**Титул**

**Задание**

**Содержание**

[Введение 4](#_Toc9)

[1 Основные теоретические сведения 5](#_Toc10)

[2 Ход работы 9](#_Toc11)

[2.1 Описание процесса выполнения работы 9](#_Toc12)

[2.2 Пример выполнения работы 10](#_Toc13)

[3 Контрольные вопросы 17](#_Toc14)

[Заключение 18](#_Toc15)

[Список использованных источников 19](#_Toc16)

# **Введение**

В ходе педагогической практики необходимо было разработать программу лабораторной работы по курсу “Цифровая обработка сигналов информационно-управляющих систем” по теме “Звуковые сигналы”.

В ходе практики был получен опыт разработки методических указаний, содержащих теоретический материал, инструкцию по выполнению практического задания и контрольные вопросы. Кроме того, инструкции для практического задания были проверены опытным путем.

# 1 Основные теоретические сведения

В своей повседневной жизни человек регулярно имеет дело с процессами формирования и обработки звуковых сигналов. Наша речь — это древнейшая способность человека, позволяющая общаться друг с другом. Она представляет собой последовательность акустических сигналов. Без этих сигналов сложно представить жизнь современного человеческого общества со всеми его достижениями. Для людей с ограниченным зрением звуковые сигналы в бытовой жизни имеют еще большее значение. Так, например, в современных городах сигналы светофора дополняют звуковыми сигналами. Кроме необходимой для жизни информации, звуки могут доставлять человеку эмоциональные переживания — музыкальное творчество с незапамятных времён являлось средством выражения чувств.

В настоящее время человек, используя современные технологии, преуспел в передаче речи на большие расстояния, записи музыки в отличном качестве. Мобильные телефоны, диктофоны, микрофоны, планшеты, компьютеры, аудиоплееры, видеоплееры, видеокамеры, микшерные пульты, динамики и наушники — все эти устройства формируют, хранят, обрабатывают или воспроизводят звук для того, чтобы наша жизнь была комфортной.

Для описания звуковых сигналов музыканты используют нотную грамоту, тем не менее каждый звуковой сигнал может быть описан и математически [x].

Любой звуковой сигнал постоянной тональности представляет собой гармонический сигнал определенной длительности:

,

где — амплитуда сигнала, соответствует уровню громкости звука;

— частота колебания, соответствует тональности звука;

— фаза сигнала (не различается человеческим слухом);

— длительность тонального сигнала.

С увеличением амплитуды звук становится громче, с увеличением частоты — выше. Среднестатистическому человеческому уху доступны частоты в диапазоне от 20 Гц до 20 кГц. При этом принято считать, что человеческая речь создает колебания в диапазоне от 300 Гц до 4 кГц. В современной музыке стандартом музыкального строя стала равномерная темперация.

Равномерная темперация} — темперированный музыкальный строй, в котором каждая октава делится на математически равные интервалы, в наиболее типичном случае на двенадцать полутонов диапазоном Гц.

Можно математически вычислить частоты для всего звукоряда, пользуясь формулой:

,

где — частота камертона (например, нота Ля — 440 Гц);

— количество полутонов в интервале от исследуемого звука к эталону.

Современный уровень технологий позволяет хранить и обрабатывать аудио сигналы в цифровом виде.

Цифровая звукозапись — технология преобразования аналогового звука в цифровой с целью сохранения его на физическом носителе для возможности последующей обработки или воспроизведения.

Основными параметрами цифрового звукозаписывающего устройства с точки зрения обеспечения качества аудио сигнала являются разрядность, частота дискретизации, количество каналов и коэффициент сжатия. Перечень наиболее распространённых современных форматов цифровой звукозаписи приведен в таблице 1.

