Государственное образовательное учреждение

высшего профессионального образования

***«Московский государственный технический***

***университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)***

Отчет по НИРС на тему:

**«Анализ методов синхронизированности**

**нотных записей и живой музыки»**

Выполнил студент группы ИУ7-38

Студент: Кочуркин И.А.

Научный руководитель: Филиппов М.В.

Оглавление

[Введение 3](#_Toc349507725)

[Проблема синхронизации нотных записей и живой музыки 4](#_Toc349507726)

[Типы нотных событий 6](#_Toc349507727)

[Классификация времени исполнения 6](#_Toc349507728)

[Вероятностная модель времени 7](#_Toc349507729)

[Марковская модель времени. 7](#_Toc349507730)

[Полумарковская модель времени. 8](#_Toc349507731)

[Стохастическая модель времени в музыкальной игре 9](#_Toc349507732)

[Предупредительная модель темпа 10](#_Toc349507733)

[Преобразование исходного аудио сигнала в последовательность состояний. 11](#_Toc349507734)

[Сопоставление цепочек состояний 12](#_Toc349507735)

[Динамическая трансформация шкалы времени 13](#_Toc349507736)

[Алгоритм Витерби 16](#_Toc349507737)

[Определение и предсказания темпа исполнения. 17](#_Toc349507738)

[Метод скользящей средней 17](#_Toc349507739)

[Фильтр Калмана 17](#_Toc349507740)

[Заключение 18](#_Toc349507741)

[Список используемых источников 19](#_Toc349507742)

# Введение

**Синхронизация нот** - это процесс сопоставления звукового сигнала с загруженными заранее нотами в символьном формате.

Существуют два типа синхронизаций:

1. Непосредственно во время исполнения музыкальной композиции (в реальном времени).
2. Время сопоставления аудиосигнала и символьных нот неограниченно (оффлайновый режим).

К достоинству первого типа синхронизации можно отнести то, что по нему возможно интерактивное обучение новым партиям, в этом случае музыкант может увидеть сразу все ошибки.

К достоинству второго типа синхронизации можно отнести то, что в этом случае точность сопоставления может быть в теории выше, чем у первого типа.

Однако так как тема диссертации подразумевает разработку методику оценки исполнения нотных записей, то второй тип синхронизации рассматриваться не будет.

Для того чтобы осуществить процесс синхронизации живой музыки и символьных нот в реальном времени, необходимо решить следующие задачи:

1. Представление исходного сигнала в виде последовательности состояний.
2. Сопоставление полученной цепочки состояний с правильной и определение текущей играемой ноты.
3. Определение и предсказания темпа исполнения.

Общая схема системы изображена на рис.

Стоит отметить, что определение темпа актуально только для первого типа синхронизации (в реальном времени).

**Целью** данной работы является исследование методов для решения задач сопоставления цепочек состояний и определения темпа в реальном времени для разработки метода в дальнейшем, наиболее подходящего для оценки их соответствия.

Далее методы и алгоритмы всех трех этапов будут рассмотрены более подробно.

# Проблема синхронизации нотных записей и живой музыки

Проблемой синхронизации нотных записей занимались многие выдающиеся ученые и исследователи. Ими было разработано и опробовано множество различных методов и способов, которые используются и по сей день, и являются основой современных разработок и исследований в области компьютерной музыки. Однако изучение методов синхронизации применительно к музыкальным мелодиям началось относительно недавно.

Технология синхронизации играемых нот впервые была представлена в 1984 Барри Версое и Рогером Данненбергом независимо. В ней использовался алгоритм Виттерби – это алгоритм поиска наиболее подходящего списка состояний (называемого путём Витерби), который в контексте цепей Маркова получает наиболее вероятную последовательность произошедших событий. Из-за вычислительных ограничений в то время, представленные системы работали с символьными данными, а не обрабатывали аудиопоток в реальном времени. Тогда проблема была сведена к сопоставлению строк между символьным входным потоком и последовательностью нот в реальном времени. Задача становилась более сложной, когда музыкант использовал более выразительную игру: менял темп, увеличивал или уменьшал громкость, добавлял трели или прочие выразительные средства. Лежащие в основе алгоритмы не были адаптированы к этим отклонениям, чтобы можно было восстановить синхронизацию с человеческой игрой. Все эти факторы создавали проблемы, даже если входной поток являлся символьный.

