



Цифровая обработка сигналов

Лабораторная работа № 4

Транскрибирование музыки

Содержание

| | |
|---|---|
| 1 Теоретические сведения..... | 3 |
| 1.1 Дискретное преобразование Фурье..... | 3 |
| 1.2 Кратковременное преобразование Фурье..... | 3 |
| 1.3 Основы нотной грамоты..... | 4 |
| 2 Практические сведения..... | 7 |
| 3 Задание на лабораторную работу..... | 7 |
| 4 Формат сдачи..... | 8 |
| 5 Контрольные вопросы..... | 8 |
| Приложение А. Пошаговая инструкция..... | 9 |

| | |
|----------------|--|
| Организация: | Самарский университет |
| Подразделение: | Кафедра геоинформатики и информационной безопасности |
| Версия: | 2023.09.28 |

Я сижу, а рядом кот.
Я пою. И кот поёт.
Весело и живо,
Но чуть-чуть фальшиво.
Это потому, что кот
До сих пор не знает нот.

«Мур» поёт он вместо «соль»
«Мяу» вместо «си бемоль»,
Завывает вместо «ре»,
Словно в марте во дворе,
Вместо «ля» зевает
И меня сбивает.
С ним сольфеджио урок
Никогда не будет впрок!
Может, лучше петь дуэтом
Мне с дворовым псом Валетом?

Надежда Радченко,
[«Сольфеджио с котом»](#)

1 Теоретические сведения

1.1 Дискретное преобразование Фурье

Ранее уже было введено и использовалось дискретное во времени преобразование Фурье (ДВПФ). Однако одним из его существенных недостатков для применения на практике является тот факт, что оно ставит в соответствие **дискретному** сигналу **непрерывный** спектр. Если рассматривать конечный по длительности сигнал, то получается, что конечному набору временных отсчётов сопоставляется несчётное множество спектральных. Другими словами, для вычисления обратного ДВПФ требуется взять интеграл. И хотя существуют методы компьютерной алгебры для символьных вычислений и методы численного интегрирования, всё это делает ДВПФ крайне неудобным для практического использования из-за непрерывной формы спектра.

Поэтому на практике повсеместно используется **дискретное преобразование Фурье (ДПФ, DFT, Discrete Fourier Transform)**, в котором количество спектральных отсчётов конечно и равно числу отсчётов во временной форме представления сигнала.

Формулы прямого и обратного ДПФ имеют следующий вид:

$$X[m] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \frac{2\pi}{N} mn},$$
$$x[n] = \frac{1}{N} \sum_{m=0}^{N-1} X[m] \cdot e^{j \frac{2\pi}{N} mn}.$$

1.2 Кратковременное преобразование Фурье

Ранее уже отмечалось, что речевой сигнал не является стационарным процессом. Например, человек произносит какое-нибудь предложение. В каждом слове доминирует свой набор частот, однако вычисление ДВПФ над всем сигналом сразу покажет только **факт наличия** каких-то доминирующих частот, **но не моменты времени**, в которые эти частоты присутствовали в сигнале. Другими словами, глядя на спектр всего сигнала, нельзя сказать, в какие моменты времени доминировали те или иные частоты.

Для решения этой проблемы может применяться **кратковременное преобразование Фурье (КВПФ, STFT, Short-Time Fourier Transform)**, иногда такое преобразование ещё называют оконным преобразованием Фурье. Основная идея заключается в том, чтобы применить преобразование не ко всему сигналу целиком, а к перекрывающимся сегментам фиксированной длительности. При этом возникает следующая дуальность. С одной стороны, чем длиннее сегмент, тем лучше разрешение по частоте, но тем хуже разрешение по времени. С другой стороны, чем короче сегмент,

тем хуже разрешение по частоте, но лучше разрешение по времени. На эту тему рекомендуется посмотреть видео [The more general uncertainty principle](#).

