

Разработка методики оценки степени синхронизированности нотных записей и живой музыки

Магистерская диссертация

Студент: Кочуркин Иван Алексеевич
Руководитель: Филиппов Михаил Владимирович

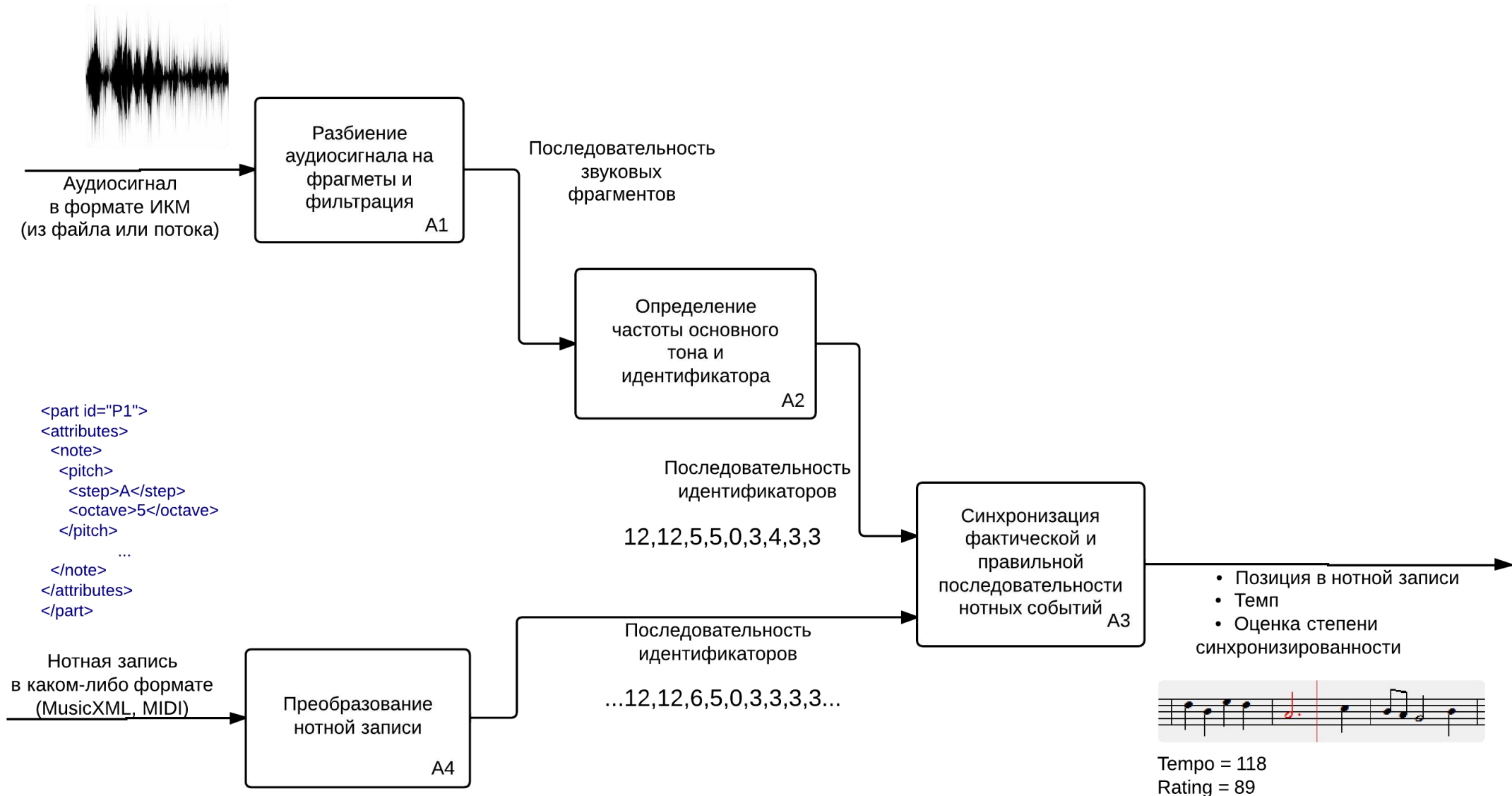
Цель:

Разработка и реализация методики оценки синхронизации нотных записей и музыки в реальном времени.