Таблица 1 — Перечень современных форматов цифровой звукозаписи.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название формата** | **Разрядность, бит** | **Частота дискретизации, кГц** | **Число каналов** | **Коэффициент сжатия** |
| CD | 16 | 44,1 | 2 | 1 |
| Dolby Digital (AC3) | 16-24 | 48 | 6 | 12 |
| DTS | 20---24 | 48; 96 | до 8 | 3 |
| DVD-Audio | 16; 20; 2 | 44,1; 48; 88,2; 96 | 6 | 2 |
| DVD-Audio | 16; 20; 24 | 176,4; 192 | 2 | 2 |
| MP3 | переменная | до 48 | 2 | 11 |
| AAC | переменная | до 96 | до 48 | с потерями |
| AAC+ (SBR) | переменная | до 48 | 2 | с потерями |
| Ogg Vorbis | до 32 | до 192 | до 255 | с потерями |
| WMA | до 24 | до 96 | до 8 | 2 |

# **2 Ход работы**

## 2.1 Описание процесса выполнения работы

Для успешного выполнения работы необходимо:

1. Выполнить имитационное моделирование сигнала, соответствующего звуковому ряду в виде последовательности из заданных нот длительности .
2. Построить график полученного сигнала во временной области.
3. Сформировать и построить спектр мощности и спектрограмму сигнала.
4. Воспроизвести с использованием звуковой карты и динамиков компьютера полученную звуковую последовательность.
5. Сохранить звуковой ряд в файл на жестком диске в произвольном аудио формате.
6. Прослушать полученный файл средствами операционной системы или установленного программного обеспечения.
7. Сформировать и наложить шум, распределенный по нормальному закону на звуковую последовательность нот, таким образом, чтобы соотношение сигнал-шум результирующего сигнала составляло .
8. Воспроизвести с использованием звуковой карты и динамиков компьютера полученную звуковую последовательность с наложенным шумом.
9. Составить, проиграть и записать в файл произвольную мелодию длительностью не менее 50 нот.

После выполнения экспериментальной части необходимо ответить на предложенные контрольные вопросы для закрепления пройденного материала и установления взаимосвязи между полученными результатами практических работ и теоретическими знаниями. Результаты работы рекомендуется оформить в виде отчета, в котором должна содержаться следующая информация: цель работы; решённые в процессе её достижения задачи; основные математические выражения, использованные при решении задач; текст программы или схема моделирования, результаты моделирования в виде графиков и заключение, позволяющее сделать вывод о сопоставимости результатов практической работы с теоретическими сведениями.

## 2.2 Пример выполнения работы

Инициализация и формирование значений основных параметров продемонстрированы в листинге 1.

Листинг 1 — инициализация и формирование значений основных параметров

|  |
| --- |
| clear all; % Очистка памяти  close all; % Закрытие всех окон с графиками  clc; % Очистка окна команд и сообщений  fontSize = 10; % Размер шрифта графиков  tColor = 'b'; % Цвет графиков во временной области  fColor = [1 0.4 0]; % Цвет графиков в частотной области  xlimit = 0.05; % Ограничение области определения на графике |

В листинге 2 приведен исходный код для визуализации звуковой волны из файла с мелодией. На рисунке 4 продемонстрирован результат.

Листинг 2 — чтение файла с мелодией

|  |
| --- |
| [data,rate] = audioread('complex.wav');  t = linspace(0, length(data)/rate, length(data))'; % Формирование области определения  figure('Renderer', 'painters', 'Position', [0 0 700 300]);  subplot(1,2,1);  plot(t, data, 'Color', tColor);  xlim([0 xlimit]); % Показать сигнал в начале мелодии  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  subplot(1,2,2);  plot(t, data, 'Color', tColor);  xlim([t(end)-xlimit t(end)]); % Показать сигнал в конце мелодии  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  sgtitle('Исходный сигнал во временной области'); |

|  |
| --- |
|  |

Рисунок 4 — исходный сигнал во временной области

В листинге 3 приведен исходный код для визуализации звуковой волны в частотной области. На рисунке 5 продемонстрирован результат.

Листинг 3 — визуализация звуковой волны в частотной области

|  |
| --- |
| f\_axis = linspace(0, rate, length(data)); % Формирование области определения  fdata = abs(fft(data)/length(data)); % Формирование значений спектра  figure; plot([-fliplr(f\_axis(1:end/2)) f\_axis(1:end/2)], fftshift(fdata),...  'Color', fColor, 'LineWidth', 3);  xlim([0 1100]); % Ограничение области определения  ylim([0 0.01]);  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  title('\rm Исходный сигнал в частотной области'); % Заголовок  xlabel('Частота,\it f\rm, Гц'); % Надпись оси абсцисс  ylabel('Амплитуда,\it A(f)\rm, В'); % Надпись оси ординат |

|  |
| --- |
|  |

Рисунок 5 — исходный сигнал во временной области

В листинге 4 приведен исходный код для построения спектрограммы сигнала. На рисунке 6 продемонстрирован результат.

