

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Актуальность исследования	3
Степень научной разработанности проблемы.....	4
Объект исследования	5
Предмет исследования	5
Цель исследования.....	5
Задачи исследования	5
Методологическая основа исследования	5
Теоретическая база исследования	6
Научно-практическая значимость исследования	6
Раздел 1.	7
1.1 Краткая история звукорежиссуры.....	7
1.2 Акустические и физические основы звука.....	9
1.3 Обработка звука.....	14
1.4 История цифрового звука.....	16
1.5 Цифровой звук	18
1.6 Особенности цифровой обработки звука.....	20
1.7 DSP (ЦСП)	24
1.8 VST	25
1.9 Программные эмуляции аналоговых устройств	26
Раздел 2	27

2.1 Обоснование выбора компрессора в качестве тестового плагина ..	27
2.2 Используемые библиотеки и обоснование их использования	27
2.3 Кроссплатформенность	28
2.4 Домены величин.....	29
2.5 Архитектура VST и архитектура плагина	29
2.6 Архитектура компрессора	32
2.6.1 Аттенюатор.....	34
2.6.2 Вычислитель управляющего напряжения	34
2.6.3 Детектор уровня.....	35
2.6.4 Детектор пиков	36
2.6.5 Собственно архитектура	37
2.7 Подготовка к разработке	39
2.8 Разработка.....	40
2.8.1 Добавление шаблона	40
2.8.2 Создание пользовательской логики. Pressor.....	41
2.8.3 Отладка.....	42
2.8.4 Тестирование	42
Заключение.....	46
Список литературы	47
ПРИЛОЖЕНИЕ А.	51
ПРИЛОЖЕНИЕ Б	52

ВВЕДЕНИЕ

Тема: Теоретические и практические основы разработки аудиопрограмм (на примере создания vst компрессора с помощью платформы .Net и звуковых библиотек)

Актуальность исследования

Развитие информационных технологий в 21-м веке принесло значительные изменения в различные сферы деятельности человека. Не обошло оно стороной и профессиональную деятельность звукорежиссеров.

Второе десятилетие 21го века по отношению к звукорежиссуре характеризуется полной и окончательной победой облачных стриминговых сервисов прослушивания музыки и цифровых технологий продакшна. Тренд на автоматизацию затронул рабочий процесс сценических звукорежиссеров, позволив им, например, создавать и сохранять скрипты работы микшерных консолей, задействовать матричную маршрутизацию для тонкого управления звуковой картиной, использовать портативные анализаторы спектра и пр. Современные цифровые усилители имеют функцию изучения амплитудно-частотной и фазово-частотной характеристик аналоговых усилителей в реальном времени и эмулируют их звучание практически неотличимо даже для профессионала.

Проектирование такого уровня цифровых систем - процесс трудоемкий и требующий от инженеров как первоклассного знания программирования (или системной инженерии, если мы говорим об устройствах), но и специфических навыков и знаний, характерных для представителей профессии звукорежиссер, а программистам, работающими со звуком в игровых движках, требуется еще и знание психоакустики. Специалистов подобного профиля найти очень непросто,

спрос на них велик, хоть и не виден стороннему наблюдателю. Зачастую, программисты просто не осведомлены о специфических особенностях звукорежиссерской работы, а звукорежиссеры, в свою очередь, очень слабо представляют себе устройство ПО, с которым работают.

Ввиду узости специальности, найти информацию по ней сложно не только в научных публикациях, но даже просто в интернете.

В связи со всем вышесказанным, данная работа призвана пролить свет на особенности создания звукового программного обеспечения, не всегда достаточно изученные в профессиональной среде звукорежиссеров и наоборот, изложить некоторые особенности специфики звукорежиссерской работы для программистов, а также продемонстрировать непосредственный процесс разработки аудиоплагина.

Степень научной разработанности проблемы

В российской академической среде существует ряд работ прямо или косвенно связанных с данной темой:

Методика обучения основам музыкального программирования, Кибиткина, Элина Вадимовна¹

Структура и содержание музыкальной информатики в системе профессионального музыкального образования, Тереньтев Юрий Юрьевич²

Музыкальные компьютерные технологии как новый инструментарий современного творчества, Пучков, Станислав Владимирович³

Их, однако, недостаточно для получения полноценного знания об изучаемом вопросе, поэтому дополнительно приложены материалы на английском языке, более полно научно прорабатывающие данную тему, а также ряд ссылок на интернет ресурсы более простые в освоении и изучении.

Объект исследования

Объектом исследования данной работы является совокупность знаний, практик и техник разработки аудиопрограмм различной направленности, а также особенности работы программиста-звукорежиссера.

Предмет исследования

Предмет исследования - возможности звуковых библиотек на платформе .NET и их применение в создании аудиопрограмм.

Цель исследования

Цель данного исследования - формирование представления о работе программиста-звукорежиссера и процессе разработки аудиопрограмм.

Задачи исследования

- 1) Изучить исторический аспект звукорежиссуры и обработки аудио
- 2) Исследовать особенности цифрового звука
- 3) Продемонстрировать создание аудиоплагина с помощью выбранных технологий и согласно выявленным требованиям

Методологическая основа исследования

Методологической основой исследования является моделирование процесса создания аудио ПО на примере разработки аудиоплагина, а также анализ требований к разрабатывающему его специалисту, самому ПО и обобщение результатов исследования.

Теоретическая база исследования

Статьи зарубежных и отечественных специалистов в области DSP-программирования (см. Приложение).

Научно-практическая значимость исследования

Научно-практическая значимость результатов данного исследования заключается в возможности его применения для обучения специалистов в области аудиопрограммирования (игровые аудиопрограммисты, разработчики ПО для звукорежиссеров), а также для обучения и повышения компетенции звукорежиссеров.

Достоверность и обоснованность результатов формируются научно-практическим подходом к изучаемой проблеме, методологией работы с материалами, их качественным и количественным анализом, подтверждением исследования результатами процесса создания реально функционирующей программы.

РАЗДЕЛ 1.

1.1 Краткая история звукорежиссуры.

Звукорежиссура, как подсказывает «словарь и справочник по материалам прессы и литературы 70х годов» 1984го года под авторством Н. З. Котеловой⁴, есть творческое руководство и организация процесса записи музыкального, драматургического, литературного произведения, документального, учебного и другого материала для последующего неоднократного воспроизведения, передачи в эфир и хранения.

В наше время этот термин дополняется всеми аспектами цифровой эпохи, аналого-цифровым и цифро-аналоговым преобразованием, кодированием, декодированием, сжатием для передачи по сети, работой в рамках войн громкости и т.д. Однако, прежде чем переходить к непосредственной теме данной работы, полезно было бы освежить историю звукорежиссуры и непосредственно работы со звуком.

Механические звуковоспроизводящие устройства известны с 9-го века н.э. Первое из них - гидравлический орган братьев Бану Муса (рис. 1)⁵, появился задолго даже до простейших шарманок, колесных лир, музыкальных шкатулок и т.п., и был способен воспроизводить различные мелодии, записанные на сменном штампованном цилиндре, однако не способен был записывать произвольные звуки, вроде человеческого голоса. Несмотря на это, стремление создавать самовоспроизводящиеся записи породила множество специфических музыкальных инструментов и в конце концов привела к появлению механических инструментов, вроде пианино, играющего мелодии, записанные на перфокары (рис. 2), а впоследствии - и MIDI-формату в 1982-м году⁶.

История же акустической звукорежиссуры начинается на заре 20го века с появлением простейших механических звукозаписывающих устройств. Первые

навыки звукорежиссуры были получены эмпирически изобретателями и первыми пользователями фоноавтографов, фонографов и после 1887го - граммофонов. Патефон, появившийся в 1907-м году, также относился к механическим звуковоспроизводящим устройствам, и отличался от граммофона более компактной конструкцией, встроенной в переносной корпус⁷.

Работали механические звукозаписывающие устройства очень просто - вибрации воздуха, возникающие при говорении или музыкальном исполнении, усиливались многократными отражениями и резонансом раструба и заставляли двигаться иглу, нарезающую дорожки на мягком материале носителя - цилиндра или диска. До наших дней дошло достаточное количество записей, сделанных на все эти устройства, не смотря на недолговечность самых первых носителей.

В те времена звукорежиссура сводилась к подбору достаточно большого помещения, работе непосредственно со звукозаписывающей техникой, и правильной рассадке музыкантов перед раструбом записывающего граммофона.

Позже, при переходе от механической к электромеханической записи, роль звукорежиссера формируется более отчетливо. Формируется и развивается практика аналоговой записи фонограмм. Качество звука совершенствуется, запись электромеханическим способом на пластинки (примерно 1910-1940), а затем и на магнитную ленту (1940 - 1990) требует дополнительных усилий по работе с микрофонами, уровнями, рассадкой музыкантов и взаимодействия с актерами в театре или на радио. Окончательно практические основы работы со звуком достигаются с появлением многодорожечной записи в середине 1950х и появлением практики сведения фонограмм, поддорожечной обработки записей и усовершенствованием методов многоканальной акустической записи. Современная звукорежиссура, безотносительно технической составляющей, в основном является прямым продолжением практик того времени.

Таким образом, как художественные, так и технические особенности работы звукорежиссера сложились к 90м годам прошлого века, а развившиеся к концу нулевых годов 21-го века цифровые технологии лишь преобразовали их в более удобную и повторяемую форму.

1.2 Акустические и физические основы звука.

Звук — распространение упругих волн механических колебаний в среде и их восприятие органами слуха⁸. Объектом звукорежиссуры как творческой практики является звук, доступный человеческому уху, а потому рассматривать звуковые явления мы будем с позиции восприятия человеком и, так как данная работа подразумевает использование цифровых методов работы со звуком, с позиции цифровой техники, то есть в виде квантов звука - семплов. Процесс формирования звуков показан на рисунке 1.2.1.

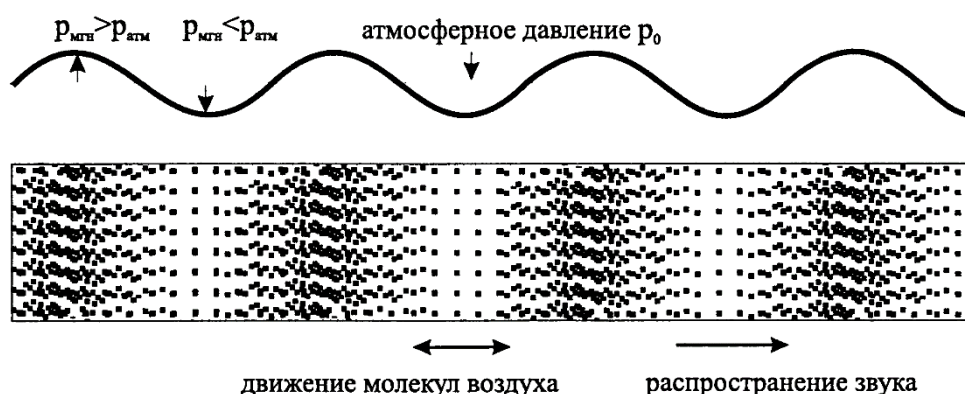


Рисунок 1.2.1. Формирование областей повышенного и пониженного давления воздуха вдоль вектора распространения звука

Звук представляет собой поперечную волну, представляющую собой формируемые источником звука чередования сжатия и разрежения среды, распространяющиеся перпендикулярно направлению распространения. Звук характеризуется **амплитудой** и **частотой**.

Амплитуда представляет собой величину отклонения волны от уровня атмосферного давления и определяет громкость звука (см. рисунок 1.2.2).

Громкость - субъективная величина, она зависит от эффективного звукового давления, частоты и формы колебаний.

