

СПбГПУ ИИТУ
Кафедра измерительных
информационных технологий

Сети и системы передачи данных.
Описание лабораторных работ

Богач Н.В.
<Bogach@iit.icc.spbstu.ru>
Вылегжанина К.Д.
<Vilegzhanina@iit.icc.spbstu.ru>

Санкт-Петербург 2014

Содержание

1	Система контроля версий SVN	3
2	Система верстки $\text{T}_{\text{E}}\text{X}$ и расширения $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$	4
3	Система для математических вычислений Octave	6
4	Визуализация сигналов во временной и частотной области	8
5	Спектры простых сигналов	10
6	Линейная фильтрация	11
7	Аналоговая модуляция	16
8	Частотная и фазовая модуляция	18
9	Цифровая модуляция	19
10	Моделирование схем кодирования сигналов	21
11	Помехоустойчивое кодирование	23

1 Система контроля версий SVN

1.1 Цель работы

Изучение приемов работы с системой контроля версий SVN, понимание основных принципов

1.2 Ход работы

1. Изучить справку для основных команд

```
svn help
svn checkout
svn update
svn add
svn commit
другие
```

2. Получить содержимое репозитория (assembla.com)
– checkout
3. Добавить новую папки и первого файла под контроль версий
4. Зафиксировать изменения
5. Получить обновления с сервера
6. Внести изменения в файл и просмотреть различия
7. Отменить локальные изменения
8. Внести изменения в файл и просмотреть различия
9. Зафиксировать изменения

1.3 Выводы

Описание целей использования систем контроля версий и возникающих при их использовании преимуществ

1.4 Материалы

1. SVN Red Book
2. Видеолекция "Системы контроля версий" на events.yandex.ru

1.5 Инструменты

1. TortoiseSVN

2 Система верстки \TeX и расширения \LaTeX

2.1 Цель работы

Изучение принципов верстки \TeX , создание первого отчета

2.2 Ход работы

Изучение

1. Создание минимального файла `.tex` в простом текстовом редакторе – преамбула, тело документа
2. Компиляция в командной строке – `latex`, `xdvi`, `pdflatex`
3. Оболочка `TexMaker`, Быстрый старт, Быстрая сборка

4. Создание титульного листа, нескольких разделов, списка, несложной формулы
5. Понятие классов документов, подключаемых пакетов
6. Верстка более сложных формул

Выполнение практического задания Создание отчетов по лабораторным работам ?? и 2

2.3 Выводы

Описание преимуществ и недостатков в подходе практикуемом при верстке для системы \TeX

Один абзац, выражающий впечатления от работы с \TeX

2.4 Материалы

1. Не очень краткое введение
2. Конспект-справочник
3. <http://www.inp.nsk.su/~baldin/LaTeX/lurs.pdf>
4. Математика в \LaTeX

2.5 Инструменты

1. \TeX для Windows Pro \TeX t
2. TeXnicCenter
3. TeXmaker

3 Система для математических вычислений Octave

3.1 Цель работы

Изучить принципы работы и команды системы для математических вычислений Octave (открытый исходный код)

3.2 Ход работы

Изучение

1. Запуск Octave
2. Элементарные арифметические операции (сложение, вычитание, умножение, деление, возведение в степень)
3. Логические операции (равно, не равно, И, ИЛИ, иск ИЛИ).
4. Присваивание (числа, строки, результаты вычислений, другие переменные)
5. Подавление вывода, вывод значений переменных, вывод списка переменных в области видимости. Удаление из области видимости.
6. Массивы и матрицы. Непосредственное определение, диапазон с шагом, нули, единицы, случайные числа, единичная матрица. Определение размера. Обращение к произвольным строкам и элементам, добавление строк и столбцов.

7. Загрузка данных из файла (текст с разделителем), сохранение в файл.
8. Арифметика. Логарифм, экспонента, модуль. Транспонирование. Максимальное значение. Сумма. Округление.
9. Визуализация. Гистограмма. Графики, подписи, оси, легенда, заголовок. Несколько графиков.
10. Циклы. For, while, break, continue. Ветвления.
11. Определения функций.

