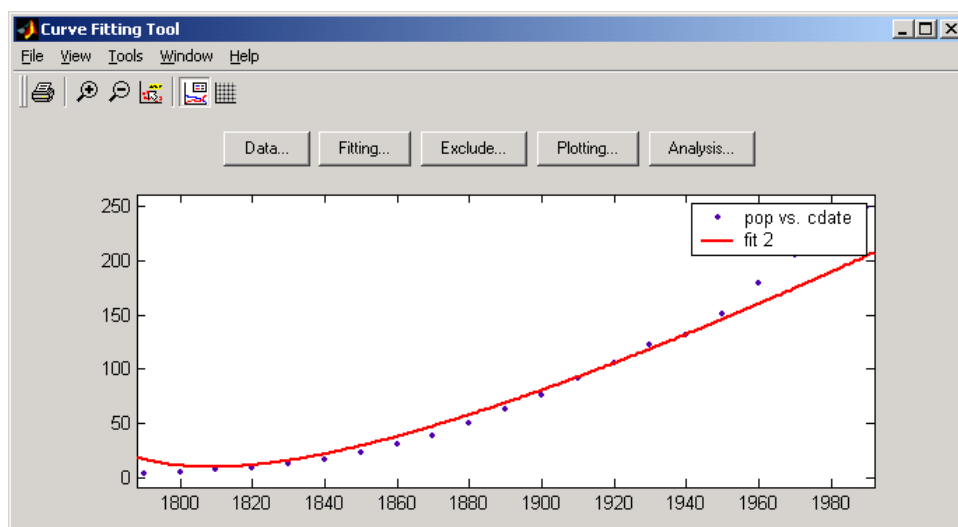


Рис. 6.1

Результаты аппроксимации будут представлены в графической форме в окне графического вывода на панели CurveFittingTool (рис.6.1) и в численной форме в окне Results на панели Tables of Fits (рис. 6.2).

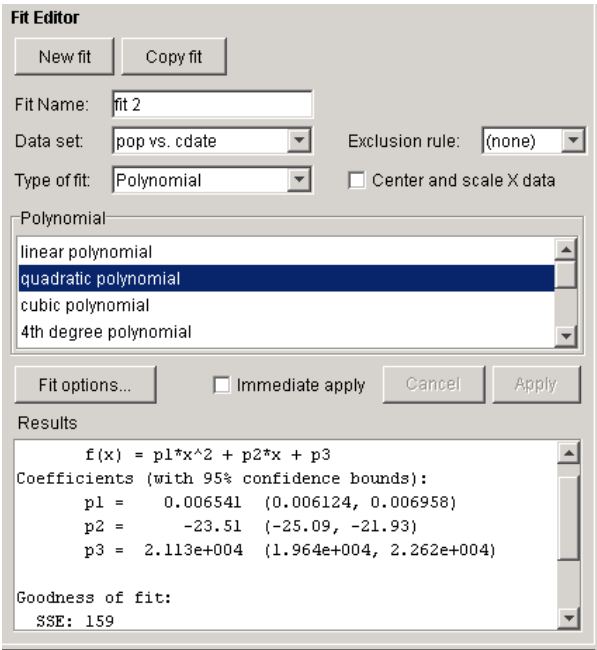


Рис. 6.2

з)выберите пункт меню View/Residuals/Line или View/Residuals/Scatter. В результате в окне графического вывода на панели CurveFittingTool будут одновременно выведен график ошибки интерполяции (см. пример на рис. 6.3).

Окно графического вывода можно вывести на печать или выполнить в виде рисунка: File/Print to Figure.

Полученные в процессе аппроксимации результаты занесите в таблицу:

Табл. 6.1.

Вид модели	Порядок модели	Качество подгонки (+ / -)	Диапазон погрешностей	R-square	Adjusted R-square	RMSE

Исследуйте таким образом все возможные виды моделей, порядок модели - от 1 до 5.

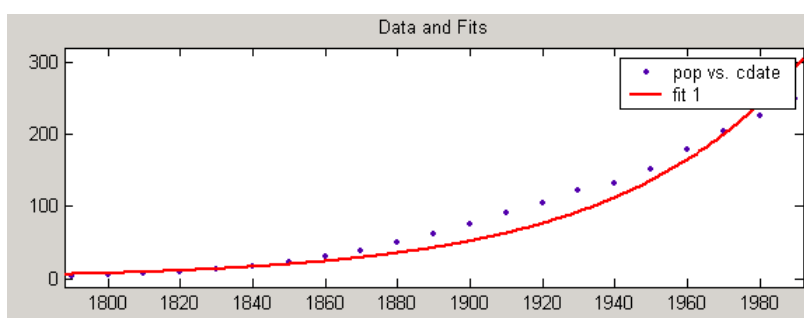
е) по полученным и отраженным в таблице данным выберите три наилучших варианта аппроксимирующих функций. Для этих вариантов сохраните график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения с численными значениями коэффициентов для аппроксимирующей зависимости.

4. Оценку качества подгонки кривых и окончательный выбор аппроксимирующей зависимости производите последовательно:

1. Выберите «хорошие» модели.

1.1. Визуально, по степени согласованности графика полученной в результате подгонки аппроксимирующей кривой с отображенными в этом же окне графического вывода значениями y_i . Качество подгонки считается удовлетворительным, если значения y_i «вытянуты» вдоль аппроксимирующей кривой. На рис.6.3 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).

А



Б

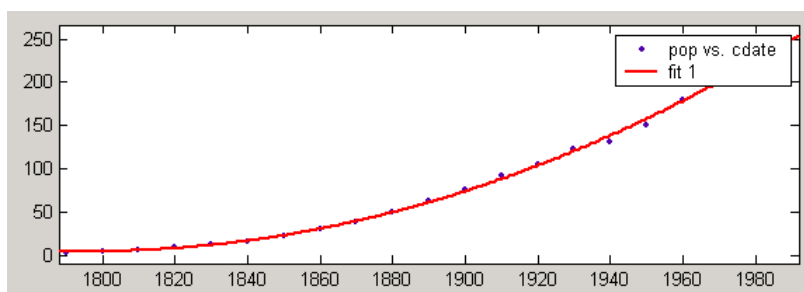


Рис. 6.3.

1.2. Визуально, по графику разности между значениями y_i и вычисленными по аппроксимирующей функции значениями $f(x_i)$. Качество подгонки считается удовлетворительным, если график разности хорошо аппроксимируется «на глаз» прямой $y=0$. На рис.6.4 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).

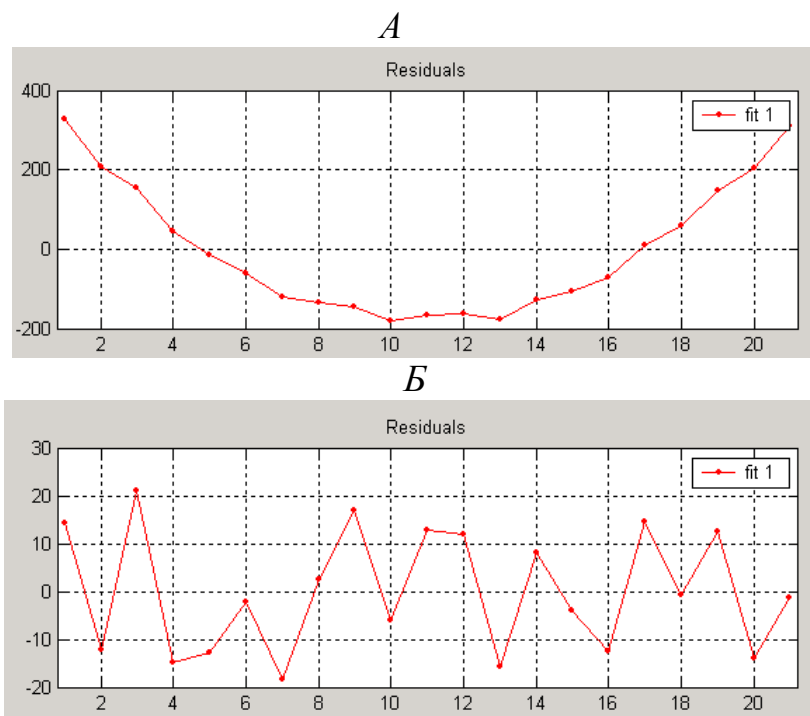


Рис. 4.

