УДК 004

Реализация алгоритмов цифровой обработки сигналов в браузерном тюнере для струнных инструментов в парадигме Zero-Dependency методами промпт-инжиниринга

Пайдершев Алексей Константинович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В статье исследуется применение больших языковых моделей для генерации кода, реализующего алгоритмы цифровой обработки сигналов в реальном времени. В качестве объекта исследования выступает универсальный браузерный тюнер для струнных инструментов, разработанный в парадигме Zero-Dependency на базе нативного Web Audio API. В отличие от типовых задач веб-разработки, данный проект требует не только построения пользовательского интерфейса, но и программной реализации математического аппарата: алгоритма автокорреляции для выделения основной частоты, вычисления логарифмических отклонений в центах и физического моделирования звука методом Карплуса-Стронга. Особое внимание уделено вопросам фильтрации фонового шума. Показано, что хотя нейросеть успешно транслирует строгие математические формулы в код, обеспечение устойчивости работы в реальных акустических условиях требует от разработчика точной формулировки эмпирических пороговых значений в тексте промпта.
Ключевые слова: Qwen Studio, нейросети, промпт-инжиниринг, вайб-кодинг, Zero-Dependency, Web Audio API.

Implementation of digital signal processing algorithms in a browser-based tuner for string instruments using the Zero-Dependency paradigm and prompt engineering methods

Paysershev Alexey Konstantinovich
Sholom-Aleichem Priamursky State University
Student

Abstract
The article examines the use of large language models to generate code that implements digital signal processing (DSP) algorithms in real time. The object of the study is a universal browser-based tuner for string instruments, developed in the Zero-Dependency paradigm based on the native Web Audio API. Unlike typical web development tasks, this project requires not only the creation of a user interface but also the software implementation of a mathematical framework: an autocorrelation algorithm for extracting the fundamental frequency, calculating logarithmic

deviations in cents, and physically modeling sound using the Karplus–Strong method. Special attention is paid to issues related to background noise filtering. It has been shown that although the neural network successfully translates strict mathematical formulas into code, ensuring stable operation in real acoustic conditions requires the developer to precisely formulate empirical threshold values in the prompt text.

Keywords: Qwen Studio, neural networks, prompt engineering, vibe coding, Zero-Dependency, Web Audio API.

1 Введение

1.1 Актуальность

Разработка программ для работы со звуком в режиме реального времени традиционно относится к числу ресурсоемких задач, требующих применения специализированных библиотек для цифровой обработки сигналов. Однако современные браузерные технологии, в частности Web Audio API, предоставляют разработчику низкоуровневый доступ к аудиопотоку, позволяя реализовывать сложные математические алгоритмы без привлечения сторонних зависимостей.

Создание музыкального тюнера в парадигме Zero-Dependency сопряжено с рядом специфических трудностей. Во-первых, необходимо реализовать надежный алгоритм выделения основной частоты из смешанного аудиосигнала. Во-вторых, требуется обеспечить обратную связь – воспроизведение эталонных звуков, которые должны быть акустически достоверными. В-третьих, приложение обязано игнорировать фоновый шум, который неизбежно улавливается микрофоном в реальных условиях.

Актуальность данного исследования заключается в проверке того, насколько большие языковые модели способны генерировать код для решения задач цифровой обработки сигналов. Если генерация интерфейсов для ИИ является стандартной задачей, то реализация алгоритмов физического моделирования звука и математической фильтрации шума требует от модели понимания специфических физических и математических закономерностей.