Практическое задание Выполнить сортировку массива действительных чисел любым выбранным Вами алгоритмом. Сравнение по быстродействию разработанной программы со встроенными функциями сортировки (представить графики зависимости среднего времени, требуемого на выполнение сортировки массива из n случайных действительных чисел от его размерности для встроенной функции сортировки и для разработанной).

3.3 Выводы

3.4 Материалы

1. Octave Wiki
2. Octave официальная документация

3.5 Инструменты

1. GNU Octave

4 Визуализация сигналов во временной и частотной области

- Цель: Познакомиться со средствами генерации сигналов и визуализации их спектров.
- Постановка задачи: В командном окне MATLAB и в среде Simulink промоделировать чистый синусоидальный сигнал, а также синусоидальный сигнал с шумом. Получить их спектры.
- Справочные материалы: В.С. Гутников Фильтрация измерительных сигналов ш.1-2, 11-12
- Примерный ход работы:

```
x = 0:0.01:4*pi;  
f0 = 5;  
%исходный сигнал  
y = sin(2*pi*f0*x);  
plot(x(1:200),y(1:200))  
grid  
%спектр исходного сигнала  
figure  
spectrum = fft(y,512);  
norm_spectrum = spectrum.*conj(spectrum)/512;  
f=100*(0:255)/512;  
plot(f, norm_spectrum(1:256))  
axis([0 max(f) 0 10])  
grid
```

Для создания модели в Simulink используйте разделы Sources и Sinks главной библиотеки.

- В выводах поясните возможные расхождения между теоретически ожидаемым и полученным спектром сигнала вследствие дискретности представления сигналов, а также их конечности во времени.

5 Спектры простых сигналов

- Цель: Получить представление о свойствах спектров.
- Постановка задачи: В командном окне MATLAB и в среде Simulink промоделируйте следующие тестовые сигналы

- полигармонический сигнал $y(t) = \sum_{n=0}^{N-1} \cos(nt)$
- прямоугольный импульсный сигнал $y(t) = \Pi(t, T_i)$
- треугольный импульсный сигнал $y(t) = \Delta(t, T_i)$

Получите их спектры.

- Справочные материалы: В.С. Гутников Фильтрация измерительных сигналов пп.3-6, 13-14 Для получения треугольного сигнала используйте операцию свертки

`conv`

Для создания модели в Simulink используйте разделы Sources и Sinks главной библиотеки. Для получения генератора треугольного сигнала используйте генератор прямоугольных импульсов каскадно с фильтром с прямоугольным окном.

- В выводах обоснуйте использование фильтра с прямоугольным окном для получения треугольного сигнала

6 Линейная фильтрация

- Цель: изучить воздействие ФНЧ на тестовый сигнал с шумом.
- Постановка задачи: сгенерировать гармонический сигнал с шумом и синтезировать ФНЧ. Получить сигнал во временной и частотной областях до и после фильтрации. Сделать выводы о воздействии ФНЧ на спектр сигнала.
- Справочные материалы: В.С. Гутников Фильтрация измерительных сигналов пп.17-19 Примерный ход работы:

```
x = 0:0.01:4*pi;
f=100*(0:255)/512;
figure
noise=rand(size(x));
y = sin(2*pi*x);
y_noisy = y+0.3*noise;
plot(x(1:200),y(1:200))
grid
[B,A] = BUTTER(16,0.98); % синтез ФНЧ Баттерворта
B=B./sum(B);
A=A./sum(A);
%обработка сигнала ФНЧ
figure
y_filtered = conv(y_noisy,[B,A]);
plot(x(1:200),y_filtered(1:200))
grid
figure
noisy_spectrum = fft(y_noisy,512);
norm_noisy_spectrum = spectrum.*conj(spectrum)/512;
```

```

plot(f,norm_noisy_spectrum(1:256))
axis([0 max(f) 0 2])
grid
figure
spectrum = fft(y_filtered,512);
norm_filtered_spectrum=
    filtered_spectrum.*conj(filtered_spectrum)/512;
plot(f,norm_filtered_spectrum(1:256))
axis([0 max(f) 0 2])
grid

```

Для создания модели в Simulink используйте блок Discrete FIR Filter раздела Discrete главной библиотеки и блок Digital Filter design из Signal Processing Blockset/Filtering/Filter Designs.

