Санкт-Петербургский Государственный Университет
Математико-Механический факультет
Кафедра Системного Программирования

Получение нотной записи одноголосного музыкального сигнала

Дипломная работа студента 545 группы Сергея Ильи Дмитриевича

Научный руководитель: ..... д.ф.-м.н., профессор Барабанов А. Е.

_____
/подпись/

Рецензент: ..... д.ф.-м.н., профессор Граничин О. Н.

_____
/подпись/

„Допустить к защите“
Заведующий кафедрой: ..... д.ф.-м.н., профессор Терехов А. Н.

_____
/подпись/

Санкт-Петербург
2008 г.

Аннотация

Целью данной работы являлось создание программного средства для выделения нотной записи из оцифрованного одноголосного музыкального сигнала. Впоследствии этот же подход планируется расширить и для многоголосных сигналов. В данный момент предлагаемый алгоритм решает задачи быстрого нахождения приближения периода основного тона музыкального сигнала в отсчётах дискретизации, последующего его уточнения, а также устранения так называемой «ошибки октав». В качестве входных данных используется файл в формате `WAV` (Waveform Audio Format), содержащий сигнал. Результатом работы программы является `MIDI`-файл либо нотная запись сигнала в виде `PDF`-документа.

Содержание

1 Введение

В последние годы всё большее значение придаётся задачам распознавания и поиска в больших объёмах специфичных данных. Таковыми могут являться, к примеру, большие хранилища изображений, видеоинформации или музыкальных данных.

1.1 Мотивация

Наверное, каждый человек, которому в какой-то степени небезразлична музыка, попадал в ситуацию, когда по услышанному мотиву или мелодии хотелось бы найти оригинал. В подобных случаях далеко не всегда удаётся формализовать свой запрос в виде текста, поэтому следует искать другие пути формализации запросов и критериев поиска. Задача поиска похожей музыки, столь простая в формулировке, до сих пор не имеет приемлемой практической реализации. Видимо, всё дело в том, что музыкальные записи (в дальнейшем называемые здесь сигналами) представляют собой особый вид данных, на которых крайне сложно определить формальную метрику без наличия какой бы то ни было дополнительной информации. Под дополнительной информацией можно понимать, к примеру, текстовые данные, содержащиеся в *тегах* музыкальных MP3-файлов (сведения об исполнителе, композиции, жанре и т. п.). Есть подходы к решению проблемы нахождения «похожей» музыки, основанные на сборе статистики по такого рода данным. По всей видимости, сейчас наиболее популярным из проектов, реализующих такой подход, является музыкальная база знаний **LastFM** [1].

Нас интересует немного другой случай, а именно, получение информации по «чистому» сигналу, не снабжённому дополнительными подсказками. В том случае, если музыкальная партия уже записана в некоторой общепринятой форме, скажем, с использованием нотной грамоты, можно применить стандартные алгоритмы поиска подстрок или выделения метрик на строках в кластеризованных хранилищах данных. Косвенным образом этой проблеме посвящена, к примеру, статья [3], где авторами рассматривается алгоритм выделения повторяющихся фрагментов мелодии («куплетов») по некоторому символьному представлению пьесы.

[1] `http://www.last.fm/`

1.2 Постановка задачи

Задачей данной работы является разработка алгоритма и программы для получения по оцифрованному музыкальному сигналу некоторого *адекватного* представления, с которым было бы удобно работать в упомянутых задачах сопоставления и поиска. Таким представлением является, к примеру, общепринятая нотная запись музыкальных произведений. Под *адекватностью* в данном случае понимается достаточно хороший уровень соответствия того, что будет получено при «исполнении» музыкантом эквивалентной представлению нотной записи и исходного сигнала.

Достаточно давно стандартом *de-facto* для унифицированного представления музыкальных данных является формат `MIDI`. В отличие от других звуковых форматов, `MIDI` хранит не оцифрованный звук, а наборы команд (проигрываемые ноты, ссылки на проигрываемые инструменты, значения изменяемых параметров звука), которые могут воспроизводиться по-разному в зависимости от устройства воспроизведения. По сути, `MIDI`-файл является всего лишь немного другой нотной записью музыкального произведения. Кроме того, он удобен для воспроизведения на ряде устройств, снабжённых `MIDI`-секвенсором. Потому было бы логичным в качестве одного из вариантов порождаемого результата иметь `MIDI`-файл. В качестве входных данных итоговая программа принимает файл в формате `WAV` (Waveform Audio Format) - наиболее простом и удобном формате для представления оцифрованных аудиоданных. Сознательно не берутся в расчёт другие форматы наподобие MP3, OGG, ATRAC, использующие дополнительные алгоритмы сжатия. Задача восстановления по ним исходного сигнала интереса не представляет, а потому не рассматривается. В дальнейшем планируется использовать существующие средства декомпрессии для удобства получения нотного представления, скажем, сразу по MP3-файлу и не требовать от пользователя программы промежуточного конвертирования файла в `WAV`-формат.

1.3 Доступные программные средства

В данный момент на рынке доступен ряд коммерческих программных инструментов, которые решают аналогичную задачу в терминах получения по файлу с оцифрованным сигналом файла в формате `MIDI`. Наиболее известными из них являются *IntelliScore*, *AKoff Music Composer* и *AmazingMIDI*. Поскольку эти средства нацелены не только на решение поставленной задачи, но и на последующую обработку `MIDI`-файла, можно отметить некоторые достаточно большие ошибки в их работе по анализу входного сигнала, которые будут рассмотрены подробнее ниже.

Кроме того, все они рассчитаны на работу в операционных средах семейства *Microsoft Windows*, что значительно сужает круг их пользователей.

1.4 Полученные результаты

В данной дипломной работе рассматриваются основные проблемы, возникающие при распознавании музыкальных сигналов, производится их классификация и предлагаются методы решения проблемы распознавания на основе Фурье-анализа сигнала.

Итогом работы является реализация алгоритма, выделяющая нотное представление по одноголосному музыкальному сигналу. Кроме того, в заключительной части работы представлен беглый обзор коммерческих программных средств, решающих сходные задачи. Проводится сравнение результатов их работы с предлагаемым алгоритмом.

2 Модель музыкального сигнала и характерные ошибки анализа

Исходный сигнал представляет собой оцифрованную с частотой 22050 и 44100 отсчётов запись игры музыкального инструмента. В большинстве рассматриваемых ниже примеров таким инструментом фортепиано.

Даже одноголосный музыкальный сигнал представляет из себя весьма сложную для анализа систему. Традиционно, в задачах выделения частотных характеристик сигнала используется Фурье-анализ. Зачастую это приводит к ряду ошибок, которые должна учитывать модель исследуемой системы.

2.1 Выбор длины сегмента

Как правило, музыкальные сигналы являются нестационарными (в противном случае, прослушивание музыки было бы весьма утомительным и однообразным занятием). Однако же, мало изменяющиеся колебательные свойства могут проявляться на небольших временных промежутках. Потому, как правило, происходит разбиение всего сигнала на сегменты фиксированной длины (фреймы) и последующий анализ свойств сигнала на каждом из таких промежутков. Целостная картина получается «склеиванием» результатов, полученных на идущих подряд сегментах.

В процессе всей работы многие действия производятся только на отдельном взятом сегменте, в том числе определение воспринимаемой *частоты основного тона*. *Частота основного тона* для одного голоса – это наибольший общий делитель частот его обертонов.

Промежутки, на которых имеет смысл считать сигнал стационарным, не учитывая погрешностей, опять же, могут весьма сильно варьироваться, что во многом зависит от темпа воспроизведения музыки и характеристик инструмента. Потому идеальным вариантом было бы изменять длину фрейма, выбирая оптимальную для каждого конкретного участка сигнала. Такой подход оправдан в том случае, если на программу не накладываются жёсткие ограничения по производительности.

2.2 Опредение частоты основного тона

Звук, который воспринимается человеком как слышимая нота, характеризуется частотой основного тона колебаний воздуха, порождаемых звучащим элементом, скажем, гитарной струной. Нота «ля» первой октавы