Когда преобразование Фурье вычисляется на искусственно ограниченном сегменте, это равносильно умножению сигнала на прямоугольную оконную функцию, соответствующую рассматриваемому сегменту. По теореме о свёртке это означает, что в спектральной области происходит свёртка спектра исходного сигнала с `sinc`-функцией (которая есть преобразование Фурье прямоугольного импульса). Данный эффект называют **растеканием спектра**. Растекание спектра приводит к тому, что боковые «паразитные» лепестки `sinc`-функции могут маскировать присутствие частот меньшей амплитуды.

Для борьбы с возможными последствиями растекания спектра используют более специальный вид **оконной функции**, спектр которой более гладок по сравнению с `sinc`-функцией. Отметим, что использование непрямоугольного окна специфичного вида не даёт выигрыш на пустом месте — при уменьшении амплитуд боковых лепестков увеличивается ширина основного и наоборот. Таким образом, выбор формы оконной функции определяется компромиссом между точностью определения расположений пиков на спектре и возможностью обнаружения небольших пиков.

С учётом вышесказанного формула вычисления дискретного КВПФ выглядит следующим образом:

$$\text{STFT}\{x[n]\}(m, k) = \text{DFT}\{x_k[n] \cdot w[n]\}[m],$$


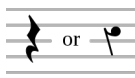


где $x_k[n]$ — k -ый сегмент сигнала $x[n]$, $w[n]$ — используемая оконная функция. Обратите внимание, что результатом вычисления КВПФ является матрица. Каждый столбец в ней соответствует ДПФ очередного сегмента.

1.3 Основы нотной грамоты

В музыке длительность измеряется в условных относительных единицах, называемых **целой нотой**. Меньшие длительности обозначаются половинной нотой (равна по длине половине целой), четвертной (половина половинной), восьмой и т. д. Аналогичным образом обозначается длительность пауз — см. [таблицу 1](#).

Таблица 1 — Обозначение длительности нот и пауз

| Длительность | Обозначение ноты | Обозначение паузы |
|--------------|---|---|
| Целая |  |  |
| Половинная |  |  |

| | | |
|------------|---|---|
| Четвертная |  |  |
| Восьмая |  |  |



Вертикальный отрезок, используемый для нот длительности меньше целой, может быть направлен как вниз, если нота располагается в верхней части нотного стана, так и вверх в противоположном случае. В нашем случае правильная ориентация ноты будет автоматически обеспечиваться соответствующим программным обеспечением.

Привязка относительных единиц длительности к абсолютному времени может быть различной и определяется **темпом**. Как правило, привязка длительности к секундам производится не для целой ноты, а четвертной, потому что они используются чаще. В нотном тексте в начале произведения указывают, сколько четвертных нот должно уместиться в одной минуте. Например, запись

$$\text{♩} = 80$$

означает, что в минуту должно уложиться 80 четвертных нот (следовательно, при таком темпе четвертная нота звучит 0,75 с, а целая нота – 3 с).

Однако темп не всегда задают точным числовым значением — человек всё-таки не машина. Поэтому темп может задаваться словом (например, *adagio* или *allegro*), обозначающим примерный темп — см., например, [темповые обозначения](#).

Высота звука определяется расположением ноты на нотном стане и используемым ключом. Одни из самых известных ключей — басовый  и скрипичный . Для сопоставления расположения ноты с частотой её основного тона см. [рисунок 1](#).