2. Выберите лучшую кривую в каждом классе аппроксимирующих зависимостей – по показателю Adjusted R-square. Лучшей кривой в классе является та, которой соответствует максимальное значение Adjusted R-square. Если максимальное значение соответствует двум или более кривым – выбирается кривая с наиболее простым аналитическим описанием.
3. Окончательный выбор самой лучшей аппроксимирующей зависимости – выбрать самую простую модель из лучших с условием примерно тех же значений диапазона погрешностей, R-square и RMSE (Root Mean Squared Error).

Содержание отчета

1. Постановка задачи.
2. Результаты аппроксимации экспериментальных данных из файла и показатели качества подгонки R-square и Adjusted R-square для всех основных классов аппроксимирующих функций в форме таблицы.
3. Для трех наиболее подходящих аппроксимирующих зависимостей - график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения для аппроксимирующей зависимости.
4. Обоснование выбора наилучшей аппроксимирующей зависимости.
5. Выводы.

Работа 7. Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга.

Целью лабораторной работы является изучение методики частотно-временного анализа нестационарных сигналов на основе эмпирического метода декомпозиции Хуанга и частотно-временного анализа на основе преобразования Гильберта.

Задание к работе

Имеется модельный сигнал, представляющий собой зашумленный гармонический сигнал с частотой, изменяющейся по линейному, квадратичному, кубическому закону или сумму двух зашумленных гармонических сигналов с изменяющейся частотой. Требуется исследовать условия их адекватного частотно-временного представления.

Теоретические основы

Под преобразованием Гильберта-Хуанга (Hilbert-Huang transform – ННТ) понимается эмпирический метод декомпозиции (EMD) нелинейных и нестационарных процессов и Гильбертов спектральный анализ (HSA). ННТ представляет собой частотно-временной анализ данных (сигналов) и не требует априорного функционального базиса. Функции базиса получаются адаптивно непосредственно из данных процедурами отсеивания функций «эмпирических мод». Мгновенные частоты вычисляются от производных фазовых функций Гильбертовым преобразованием функций базиса. Заключительный результат представляется в частотно-временном пространстве

EMD-HSA был предложен Норденом Хуангом в 1995 в США (NASA) для изучения поверхностных волн тайфунов, с обобщением на анализ произвольных временных рядов коллективом соавторов в 1998 г. В последующие годы, по мере расширения применения EMD-HSA для других отраслей науки и техники, вместо термина EMD-HSA был принят более короткий термин преобразования: ННТ.

EMD (Empirical Mode Decomposition) - метод разложения сигналов на функции, которые получили название «эмпирических мод». Метод представляет собой итерационную вычислительную процедуру разложения

исходных данных (непрерывных или дискретных сигналов) на эмпирические моды или внутренние колебания (intrinsic mode functions, IMF).

IMF представляет собой колебательный режим, как часть простой гармонической функции, но вместо постоянной амплитуды и частоты, как в простой гармонике, у IMF могут быть переменная амплитуда и частота, как функции времени.

IMF, определенные вышеприведенным способом, допускают вычисление физически значимых мгновенных частот, что дает возможность создать частотно-временное представление сигнала на основе преобразования Гильберта.

Преобразование Гильберта позволяет разложить исходный процесс на две составляющие: амплитудную и фазовую. Амплитуда не меняется во времени, а фаза линейно растет с коэффициентом пропорциональности, который носит название частоты. Частота определяет число периодов (повторений значений функции) в единицу времени и является постоянной во времени величиной. Амплитуда и фаза гармонической функции отражают различную информацию: амплитуда описывает энергию, а фаза характеризует повторяемость процесса во времени, и в этом смысле они могут рассматриваться как независимые характеристики гармонической функции. Поскольку фаза зависит от времени произвольным образом, то вводится понятие мгновенной частоты.

—

Где $\phi(t)$ — это фаза процесса.

Преобразование Гильберта действительной функции $x(t)$, определенной во всей временной области $-\infty < t < \infty$, есть действительная функция $\tilde{x}(t)$, также определенная на всей временной оси и задаваемая формулой:

$$\tilde{x}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau$$

Применение преобразования Гильберта позволяет выполнять квадратурную модуляцию сигналов, в каждой текущей координате модулированных сигналов производить определение огибающей и мгновенной фазы и частоты сигналов, выполнять анализ каузальных систем обработки сигналов.

Декомпозиция Хуанга

Декомпозиция основана на предположении, что любые данные состоят из различных внутренних колебаний. В любой момент времени данные могут иметь различные сосуществующие внутренние колебания, нанесенные одно на другое. Каждое колебание, линейное или нелинейное, представляет собой модовую функцию, которая имеет экстремумы и нулевые пересечения. Кроме того, колебание будет в определенной степени «симметрично» относительно локального среднего значения. Результат – конечные сложные данные.

Допустим, что имеется произвольный сигнал $x(t)$. Сущность метода EMD заключается в последовательном вычислении функций эмпирических мод $c_j(t)$ и остатков $r_j(t) = r_{j-1}(t) - c_j(t)$, где $j = 1, 2, 3, \dots, n$ при $r_0 = x(t)$. Результатом разложения будет представление сигнала в виде суммы модовых функций и конечного остатка.

Входной сигнал $x(t)$ раскладывается по базису, который удовлетворяет всем традиционным требованиям базиса. На основании проверки на модельных и опытных данных он является:

- законченным,
- сходящимся,
- ортогональным,
- единственным.

Также он является адаптивным, так как получен непосредственно из анализируемых данных эмпирическим методом.

Алгоритм эмпирической декомпозиции сигнала⁴

Операция 1. Идентифицируем по координатам и амплитудам все локальные экстремумы (максимумы и минимумы) сигнала. Группируем отдельно массивы векторов координат (номеров отсчетов) $x_{\max}(k)$ и соответствующих амплитудных значений $y_{\max}(k)$ максимумов, и аналогичные массивы векторов $x_{\min}(k)$ и $y_{\min}(k)$ минимумов всех выделенных экстремумов.

⁴ Описание алгоритма и иллюстрации взяты из [6]

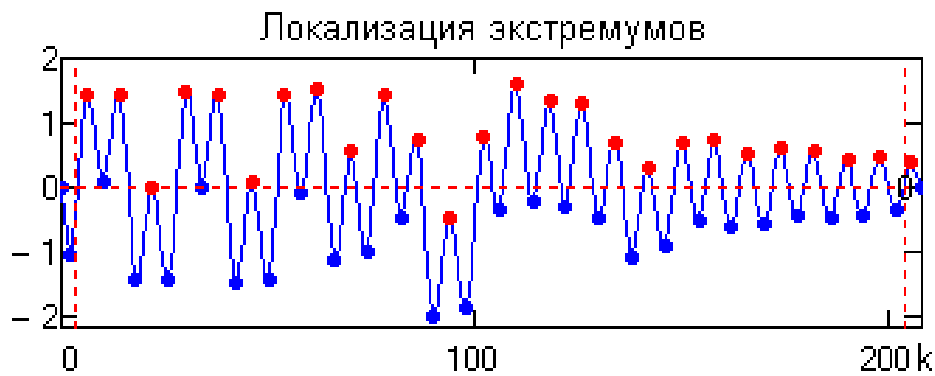


Рис. 7.1

Операция 2. Кубическим (или каким либо другим) сплайном вычисляем верхнюю и нижнюю огибающие сигнала по выделенным максимумам и минимумам, как это показано на рисунке (красный и синий цвет соответственно). Определяем функцию средних значений $m_1(k)$ между огибающими (черный цвет) и находим первое приближение к первой функции IMF:

$$h_1(k) = y(k) - m_1(k).$$

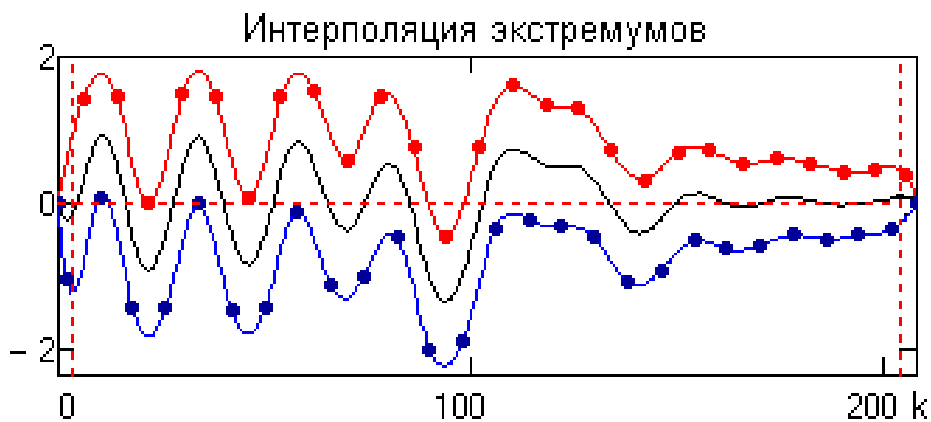


Рис. 7.2.

Операция 3. Повторяем операции 1 и 2, принимая вместо $y(k)$ функцию $h_1(k)$, и находим второе приближение к первой функции IMF – функцию $h_2(k)$.

$$h_2(k) = h_1(k) - m_2(k).$$

Аналогично находим третье и последующие приближения к первой функции IMF. По мере увеличения количества итераций функция $m_i(k)$, равно как и функция $h_i(k)$, стремится к неизменяемой форме. С учетом этого, естественным критерием останова итераций является задание определенного предела по нормализованной квадратичной разности между двумя последовательными операциями приближения, определяемой как

$$d = \frac{\sum_k |h_i(k) - h_{i-1}(k)|}{\sum_k h_{i-1}^2(k)}$$

Результат разложения

Последнее значение $h_i(k)$ итераций принимается за наиболее высокочастотную функцию $c_1(k) = h_i(k)$ семейства IMF, которая непосредственно входит в состав исходного сигнала $y(k)$. Это позволяет вычесть $c_1(k)$ из состава сигнала и оставить в нем более низкочастотные составляющие:

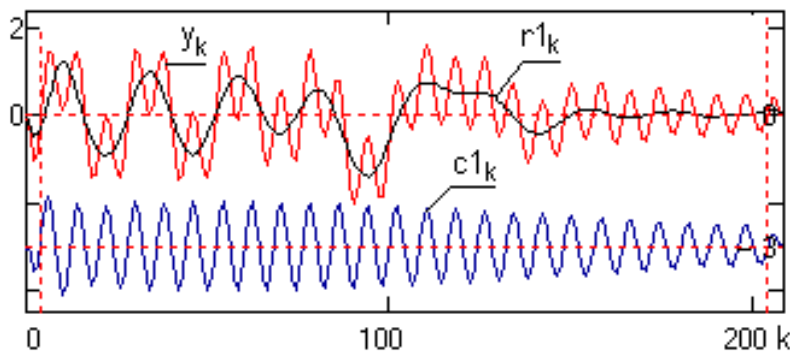


Рис. 7.3.

$$r_1(k) = y(k) - c_1(k).$$

Функция $r_1(k)$ обрабатывается как новые данные по аналогичной методике с нахождением второй функции IMF – $c_2(k)$, после чего процесс продолжается:

$$r_2(k) = r_1(k) - c_2(k), \text{ и т.д.}$$

Таким образом, достигается декомпозиция сигнала в n – эмпирическом приближении: $y(t) = c_n(t) + r_n(t)$.

Критерии останова процесса декомпозиции

1. Остаток $r_n(k)$ во всем интервале задания сигнала становятся незначительными по своим значениям по сравнению с сигналом.
2. Остаток $r_n(k)$ становится монотонной функцией, из которой больше не может быть извлечено функций IMF.
3. Так как в конечном итоге суммирование всех функций IMF (реконструкция сигнала) должно давать исходный сигнал, то можно останавливать разложение заданием относительной погрешности среднеквадратической реконструкции (без учета остатка $r_n(k)$).
4. По мере увеличения количества функций IMF относительная среднеквадратическая погрешность реконструкции достаточно сложных и протяженных сигналов уменьшается, но, как правило, имеет определенный минимум. По-видимому, это определяется попытками алгоритма разложить остаток на функции, частично компенсирующие друг друга. Соответственно, останов программы может выполняться, если следующая выделенная функция IMF увеличивает погрешность реконструкции.

Практический критерий останова процесса декомпозиции

Другими словами, остановка декомпозиции сигнала должна происходить при максимальном «выпрямлении» остатка, т.е. превращения его в тренд сигнала по интервалу задания с числом экстремумов не более 3. Даже для данных с нулевым средним значением конечный остаток может отличаться от нуля. Чтобы применять метод EMD, центрирования данных не требуется, метод нуждается только в локализациях экстремумов. Нулевая линия для каждого компонента декомпозиции будет сформирована процессом отсеивания.

Очистка сигналов от шумов с использованием разложения Хуанга

Особенностью сигналов, получаемых при физических экспериментах, является то, что они, как правило, нестационарные и значительно зашумлены. Применение традиционных способов очистки сигналов от шумов с помощью частотных фильтров (сглаживающих, медианных, Баттерворта и др.) приводят к значительному искажению формы сигнала, что может приводить к ошибочной физической интерпретации изучаемого процесса.

Известен другой способ очистки сигналов от шумов, основанный на использовании эмпирической модовой декомпозиции Хуанга [7,8]. Способ заключается в разложении исходного сигнала на составляющие (моды), затем в исключении первых составляющих (от 1 до 3), которые обычно представляют собой шум, содержащийся в исходном сигнале.

Преобразование Гильберта

Одним из достоинств преобразования Гильберта является то, что с помощью него можно $\varphi(t) = \arctan \left[\frac{Hs(t)}{s(t)} \right]$ вычислить мгновенную частоту сигнала.

Для этого используется понятие аналитического сигнала. Аналитический сигнал – комплексный сигнал $z(t) = s(t) + jHs(t)$, где $s(t)$ – исходный сигнал, который представляется в качестве действительной части комплексного сигнала, $Hs(t)$ – мнимая часть и является преобразованием Гильберта исходного сигнала $s(t)$. Тогда амплитуда сигнала $a(t)$ и фаза $\varphi(t)$ вычисляются по формулам:

$$a(t) = \sqrt{s(t)^2 + Hs(t)^2}$$

Мгновенная частота находится путем вычисления производной от фазы по времени. На рис.7.4 приведен пример частотно-временного представления сигнала с линейной частотной модуляцией.