В начале 1990-ых, с появлением быстрых компьютеров, стало возможно использование прямого входного аудиопотока вместо символьных данных, позволяющее музыкантам использовать их инструменты в реальном времени. В этом новом подходе, символьный уровень нот не анализируется напрямую и практически скрыт от системы. Первые попытки построения таких систем заключались в использовании монофонического распознавании тонов, с дальнейшей передачей этой информации в алгоритм сопоставления для обработки, таким образом, усугубляя проблему синхронизации и адаптированности.

В середине 90 Граб и Рафаэль, параллельно с разработкой системы распознавания речи, внедрили стохастические системы. Они базировались на скрытых Марковских моделях и статическом наблюдении за аудиопотоком в реальном времени. Достижение Рафаэля позже было модифицировано различными учеными. В более поздних разработках, Рафаэль внедрил полифоническую систему. В той системе было два этапа обработки:

* Обработка нотных позиций, которая включала в себя скрытую марковскую сеть, отвечающую за сопоставление играемых музыкантом нот и нот из партии.
* Во втором этапе использовалась детально проработанная Байесовская сеть для сглаживания темпа во время исполнения.

У музыкантов и ученых существую различные мнения в отношении синхронизации играемых нот. Ученые требуют точного сопоставления, в то время как музыканты хотят, чтобы система была более гибкой и подстраивалась под музыканта. В 2007, благодаря композитору Макро Стропа, появилась новая парадигма технологии отслеживания нот, которая называется «предупреждающая». В ней и темп и ноты обрабатываются в реальном времени таким образом, что система старается предсказать или угадать, что хочет сыграть музыкант. Данные соображения привели к разработке проекта «Антескофо», которая стала стандартом во многих сферах, включая живую электронику.

Благодаря хорошим результатам программы «Антескофо», цели системы синхронизации нот становились более и более сложными. Это привело к расширению парадигмы синхронизации нот к парадигме «синхронного программирования». Электронные события стали полифоническим программами, работающими одновременно и параллельно с исполнителем. Эта была попытка уменьшения разрыва между перформативными и композиционными аспектами компьютерной музыки. Синхронный язык программирования «Антескофо» стал удовлетворять многие нужды композиторов.

Также существуют так называемые оффлайновые системы для синхронизации нот. Такие системы не работают в реальном времени. В них вся входная аудиопоследовательность заранее известна.

Таким образом, данное направление является молодым и перспективным, имеющим много проблем, которые или вовсе не решены, или для которых можно найти более оптимальное решение.

Восприятие структуры музыки во времени это не просто анализ ритма. Напротив, оно подразумевает стратегию «активного прослушивания», в которой ожидания слушателя о бушующих событиях имеют такое же важное место, как и текущие музыкальные события. Данные соображения указывают на то, что временная структура музыкальных ожиданий это динамическая структура, и она должна быть заложена в архитектуре, в отличие от методов сопоставления речи и фонем. Анализируя традиционные западные музыкальные нотации, самый простой путь описания темпоральной динамики заключается в создании множества дискретных событий, состоящих из нот и тишины, которые занимают определенное количество времени. Различные длительности этих событий сильно влияют на экспрессивность исполнения, которая обычно управляется темпом, представляющимся в ударах в минуту (BPM).

## Типы нотных событий

Существует два типа нотных событий: **темпоральные** и **атемпоральные**. Атемпоральные (вне времени) события соответствуют внутренней темпоральной структуре независимо от общей темпоральной структуры композиции. Две структуры обычно рассматриваются независимыми в музыкальной теории. Для согласования этих различий с вероятностной моделью, нужно определить атемпоральные объекты или события. Типичными примерами являются подчеркнутые ноты, трели, которые используются в барокко-стиле в западной музыкальной нотации. В этих случаях, индивидуальные события не вносят поправок в темп, но их общее темпоральное представление влияет на темп.