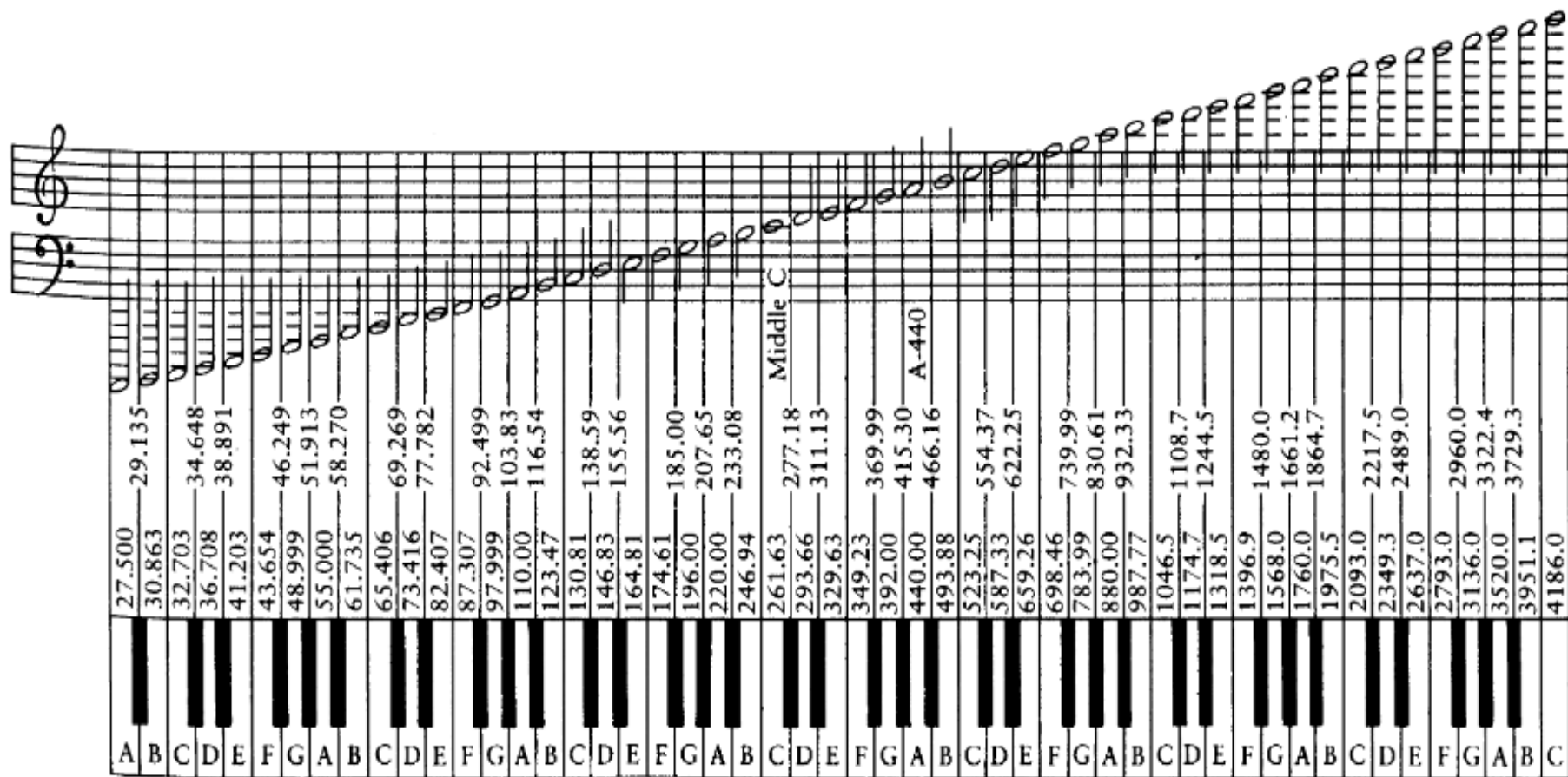


Рисунок 1 — Расположение нот на нотном стане

Источник: [Frequenze della Musica, perché è importante riconoscerle](#)

2 Практические сведения

| Пакет | Функция |
|------------|--|
| matplotlib | matplotlib.pyplot.pcolormesh |
| numpy | numpy.abs |
| scipy | scipy.signal.stft |

3 Задание на лабораторную работу

Часть 1. Из репозитория курса скачайте заготовку `lab4.py`. В скачанном файле имплементируйте функцию `dft` для вычисления ДПФ (см. формулу в подразделе [1.1](#)). Реализация данной функции должна быть своя (запрещается использовать готовые функции `dft/fft` из каких-либо библиотек).

В файле уже предоставлен готовый тест для проверки вашей реализации. Для проверки запустите файл — вы должны увидеть строку `OK (skipped=1)`.