Задачи:

- Анализ алгоритмов определения частоты основного тона.
- Анализ алгоритмов синхронизации временных рядов.
- Разработка алгоритма синхронизации нотных записей в реальном времени.
- Разработка алгоритма оценивания темпа исполнения.
- Разработка методики оценки степени синхронизации.
- Реализация программного комплекса.
- Исследование разработанных алгоритмов.

Синхронизация нотных записей и музыки



Алгоритмы определения частоты основного тона

Частота основного тона – наименьшая частота колебаний квазипериодического сигнала.

Алгоритм	Достоинства	Недостатки
Временная область		
Пересечение с нулем (ZCR)	<ul style="list-style-type: none">• Простота реализации• Высокая скорость работы	<ul style="list-style-type: none">• Неправильная работа в случае наличия высокочастотных спектральных компонент
Автокорреляция	<ul style="list-style-type: none">• Простота реализации	<ul style="list-style-type: none">• Ограниченный набор частот• Скорость работы
YIN	<ul style="list-style-type: none">• Большой диапазон частот	<ul style="list-style-type: none">• Вычислительная сложность
Частотная область		
Гармоническое перемножение спектров (HPS)	<ul style="list-style-type: none">• Плохое качество работы на низких частотах	<ul style="list-style-type: none">• Требуется спектр сигнала

Модифицированный алгоритм определения частоты основного тона

- Основан на автокорреляции.
- Два этапа:
 - Тест фрагмента низкого разрешения (используется каждый 8 отсчет).
 - Тест фрагмента высокого разрешения (используется фрагмент в окрестности частоты, определенной на 1 этапе).
- Интерполяция Хермита 3-го порядка для минимизации ошибок высоких частот (так как индексы отсчетов не ограничены целыми числами).
- Вычисление номера ноты: $MidiNote = \log_{\sqrt[12]{2}}\left(\frac{F}{F_0}\right)$
 - F - частота основного тона.
 - F_0 - частота ноты с номером 0.

Алгоритмы сопоставления временных последовательностей

Алгоритм	Достоинства	Недостатки
Сопоставление строк (String Matching)	<ul style="list-style-type: none">• Простота реализации.	<ul style="list-style-type: none">• Невозможность возврата в предыдущее состояние.• Сложность применения в системах реального времени.• Сложность гибкой настройки.
Динамическая трансформация времени (DTW)	<ul style="list-style-type: none">• Не требуется определение частоты основного тона.	<ul style="list-style-type: none">• Сложность применения в системах реального времени.
Алгоритм Витерби на Скрытой Марковской Модели (HMM)	<ul style="list-style-type: none">• Большое количество исследований и применений в различных областях.• Возможность применения алгоритмов обучения и гибкой настройки.	<ul style="list-style-type: none">• Потребление большого количества памяти без оптимизаций.

Использование СММ для нотных записей

СММ:

- Алгоритм прямого-обратного хода (оценка).
- Алгоритм Витерби (поиск скрытых состояний).
- Алгоритм Баума-Велша (обучение).

Музыкальные записи как события СММ:

- Ноты.
- Часть ноты.
- ADSR-кривые.

Решенная задача: подбор и генерация коэффициентов в матрицах A , B , π .

СММ: $\{N, M, A, B, \pi\}$

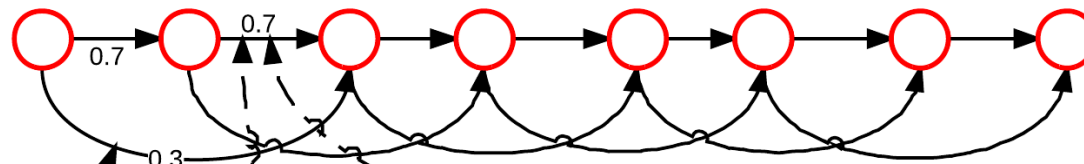
$N = 8; M = 13;$

$\pi = (0.75 \quad 0.05 \quad 0.04 \quad 0.03 \quad 0.02 \quad 0.007 \quad 0.003 \quad 0.0)$

120



time, $t =$ 1 2 3 4 5 6 7 8



$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} & \begin{pmatrix} \cdot & & & & & & & \\ & \cdot & & & & & & \\ & & \cdot & & & & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & & \cdot & & \\ & & & & & & \cdot & \\ & & & & & & & \cdot \end{pmatrix} \end{pmatrix}$$