## Классификация времени исполнения

Время можно классифицировать по категориям: **прерывистое** и **сглаженное**. Прерывистое время это такое время, которое основано на текущем темпоральном порядке, в то время как сглаженное время это непрерывное представление времени как потока информации. Скачкообразное время, используемое в большинстве западных классических музыкальных нотациях, это регулируемый прерывистый временной поток, который использует внутренний музыкальный таймер, управляемый темпом, в ударах в минуту. В работе различаются понятия между скачкообразной временной шкалой, где время зависит от константного темпа, и сглаженной временной шкалой, где информация на микроуровне состоит из единичных вневременных элементов или определена относительно скачков. Типичным примером события в сглаженном времени в западной традиционной нотации – это свободное глиссандо. Важно заметить, что большинство доступных классических и популярных мелодий относится к прерывистому времени.

Вероятностная модель времени

Из-за внутренне присущей темпоральной структуры музыки, способности представления и декодирования темпоральных событий, должны быть присуще любым системам следования нотам. В общем, система синхронизации в реальном времени оценивает живой аудиопоток в соответствие с временной моделью символьных нот в памяти. С тех пор как такие системы оперируют в неопределенных ситуациях, вероятностная модель стала трендом в моделировании с конца 1990. Целью данной методики является декодирование темпоральной динамики с внешним процессом. Таким образом, производительность таких систем во многом зависит от их способности представлять такую динамику через их внутреннюю топологию. В этих задачах, любое состояние процесса занимает некоторое время, которое может быть детерминированным или нет. В моей работе затрагивается вероятностная модель длительности макро-состояний и ожидаемых событий. В музыкальном контексте, макро-состояниями являются музыкальные события (ноты, тишина, аккорды, трели и т.д.) с ожидаемой длительностью, которые в свою очередь могут состоять из микро-состояний. Наиболее общий способ моделирования временных рядов достигается с помощью моделей пространственных состояний (state-space models). Модель пространственных состояний последовательности – это помеченная во времени последовательность графов, где каждый узел указывает на состояние системы во времени. Таким образом, каждое состояние имеет внешнюю временную протяжённость, которая может использоваться для вероятностной модели локализации и длительности рассматриваемых событий. В данной работе будут рассматриваться два широких класса моделей пространственных состояний и их моделей времени, которые покрывают большинство существующих достижений: это марковские и полу-марковские процессы.

Марковская модель времени.

В параметрической модели времени Маркова, ожидаемые длительности макро состояний (такие события как ноты, аккорды и другие, которые занимают время) моделируются с помощью множества Марковских цепочек (или микро-состояний) со случайными переменными, упитывающихся в переходных вероятностях, которые параметризуют временные вероятности , где случайная переменная подсчитывает количество времени, проведенного в макросостоянии . Рисунок ниже демонстрирует параметрическую топологию макросостояний Марковской цепочки, широко использующейся для моделирования продолженности. Таким образом, макросостояние состоит из Марковских состояний и двух параметров и , означающих выходную вероятность и вероятность перехода в следующее состояние соответственно. Распределение длительностей макросостояний, связанное с общей топологией, выглядит следующим образом:

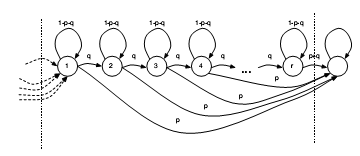


Рисунок . Параметрическая марковская сеть.

Если , это распределение выглядит следующим образом:

которое соответствует набору состояний без прыжков в выходное состояние с недостатком, заключающемся в минимальном времени, проведенном в макросостоянии . Эта упрощенная версия была широко используемой в различных системах отслеживания нот, где два параметра и извлекались из оптимизации над продолжительностью макросостояний из музыкальных нот.

Полумарковская модель времени.

В полумарковской модели макросостояния могут быть смоделированы с помощью одного состояния (вместо фиксированного набора микросостояний) с использованием явно заданного вероятностного распределения времени , для каждого состояния и времени . Допуская что это дискретная случайная переменная обозначающая макросостояния в момент времени i из пространства состояний , и это время проведенное в каждом состоянии m, тогда всякий раз когда

Проще говоря, мы находимся в состоянии во времени когда модели длительностей для всех состояний до и соответствует рассчитанному времени. В такой конфигурации весь процесс в целом не является Марковским в контексте макросостояний, но является марковским в контексте микросостояний.