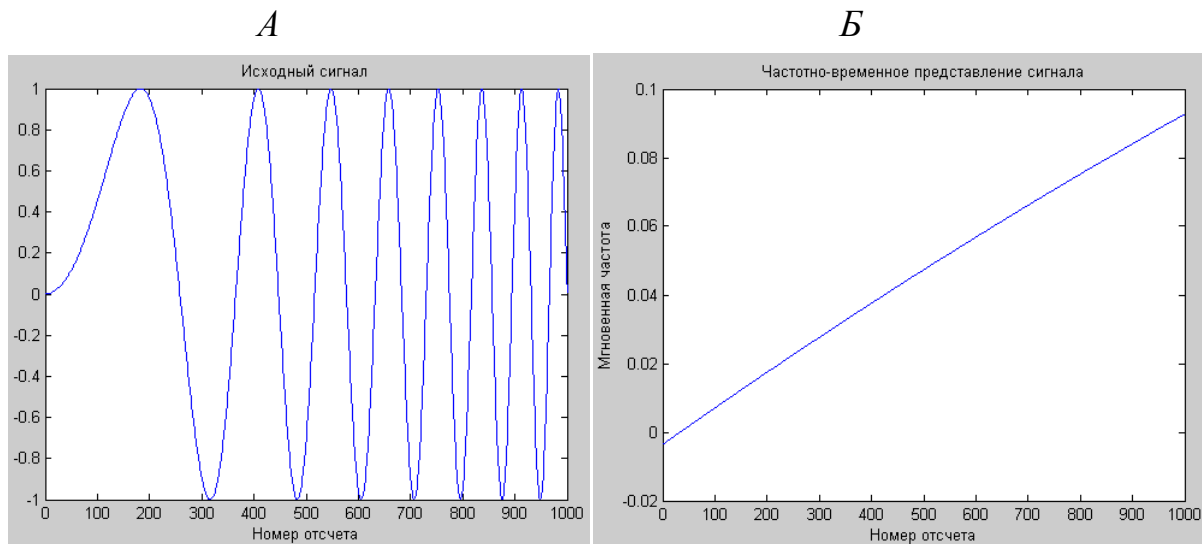


Рис. 7.4. Сигнал с линейной частотной модуляцией (А) и его частотно-временное представление (Б).

Указания к выполнению работы

1. Для изучения эмпирической декомпозиции Хуанга возьмите за основу базовую программу (см. Приложение 1) и выполните декомпозицию для трех видов сигналов:

- суммы двух стационарных гармонических сигналов;
- нестационарного гармонического сигнала;
- зашумленного гармонического сигнала.

В первом случае представляет интерес возможность разделения гармонических составляющих сигналов в функциях IMF.

Во втором и третьем случае представляет интерес определение зависимости оптимального количества удаляемых функций IMF для очистки сигнала от шума в зависимости от уровня шума в исходном сигнале.

2. Для изучения преобразования Гильберта возьмите за основу базовую программу (см. Приложение 2)

Содержание отчета

1. Задание к работе.
2. Тексты программ эмпирической декомпозиции Хуанга и частотно-временного преобразования Гильберта.
3. Графики исходных сигналов и результатов обработки.
4. Интерпретация полученных результатов обработки.
5. Выводы.

Приложение 1. Текст базовой программы модовой декомпозиции Хуанга

%Эмпирическая модовая декомпозиция Хуанга.

%Используется для очистки сигналов от шумов

clc;%очистка Command Window

N=2401;%количество точек

d=0.005;%шаг изменения переменной time

M=(N-1)*d;%

time = 0:d:M;

Q=0.2;%уровень шума в долях СКО

NF=3;%кол-во функций разложения, содержащих шум (NF=1..3)

```
KP1=5;%кол-во периодов первого сигнала
KP2=7;%кол-во периодов второго сигнала
```

%ПРОГРАММНАЯ ГЕНЕРАЦИЯ СИГНАЛА

```
noise=rand(N);%генерация массива нормально распределенного шума
% noise=wgn(kf,1,0);%генерация массива белого Гауссова шума
for k=1:N % генерация гармонического сигнала
s(k) = sin(2*pi*KP1*k/N)+ sin(2*pi*KP2*k/N);% 1-й вид сигнала
f(k)=s(k)+ Q*noise(k); % суммирование сигнала и шума
end
%Генерация нестационарного сигнала
s = 4*sin(10*time.*time./100.0);
f = s(1:N)+ Q*noise(1:N);
% Чтение сигнала из файла
% filename = sensor_signal.xls';%3-й вид сигнала
% x = xlsread(filename);%3-й вид сигнала
% f = x(1:N);
% f(N) = f(N-1);
% f = zeros(1,1+M/d);
% i=1:1+M/d;
% f(i)=x(i);
figure
plot(f);
title ('Исходный сигнал')
```

%Обнуление матрицы остатков

```
r = zeros(10,1+M/d);
r(1,:) = f;%Операция создания 1-й строки матрицы остатков r из массива f
%первая строка матрицы остатков равна самому модельному сигналу
```

%ПОЛУЧЕНИЕ ФУНКЦИЙ РАЗЛОЖЕНИЯ

```
for j = 1:10 % 10 - максимальное количество функций разложения IMF
h = r(j,:);%r(j,:)-операция выделения строки j из матрицы r
maxLastCount = -10;
for i = 1:N % N - максимальное количество экстремумов
```

```

[ymax,imax,ymin,imin] = extrema(h);%нахождение массивов локальных
максимумов
%i минимумов с помощью стандартной функции extrema, имеющейся в
MATLAB
if((length(imax) < 2) || (length(imin) < 2))
    break;
end;
kUp = spline(time(imax),ymax,time);%сплайн-интерполяция локальных
максимумов
kDown = spline(time(imin),ymin,time);%сплайн-интерполяция локальных
минимумов
kMean = (kUp + kDown)/2;
h = h - kMean;
[one, maxCount] = size(ymax);
if(abs(maxCount - maxLastCount) < 4)
    break;
end;
maxLastCount = maxCount;
end;
c(j,:) = h;

%Формирование искусственных экстремумов в начале и в конце массива
c(j,1)=c(j,2);% 1-ый элемент массива Cj устанавливается равным 2-му
c(j,2)=c(j,3)+0.1*abs(c(j,3));% 2-ой элемент массива Cj равен 3-му + 10% от
модуля 3-го
c(j,3)=c(j,4)-0.1*abs(c(j,4));% 3-ий элемент массива Cj равен 4-му минус 10% от
модуля 4-го
c(j,N)=c(j,N-1); %N-й элемент массива Cj устанавливается равным N-1-му
c(j,N-2)=c(j,N-3)+0.1*abs(c(j,N-3)); %N-2-й элемент массива Cj равен N-3-му
+10% от модуля N-3-го
c(j,N-3)=c(j,N-4)-0.1*abs(c(j,N-4)) ; %N-3-й элемент массива Cj N-4-му минус
10% от модуля N-4-го

r(j + 1,:) = r(j,:) - c(j,:);
end;
%Вывод функций разложения IMF

```

```

figure; plot(c(1,:));title('IMF_1');
figure;plot(c(2,:)); title('IMF_2');
figure; plot(c(3,:));title('IMF_3');
figure;plot(c(4,:));title('IMF_4');
figure; plot(c(5,:));title('IMF_5');
figure;plot(c(6,:));title('IMF_6');
figure;plot(c(7,:));title('IMF_7');

%Вывод остатков
figure; plot(r(1,:)); title('Остаток_1');
figure;plot(r(2,:));title('Остаток_2');
figure;plot(r(3,:));title('Остаток_3');
figure;plot(r(4,:));title('Остаток_4');
figure;plot(r(5,:));title('Остаток_5');
figure;plot(r(6,:));title('Остаток_6');
figure;plot(r(7,:));title('Остаток_7');
C1 = zeros(1,1+M/d);
%Суммирование функций разложения, содержащих только шум
for i = 1:NF
    C1=C1+c(i,:);%накопление шумовых функций IMF
end
i=1:N;
f1(i)=f(i)-C1(i);%исходный зашумленный сигнал за вычетом шумовых IMF
%f1(i)=r(kki,i);
figure;plot(i,f(i),'r-');title('Исходный зашумленный сигнал');
hold on;
%figure
plot(i,f1(i));title('Отфильтрованный сигнал');
hold off;
% Следующий блок программы комментируется, если зашумленные
% данные берутся из файла (в этом случае
% чистый незашумленный сигнал s неизвестен)
for i=1:N
    DZ(i)=f1(i)-s(i);%уровень зашумления в сигнале после
    %вычитания функций разложения, содержащих только шум
end

