

Лабораторная работа №1

Краткая информационная справка

Звук - это колебательное движение частиц упругой среды: газообразной, жидкой или твёрдой. Нота – графическое изображение музыкального звука.

Звуковой (аудио) сигнал является представлением звука, с использованием уровня электрического напряжения для аналоговых сигналов, а также ряда двоичных чисел для цифровых сигналов. Человеческое ухо способно воспринимать звук в частотном диапазоне от 20 Гц до 20 кГц, что соответствует верхнему и нижнему пределам человеческого слуха

Чистой ноте «ля» соответствует аудио сигнал с частотой 440 Гц. Математическая модель такого аудио сигнала:

$$s(t) = A * \sin(2\pi f_{\text{ноты}} t + \varphi_0)$$

Аналоговый аудиосигнал является непрерывным во времени, дискретный аудиосигнал представляет собой последовательность значений в определённые моменты времени. Как правило, моменты, в которые берутся значения отчётов сигнала распределены равномерно. Промежуток времени между двумя соседними отчётами времени называют интервалом дискретизации T_s , а величину обратную интервалу дискретизации – частотой дискретизации $F_s = 1/T_s$ (от англ. *Sample* - отчет). Для восстановления без потерь аналогового сигнала из дискретного частота дискретизации должна быть больше удвоенной верхней частоты, поэтому при работе со звуком частота дискретизации, как правило, больше 40 кГц (20 кГц * 2).

На рис.1, 2 показаны аналоговый и дискретный аудиосигналы, соответствующие ноте «ля».

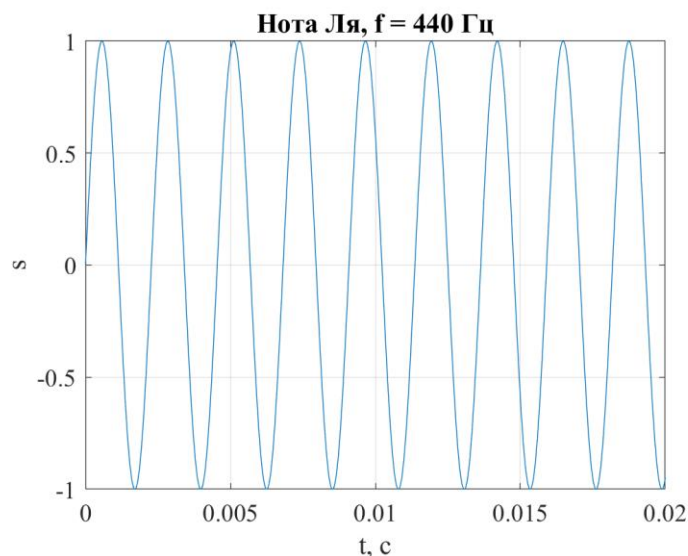


Рис. 1. Аналоговый аудиосигнал во временной области (нота Ля)

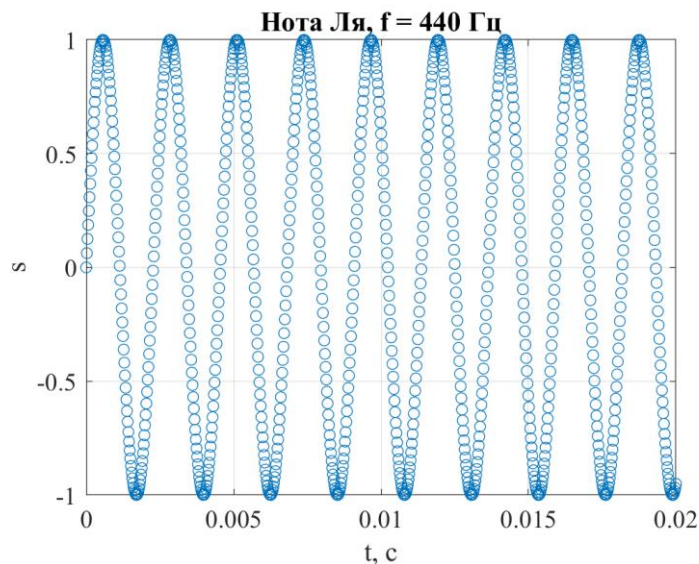


Рис. 2. Дискретный аудиосигнал во временной области (нота Ля)

Последовательность базовых заданий

1. Включите дневник записи команд в Command Window с помощью команды `diary` с указанием файла `lab1_ASP_SurnameN.txt` и приступайте к работе с Command Window.

Создайте в Command Window переменные для длительности ноты `pitchDur` и частоты дискретизации `fS` со значениями 1 и 8000 соответственно.

Вопросы для самоконтроля. В каком окне интерфейса Matlab отображаются созданные переменные? Какой тип данных у созданных переменных?

2. Создайте вектор-строку `timeArray` со значениями моментов времени в соответствие с заданной частотой дискретизации.

```
timeArray=0:1/fS: pitchDur;
```

3. Создайте вектор-строку со значениями аудиосигнала, соответствующего ноте «до/С» первой октавы. Исходные данные: $\varphi_0 = 0, A = 1, f_{\text{ноты}} = 261.63$ Гц

4. Выключите дневник записи команд.

Вопросы для самоконтроля. Проверьте созданный командой `diary` файл, что он содержит?

5. Очистите историю команд с помощью команды `clc` и очистите рабочее пространство с помощью команды `clear`.

6. Скачайте файл `pitchVSfreq.csv` соответствия нот и частот. Импортируйте в Matlab частоты для нот первой октавы в числовой массив `freqPitch`.