$a_{13} = 0.3$
 $a_{23} = 0.7$

$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} & \begin{pmatrix} \cdot & & & & & & & & & & & & \\ & \cdot & & & & & & & & & & & \\ & & \cdot & & & & & & & & & & \\ & & & \cdot & & & & & & & & & \\ & & & & \cdot & & & & & & & & \\ & & & & & \cdot & & & & & & & \\ & & & & & & \cdot & & & & & & \\ & & & & & & & \cdot & & & & & \\ & & & & & & & & \cdot & & & & \\ & & & & & & & & & \cdot & & & \\ & & & & & & & & & & \cdot & & \\ & & & & & & & & & & & \cdot & \\ & & & & & & & & & & & & \cdot \end{pmatrix} \end{pmatrix}$$

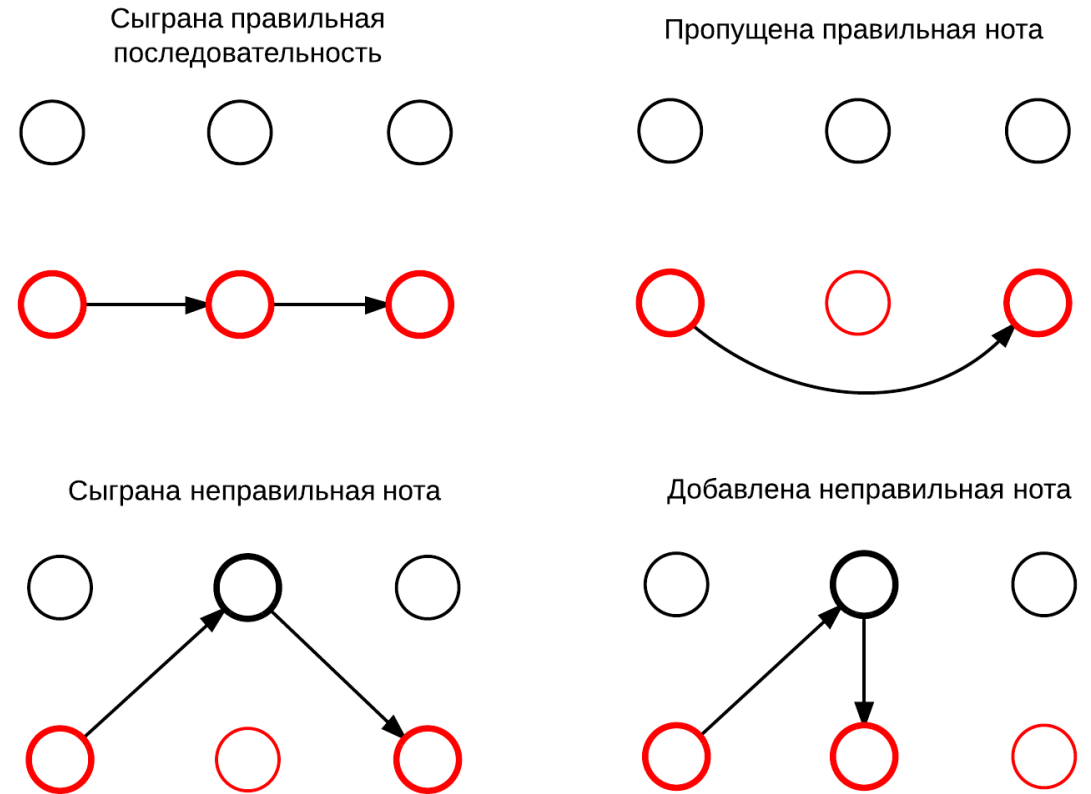
$b_{21} = 0.8 \quad b_{2i} = 0$

Ошибки исполнения и различные длительности нот.

- Виртуальные состояния.
- Ноты с различной длительностью:
 - Использование состояния с самопереходам.
 - Использование скрытых полумарковских моделей.
 - Разбиение всех нотных событий на фрагменты с одинаковой длительностью, *кванты*.