1.2 Обзор исследований

Согласно работам Д. С. Пащенко, применение инструментов на основе искусственного интеллекта превращается в важнейшее условие сохранения конкурентоспособности ИТ-компаний, поскольку даёт возможность значительно ускорить разработку и автоматизировать повторяющиеся операции [1]. С. В. Рогатин и В. А. Антошкин доказывают принципиальную возможность разработки сложных интерактивных приложений, используя только встроенные средства веб-технологий и не привлекая сторонние зависимости [2]. Д. Р. Мухаметов, изучающий воздействие искусственного интеллекта на создание политических образов, подтверждает эффективность генеративных моделей в узкоспециализированных предметных сферах и творческих отраслях [3]. Тем не менее, как показывает И. В. Елманов, автоматическое создание кода нуждается в жёстком контроле; для этого он

предлагает применять методологию Spec Driven Development, обеспечивающую надёжную проверку программ, разработанных в рамках вайб-кодинга [4]. А. М. Титеев на практике показывает, что большие языковые модели успешно справляются со сложной организацией веб-приложений: он применяет их для ускорения разработки автотестов и разбора HTML-структуры [5].

1.3 Цель исследования

Цель работы – оценить способность большой языковой модели (на примере Qwen Studio) спроектировать универсальный браузерный тюнер для струнных инструментов в рамках парадигмы Zero-Dependency, а также проанализировать эффективность промпт-инжиниринга при настройке алгоритмов цифровой обработки сигналов и фильтрации шума.

2 Материалы и методы

Методология исследования базируется на гибридном подходе, объединяющем структурированный промпт-инжиниринг и отказ от традиционных методов веб-разработки в пользу процедурной генерации. В качестве жесткого архитектурного ограничения утверждается применение исключительно нативных HTML, CSS и JavaScript с Web Audio API. Всё приложение должно размещаться в едином файле index.html.

В качестве платформы для написания кода применяется среда Qwen Studio. Ключевой особенностью методологии является итеративный подход: первоначальная генерация рассматривается как прототип, который последовательно дорабатывается через серию уточняющих промптов.

Тюнер должен безошибочно определять ноту и отклонение в центах при подаче эталонного сигнала, игнорируя при этом человеческую речь и фоновый гул.

3 Результаты

Первоначальная задача – создание ядра приложения, способного считывать данные с микрофона и преобразовывать их в музыкальную ноту.

Промпт для генерации: «Действуй как Senior Frontend разработчик и специалист по цифровой обработке аудиосигналов. Твоя задача – написать полный, готовый к запуску код веб-приложения «Хроматический гитарный тюнер». Приложение должно быть реализовано в одном файле и работать локально в браузере. Никаких сторонних библиотек и аудио-файлов – только нативные HTML, CSS и JavaScript с Web Audio API.

Архитектурные требования к аудио:

– Реализуй алгоритм автокорреляции для точного определения основной частоты сигнала в диапазоне от 70 Гц до 350 Гц.

– Вычисляй ближайшую ноту хроматического звукоряда и отклонение от её точной частоты в центах (1/100 полутона).

– Если отклонение менее ±5 центов, нота считается «в строе» (визуально выделяется зелёным цветом).

Требования к интерфейсу:

– Тёмная тема, панель выбора гитарных струн (E2, A2, D3, G3, B3, E4), крупный индикатор ноты и визуальная шкала отклонения.

Выведи полный код в одном блоке».

Результат генерации: https://disk.yandex.ru/d/cDcvp5aT_IgYzg

Нейросеть успешно справилась с задачей, реализовав классический алгоритм автокорреляции, который ищет периодичность сигнала во временной области. Математический аппарат вычисления центов также реализован без ошибок.

Сгенерированное приложение обладает минималистичным интерфейсом (рис. 1). Вверху находится динамическая сетка кнопок, состав которой опирается на название нот.

Центральная часть экрана отведена под блок отображения результата. Он содержит крупный индикатор ноты, текущую частоту в герцах и визуальную шкалу отклонения. Шкала представляет собой горизонтальную полосу со стрелкой: при точном попадании в ноту (отклонение менее ±5 центов) индикаторы окрашиваются в зелёный цвет (рис. 2). Также присутствует кнопка управления микрофоном и строка статуса.