7. Создайте скрипт с названием *lab1_ASP_SurnameN.m* в директории /Documents/Matlab. Убедитесь, что в окне интерфейса *Current Folder* появился Ваш скрипт.

Surname - Фамилия

N - первая буква имени

Например, *lab1_I_PopovE.m*

Создайте в скрипте первую секцию с названием *Creating pitches*. Первая секция должна включать в себя создание переменных *pitchDur*, *fS* и массивов *timeArray*, *pitchA*, *pitchB*, *pitchC*, *pitchD*, *pitchE*, *pitchF*, *pitchG*. После запуска данной секции в *Workspace* должны остаться 11 переменных: *pitchDur*, *fS*, *timeArray*, *pitchA*, *pitchB*, *pitchC*, *pitchD*, *pitchE*, *pitchF*, *pitchG*, *freqPitch*. Создайте вторую секцию скрипта с временным названием *Temporary section*.

Вопросы для самоконтроля. Запустите первую секцию. Что содержит *Workspace*? Запустите весь скрипт целиком. Что содержит *Workspace*?

8. Добавьте во вторую секцию команду `clear freqPitch`.

Вопросы для самоконтроля. Запустите первую секцию. Что содержит *Workspace*? Запустите весь скрипт целиком. Что содержит *Workspace*? Чем отличается результат запуска всего скрипта от запуска только первой секции?

9. Скачайте файл *song.txt* с последовательностью нот

10. Измените название второй секции на *Read the file with pitches*. Прочитайте с помощью функции `fscanf` файл из пункта 9. Сохраните последовательность нот песни в переменную *song*. Используйте команду `whos` для определения типа данных переменной *song*.

Вопросы для самоконтроля. Какой тип данных у переменной *song*?

11. Создайте третью секцию скрипта с временным названием *Create the song and the file.wav*. Создайте массив отсчетов аудио сигнала *sigSong*, соответствующий последовательности нот в переменной *song*. Например, песня состоит из нот *A B C*, тогда `singsong=[pitchA pitchB pitchC];`

12. Прослушайте полученный аудиосигнал с помощью функции `sound` (третья секция).

Вопросы для самоконтроля. Что за песня играет?

13. Сохраните созданную песню в аудиофайл *firstSong.wav* с помощью функции `audiowrite` (третья секция).

14. Посмотрите информацию о созданном аудиофайле *firstSong.wav* с помощью функции `audioinfo('firstSong.wav')`, результат работы функции сохраните в переменную `audioWavInfo` (третья секция). Создайте 7 переменных следующим образом (третья секция) .:

```
wavFileName = audioWavInfo.Filename
wavFileCompMet = audioWavInfo.CompressionMethod
wavFileNumCh = audioWavInfo.NumChannels
wavFileRate = audioWavInfo.SampleRate
wavFileSamples = imgPngInfo.TotalSamples
wavFileDur = imgPngInfo.Duration
```

Вопросы для самоконтроля. Какой тип данных и размер у созданных переменных?

15. Скачайте отрывок песни *song.mp3*. Создайте четвертую секцию скрипта с временным названием *Listen the mp3 file*. В четвертой секции с помощью функции `audioread` прочитайте *mp3* файл. С помощью функции `sound` прослушайте его. С помощью функции `audioinfo` проанализируйте информацию об аудиофайле и сохраните ее в переменные разных типов по аналогии с пунктом 14. Сохраните переменные с информацией о файле с помощью команды `save` в файл *data.mat*.

Вопросы для самоконтроля. Какие новые свойства появились у файла *.mp3* по сравнению к файлу *.wav*? В каком окне интерфейса отображается созданный файл *data.mat*?

16. Используйте команду `fprintf` и выведите в Command Window информацию об аудиофайлах в следующем формате:

- в первой строке напечатать 10 тире, потом '*.MP3 VS .WAV*', потом повторить 10 тире,
- во второй строке отступить 10 пробелов и вывести значение переменной `wavFileName`, потом пробел, потом '*VS*' и закончить переменной, содержащей имя *mp3 файла*,
- в третьей строке отступить 10 пробелов и вывести значения двух переменных, содержащих информацию о сжатии, используемом в файлах (аналогично второй строке)
- в четвертой строке отступить 10 пробелов и вывести значения двух переменных, содержащих информацию о длительности аудиосигналов в файлах (аналогично второй строке)

- в пятой строке отступить 10 пробелов и вывести значения двух переменных, содержащих информацию о частоте дискретизации каждого из аудиосигналов (аналогично второй строке)
- в шестой строке вывести 30 тире .

-----*.MP3 VS .WAV*-----

Имя файла *.wav* VS Имя файла *.mp3*

Метод сжатия *.wav* VS Метод сжатия *.mp3*

Длительность аудиосигнала *.wav* VS Длительность аудиосигнала *.mp3*

Частота дискретизации *.wav* VS Частота дискретизации *.mp3*

17. Используйте команду `disp` и выведите в Command Window информацию об аудио в формате, представленном в пункте 16.

Дополнительные задания

1. Каждую строку песни (*song.txt*) запишите в *.wav* файл с затуханием по отношению к предыдущей строке на 10дБ.
2. Создайте разные тембры любой из нот посредством добавления кратных гармоник.

Чистая нота:

$$s = A * \sin(2\pi f_{\text{ноты}} t)$$

Тембр:

$$s = A_1 * \sin(2\pi f_{\text{ноты}} t) + A_2 * \sin(2\pi * 2f_{\text{ноты}} t) + \\ + A_3 * \sin(2\pi * 3f_{\text{ноты}} t) + \dots A_n * \sin(2\pi * nf_{\text{ноты}} t),$$

где $A_1..A_n$ случайные числа.