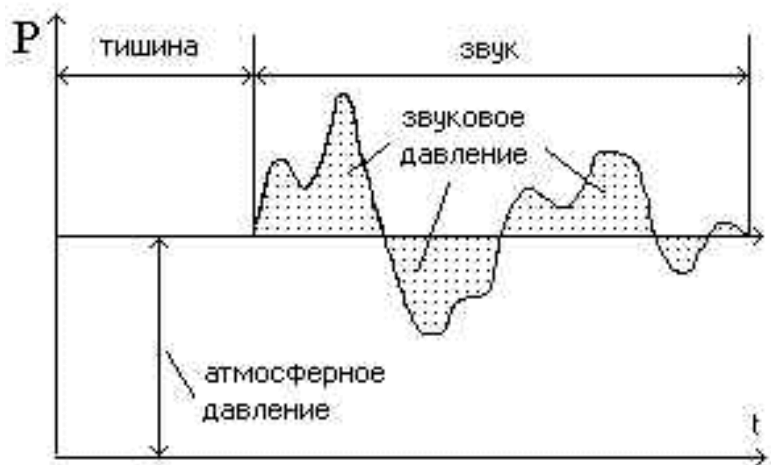


Рисунок 1.2.2. Формирование звука звуковым давлением

Громкость относительна, мы не можем измерить ее абсолютно, лишь посчитать отношение звукового давления (или величину семпла) к некому пороговому уровню. Величиной отношения текущего уровня громкости к минимальному является децибел (dB или дБ) ⁹. Таблица 1.2.1 дает некоторое представление об уровнях громкости некоторых звуков.

Источник звука	Расстояние до источника, м	Звуковое давление, Па	Уровень звукового давления, дБ
Шум самолета	5	200	120
Большой оркестр (<i>ff</i>)	10	2–4	100–106
орган (<i>f</i>)	3,6	2	100
Тарелки	0,9	1,67	99
Камерный ансамбль	4	0,8–1	92–94
Труба	0,9	0,80	92
Кларнет	0,9	0,36	85
Флейта	1	0,063	70
Речь	1	0,02–0,05	60–68
Шепот	1	2×10^{-4}	20
Писк комара	0,5	2×10^{-5}	0

Таблица 1.2.1 Сравнительная таблица уровней звукового давления и громкости различных источников звука

В общем случае громкость L описывается как:

$$L = 20 \lg(p/p_0)$$

Где p - текущий уровень звукового давления (слышимого человеком или записанного и оцифрованного), а p_0 - константа, равная $2 \cdot 10^{-5}$ Па при восприятии звука человеком (среднее значение минимального звукового давления, которое мы можем расслышать) и 0.001 - при обработке цифрового сигнала (значение, меньшее по модулю, будет вызывать выключение аппаратуры).

Частота есть количество колебаний (полных проходов фазы) на единицу времени, она формирует тон, высоту звука. Измеряется в Гц («разы в секунды»).

Обычный человек способен слышать звуковые колебания в диапазоне частот от 16—20 Гц (который в большей степени воспринимается, как чувство давления) до 15—20 кГц (самые тонкие «шипящие» звуки, способность слышать которые уменьшается с возрастом).

Особенность устройства человеческого уха в том, что его чувствительность нелинейна и неодинакова на разных частотах и на разных уровнях звукового давления. Тихий звук мы лучше всего воспринимаем на частотах от 500 до 6000 Гц. Чем выше уровень звукового давления, тем меньше субъективная разница между уровнями слышимости звуков разной частоты.

В процессе экспериментов и измерений были выведены так называемые **кривые равной громкости**¹⁰, описывающие нелинейную характеристику уха (Рис. 1.2.3). Читается она следующим образом: звук частотой 1000 Гц на уровне громкости в 20 дБ будет казаться настолько же громким, как звук с частотой 20 Гц на уровне 80 дБ.

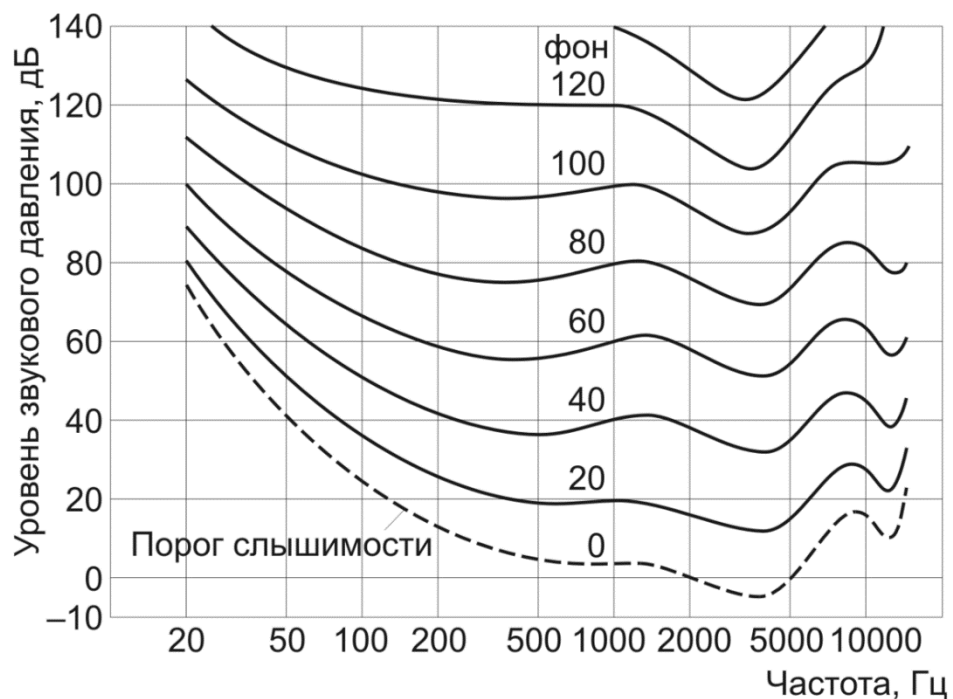


Рисунок 1.2.3. Кривые равной громкости

Среди слышимых звуков различаются звуки речи и музыкальные звуки. Стоит отметить, что все привычные нам звуки голоса или музыкальных не состоят из одной конкретной частоты. Помимо фундаментальной частоты тон может содержать **обертона** (все дополнительные собственные частоты), часть из которых может быть **гармониками** (обертонами, частоты которых кратны фундаментальной, например 440Гц при фундаментальной частоте 220Гц).

Связь амплитуды и частоты называется **амплитудно-частотной характеристикой (АЧХ)**. На рисунке 1.2.4 мы можем увидеть фундаментальную частоту - самый высокий пик, пики поменьше - обертона, и множество невысоких значений других частот - шумовые компоненты. Все эти частоты необходимы для формирования тембра инструмента.

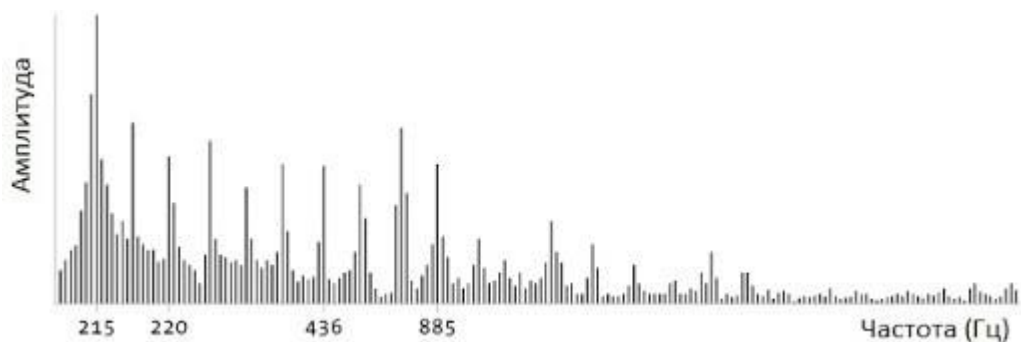


Рисунок 1.2.4. АЧХ оцифрованного звука ноты ля виолончели

Помимо амплитуды и частоты, у звуковой волны, как у любой другой, есть **фаза**.

Фаза описывает положение волны относительно начала координат, т.е. с какого момента началось колебание

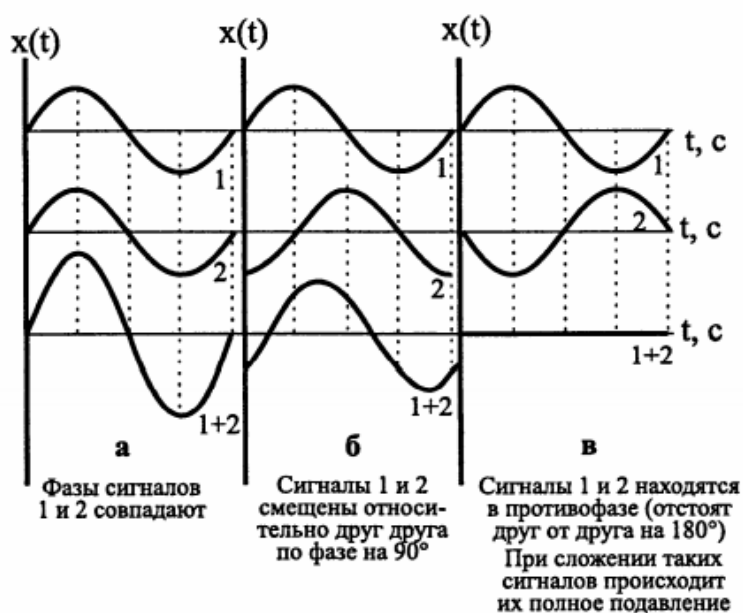


Рисунок 1.2.5. Не совпадающие по фазе сигналы: а - сдвиг на 0° , б - 90° , в - 180°

Фаза может оказывать существенное влияние на тембр звука в целом и в некоторых ситуациях портить панораму стерео. Более полно о физических основах акустических процессов и характеристиках звуковых полей можно почитать в книге И. Алдошиной и Р. Приттса «Музыкальная акустика».

1.3 Обработка звука

Общие сведения о способах аналоговой записи и обработке звука

Аналоговая запись - способ записи на носитель, при котором создается электрический эквивалент записанных звуков. Переменный электрический ток, колебания которого происходят в заранее известных границах, был на момент появления электромеханической звукозаписи единственным удобным способом упаковки звука на носитель. Такой тип записи просуществовал более пятидесяти лет, и кое-где до сих пор весьма ценится. Лишь с достаточным развитием цифровых технологий и появлением облачных стриминговых сервисов, он был вытеснен с массового рынка.

С 50-х годов 20-го века, с целью удешевления и ускорения ручного труда звукорежиссеров и технического персонала, а также для расширения творческих возможностей музыкантов и саунд-дизайнеров радио, кино и театра, получили распространение несколько групп устройств обработки звука. Все они были аналоговыми, то есть работали с электрическим напряжением, полученным с помощью микрофона или генератора (синтезатора), усиленным с помощью усилителя и управляемым с помощью микшерного пульта. Это напряжение соответствовало (с определенными технически обусловленными коэффициентами) амплитуде реального звука, и обработка его шла путем пропускания этого напряжения через определенным образом скомпонованные электрические цепи¹¹.

В первую группу входили устройства для работы с **динамическим диапазоном сигнала** - компрессоры, эквалайзеры, лимитеры, гейты, экспандеры.

Компрессор автоматизировал работу звукорежиссера при записи, позволяя ограничивать динамический диапазон записываемого звука, делая его пригодным для записи на носители того времени. Позже, звук компрессии

пришелся по вкусу как звукорежиссерам, так и музыкантам, артистам и потребителям, а носители стали менее требовательны к уровням, и компрессоры стали применяться уже целенаправленно как художественный эффект.

Эквалайзер дал возможность вносить изменения в амплитуду отдельных частей звукового спектра, вырезая лишние шумы или добавляя красок характерным частотам инструмента, голоса или микса.