Рисунок 1. Интерфейс приложения

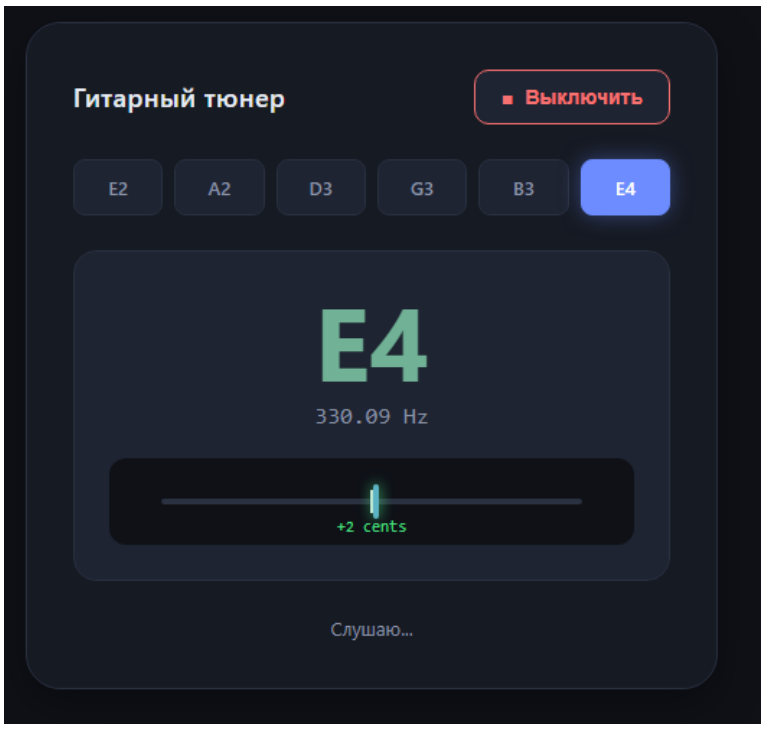


Рисунок 2. Индикатор точного попадания в ноту

Следующий этап – добавление звуковой обратной связи.

Промт для генерации: «Добавь следующее:

– При нажатии на кнопку выбора струны должен проигрываться кратковременный эталонный звук этой ноты.

– Звук должен быть реалистичным, как звук щипковой гитарной струны.

– Для синтеза используй алгоритм Карплуса-Стронга (физическое моделирование струны).

– Для басовых струн (частота ниже 100 Гц) используй более высокий коэффициент затухания, чтобы звук тянулся дольше».

Результат генерации: https://disk.yandex.ru/d/pv0HtvJKOvzf7g

ИИ успешно реализовал алгоритм, который имитирует физику колеблющейся струны. Генерация начального белого шума имитирует момент щипка, а рекурсивный фильтр низких частот создает гармонические обертоны, плавно затухающие во времени.

Следующий этап – расширение функционала для поддержки бас-гитары, укулеле и хроматического режима.

Промпт для доработки: «Добавь следующее:

– Возможность выбора инструмента: Гитара, Бас-гитара, Укулеле, а также режим «Все ноты» (хроматический).

– Интерфейс должен динамически перестраиваться под выбранный инструмент.

– Для бас-гитары используй стандартный строй: E1 (41.20 Гц), A1 (55.00 Гц), D2 (73.42 Гц), G2 (98.00 Гц).

– Для укулеле: G4, C4, E4, A4.

– В режиме «Все ноты» отображай 12 кнопок хроматического звукоряда. При нажатии на них проигрывай звук синтезатора, а для струнных инструментов оставь звук щипковой струны.

– Обнови алгоритм автокорреляции: опусти нижний порог определения частоты до 30 Гц, чтобы тюнер корректно распознавал басовые струны».

Результат генерации: https://disk.yandex.ru/d/4brHD77TvfZ6_w

Теперь в интерфейс приложения добавились кнопки с выбором инструмента, при нажатии на которые меняется сетка кнопок с нотами в соответствии с выбранным инструментом (рис. 3-5).



Рисунок 3. Режим «Бас»



Рисунок 4. Режим «Укулеле»



Рисунок 5. Режим «Все ноты»

Внедрение пресетов потребовало изменения логики расчета: в режиме конкретной струны тюнер сравнивает входящий сигнал с фиксированной частотой, а в хроматическом режиме – ищет ближайшую ноту по всей сетке.

