**Федеральное государственное автономное образовательное учреждение   
высшего образования**

**КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ИНФОРМАЦИОННЫХ СИСТЕМ**

Направление подготовки: 09.03.03 Прикладная информатика

Профиль: Прикладная информатика в экономике

**ДИПЛОМНАЯ РАБОТА**

**Методы "оживления" midi-партий ударных музыкальных инструментов**

**Работа завершена:**

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Студент группы 11-307 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шайхутдинов А. Л.

**Работа допущена к защите:**

Научный руководитель

(преподаватель/куратор лаборатории)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кугуракова В. В.

Директор Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хасьянов А. Ф.

Казань – 2017 г.

**Содержание**

[Введение 3](#_Toc484429247)

[1. Работа с MIDI 6](#_Toc484429248)

[2. "Оживление" MIDI-партий ударной установки 11](#_Toc484429249)

[3. Байесовский метод 19](#_Toc484429250)

[3.1. Байесовский подход к оживлению 20](#_Toc484429251)

[3.2. Алгоритм 21](#_Toc484429252)

[3.3. Субъективная оценка 24](#_Toc484429253)

[4. Собственная модификация 27](#_Toc484429254)

[4.1. Создание алгоритма изменения громкости нот 28](#_Toc484429255)

[4.2. Создание алгоритма сдвигов нот 32](#_Toc484429256)

[4.3. Создание примеров 36](#_Toc484429257)

[4.4. Сравнение результатов 36](#_Toc484429258)

[5. Обсуждение и вывод 39](#_Toc484429259)

[Заключение 41](#_Toc484429260)

[Список использованных источников 42](#_Toc484429261)

[Приложение 43](#_Toc484429262)

## Введение

В программируемой электронной музыке музыкальные партии обычно записываются с использованием секвенсоров и пианоролла. В этих системах музыкальный ритм сохраняется при использовании сетки с равными делениями. Поскольку ноты располагаются по сетке, они квантуются по выбранным интервалам. По ощущениям квантованные ноты звучат слишком ровно и не звучат так же, как если бы на музыкальном инструменте играл человек. Для того, чтобы эмулировать воспринимаемую естественность стиля человеческой игры, к последовательности может быть применен гуманизатор.

В статье ученых из Бирмингема [1] дается такое определение гуманизации: гуманизация - это процесс, который влияет на различные параметры сигнала для того, чтобы создать менее роботизированный результат. В партиях ударных инструментов она в первую очередь влияет на момент возникновения ноты по отношению к метрономной сетке и на амплитуду или громкость каждой соответствующей ноты. В большинстве существующих систем гуманизации используются гауссовские и равномерные генераторы случайных значений.

Хотя эти методы генерируют переменные, которые увеличивают случайность музыкальных событий, вопрос о том, до какой степени они увеличивает воспринимаемую естественность последовательности, открыт для обсуждения.

В качестве примера, для сравнения, на рисунке 1 изображены отклонения от метрономной сетки нот партии, сыгранной человеком на клавиатуре компьютера и той же партии, только квантованной, гуманизированной гауссовым методом. По рисунку можно сделать вывод, что человеческая партия заметно отличается от искусственной, с применением существующих технологий.

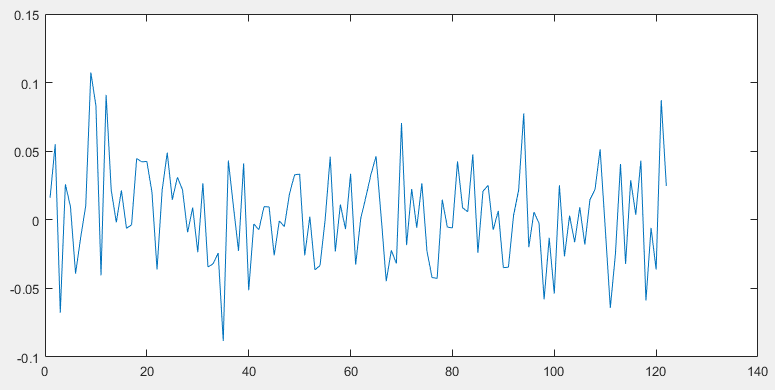
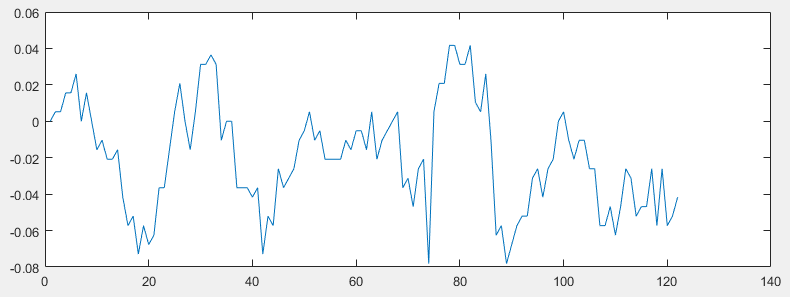


Рисунок 1. Отклонения нот от метрономной сетки (по вертикали) и количество сыгранных нот (по горизонтали) одинаковых партий, сыгранной человеком (выше) и гуманизированной гауссовым методом (ниже)

В отличие от электронных и танцевальных жанров в музыке, в которых звучит безупречно ровный бит драм-машины, в поп- и рок-музыке есть требования к динамике, и слушатели ожидают услышать более естественный звук барабанов с нюансировкой живого исполнителя.

В статье [2] рассказывается о нюансах человеческой игры и о программировании партий ударных музыкальных инструментов. В процессе написания MIDI-партий при создании цифрового музыкального контента важной проблемой часто является синтетичность и неестественность звучания, а основной сложностью — оживление партий, придание партиям «человечности», их гуманизация. Это относится и к барабанным партиям. И не только из-за трудности имитации средствами синтезаторов исэмплеровнастоящего,живого звука, но и по причине более важной — даже самым «похожим на живые» звукам необходимо придать ощущение, что они играются реальным, живым барабанщиком.

Цель данной дипломной работы заключается в создании алгоритма, приближающего midi-партии ударных музыкальных инструментов, исполняемых компьютером к человеческому исполнению.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Изучение средств обработки midi-партий.
2. Изучение методов оживления midi-партий ударных музыкальных инструментов.
3. Оценка и сравнение результативности методов.

В первую очередь рассмотрим, как выглядит классическая барабанная установка (рис. 2). Основные её составляющие – тарелки и барабаны.



Рисунок 2. Барабанная установка | 1. Тарелки | 2. Напольный том-том | 3. Том-том | 4. Бас-барабан | 5. Малый барабан | 6. Хай-хэт

**Программирование ударных музыкальных инструментов** подразумевает наличие некоторых базовых знаний работы с MIDI.

# Работа с MIDI

По определению из Википедии [3], MIDI (Musical Instrument Digital Interface — цифровой интерфейс музыкальных инструментов) — стандарт цифровой звукозаписи, формат обмена данными между электронными музыкальными инструментами.

Интерфейс позволяет единообразно кодировать в цифровой форме такие данные как нажатие клавиш, настройку громкости и других акустических параметров, выбор тембра, темпа, тональности и др., с точной привязкой во времени. В системе кодировок присутствует множество свободных команд, которые производители, программисты и пользователи могут использовать по своему усмотрению. Поэтому интерфейс MIDI позволяет, помимо исполнения музыки, синхронизировать управление другим оборудованием, например, осветительным, пиротехническим и т. п.

В отличие от других форматов это 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. Удобство формата MIDI как формата представления данных позволяет реализовывать устройства, производящие автоматическую аранжировку по заданным аккордам, а также приложения 3D-визуализации звука. Кроме того, такие файлы, как правило, имеют на несколько порядков меньший размер, чем оцифрованный звук сравнимого качества.

Стандартный MIDI-файл (SMF — Standard MIDI File) — это специально разработанный формат файлов, предназначенный для хранения данных, записываемых и/или исполняемых секвенсором, секвенсор может быть как программой для компьютера, так и аппаратно выполненным модулем.

В этом формате хранятся стандартные MIDI-сообщения (то есть статус-байты и соответствующие им байты данных), а также временные метки или маркеры для каждого сообщения (то есть последовательности байтов, указывающие, какое количество условных единиц времени (импульсов, тиков) необходимо подождать перед тем, как исполнить следующее событие MIDI). Этот формат позволяет сохранять информацию о темпе, временном разрешении, выраженном в количестве тиков на одну четвертную длительность, обозначения размера, информацию о музыкальных ключах, а также хранить названия треков и паттернов. Формат предусматривает возможность сохранения в одном файле нескольких паттернов и треков таким образом, что программы-приложения могут выбирать из всего набора хранимой информации ту, которая будет понятна данному приложению.

Как правило, трек представляет собой аналог музыкальной партии, например, партии трубы. Аналогом паттерна может служить весь набор партий, взятых вместе, например: совокупность партий трубы, ударных, фортепиано и т. д., которые используются в данном произведении или его части и исполняются одновременно.

Процесс передачи MIDI-сообщения может осуществляться в реальном времени (во время исполнения или воспроизведения музыки), но может быть и разорванным во времени. В этом случае MIDI-сообщение записывается в виде файла в память компьютера, а потом считывается устройством-получателем. В данной работе обрабатывается заранее подготовленный файл.

Дополнительная информация о MIDI взята из источника [4]. MIDI является протоколом последовательной передачи данных между главным и подчиненным устройством. Главное устройство генерирует сообщения и отправляет их подчиненному устройству, который выполняет полученные команды. Последовательный — значит информация передается по одному биту. Отсюда следует невозможность передачи нескольких сообщений одновременно.

Протокол состоит из трех частей: спецификация формата данных, аппаратная спецификация интерфейса и спецификация хранения данных.