Лимитер, как вариация компрессора, позволил управляемо увеличивать громкость фонограмм, не выводя ее за пределы динамического диапазона. Из-за повсеместного применения лимитеров даже возникло культурное явление, известное как «войны громкости».

Гейт, как устройство подавления звука, не достигающего необходимой амплитуды, нашел применение при записи голоса в недостаточно звукоизолированном окружении, например при записи публичных мероприятий или на радио.

Экспандер, устройство обратное компрессору, нашел применение, например, при работе на больших концертных площадках в составе компандерных систем управления динамикой звука или чуть позже для расширения динамического диапазона звука на цифровых аудиокассетах.

Вторая группа - **устройства модуляции**. Все они так или иначе работают за счет внесения задержки в копию сигнала и управлением ей с помощью генератора низкой частоты (LFO). К ним относятся хорус, фэйзер, фланжер, задержка (дилей).

Хорус, например, сдвигает фазу копии сигнала вперед и назад по фазе с периодом около 20-30 мс, и затем складывает эти сигналы, создавая так называемый расчесочный фильтр.

Если же уменьшить задержку корпуса до 5-15 мс и добавить обратную связь, получится **флэнжер**, создающий эффект летающего по стереополю звука.

Если еще уменьшить время задержки, получится **фэйзер**, создающий специфический круговой эффект (sweeping).

Если сигнал добавлять через равные промежутки времени с затуханиями, получится эффект эха или задержки (**дилей**).

Третья группа - **устройства искажения**, дисторшн, овердрайв, фузз. Особенно применимы к музыке, исполняемой на электрогитаре. Все перечисленные эффекты вносят нелинейные искажения, создаваемые резким ограничением амплитуды сигнала, что делает звук острым, агрессивным и богатым на гармоники.

Четвертая группа - устройства **пространственной обработки, ревербераторы**. Помогают в позиционировании инструментов или голосов актеров внутри виртуального или реального (например, в кадре кинофильма или в театре) звукового пространства. Создают эффект многократного отражения звука с затуханием. Аналоговая реверберация достигалась либо с помощью использования реального звукового окружения, например - с помощью записи акустики студии или концертного зала, либо с помощью пружинной или плиточной реверберации. В последних двух случаях электромеханическая энергия звука передается физическому объекту (пружине или металлической плите), а затем снимается микрофоном, после чего полученный звук добавляется к исходному.

1.4 История цифрового звука

Еще с 20-х годов 20-го века инженеры и ученые искали способ избавиться от шумов при работе с аналоговым звуком. В 1928-м году Г. Найквист в работе

«Определённые проблемы теории телеграфной передачи» определил особенности необходимой полосы пропускания для передачи импульсного сигнала.¹²

В 1933-м В. А. Котельниковым в работе «О пропускной способности эфира и проволоки в электросвязи» предложена и доказана Теорема Котельникова, согласно которой аналоговый сигнал с ограниченным спектром может быть восстановлен однозначно и без потерь по своим дискретным отсчётам, взятым с частотой строго большей удвоенной максимальной частоты спектра.¹³

В западной науке эта же теорема часто называется теоремой Шеннона, по фамилии ученого, опубликовавшего в 1948-1949 годах труды «Математическая теория связи» и «Передача данных при наличии шума», независимо придя в них к аналогичным Котельникову результатам.¹⁴

В 1950-1960 годах было написано немало научных работ о физических и математических основах цифровой связи. Разработками в этой области занимались Р. Хэмминг, Д. Хаффман, А. Хоквингем, И. Рид, Г. Соломон и другие. К началу 1970х разработкой технологического приложения математических и физических основ подключились крупные компании -Mitsubishi, Sony, NHK, Hitachi и прочие. Все 70-е годы шли разработки стандартов и решений, и лишь к 1982-м году лицензируется и стандартизируется формат компакт-диска, первого полностью цифрового устройства хранения звука. В этом же году появляется формат цифровой многодорожечной записи на магнитную катушечную ленту DASH от компании Sony¹⁵.

В 1991-м году появляется формат хранения битовых потоков аудио - WAV, позволивший хранить цифровые аудиоданные в компьютерных системах IBM и Microsoft¹⁶. Затем, с распространением интернета и цифровых неспециализированных носителей данных (флэш-дисков, внешних жестких дисков), для более компактного хранения аудиоданных разработаны форматы

сжатия аудио с потерями (MP3, OGG и др.), форматы сжатия без потерь, занимающие больший объем на диске, чем форматы с потерями, но меньший, чем WAV (FLAC, ALAC, Monkey's Audio и др.). В конце концов, к концу 2000-х годов повсеместное распространение широкополосного доступа в интернет почти полностью вытеснило физические носители данных, позволив скачивать музыку прямо при прослушивании с помощью стриминговых сервисов, оставив эпоху носителей позади. Согласно отчету международной федерации производителей фонограмм (IFPI), около 89% любителей музыки по всему миру используют стриминговые сервисы.

В профессиональной же среде распространились современные Hi-End форматы, с частотой дискретизации 192кГц и глубиной до 64 бит, неотличимые от лучшего ленточного аналогового аудио как для потребителей, так и профессионалов и окончательно завершили эпоху массового использования аналоговых устройств. В быту крупных студий звукозаписи аналоговые устройства встретить все еще можно, но играют они роль исключительно конкурентного преимущества и используются лишь на определенных этапах производства.

1.5 Цифровой звук

Если аналоговый звук представляет собой прямую конверсию звуковых волн среды в переменный электрический ток, то цифровой - дискретное по времени импульсное измерение этого тока с определенной частотой (частотой дискретизации) и разрядностью (уровнями квантования) за равные промежутки времени¹⁷. Пример можно увидеть на рисунке 1.5.1.

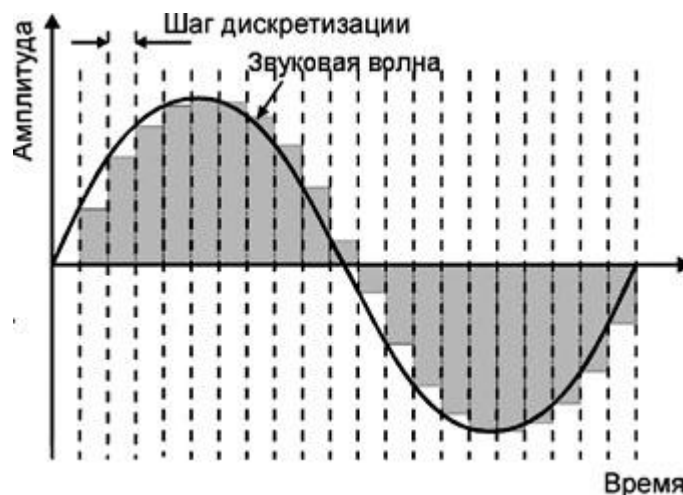


Рисунок 1.5.1. Дискретизация синусоидального сигнала

Дискретизация аналогового звука в цифровой выполняется специальным модулем - аналогово-цифровым преобразователем (АЦП).

Декодирование сигнала в аналоговый выполняется обратным устройством - цифро-аналоговым преобразователем (ЦАП). Пример - в Приложении А.

Алгоритм работы АЦП в общем случае включает следующие действия:

- 1) Фильтрация - аналоговый сигнал ограничивается по частоте с помощью фильтра низких частот (для отсеечения частот выше половины частоты дискретизации)
- 2) Дискретизация - устройство выборки-хранения делит сигнал на дискретные участки согласно частоте дискретизации (дискретный участок сигнала на отсчет)
- 3) Квантирование - квантователь высчитывает из участка сигнала наиболее близкую фиксированную величину
- 4) Оцифровка - кодирование фиксированной величины сигнала в последовательность бит определенного формата

ЦАП же работает обратным образом, но имеет несколько нюансов:

- 1) Исправление ошибок - при канальном декодировании исправляются несовпадающие с ожидаемыми значения
- 2) Декодирование со сглаживанием промежуточных значений (например, с помощью сглаживающего фильтра)
- 3) Восстанавливающий фильтр низкой частоты.

Получаемый аудиопоток со значениями обычно от -1 до 1 (семплами) и является цифровым представлением звука. Его можно преобразовать в определенный формат и сохранить на носителе.

Существует множество форматов хранения цифрового звука. К ним относятся:

- 1) Форматы без сжатия (битовые потоки без заголовков - .sam, .raw, или с заголовками - .wav)
- 2) Форматы со сжатием без потерь (lossless - FLAC, ALAC, APE и др.)
- 3) Форматы со сжатием с потерями (lossy - MP3, OGG, AAC, WMA и др.)

1.6 Особенности цифровой обработки звука

На заре своего существования, цифровой звук не мог похвастаться даже тенью превосходства над аналоговым. Из-за особенностей алгоритмов и технических реализаций цифровых устройств, могли возникать потери в качестве и шумы при преобразовании. Выделяют следующие типы шумов¹⁸:

Джиттер - колебания несущей частоты генератора устройства могут вносить нежелательные частотные искажения.

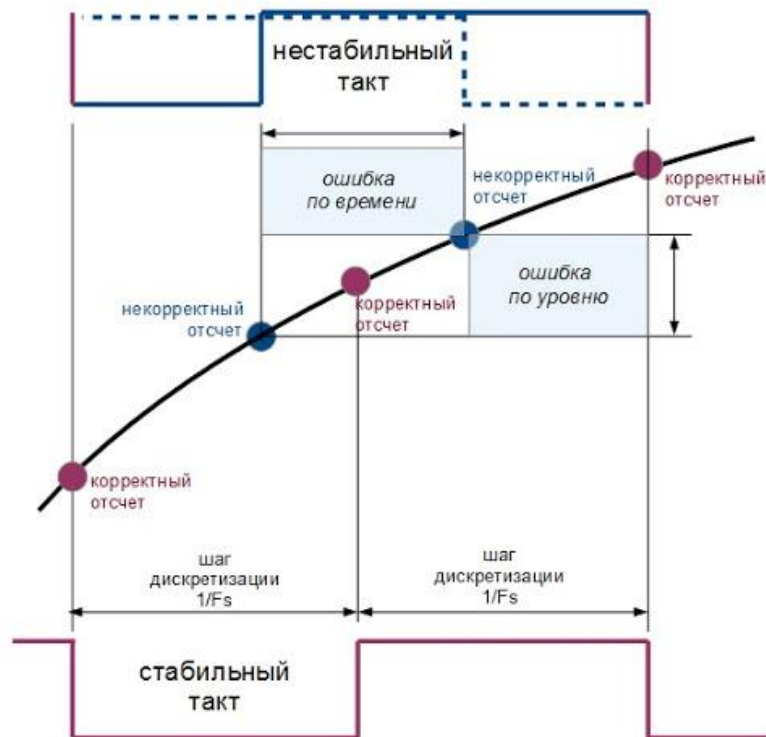


Рисунок 1.6.1. Джиттер - неустойчивость частоты квантования приводит к некорректным отсчетам при определении параметров точки квантования

Джиттер исправляется использованием устройств с точными генераторами тактовой частоты.