Явное распределение времени определяется следующим образом:

Где , а – верхняя граница времени, проведенного в макросостоянии.

Полумарковские модели были впервые введены для распознавания речи и были широко использованы из-за их интуитивного применения к структуре времени. Полумарковские топологии обычно менее ресурсоемкие и более контролируемые по сравнению с обычными Марковскими. Более того, они обеспечивают явный доступ к моделям времени, выраженных как распределение событий. Несмотря на эти достоинства, явные протяженные модели могут требовать существенных разработок в стандартных алгоритмов статического вывода. Такие разработки могут стать слишком громоздкими, если модели времени должны быть динамическими, а не статическими (как в распознавании речи).

Структура системы отслеживания нот представлена на рисунке.

Стоит отметить, что музыкальные ноты в данном случае представлены как вероятностная пространственная модель из состояний, сконструированная непосредственно из символьной нотации. Система реального времени извлекает наблюдаемые вероятности из аудиопотока, учитывая состояния из нотного графа. Цель такой системы – интеграция мгновенных событий с прошлыми и бедующими для того чтобы определить позицию и темп исполнения.

## Стохастическая модель времени в музыкальной игре

Как было замочено ранее, любая модель для синхронизации времени музыкальных событий должна рассматривать гипотезы что темпоральные структуры ожиданий слушателей – это динамические системы. Первичная функция таких структур – предсказание будущих событий, включая восприятие и координацию действий с музыкальными событиями.

Формула для определения абсолютного времени события в секундах описывается следующим рекуррентным соотношением:

В которой – темп (секунды/такты).

Тем не менее, даже если целая часть имеет фиксированный темп, переменная подвергается различным изменениям из-за экспрессивности музыкального исполнения. Цель работы – вывести динамику темпа как случайную переменную во времени.

## Предупредительная модель темпа

Внутренний темп представляется через случайную переменную означающая насколько быстро музыка исполняется в соответствие с физическим временем. Следуя [9], нужно смоделировать поведение такой случайно переменной как внутренний осциллятор, погруженный в музыкальное исполнение. Такой осциллятор может быть представлен с помощью sine карт окружностей. Эти модели хорошо изучены в литературе и могут рассматриваться как нелинейные модели осцилляторов, которые погружены в периодический сигнал с помощью дискретно-временного формализма. В этом случае фаза карты окружностей – это абстракций времени и она соответствует времени прохода одного кругового периода локального темпа. Таким образом, нужно представить

# Преобразование исходного аудио сигнала в последовательность состояний.

Для определения играемой музыкантом позиции в нотах, исходный звуковой сигнал сначала необходимо привести к виду, который можно было бы сопоставлять с символьными нотами. Для этого исходный поток и ноты представляются в виде последовательности состояний.

Для этого этапа необходимо решить следующие задачи:

* Оцифровка сигнала.
* Нарезка на звуковые фрагменты.
* Фильтрация фрагметнов.
* Определение частоты основного тона фрагмента.
  + Временная область (автокорреляция).
  + Частотная область (гармоническое перемножение спектров, HFC).
* Преобразование частоты в ноты (состояния).

Поскольку данная работа не затрагивает данный этап, к тому же эти задачи были частично решены в бакалаврской работе, эти задачи не будут рассматриваться подробно.

# Сопоставление цепочек состояний

Существуют следующие методы для решения данной задачи:

* Методы, основанные на динамическом программировании
  + DTW
* Методы, основанные на скрытых марковских сетях
  + Алгоритм Витерби