MIDI-сообщение — это управляющая команда, использующая протокол MIDI.

MIDI-сообщения делятся на два типа: сообщения канала (channel messages) и системные сообщения (system messages). Первые управляют звукообразованием, а вторые выполняют служебные функции, например, синхронизация.

Сообщение обычно состоит из двух или трех байт. Первый байт называется статус байтом. В нем задается тип сообщения и номер канала, к которому оно относится. Все последующие байты называются байтами данных. Статус-байт всегда начинается с единицы, а байт-данных с нуля — таким образом система их различает. Получается, что для MIDI информации остается только 7 бит, с помощью которых можно закодировать целые числа от 0 до 127. Отсюда и берется известное ограничение на количество нот и значения контроллеров.

Информации о типе сообщений отводится всего 3 бита, в которых можно закодировать только 8 чисел. 7 из них отведены под наиболее часто используемые команды, а последнее используется для системных сообщений. Когда передается системное сообщение, последние 4 бита статус байта (в которых обычно передается номер канала) определяют тип системного сообщения.

Большая часть информации, генерируемой при игре музыкантом на устройстве с поддержкой MIDI, относится к типу событий Channel Voice. В источнике [5] указано, что Channel Voice включает в себя 7 видов голосовых событий:

* Note-On — активация ноты, начало ее звучания;
* Note-Off — деактивация ноты, окончание ее звучания;
* Monophonic (Channel) Pressure/Aftertouch — параметры силы нажатия клавиши или силы извлечения монофонического звука (эффект послекасания);
* Polyphonic (Key) Pressure/Aftertouch — параметры силы извлечения полифонического звука (эффект послекасания);
* PitchBend — изменение высоты звука;
* Program Change — изменение программы;
* Control Change/Continuous Controller — события потокового управления (127 штук), участвующие в управлении извлеченным звуком и обозначаемые при помощи сокращения CC.

Удобным средством анализа и обработки MIDI-партий является набор инструментов пакета MATLAB — MIDI Toolbox.

MATLAB (сокращение от англ. «Matrix Laboratory») — пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете. Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования. Для MATLAB имеется возможность создавать специальные наборы инструментов (англ. toolbox), расширяющие функциональность. Наборы инструментов представляют собой коллекции функций, написанных на языке MATLAB для решения определённого класса задач.

Набор инструментов MIDI Toolbox, разработанный финскими учеными из университета Ювяскюля Томасом Эуролой и Петри Тойваненом, является самым мощным набором инструментов для статистического анализа, визуализации, генерации MIDI-файлов.

В статье [6] показано как использовать данный набор инструментов. Считывание MIDI-файла в программу происходит посредством функции readmidi, например, так: nmat = readmidi('laksin.mid'). После этого файл представляется в виде так называемой нотной матрицы со следующими столбцами: первая колонка отображает начало ноты исходя из доли, вторая — продолжительность ноты в долях, третья — канал midi, в котором звучит нота, четвертая — высоту ноты, пятая — силу нажатия, а шестая и седьмая — то же, что и первая и вторая, только в секундной системе отсчета.

Озвучивание партий производилось в цифровой звуковой рабочей станции FL Studio 12.

В качестве синтезатора, для воспроизведения midi-партий ударных, был использован плагин EZdrummer 2 с библиотекой Rock (рис. 3).



Рисунок 3. Плагин EZdrummer 2 с библиотекой Rock

В плагине EZdrummer 2 есть готовые образцы midi-партий, записанные на электронных барабанных установках профессиональными барабанщиками паттерны (рис. 4), грувы, которые можно использовать для создания своего цифрового музыкального контента, собирая по частям для различных разделов музыкального произведения.



Рисунок 4. Паттерны

# "Оживление" MIDI-партий ударной установки

Для написания партий барабанов существуют некоторые советы, указанные в статьях [7] и [8], следуя которым можно достичь наиболее эмоциональной и живо звучащей партии.

Первым шагом к оживлению является придание динамики барабанной партии. Одна из самых грубых ошибок, встречающихся у музыкантов, программирующих барабаны, но не являющихся барабанщиками, заключается в выставлении всем MIDI-нотам максимальной или одинаковой громкости (параметр Velocity). При использовании хай-хэта это особенно заметно. Необходимо настроить громкость каждого удара. Можно, например, снизить громкость всех звуков хай-хэта, которые не являются сильной долей, что позволяет значительно оживить игру виртуального барабанщика.

При написании партии барабанов важно учесть, что барабанщик имеет только две руки. Таким образом, извлечение с помощью рук более двух звуков одновременно звучит неестественно и так же является грубой ошибкой. Подобная проблема касается и барабанной дроби. Во время барабанной дроби, при большой ее скорости, барабанщик не способен успевать извлекать с помощью рук одновременно более одного звука.

Также для оживления партий хорошо бывает использовать так называемые ноты-призраки, которые звучат значительно тише. Комбинация разных ударов по рабочему барабану и нот-призраков придает драйва партии. Нужно помнить, что ни один живой барабанщик не играет статично. В живых партиях всегда присутствуют интересные детали, которые не очень заметны на первый взгляд, но значительно украшают партии. Конечно, если речь идет о партии для песен в стиле рока 1980-х, где все очень статично или же о иных экстремальных стилях музыки, статичность рабочего может возрастать.

Удары по крэшу в композициях обычно используются для выделения ее отдельных частей. Крэш всегда звучит немного изолированно от других инструментов, поэтому реальный барабанщик вряд ли станет использовать крэш сам по себе в отрыве от других частей ударной установки. Следовательно, виртуальный барабанщик также не должен допускать этого. В большинстве случаев крэш используется только вместе с бас-бочкой и ничем другим.

Реальные барабанщики ускоряются и замедляются - иногда преднамеренно, иногда нет - и эти изменения темпа могут придать партии более натуральное звучание.

Много реальных ударных стилей фактически зависят от логичных отклонений в пределах теоретически точного ритма. Иногда это весьма очевидно, как в случае жёсткого синкопирования или ‘свинга’, которое налагает ощущение триолей на ритм с четырьмя долями, но это может быть и намного более тонким элементом. Например, игра с опережением доли (early snare), особенно на первых и третьих долях такта с четырьмя ударами, является одним из приёмов, способным добавить больше «задора» ритму, что характерно для большинства диско, поп и кантри барабанщиков. В таких жанрах как блюз, наоборот: иногда барабанщики преднамеренно задерживают удары, чтобы создать чувство замедления (laid-back).

В музыкальных произведениях могут быть очень ощутимы небольшие изменения темпа, например, такие как rallentando (замедление к концу песни) и переходы между медленными и быстрыми частями композиции. Другие изменения, однако, являются более тонкими: для барабанщиков весьма характерно ускорять темп при переходе к припеву. В некоторых классических записях встречается постепенное увеличение темпа в некоторых местах, или даже, на протяжении всей песни.

При прослушивании различных записей опытный слух обязательно отмечает характер звучания любого инструмента. При этом значительное внимание уделяется тому, что принято называть живой игрой. Исполнение любой партии реальным барабанщиком (играет ли он на акустических барабанах, или на электронных) всегда отличимо от «игры» автоматом (например, драм-машиной). В статье [2] указаны причины того, что даже от самой ровной игры живым барабанщиком не возникает ощущения механистичности.

– Каждый последующий удар даже в самых маленьких нюансах отличается от предыдущего.

– Человек - не машина, неровности в игре могут проявляться из-за простой «физиологии».

– Извлекаемые из инструмента звуки отличаются высотой тона и тембром в зависимости от силы удара.

Проблема механистичности звучания партий MIDI-барабанов возникает в первую очередь из-за того, что в данной MIDI-партии отсутствуют нюансы, либо они слабо выражены, либо аранжировщик не имеет полного понимания процесса игры на ударных инструментах. Нужно учитывать и тот факт, что по умолчанию у электронных барабанов отсутствует ярко выраженная нюансировка, и они не реагируют ни на что, кроме контроллера Velocity, отвечающего в партии ударных только за «громкость» каждого конкретного удара. Поэтому, при определении характера аранжировки, первым делом нужно решить, какие средства будут применяться.

Для достижения цели у аранжировщика должен иметься сложный инструмент — синтезатор, сэмплер, ромплер с поддержкой многослойности сэмплов: функции, которая в зависимости от силы удара по MIDI-клавише (значения параметра Velocity) способна вызывать различные сэмплы — высокие по тону и яркие по тембру при сильном ударе, более матовые и низкие при более слабом (высота тона, хотя бы в долях цента, изменчива в первую очередь у барабанов, для тарелок характерна тембровая окраска). Но и этого недостаточно.

Даже очень опытный музыкант вряд ли способен при каждом новом ударе «попадать» в одну и ту же точку малого барабана с одинаковой силой. Удары правой и левой руками так же обязательно будут отличаться.  Из этих фактов вытекает отмеченное правило — каждый последующий удар должен отличаться от предыдущего.

Для решения этой проблемы аранжировщику придется искать вариант с применением инструмента, в алгоритм которого вложена функция рандомизации (от англ. «randomize», «randomization»), суть которой заключается в том, что программа в режиме реального времени (либо при игре на MIDI-клавиатуре, либо при воспроизведении уже записанной MIDI-партии) случайным образом «подставляет» к MIDI-нотам сэмплы разного характера звучания. Например, после удара в самый центр мембраны малого барабана может последовать удар, немного смещенный чуть влево или чуть вправо от центра. Получится, что созданная партия не будет иметь повторов в звуковых нюансах. Такой «разнобой», плюс многослойность и студийное качество сэмплов являются одними из самых важных моментов в имитации игры живого барабанщика.