Шум квантования - возникает при недостаточной разрядности квантования

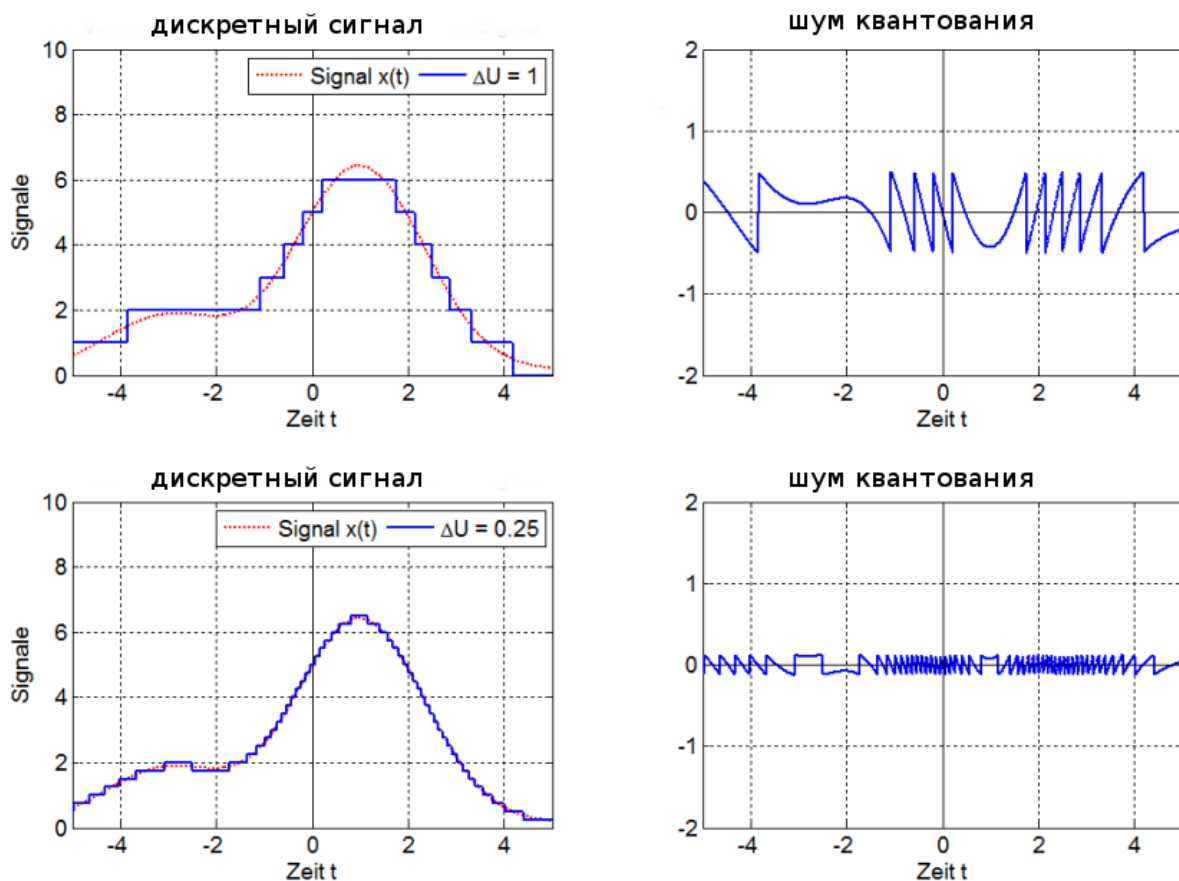


Рисунок 1.6.2. Зависимость уровня шума квантования от глубины квантования

Исправляется использованием аппаратуры, поддерживающей разрядность 24 бит и более.

Алиасинг (Aliasing) - эффект, приводящий к неразличимости сигналов разной частоты при дискретизации.

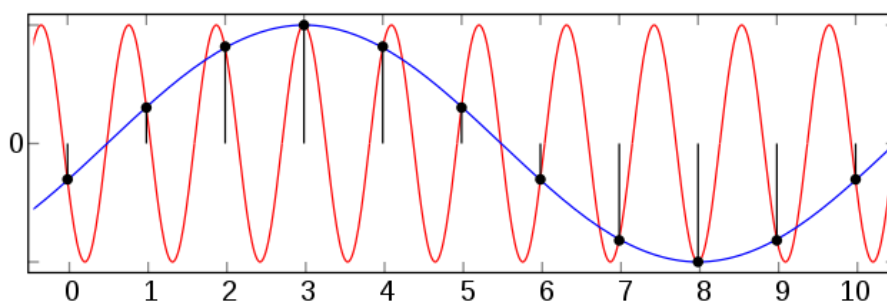


Рисунок 1.6.3. Алиасинг. Риски сверху - шаги дискретизации, синусоида высокой частоты (красная) квантуется точно так же, как и синусоида более низкой частоты (синяя)

Исправляется более высокой частотой дискретизации - 88.2 КГц, 96 КГц и выше.

Общие принципы работы с цифровым звуком сходны с таковыми при работе с аналоговыми, разница в методиках преобразований и учете специфики цифрового звука в принципе. Так работа с квантами звука производится напрямую с семплами, то есть уровнями, являющимися суммой уровней всех частот, оцифрованных в этом такте. Восстановить частотный спектр можно с помощью быстрых преобразований Фурье.

Так как мы имеем дело с абсолютными уровнями семплов, стоит отдельно упомянуть конвертирование уровней в более привычные для человека децибелы.

0 дБ - максимальный уровень, который может достигать система без искажений и перегрузок. Значения уровней, с которыми работает звукорежиссер обычно существует в пределах от некоего минимального значения (SPL в случае звука в воздухе или уровень выключения оборудования 0.001) до 0дБ. При необходимости управляемой обработки семпла и сравнении его значений с вводимыми пользователем (например, в компрессоре) необходимо осуществить преобразования уровня в дБ в абсолютный уровень семпла или наоборот, для выведения показаний редакции в дБ необходимо семпл преобразовать в децибелы.

Так как дБ единица измерения относительной величины, заданное значение будет являться отношением текущего значения к максимальному - 0дБ.

Каждые $20 \log_{10}(2)$, то есть 6.02 дБ будут означать изменение сигнала в 2 раза. То есть -6дБ вдвое тише максимума, -12дБ вчетверо и т.д.

Таким образом, преобразование семпла в уровень дБ выглядит так:

$$Df = 20 \log_{10}(s)$$

Где s - абсолютное значение семпла

Обратное преобразование из семпла в дБ:

$$s = 10 ^ { (1/20 * Df)}$$

Умножение семплов на значения от 0 до 1 вызывает их ослабление, на значения более 1 - усиление.

Выход семпла по модулю за пределы 1 вызовет клиппирование - обрезку с искажением частотного спектра.

Уменьшение же уровня по модулю до значения 0.001 приведет к принудительному отключению аудиодорожки в рабочей станции.

1.7 DSP (ЦСП)

В 1980-х, когда цифровые технологии только начали распространяться, процессоры общего назначения обладали весьма скромными характеристиками и для обработки цифровых сигналов не годились. В целях ускорения обработки, типовые операции упаковывали в виде аппаратного решения, создав класс устройств под названием DSP (digital signal processor) или ЦСП (цифровой сигнальный процессор). До появления стандарта VST и роста производительности процессоров общего назначения, ЦСП были единственной опцией для работы с цифровыми сигналами в реальном времени.

DSP представляет собой процессор, специализированный на обработке цифровых сигналов в реальном времени. Его техническая реализация позволяет производить типичные операции работы с семплами значительно быстрее и с меньшими затратами ресурсов, чем на процессорах общего назначения.

Особенности DSP:

- Быстрая аппаратная реализация типовых операций - цифровой фильтрации, поиска сигналов, быстрых преобразований Фурье и т.д.
- Векторизация вычислений с действительными числами
- Аппаратная реализация заранее описанных последовательностей команд для циклического выполнения
- Разделение памяти на буферы

Современные ЦСП имеют дополнительно имеют следующие преимущества:

- Высокая тактовая частота процессора и частота шины
- Многоядерность
- Многоуровневый кэш
- Встроенные контроллеры прямого доступа к памяти

DSP долгое время использовались в оборудовании, поставляемом вместе с программой ProTools, во многих аудиоинтерфейсах есть чипы DSP для обработки звука в реальном времени. Использовались в цифровых синтезаторах и музыкальных рабочих станциях. Могут входить в состав современных цифровых устройств вместе с процессорами общего назначения, облегчая выполнение ресурсозатратных операций.

1.8 VST

Параллельно с развитием DSP развивались и процессоры общего назначения. С начала 80-х годов и до середины 90-х производительность компьютеров выросла многократно, что позволило выполнять работу над семплами в реальном времени. В 1996-м году компания Steinberg разработала технологию VST (Virtual Studio Technology), позволившую выполнять обработку семплов в реальном времени на процессорах общего назначения. В составе VST был разработан интерфейс связи между цифровой рабочей станцией,

работающей на обыкновенном компьютере, и подключаемыми подпрограммами - плагинами. Интерфейс включал в себя как прямое управление пользователем настройками плагинов, так и автоматизацию этих настроек.

В 1999-м году появилась вторая версия технологии VST, позволившая передавать плагинам информацию в формате MIDI, что привело к появлению виртуальных музыкальных инструментов в формате VST.

Таким образом VST плагины делятся на три типа:

- 1) VST-эффекты - обработки, эффекты, любые плагины изменяющие характеристики семплов
- 2) VST-инструменты - плагины, генерирующие звук
- 3) VST-MIDI-эффекты - обработчики MIDI-сообщений (агрегаторы, маршрутизаторы и т.д.)

1.9 Программные эмуляции аналоговых устройств

В последние годы в звукорежиссерской и исполнительской среде стали популярны так называемые плагины-эмуляции. Они представляют собой алгоритмы, эмулирующие импульсную переходную характеристику аналоговых устройств (дельта-функция Дирака).

Преимущество таких плагинов в их практически неотличимом от оригинальных аналоговых устройств звучании.

С помощью таких плагинов можно достоверно воссоздать звучание винтажного компрессора, усилителя, и даже в полной мере передать звучание определенной комнаты или площадки.

РАЗДЕЛ 2

2.1 Обоснование выбора компрессора в качестве тестового плагина

Компрессор - устройство часто используемое, крайне вариативное. Логика работы компрессора довольно удобно ложится в основу алгоритма, интуитивна и легко имплементируется. К тому же, реализация компрессора несет в себе ряд технологических решений, способных углубить понимание проблематики работы цифровых устройств и цифрового звука.

Поэтому компрессор является наиболее подходящим устройством для демонстрационной разработки.

2.2 Используемые библиотеки и обоснование их использования

В 2020-м году на рынке разработки ПО существует огромное множество языков программирования, технологий, платформ и т.д. Многие из них позволяют путем несложных манипуляций добиться компиляции и сборки плагина VST.

Изначально технология VST написана на языке C/C++, дающем ей максимальную производительность, многопоточность, прямой доступ к памяти и кроссплатформенность. Большинство плагинов написаны именно на этом языке программирования. Однако, есть у него и свои минусы - относительно медленная разработка, потенциальная уязвимость неуправляемого кода, сложность в стандартизации.¹⁹

Главной целью данной работы является объяснение причин принятия тех или иных решений при разработке плагина и демонстрация этого процесса. Поэтому технологическую платформу автор выбирал исходя из собственных предпочтений и опыта разработки (на момент написания - 3 года коммерческой разработки на платформе .NET и языке C#).

Хотя платформа .NET и язык C# не так часто, как C/C++, задействуются для создания высокопроизводительных приложений для обработки сигналов в режиме реального времени, предоставляемого ими инструментария достаточно.

Вдобавок, для платформы .NET есть ряд удобных библиотек с открытым исходным кодом, бесплатным для любого использования, позволяющим генерировать рабочий VST плагин, интегрируя в него собственную логику.

Также, C# удобен и быстр в разработке, безопасен и глубоко интегрирован в логику системы Windows, а его производительность растет с каждой новой версией.

Наконец, компания Microsoft предоставляет профессиональную среду разработки на этой платформе бесплатно для некоммерческого использования (IDE Microsoft Visual Studio Community²⁰) и редактор кода Visual Studio Code бесплатно даже для коммерческой разработки²¹. А значит, никакие соглашения и лицензии не будут нарушены.