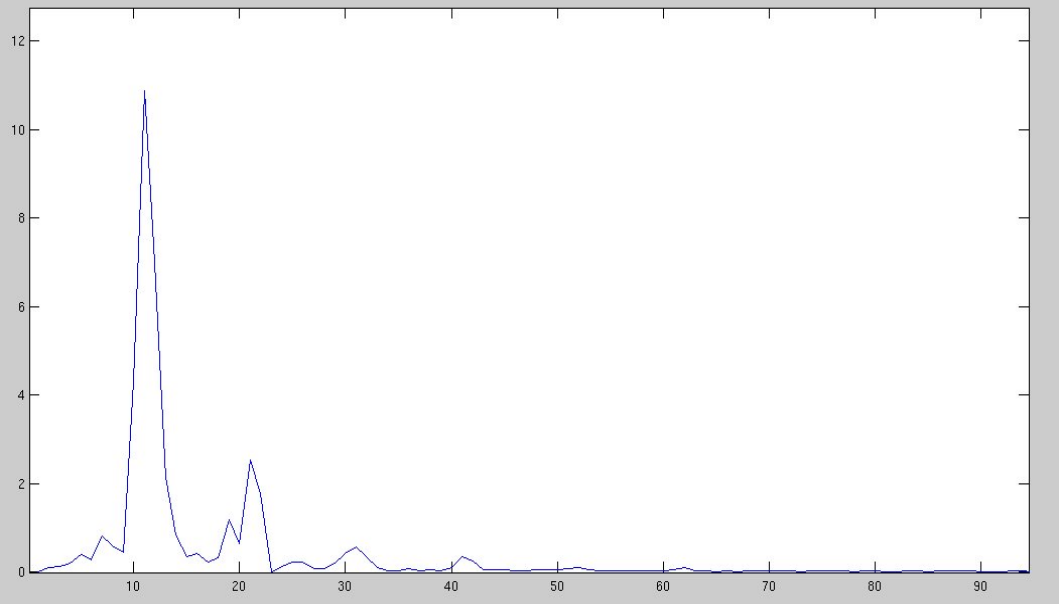


Рис. 1: Дискретное значение частоты основного тона

фортепиано обладает частотой основного тона 440 Hz. Так как анализируемый программой сигнал содержит некоторое количество шумов, а также из-за погрешности дискретизации, частота основного тона может быть определена с помощью дискретного преобразования Фурье лишь с точностью до некоторого целого значения, зависящего от частоты дискретизации и выбранной длины сегмента, на котором производился анализ. На рис. 1 спектральный максимум приходится на 11 отсчёт ДПФ, что при заданных параметрах дискретизации и выбора сегмента даёт частоту ноты «ля» первой октавы 470 Hz, тогда как 10 отсчёт соответствует 430 Hz. Очевидно, что правильное значение находится где-то посередине. После такого «грубого» первого приближения требуется «более точная подстройка» для выяснения настоящей частоты основного тона.

2.3 Ошибка октав

Но даже достаточно точно определив частоту основного тона по методу спектрального максимума, мы не можем гарантировать, что не ошиблись. Это может произойти в том случае, когда большой амплитудой обладала синусоидальная составляющая, соответствующая какой-либо из кратных гармоник. Из-за этого происходит так называемая «ошибка октав». Получающийся в результате анализа звук, отличается от реаль-

но слышимого в исходном сигнале на несколько октав либо октаву и квинту (см. [1]), что соответствует получению частоты, отличающейся от истинной в натуральное число раз. При этом, к примеру, ошибке на одну октаву вверх соответствует определённая частота в два раза выше истинной.

3 Теоретическая часть

В основном алгоритме по выделению нотного представления музыкального сигнала используются идеи по выделению формант, используемые при создании голосовых вокодеров. Далее описаны основные этапы работы алгоритма:

1. Разбиение исходного музыкального сигнала на перекрывающиеся сегменты.

2. Вычисление целого приближения периода основного тона сигнала для каждого из сегментов посредством минимизации функционала качества на множестве допустимых целых приближений.

3. Уточнение периода основного тона. Для этого производится минимизация невязки между известной линейной огибающей дискретного спектра и интерполяционной функцией специального вида с узлами в предполагаемых точках нахождения уточнённого периода основного тона.

4. Устранение «ошибки октав» производится путём подсчёта спектральной огибающей для сегментов сигнала как стационарного случайного процесса методом Левинсона-Дурбина. Далее методом наименьших квадратов оценивается степень приближения сигналом, восстановленным по полученной частоте основного тона данной спектральной огибающей. При этом перебираются различные кратные частоты и выбирается та, на которой достигается наилучшее приближение.

3.1 Деление сигнала на сегменты

Изначально весь сигнал делится на сегменты. В простейшем случае сегменты имеют одинаковую длину. Эта длина может задаваться в зависимости от заранее известных характеристики инструмента. Так, к примеру, для анализа большинства записей фортепиано подходит длина сегмента в 512 или 1024 отсчёта, тогда как для гитары наиболее оптимальной является длина в 2048 отсчётов. Кроме того, длина сегмента может изменяться в зависимости от скорости изменения сигнала, но этот подход неприменим в том случае, если от программы требуется большая производительность. Для определения оптимальной длины сегмента используется метод минимизации функционала качества, рассматриваемый ниже.

Сегменты имеют общие части, другими словами, каждый следующий сегмент получается небольшим сдвигом из предыдущего. В нашем случае величина сдвига равняется 64 или 128 отсчётам в зависимости от частоты квантования. Итоговое значение частоты основного тона (то есть звучащей ноты), полученного на сегменте, приписывается середине этого сегмента. Анализ производится на каждом отдельно взятом сегменте. В заключительной стадии работы алгоритма происходит «склеивание» данных по соседним сегментам для получения продолжительных нот.

3.2 Вычисление целого значения периода основного тона

В этом разделе предполагается, что частота дискретизации и соответствующее количество измерений достаточно велики, чтобы можно было пренебречь погрешностями дискретизации. При этом вычисляется грубая оценка периода основного тона, обычно равная ближайшему к нему целому числу. Важно заметить, что модель сигнала, рассматриваемая в этом разделе, предполагает, что период основного тона - целое число. Разумеется, в общем случае это не так, но, как показывают эксперименты, данный подход даёт хорошее приближение реального периода основного тона целым числом.

Пусть исходный сигнал на данном сегменте длины N представляет собой последовательность вида

$$s(n) = s_0(n) + d(n)$$

где $s_0(n)$ - исходный идеальный сигнал, а $d(n)$ - возмущение. После умножения на оконную функцию $w(n)$ сигнал представляется в виде

$$s_w(n) = w(n)(s_0(n) + d(n))$$

а его преобразование Фурье соответственно

$$S_w(\omega) = \sum_{n=0}^{N-1} s_w(n) e^{-i\omega n}$$

3.2.1 Модель музыкального сигнала и функционал качества

Для одноголосного музыкального сигнала период основного тона обозначим P^0. Тогда спектральный диапазон $[0, 2\pi]$ разделяется на P^0 спектральных полос вида $[pF^0, (p+1)F^0)$, $0 \le p \le P^0 - 1$, где $F^0 = 2\pi/P^0$ - искомая частота основного тона.

Определим класс моделей данного сигнала, которые различаются только периодом основного тона P. На данном этапе рассуждений предполагаем преобразование Фурье непрерывным, либо считаем частоту дискретизации достаточно большой. В этом случае идеальный сигнал s_0 представляет собой сумму синусоид с частотами kF, где $k \in [1 \dots (P-1)]$. Преобразование Фурье такого сигнала, очевидно, является линейной комбинацией δ-функций с коэффициентами $A_p \in \mathbb{C}$, представляющими собой амплитуду и фазу соответствующих гармоник. Домножение на оконную функцию $w(\cdot)$ даёт в преобразовании Фурье свёртки $W(\omega)$ с δ-функциями $\delta(\omega - c_p)$.

На этом классе моделей зададим функционал качества как минимальное значение выражения

$$\mathcal{E} = \frac{1}{2\pi} \sum_{p=1}^{P-1} \int_{a_p}^{b_p} |S_w(\omega) - A_p W(\omega - c_p)|^2 d\omega \tag{1}$$

по набору комплексных чисел A_p при $0 \le p \le P^0 - 1$, где $c_p = pF$, $a_p = c_p - F/2$, $b_p = c_p + F/2$ и $W(\omega)$ - преобразование Фурье оконной функции $w(n)$. Заметим, что суммирование по p происходит не с нуля, а с единицы, так как в реальных сигналах нулевая частота означает некоторое изначально заданное постоянное напряжение ЦАП, которое далее убирается путём фильтрации.

В силу вышесказанного минимум по P на классе моделей достигается на истинном значении P^0, так как спектр сконцентрирован в точках c_p, после умножения на окно каждая единичная нагрузка в соответствующих точках преобразуется в функцию W, а параметры A_p равны значениям спектра идеального сигнала s_0.

3.2.2 Минимизация функционала качества

Воспользовавшись методом наименьших квадратов для минимизации выражения (1) по по комплексным числам A_p, получаем, что минимум нашего функционала равен

$$\tilde{\mathcal{E}} = \frac{1}{2\pi} \sum_{p=1}^{P-1} \left(\int_{a_p}^{b_p} |S_w(\omega)|^2 d\omega - |\int_{a_p}^{b_p} S_w^*(\omega) W(\omega - c_p) d\omega|^2 / \int_{a_p}^{b_p} |W(\omega - c_p)|^2 d\omega \right)$$

Далее заметим, что величина

$$F_w = \int_{a_p}^{b_p} |W(\omega - c_p)|^2 d\omega = \int_{-F/2}^{F/2} |W(\omega)|^2 d\omega$$

не зависит от p и приближённо равна энергии последовательности $W(n)$, если положить, что $W(\omega) = 0$ при $|w| > F/2$. Далее будем полагать, что оконная функция $w(\cdot)$ выбрана таким образом, чтобы удовлетворять этому условию. Кроме того, мы будем считать, что оконная функция нормирована на данном сегменте длины N, так, что $||w(n)||_{L_2} = N$ а потому, согласно равенству Парсеваля, в вычислительной части алгоритма будем полагать $F_w = N$.