- В выводе поясните неполное удаление шума линейным фильтром

Технологии модуляции

ФУНКЦИИ МОДУЛЯЦИИ/ДЕМОДУЛЯЦИИ

<http://matlab.exponenta.ru/communication/book1/index.php>

Для передачи по любому каналу связи цифровое сообщение, представляющее собой последовательность символов (чисел), необходимо преобразовать в аналоговый сигнал — изменяющуюся во времени физическую величину (например, напряжение). Кроме того, канал связи способен пропускать лишь определенную полосу частот. Преобразование сигнала для переноса в заданный частотный диапазон осуществляется путем модуляции. Обратный процесс называется демодуляцией. Пакет MatLab Communications содержит функции для реализации аналоговой и цифровой модуляции/демодуляции. При аналоговой модуляции входным сигналом является непрерывная функция, при цифровой — последовательность символов. Модулированный сигнал может представляться либо в вещественном виде (*passband simulation*), либо в виде комплексной огибающей (*baseband simulation*). Соответственно приведенной классификации имеется 8 функций: (в новых версиях пакета, начиная с Matlab 2008, имеется возможность использовать класс `modem` для построения модемов, для справки: `help modem/types`)

amod — аналоговая модуляция, вещественный выходной сигнал;

amodce — аналоговая модуляция, выходной сигнал в виде комплексной огибающей;

ademod — аналоговая демодуляция, вещественный входной сигнал;

ademodce — аналоговая демодуляция, входной сигнал в виде комплексной огибающей;

dmod — цифровая модуляция, вещественный выходной сигнал;

dmodce — цифровая модуляция, выходной сигнал в виде комплексной огибающей;

ddemod — цифровая демодуляция, вещественный входной сигнал;

ddemodce — цифровая демодуляция, входной сигнал в виде комплексной огибающей;

Функциями пакета поддерживаются следующие виды аналоговой модуляции:

- амплитудная модуляция;
- амплитудная модуляция с подавленной несущей;
- однополосная модуляция;
- частотная модуляция;
- фазовая модуляция;
- квадратурная модуляция

При цифровой модуляции возможны следующие ее виды:

- амплитудная манипуляция;
- частотная манипуляция;
- минимальная частотная манипуляция;
- фазовая манипуляция;

- квадратурная манипуляция

Цифровая модуляция и демодуляция включают в себя две стадии. При модуляции цифровое сообщение сначала преобразуется в аналоговый модулирующий сигнал с помощью функции `modmtr`, а затем осуществляется аналоговая модуляция. При демодуляции сначала получается аналоговый демодулированный сигнал, а затем он преобразуется в цифровое сообщение с помощью функции `demodmtr`. Наконец, три последних функции этой группы предназначены для работы с конкретными сигнальными созвездиями квадратурной манипуляции. Функции `qaskenco` и `qaskdeco` производят кодирование и декодирование сообщения с использованием “квадратного” созвездия, а функция `arkconst` позволяет вывести на экран изображение “концентрического” созвездия.

МОДЕЛИРОВАНИЕ КАНАЛОВ СВЯЗИ

Сформированный в результате модуляции сигнал поступает в канал связи, где он подвергается воздействию шумов и помех. Поэтому функции моделирования каналов связи должны обеспечивать внесение в сигнал искажений согласно используемым статистическим моделям. Данная группа функций пакета `Communications` в данный момент представлена лишь одной функцией `awgn`, которая позволяет добавить к сигналу аддитивный белый нормальный шум, реализовав при этом заданное отношение сигнал/шум.