2.3 Кроссплатформенность

Библиотека .NET Core 3.1 кроссплатформенна. Код, собранный на ней, будет работать на устройствах Windows, Linux и Mac. Однако есть ряд нюансов, не позволяющих сделать будущий плагин полностью кроссплатформенным по умолчанию.

Дело в том, что Linux поддерживает VST лишь номинально. Даже если получится запустить образ с помощью программы Wine или расширить базовую логику плагина для поддержки отдельных Linux-систем, полной функциональности достичь скорее всего не удастся.

С системами Mac еще сложнее. В Mac поддержка VST отсутствует и может быть реализована только с помощью сторонних библиотек. Вместо него используется собственные стандарты плагинов - AAX и RTAS.

Поэтому целью данной работы является разработка плагина запуска исключительно для Windows.

2.4 Домены величин

Работ с компрессором будет сосредоточена в двух доменах - линейном и логарифмическом. В **линейном домене** значения представляются как реальное число в промежутке $[-1;1]$. В логарифмическом домене значения будут представлены в **децибелах (дБ или dB)** в промежутке от минимального (-96.33 дБ для 16 бит или 144.49 дБ) до 0дБ. Формулы конверсии можно найти в разделе 1.6.

Имплементация конверсии в децибелы и обратно представлена в классе DomainConverter.

2.5 Архитектура VST и архитектура плагина

Разработанный компанией Steinberg стандарт VST представляет собой упакованную в код конвенцию (соглашение) по обработке цифрового аудио внутри рабочей станции (DAW).

При добавлении плагина в DAW впервые, происходит считывание базовой информации о плагине (производитель, версия, количество каналов, категория и т.д.), прочтение настроек и параметров плагина и производится его первичная инициализация для проверки на наличие ошибок. Если плагин инициализировался штатно, он добавляется в список доступных в рабочей станции.

Каждый канал рабочей станции является по сути конвейером, проводящим буферы семплов сквозь плагины обработки. Будучи установленным на канал аудио, VST-плагин подключается к конвейеру обработки буферов семплов, получая для обработки входной и выходной буфер

семплов. При выключении обработки плагином буферов, входной и выходной буферы проходят сквозь логику плагина без изменений, оставаясь равны друг другу. При включении обработки, алгоритм преобразует семплы входного буфера и записывает в выходной, по общему индексу, если иное не продиктовано логикой работы создаваемого плагина.

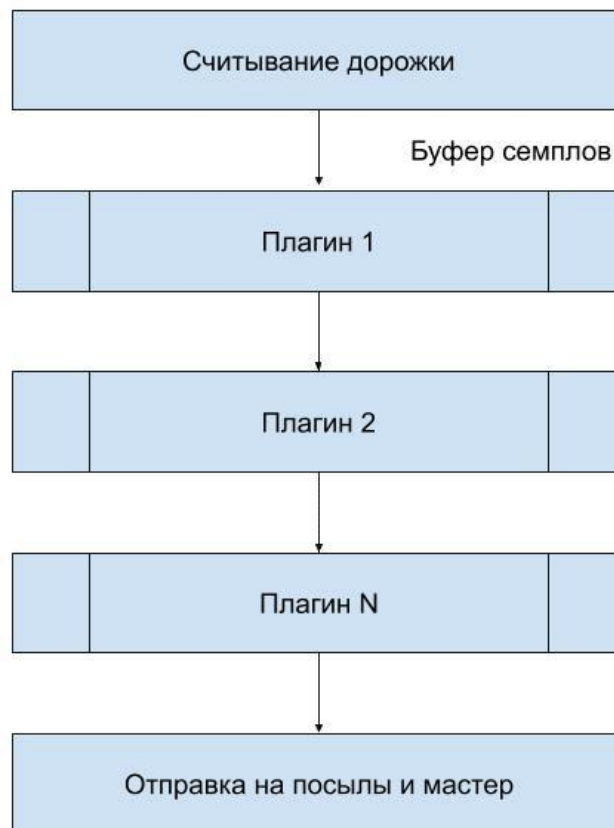


Рисунок 2.5.1. Схема передачи буфера семплов плагинов в DAW в режиме постобработки

Механизмы VST, используемые в этой работы, были имплементированы в библиотеке Jacobі VST²² с открытым исходным кодом и использованы без изменений. При компиляции плагин создается в виде двух файлов - Pressor.dll и Pressor.net.vst2. Первый - неуправляемая библиотека VST, которая вызывается рабочей станцией. Она в свою очередь обращается к управляемому коду, хранящемуся во втором файле.

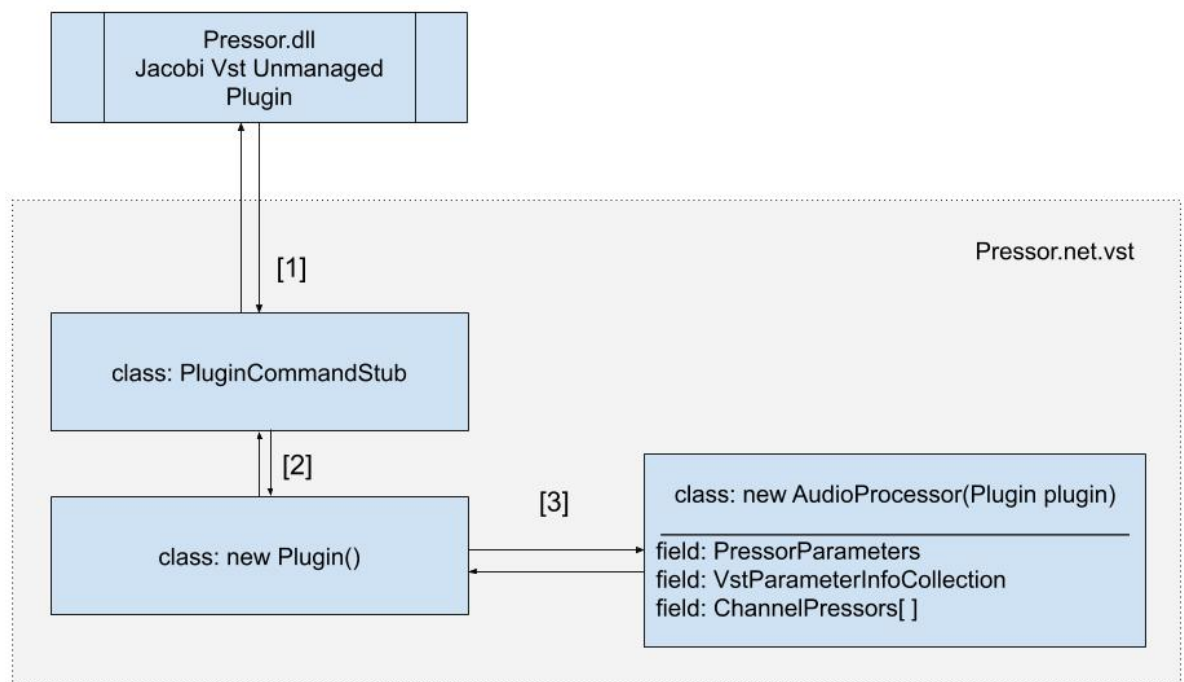


Рисунок 2.5.2. Архитектура плагина

Обращение хоста к неуправляемому плагину происходит через заглушку `PluginCommandStub` и проксируется на управляемый, и затем выполняется классом `Plugin`, который реализует инъекцию зависимостей в виде основного класса `AudioProcessor`, и вспомогательных - `PluginPersistence` и `PluginPrograms`.

`AudioProcessor` инициализируется один раз при первом вызове компрессора и загружает всю внутреннюю логику обработки семплов - класс компрессора, вспомогательные классы конечного автомата расчета тау-функции, класс параметров и класс состояния.

Экземпляр класса `Pressor` содержит внутреннюю логику работы компрессора, при которой создаются машины состояния, связываются параметры регуляторов плагина с переменными объекта `PressorParameters`, объявляются перманентные значения в классе состояний `PressorState`.

При выполнении метода `void AudioProcessor.Process (...)` происходит вызов метода `void Pressor.ProcessChannel (...)` для каждого буфера.

PluginPersistence обрабатывает состояние плагина и не управляется из кода.

PluginPrograms позволяет загрузить вместе с плагином набор пресетов.

2.6 Архитектура компрессора

Прежде, чем говорить о собственно архитектуре, стоит разобраться подробнее в принципах работы компрессоров.

Компрессия — это уменьшение динамического диапазона звука, то есть сокращение более громких участков звука без воздействия на тихие. Процесс компрессии можно описать так. Превышение семплом **X** порогового уровня **T**, вызовет уменьшение величины сигнала в зависимости от **R** (соотношения) и **W** (ширины колена), с поправкой на **T_A** и **T_R** (Время атаки и релиза) и с восстановлением конечного семпла **Y** на величину **M** (маскирующее усиление).

T - Threshold - порог срабатывания компрессора, устанавливается пользователем в дБ.

R - Ratio - отношение величины выходного уровня к входному.

W - Knee Width - ширина сглаживающей характеристики (колена)

T_A - Attack Time - время, необходимое компрессору для развития заданного уровня редакции, в мс

T_R - Release Time - время спада редакции до нуля, в мс.

M - Makeup Gain - величина постусиления всех выходных семплов компрессора, в дБ.

Вышеперечисленное - органы управления стандартным компрессором. В некоторых реализациях часть из них может отсутствовать, либо присутствовать дополнительные (lookahead, input/output, sidechain и т.д.).

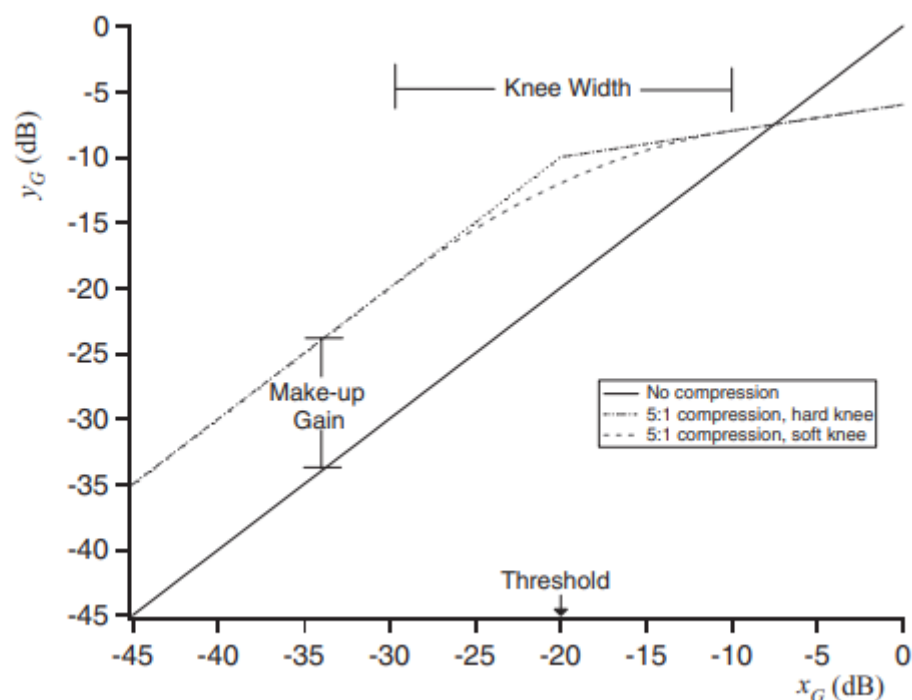


Рисунок 2.6.1. Статическая характеристика компрессии с маскирующим усилением, жесткой и мягкой характеристикой («коленом»).

В общем случае сигнал, приходящий на вход компрессора, делится на две копии. Одна отправляется на боковую цепь с детектором и вычислителем ослабления, где происходит сравнение участка сигнала с пороговой характеристикой и рассчитывается величина ослабления. Другая идет в аттенюатор, где сигнал ослабляется согласно параметрам первой. Биполярный сигнал на боковой цепи складывается в однополярную огибающую уровня, которая используется для определения пика или среднеквадратичного уровня.