Далее перепишем $\tilde{\mathcal{E}}$, сделав замену переменной во втором интеграле:

$$\tilde{\mathcal{E}} = \frac{1}{2\pi}\int_{-\pi}^{\pi} |S_w(\omega)|^2 d\omega - \sum_{p=1}^{P-1} |\frac{1}{2\pi}\int_{-\pi}^{\pi} S_w^*(\omega + c_p)W(\omega)d\omega|^2/F_w$$

Заметим также, что, согласно равенству Парсеваля выполняются следующие тождества:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |S_w(\omega)|^2 d\omega = \sum_{n=0}^{N-1} w^2(n)(s_0(n) + d(n))^2$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} S_w^*(\omega + c_p)W(\omega)d\omega = \sum_{n=0}^{N-1} w(n)w_0(n)(s_0(n) + d(n))e^{inc_p}$$

Далее предположим, что случайный шум $d(n)$ - *белый*, то есть обладает нулевым средним и постоянной дисперсией σ^2. В таких предположениях получаем, что несмещённым (т. е. математическое ожидание оценки совпадает с вектором правильных параметров) является следующий критерий:

$$\tilde{\mathcal{E}}_{UB} = \tilde{\mathcal{E}}/(F_w - P\sum_n w^4(n)/F_w) - \sigma^2$$

3.2.3 Упрощение вычислений

Дальнейшая задача стоит в упрощении вычислительных аспектов расчёта значения $\tilde{\mathcal{E}}$. Уже показано, что

$$\tilde{\mathcal{E}} = \sum_{n=0}^{N-1} w^2(n)s^2(n) - \sum_{p=1}^{P-1} |\sum_n w^2(n)s(n)e^{inc_p}|^2/F_w \tag{2}$$

Заметим, что

$$|\sum_n w^2(n)s(n)e^{inc_p}|^2 = |\sum_{n=0}^{N-1} \underbrace{w^2(n)s(n)e^{2\pi inp/P}}_{\eta_n}|^2 \tag{3}$$

Последняя сумма может быть переписана при помощи значений эмпирической корреляционной функции r_m от последовательности $w^2(n)s(n)$ как стационарного случайного процесса.

В нашем случае

$$r_m = \sum_n w^2(n)s(n)w^2(n-m)s(n-m) \tag{4}$$

Раскрывая сумму

$$|\sum_{n=0}^{N-1} \eta_n|^2 = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} \eta_n\overline{\eta_k}$$

и дополняя её нулями до $-\infty$ и $+\infty$, делаем замену переменных, группируя слагаемые в обеих суммах, отстоящие друг от друга на m, где $m = 0 \dots (N-1)$, можем записать выражение (3) как

$$|\sum_n w^2(n)s(n)e^{inc_p}|^2 = \sum_{m=0}^{N-1} r_m e^{2\pi ipm/P} \tag{5}$$

Можно также заметить, что вторая сумма из в (2) принимает в этом случае вид

$$\sum_{p=1}^{P-1}\sum_{m=0}^{N-1} r_m e^{2\pi ipm/P}/F_w \tag{6}$$

из чего видно, что слагаемые, для которых mp не кратно P дают в сумме ноль. Потому остаются только те слагаемые, для которых $mp = Pk$, $k \in \mathbb{Z}$. Так как $p < P$, то в сумме (6) остаётся P слагаемых. Такие слагаемые обозначим r_{P_m}. При реализации алгоритма они считаются только один раз для всего сегмента. Окончательно получаем:

$$\tilde{\mathcal{E}} = \sum_n w^2(n)s^2(n) - P\sum_m r_{P_m}/F_w \tag{7}$$

Таким образом справедливо следующее утверждение.

Утверждение:

Пусть для зашумлённого сигнала $s(n) = s_0(n) + d(n)$, где $s_0(n)$ - полезный сигнал, а $d(n)$ - случайный шум, выполнены следующие условия:

1. Оконная функция $w(\cdot)$ выбирается так, что преобразование Фурье $W(\omega)$ равно нулю при $w > 2\pi/P$, где P-период основного тона сигнала $s_0(n)$.

2. Случайный шум $d(n)$ - белый.

3. Полезный сигнал s_0 представляет собой сумму синусоид с кратными частотами.

Тогда несмещённой оценкой числа P^0 - истинного периода основного тона сигнала $s(n)$ является точка минимума функционала

$$\boxed{\tilde{\mathcal{E}}_{UB} = (\sum_n w^2(n)s^2(n) - P\sum_m r_{P_m}/F_w)/(F_w - P\sum_n w^4(n)/F_w)} \quad (8)$$

где $F_w = \sum_n w^2(n)$ - энергия оконной функции, и r_t - корелляция последовательности $w^2(n)s(n)$:

$$r_t = \sum_n w^2(n)s(n)w^2(n-t)s(n-t)$$

При реализации этой части алгоритма программы минимум ищется на множестве «достоверных» периодов основного тона. В качестве минимального значения брался период ноты «ля» октавы. В то же время, нижняя граница выбиралась из тех соображений, что в одном сегменте должно уложиться хотя бы три периода длины P^0.

Кроме того, можно заметить что (4) очень похоже на свёртку последовательности $w^2(n)s(n)$ с самой собой с той лишь разницей, что «неподвижный» параметр m взят с минусом. Удобно считать это выражение, взяв преобразование Фурье от данной последовательности, домножив на сопряжённое и обратив, сведя таким образом задачу подсчёта свёртки во временной области к произведению функций в частотной области.

3.3 Уточнение периода основного тона

Следующая задача состоит в получении более точного периода основного тона. Будем считать, что далее мы имеем дело с идеальным сигналом $s(n)$, домноженным на оконную функцию $w(n)$.

3.3.1 ДПФ оконной функции и «колокольчики»

Как видно из приведённых выше рассуждений модуль спектра сигнала, умноженного на окно представляет собой линейную комбинацию «колокольчиков» - модулей образов преобразования Фурье оконной функции с центрами в точках, где сосредоточен спектр изначального сигнала. Вид такого «колокольчика» зависит от его положения относительно узлов сетки. Например, если значение частоты попадает в узел сетки, то колокольчик имеет один максимум в этой точке и симметричные склоны влево и вправо. Если же частота оказывается ровно в середине между

отсчетами, то колокольчик содержит два одинаковых максимума. В нашем случае каждый «колокольчик» определяется пятью числами, которые соответствуют отсчетам на данном сегменте. По виду колокольчика можно предсказать положение порождающей его частоты с точностью, намного превышающей расстояние между отсчетами. На этом наблюдении основана процедура более точного определения периода основного тона, чем это было сделано в предыдущем разделе.

Пусть далее N - достаточно большое количество отсчётов в одном сегменте. В нашем случае это обычно степень двойки. Пусть f_0 - некоторое целое число, такое что $3 \leq f_0 \leq \frac{N}{2} - 3$, и пусть $\delta \in [-0.5, 0.5]$. Под оконной функцией будем далее понимать окно Ханнинга

$$w(k) = \frac{1}{2}\left[1 - cos\left(\frac{2\pi}{N}k\right)\right], 0 \leq k \leq N-1$$

S_n - преобразование Фурье нашего сигнала, домноженного на оконную функцию.

$$S_n = \sum_{k=0}^{N-1} s(k)w(k)e^{2\pi ikn/N}$$

Пусть f_0 из наших обозначений - некоторое целое значение частоты. Нормированный «колокольчик» для неё задаётся формулой.

$$C_n = \begin{cases} 0, & \text{при } n = -2 \text{ или } n = 2 \\ -1/\sqrt{6}, & \text{при } n = -1 \text{ или } n = 1 \\ \sqrt{2/3}, & \text{при } n = 0 \end{cases} \tag{9}$$

Построение *нормированного* «колокольчика» для некоторой частоты $f = f_0 + \delta$, $\delta \neq 0$ также требует пяти точек вокруг f_0 и определяется формулой:

$$C_n = \frac{1}{N\sqrt{3/2}} \sin(\pi\delta) \sin^2 \frac{\pi}{N} \frac{\cos\frac{\pi(n-\delta)}{N}}{\sin\frac{\pi(n-\delta)}{N} \sin\frac{\pi(n-1-\delta)}{N} \sin\frac{\pi(n+1-\delta)}{N}} \tag{10}$$

3.3.2 Аппроксимация ДПФ сигнала «колокольчиками»

Вернёмся к нашей задаче. На данный момент уже найдено некоторое целое значение частоты основного тона P^0, а значит и соответствующей частоты $F^0 = N/P^0$. Будем искать наилучшего кандидата на роль уточнённой частоты основного тона на промежутке $F \in [F_{min}, F_{max}]$, где $F_{min} = N/(P^0 + 1)$, $F_{max} = N/(P^0 - 1)]$, следующим образом.

Для каждого кандидата из этого промежутка будем строить набор кратных частот [2]. Для всех кратных частот f_k из этого набора находим ближайшую целую частоту f_{0_k}, а также δ_k такое, что $f_k = f_{0_k} + \delta_k$. Далее для каждой частоты f_{0_k} рассматриваем вырезку 5 точек из S_n, сконцентрированных вокруг f_{0_k}: $S_n[f_{0_k} - 2] \dots S_n[f_{0_k} + 2]$. Кроме того, строим соответствующий «колокольчик» по формулам (9) или (10). Дальнейшая задача сводится к уменьшению суммарной невязки между такими вырезками сигнала и соответствующими «колокольчиками» по всем кратным частотам. Другими словами, мы рассматриваем задачу вида

$$\mathbf{J}(F) = \sum_{k=1}^{K} |S_k(F) - c_k B_k(F)|^2 \to \min \tag{11}$$

В (11) K - выбранное число рассматриваемых кратных частот, $S_k(F)$ - 5-точечная вырезка из S_n, соответствующая k-й кратной частоте кандидата F, $B_k(F)$ - соответствующий колокольчик. Задачи такого рода решаются методом наименьших квадратов, и в данном случае минимальное значение такого функционала по вектору $< c_k >$ выглядит как

$$\tilde{\mathbf{J}}(F) = \sum_k |S_k(F)|^2 - \sum_k | < S_k(F), B_k(F) > |^2$$

Здесь под $< S_k(F), B_k(F) >$ понимается скалярное произведение соответствующих векторов.