В данном исследовании обсуждаются некоторые недостатки гауссовой гуманизации и рассматриваются вопросы о том, каким образом модели артикуляции, проявляемые ударниками можно эмулировать с помощью вероятностной модели. В исследовании [9] для получения априорного распределения вероятностей и функции правдоподобия были проанализированы партии профессиональных барабанщиков, чтобы создать ряд выборочных распределений. Затем они используются для изменений амплитуд и сдвигов нот квантованной последовательности с использованием рекурсивного байесовского метода. И, наконец, с помощью парного прослушивания тестов сравнивается результативность моделей. Продемонстрировано, что вероятностные модели работают лучше, чем мгновенные гауссовские модели, после оценки рок-ритма в темпе 120 ударов в минуту.

В работе [1] продемонстрировано, что основные временные структуры, которые можно наблюдать в последовательности ударов, часто тесно связаны с длиной музыкальных фраз. Это показано на рисунке 5. При этом, структурные закономерности наблюдаются в местах возникновения и амплитудах соответствующих нот музыкального произведения. Если из последовательности ударов удалить эти неровности с помощью квантования, то это может привести к неестественности, созданию ритма более синтетического звучания.

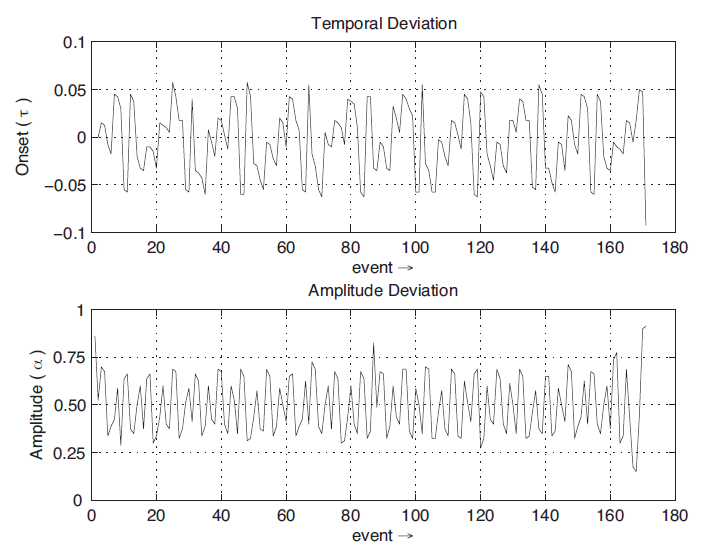


Рисунок 5. Отклонения нот от метрономной сетки (выше) и значений амплитуд нот (ниже) партии профессионального ударника, играющего рок-ритм в размере 4/4

При производстве цифровой музыки часто желательно эмулировать качества реальных барабанщиков. Этот процесс позволяет компьютерным музыкантам создавать реалистичные последовательности с использованием синтезаторов и семплеров, без присутствия ударника. В этих ситуациях, процесс гуманизации может быть использован, чтобы придать выражение и артикуляцию, которые проявляются ударниками на квантованных последовательностях запрограммированных событий. Этот процесс обычно делается в цифровой звуковой рабочей станции (digital audio workstation, DAW) и включает в себя изменение места возникновения и амплитуды нот по отношению к неподвижной метрономной сетке.

В исследовании [9] определяются моменты возникновения нот (*tn*), как отклонение от метронома цифровой звуковой рабочей станции (*mn*), где *tmin* соответствует точке, лежащей в промежутке между *mn* и *mn-1*, а *tmax* соответствует точке, лежащей в промежутке между *mn* и *mn+1*, нормализованное к -1 < 0 < +1. В то время как амплитудные (*an*) параметры нормализованы к 0 < *a* < 1, где 1 - значение максимальной амплитуды, и представляют абсолютную амплитуду *n*-го события.

В гауссовой модели, каждый из параметров модулируется независимо друг от друга, используя распределение, определенное в уравнении (1), где среднее (*μ*) и стандартное отклонение (σ) часто назначается эмпирически пользователем через интерфейс. Для момента возникновения ноты (*х = t*), *μ* устанавливается в '0' и представляет точку, в которой *tn* соответствует *mn*, тогда как σ часто параметрическая, и представляет собой количество «изменчивости» в последовательности. Точно так же для изменения амплитуды (*х = a*), *μ* установлен на произвольное значение средней громкости, а σ представляет изменчивость в динамическом диапазоне.

Хотя этот метод модуляции параметров увеличивает воспринимаемую хаотичность последовательности, он не обязательно увеличивает количество человекоподобного выражения или артикуляции, содержащиеся в последовательности. Так как распределения в гауссовой модели присваиваются мгновенно, предыдущие события не влияют на текущие события. Это значит, что при гауссовом распределении с *μ* = 0 вероятности получения одного и того же значения слухового временного интервала различения с опережением и задержкой будут равны, т.е. *P (-tn+1) = P (tn+1)*. В действительности это маловероятно. На рис. 3 видно, что и *t*, и *a* демонстрируют структуру, по которой видно, что вероятность n-го события условно зависит от n-1-го. Например, ударники часто слегка задерживают или излишне подчеркивают событие на основе *t* и *a* предшествующих событий, чтобы добавить в партию определенные смысловые признаки, такие как плотность или разреженность.

Также модели модуляции параметров могут часто наблюдаться локально в компонентах ударной установки. Например, если на протяжении музыкального произведения последовательные удары по бочке большей силы, чем по малому барабану, два компонента могут быть смоделированы, с использованием независимых распределений, с дискретными параметрами *μ* и σ из исследования [1]. Эта форма модуляции распространена в человеческой игре, однако она опущена в моделях с глобальным распределением. Кроме того, нюансы, которые артикулированы определенными ударниками, вряд ли могут возникать, когда используется глобальное гауссово распределение. Исследование может быть расширено до жанров или ударных стилей, все из которых имеют особенности, которые теряются при гауссовой гуманизации. Эта разница между человеком и гауссовыми распределениями выделена на рисунке 6. Последовательности, которые сгенерированы с использованием каждого из распределений, вряд ли будут иметь подобные атрибуты.

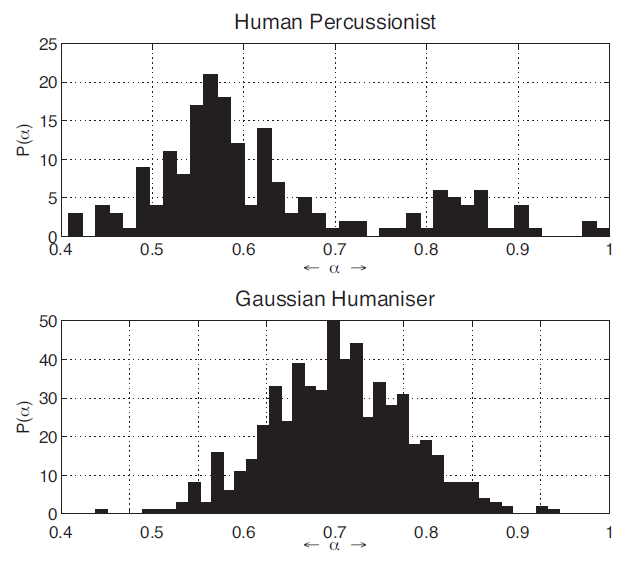


Рисунок 6. Сравнение между человеческим и гауссовым распределениями амплитуд ударной последовательности, измеренной в течение 12 тактов

Для решения этих проблем О'Салливан и Боланд в исследовании [10] рассмотрели вопрос об использовании нечеткой логики для того, чтобы менять силу удара. Чтобы сделать это, данные записанной партии ударных используются для изменения квантованных последовательностей. По аналогии, вероятностные модели могут быть использованы для взвешивания псевдослучайных модуляторов, управляющих *t* и *a*. В исследовании [1] это делается с использованием скрытой модели Маркова (СММ). Для реализации этого, эмпирические распределения взяты из набора данных записанных барабанщиков и условно присваиваются на основе предыдущих событий. И *t,* и *a* рассматриваются как взаимосвязанные состояния в модели, где вероятность события (*bn*), выбирается из перехода между состояниями матрицы, на основе предыдущих N событий.

bn = P(bn|bn-1, bn-2,…, bn-N ) (2)

# Байесовский метод

В исследовании [9] рассматривается использование рекурсивного байесовского метода с целью увеличения параметров событий. Этот метод позволяет картографировать значения параметров вероятностных распределений и решает проблемы, вызванные условной независимостью и мгновенным распределением переменной, наблюдающиеся в гауссовой модели. Используется набор данных классически обученных барабанщиков для выведения эмпирических кумулятивных функций распределения, а затем оценивается естественность модели с использованием парных прослушиваний тестов.

В теории вероятностей байесовский логический вывод может быть использован для формирования распределения апостериорной вероятности, основанной на первоначальной гипотезе и некоторых наблюдаемых событиях в отношении указанной гипотезы. Формально правило Байеса (3), определяет метод, который параметрически моделируется априорным распределением вероятности P(A).

Это распределение представляет неопределенность в отношении данной гипотезы "А" до введения любого другого события и подправляет вероятность гипотезы, данную новым свидетельством. Параметр правдоподобия в модели P(B|A) является условной вероятностью новых событий, учитывая предварительные знания. Р(В) относится к распределению, характеризующему признаки при всех вариантах А. Это вытекает из суммы и результата правила, показанного в уравнении 4.

P(B) = P(B|A)P(A) + P(B|¬A)P(¬A) (4)

Такое распределение может быть оценено путем взятия интеграла по Р(А)Р(B|A), т.е. P(B) = ∫P(B|A)P(A)dA. В рекурсивных численных моделях алгоритм работает итеративно, обновляя априорное распределение апостериорным по завершении каждого цикла.

В любом случае рекурсия используется для обновления первоначальной гипотезы и принимает обоснованные решения в отношении конкретного события, учитывая ряд наблюдений.