Количество квантов:

$$Count = \left(\sum_{events} event. quarterDur \cdot kvantsInQuater \right) \cdot (VirtualCount + 1)$$



Модифицированный алгоритм Витерби в СММ для работы в реальном времени

Матрица A:

	159	160	161	162	163	164	165	166	167	168	169	170	171
156	0.45	0	0	0	0	0	0	0	0	0	0	0	0
157	0.95	0	0.01	0	0.01	0	0.01	0	0	0	0	0	0
158	0.45	0.1	0.45	0	0	0	0	0	0	0	0	0	0
159	0	0.02	0.95	0	0.01	0	0.01	0	0.01	0	0	0	0
160	0	0	0.45	0.1	0.45	0	0	0	0	0	0	0	0
161	0	0	0	0.02	0.95	0	0.01	0	0.01	0	0.01	0	0
162	0	0	0	0	0.45	0.1	0.45	0	0	0	0	0	0
163	0	0	0	0	0	0.02	0.95	0	0.01	0	0.01	0	0.01
164	0	0	0	0	0	0	0.45	0.1	0.45	0	0	0	0
165	0	0	0	0	0	0	0	0.02	0.95	0	0.01	0	0.01
166	0	0	0	0	0	0	0	0	0.45	0.1	0.45	0	0
167	0	0	0	0	0	0	0	0	0	0.02	0.95	0	0.01
168	0	0	0	0	0	0	0	0	0	0	0.45	0.1	0.45
169	0	0	0	0	0	0	0	0	0	0	0	0.02	0.95
170	0	0	0	0	0	0	0	0	0	0	0	0	0.45

$N = 10; M = 3; P = 4$

- Идентификатор ноты в матрице B:

$$HmmlId = \begin{cases} 12, & MidlId \leq 0 \\ MidlId \% 12, & MidlId > 0 \end{cases}$$
- Использование скользящей матрицы начальных вероятностей (pi).
- Распараллеливание.
- Сложность алгоритма: $O(\frac{M \cdot N^2}{P})$
 - N – размер окна
 - M – рассматриваемые наблюдения ($P \geq N \cdot 4$)
 - P – количество потоков.
- Не зависит от глобального пути.

Алгоритм отслеживания ритма в реальном времени

Существующие методы неприменимы для задач реального времени на среднестатистическом компьютере пользователя.

Шаги:

- Замер времени последних N не гостевых событий в возрастающей последовательности (t_n)
- Измерение разницы между каждыми соседними двумя полученными событиями (d_n)
- Удаление 2 наименьших или наибольших значений (d_k)
- Измерение времени между полученными 6 величинами последующей формуле:

$$d_{mean} = \frac{\sum d_k}{K}$$

$$Tempo = \frac{60 \cdot Kvant Note Value}{d_{mean}}$$

Разработанный алгоритм показал незначительно худшие результаты по сравнению с последними достижениями (Диксон, 2006).

$$t_1 t_3 t_2 t_7 t_5 t_9 t_{11} t_{12} t_{13} t_{15} t_{17} \\ \rightarrow t_1 t_3 t_7 t_9 t_{11} t_{13} t_{15} t_{17}$$



$$t_1 t_3 t_7 t_9 t_{11} t_{13} t_{15} t_{17} \rightarrow \\ d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7 \in D_0$$