3.3.3 Метод Ньютона

Таким образом задача свелась к минимизации полученных значений $\tilde{\mathbf{J}}(F)$ на множестве значений, принимаемых F. Учитывая то, что данное множество представляет собой отрезок, а $\tilde{\mathbf{J}}(F)$ задана аналитически, $\tilde{\mathbf{J}}(F) \in C^2([F_{min}, F_{max}])$ и, как правило, имеет на данном отрезке один глобальный минимум, мы можем воспользоваться методом Ньютона, применённым к производной $\tilde{\mathbf{J}}(F)$ для быстрого поиска значения F_*, такого что

$$\left.\frac{d\tilde{\mathbf{J}}(F)}{dF}\right|_{F=F_*} = 0$$

На рис. 2 изображён график типичного вида функции $\tilde{\mathbf{J}}(F)$. По оси абс-

[2]Количество таких кратных частот ограничивается из соображений некоторого выбранного энергетического порога. К примеру, можно сосчитать энергию преобразования Фурье по частотам от нулевой до частоты Найквиста и далее рассматривать только те частоты, для которых эта суммарная энергия составляет менее 80% от общей. Как показывает практика, идущее далее частоты не вносят существенного вклада в общую картину.

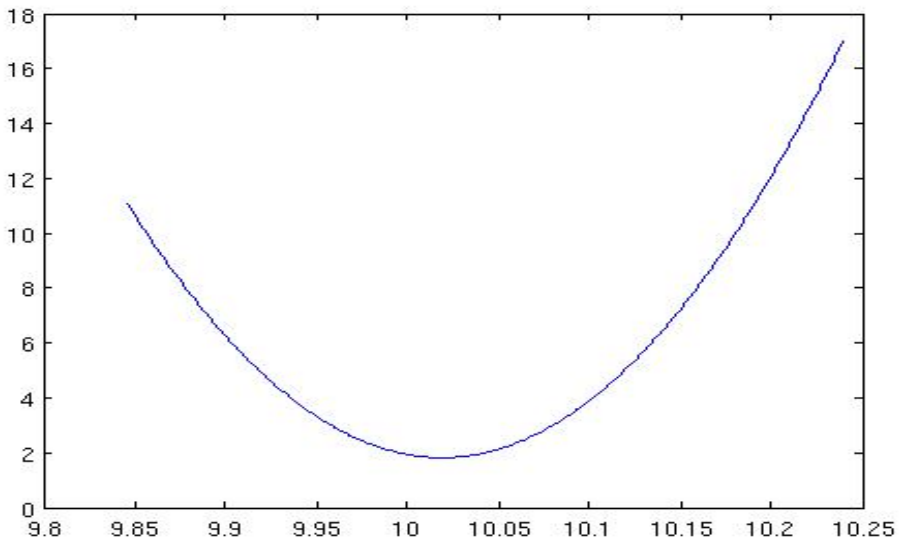


Рис. 2: График зависимости значений $\tilde{\mathbf{J}}(F)$ от частоты F

цисс расположены значения F, на которых производится поиск. Важно понимать, что частота F в данном случае измеряется в отсчётах дискретного преобразования Фурье.

Необходимые для реализации метода Ньютона первая и вторая производные функции $\tilde{\mathbf{J}}(F)$ могут быть посчитаны аналитически, но, как показали эксперименты, быстродействие программы увеличивается, если считать их численно. Кроме того, при реализации следует проверять, что в процессе приближения значения мы не ушли за границу отрезка $[F_{min}, F_{max}]$. В этом случае ищем середину отрезка между значениями, полученными на последней и предпоследней итерациях пополам и продолжаем поиск. Правило остановки для итераций метода Ньютона определяем по некоторой заранее заданной величине ϵ.

3.4 Исправление ошибки октав

Минимум функционала (8) может достигаться как на «истинном» значении периода основного тона P, так и на всех его кратных, что ведёт к так называемой «ошибке октав». Для её устранения используется следующий алгоритм.

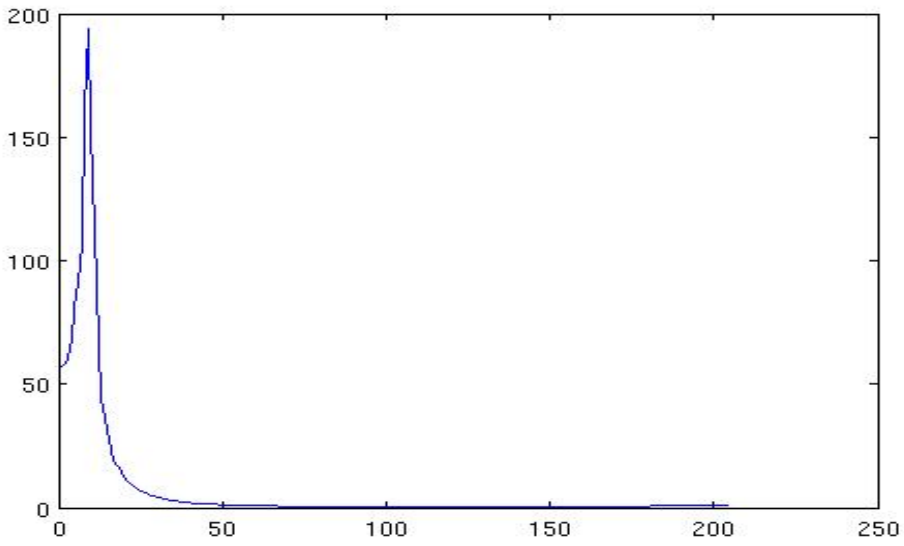


Рис. 3: График модуля спектральной огибающей сигнала на сегменте

3.4.1 Вычисление спектральной плотности сигнала

В противном случае в очередной раз обратимся к теории стационарных случайных процессов. Наш сигнал $s(n)$ на сегменте длины N можно рассматривать как стационарный процесс с дискретным спектром, то есть процесс, представляющий собой случайные колебания, имеющий своими составляющими гармонические колебания $\xi_k(n) = e^{i\lambda_k n}\Phi_k$ с частотами $\omega_k = |\lambda_k|, k = -N/2 \dots N/2$, фаза и амплитуда которых являются случайными величинами. Такой процесс имеет вид

$$\xi(n) = \sum_{k=-N/2}^{N/2} e^{i\lambda_k n}\Phi_k$$

где $\Phi_{-N/2} \dots \Phi_{N/2}$ - некореллированные случайные величины с нулевыми средними значениями[3]. В этом случае случайный процесс $\xi(n)$ является *стационарным* и его корреляционная функция $r_k = E\xi(n)\xi(n-k)$ зависит только от k.

Полагая далее наш сигнал *авторегрессионным*, мы подразумеваем существование вектора коэффициентов $(a_1, \dots, a_m)$ для некоторого m, а также последовательности нормированных случайных величин w_n с

[3]Подробнее о теории стационарных случайных процессов см. [6]

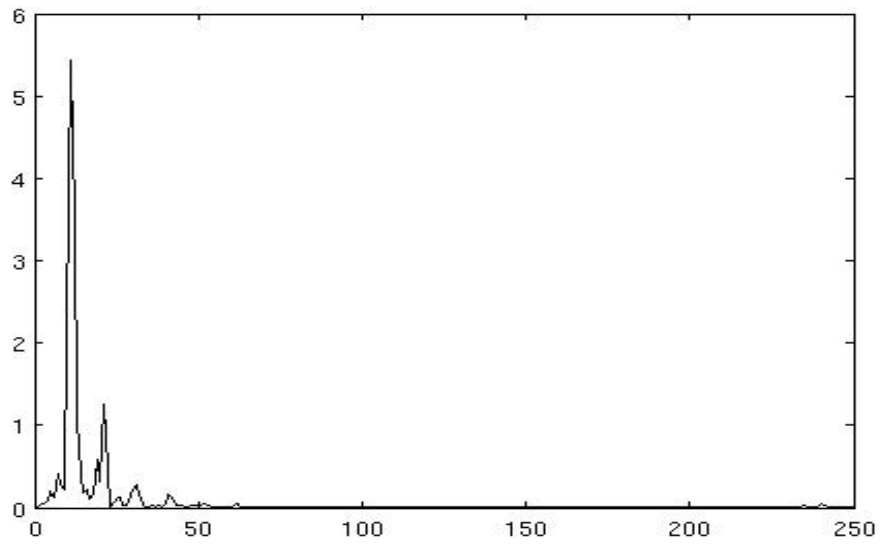


Рис. 4: График модуля дискретного преобразования Фурье на сегмента

нулевым средним и дисперсией σ^2, что

$$s(n) + a_1 s(n-1) + \ldots + a_m s(n-m) = w_n$$

для всех n.

Как известно, авторегрессионный процесс является стационарным процессом со спектральной плотностью

$$S(z) = \frac{\sigma^2}{|a(z)|^2}, a(z) = 1 + a_1 z + \ldots + a_m z^m, |z| = 1$$

В нашем случае спектральная плотность случайного процесса имеет смысл «усреднённой» спектральной огибающей. На рис. 3 представлен график спектральной огибающей для сигнала на некотором сегменте, тогда как на рис 4 представлен график модуля дискретного преобразования Фурье того же сигнала.