```

```
SKO=std(DZ)%CKO полной погрешности
KSKO=Q/SKO %коэффициент подавления шума
pause;
close all; %закрытие всех окон графического вывода
clear; %очистка Workspace
```

Приложение 2.Текст базовой программы преобразования Гильберта

```
%Частотно-временной анализ нестационарного сигнала
%с помощью преобразования Гильберта
n=1000; % количество отсчетов
k=5 ; % количество периодов
h=zeros(1,n);
for i=1:n
    h(i)= sin(3*pi*k*i*i/n/n);%линейная частотная модуляция (ЛЧМ)
    % h(i)=sin(2*pi*k*i*i*i/n/n/n); %квадратичная частотная модуляция
end
figure
plot(h)
title('Исходный сигнал')
xlabel('Номер отсчета');
hx=hilbert(h); %преобразование Гильберта
phi = angle(hx); %фаза
phi2 = unwrap(phi);
p = polyfit(0:n-1,phi2,2); %аппроксимация полиномом второй степени
dp = polyder(p); %производная полинома
f = polyval(dp, 0:n-1); %Вычисление значений полинома
figure
plot(f)
title('Частотно-временное представление сигнала')
xlabel('Номер отсчета');
ylabel('Мгновенная частота');
pause
close all;
clear;
```

Работа 8. Базовые средства фильтрации шумов на изображениях

Целью лабораторной работы является изучение методики использования базовых средств фильтрации шумов на изображениях в среде MATLAB, включающих усредняющий и медианный фильтры, фильтр Гаусса, адаптивный фильтр Винера, фильтр повышения резкости и ранговый фильтр и оценки их сравнительной эффективности.

Задание к работе

1. Исследуйте связь эффективности фильтрации зашумленных изображений с визуальным восприятием изображений и степенью сходства отфильтрованного изображения с исходным незашумленным изображением, найдите оптимальные параметры фильтров;
2. Исследуйте связь эффективности фильтрации от показателей яркости и контрастности исходного изображения;
3. Спроектируйте на внутреннем языке MATLAB программу цифровой фильтрации изображений, содержащей средства интерфейса пользователя, позволяющие выбирать вид и задавать параметры фильтрации.

Описание программного обеспечения и порядок работы

Программное обеспечение включает базовую программу фильтрации шумов и файлы фотоизображений. Программа фильтрации шумов позволяет загрузить два изображения, выполнить зашумление одного из них, произвести фильтрацию зашумленного изображения с помощью различных фильтров, вычислить коэффициент корреляции зашумленного и отфильтрованных изображений с исходным, выполнить построение 2D и 3D графика коэффициента корреляции. Виды и параметры шума и параметры фильтров могут задаваться. Текст программы приведен в приложении.

Указания к выполнению работы

1. При выполнении п.1 Программы установите в программе уровень шума 0.005, вид шума – Гауссов, запустите программу Filtering.m, произведите кадрирование изображения, выделив центральную часть лица, и зафиксируйте в таблице получаемые в окне Command Window при выполнении программы максимальные значения коэффициентов корреляции зашумленного изображения и отфильтрованных разными фильтрами изображений. Затем, изменяя параметры фильтров, добейтесь получения наиболее высоких значений коэффициентов корреляции отфильтрованных сигналов с исходным.

2. Повторите выполнение п. 1 при шуме `salt & pepper` с параметрами $d=0.01$, 0.05 и 0.1 и при мультипликативном шуме `speckle` с нулевым средним значением и среднеквадратичным отклонением $v=0.02$, 0.04 и 0.06 .

3. При выполнении п. 2 Программы спроектируйте на внутреннем языке MATLAB программу цифровой обработки изображений, реализующую уменьшение шумов на изображениях, используя стандартные базовые программы:

- `average` – усредняющий фильтр;
- `medfilt` – медианный фильтр;
- `gaussian` – фильтр Гаусса;
- `ordfilt` – ранговый фильтр;
- `unsharp` – фильтр повышения резкости;
- `wiener2` – адаптивный фильтр Винера;

Программа должна позволять производить интерактивный выбор вида фильтра и параметров фильтра.

Содержание отчета

1. Задание к работе.
2. Таблицы с показателями корреляционного сравнения изображений по п. 1 и 2 Указаний по выполнению работы с комментариями.
3. Текст программы по п. 3 Указаний по выполнению работы и экранная форма.
4. Выводы.

Приложение. Текст базовой программы фильтрации шумов

```
%Программа производит зашумление изображения
%фильтрацию зашумленного изображения с помощью
%различных фильтров, определение коэффициента корреляции
%зашумленного и отфильтрованных изображений с исходным
% img=imread ('m00200021.tif');
img2=imread ('m00200001.tif');
img=imread ('m00200117.tif');
% img3=imread ('m00200118.tif');
% img4=imread ('m00200119.tif');
% img5=imread ('m00200120.tif');
% img6=imread ('m00200121.tif');
I=imcrop(img);
figure;
imshow (img);
title('Исходное изображение');
img1=imnoise(img,'gaussian',0,0.015);% 0.005
figure;
imshow (img1);
title('Зашумленное изображение');

%Медианный фильтр 'medfilt2'
hsize=[3 3 ];
F4= medfilt2(img1,hsize);
figure;
imshow(F4);
title('После медианного фильтра');
pause;

%Усредняющий фильтр 'Average'
hsize=[3 3 ];
h= fspecial('average',hsize);
F1=imfilter(img1,h,'replicate');
figure;
```

```

imshow (F1);
title('После усредняющего фильтра');
pause;

ncorr = normxcorr2(I(:,:,1),img2(:,:,1));
figure, surf(ncorr), shading flat;
title('Сравнение с изображением другого человека');
max_c_other_man = max(abs(ncorr(:)))
pause;

ncorr = normxcorr2(I(:,:,1),img1(:,:,1));
figure, surf(ncorr), shading flat;
title('После зашумления');
max_c_noised = max(abs(ncorr(:)))
pause;

ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
figure, surf(ncorr), shading flat;
title('После медианного фильтра');
max_c_median = max(abs(ncorr(:)))
pause;

ncorr = normxcorr2(I(:,:,1),F1(:,:,1));
figure, surf(ncorr), shading flat;
title('После усредняющего фильтра');
max_c_average = max(abs(ncorr(:)))
pause;

%Фильтр Гаусса 'gaussian'
hsize=[9 9];
sigma=0.99;
h= fspecial('gaussian',hsize,sigma);
F2=imfilter(img1,h,'replicate');
figure;
imshow(F2);title('После фильтра Гаусса');
pause;