$$D_1: d \in D_0, \\ d \neq \max(D_0), d \neq \min(D_0)$$



$$Tempo = \frac{60 \cdot 0.5 \cdot 6}{d_0 + d_1 + d_2 + d_3 + d_4 + d_5}$$

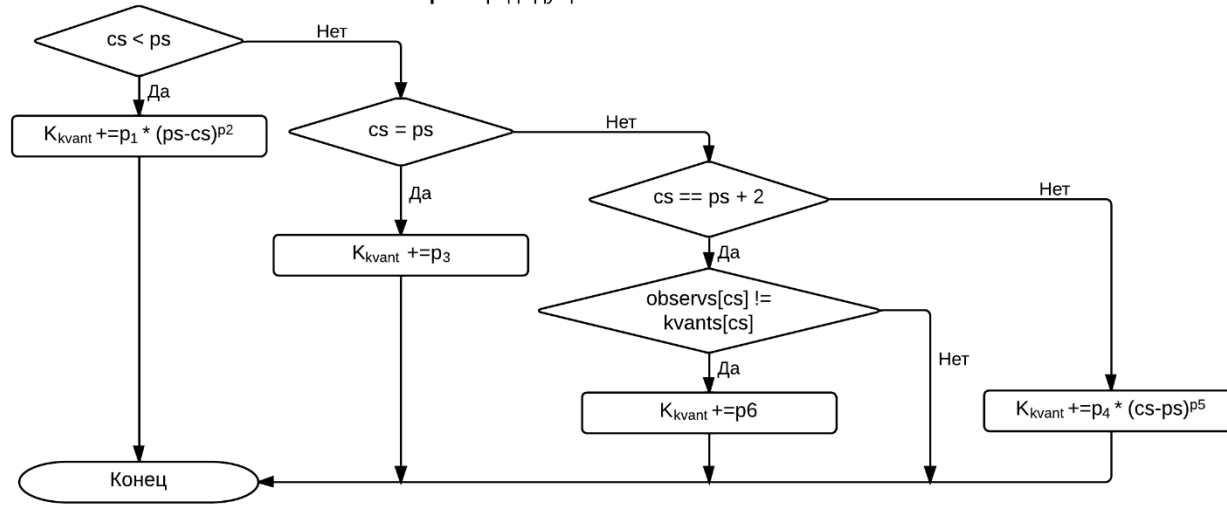
Методика оценки степени синхронизации

- Требования к методике:
 - Учет позиционных ошибок исполнения.
 - Учет темпа исполнения.
 - Учет ADSR кривых звуковых сигналов (без расширения матрицы **B**):
 - Залигованные ноты.
 - Октавные ошибки.
 - Другие музыкальные орнаменты.
 - Функционирование в реальном времени.
- Недостатки алгоритма прямого-обратного хода в качестве позиционного критерия:
 - Не учитываются виртуальные состояния.
 - Возможность оценки только локальной рассматриваемой цепочки наблюдений.
 - Невозможность учета музыкальных событий (а не квантов).

Методика оценки степени синхронизации

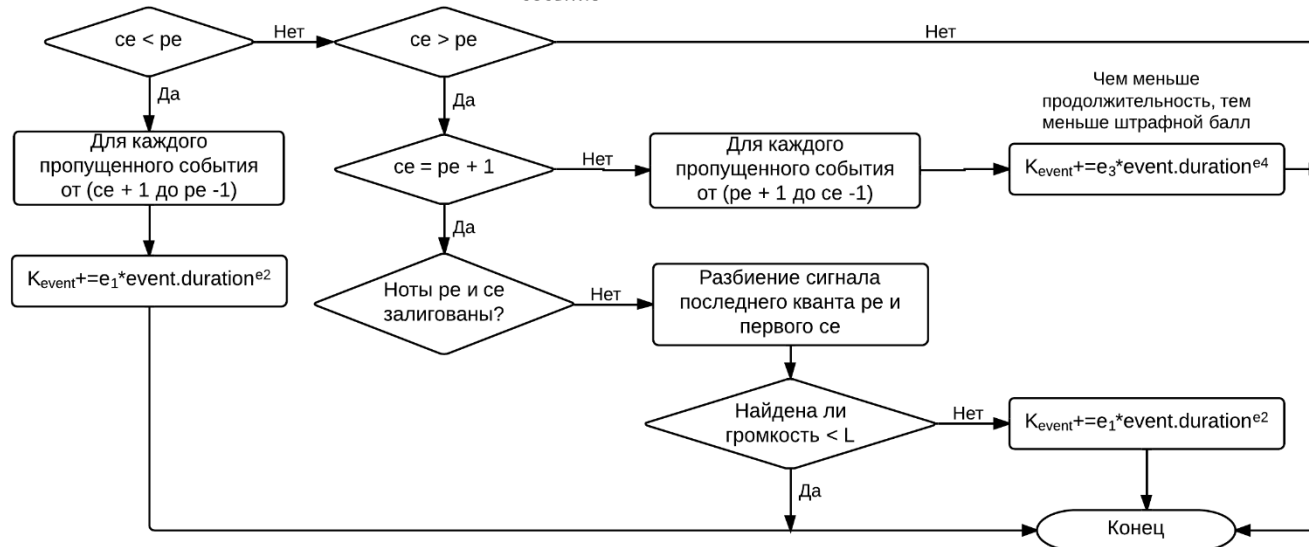
Позиция в нотах (для квантов)

cs - Текущее состояние в CMM
ps - Предыдущее состояние в CMM



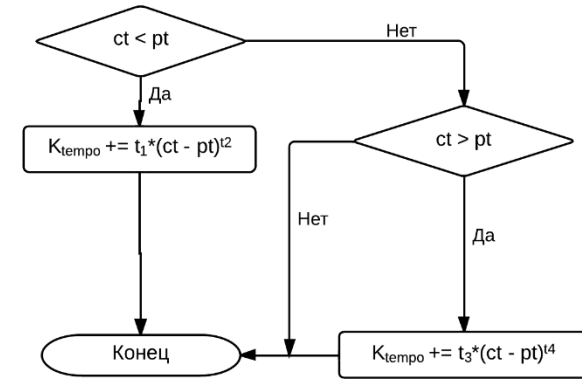
Позиция в нотах (для нотных событий)