7 Аналоговая модуляция

- Цель: изучение амплитудной модуляции/демодуляции сигнала.
- Постановка задачи:
 1. Сгенерировать однотоновый сигнал низкой частоты.
 2. Выполнить амплитудную модуляцию (АМ) сигнала по закону $u(t) = (1 + MU_m \cos(\Omega t)) \cos(\omega_0 t + \phi_0)$ для различных значений глубины модуляции M . Используйте встроенную функцию MatLab `ammod`¹
 3. Получить спектр модулированного сигнала.
 4. Выполнить модуляцию с подавлением несущей $u(t) = MU_m \cos(\Omega t) \cos(\omega_0 t + \phi_0)$. Получить спектр.
 5. Выполнить однополосную модуляцию:

$$u(t) = U_m \cos(\Omega t) \cos(\omega_0 t + \phi_0) + \frac{U_m}{2} \sum_{n=1}^N M_n (\cos(\omega_0 + \Omega_n)t + \phi_0 + \Phi_n)$$

положив $n=1$

6. Выполнить синхронное детектирование и получить исходный однополосный сигнал.
7. Рассчитать КПД модуляции.

$$\eta_A M = \frac{U_m^2 M^2 / 4}{P_U} = \frac{M^2}{M^2 + 2}$$

¹Используйте при необходимости помощь MatLab: `help <имя функции>`

- Для создания модели используйте раздел Communication Blockset Simulink
- Справочные материалы: Н.В. Богач и др. Обработка сигналов в информационных системах, с. 110-118, 125-127

8 Частотная и фазовая модуляция

- Цель: изучение частотной и фазовой модуляции/демодуляции сигнала.
- Постановка задачи:
 1. Сгенерировать однотоновый сигнал низкой частоты.
 2. Выполнить фазовую модуляцию/демодуляцию сигнала по закону $u(t) = (U_m \cos(\Omega t + k s(t)))$, используя встроенную функцию MatLab `pmmod`, `pmdemod`
 3. Получить спектр модулированного сигнала.
 4. Выполнить частотную модуляцию/демодуляцию по закону

$$u(t) = U_m \cos(\omega_0 t + k \int_0^t s(t) dt + \phi_0)$$

используя встроенные функции MatLab `fmmod`, `fmdemod`

- Для создания модели используйте раздел Communication Blockset Simulink. Выполните демодуляцию, в том числе с помощью блока захвата фазы (фазовой автоподстройки частоты) Phase-Locked Loop (Communication Blockset/ Synchronization/ Components).
- Справочные материалы: Н.В. Богач и др. Обработка сигналов в информационных системах, с. 118-125, 127-133

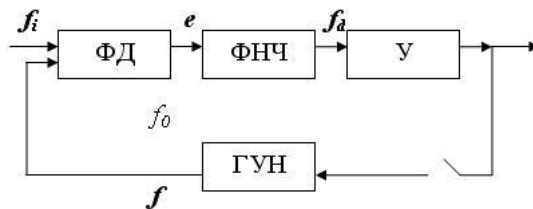


Рис. 1: Структурная схема блока PLL

9 Цифровая модуляция

- Цель: изучение методов модуляции цифровых сигналов.

Функция **randerr** предназначена для формирования ошибок в цифровом сигнале. Она возвращает матрицу, в каждой строке которой имеется заданное число случайно расположенных ненулевых элементов. Для оценки помехоустойчивости системы связи необходимо произвести сравнение исходного (передаваемого) сообщения с сообщением, полученным в результате приема, и определить число ошибок, возникших в процессе передачи, а также вероятность ошибки. Это можно выполнить функциями **symerr** и **biterr**, первая из которых подсчитывает число несовпадающих символов в двух сообщениях, а вторая — число несовпадающих битов в двоичных представлениях этих символов. Кроме числа ошибок, обе функции могут возвращать долю ошибок в общем числе символов (битов) и индикаторы мест возникновения ошибок. Последние две функции данной группы предназначены для графического отображения сигналов с квадратурной манипуляцией. Функ-

ция `eyediagram` выводит глазковую диаграмму, а функция `scatterplot` — диаграмму рассеяния.