# Байесовский подход к оживлению

В исследовании [9] использовались эмпирические данные для того, чтобы создать естественно звучащую артикуляцию, используя байесовскую модель. Для этого сначала был записан набор данных ударников, играющих предопределенные последовательности на электронной барабанной установке. Моменты возникновения нот и соответствующие амплитуды нот затем были извлечены из игры с помощью миди, т.е. электронной барабанной установки. Значения амплитуд были нормализованы, чтобы представить параметры амплитуды *a*, а моменты возникновения нот, вычисленные из массива, представляющего компьютерную метрономную дорожку, представляют различные параметры *t*. Были взяты образцы исполнений шести классически обученных ударников, играющих в темпе 120 ударов в минуту в размерности 4/4. Каждому из ударников было предложено играть в традиционном рок стиле, 32 такта предопределенной последовательности одной композиции.

Эти параметры затем использовались для создания эмпирических распределений, из которых были выбраны случайные переменные основываясь на модели, описанной в разделе 3.2. Для того, чтобы проверить модель, брался новый квантованный шаблон (показано на рисунке 7) и оживлялся с использованием байесовского метода. Затем проводилось прослушивание парных тестов с использованием ударных последовательностей, взятых из четырех групп: квантованные последовательности без применения гуманизации, последовательности с применением гауссовой гуманизации, последовательности, которые очеловечены с помощью эмпирического байесовского метода и последовательности, которые оживлены с использованием СММ, описанной в [1]. Каждая из последовательностей длиной 16 тактов проигрывалась с тем же тактовым размером и темпом как в исходном наборе данных.

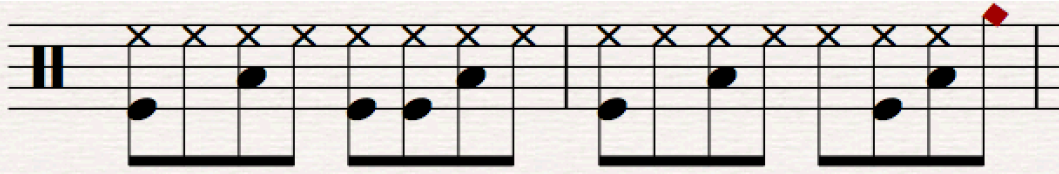


Рисунок 7. Баллы последовательности используются для того, чтобы субъективно оценить естественность Байесовской модели

# Алгоритм

В статье [9] рассказывается, что используя правило Байеса, приведенное в формуле (3), можно одновременно обновлять распределения для моментов возникновения нот и значений амплитуды с целью включения эмпирической вероятности в процесс гуманизации. Оцениваются моменты возникновения нот и значения амплитуды независимо друг от друга, на основе предыдущих событий в соответствующей группе. Перед тем как этот процесс осуществляется, каждая из групп параметров (*a* и *t*) сегментирована в группы компонентов, представляющие диапазоны значений миди-ноты. Эти группы помечены тегами, представляющими составные части: большой барабан, малый барабан, хэт, тарелки и другие. Эти группы определены эмпирически, основаны на ударных последовательностях и комплектах в наборе данных. Этот этап сегментации делает возможным сохранение локальной амплитуды и диапазонов времени возникновения нот. *a* и *t* модулируются с использованием одного и того же метода, однако значения параметров различны. Амплитудные диапазоны 0 < *a* < 1 (это нормализованное абсолютное измерение силы удара), в то время как моменты возникновения нот в диапазоне -1 < *t* < 1, (измеряется как относительные отклонения от неподвижной сетки).

Априорное распределение Р(А)рассматривается как эмпирическая кумулятивная функция распределения, основанная на группах компонентов из набора данных. Каждая из локальных групп компонентов имеет дискретное распределение, которое представляет вероятность всех отклонений в заданном диапазоне. Эти распределения затем используются для выбора нормализованных случайных значений параметров переменных между минимумом и максимумом.

Функция правдоподобия Р(Bi|A) рассматривается как динамическое вычисление способности барабанщика контролировать последовательность, описанную распределением вероятности Р(А) и представлена в виде измерения согласованности через серию N событий. Это может интерпретироваться как вероятность текущего параметра, основанного на N ранее наблюдаемых параметрах событий.

Чтобы вычислить эту функцию, нормализованные распределения были поделены на пять равноотстоящих диапазонов и использовались для квантизации дельта-измерений. Каждый раз, когда событие (*xi-1*) попадает в пределы диапазона, распределение вероятности описывает вероятность следующих событий (*xi*), попадающих в каждый из пяти диапазонов. Для того чтобы включить события, произошедшие раньше во временной последовательности, с затухающим уровнем влияния на текущее событие, для каждой итерации *xi-n*, где 0 < n < N, повторяется вычисление функции правдоподобия, которое дается в уравнении 5.

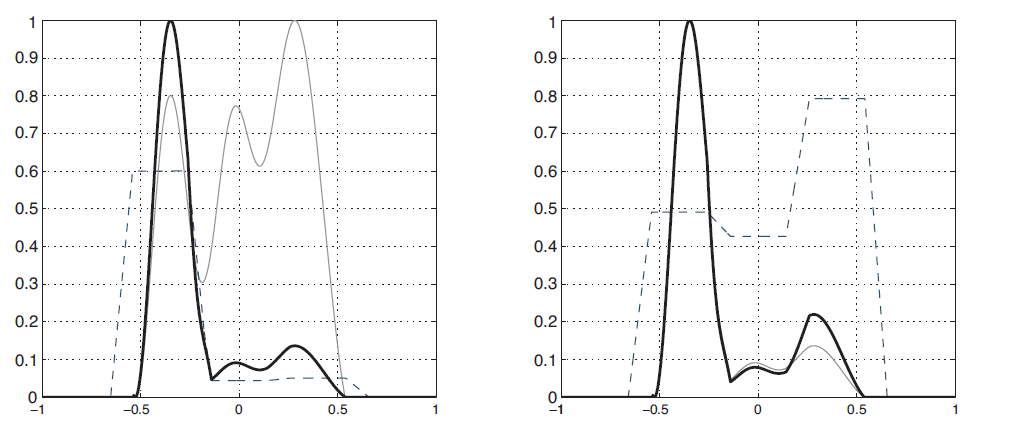
Здесь параметр регуляризации λ = 0,2 присваивается эмпирически, а последовательность вычисляется по N = 4 предыдущим событиям.

Уменьшение разложения распределений до пяти диапазонов предотвращает ограничение выразительности, вызванного переобучением и уменьшает количество вычислений в алгоритме. Количество диапазонов было выбрано эмпирически, чтобы контролировать влияние функции правдоподобия при поддержании достаточного уровня отклонения между распределениями. Распределение Р(В) используется в качестве средства нормализации и оценивается путем взятия интеграла по всем возможным значениям P(A)P(В|A).

Такая нормализация позволяет апостериорному распределению Р(A|B) быть представленным в виде распределения вероятностей. Коэффициент, использующийся для модуляции, затем генерируется с помощью этого распределения вероятности, таким образом, позволяя ему влиять на амплитуду определенного события. Для того чтобы сохранить динамичный характер последовательности и правильно обновлять будущие распределения, производится присвоение предыдущего апостериорного распределения априорному распределению P(A), иллюстрированное в уравнении 6.

(6)

Эта рекурсия обеспечивает непрерывное обновление априорного распределения, гарантируя, что текущее событие в последовательности зависит от всех предыдущих событий. Здесь каждая итерация алгоритма соответствует появлению нового события. На изображении 8 показана последовательность из четырех событий из одной партии, измененной с использованием байесовского метода.



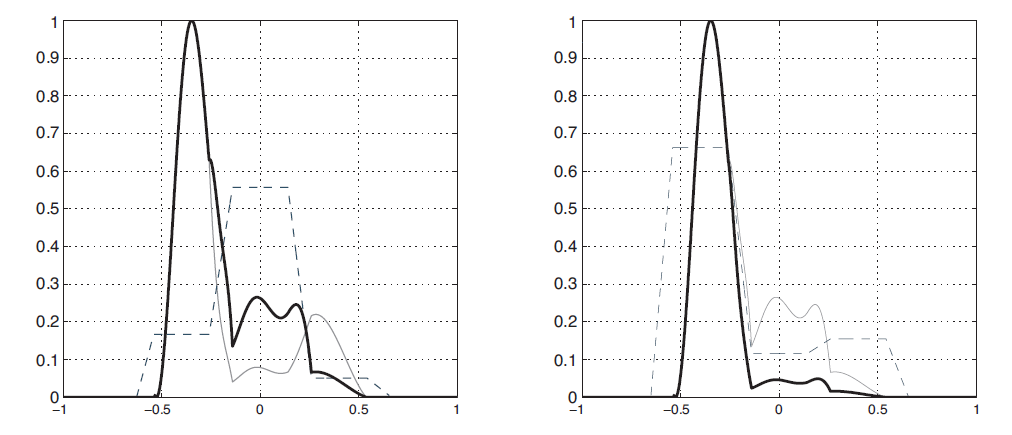


Рисунок 8. Последовательность 4 нормализованных t распределений, сгенерированных с использованием рекурсивного метода Байеса для 4 последовательных событий. Пунктирные линии представляют распределение вероятности, серые сплошные линии представляют собой априорные распределения, а видные черные линии представляют апостериорные распределения для каждого события.

* 1. Субъективная оценка

Для получения оценки моделей, учеными из Бирмингема были проведены слепые тесты, описанные в статье [9].