Листинг 4 — построение спектрограммы сигнала

|  |
| --- |
| spectrogram(data,4096,64,4096,rate,'yaxis');  ylim([0 1.1]); % Ограничение области определения в кГц  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  title('\rm Спектрограмма исходного сигнала'); % Заголовок  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Частота,\it f\rm, Гц'); % Надпись оси ординат  yt = get(gca, 'YTick'); % Перевод единиц измерения частоты в Гц  set(gca, 'YTick', yt, 'YTickLabel', yt\*1E+3);  ylabel('Амплитуда,\it A(f)\rm, В'); % Надпись оси ординат |

|  |
| --- |
|  |

Рисунок 6 — спектрограмма сигнала

Далее перейдем непосредственно к преобразованию Гильберта-Хуанга. Первым его шагом является выполнение эмпирической модовой декомпозиции (листинг 5, рисунки 7-8).

Листинг 5 — выполнение эмпирической модовой декомпозиции сигнала

|  |
| --- |
| c = zeros(length(data),5); % массив для пяти эмпирических модовых функций  r = zeros(length(data),5); % массив для пяти остатков  c\_index = 1; % индекс эмпирической модовой функции  y = data; % y - раскладываемый сигнал  while 1  h\_prev = y; % h\_prev - предыдущее приближение модовой функции  h\_index = 1; % номер приближения модовой функции  while 1  [pks\_max,locs] = findpeaks(h\_prev);  t\_pks\_max = t(locs); % Нахождение локальных максимумов  [pks\_min,locs] = findpeaks(-h\_prev);  pks\_min = -pks\_min;  t\_pks\_min = t(locs); % Нахождение локальных минимумов  polynom\_max = spline(t\_pks\_max,pks\_max,t); % Вычисление огибающих  polynom\_min = spline(t\_pks\_min,pks\_min,t);  m = (polynom\_max + polynom\_min)./2; % Вычисление функции средних значений  if (h\_index == 1) % Построить графики вычисленных функций для первого приближения  figure('Renderer', 'painters', 'Position', [0 0 700 300]);  subplot(1,2,1);  plot(t, h\_prev, '-b', t, polynom\_max, '-r', t, polynom\_min, '-g', t, m, '-k');  xlim([0 xlimit]); % Ограничение области определения  ylim([-1 1]); % Ограничение области определения  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  subplot(1,2,2);  plot(t, h\_prev, '-b', t, polynom\_max, '-r', t, polynom\_min, '-g', t, m, '-k');  xlim([t(end)-xlimit t(end)]); % Ограничение области определения  ylim([-1 1]); % Ограничение области определения  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  sgtitle(sprintf('Интерполяция экстремумов для нахождения %d модовой функции', c\_index));  end  h = h\_prev - m; % Нахождение следующего приближения  eps = sum((h\_prev - h).^2)/sum(h\_prev.^2);  if(eps < 1e-6) % Выход из цикла при малой разности между двумя приближениями  break;  end  h\_prev = h; % Переход к расчeту следующего приближения  h\_index = h\_index + 1;  end  c(:,c\_index) = h; % Сохранить последнее приближение как модовую функцию  r(:,c\_index) = y - c(:,c\_index); % Вычисление остатка  y = r(:,c\_index); % Остаток - следующий сигнал для разложения  eps2 = sum(r(:,c\_index).^2)/length(r(:,c\_index));  if(eps2 < 1e-6) % Выход из цикла при нулевом остатке  break;  end  c\_index = c\_index + 1; % Переход к вычислению следующей модовой функции  end |

|  |
| --- |
|  |

Рисунок 7 — интерполяция экстремумов для нахождения первой модовой функции

|  |
| --- |
|  |

Рисунок 8 — интерполяция экстремумов для нахождения второй модовой функции

Теперь визуализируем зависимости значения модовых функций и их частоты от времени (листинг 8).