23

Порядок расположения элементов и домен, в котором они выполняют свою задачу, не случайны и могут существенно влиять на результат работы компрессора. Рассмотрим их подробнее.

2.6.1 Аттенюатор

Аттенюатор выполняет ослабление сигнала. В различных аппаратных реализациях компрессоров для этого использовались управляемые напряжением усилители на основе радиоламп, фотоэлементов, транзисторов и т.д. В цифровом мире можно воплотить практически идеальный аттенюатор, умножающий управляющее напряжение на полезный сигнал. В случае линейного домена, если $x[n]$ — это входной семпл, $y[n]$ - выходной, $c[n]$ - управляющее напряжение, формула его работы будет выглядеть так:

$$y[n] = x[n] * c[n]$$

А в логарифмическом домене вот так:

$$Y(db)[n] = X(db)[n] + C(db)[n] + M$$

Где Y , X и C - все то же самое, но в дБ, а M - величина маскирующего усиления

2.6.2 Вычислитель управляющего напряжения

Вычислитель генерирует управляющее напряжение согласно параметрам T , R и W , и величине входного сигнала и определяет статическую характеристику компрессии. Как только сигнал превосходит пороговое значение, он ослабляется согласно величине $Ratio$.

$$R = \frac{x_G - T}{y_G - T} \quad for \quad x_G > T$$

Рис. 1. Зависимость величины $Ratio$ от входного(x_G) и выходного семпла(y_G), порога (T)

Статическая функция компрессии вычисляется так:

$$y_G = \begin{cases} x_G & x_G \leq T \\ T + (x_G - T)/R & x_G > T \end{cases}$$

Рис. 2. Формула расчета выходного значения при $W = 0$

Такая характеристика присуща компрессору с так называемым «Твердым коленом» (Hard knee), способному вызывать появлением цифровых артефактов (шумов квантования и дисторшна от резкого сокращения амплитуды сигнала).

Для улучшений характеристик переходных процессов вводят параметр колена (knee width) - ширину зоны, в которой срабатывает компрессия.

С его добавлением формула принимает следующий вид:

Рис. 3.

Формула расчета выходного значения при $W > 0$, где W - ширина колена

$$y_G = \begin{cases} x_G & 2(x_G - T) < -W \\ x_G + (1/R - 1)(x_G - T + W/2)^2 / (2W) & 2|(x_G - T)| \leq W \\ T + (x_G - T)/R & 2(x_G - T) > W \end{cases}$$

2.6.3 Детектор уровня

Детектор предоставляет сглаженную огибающую сигнала. Во избежание искажений и звуковых артефактов, ослабление сигнала должно происходить с определенными задержками. Обычно они вводятся в систему параметрами атаки (T_a) и релиза (T_r) через цифровой сглаживающий фильтр.

$$s[n] = \alpha s[n-1] + (1 - \alpha)r[n]$$

Где α - коэффициент сглаживания, $r[n]$ - входящий и $s[n]$ - выходящий параметры.

Шаг фильтра соответствует формуле:

$$s[n] = 1 - \alpha^n; \quad x[n] = 1, n \geq 1$$

Временная константа T определяет время, необходимое системе, чтобы достичь $1 - 1/e \sim 0.63$. Таким образом коэффициент сглаживания

$$\alpha = e^{-1 / (T * fs)}$$

Где e - экспонента, T - параметр времени атаки или релиза (в зависимости от состояния детектора), fs - частота дискретизации

2.6.4 Детектор пиков

На сегодняшний день существует несколько реализаций детектора пиков огибающей, о которых можно прочитать в профильной литературе. В данной работе была использована схема сглаживающего разветвленного детектора (smooth branching detection). Особенность такой схемы в том, что она выдает наиболее плавную кривую атаки-релиза и вносит минимальные искажения в сигнал.

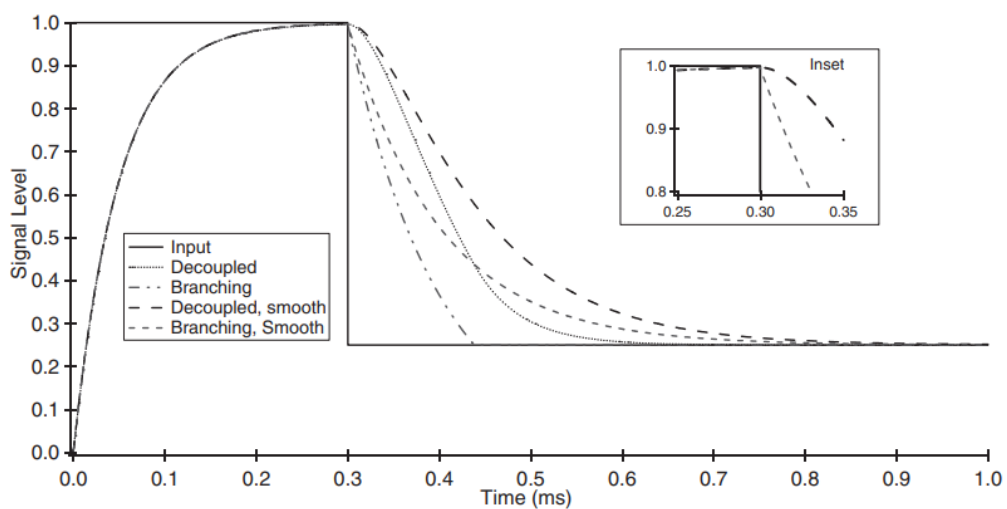


Рисунок 2.6.4.1. Формы спада волны для детекторов различной конфигурации

Работа такого детектора зависит от предыдущего результата, сохраняемого в переменной. Она соответствует формуле

$$y_L[n] = \begin{cases} \alpha_A y_L[n-1] + (1 - \alpha_A) x_L[n] & x_L[n] > y_L[n-1] \\ \alpha_R y_L[n-1] + (1 - \alpha_R) x_L[n] & x_L[n] \leq y_L[n-1] \end{cases}$$

Где α_A и α_R соответственно сглаживающие коэффициенты атаки и релиза, n - индекс текущего семпла, $x_L[n]$ и $y_L[n]$ - входной и выходной уровни детектора.

2.6.5 Собственно архитектура

Как уже было сказано, порядок и свойства элементов компрессора могут существенно влиять на его работу. Именно по этой причине существует множество конфигураций, схем и архитектур компрессоров, а сами они так сильно отличаются как по применимости, так и по звучанию.

По схеме включения вычислителя управляющего напряжения разделяют компрессоры с простой обратной связью (feedback-compressor), с разделенной обратной связью (alternate feedback-compressor) и компрессоры прямого действия (feedforward compressor) (рисунок 2.6.5.1). Первые складывают входной семпл с управляющим напряжением из предыдущего такта сразу на входе, а затем используют результат и как выходное значение, и для расчета управляющего напряжения для следующего такта. Вторые и третьи складывают управляющее напряжение и исходный семпл в конце цикла, но компрессоры прямого действия делают это напрямую за один такт, тогда как компрессоры с разделенной обратной связью перед расчетом управляющего напряжения складывают результат из предыдущего такта с исходным семплом. Компрессоры прямого действия зарекомендовали себя как наиболее

совершенное решение и включены в архитектуру разрабатываемого компрессора.

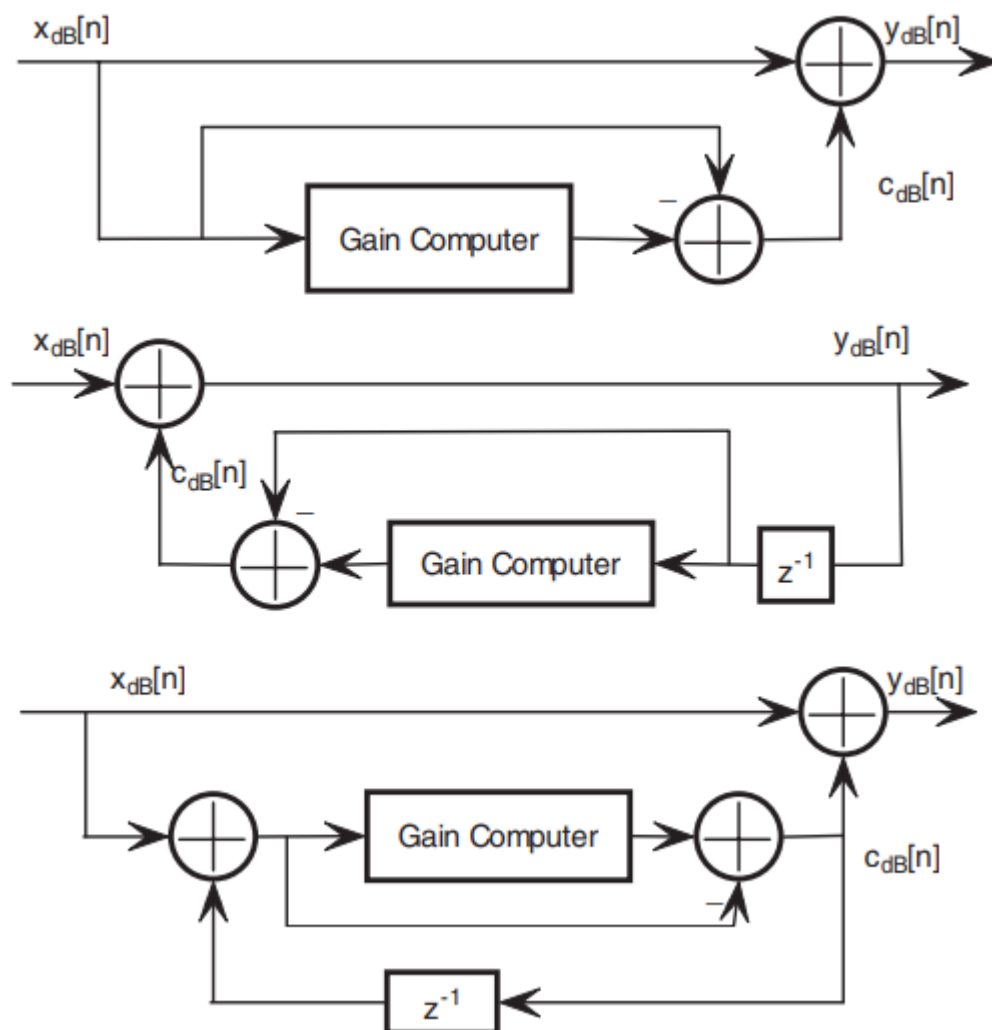


Рисунок 2.6.5.1. Схемы подключения вычислителей управляющего напряжения сверху вниз: прямого действия, с обратной связью, с разделенной обратной связью

Детекторы компрессоров подразделяются на детекторы возврата к нулю, детекторы возврата к пороговому уровню и детекторы логарифмического домена (Приложение Б). Поскольку детекторы возврата к нулю или пороговому значению выполняют ряд лишних операций и не всегда корректно обрабатывают атаку и релиз, в состав архитектуры разрабатываемого компрессора вошел детектор логарифмического домена.

Таким образом, разработанный компрессор реализует следующие особенности:

- 1) Изменяемое (мягкое и жесткое) колено
- 2) Вычислитель управляющего напряжения, подключенный по принципу прямого действия
- 3) Детектор уровней логарифмического домена
- 4) Аттенюатор линейного домена

2.7 Подготовка к разработке

Чтобы приступить к непосредственной разработке плагина необходимо установить ряд программ.