- Постановка задачи:

1. Получить сигналы BPSK, PSK, OQPSK, genQAM, MSK, M-FSK модуляторов ²
2. Построить их сигнальные созвездия
3. Провести сравнение изученных методов модуляции цифровых сигналов

```
% EXAMPLE: Construct modulation objects
% to perform QPSK modulation and demodulation
% Modulator object
h = modem.pskmod('M', 4);
% Demodulator object
g = modem.pskdemod('M', 4);
% Modulating message
msg = randint(10,1,4);
% Modulate signal
modSignal = modulate(h,msg);
% Demodulate signal
demodSignal = demodulate(g,modSignal);
```

- Для создания модели используйте раздел Communication Blockset/ Digital Baseband/Passband Modulation.
- Справочные материалы: Н.В. Богач и др. Обработка сигналов в информационных системах, с. 137-141

²Используйте "help modem/types" для справки по конструктору объектов `modem`

10 Моделирование схем кодирования сигналов

Функции `encode` и `decode` осуществляют, соответственно, кодирование и декодирование сообщения с использованием блочного кода. Тип используемого кода задается в числе параметров функций. Линейный блочный код в общем случае описывается порождающей матрицей (`generator matrix`). Кодирование блока (вектора) производится путем его умножения на порождающую матрицу. При контроле ошибок на приемной стороне используется проверочная матрица кода (`parity-check matrix`). Преобразование порождающей матрицы в проверочную и обратно осуществляется функцией `gen2par`. Если умножение кодированного блока на проверочную матрицу не дает нулевого вектора, то полученный результат (его называют синдром — `syndrome`) позволяет определить, какие именно символы были искажены в процессе передачи. Для двоичного кода это позволяет исправить ошибки. Декодирование линейного блочного кода, таким образом, можно осуществить с помощью таблицы, в которой для каждого значения синдрома указан соответствующий вектор ошибок. Создать такую таблицу на основании проверочной матрицы кода позволяет функция `syndtable`. Функция `gfweight` позволяет определить кодовое расстояние для линейного блочного кода по его порождающей или проверочной матрице.

Циклические коды

Циклические коды — это подкласс линейных кодов, обладающие тем свойством, что циклическая перестановка символов в кодированном блоке дает другое возможное кодовое слово того же кода. Для работы с циклическими кодами в пакете `Communications` есть две функ-

ции. Задав число символов в кодируемом и закодированном блоках, с помощью функции `cyclpoly` можно получить порождающий полином циклического кода. Далее, используя этот полином в качестве одного из параметров функции `cyclgen`, можно получить порождающую и проверочную матрицы для данного кода. Коды БЧХ Коды БЧХ являются одним из подклассов циклических блочных кодов. Для работы с ними функции высокого уровня вызывают специализированные функции `bchenco` (кодирование) и `bchdeco` (декодирование). Кроме того, функция `bchpoly` позволяет рассчитывать параметры или порождающий полином для двоичных кодов БЧХ.

Коды Хэмминга

Коды Хэмминга являются одним из подклассов циклических блочных кодов. Порождающий полином для кодов Хэмминга неприводим и примитивен, а длина кодированного блока равна $2m - 1$. Порождающая и проверочная матрицы для кодов Хэмминга генерируются функцией `hammgen`.

Коды Рида—Соломона

Коды Рида—Соломона являются одним из подклассов циклических блочных кодов. Это единственные поддерживаемые пакетом Communications не двоичные коды. Для работы с кодами Рида—Соломона функции высокого уровня вызывают специализированные функции `rsenco` (кодирование) и `rsdeco` (декодирование). Кроме того, функции `rsencode` и `rsdecode` позволяют использовать при кодировании и декодировании экспоненциальный формат данных, а функции `rsencof` и `rsdecof` осуществляют кодирование и декодирование текстового файла. Функция `rspoly` генерирует порождающие полиномы для кодов Рида—Соломона.

11 Помехоустойчивое кодирование

- Цель: Изучение методов помехоустойчивого кодирования и сравнение их свойств.
- Постановка задачи:
 1. Провести кодирование/декодирование сигнала, полученного с помощью функции `randert` кодом Хэмминга 2-мя способами: с помощью встроенных функций `encode/decode`, а также через создание проверочной и генераторной матриц и вычисление синдрома. Оценить корректирующую способность кода.
 2. Выполнить кодирование/декодирование циклическим кодом, кодом БЧХ, кодом Рида-Соломона. Оценить корректирующую способность кода.