```

```
ncorr = normxcorr2(I(:,:,1),F2(:,:,1));
figure, surf(ncorr), shading flat;
title('После фильтра Гаусса');
max_c_gaussian = max(abs(ncorr(:)))
pause;
```

```
%Фильтр Лапласа 'laplacian'
alpha=0.5;
h= fspecial('laplacian',alpha);
F3=imfilter(img1,h,'replicate');
figure;
imshow(F3);title('После фильтра Лапласа');
pause;
```

```
ncorr = normxcorr2(I(:,:,1),F3(:,:,1));
figure, surf(ncorr), shading flat;
title('После фильтра Лапласа');
max_c_laplasian = max(abs(ncorr(:)))
pause;
```

```
%Фильтр повышения резкости 'unsharp'
alpha=0.2;
h= fspecial('unsharp',alpha);
F4=imfilter(img1,h);
figure;
imshow(F4);title('После фильтра unsharp');
pause;
```

```
ncorr = normxcorr2(I(:,:,1),F4(:,:,1));
figure, surf(ncorr), shading flat;
title('После фильтра unsharp');
max_c_unsharp = max(abs(ncorr(:)))
pause;
```

```
%Ранговый фильтр
```

```

m=5;n=5;
J = imnoise(img,'gaussian',0,0.015);
imshow(J);
J6 = ordfilt2(J, 7, ones(4,4));
figure;
imshow(J6);title('После рангового фильтра');
pause;

ncorr = normxcorr2(I(:,:,1),J6(:,:,1));
figure, surf(ncorr), shading flat;
title('После рангового фильтра');
max_c_ordfilt2 = max(abs(ncorr(:)))
pause;

%Фильтр Винера
m=5;n=5;
[J,noise] = wiener2(img1,[m n]);
J6 = wiener2(J,[5 5]);
figure;
imshow(J6);title('После фильтра Винера');
pause;

ncorr = normxcorr2(I(:,:,1),J6(:,:,1));
figure, surf(ncorr), shading flat;
title('После фильтра Винера');
max_c_wiener = max(abs(ncorr(:)))
pause;

close all; %закрытие всех окон графического вывода
clear; %очистка Workspace

```

Работа 9. Средства идентификации объектов на изображениях

Целью лабораторной работы является изучение методики разработки программ цифровой корреляционной обработки изображений, включающей корреляционное сравнение изображений, формирование порогового правила для идентификации объектов на изображениях.

Задание к работе

Имеется набор изображений центральной части лица нескольких человек. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки изображений, реализующую сравнение изображения человека с различными изображениями других людей из базы данных или различными изображениями того же человека, формирование порогового правила, позволяющего отличать «своего» от «чужого».

Описание программного обеспечения и порядок работы

Основные программы системы лабораторной работы функционируют в среде MATLAB.

Программное обеспечение включает файлы 150 полноформатных фотоизображений, программу формирования эталонных фотоизображений и программу сравнения фотоизображений. Первый этап подготовки к работе системы заключается в создании базы данных эталонных изображений. В результате работы первой программы на основе каждого полноформатного изображения формируются 16 эталонных изображений центральной части лица, отличающиеся масштабом от исходного в пределах $\pm 10\%$.



Рис. 9.1

На рис. 9.1 приведен пример полноформатного фотоизображения, на рис. 9.2 – 16 эталонных фотоизображений центральной части лица, отличающихся масштабом в пределах $\pm 10\%$.

Для выполнения основной функции – сравнения фотоизображений нужно запустить вторую программу. Программа

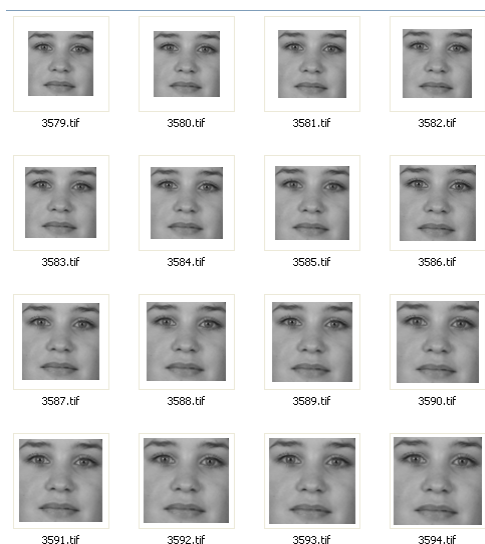


Рис. 9.2

производит корреляционное сравнение выбранного изображения с эталонными. Для исследования эффективности корреляционного принципа распознавания и формирования оптимального порога распознавания необходимо получить функции распределения коэффициентов корреляции при сравнении фотоизображений одних и тех же людей и разных людей.

В первом случае каждое исходное фотоизображение из базы данных масштабируется случайным образом, сравнивается с 16 эталонными изображениями этого же человека и находится максимальное значение коэффициента корреляции для каждого человека.

Во втором случае каждое исходное фотоизображение из базы данных масштабируется случайным образом, сравнивается с 16 эталонными изображениями остальных 149 людей и находится максимальное значение коэффициента корреляции для каждой пары сравниваемых фотоизображений.

Указания к выполнению работы

1. Для формирования эталонных изображений запустите программу формирования эталонных фотоизображений. При появлении окна с полноформатным изображением с помощью курсора выделите центральную часть лица, затем из контекстного меню выберите пункт **Imcrop**. Далее будет автоматически производиться программная генерация 16 эталонных изображений, отличающихся масштабом от исходного в пределах $\pm 10\%$.
2. Корреляционное сравнение текущих изображений с эталонными производите в следующем порядке:

- a) Произведите корреляционное сравнение выбранного изображения, масштаб которого измените случайным образом в пределах $\pm 10\%$ от исходного, с серией эталонных изображений, созданных при выполнении п. 1. Для этого запустите программу сравнения фотоизображений, после чего зафиксируйте результаты корреляционного сравнения, выводимые в окно Command Window (максимальное значение коэффициента корреляции). Аналогичные действия произведите для всех полноформатных изображений;
 - b) Произведите корреляционное сравнение выбранного изображения с серией эталонных изображений других людей, созданных при выполнении п. 1. Для этого запустите программу сравнения фотоизображений, после чего зафиксируйте результаты корреляционного сравнения, выводимые в окно Command Window (максимальные значения коэффициентов корреляции). Аналогичные действия произведите для всех полноформатных изображений.
3. По полученным массивам коэффициентов корреляции постройте интервальные гистограммы, затем массивы гистограмм аппроксимируйте непрерывными функциями, используя программу подгонки кривых Curve Fitting и произведите проверку правильности гипотезы о законе распределения по критерию χ^2 .
4. Произведите анализ результатов корреляционного сравнения изображений «свой - свой» и «свой - чужой» и на основании этого определите пороговое правило по принципу максимума правдоподобия для разделения по значению коэффициента корреляции «своих» и «чужих». Оценить вероятность ошибок 1-го (принять «своего» за «чужого») и 2-го рода (принять «чужого» за «своего»).
5. Вычислите порог корреляционного сравнения по принципу «идеального наблюдателя» и Неймана-Пирсона, считая допустимой ошибкой «пропуска чужого» - 10^{-3} .

Содержание отчета

1. Задание к работе
2. Текст программы с комментариями.
3. Таблицы полученных значений коэффициентов корреляции.
4. Расчет и обоснование порогового правила для различения «своих» и «чужих» по принципу максимума правдоподобия.

5. Расчет вероятности ошибок 1-го и 2-го рода.

6. Выводы.

