

Два метода восстановления промежуточных значений между характерными точками и их сравнение

В. С. Родионов, email: tuukvadim@live.com
Н. Е. Балакирев, email: balakirev1949@yandex.ru
М. В. Зеленова, email: businkakatilas@mail.ru
А. А. Рубан, email: andreygrigorev0@gmail.com
М. М. Фадеев, email: fadeev_mix@bk.ru

Московский авиационный институт
(национальный исследовательский университет)

***Аннотация.** В данной работе рассматриваются два метода восстановления информации по характерным точкам. Особо рассматривается алгоритм, по которому происходит восстановление промежуточных значений, процесс обращения к реализованным методам и непосредственно скорость восстановления обоими методами. Необходимым условием алгоритма является значения начального и конечного значения амплитуды характерной точки и количества отсчетов между ними.*

***Ключевые слова:** характерные точки, табличное восстановление, RDTSCP, амплитуды, математический сопроцессор, восстановление промежуточных значений, фрагмент волны, идеальный образ.*

Введение

Имея возможность фиксировать поведение волны, в том числе и звуковой, по определяемым **характерным точкам** [1] и зная временной промежуток между ними, требуется восстановить (вычислить) промежуточные значения между этими точками в соответствии с теоретической формулой некоторой непрерывной аналитической функции и соответствующим способом экстраполяции [2]. Восстановление на практике может отходить от строгих канонов теории, исходя из имеющихся дополнительных условий и менее строгих требований точности, не влияющих на качество воспроизведения. Именно этому вопросу и посвящено дальнейшее рассуждение.

В настоящее время время для практического применения волны оцифровываются путем фиксации значения амплитуд через равные временные промежутки, называемые отсчетами. Время соответствует

© Родионов В. С., Балакирев Н. Е., Зеленова М. В., Рубан А. А.,
Фадеев М. М., 2021

количеству отсчетов между характерными точками при условии фиксации значения дельты между отсчетами. Известно, что для звуковой волны оцифровываемые значения амплитуд представляются в ограниченных пределах ($2^8 = 256$ – трехзначное число, $2^{16} = 65536$ пятизначное число, $2^{24} = 16777216$ восьмизначное число). При сравнительно малом значении количества отсчетов (по отношению к значениям амплитуд) между характерными точками диапазон в интервале от 0 до 90° разбивается на сравнительно большие угловые участки, что позволяет получить различные значения амплитуд при понижении требований высокой точности. Теоретически промежуточное значение амплитуды вычисляется как произведение экстремального значения в точке, где угол 90° , на значение SIN угла, соответствующее отсчету. С учетом вышесказанного требования точности получения значений могут быть в определенной степени снижены, при условии отсутствия совпадения значений амплитуд при разных значениях углов (соответствуют отчетам) в рассматриваемых пределах.

Если внимательно посмотреть на таблицу 1 вычисления SIN в пределах 90 градусов, то мы видим, что при вычислении относительно экстремального значения на рассматриваемом промежутке восстанавливаемых значений амплитуд для их различения требуется:

$0^\circ - 62^\circ$ – двухзначное число

$63^\circ - 87^\circ$ – трехзначное число

$88^\circ - 90^\circ$ – четырехзначное число

Такая точность позволяет иметь интервал до 90 отсчетов между характерными точками.

Возможное превышение этой величины для «разметки» до 90° , но с точностью до $60'$ (минут) требуется уже 5-6 знаков, и то только для диапазона $88^\circ - 90^\circ$. В этом случае такой диапазон значений угла (соответствующее определенному отсчету) при умножении на максимальное значение экстремума позволяет иметь до 5400 отсчетов между характерными точками. На практике количество отсчетов между двумя характерными точками с очень большой вероятностью (для записи звука это по факту правило) не может превышать 90 отсчетов, поэтому для большинства участков восстановления достаточно точности до 4 знаков.

Такое снижение требований, практически не влияющее на качество воспроизведения восстановленного сигнала, позволяет перейти на табличный (индексный) способ вычисления SIN, что обеспечивает увеличение скорости получения результата. Это особенно актуально для

больших объемов данных, обрабатываемых (например, для звука) в единицу времени до десятков и даже сотен тысяч параметров. Получаемый эффект при использовании такого способа вычисления по отношению к стандартному рассматривается далее. Будет сравнивать восстановление в стандартном варианте с вариантом, упрощенным – табличным. Для этого потребуется внести коды предлагаемых алгоритмов в программу-измеритель, которая будет фиксировать время, потраченное на восстановление заданного объема данных по тестовым значениям амплитуд характерных точек и количеству отсчетов между ними.

Для вычисления промежуточных значений амплитуд между характерными точками, так или иначе, требуется получить значение SIN по значению номера отсчета, которое соответствует значению угла. Традиционно можно обратиться к вычислению по стандартной команде вычисления FSIN в сопроцессоре [3]. Но это вычисление с плавающей точкой (с плавающей запятой для русской литературы), и оно требует достаточно большого времени на вычисление по сравнению с операциями для целых чисел. Кроме этого, это значение затем надо преобразовать в целое, не говоря о том, что для подготовки обращения к сопроцессору и извлечения результата также требуется несколько подготовительных команд. И, самое главное, точность даже в 32 разрядной сетке представления числа является излишней. Требуемая величина восстанавливаемых амплитуд не выходит за пределы 16 разрядного представления.