В связи с этим мы можем вычислить значения корреляционной функции r_k как

$$r_k = \sum_{n=1}^{N-k} s(n)s(n+k), k = 1 \ldots m$$

Algorithm 1 pitch_multiple_decision(P, N, s(n))

Require: Вычислить значения $\Phi(z)$ и l

$Res = []$ {Вектор приближений для различных соотношений}
for $i_F \in \{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1, \frac{3}{2}, 2, 3, 4\}$ **do**
  $V_i = []$ {вектор значений спектра кандидатов для i_F}
  $\Phi_i = []$ {вектор соответствующих значений Φ}
  for $f \in l$ **do**
    if f делится на i_F без остатка **then**
      $B =$ «колокольчик» для f
      $P = < B, FFT(s(n)) >$
      $V_i = V_i \leftarrow P$ {Добавили элемент P в вектор V_i}
      $\Phi_i = \Phi_i \leftarrow \Phi[l.index(f)]$ {Добавили в вектор Φ_i соответствующий f элемент огибающей Φ}
  $Res[i_F] = \min_\tau ||\Phi_i \tau - V_i||$ {Ищем минимум данного функционала методом наименьших квадратов}
$[m, i_{F*}] = \min_{i_F} Res[i_F]$ {Минимум по всем приближениям и кандидат, на котором он достигается}
return $\frac{P}{i_{F*}}$

а далее воспользоваться уравнениями Юла-Уокера $R\overline{a} = \chi$, где

$$R = \begin{pmatrix} r_0 & r_1 & \dots & r_m \\ r_1 & r_0 & \dots & r_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_m & r_{m-1} & \dots & r_0 \end{pmatrix}, \overline{a} = \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}, \chi = \begin{pmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

При известных значениях корреляционной функции r_k, $0 \leq k \leq m$ коэффициенты $a(z)$ определяются из последних m уравнений, а дисперсия σ^2 - из первого уравнения. Мы находим искомые коэффициенты, решая рекуррентную систему уравнений алгоритмом Левинсона-Дурбина.

3.4.2 Алгоритм устранения ошибки октав

Построив таким образом спектральную огибающую, мы составляем вектор возможных «кандидатов» на роль частоты основного тона и кратных ей. Для этого вектор $< \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1 >$ возможных соотношений определённой частоты к истинной сдвигается по $P/2$ спектральным полосам. Получается вектор l значений, где могут располагаться частоты основного тона и кратные им. Далее во всех точках $z = e^{2\pi i f/P}, f \in l$ считается вектор спектральной огибающей $\Phi(z) = S(z)$.

Предполагается, что возможные соотношения известной частоты основного тона F к истинной могут принимать значения из множества $\{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1, \frac{3}{2}, 2, 3, 4\}$, поэтому далее выбор наилучшего кандидата описывается алгоритмом 1.

Суть алгоритма сводится к следующему. Мы минимизируем невязку между вектором значений спектральной огибающей, вычисленными в точках, соответствующих частотам, кратным возможному кандидату, и значениями истинного спектра в этих же точках. Последние вычисляются по ДПФ исходного сигнала и «колокольчику», восстановленному в данной точке.

Как можно видеть, компоненты f вектора l являются по сути компонентами вектора $< \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1 >$, «растянутого» по всем спектральным полосам. Вектор V_i носит смысл спектра исходного сигнала, вычисленного в точках f. Выражение $\Phi[l.index(f)]$ означает взятие компоненты вектора Φ, сосчитанной в точке $z = e^{2\pi if/P}$. Таким образом векторы Φ_i и V_i имеют одинаковые размерности. На шаге 12 ищется минимум по всем возможным приближениям различных векторов Φ_i векторами же V_i, соответствующими различными значениям i_F соотношения реальной и изначальной частоты. Значение i_{F*}, на котором достигается минимум - и есть искомое соотношение.

Найдя таким образом наилучшее приближение для частоты основного тона, мы ещё раз уточняем его посредством алгоритма, описанного в разделе 3.3.

3.4.3 Случай явного спектрального максимума

Как показади эксперименты, алгоритм, описанный в разделе 3.4.2 может приводить к ошибке в том случае, если изначально определённая частота основного тона была меньше истинной в целое число раз. Это обуславливается излишней гладкостью спектральной огибающей. Другими словами, невязка получается меньше в том случае, когда расставлены «лишние» колокольчики в дополнительных точках, соответствующих частоте основного тона, меньшей истинной.

В случае, когда алгоритм компенсации ошибки октав работает неудовлетворительно, используется следующее простое улучшение. Если на некоторую частоту F в дискретном преобразовании Фурье приходится ярко выраженный энергетический максимум[4], то эта частота выбирается как изначальное приближение. В случае нахождения другого существенного локального максимума проверяется, не приходится ли он на частоту

[4]К примеру, все другие локальные максимумы модуля спектра сигнала составляют 10% от данного.

F', кратную F. Так как спектр здесь всё-таки дискретный, учитывается и тот вариант, когда кратной F является частота $F' + 1$ либо $F' - 1$. В этих случаях F также выбирается в качестве изначального приближения, после чего происходит уточнение значения $\frac{N}{F}$ как периода основного тона[5].

[5]Здесь N, как и прежде, обозначает длину исследуемого сегмента сигнала.

4 Детали реализации

Далее остановимя подробнее на деталях реализации алгоритма, описанного в разделе 3.

4.1 Расчётная часть

Основные вычисления, связанные с определением и уточнением периода основного тона, реализованы посредством пакета прикладных программ *MathWorks Matlab*. Данный программный модуль может быть исполнен на любой целевой платформе, на которой установлена среда *Matlab*. В случае необходимости создания отдельного программного модуля, не зависящего от среды *Matlab*, в наличии имеется компилятор[6], создающий по конкретной *Matlab*-программе программу на языке C либо C++, которая впоследствии может быть скомпилирована и запущена на целевой платформе. В последнем случае для корректной работы также понадобятся специфичные для *Matlab* библиотеки функций, которые могут быть установлены при помощи инсталлятора, поставляющегося непосредственно с *Matlab*.

В результате сборки данного программного модуля создаётся исполняемый файл `notegen`[7], а также запускающий его скрипт `run_notegen[.ext]`[8]. Команда `run_notegen` имеет следующий формат:

run_notegen `matlab_path [OPTIONS] wav_file`

где

`matlab_path`
путь к директории с библиотеками, необходимыми для запуска программы, скомпилированной с помощью `mcc`

`wav_file`
входной файл ф формате WAVE с расширением `.wav`.

`[OPTIONS]`
необязательные опции следующего вида

[6] `mcc` - *Matlab C Compiler*
[7] В данный момент это рабочее название программы
[8] Расширение `ext` исполняемого скрипта специфично для конкретной платформы

`-n`
Не генерировать `NEF`[9]-файл с промежуточным представлением распределения частот основного тона по времени. По умолчанию данный файл генерируется и его имя без расширения совпадает с именем исходного `WAV`-файла. Впоследствии этот файл используется программным модулем, описанным в 4.2 для генерации TEX-документа с нотами.

`-m`
Генерировать `MIDI`-файл. Имя файла без расширения совпадает с именем исходного `WAV`-файла.

`-g`
Вывести на экран график зависимости распознанных нот в зависимости от времени. К примеру, если указать эту опцию для `WAV`-файла, содержащего последовательно проигранные ноты *a*, *b* первой и *c* второй октавы, на экран будут выведен графики вида 5, где верхний представляет собой распределение амплитуд исходного сигнала, а нижний, соответственно, полученные значения нот. За 0 в нижнем представлении принимается нота *a* первой октавы.

4.2 Генерация нотной записи

Для получения нотной записи `NEF`-представление сигнала обрабатывается скриптовой программой на языке *Python*. Он был выбран как наиболее удобный инструмент для работы с текстами, так как на выходе порождается файл в формате TEX с нотным содержимым. Этот файл может быть как скомпилирован отдельно с помощью утилиты MusiXTEX - надстройки над пакетом LATEX, так и вставлен в качестве составляющей части в любой LATEX-документ. Все ноты, представленные далее в этой статье, получены именно таким образом. Утилита MusiXTEX была выбрана благодаря её простоте в использовании и широте возможностей. Подробнее ознакомиться с ней можно в [4].

Уже упоминавшееся `NEF`-представление является по сути текстовым файлом и содержит в себе следующую информацию:

- `bitrate`
  Частота квантования входного `WAV`-файла. К примеру, 22050 и 44100 отсчётов в секунду.