Для того чтобы оценить результат байесовской модели, были предварительно сформированы тесты прослушивания. Участникам с нормальным слухом было предложено классифицировать ударные последовательности, основанные на том, как похожи они на партию профессионального ударника. Участники были отобраны по их опыту с производством музыкальных произведений и ударными. Участникам воспроизвели в общей сложности 20 аудио файлов, все из которых состояли из одинаковых ударных последовательностей, которые были сыграны в одинаковом темпе. Образцы разделены на 4 различных группы, подробно описанных в разделе 3. Выбранная ударная последовательность часто используется в рок-ритме, сыгранная в размере 4/4 в темпе 120 ударов в минуту.

Образцы были представлены в парных тестах прослушивания, где участники могли выбрать пример из пары, который звучал наиболее по-человечески сыгранным. Средние баллы были взяты из тестов и представлены на рисунке 9. Предложенная байесовская модель выполнена аналогично СММ, основанной на системе, которая также использует вероятностный метод гуманизации событий. Байесовский подход имел немного более низкий балл 0,64, по сравнению с СММ, у которого был 0,69. Обе этих модели работают лучше, чем гауссовый метод (0,29), который имеет небольшое преимущество над квантованной последовательностью (0,22). Это свидетельствует о том, что, во-первых, гауссовские примеры кажутся более живыми, чем квантованные примеры. Это также показывает, что примеры, которые создаются с использованием вероятностных моделей звучат значительно более живо.

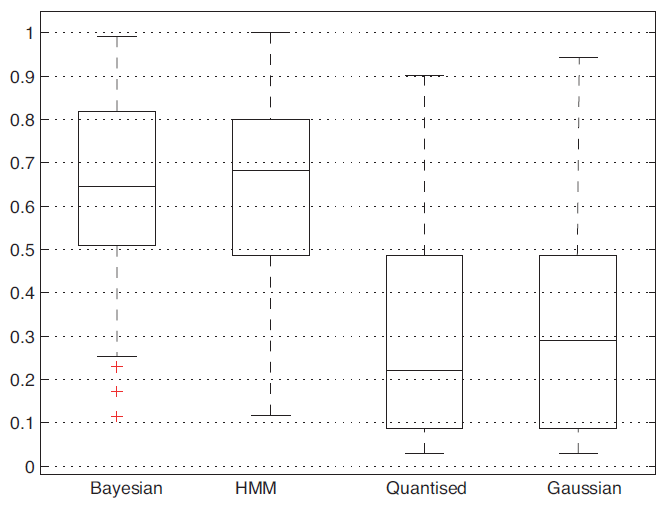


Рисунок 9. По оси Y представлены результаты прослушивания тестов с использованием 3 типов гуманизации и квантованной последовательности

# Собственная модификация

Для создания алгоритма оживления партий были изучены методы, описанные разделе 2.

Для примера был выбран образец из рисунка 4. К нему было применено квантование (рис. 10) для последующего применения методов оживления и были выровнены громкости нот (для нот-призраков малого барабана была выделена отдельная высота, и их громкость была снижена примерно на половину) для создания образца квантованной партии для субъективного сравнения с результатами оживления различными методами. Партия была экспортирована в форматах MIDI и mp3.

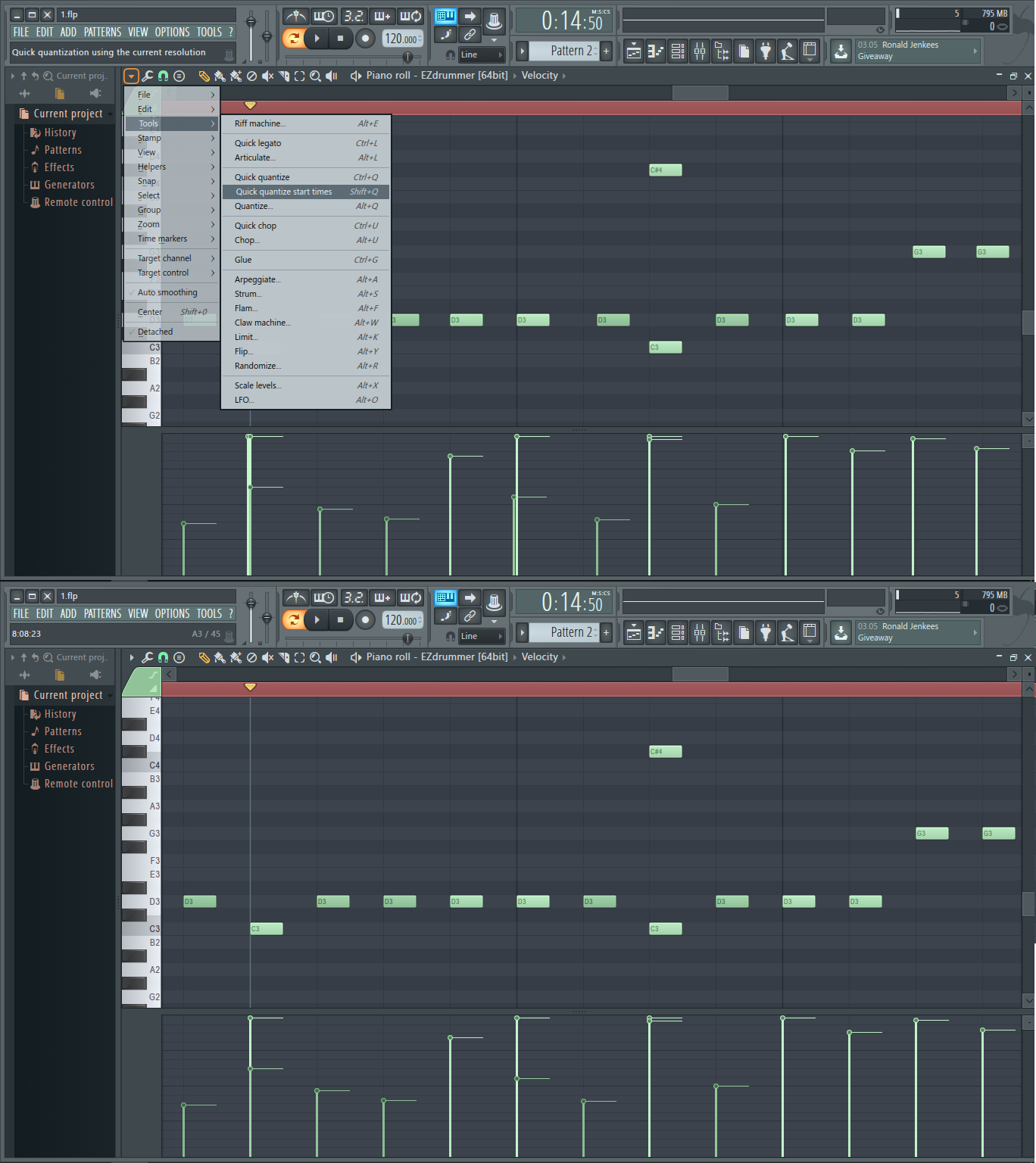


Рисунок 10. Пример квантизации

Для обработки партии в MATLAB считывается MIDI-файл с квантованной партией ударных инструментов:

nmat = readmidi('q.mid');

* 1. Создание алгоритма изменения громкости нот

Как уже известно, переменная nmat представляет собой матрицу, в которой строки соответствуют нотам, а столбцы – параметрам.

Для изменения громкости нот создается цикл для изменения параметров нот с первой до последней:

for i = 1:length(nmat(:,3))

Чтобы партия звучала наиболее живо, громкость нот различных ударных музыкальных инструментов изменяется по разным законам. Для этого сначала определяется часть ударной установки, например, если текущая нота – удар по большому барабану, то условие выглядит следующим образом:

if(nmat(i,4) >= 34 && nmat(i,4) <= 36)

Для придания динамики барабанной партии для всех нот, которые должны звучать максимально громко, можно изменить громкость при помощи следующей строки:

nmat(i,5) = 127 - rand \* 10;

В партиях часто встречаются двойные удары по большому барабану. Чаще всего, если второй из них в сильной доле, то первый удар тише.

Для оживления партии большого барабана снижается громкость предыдущего удара, что не в сильной доле, стоящего близко к текущему. Чтобы реализовать эту идею, для каждой ноты должен быть известен индекс предыдущей ноты - удара по большому барабану. Это было достигнуто с помощью условия, в котором присваивается индекс текущей ноты, которая при появлении следующей ноты становится предыдущей:

if(k == 0)

k1 = i;

k = k + 1;

else

k2 = i;

k = 0;

k3 = 1;

end

Если переменная k равна нулю, то индекс предыдущей ноты большого барабана хранится в переменной k2. Если переменная k равна единице, то индекс предыдущей ноты большого барабана хранится в переменной k1. Переменная k3 нужна как обозначение того, что известны обе переменные k1 и k2.