Заключительный этап – борьба с ложными срабатываниями. В реальных условиях микрофон улавливает человеческую речь и фоновый гул, которые алгоритм автокорреляции может ошибочно принять за музыкальный тон.

Промпт для отладки: «Исправь следующее:

Приложение реагирует на обычный шум (разговоры, шорохи, гул вентилятора). Тюнер должен срабатывать только на чёткий музыкальный тон.

Добавь следующее:

– Подними порог среднеквадратичного отклонения до 0.02, чтобы отсекать тихий фоновый шум.

– Реализуй проверку ясности: после нахождения пика автокорреляции вычисляй отношение пика к нулевому лагу (c[T0] / c[0]). Если значение меньше 0.75, сигнал считается шумом или речью, и функция должна возвращать -1 (игнорировать сигнал).

– Добавь сглаживание частоты: заведи массив, в который при каждом кадре добавляется найденная частота (не более 3 последних значений). Итоговую частоту для отображения считай как среднее арифметическое. Это уберёт дёргание стрелки».

Результат генерации: https://disk.yandex.ru/d/YSaWFTmHXdxYxQ

Реализация трехуровневой фильтрации полностью решила проблему. Приложение отсекает тихие звуки, проверка ясности отличает периодический музыкальный сигнал от непериодического шума, а скользящее среднее обеспечивает плавность анимации.

4 Вывод

Проведенное исследование демонстрирует, что большие языковые модели обладают значительным потенциалом в области генерации кода для цифровой обработки сигналов. Эксперимент по созданию универсального тюнера выявил несколько важных закономерностей.

ИИ успешно справляется со строгими математическими и физическими алгоритмами. Реализация автокорреляции, расчет центов по логарифмической шкале и синтез звука методом Карплуса-Стронга выполнены корректно. Нейросеть способна понимать терминологию из области акустики и переводить её в программные константы.

Опыт показал, что для обеспечения устойчивости работы в реальных условиях от разработчика требуется точная формулировка эмпирических параметров. Нейросеть не может самостоятельно догадаться, какой именно порог ясности использовать для отсечения речи, однако она безупречно интегрирует в код значения, указанные в промпте.

Разработанное приложение полностью соответствует парадигме Zero-Dependency. Оно работает локально в браузере, не требует установки сторонних библиотек и использует только нативные API, что делает его универсальным и кроссплатформенным инструментом для музыкантов.

Методология промпт-инжиниринга доказала свою эффективность при создании узкоспециализированных инженерных инструментов.

Библиографический список

1. Пащенко Д. С. Использование ИИ-инструментов в конкурентном развитии ИТ-компаний: прикладное исследование 2026 //Информатика. Экономика. Управление-Informatics. Economics. Management. 2026. Т. 5. №. 3. С. 2012-2024. URL: https://oajiem.com/index.php/24/article/view/297
2. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С.12-16. URL: https://elibrary.ru/item.asp?id=25030492
3. Мухаметов Д. Р. Политический вайб-маркетинг: возможности искусственного интеллекта для продвижения образа будущего //Известия Тульского государственного университета. Гуманитарные науки. 2025. №. 3. С. 27-33. URL: https://cyberleninka.ru/article/n/politicheskiy-vayb-marketing-vozmozhnosti-iskusstvennogo-intellekta-dlya-prodvizheniya-obraza-buduschego
4. Елманов И. В. Инженерный подход к ИИ: использование spec driven development для валидации vibe coding //Вестник науки. 2025. Т. 4. №. 12 (93). С. 716-723. URL: https://cyberleninka.ru/article/n/inzhenernyy-podhod-kii-ispolzovanie-spec-driven-development-dlya-validatsii-vibe-coding
5. Титеев А. М., Оптимизация процесса создания автотестов веб-приложений с использованием LLM и структурного анализа HTML //Моделирование и анализ данных. 2025. Т. 15. №4. С. 87–103. URL: https://psyjournals.ru/journals/mda/archive/2025_n4/Titeev