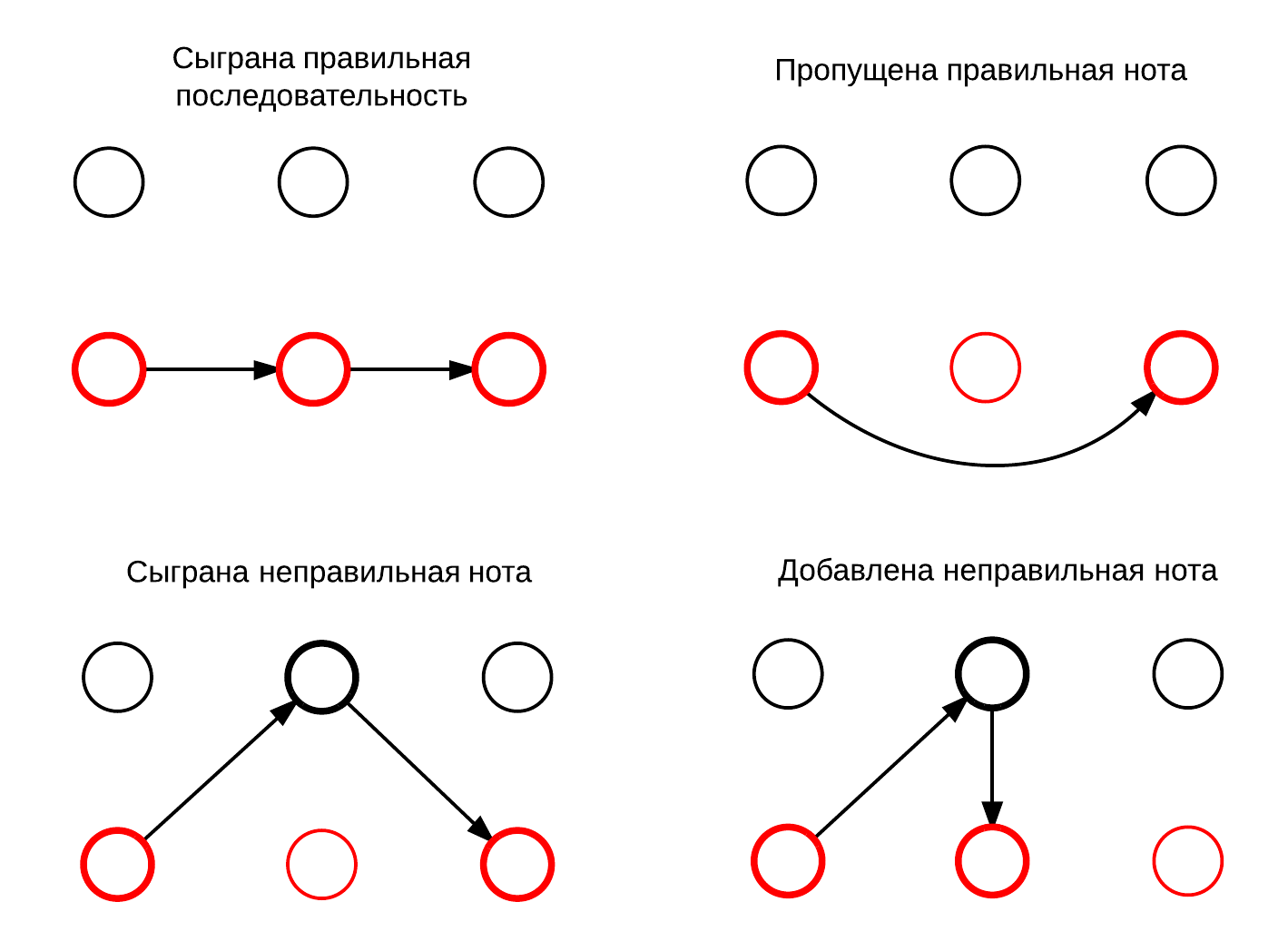


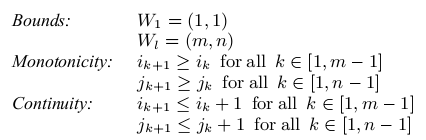
Рисунок . Различные варианты исполнения мелодии в контексте HMM.

## Динамическая трансформация шкалы времени

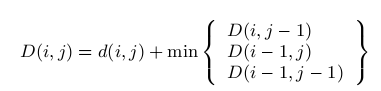
Алгоритм динамической трансформации шкалы времени находит оптимальное выравнивание двух временных последовательностей, но этого не достаточно для приложений в реальном времени, так как он требует сразу две цепочки последовательностей.