[9]Notes & Energies File

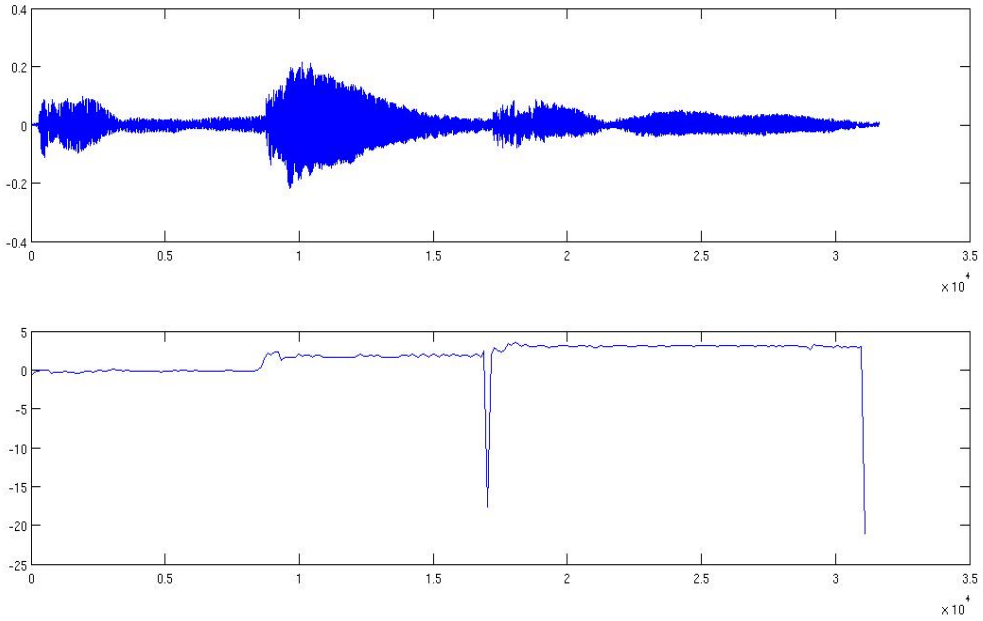


Рис. 5: График зависимости распознанных нот от времени

- `seglength`
  Длина наименьшего сегмента («кванта»), которому приписывается конкретное значение ноты и энергии и громкости. В случае музыкального мы использовали значение 128 отсчётов.

- Набор пар `(n, e)`, где `n` - нота, приписанная данному кванту, а `e` - соответственно, энергия данного кванта.

Сам алгоритм генерации нотной записи заключается в анализе и фильтрации полученного массива данных:

1. Вычисляется абсолютная длина одного кванта (в секундах).

2. Изходя из полученной величины кванты разбиваются на последовательные группы g_i исходя из того факта, что в секунду не может быть исполнено более 20 нот.

3. Те группы g_i, в которых суммарная энергия не превышает 5% от средней энергии по всем таким группа, считаются «пустыми», то есть в итоговой нотной записи они будут представлены знаками пауз.

4. Во всех остальных группах g_i происходит факторизация исполненных нот по следующему критерию: в один класс эквивалентности попадают ноты, отличные на 12, 19 и 24 полутона[10]. Другими словами, происходит дополнительная фильтрация не устранённой ошибки октав. В каждом классе эквивалентности выбирается наиболее «многочисленный» участник, который в итоге представляет весь класс.

5. Среди всех выделенных в группе g_i представителей «октавных» классов выбирается выбирается тот, чей класс обладает наибольшей мощностью. Он и приписывается данной группе как звучащая нота.

6. Производится «склеивание» полученных в группах g_i данных об определённых нотах в постоянные участки. В результате получается список пар вида `(n, e)`, где `n` - нота на временном участке, `t` - продолжительность данного участка. В случае паузы на данном участке в первой компоненте пары стоит символ `'p'`.

7. Согласно установленной изначально продолжительности четвертной ноты по полученному списку пар происходит подбор надлежащих нотных длительностей и формирования текста TEX-документа.

Исполняемый *Python*-скрипт, реализующий данный алгоритм, принимает на входе файл с `NEF`-представлением и порождает на выходе TEX-документ. При наличии на целевой платформе установленных утилит `latex` и `musixtex` происходит компиляция полученного документа d `PDF`-документ. Примеры полученных нотных записей мы приводим в разделе 5.1.

4.3 Генерация `MIDI`-файла

Распределение частот основного тона по времени, полученное в результате основного алгоритма, описанного в разделе 3, дало возможность генерации `MIDI`-файлов.

Для их создания мы использовали бесплатную библиотеку функций для среды *MatLab*, предоставленную Кеном Шаттом (*Ken Shutte*)[11]. Примеры использования функций по генерации `MIDI`-файлов а также построенные диаграммы распределения нот по времени можно найти в приложении А.

[10] Одна октава - 12 полутонов, 19 полутонов соответствуют октаве с квинтой или ошибке по определённой частоте в три раза.

[11] Подробное описание данной библиотеки может быть найдено по адресу `http://www.kenschutte.com/midi/`

Алгоритм генерации входа почти не отличается от аналогичного для нот, описанного в 4.2 с той лишь разницей, что в данном случае нет необходимость «склеивать» полученные кванты времени с одинаковым значением частоты основного тона. Достаточно лишь вычислить «интервалы постоянства» той или иной ноты и её громкость.

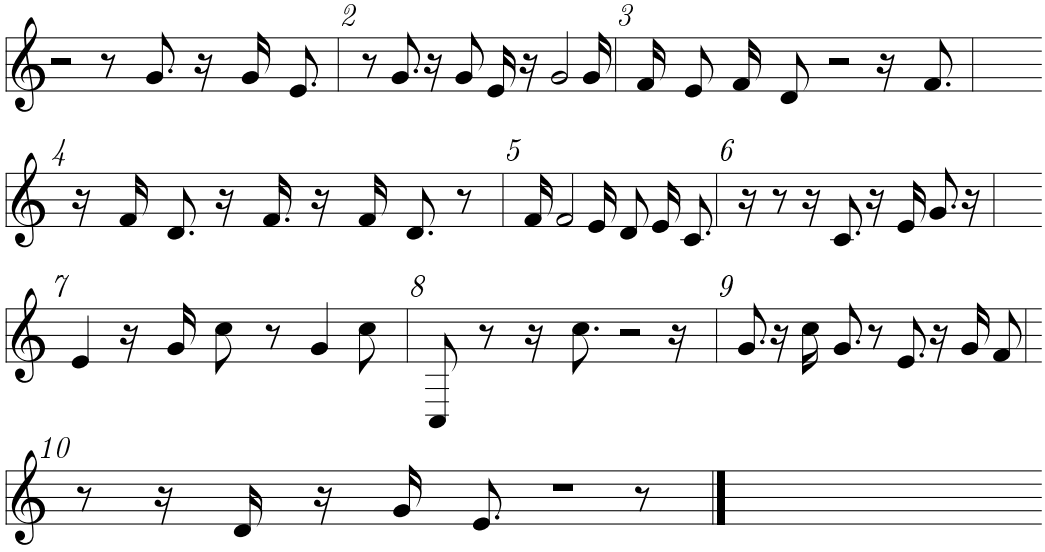
5 Результаты

Получение символьного представления по необработанному музыкальному сигналу является востребованной задачей во многих областях, касающихся обработки звука и классификации музыкальных данных. Представленное в данной работе приложение, созданное «с нуля», даёт возможность легко получать качественную нотную запись одноголосного сигнала, `MIDI`-представление, а также промежуточное `NEF`-представление, которое может быть в дальнейшем использовано для задач сравнения и поиска.

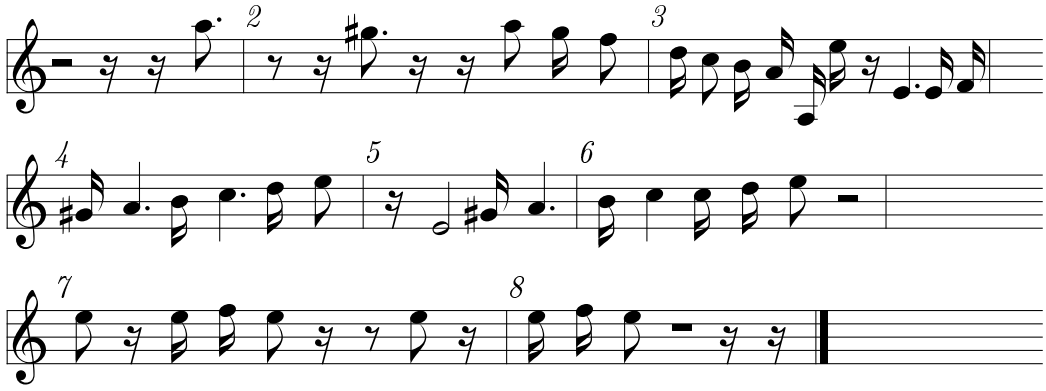
5.1 Полученные ноты

Возможность генерировать по музыкальному сигналу файлы в формате TEX позволила представить в этой статье результат работы программы по генерации нот. В качестве музыкальных сигналов для визуализации были выбраны начало арии Фигаро из оперы «Женитьба Фигаро» В. А. Моцарта, а также начало основного мотива концерта *a-moll* для двух скрипок с оркестром А. Вивальди. Оба музыкальных фрагмента исполнены автором на домашнем фортепиано.

1. В. А. Моцарт, Фрагмент арии Фигаро из оперы «Женитьба Фигаро»



2. А. Вивальди, Концерт *a-moll* для двух скрипок с оркестром



5.2 Получение MIDI-файла

Получаемое в результате анализа промежуточное символьное представление легко транслировать в MIDI-файл. Визуализированные результаты работы этой части программы представлены в приложении А.

Кроме иллюстраций, относящихся к нашей программе там же представлена визуализация результатов работы упомянутых коммерческих приложений, таких как *IntelliScore*, *AKoff Music Composer* и *AmazingMIDI*. На рис. 6-8 заметна не устранённая ошибка октав. Особенно хорошо это заметно на рис. 8. Причина этого, судя по всему, в том, что в этом случае для выделения звучащих голосов использовался исключительно метод поиска спектральных максимумов. Локальные максимумы приходились же как на истинную частоту основного тона, так и на кратные ей.