Переменные k и k3 были объявлены сразу после считывания MIDI-файла:

k = 0;

k3 = 0;

Величины изменения громкости нот были определены сравнением с партией живой игры на ударных музыкальных инструментах. В результате код выглядит так:

if(k3 == 1)

if(k == 0)

if(mod(nmat(k1,1),1) ~= 0 && nmat(i,1) - nmat(k1,1) == 0.5)

nmat(k1,5) = 100 - rand \* 30;

end

else

if(mod(nmat(k2,1),1) ~= 0 && nmat(i,1) - nmat(k2,1) == 0.5)

nmat(k2,5) = 100 - rand \* 30;

end

end

end

Код оживления партии малого барабана:

if(nmat(i,4) >= 39 && nmat(i,4) <= 40)

nmat(i,5) = 127 - rand \* 10;

end

Код оживления партии малого барабана для нот-призраков:

if(nmat(i,4) == 38)

nmat(i,5) = 100 - rand \* 70;

end

Код оживления партии том-томов:

if(nmat(i,4) == 41 || nmat(i,4) == 43 || nmat(i,4) == 45 || nmat(i,4) == 47 || nmat(i,4) == == 48)

nmat(i,5) = 127 - rand \* 10;

end

При игре на райде барабанщики часто акцентируют сильные доли и подчеркивают большой барабан. Таким образом, если текущая нота - райд, то если она в слабой доле, то она звучит тише, но только если не совпадает с большим барабаном:

if(nmat(i,4) >= 51 && nmat(i,4) <= 53)

if(mod(nmat(i,1),1) ~= 0)

if(i > 2 && i < length(nmat(:,5))-1)

if~((nmat(i-2,4) >= 34 && nmat(i-2,4) <= 36 && nmat(i-2,1) == nmat(i,1)) || || (nmat(i- -1,4) >= 34 && nmat(i-1,4) <= 36 && nmat(i-1,1) == nmat(i,1)) || (nmat(i+1,4) >= >= 34 && nmat(i+1,4) <= 36 && nmat(i+1,1) == nmat(i,1)) || (nmat(i+2,4) >= 34 && && nmat(i+2,4) <= 36 && nmat(i+2,1) == nmat(i,1)))

nmat(i,5) = 110 - rand \* 20;

end

end

else

nmat(i,5) = 127 - rand \* 10;

end

end

Барабанщики при совпадении хай-хэта с малым барабаном увеличивают громкость хай-хэта, подчеркивая малый барабан. При этом громкость обычно нарастает и в предыдущих ударах по хай-хэту.

Здесь определяется нота, соответствующая хай-хэту:

if(nmat(i,4) >= 7 && nmat(i,4) <= 26 || nmat(i,4) == 42 || nmat(i,4) == 44 || nmat(i,4) == == 46 || nmat(i,4) >= 60 && nmat(i,4) <= 65)

Как и на райде, так и при игре на хай-хэте, барабанщик сильные доли играет громче. Таким образом был написан следующий код:

if(mod(nmat(i,1),4) ~= 0) % если в слабую долю

nmat(i,5) = 80 - rand \* 15; % то удар слабее

end

Чтобы создать нарастание громкости в предыдущих ударах по хай-хэту, было решено, что для каждой ноты должны быть известны индексы двух предыдущих ударов по хай-хэту. Здесь было использовано условие, аналогичное условию для определения индекса предыдущего удара по большому барабану, которое присваивает индекс текущей ноты, которая при появлении следующей ноты становится предыдущей, а предыдущая становится предыдущей для предыдущей:

if(h == 0)

h1 = i;

h = h + 1;

else

if(h == 1)

h2 = i;

h = h + 1;

else

h3 = i;

h = 0;

h4 = 1;

end

end

Если переменная h равна нулю, то индекс предыдущей ноты хай-хэта хранится в переменной h2, а индекс предыдущей для h2 – в h1. Если переменная h равна единице, то индекс предыдущей ноты хай-хэта хранится в переменной h3, а индекс предыдущей для h3 – в h2. Если переменная h равна двум, то индекс предыдущей ноты хай-хэта хранится в переменной h1, а индекс предыдущей для h1 – в h3. Переменная h4 нужна как обозначение того, что известны переменные h1, h2 и h3.

Переменные h и h4 были объявлены сразу после считывания MIDI-файла:

h = 0;

h4 = 0;

В результате код выглядит следующим образом:

if(h4 == 1 && i < length(nmat(:,5))-1)

if((nmat(i-2,4) >= 38 && nmat(i-2,4) <= 40 && nmat(i-2,1) == nmat(i,1)) || || (nmat(i-1,4) >= 38 && nmat(i-1,4) <= 40 && nmat(i-1,1) == nmat(i,1)) || || (nmat(i+1,4) >= 38 && nmat(i+1,4) <= 40 && nmat(i+1,1) == nmat(i,1)) || || (nmat(i+2,4) >= 38 && nmat(i+2,4) <= 40 && nmat(i+2,1) == nmat(i,1)))

if(h == 0)

nmat(h1,5) = nmat(h1,5) + 10;

nmat(h2,5) = nmat(h2,5) + 20;

else

if(h == 1)

nmat(h2,5) = nmat(h2,5) + 10;

nmat(h3,5) = nmat(h3,5) + 20;

else

nmat(h3,5) = nmat(h3,5) + 10;

nmat(h1,5) = nmat(h1,5) + 20;

end

end

nmat(i,5) = 85 + rand \* 20;

end

end

* 1. Создание алгоритма сдвигов нот

На рисунке 1 можно заметить, что в партии, сыгранной человеком, у сдвигов есть определенная зависимость от предыдущих сдвигов, в отличие от гуманизированной гауссовым методом.

Было предположено, что каждая новая нота должна сдвигаться с разной вероятностью на определенные диапазоны отклонения от метрономной сетки в зависимости от предыдущего отклонения (рис. 11).

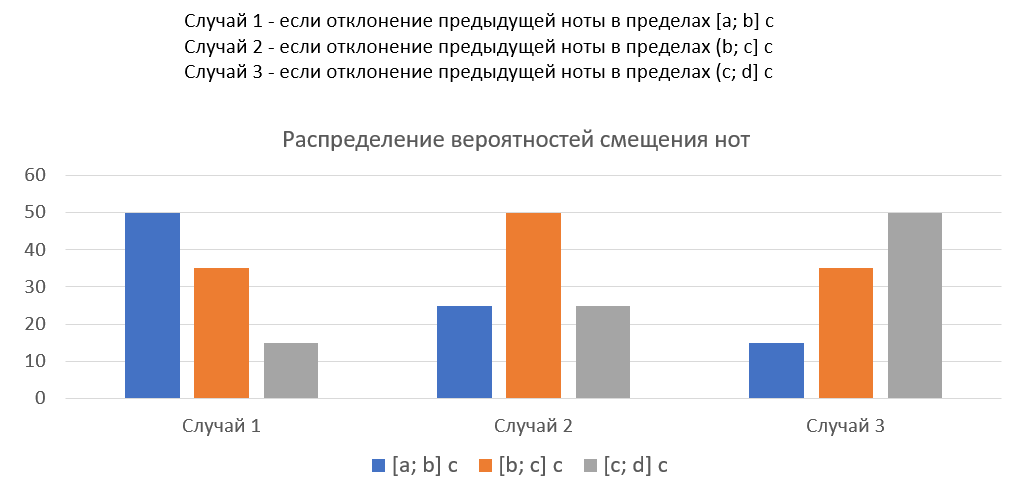


Рисунок 11. Распределение вероятностей смещения нот

Так как чаще всего наиболее ровные партии не имеют отклонения более 15 миллисекунд (на рисунке 12 изображена гистограмма, показывающая статистику отклонений от метрономной сетки партии из рисунка 4), сразу после считывания MIDI-файла были определены границы или диапазоны отклонений от метрономной сетки следующим образом:

a = -0.015;

b = -0.005;

c = 0.005;

d = 0.015;

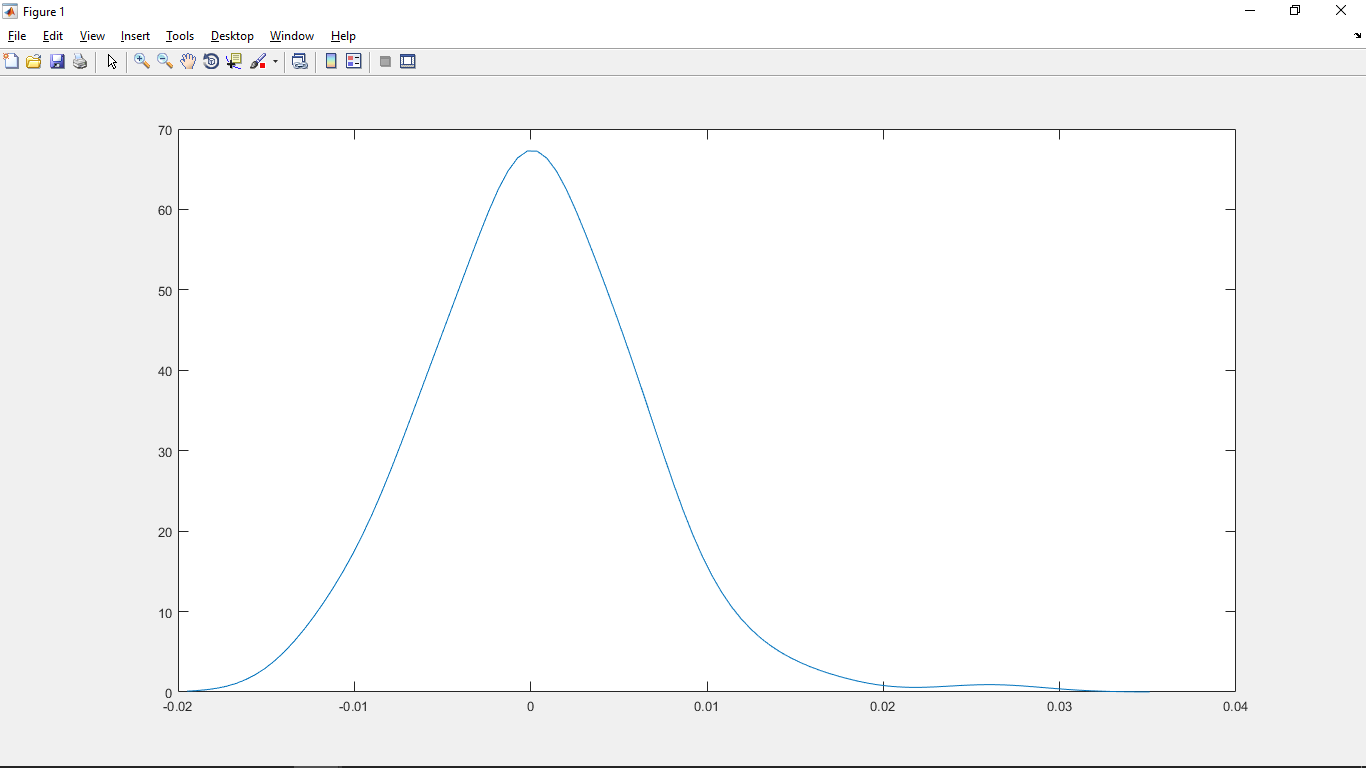


Рисунок 12. Гистограмма отклонений

Там же были объявлены переменные для распределений вероятности и сдвиг для первой ноты:

p1=50;

p2=85;

p3=75;

t(1) = 0;