DTW алгоритм сопоставляет временные последовательности и , отыскивая минимальный по стоимости путь , где каждое – упорядоченная пара , такая что означает что точки и являются выровненными. Выравнивание определяется с помощью функции локальных стоимостей , которая обычно представляется в виде матрицы, которая назначает соответствующий вес для выравнивания каждой паре . Стоимость является нулевой для идеального совпадения и положительной в другом случае. Стоимость пути – это сумма совпадений локальных весов на продолжении пути.

Несколько ограничений накладываются на , которая обозначает путь, ограниченный концами двух последовательностей. Он является продолженным. Формально это выглядит так:



Минимальная стоимость пути может быть рассчитана за квадратичное время с использованием динамического программирования, используя рекурсию:

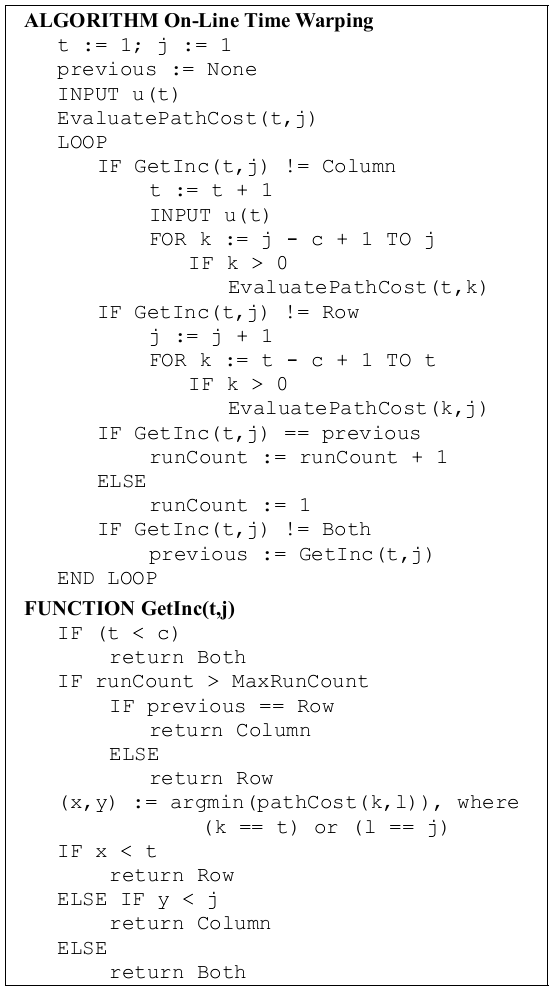


В этой формуле – стоимость минимального пути из в и . Таким образом, путь может быть получен с помощью обратной рекурсии из .

Тем не менее, в случае выравнивания в реальном времени с частично неизвестными последовательностями, несколько изменений должны быть внесены в DTW алгоритм. В стандартном DTW, длины последовательностей предоставляют одно из граничных условий поиска. В случае реального времени, это условие должно быть оценено вдоль оптимального пути. Вследствие этого, диагонали матрицы стоимостей неизвестны, таким образом, ограничения глобального пути не могут быть реализованы напрямую. Другое изменение которое нужно привнести состоит в том, чтобы минимальный по стоимости путь должен быть рассчитан только в прямом направлении. Далее, для того чтобы запустить в реальном времени с продолжительными длинными последовательностями, полный алгоритм должен быть линеен по всей длине последовательности, таким образом инкрементальный шаг должен быть постоянным.

Допустим что частично неизвестная последовательность. Тогда для каждого (которое измеряется в целых единицах, соответствующих индексам ), нужно найти лучшее выравнивание последовательности к другой начальной подпоследовательности . Данное решение удовлетворяет условием выше и алгоритму DTW в офлайн режиме. Переменные и указывают на текущие позиции в последовательностях и соответственно, которые инициализировались точкой старта каждой серии.

Главный цикл алгоритма вычисляет некоторые строки и столбцы пути матрицы стоимостей. В вычислениях обычно задействована стандартная рекурсивная формула DTW алгоритма, ограниченная значениями матрицы, которые уже были вычислены. Стоимость пути нормализуется длинной пути, таким образом, длины меняющихся длин могут быть сравнены в функции . Количество рассчитанных ячеек определяются параметром . Если новая строка становится вычисленной, номер ряда инкрементируется и ячейки в последних колонках, включая, рассчитываются.