Часть 2. В том же файле имплементируйте функцию `real_stft` для вычисления дискретного КВПФ (см. формулу в подразделе [1.2](#)). Обратите внимание на следующие моменты.

1. Функция должна работать аналогично функции `stft` из библиотеки `scipy` (ознакомьтесь с документацией!). См. аргументы вызова этой функции в проверяющем методе-тесте `test_stft`, чтобы понять, какое именно поведение ожидается от вашей реализации.

2. При реализации используйте функцию `dft`, реализованную в первой части задания. Библиотечная `stft` использует нормированное прямое ДПФ (с коэффициентом $1/N$), тогда как в текущем курсе нормировка используется в обратном ДПФ. Это различие *учитывается* проверяющим тестом.

3. Реализация данной функции должна быть своя — запрещается использовать готовые функции `dft/fft/stft` из каких-либо библиотек.

В файле уже предоставлен готовый тест для проверки вашей реализации, но он помечен аннотацией `@unittest.skip` (чтобы этот тест не мешался в первой части). Для проверки уберите вышеупомянутую аннотацию и запустите файл — вы должны увидеть строку `OK`.

Часть 3. В том же файле `lab4.py` в функции `main` напишите код для загрузки музыкального фрагмента (нужный файл скачивается из репозитория курса согласно вашему варианту). Далее вычислите и визуализируйте STFT. По имеющейся визуализации транскрибируйте музыкальный фрагмент в нотную запись. Для вычисления STFT используйте более быструю библиотечную реализацию.

Визуализация квадрата модуля КВПФ называется [спектрограммой](#). Для разметки оси координат в абсолютных единицах (герцы, секунды) используются массивы `f` и `t`, которые возвращает функция [stft](#). Чтобы значения этих массивов соответствовали действительности, при вызове необходимо указать корректное значение частоты дискретизации `fs`. Пример кода для вычисления и визуализации STFT:

```
import matplotlib.pyplot as plt
import numpy as np

from scipy.signal import stft

x, fs = ...

f, t, spectrum = stft(x, fs)
plt.figure('Spectrogram')
plt.pcolormesh(t, f, np.abs(spectrum) ** 2)
plt.xlabel('Time [sec]')
plt.ylabel('Frequency [Hz]')
```

По спектрограмме определите, какая длительность встречается в файле чаще всего и примите эту длительность за четвертную ноту. Далее вычислите темп (количество четвертных нот в минуту).

Создайте с помощью программы [MuseScore](#) новый `mscz`-документ (или используйте готовый шаблон `lab4.mscz` из репозитория курса) и транскрибируйте музыкальный фрагмент в нотную запись, используя полученную спектрограмму. Для минимизации времени ознакомления с графическим интерфейсом можете воспользоваться [пошаговой инструкцией](#).

Опциональное усложнение. Автоматизируйте процесс определения темпа произведения, высоты и длительности нот. Скрипт должен не только визуализировать STFT, но и выводить в стандартный поток вывода автоматически определённые темп произведения, высоту и длительность нот, длительность пауз. Высоту ноты выводить либо в [американской системе нотации](#), либо в герцах. Длительность ноты и пауз выводить в относительных единицах: `1` для целой ноты; `1/2`, `1/2` или `0,5` для половинной и т. д.

4 Формат сдачи

Предоставить скрипт с кодом визуализации STFT и `mscz`-документ с транскрибированными нотами.

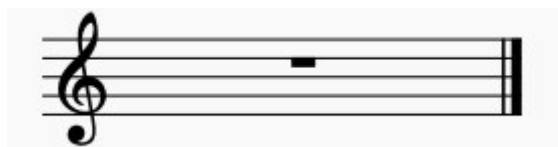
5 Контрольные вопросы

ДПФ и КВПФ. Основы нотной грамоты.