Приложение 1. Критерий χ^2 с $k = l - r - 1$ степенями свободы.

$$\chi^2 = \sum_{j=1}^l \frac{(m_j - np_j)^2}{np_j}$$

где

r – число параметров распределения, определяемых по выборке;

m_j - табличная частота;

np_j - теоретич. частота;

p_j - вероятность функции распределения, вычисляемая по формуле:

$$P_i^{\text{theor}} = \int_{x_{i-1}}^{x_i} f(x) dx$$

Табл.9.1

Критерий Пирсона (хи - квадрат) при уровне значимости
(0,01; 0,05; 0,1; 0,15; 0,20; 0,25; 0,30)

n	P - 0,01	P - 0,05	P - 0,1	P - 0,15	P - 0,2	P - 0,25	P - 0,3
1	6,6349	3,8415	2,7055	2,0723	1,6424	1,3233	1,0742
2	9,2103	5,9915	4,6052	3,7942	3,2189	2,7726	2,4079
3	11,3449	7,8147	6,2514	5,3170	4,6416	4,1083	3,6649
4	13,2767	9,4877	7,7794	6,7449	5,9886	5,3853	4,8784
5	15,0863	11,0705	9,2364	8,1152	7,2893	6,6257	6,0644
6	16,8119	12,5916	10,6446	9,4461	8,5581	7,8408	7,2311
7	18,4753	14,0671	12,0170	10,7479	9,8032	9,0371	8,3834
8	20,0902	15,5073	13,3616	12,0271	11,0301	10,2189	9,5245
9	21,6660	16,9190	14,6837	13,2880	12,2421	11,3888	10,6564
10	23,2092	18,3070	15,9872	14,5339	13,4420	12,5489	11,7807
11	24,7249	19,6751	17,2750	15,7671	14,6314	13,7007	12,8987
12	26,2169	21,0260	18,5493	16,9893	15,8120	14,8454	14,0111
13	27,6882	22,3620	19,8119	18,2020	16,9848	15,9839	15,1187
14	29,1412	23,6848	21,0641	19,4062	18,1508	17,1169	16,2221
15	30,5779	24,9958	22,3071	20,6030	19,3107	18,2451	17,3217
16	31,9999	26,2962	23,5418	21,7930	20,4651	19,3689	18,4179
17	33,4087	27,5871	24,7690	22,9770	21,6146	20,4887	19,5110
18	34,8053	28,8693	25,9894	24,1555	22,7595	21,6049	20,6014
19	36,1909	30,1435	27,2036	25,3288	23,9004	22,7178	21,6891
20	37,5662	31,4104	28,4120	26,4976	25,0375	23,8277	22,7745

Приложение 2. Программа создания эталонных изображений

%Программа создания эталонных изображений

%из фотоизображений 480x640 формата .tif

clc;

for i=1:10

t=num2str(i);%номер исходного фотоизображения

password=num2str(i);%номер группы эталонов

path='D:\LifeCamFiles\P1\';

fname=strcat(path,t,'.tif');

J=imread (fname);%Чтение файла с изображением для создания

%группы эталонов

J3=imresize(J,0.5,'bicubic');%масштабирование

figure; imshow(J3);

I=imcrop(J3);%интерактивное выделение центральной части лица

figure; imshow(I);

path='D:\LifeCamFiles\';

h = waitbar(0,'Please wait...');

for i=1:16

 t=num2str(10+i);

 k=0.84+i*0.02;

 J=imresize(I,k,'bicubic');

 fname=strcat(path,password,t,'.tif');

 imwrite(J,fname);

% figure;imshow(J);

 waitbar(i/16)

end

pause(2);

set(0,'ShowHiddenHandles','on')%автоматическое удаление всех figures

close all;

clear;

end

Приложение 3. Программа корреляционного сравнения изображений

```
%Программа корреляционного сравнения полноформатного
% фотоизображения с эталонными
clc;
msgbox('Введите номер полноформатного фотоизображения','Выбор номера
изображения','warn');
answer=input('Введите номер фотоизображения ','s');
t=char(answer);
msgbox('Введите номер группы эталонов ','Выбор группы эталонов','warn');
answer=input('Введите номер группы эталонов ','s');
password=char(answer);
path='D:\LifeCamFiles\P1\';
fname=strcat(path,t,'.jpg');
I=imread(fname);

%Изменение масштаба полноформатного изображения по нормальному закону
% 100 - количество случайных значений, 0,05 - СКО,
% макс. отклонение с доверит. вероятностью 0,95 - 3*СКО
k=abs(1.0+ 0.05*(randn(size(100)))));
J=imresize(I,k,'bicubic');%изменение масштаба

X1=imresize(J,0.5,'bicubic');
figure;imshow(X1);
path='D:\LifeCamFiles\';
h = waitbar(0,'Please wait...');
for i=1:16
    t=num2str(78+i);
    fname=strcat(path,password,t,'.tif');
    X2=imread(fname);
    figure;imshow(X2);
    % изменен порядок сравнения: X2 с X1
    ncorr = normxcorr2(X2(:,1),X1(:,1));
    % figure, surf(ncorr), shading flat;
    max_c = max(abs(ncorr(:)));
    y(i)=max_c;
```

```
        waitbar(i/16)
    end

    set(0,'ShowHiddenHandles','on')%автоматическое удаление всех figures
    delete(get(0,'Children'))%автоматическое удаление всех figures
    y(9)=0;
    corr=y;
    max_corr=max(y)
    set(0,'ShowHiddenHandles','on')%автоматическое удаление всех figures

    close all;
    clear;
```

Расчетные задания

Расчетное задание 1

1. Могут ли векторы v_1 и v_2 образовывать базис, если

$$v_1 = \left(\frac{\sqrt{3}}{2}, \frac{\sqrt{1}}{2}\right), \quad v_2 = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$$

1. Разложите вектор

$$f = \left(\frac{\sqrt{3}}{2}, \frac{5}{2}\right)$$

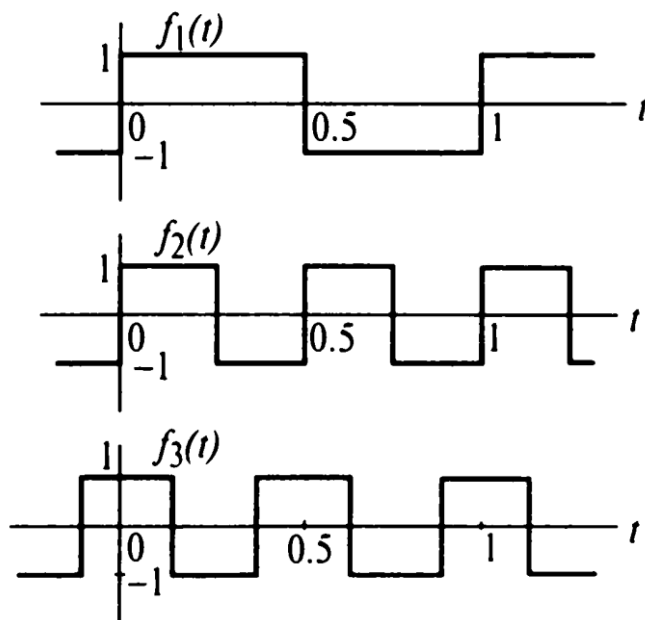
по базису $\{v_1, v_2\}$

2. Найдите расстояние между векторами

$$f = (4, -4, 7), \quad g = (3, -2, 6),$$

их скалярное произведение и коэффициент корреляции.

4. Докажите, что сигналы $f_1(t)$, $f_2(t)$, $f_3(t)$, изображенные на рисунке, взаимно перпендикулярны на отрезке $[0, 1]$



Расчетное задание 2

Вариант 1.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t)=t^2$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Получите аналитическое выражение для коэффициента корреляции R двух гармонических сигналов на интервале $[0, T]$, отличающихся по частоте (по количеству периодов на интервале определения):

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right),$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

Получите аналитическое выражение, постройте зависимость коэффициента корреляции от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = +/\!- 0.15$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией:

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right)w(t), \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right)w(t),$$

$$w(t) = \exp[-(n(t - T/2)/T)^2]$$

- 2**. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right)w(t) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right)w(t) + \xi(t)$$

где $\xi(t)$ – шум.

Расчетное задание 2

Вариант 2.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = |t|$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.
2. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов на интервале $[0, T]$, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right), \quad \text{shift1}, \text{shift2} \in [0, T]$$

где n – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$.

Получите аналитическое выражение и постройте зависимость коэффициента ковариации от относительного сдвига по времени δ :

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = \text{shift2} - \text{shift1}, \quad n = 9, \quad \delta \in [-0.05, 0.05]$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента ковариации R двух отличающихся сдвигом по времени гармонических сигналов, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right) w_2(t),$$

$$w_1(t) = \exp\left[-\left(\frac{m(t - \text{shift1})}{T}\right)^2\right], \quad w_2(t) = \exp\left[-\left(\frac{m(t - \text{shift2})}{T}\right)^2\right]$$

$m = 10$ – количество периодов несущей частоты радиоимпульса на интервале $[0, T]$

- 2**. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов, модулированных экспонентами и отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right) w_2(t) + \xi(t),$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

Расчетное задание 2

Вариант 3.

Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = |\sin(t)|$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.

В таблице представлены значения среднемесячных температур для двух городов F и G (измерения проводились через месяц).

Температура [°C]

i	1	2	3	4	5	6
Месяцы	1	3	5	7	9	11
Город F (f_i)	4	10	20	25	20	5
Город G (g_i)	20	18	15	10	17	22

Составьте таблицу и график значений функции взаимной корреляции температур атмосферы этих городов и прокомментируйте ее. Функция взаимной корреляции определяется соотношением:

$$R_{fg}(j) = \frac{1}{6} \sum_{i=1}^6 (f_i - \bar{f})(g_{i+j} - \bar{g}),$$

где \bar{f}, \bar{g} - среднеарифметические значения f_i и g_i .

Расчетное задание 2

Вариант 4.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = t \cdot \sin(t)$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Вычислите⁵ и постройте зависимость значения интеграла R модуля суммы двух гармонических сигналов, отличающихся по частоте (по количеству периодов на интервале $[0, T]$):

$$R(\delta) = \frac{1}{T} \int_0^T \left| \sin\left(\frac{2\pi n t}{T}\right) + \sin\left(\frac{2\pi n_2 t}{T}\right) \right| dt$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = + / - 0.15.$$

- 2*. Вычислите и постройте зависимость интеграла R модуля суммы двух гармонических сигналов, модулированных экспоненциальной оконной функцией $w(t)$, и отличающихся по частоте:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right) w(t), \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right) w(t),$$

$$w(t) = \exp\left(-\left(n(t - T/2)/T\right)^2\right)$$

- 2**. Вычислите и постройте зависимость значения интеграла R модуля суммы двух гармонических сигналов, отличающихся по частоте, в условиях, когда один из сигналов зашумлен:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right) w(t) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right) w(t) + \xi(t)$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

⁵ Использовать численное интегрирование методом трапеций. Пример программы приведен ниже
x1=0:0.01:2;
y1=x1.^2;
intl=trapz(x1,y1) %числ.интегр.методом трапеций

Расчетное задание 2

Вариант 5.

1. Вычислить аналитически коэффициенты разложения в ряд Фурье функции $f(t) = \exp(at)$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.
2. Вычислите⁶ и постройте на интервале $[0, T]$ зависимость значения интеграла R модуля разности двух гармонических сигналов $x(t)$ и $y(t)$, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени:

$$R(\delta) = \frac{1}{T} \int_0^T \left| \cos\left(\frac{2\pi n(t - shift1)}{T}\right) - \cos\left(\frac{2\pi n(t - shift2)}{T}\right) \right| dt, \quad shift1, shift2 \in [0, T]$$

где n – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$.

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = shift2 - shift1, \quad n = 9, \quad \delta = [-0.1, 0.1]$$

- 2*. Вычислите и постройте зависимость интеграла R от модуля разности двух гармонических сигналов, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$w_1(t) = \exp\left[-\left(\frac{m(t - shift1)}{T}\right)^2\right], \quad w_2(t) = \exp\left[-\left(\frac{m(t - shift2)}{T}\right)^2\right]$$

$m = 10$ – количество периодов несущей частоты радиоимпульса на интервале $[0, T]$

- 2**. Вычислите и постройте зависимость значения математического ожидания интеграла R от модуля суммы двух гармонических сигналов, отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - shift1)}{T}\right) w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - shift2)}{T}\right) w_2(t) + \xi(t),$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

⁶ Использовать численное интегрирование методом трапеций. Пример программы приведен ниже
x1=0:0.01:2;
y1=x1.^2;
int1=trapz(x1,y1) %числ.интегр.методом трапеций

Расчетное задание 2

Вариант 6.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t)=x*\exp[ax]$ на интервале $[-\pi,\pi]$ $(-\frac{T}{2},\frac{T}{2})$.
2. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени, на интервале $[0,T]$:

$$x(t) = \cos(\frac{2\pi n(t - shift1)}{T}) \quad \text{и} \quad y(t) = \cos(\frac{2\pi n(t - shift2)}{T}), \quad shift1, shift2 \in [0,T]$$

где n – количество периодов сигнала $x(t)$ и сигнала $y(t)$ на интервале $[0,T]$.

Получите аналитическое выражение, постройте зависимость от относительного сдвига по времени:

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = shift2 - shift1, \quad n = 5, \quad \delta = [-0.1, 0.1]$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов $x(t)$ и $y(t)$, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$w_1(t) = \exp[-(\frac{m(t - shift1)}{T})^2], \quad w_2(t) = \exp[-(\frac{m(t - shift2)}{T})^2]$$

- 2**. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов $x(t)$ и $y(t)$, отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos(\frac{2\pi n(t - shift1)}{T})w_1(t), \quad y(t) = \cos(\frac{2\pi n(t - shift2)}{T})w_2(t) + \xi(t),$$

где $\xi(t)$ – шум.

Расчетное задание 2

Вариант 7.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = x \cdot \cos(x)$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Получите аналитическое выражение для коэффициента корреляции R двух гармонических сигналов на интервале $[0, T]$, отличающихся по частоте (по количеству периодов на интервале определения):

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right),$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

Получите аналитическое выражение, постройте зависимость коэффициента корреляции от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = +/\!- 0.15$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right)w(t), \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t),$$

$$w(t) = \exp\left[-(n(t - T/2)/T)^2\right]$$

- 2**. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией, в условиях, когда один из сигналов зашумлен:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right)w(t) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t) + \xi(t)$$

где $\xi(t)$ – шум.

Литература

1. Сергиенко А.Б. Цифровая обработка сигналов. Учебник для вузов. 3-е изд.-СПб.: Питер, 2013.-768с.
2. Тутыгин В.С. Цифровая обработка коротких сигналов. Издательство Политехн. ун-та, 2012 – 164с.
3. А.И.Солонина, С.М.Арбузов. Цифровая обработка сигналов. Моделирование в MATLAB: Учебное пособие. – СПб.: БХВ-Петербург, 2008 – 816с. ISBN 978-5-9775-0259-7
4. Юкио Сато Без паники! Цифровая обработка сигналов: Пер. с яп.. – М.: Изд. дом Додэка – XXI, 2010 - 176с. ISBN 978-5-94120-251-5.
5. С.Л.Марпл-мл. Цифровой спектральный анализ и его приложения: Пер. с англ. – М.:Мир, 1990 – 584с. ISBN 5-03-001191-9
6. Давыдов А.В. Цифровая обработка сигналов. Конспект лекций.
<http://prodav.narod.ru/textbook/index.html>
7. Давыдов В.А., Давыдов А.В. Очистка сигналов от шумов методом эмпирической модовой декомпозиции в диалоговом режиме.
www.geoin.org/hht/app/hht6.doc
8. Тутыгин В.С. Способ очистки сигналов от шумов с использованием неполной декомпозиции Хуанга. В сб. докладов X Международной научно — практической конференции «Актуальные вопросы науки, технологии и производства», СПб., 2015.
9. Гонсалес Р., Вудс Р. Цифровая обработка изображений.// Техносфера. – 2006 г.-1072с.
10. Гонсалес Р., Вудс Р., Эддинс С., Цифровая обработка изображений в среде MATLAB.// Техносфера. – 2006 г.-616с.
11. В.Дьяконов, И.Абраменкова. MATLAB. Обработка сигналов и изображений. Специальный справочник. –СПб.: Питер, 2002.-608с.