Для сдвигов нот создается цикл для изменения параметров нот со второй до последней:

for i = 2:length(nmat(:,6))

Затем генерируется целое случайное число от 1 до 100:

r=randi(100);

Далее, следуя алгоритму, изображенному на рисунке 5, если отклонение предыдущей ноты в пределах [a; b], если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1% смещение ноты приходится на промежуток [a; b]:

if(t(i-1) <= b)

if (r <= p1)

t(i) = a + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p1; p2], то с вероятностью p2-p1% смещение ноты приходится на промежуток [b; c]:

else

if (r <= p2)

t(i) = b + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p2; 100], то с вероятностью 100-p2% смещение ноты приходится на промежуток [c; d]:

else

if (r > p2)

t(i)= c + rand \* 2 \* c;

Если отклонение предыдущей ноты в пределах (b; c], если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1% смещение ноты приходится на промежуток [b; c]:

if(t(i-1) <= c)

if (r <= p1)

t(i) = b + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p1; p3], то с вероятностью p3-p1% смещение ноты приходится на промежуток [a; b]:

else

if (r <= p3)

t(i) = a + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p3; 100], то с вероятностью 100-p3% смещение ноты приходится на промежуток [c; d]:

else

if (r > p3)

t(i)= c + rand \* 2 \* c;

При отклонении предыдущей ноты в пределах (c; d] код аналогичен коду при отклонении предыдущей ноты в пределах [a; b]. Если отклонение предыдущей ноты в пределах (c; d], если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1% смещение ноты приходится на промежуток [c; d]:

if(t(i-1) <= d)

if (r <= p1)

t(i) = c + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p1; p2], то с вероятностью p2-p1% смещение ноты приходится на промежуток [b; c]:

else

if (r <= p2)

t(i) = b + rand \* 2 \* c;

Иначе, если возвратилось число в пределах (p2; 100], то с вероятностью 100-p2% смещение ноты приходится на промежуток [a; b]:

else

if (r > p2)

t(i)= a + rand \* 2 \* c;

В конце итерации происходит смещение текущей ноты в секундной системе отсчета:

nmat(i,6) = nmat(i,6) + t(i);

Для экспорта матрицы в MIDI-файл используется следующая команда:

writemidi(nmat,'modified.mid');

* 1. Создание примеров

Для создания примера гауссовой гуманизации был использован следующий код:

nmat = readmidi('q.mid');

nmat(:,6)=normrnd(nmat(:,6),0.007)

nmat(:,5)=normrnd(nmat(:,5),5)

writemidi(nmat,'gauss.mid');

После экспорта, для создания примеров, файлы были импортированы в цифровую звуковую рабочую станцию. Таким образом партии приобрели звучание барабанной установки плагина EZdrummer 2. Из цифровой звуковой рабочей станции были экспортированы mp3-файлы.

* 1. Сравнение результатов

Для сравнения результатов, с помощью следующих команд MATLAB были получены графики, показывающие насколько близка динамика партий, гуманизированных гауссовым и модифицированным методами, к партии, сыгранной профессиональным барабанщиком:

ksdensity(g(:,5)-h(:,5))

ksdensity(nmat(:,5)-h(:,5))

Первый график показывает разницу в амплитудах (по оси x) соответствующих нот партий, насколько партия, гуманизированная гауссовым методом, отличается от сыгранной человеком:

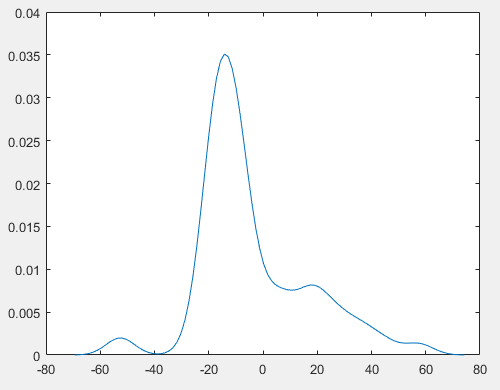


Рисунок 13. Сравнение гауссового метода

Второй график показывает разницу в амплитудах соответствующих нот партий, насколько партия, гуманизированная модифицированным методом, отличается от сыгранной человеком:

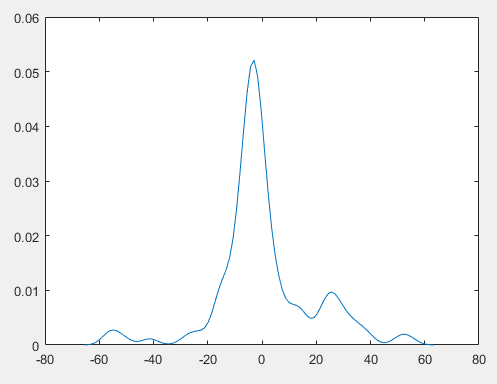


Рисунок 14. Сравнение модифицированного метода

На графиках видно, что при использовании модифицированного метода, разница в динамике партий ближе к нулю:

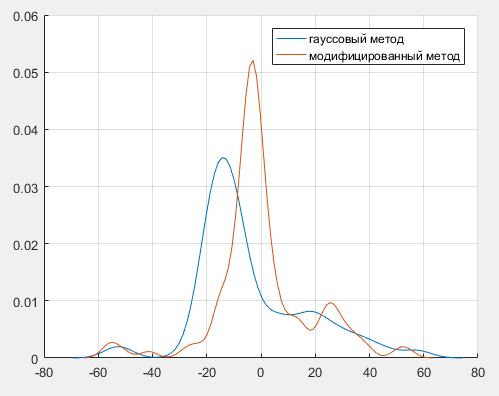


Рисунок 15. Сравнение методов

Результаты сравнения с игрой реального барабанщика можно представить в виде чисел. Для гауссового метода был использован следующий код:

dif = 0;

for i = 1:length(h(:,5))

dif = dif + abs(g(i,5) - h(i,5));

end

dif

И получен результат: 1951.

Для модифицированного метода был использован следующий код:

dif = 0;

for i = 1:length(h(:,5))

dif = dif + abs(nmat(i,5) - h(i,5));

end

dif

И получен результат: 1346.

Это показывает, что при использовании модифицированного метода разница в динамике партий меньше, в сравнении с использованием гауссового метода, примерно на 31%.

# Обсуждение и вывод

Правило Байеса было использовано для применения условной вероятности к амплитуде и сдвигам нот в партии ударных и этот метод был протестирован сравнением с широко используемой гауссовой моделью, квантованной последовательностью и альтернативной вероятностной моделью. Из результатов на рисунке 9 очевидно, что модель работает очень сходно с моделью, основанной на СММ, которая превосходит гауссову модель в эмуляции естественной человеческой артикуляции. Модель позволяет решить некоторые проблемы, связанные с гауссовой гуманизацией, которые включают в себя отсутствие временной зависимости, опущение локальных диапазонов и упрощение человеческого выражения.

Для того, чтобы сохранить структурные элементы последовательности, была использована функция правдоподобия P(В|A), которая представляет собой определение согласованности между текущими и предыдущими событиями, в виде ряда условных дельта измерений. Эти измерения параметризируются распределениями, состоящих из пяти равноудаленных диапазонов, представляющие подраздел амплитуды и (слухового) временного интервала различения, таким образом, позволяя распределениям условно зависеть от предыдущих событий. Это совпадает с наблюдениями, сделанными на наборе данных ударников, в которых структурная схема была очевидна и в амплитуде (*a*) и моментах возникновения нот (*t*).

Сегментация в группы компонентов в последовательности позволила рассмотреть существование локальных диапазонов. Здесь присваиваются дискретные распределения к различным элементам ударной установки для того, чтобы поддерживать согласованность амплитуды и места возникновения нот. Это устраняет предположение о том, что удары по разным элементам ударной установки имеют равные шансы возникновения с той же амплитудой и с тем же сдвигом по сетке.

Использование эмпирических распределений позволяет применить вероятность, которая зависит от ударника, играющего в предопределенном стиле. Это решает вопрос упрощения в гауссовских моделях и позволяет изменять исходный материал в наборе данных, чтобы достичь изменений в распределении вероятностей. Это полезно, если нужен стиль конкретного ударника или определенный жанр. Решая основную проблему гауссового метода, становятся доступны нюансы, которые проявляются отдельными ударниками, и система имеет больше сходств с человеком.

В то время как байесовская модель превосходит гауссовский аналог, есть еще ряд вопросов, которые еще не рассматривались в исследовании [9]. Например, модель не несет никакой взаимозависимости между параметрами *a* и *t*. В акцентированных последовательностях часто существуют события, которые можно выделить с помощью смещения и амплитуды. Данные события опущены с использованием текущего байесовского метода. Также, система не отслеживает местоположение в музыкальной композиции. Поэтому такие механизмы, как крещендо и диминуэндо, которые часто встречаются в перцептивно соответствующих местах в музыкальной композиции отсутствуют в рамках существующей системы.