Приложение А. Пошаговая инструкция

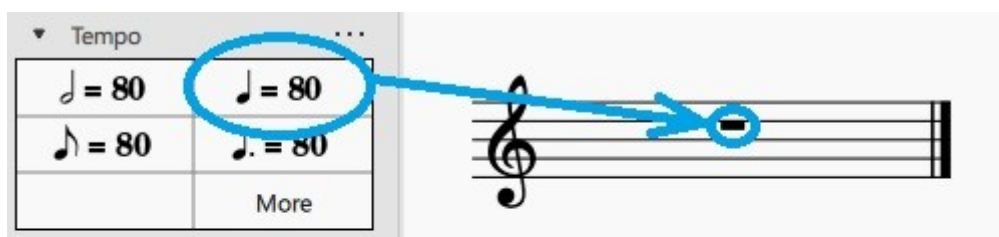
Шаг 1. Скачайте и установите [MuseScore](#).

Шаг 2. Скачайте из репозитория курса и откройте в установленном приложении готовый шаблон документа `lab4.mscz`. Вы должны увидеть подготовленный нотный стан со скрипичным ключом и целой паузой:



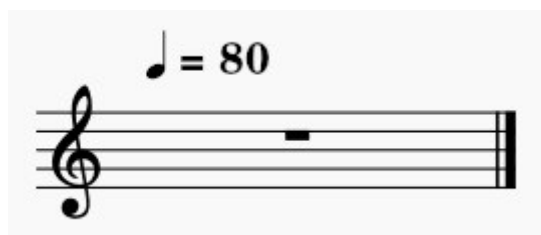
Шаг 3. Укажите темп произведения одним из следующих способов.

Способ 1. В панели инструментов слева найдите секцию **Темпо** и перетащите указанный значок на символ паузы.



Способ 2. Выберите символ паузы на нотном стане (он должен стать голубым), после чего щёлкните на значок **♩ = 80** в секции **Темпо** на панели инструментов слева.

В итоге должно получиться следующее:



Замените число 80 на определённый вами темп.

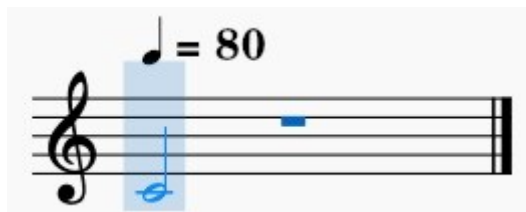
Шаг 4. Войдите в режим ввода нот, щелкнув на первую кнопку в панели инструментов сверху:



Шаг 5. На той же панели инструментов выберите необходимую длительность ноты (на примере выбрана половинная):

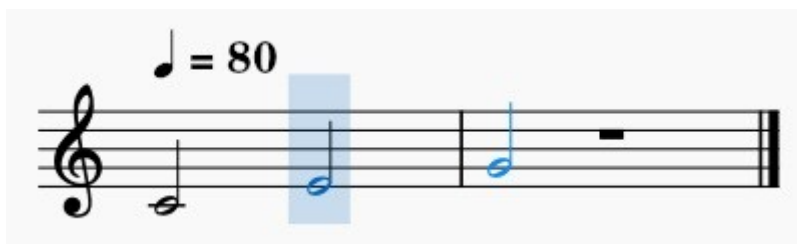


Разместите ноту на нотном стане на нужной высоте кликом мышки:



Альтернативный способ размещения нот — это показать панель виртуального пианино и набирать ноты на нём. Для включения/выключения панели используйте клавишу **P** или пункт меню **View → Piano Keyboard**.

Шаг 6. После того, как вы исчерпаете доступное время такта (по умолчанию это целая нота), добавьте при необходимости следующий такт с помощью комбинации **Ctrl+B** или через пункт меню **Add → Measures → Append One Measure**. После добавления второго такта должно получиться следующее:



Шаг 7. После завершения размещения всех нот выйдите из режима редактирования ввода нот (нажав на ту же кнопку, что и на шаге 4). Используйте **Space** для прослушивания получившегося результата или нажмите на соответствующую кнопку воспроизведения в панели инструментов сверху:



Шаг 8. Убедитесь на слух, что работа выполнена верно — воспроизведение подготовленного нотного документа должно быть максимально похоже на выданный вам музыкальный фрагмент.