Листинг 8 — зависимость значения модовых функций и их частоты от времени

|  |
| --- |
| freq = zeros(length(data),5); % Значения частоты в зависимоти от времени  for index = 1:c\_index % Отобразить каждую модовую функцию и зависимости частоты от времени  figure('Renderer', 'painters', 'Position', [0 0 700 300]);  subplot(1,2,1);  plot(t, c(:,index), '-b');  xlim([0 xlimit]); % Ограничение области определения  ylim([-1 1]); % Ограничение области определения  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  subplot(1,2,2);  plot(t, c(:,index), '-b');  xlim([t(end)-xlimit t(end)]); % Ограничение области определения  ylim([-1 1]); % Ограничение области определения  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Сигнал,\it x(nT\_д )\rm, В'); % Надпись оси ординат  sgtitle(sprintf('%d Эмпирическая модовая функция',index));  % Нахождение зависимости частоты от времени  hx = hilbert(c(:,index)); % Преобразование Гильберта  phi = angle(hx); % Вычисление фазы  phi2 = unwrap(phi);  vector = linspace(0, length(c(:,index))-1, length(c(:,index)))';  p = polyfit(vector,phi2,3); % Аппроксимация полиномом третьей степени  dp = polyder(p); % Производная полинома  freq(:,index) = polyval(dp, 0:length(c(:,index))-1).\*7000; % Вычисление значений частоты от времени  figure; plot(t, freq(:,index), 'Color', tColor);  ylim([0 1100]); % Ограничение области определения  set(get(gcf, 'CurrentAxes'), 'FontSize', fontSize); % Изменение шрифта  title(sprintf('\\rm Частотная зависимость от времени %d модовой функции', index)); % Заголовок  xlabel('Время,\it nT\_д\rm, с'); % Надпись оси абсцисс  ylabel('Частота,\it f\rm, Гц'); % Надпись оси ординат  end |

Результаты визуализации показаны на рисунках 9 и 10.

|  |
| --- |
|  |

Рисунок 9 — первая модовая функция во временной области

|  |
| --- |
|  |

Рисунок 10 — зависимость частоты модовых функций от времени

Очевидно сходство полученной зависимости со спектрограммой исходного сигнала, что подтверждает, что преобразование Гильберта-Хуанга было выполнено корректно.

# 3 Контрольные вопросы

В заключение лабораторной работы студенту необходимо ответить на контрольные вопросы, приведенные ниже:

1. Чем отличаются области применения преобразования Фурье и преобразования Гильберта-Хуанга?
2. Приведите аналитическое выражение для преобразования Гильберта- Хуанга.
3. Что такое эмпирическая модовая декомпозиция и для чего она нужна?
4. Опишите алгоритм получения эмпирических модовых функций.
5. Каким выражением определяется мгновенная частота при использовании преобразования Гильберта-Хуанга?

# Заключение

В ходе педагогической была разработана программа лабораторной работы по курсу “Цифровая обработка сигналов информационно-управляющих систем” по теме “Преобразование Гильберта-Хуанга” и углублены собственные знания в этой теме.

Разработанные методические указания содержат теоретический материал, инструкцию по выполнению практического задания и контрольные вопросы. Кроме того, инструкции для практического задания были проверены опытным путем.

# **Список использованных источников**

1. Ястребов И. П. О свойствах и применениях преобразования Гильберта-Хуанга //Проектирование и технология электронных средств. – 2016. – №. 1. – С. 26-33.
2. Алимурадов А. К., Тычков А. Ю. Применение преобразования Гильберта-Хуанга в задаче выделения информативных признаков речевых сигналов //Международный научно-исследовательский журнал. – 2013. – №. 5-1 (12). – С. 56-57.
3. ГИЛЬБЕРТА—ХУАНГА П. ПОДХОД К ОБНАРУЖЕНИЮ АНОМАЛИЙ В ТЕХНОЛОГИЧЕСКИХ СИГНАЛАХ С ПРИМЕНЕНИЕМ ПРЕОБРАЗОВАНИЯ ГИЛЬБЕРТА—ХУАНГА //Автометрия. – 2021. – Т. 57. – №. 1. – С. 31.
4. Huang N. E. Introduction to the Hilbert–Huang transform and its related mathematical problems //Hilbert–Huang transform and its applications. – 2014. – С. 1-26.
5. Daji H., Jinping Z., Jilan S. Practical implementation of Hilbert-Huang transform algorithm //ACTA OCEANOLOGICA SINICA-ENGLISH EDITION-. – 2003. – Т. 22. – №. 1. – С. 1-14.