Другая проблема рассмотренных коммерческих приложений хорошо заметна на рис. 11, на 4 и 5 секундах воспроизведения, где происходит «склеивание» нот, отличающихся на секунду (в данном случае это «e» и «f» первой октавы). Данная ошибка могла быть вызвана проблемами округления полученного значения частоты при переводе в ноты по формуле

$$n = 12 \cdot log_2\left(\frac{f}{440}\right)$$

где n - нота, f - частота основного тона в герцах, а константа 440 - частота ноты «ля» первой октавы фортепиано, а 12 - количество полутонов в октаве. Ошибки могут возникать в том случае, если фортепиано немного расстроено, и ноте «ля» первой октавы соответствует другое значение[12]

[12]В случае фортепиано автора было установлено, что нота «ля» первой октавы звучит на частоте 437 герц.

5.3 Хранилища музыкальных данных

Область применения полученного символьного представления весьма широка. В дальнейшей работе планируется использовать рассмотренный подход выделения нотных последовательностей для выяснения «похожести» фрагментов различных музыкальных произведений. Подобная проблема подробно рассматривается в работе [3]. Правда, в этом случае авторы имеют дело с уже готовыми «чистыми» символьными данными, тогда как в нашем случае получающаяся на выходе последовательность нот и пауз может достаточно сильно отличаться от того, что видел перед собой музыкант, исполняя произведение.

Применимые в данной задаче алгоритмы сходны с теми, которые используются при исправлении ошибок в последовательностях слов. В нашем случае можно выделить несколько видов преобразований, переводящих «чистую» нотную последовательность в то, что получается на выходе нашей программы:

- Растяжение или сжатие по времени. Другими словами, результирующая символьная последовательность $y(t)$ может выражаться через исходную $x(t)$ как

  $$y(t) = x(\alpha(t))$$

  где $\alpha(t)$ - некоторая функция, обладающая положительной производной на всей области определения $x(\cdot)$

- Незначительные добавления новых нот в исходную последовательность, что вызвано погрешностями идентификации. То же, относится и к удалении. коротко звучащих нот. Эти проблемы сходны с задачами, решаемыми современными поисковыми системами при вариантах исправления некорректного словесного запроса.

- *Альтерация.* Нотная последовательность, взятая за образец, может быть записана в другой тональности, нежели анализируемая, хотя с семантической точки зрения они могут быть эквивалентны. В этом случае стоит производить операции сопоставления на множестве нотных фрагментов, факторизованных относительно операции *альтерации* (то есть изменения тональности).

6 Заключение

Результатом данной работы явилось создание программного средства для анализа одноголосного музыкального сигнала и получения по нему символьного представления.

Построенное приложение эффективно устраняет ряд классических ошибок, возникающих при нахождении периода основного тона, таких как погрешности, возникающие при дискретизации сигнала, ошибку октав и т. п.

В работе использованы методы анализа звукового сигнала, разработанные А. Е. Барабановым, а также описанные в работе [2] и использующиеся в создании голосовых вокодеров. Существенно улучшена производительность алгоритма уточнения целого периода основного тона за счёт применения метода Ньютона. Кроме того, улучшен изначально предложенный алгоритм устранения ошибки октав за счёт рассмотрения случая явного спектрального максимума.

Создан ряд программных модулей, позволяющих получать как абстрактное символьное представление исходного музыкального сигнала, так и файлы в формате `MIDI`, а также непосредственно нотную запись в виде `PDF`-документа.

В финальной части работы произведён сравнительный анализ результатов работы созданного приложения и ряда коммерческих средств на примере сравнения получаемых `MIDI`-файлов.

A Визуализация полученных MIDI-файлов

В данном приложении представлены визуализированные представления MIDI-файлов, полученных в результате работы алгоритма, описанного в разделе 4.3. В качестве тестовых музыкальных фрагментов мы взяли те же музыкальные произведения, что были использованы для генерации нотной записи. В представленных ниже визуализированных MIDI-файлах по оси абсцисс откладываются временные отсчёты (в секундах), а по оси ординат - ноты в полутонах в численном выражении. Так, к примеру, ноте «ля» первой октавы соответствует значение 69, ноте «до» второй - 72 и т. д.

A.1 В. А. Моцарт, Фрагмент арии Фигаро из оперы «Женитьба Фигаро»