се - Текущее музыкальное событие
ре - Предыдущее музыкальное событие



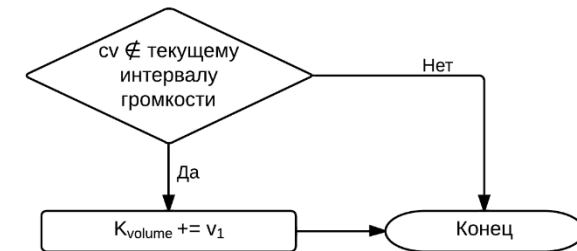
Темп

ct - текущий темп
pt - предыдущий темп



Громкость

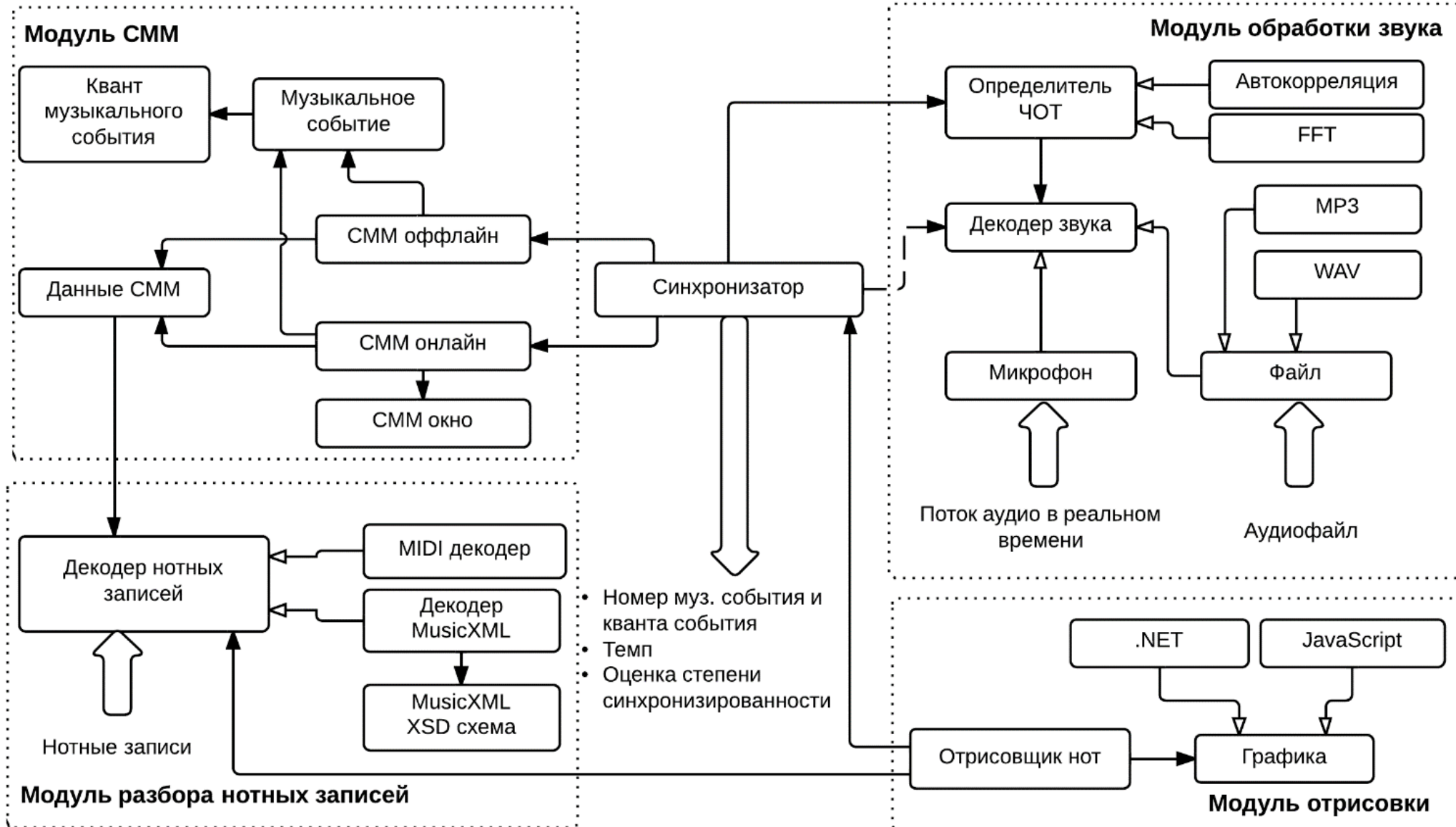
cv - Текущая громкость



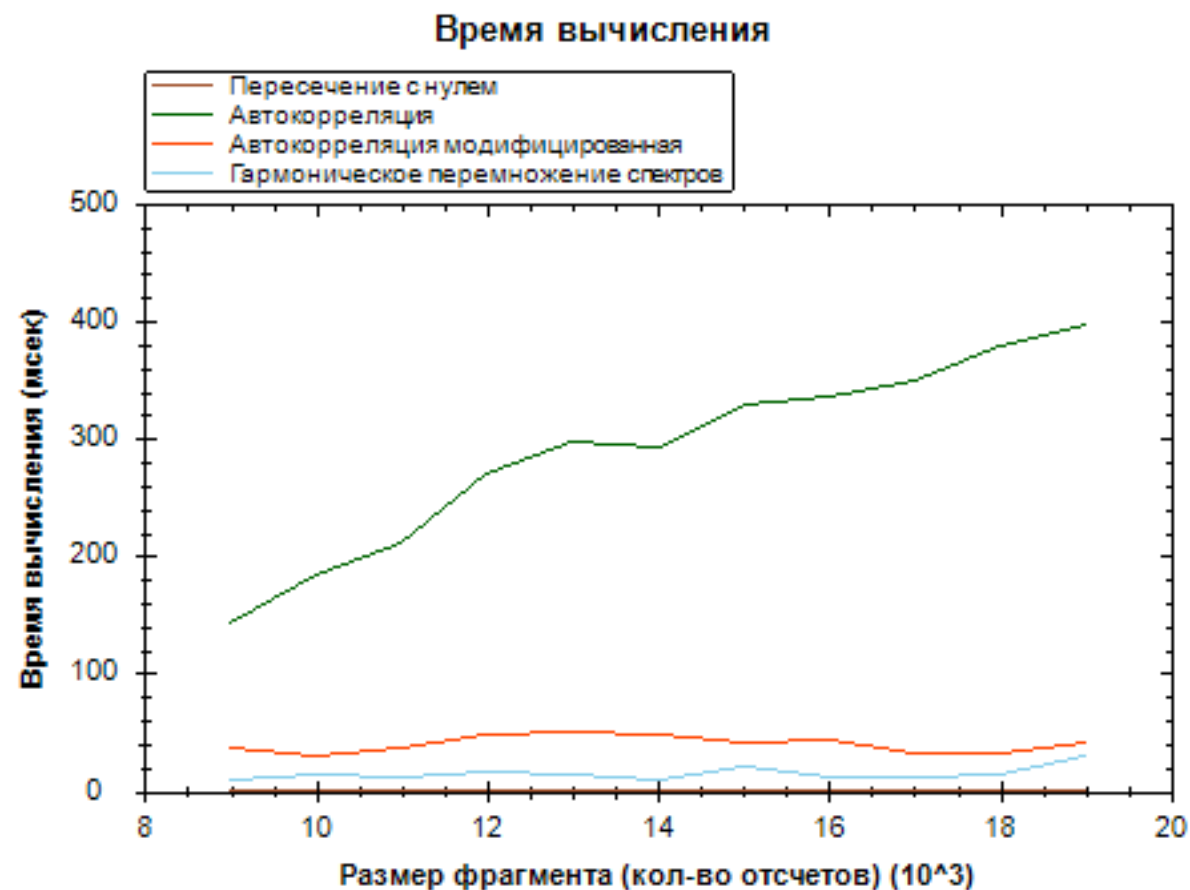
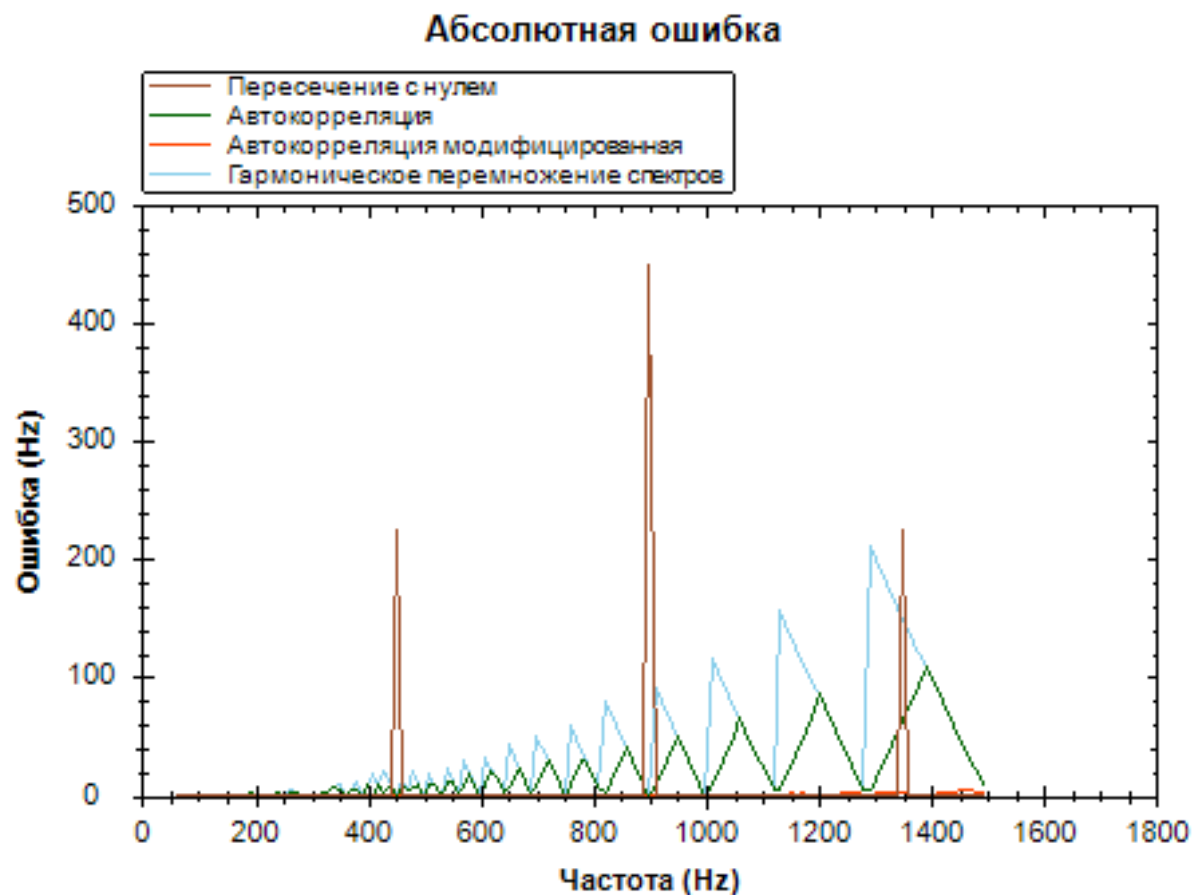
Итоговая формула оценки:

$$K = k_1 * K_{kvant} + k_2 * K_{event} + k_3 * K_{tempo} + k_4 * K_{volume}$$

Структура программной системы



Тестирование алгоритмов определения частоты основного тона



Тестирование алгоритма Витерби в реальном времени



Тестирование метода синхронизации нот

- Тестовые мелодии:
 - Twinkle Twinkle Little Star
 - All I Ask of You
 - On a High Mountain
- Простые мелодии:
 - Высокая точность (параметр соответствует субъективной оценке).
 - Отсутствуют задержки.
- Сложные (с большим количеством орнаментов) или продолжительные мелодии:
 - Довольно плохая точность (штрафной балл не всегда соответствует субъективной оценке).
 - Появляются задержки синхронизации.
 - Большое количество **повторяющихся** или **похожих** частей может являться проблемой.

Заключение и выводы

- Проанализированы алгоритмы определения частоты основного тона и реализован модифицированный метод автокорреляции.
- Проанализированы алгоритмы сопоставления временных рядов и реализован алгоритм, основанный на алгоритме Витерби для синхронизации нотных записей в реальном времени.
- Разработан алгоритм определения ритма.
- Разработана методика оценки степени синхронизации нотных записей и музыки в реальном времени.
- Реализовано кроссплатформенное ПО.
- Исследованы реализованные алгоритмы.
- По технологической части работы опубликованы 2 статьи.