- 1) Среда разработки с поддержкой языка C# и платформы .NET Core (JetBrains Rider, MS Visual Studio, Visual Studio Code, etc.)
- 2) Установить SDK .NET Core 3.1 ²⁴
- 3) Установить DAW, если нет, создать дорожку и вставить на нее набор тестовых звуков
- 4) Скопировать из папки с репозиторием .NET Core библиотеку ljwhost.dll в ту же папку, где находится исполняемый файл DAW (.exe)
- 5) Создать проект плагина (библиотека классов C#)
- 6) Открыть менеджер пакетов NuGet, установить флажок «Включать пререлизы» и установить пакет VST.NET2-Plugin (как на рисунке 2.7.1)
- 7) Перезагрузить решение

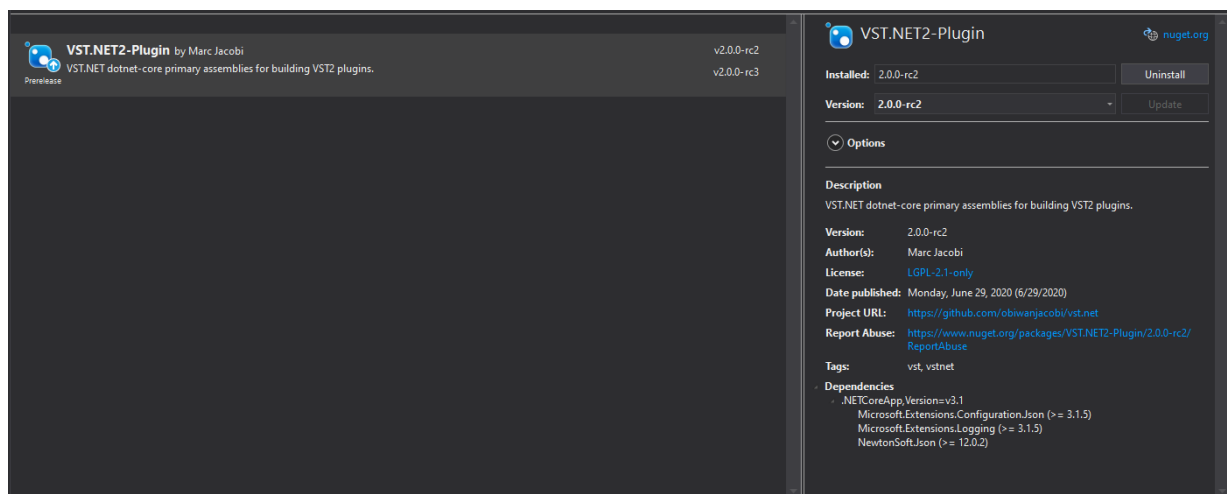


Рисунок 2.7.1. Установка плагина VST.NET2-Plugin

2.8 Разработка

В дальнейшем по умолчанию все поля будут создаваться приватными, все свойства - публичными, все классы - запечатанными. Пространства имен соответствуют именам папок, в которых лежат. Однако, это ни в коем случае не обязательно. Дополнительно, стоит отметить, что изучение этого раздела значительно упрощается наличием кода перед глазами. Поэтому рекомендуется открыть IDE с ним, или хотя бы страничку на GitHub.

2.8.1 Добавление шаблона

В списке литературы можно найти ссылку на GitHub-репозиторий с пустым шаблоном решения. При клонировании его в репозиторий создастся структура решения.²⁵

Шаблон решения содержит классы, необходимые для работы плагина на основе библиотеки .NET VST. В папке VST содержатся библиотечные классы, функциональность которых может быть расширена. В папке Logic находится класс AudioProcessor.cs, который будет вызывать пользовательскую логику

работы плагина. В папке Calc находятся вспомогательные классы с полезными математическими функциями.

2.8.2 Создание пользовательской логики. Pressor

Для начала разработки пользовательской логики плагина необходимо создать класс с названием плагина, в текущем случае Pressor.cs. Данный класс будет включать в себя всю логику обработки буферов семплов.

Класс Pressor реализован со следующими важными членами:

- 1) Свойство состояния типа PressorState, ссылающееся на один из объектов массива `_pressorStates[]`. Оно адресуется к нему по текущему индексу канала, сохраненному в отдельной переменной. В каждом инстансе данного класса хранятся значения, изменяемые каждый такт - уровни, управляющее напряжение и т.д.
- 2) Публичное свойство PressorParameters, хранящее в себе инициализированные менеджеры параметров VST. Гораздо меньше подвержены изменениям, чем данные класса PressorState, поэтому отделены

Все параметры и переменные можно хранить и в самом классе Pressor, но с целью соблюдения принципов объектно-ориентированного программирования и SOLID, были разделены между отдельными классами. К тому же, класс содержит ряд переменных, инициализирующихся и использующихся только в конфигурации DEBUG, то есть при отладке.

Основной метод класса ProcessChannel принимает на вход массивы входящих и выходящих семплов и индекс текущего канала. Индекс канала сохраняется для адресации к состоянию, т.к. стереоканалы обрабатываются независимо. Далее идет процесс обработки, описанный в разделе 2.6. Примеры разработанного кода можно увидеть в репозитории.²⁶

Следует отметить, что модули, указанные в разделе 2.6, не обязательно должны реализоваться буквально. Достаточно того, что существуют отдельные фрагменты, когда, реализующие этот функционал.

2.8.3 Отладка

В текущей работе для отладки кода использовалась DAW Reaper. Из статьи про разработку синтезатора (Приложение 2.8.3.1) ясно, что отладка в рабочей станции FL Studio так же возможна. Автор не тестировал остальные рабочие станции на совместимость с отладкой кода .NET VST, но технически препятствий для этого нет. Необходимо соблюсти два условия:

- 1) В папке DAW, где лежит исполняемый файл, обязательно должна находиться библиотека `ljwhost.dll`
- 2) Необходимо найти и добавить разрабатываемый плагин (даже пустой) в список доступных плагинов DAW и повесить его на дорожку

Далее в IDE необходимо нажать на кнопку «Присоединиться к процессу» и в списке процессов найти название DAW (Приложение 2.8.3.2) и подтвердить. Запустится отладка кода. Весь код плагина станет интерактивным и при достижении точек останова будет показывать содержимое переменных. Для дополнительного упрощения работы с массивами семплов можно установить расширение Visual Studio (или любое подобное для Rider или VSCode) Plot View (Приложение 2.8.3.3). При отладке каждый промежуточный результат работы плагина можно будет упаковать в массив данных и увидеть на графике в плагине.

2.8.4 Тестирование

Промежуточное тестирования результатов работы плагина может быть выполнено в процессе отладки при визуальном осмотре на графике или прямо в DAW. В первом случае мы смотрим на наличие артефактов в огибающей и путем анализа определяем тип характеристики, наложившей данный тип

артефактов. Во втором случае необходимо выполнить следующее - выход дорожки с плагином направить на вход другой дорожки, замутировав посыл на мастер последней. В таком случае можно быть записывать результат работы плагина на дублирующую дорожку и сравнивать огибающую увеличивая масштаб.

Конечное тестирование подразумевает более точную методологию. Необходимо протестировать плагин на уровни искажений, вносимых в сигналы различной формы определенной частоты на разных настройках. Для этого сгенерированный синусоидальный сигнал записывается через компрессор и сравнивается по следующим параметрам:

- 1) Соответствие спектра - любой компрессор так или иначе искажает сигнал, но уровень искажений должен быть в заданных рамках. Как видно на спектрограмме, спектр звука существенно не затронут, однако интенсивность звука снижена, значит компрессор отработал, не искажая сигнал (рисунок 2.8.4.1).
- 2) Величина редакции - сжатый сигнал должен укладываться в заданный динамический диапазон. Как видно на скриншоте, огибающая сигнала разработанного компрессора практически совпадает с эталонной, сделанной с помощью компрессора Waves C1 (рисунок 2.8.4.2).

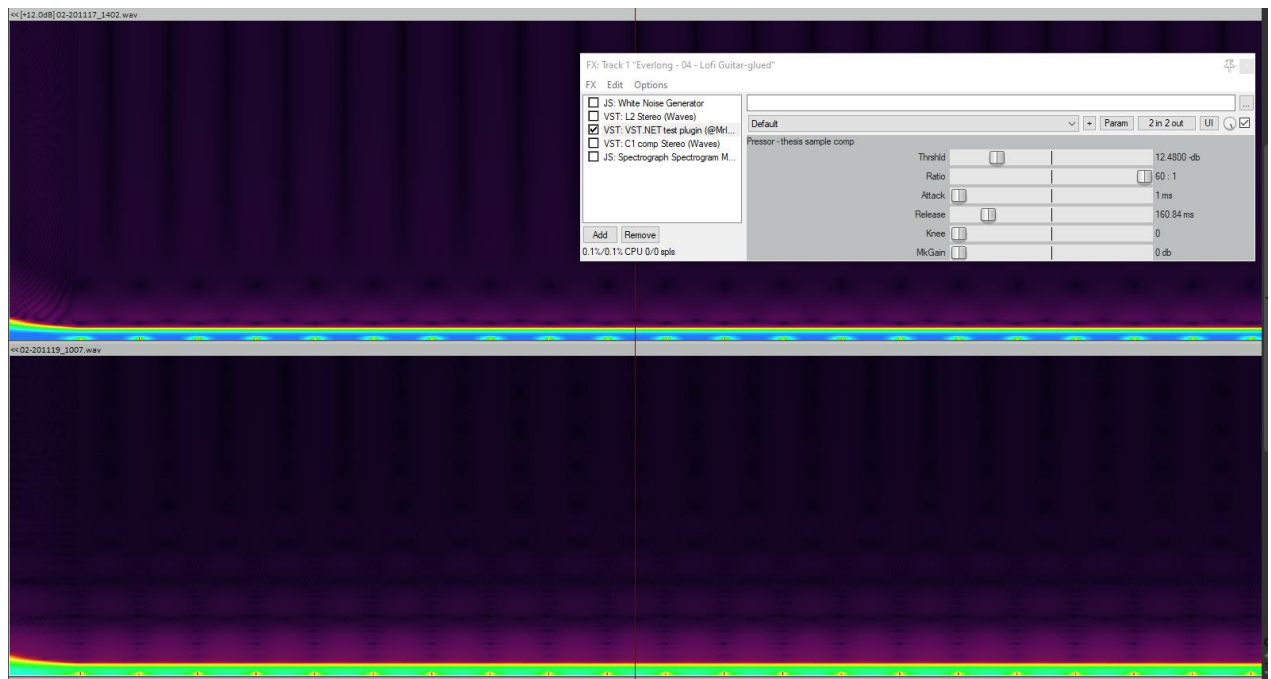


Рисунок 2.8.4.1. Сверху - исходная синусоида 0дб, снизу - скомпрессированная при $T=-12\text{дб}$, $R=60:1$