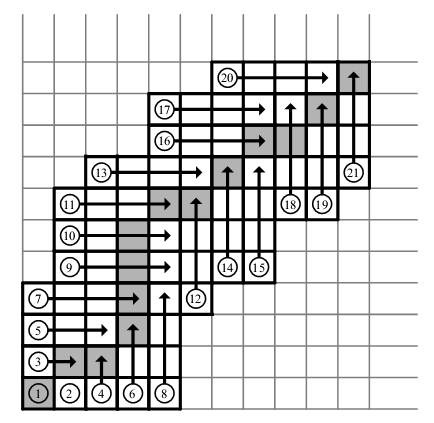


Рисунок . Пример работы алгоритма DTW с окном с = 4. Показан порядок вычисления определенной последовательности с инкрементами строк и столбцов. Оси представляют переменные t и j. Оптимальный путь окрашен в серый цвет.

## Алгоритм Витерби

Алгоритм делает несколько предположений:

* Наблюдаемые и скрытые события должны быть последовательностью. Последовательность чаще всего упорядочена по времени.
* Две последовательности должны быть выровнены: каждое наблюдаемое событие должно соответствовать ровно одному скрытому событию
* Вычисление наиболее вероятной скрытой последовательности до момента t должно зависеть только от наблюдаемого события в момент времени t, и наиболее вероятной последовательности до момента t − 1.

# Определение и предсказания темпа исполнения.

Существуют следующие методы для решения данной задачи:

* Метод скользящей средней
* Фильтр Калмана

## Метод скользящей средней

Скользя́щая сре́дняя, скользя́щее сре́днее (англ. moving average, англ. MA) — общее название для семейства функций, значения которых в каждой точке определения равны среднему значению исходной функции за предыдущий период. Скользящие средние обычно используются с данными временных рядов для сглаживания краткосрочных колебаний и выделения основных тенденций или циклов[1][2]. Математически скользящее среднее является одним из видов свёртки, и поэтому его можно рассматривать как фильтр нижних частот, используемых в обработке сигналов.

## Фильтр Калмана

Фальтр Калмана — эффективный рекурсивный фильтр, оценивающий вектор состояния.

В большинстве приложений количество параметров, задающих состояние объекта, больше, чем количество наблюдаемых параметров, доступных для измерения. При помощи модели объекта по ряду доступных измерений фильтр Калмана позволяет получить оценку внутреннего состояния.

Фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной динамической системы, то есть для расчёта текущего состояния системы необходимо знать текущее измерение, а также предыдущее состояние самого фильтра. Таким образом, фильтр Калмана, как и множество других рекурсивных фильтров, реализован во временном, а не в частотном представлении.

Наглядный пример возможностей фильтра — получение точных, непрерывно обновляемых оценок положения и скорости некоторого объекта по результатам временного ряда неточных измерений его местоположения. Например, в радиолокации стоит задача сопровождения цели, определения её местоположения, скорости и ускорения, при этом результаты измерений поступают постепенно и сильно зашумлены. Фильтр Калмана использует вероятностную модель динамики цели, задающую тип вероятного движения объекта, что позволяет снизить воздействие шума и получить хорошие оценки положения объекта в настоящий, будущий или прошедший момент времени.

# Заключение

Исходя из вышеизложенного, можно сделать вывод, что проблема синхронизации живой музыки и музыкальных записей существует и не является тривиальной. В рамках данной работы были рассмотрены этапы, которые необходимо разрешить для реализации системы отслеживания нот в реальном времени, а также подробно рассмотрены методы синхронизации временных рядов.

# Список используемых источников

1. Arshia Cont, «A coupled duration-focused architecture for realtime music to score alignment», 2010.
2. Simon Dixon, «Live tracking of musical performances using on-line time warping », 2005.
3. J. D. Ferguson, “Variable duration models for speech,” in Symposium on the Applications of Hidden Markov Models to text and Speech, Princeton, New Jersey, October 1980, pp. 143–179.
4. Arshia Cont, «On the creative use of score following and its impact on research», 2011.
5. Nicola Montecchio, Arshia Cont, «A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques», 2011.