В собственной модификации также были решены проблемы, проявляющиеся при гауссовой гуманизации. Была создана зависимость параметров нот не только от параметров предыдущих нот, но и от последующих нот соответствующих частей ударной установки. Создана зависимость силы удара от положения ноты в такте. Таким образом акцентируются ноты в сильной доле. Также была создана зависимость громкости ноты от совпадения с нотами других частей ударной установки. Для увеличения динамичности и живости партии было реализовано нарастание громкости хэта перед ударом по малому барабану.

При прослушивании тестов, партии, обработанные с использованием модифицированного метода, звучат живее, чем с использованием гауссового и квантованные партии.

# Заключение

В результате выполнения дипломной работы было исследовано применение рекурсивного байесовского метода к "оживлению" midi-партий ударных музыкальных инструментов при производстве цифрового музыкального контента и создан алгоритм, приближающий midi-партии ударных музыкальных инструментов, исполненных компьютером к человеческому исполнению.

В процессе выполнения дипломной работы были решены следующие задачи:

1. Изучены средства обработки midi-партий.
2. Изучены методы оживления midi-партий ударных музыкальных инструментов.
3. Проведены оценка и сравнение результативности методов.

Данный алгоритм оптимизирован не для всех композиций. Для улучшения алгоритма, его можно оптимизировать для остальных композиций, например таких, в которых используются нестандартные размеры или которые сыграны в специфическом стиле.

# Список использованных источников

1. R.I. Stables, J. Bullock, and I. Williams, "Perceptually relevant models for drum pattern humanisation" in Proceedings of 131st Audio Engineering Society Convention, New York, 2011, AES.
2. Журнал «Звуковые виртуальные студии». – 2009. – №1 - Режим доступа: http://zstudio-n.narod.ru/zvs5.html, свободный.
3. MIDI. - Режим доступа: https://ru.wikipedia.org/wiki/MIDI, свободный.
4. MIDI и OSC — основные протоколы взаимодействия музыкальных приложений. - Режим доступа: https://habrahabr.ru/post/139226/, свободный.
5. Что означают MIDI-события. - Режим доступа: https://samesound.ru/studio/midi-work/70572-midi-messages-explain, свободный.
6. T. Eerola and P. Toiviainen, "Mir in matlab: The midi toolbox" in Proceedings of 5th International Conference on Music Information Retrieval, Barcelona, Spain, 2004, ISMIR.
7. Программирование ударных: самые распространенные ошибки. - Режим доступа: https://samesound.ru/studio/midi-work/302-programmirovanie-udarnyh-samye-rasprostranennye-oshibki, свободный.
8. 20 Советов По Оживлению MIDI Барабанов. - Режим доступа: http://recording-studio.ru/2008/07/03/20-sovetov-po-ozhivleniyu-midi-barabanov/, свободный.
9. R.I. Stables, C. Athwal, and R. Cade, "Drum pattern humanisation using a recursive bayesian framework" in Proceedings of 133st Audio Engineering Society Convention, San Francisco, 2012, AES.
10. L. O'Sullivan and F. Boland, "Towards a Fuzzy Logic Approach To Drum Pattern Humanisation" in Proc. of the 13th Itl. Conference on Digital Audio Effects (DAFx-10), Graz, Austria, Sept. 19-21, 2010.

# Приложение

nmat = readmidi('q.mid');

k = 0;

k3 = 0;

h = 0;

h4 = 0;

t(1) = 0;

% границы

a = -0.015;

b = -0.005;

c = 0.005;

d = 0.015;

% probability

p1=50;

p2=85;

p3=75;

for i = 1:length(nmat(:,3))

disp(i)

if(nmat(i,4) >= 34 && nmat(i,4) <= 36) % если это бочка

nmat(i,5) = 127 - rand \* 10;

if(k == 0)

k1 = i;

k = k + 1;

else

k2 = i;

k = 0;

k3 = 1;

end

if(k3 == 1)

if(k == 0)

% если предыдущий удар не на сильную долю и стоит близко к текущему

if(mod(nmat(k1,1),1) ~= 0 && nmat(i,1) - nmat(k1,1) == 0.5)

nmat(k1,5) = 100 - rand \* 30; % то предыдущий удар слабее

end

else

if(mod(nmat(k2,1),1) ~= 0 && nmat(i,1) - nmat(k2,1) == 0.5)

nmat(k2,5) = 100 - rand \* 30;

end

end

end

end

% если это рабочий барабан

if(nmat(i,4) == 38) % ноты-призраки

nmat(i,5) = 80 - rand \* 60;

end

if(nmat(i,4) >= 39 && nmat(i,4) <= 40)

nmat(i,5) = 127 - rand \* 10;

end

% если это томы

if(nmat(i,4) == 41 || nmat(i,4) == 43 || nmat(i,4) == 45 || nmat(i,4) == 47 || nmat(i,4) == 48)

nmat(i,5) = 127 - rand \* 10;

end

% если это райд

if(nmat(i,4) >= 51 && nmat(i,4) <= 53)

if(mod(nmat(i,1),1) ~= 0) % если в слабую долю

if(i > 2 && i < length(nmat(:,5))-1)

if~((nmat(i-2,4) >= 34 && nmat(i-2,4) <= 36 && nmat(i-2,1) == nmat(i,1)) || (nmat(i-1,4) >= >= 34 && nmat(i-1,4) <= 36 && nmat(i-1,1) == nmat(i,1)) || (nmat(i+1,4) >= 34 && nmat(i+1,4) <= 36 && && nmat(i+1,1) == nmat(i,1)) || (nmat(i+2,4) >= 34 && nmat(i+2,4) <= 36 && nmat(i+2,1) == nmat(i,1)))

nmat(i,5) = 110 - rand \* 20; % то удар слабее, если не совпадает с бочкой

end

end

else

nmat(i,5) = 127 - rand \* 10;

end

end

% если это хет

if(nmat(i,4) >= 7 && nmat(i,4) <= 26 || nmat(i,4) == 42 || nmat(i,4) == 44 || nmat(i,4) == 46 || || nmat(i,4) >= 60 && nmat(i,4) <= 65)

if(h == 0)

h1 = i;

h = h + 1;

else

if(h == 1)

h2 = i;

h = h + 1;

else

h3 = i;

h = 0;

h4 = 1;

end

end

if(mod(nmat(i,1),4) ~= 0) % если в слабую долю

nmat(i,5) = 80 - rand \* 15; % то удар слабее

end

if(h4 == 1 && i < length(nmat(:,5))-1) % если совпадает с рабочим, то сильнее

if((nmat(i-2,4) >= 38 && nmat(i-2,4) <= 40 && nmat(i-2,1) == nmat(i,1)) || (nmat(i-1,4) >= 38 && && nmat(i-1,4) <= 40 && nmat(i-1,1) == nmat(i,1)) || (nmat(i+1,4) >= 38 && nmat(i+1,4) <= 40 && && nmat(i+1,1) == nmat(i,1)) || (nmat(i+2,4) >= 38 && nmat(i+2,4) <= 40 && nmat(i+2,1) == nmat(i,1)))

% нарастание громкости хета подчеркивет рабочий барабан

if(h == 0)

nmat(h1,5) = nmat(h1,5) + 10;

nmat(h2,5) = nmat(h2,5) + 20;

else

if(h == 1)

nmat(h2,5) = nmat(h2,5) + 10;

nmat(h3,5) = nmat(h3,5) + 20;

else

nmat(h3,5) = nmat(h3,5) + 10;

nmat(h1,5) = nmat(h1,5) + 20;

end

end

nmat(i,5) = 85 + rand \* 20;

end

end

end

end

%смещения onset

for i = 2:length(nmat(:,6))

r=randi(100); % генерация целого случайного числа от 1 до 100

if(t(i-1) <= b) % если отклонение предыдущей ноты в пределах [a; b]

% если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1%

% смещение ноты приходится на промежуток [a; b]

if (r <= p1)

t(i) = a + rand \* 2 \* c;

else

% если возвратилось число в пределах (p1; p2], то с вероятностью p2-p1%

% смещение ноты приходится на промежуток [b; c]

if (r <= p2)

t(i) = b + rand \* 2 \* c;

else

% если возвратилось число в пределах (p2; 100], то с вероятностью 100-p2%

% смещение ноты приходится на промежуток [c; d]

if (r > p2)

t(i)= c + rand \* 2 \* c;

end

end

end

else

if(t(i-1) <= c) % если отклонение предыдущей ноты в пределах (b; c]

% если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1%

% смещение ноты приходится на промежуток [b; c]

if (r <= p1)

t(i) = b + rand \* 2 \* c;

else

% если возвратилось число в пределах (p1; p3], то с вероятностью p3-p1%

% смещение ноты приходится на промежуток [a; b]

if (r <= p3)

t(i) = a + rand \* 2 \* c;

else

% если возвратилось число в пределах (p3; 100], то с вероятностью 100-p3%

% смещение ноты приходится на промежуток [c; d]

if (r > p3)

t(i)= c + rand \* 2 \* c;

end

end

end

else

if(t(i-1) <= d) % если отклонение предыдущей ноты в пределах (c; d]

% если возвратилось число меньше или равное p1 [1; p1], то с вероятностью p1%

% смещение ноты приходится на промежуток [c; d]

if (r <= p1)

t(i) = c + rand \* 2 \* c;

else

% если возвратилось число в пределах (p1; p2], то с вероятностью p2-p1%

% смещение ноты приходится на промежуток [b; c]

if (r <= p2)

t(i) = b + rand \* 2 \* c;

else

% если возвратилось число в пределах (p2; 100], то с вероятностью 100-p2%

% смещение ноты приходится на промежуток [a; b]

if (r > p2)

t(i)= a + rand \* 2 \* c;

end

end

end

end

end

end

nmat(i,6) = nmat(i,6) + t(i); % смещение текущей ноты

end

writemidi(nmat,'modified.mid');

disp('готово!')