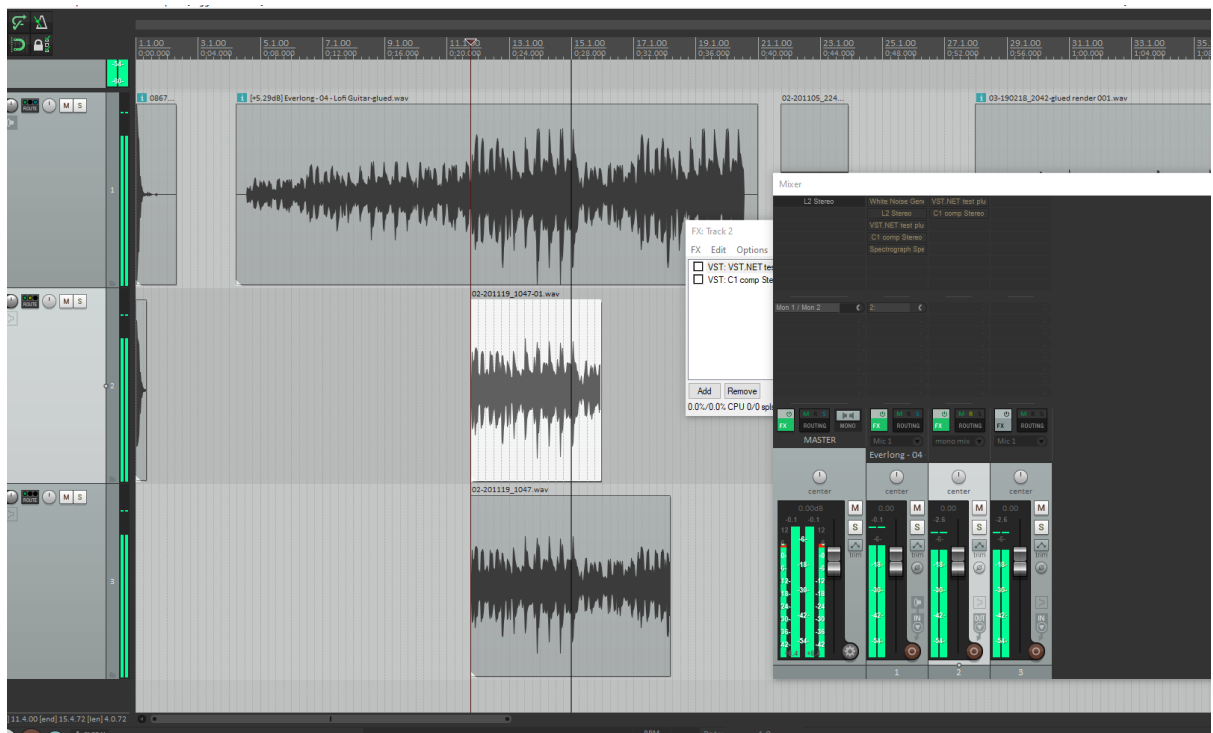


Рисунок 2.8.4.2. Сравнение уровней сигнала оригинала (наверху), эталона (внизу), скомпрессированного с теми же настройками на Waves C1 и скомпрессированного разработанным плагином (посередине).

По результатам тестов, созданный плагин полностью соответствует требованиям.

ЗАКЛЮЧЕНИЕ

Выводы

В ходе данной работы была изучена и кратко изложена история звукорежиссуры, сформированы и описаны способы обработки звука, исследованы особенности работы с цифровым звуком.

В практической части данной работы был полностью спроектирован, разработан и протестирован цифровой программный компрессор, рассмотрены особенности работы компрессоров и их реализации.

В целом, данная работа показала, что программист-звукорежиссер должен понимать особенности программных платформ, разбираться алгоритмах и проектировании, иметь минимальные знания математики, понимать, осознавать, как технические параметры разрабатываемого устройства превращаются в художественные возможности, уметь искать информацию и работать с ее источниками. Данная работа может помочь будущему специалисту в смежных областях обрести первичные навыки и понимание фундаментальных основ разработки и звукорежиссуры, а параллельное изучение источников и профильной литературы помогут обрести опыт и начать самостоятельную работу в этой области.

СПИСОК ЛИТЕРАТУРЫ

- 1) ¹ Кибиткина, Э.В. Методика обучения основам музыкального программирования [Электронный ресурс] -
<https://www.dissercat.com/content/metodika-obucheniya-osnovam-muzykalnogo-programmirovaniya>
- 2) ² Терентьев, Ю.Ю. Структура и содержание музыкальной информатики в системе музыкального образования
(<https://www.dissercat.com/content/struktura-i-soderzhanie-muzykalnoi-informatiki-v-sisteme-professionalnogo-muzykalnogo-obrazo>)
- 3) ³ Пучков, В.С. Музыкальные компьютерные технологии как новый инструментарий современного творчества [электронный ресурс] -
<https://www.dissercat.com/content/muzykalnye-kompyuternye-tekhnologii-kak-novyi-instrumentarii-sovremennogo-tvorchestva>
- 4) ⁴ Новые слова и значения. Словарь-справочник по материалам прессы и литературы 70-х годов / Под ред. Н. З. Котеловой. — М. : Русский язык, 1984.
- 5) ⁵ Koetsier, T. On the prehistory of programmable machines: musical automata, looms, calculators [Текст] / Teun Koetsier // Mechanism and machine theory. - 2001. - 36. - С.590
- 6) ⁶ История звукозаписи [Электронный ресурс] -
https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F_%D0%B7%D0%B2%D1%83%D0%BA%D0%BE%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D0%B8
- 7) ⁷ Музыкальная энциклопедия. — М.: Советская энциклопедия, Советский композитор. Под ред. Ю. В. Келдыша. 1973—1982.

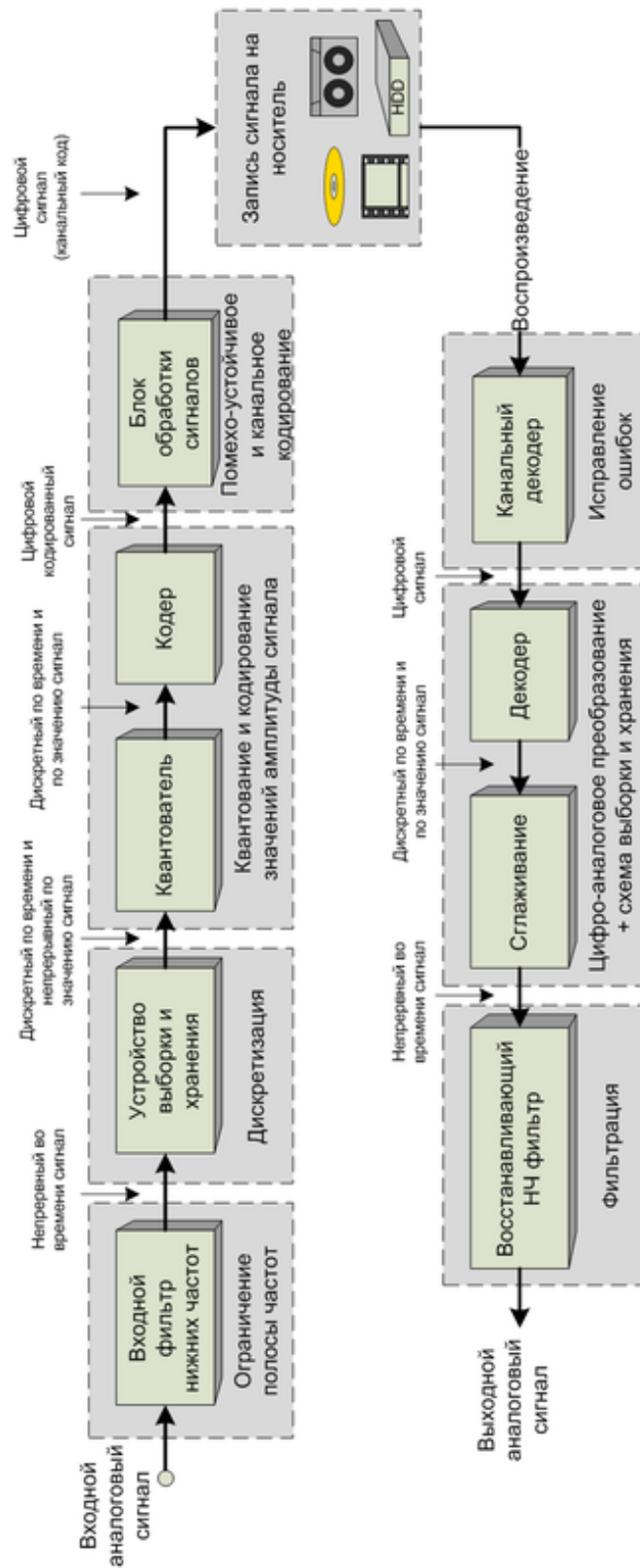
-
- 8) ⁸ Алдошина, И. Музыкальная акустика [Текст]: учебник для высших учебных заведений / И. Алдошина, Р. Приттс. - СПб.: Композитор - Санкт-Петербург, 2006. - 40 с.
- 9) ⁹ РЕКОМЕНДАЦИЯ МСЭ-R V.574-4. Использование децибела и непера в электросвязи [Текст] // Ассамблея радиосвязи МСЭ. 1978-2000. - 1-2 с.
- 10) ¹⁰ ГОСТ Р ИСО 226-2009. Акустика. Стандартные кривые равной громкости [Электронный ресурс] // "Научно-исследовательский центр контроля и диагностики технических систем" (АНО "НИЦ КД") - <http://docs.cntd.ru/document/gost-r-iso-226-2009>
- 11) ¹¹ Звуковой эффект [Электронный ресурс] - https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B9_%D1%8D%D1%84%D1%84%D0%B5%D0%BA%D1%82
- 12) ¹² Н. Nyquist, "Certain topics in telegraph transmission theory, " Trans. AIEE, vol. 47, pp. 617—644, Apr. 1928
- 13) ¹³ Котельников В. А. О пропускной способности «эфира» и проволоки в электросвязи // Успехи физических наук : Журнал. — 2006. — № 7. — С. 762-770.
- 14) ¹⁴ С. Е. Shannon. Communication in the presence of noise. Proc. Institute of Radio Engineers. Vol. 37. No. 1. P. 10—21. Jan. 1949.
- 15) ¹⁵ Цифровая звукозапись [Электронный ресурс] - https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B7%D0%B2%D1%83%D0%BA%D0%BE%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C
- 16) ¹⁶ Microsoft Corporation (June 20, 1999). "Waveform Audio File Format, MIME Sub-type Registration - INTERNET-DRAFT". IETF.

-
- 17) ¹⁷ Цифровая звукозапись [Электронный ресурс] -
https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B7%D0%B2%D1%83%D0%BA%D0%BE%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C
- 18) ¹⁸ Силаев, М. Теория звука. Что нужно знать о звуке, чтобы с ним работать. Опыт Яндекс.Музыки [Электронный ресурс]. -
<https://habr.com/ru/company/yandex/blog/270765/>
- 19) ¹⁹ Virtual Studio Technology [Электронный ресурс] -
https://en.wikipedia.org/wiki/Virtual_Studio_Technology
- 20) ²⁰ Visual Studio Community Edition [Электронный ресурс]:
<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16&apptype=desktop&tech=dotnetCFV&os=windows>
- 21) ²¹ Visual Studio Code [Электронный ресурс] -
<https://code.visualstudio.com/>
- 22) ²² obiwanjacobi/vst.net: Virtual Studio Technology (VST) for .NET. Plugins and Host applications. (github.com) [Электронный ресурс] -
<https://github.com/obiwanjacobi/vst.net>
- 23) ²³ Giannoulis, D. Digital Dynamic Range Compressor Design— A Tutorial and Analysis [Текст] / Dimitrios Giannoulis, Michael Massberg, Joshua D. Reiss. // Journal of the Audio Engineering Society. - 2012. - Vol. 60, No. 6. – 399 с.
- 24) ²⁴ Загрузить пакет компонентов для разработчиков .NET [Электронный ресурс] - <https://dotnet.microsoft.com/download>

-
- 25) ²⁵ MrImbolo/vstnet2-template: Public project with the minimal vst.net solution structure for quick start (github.com) [Электронный ресурс] - <https://github.com/MrImbolo/vstnet2-template>
- 26) ²⁶ MrImbolo/PRESSOR: Sound design thesis practical part - simple vst compressor for win x86 or x64 (github.com) [Электронный ресурс] - <https://github.com/MrImbolo/PRESSOR>

ПРИЛОЖЕНИЕ А.

Примерный алгоритм работы АЦП (вверху) и ЦАП (внизу)



ПРИЛОЖЕНИЕ Б

Схемы компрессора в зависимости от расположения детектора и домента.

- а) детектор возврата к нулю
- б) детектор возврата к пороговому уровню,
- с) детектор логарифмического домена