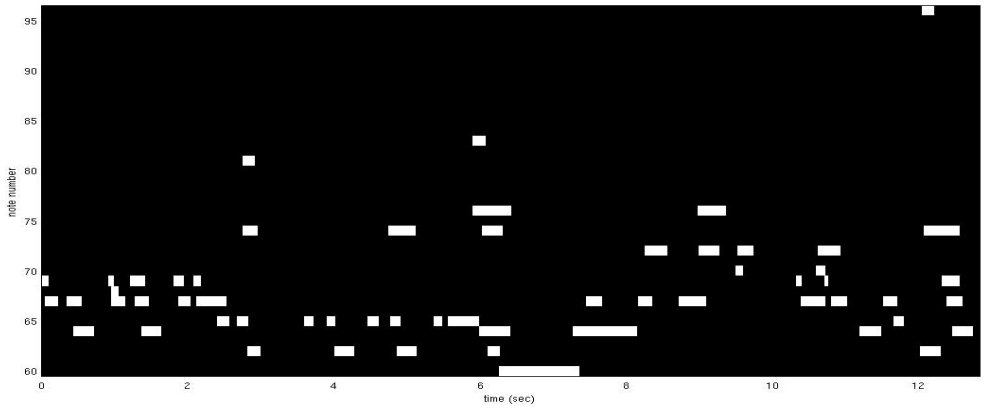


Рис. 6: Ария Фигаро - AKoff Music Composer

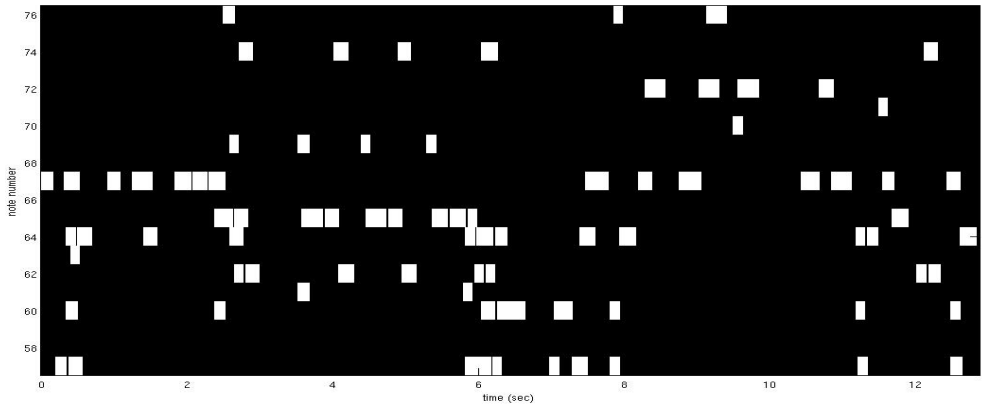


Рис. 7: Ария Фигаро - AmazingMIDI

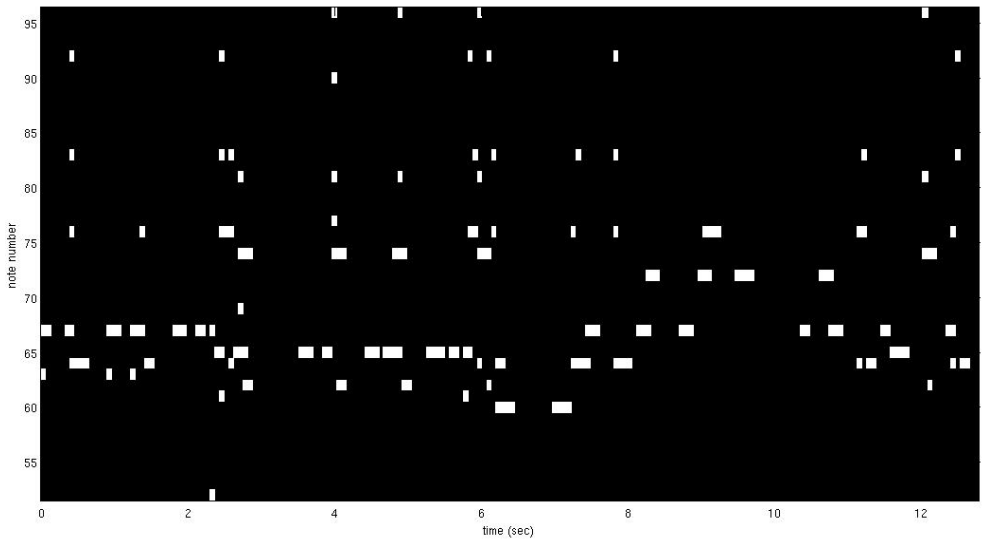


Рис. 8: Ария Фигаро - IntelliScore

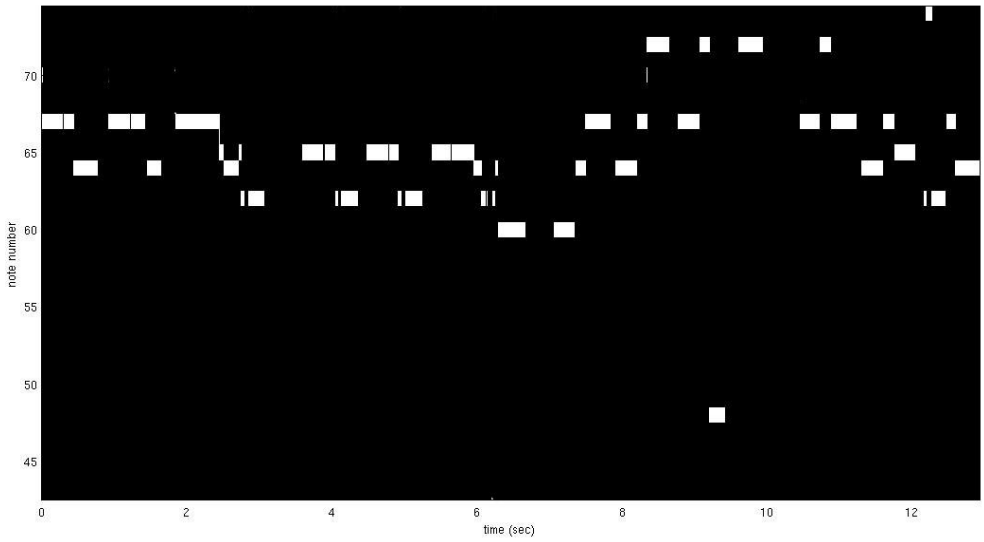


Рис. 9: Ария Фигаро - наше MIDI-представление

A.2 А. Вивальди, Концерт *a-moll* для двух скрипок с оркестром

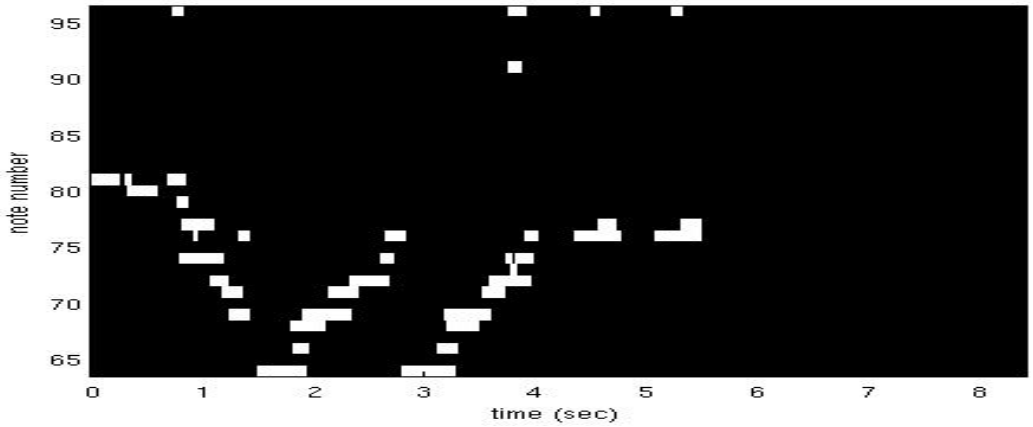


Рис. 10: Концерт a-moll - AKoff Music Composer

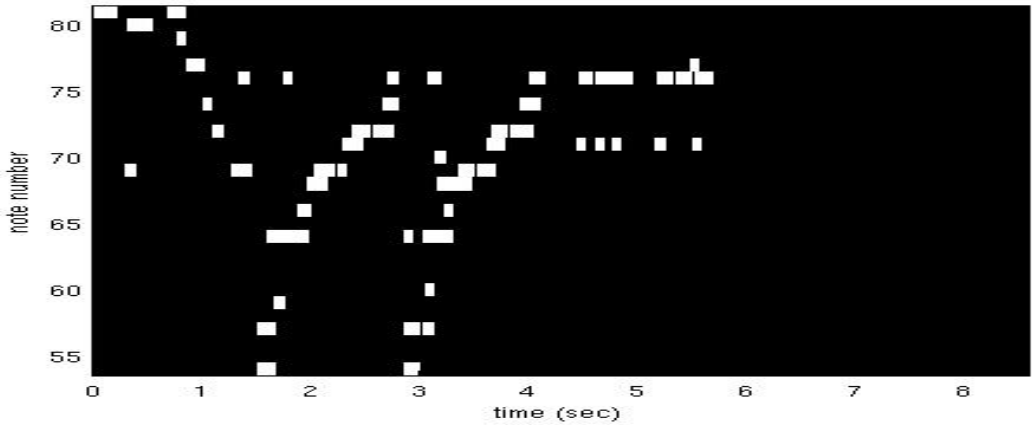


Рис. 11: Концерт a-moll - AmazingMIDI

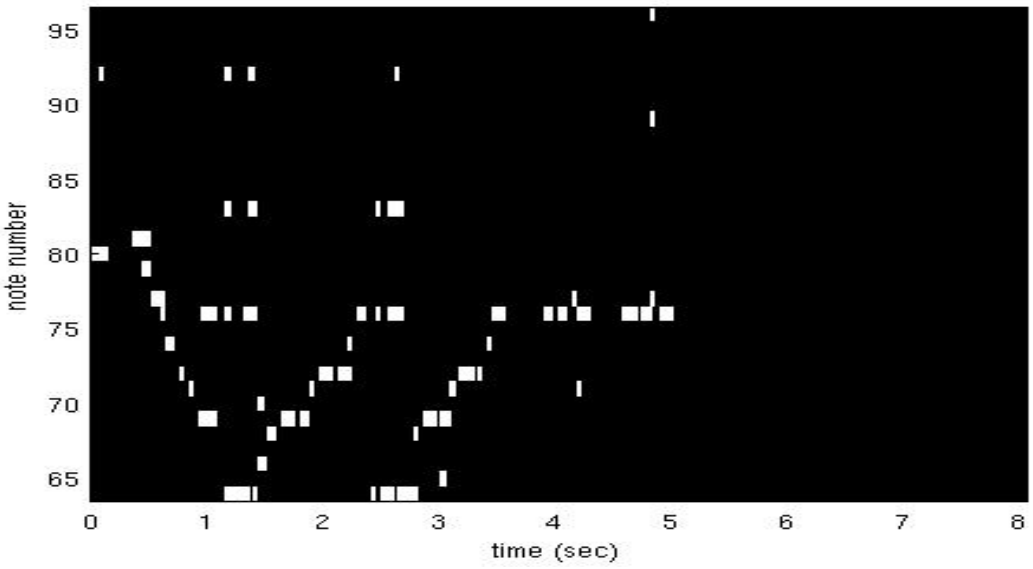


Рис. 12: Концерт a-moll - IntelliScore

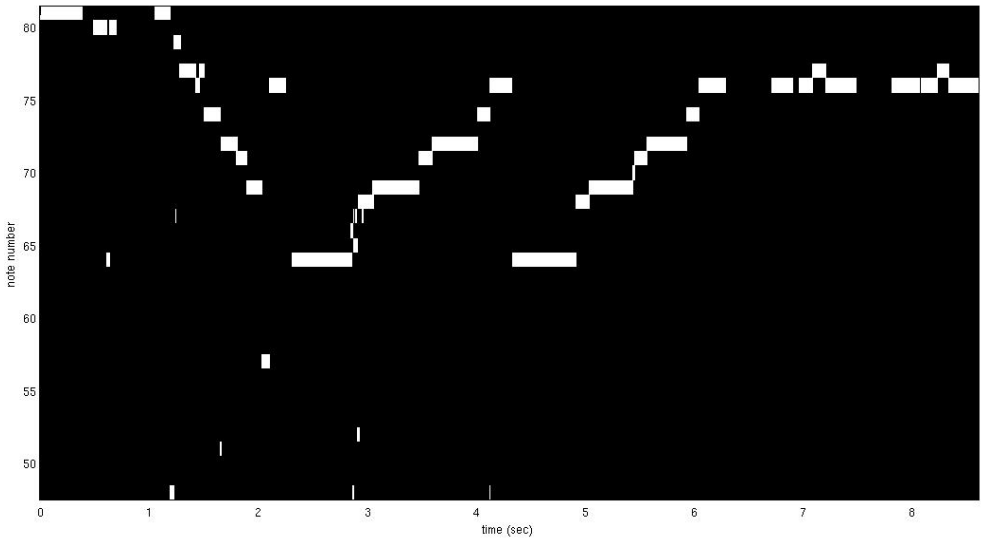


Рис. 13: Концерт a-moll - наше MIDI-представление

Список литературы

[1] Dziubinski M., Kostek B., *High Accuracy And Octave Error Immune Pitch Detection Algorithms*, Archives of Acoustics, 2004

[2] Griffin D.W., Lim J.S. *Multiband excitation vocoder*. IEEE ASSP-36 (8), 1988, pp. 1223-1235.

[3] Lartillot O., *Efficient Extraction of Closed Motivic Patterns in Multi-Dimensional Symbolic Representations of Music*, *IEEE/WIC/ACM International Conference on Web Intelligence*, Compiègne, IEEE Computer Society Press, 2005

[4] Taupin D., Mitchell R., Egler A., *Musixtex. Using TEX to write polyphonic or instrumental music*

[5] Оппенгейм А, Шафер Р., *Цифровая обработка сигналов*, М, «Техносфера» 2006. - 856с.

[6] Розанов Ю. А., *Случайные процессы. Краткий курс*, М, «Наука» 1979. - 184 с.

[7] *MIDI protocol guide*
Available at `http://hinton-instruments.co.uk/reference/midi/protocol/`

[8] Ken Shutte. *MIDI and Matlab*
Available at `http://www.kenschutte.com/midi/`