Тогда возникает альтернативное предложение, а почему бы не использовать уже вычисленные значения синусов по таблице [4], которую в школах и ВУЗах интенсивно использовали в прошлом веке. При наличии большого объема оперативной памяти использование таблицы значений для 90° потребует 180 байт, и даже с более «мелкой» градацией до 60 минут между градусами потребует для записи $90 \cdot 60 = 5400$ чисел, в которых значение синуса определяется четырехзначным числом, т.е. числа 0-10000 (2710h) включительно. Для этого достаточно 2-х байт, тогда для всей таблицы необходимо всего-то 10800 (4E20h) байт памяти. При наличии практически гигабайтной (1 Гб) оперативной памяти такие издержки в пределах килобайт ничтожно малы.

Вычисления амплитуд по таблице Брадиса [4] охватывают диапазон $0-90^\circ$, что составляет четверть волны. Оставшиеся три четверти волны вычисляются по той же таблице, но с изменением знака и (или) последовательности получения результата от $[-90^\circ - 0^\circ]$ $[0^\circ - 90^\circ]$ $[90^\circ - 0^\circ]$ $[0^\circ - -90^\circ]$.

Далее нас будет интересовать охват и определение значения амплитуды максимальных и минимальных частот (или максимальной длины волны).

Для **максимальных частот** и минимальной длины волны (от $2 \div 10$ отсчетов) 90° -ый фрагмент будет делиться на градусные участки длиной в десятки градусов. В этом случае для трехзначного числа достаточно для получения отличающихся значений SIN.

Рассмотрим минимальную частоту. С учетом восприятия слухом до 20 Гц (1/20 сек) для малых частот у нас есть дополнительная возможность фиксировать еще меньшие частоты близкие к 1 Гц (например, при значении 22050 отсчетов – один из используемых вариантов фиксации амплитуды для 1 секунды записи звука). Таким образом, для наименьшей частоты, но в то же время для максимально «большой» длины волны ($5400 + 5400 + 5400 + 5400 = 21600$) в 21600 отсчетов (это меньшее число отсчетов относительно 1 Гц колебания, т.к. $21600 < 22050$) нужно охватить, соответственно для 1/4 отсчетов рассматриваемой длины волны, (т.к. работаем по фрагментам 90°) 5400 разных значений Sin угла в целых числах из таблицы Брадиса. Это случай максимально требуемой точности от этих таблиц, именно в этом случае потребуется максимальная точность извлекаемых значений из таблицы Брадиса. так как меньшая длина волны, включает меньшее число отсчетов и тогда потребуется меньшее число обращений в таблицу.

При 44100 отсчетах на 1 секунду (другой вариант квантования), это составляет охват 1/2 секунды. Таким образом, для предела 20 Гц (это 20 колебаний в секунду) такого уровня точности вполне достаточно как для 44,1 кГц, так и для 22,05 кГц. Возможно, потребуется точность представления значений, увеличенных в 10 раз. Тогда охват 1/4 секунды составит 54000 отсчетов (D2F0h), что увеличивает требуемую память под таблицу в 10 раз большую, но и это представление не выходит за пределы 2-х байтового представления значения SIN.

Стоит заметить, что для байтовой предельной размерности амплитуды это однозначно излишне по следующей причине: если $5400 = 4E20h / 128$ (для 128 значений амплитуд для 90°) что составляет 42,18 минуты или округленно 10 из указанной таблицы будут требоваться значения из 128 строк таблицы. Если рассматривать для 2-х отсчетов такое представление, на четверть волны ($0^\circ - 45^\circ - 90^\circ$) для максимального значения амплитуды, равного 128, разбивает эти значения на три варианта 0–71–128 ($\Delta = 71, \Delta = 57$; средняя $\Delta = 64$).

При представлении амплитуды в пределах 2-х байт или 1 слова имеем диапазон представлений $0 - 65536 \cdot 1/4 = 16384$. Тогда $(16384 \text{ макс. кол-ва отсчетов на } 90^\circ) / (5400 \text{ значений SIN})$ немного больше 3. различных значений амплитуд на 1 минуты (1 отсчет) точности квантования. Начиная с $1/3$ Гц, мы будем иметь достаточно точные значения.

Идеальное соотношение было бы не более 5400 отсчетов на 5400 значений SIN для вычисления амплитуды.

Таким образом, всё эти рассуждения будем иметь в виду при восстановлении амплитуд по характерным точкам. Далее рассмотрим задачу и алгоритм её решения через таблицу на основе таблицы Брадиса [3] для вычисления SIN.

1. Практическая постановка задачи

Имеется первое значение экстремума (максимума или минимума S_A) и второе значение экстремума (минимума или максимума S_B). И это соответствует одной второй фрагмента или цикла волны. Известно количество отсчетов N между ними. Требуется восстановить промежуточные значения амплитуд.

Здесь и далее используется **фрагмент волны**, так при оцифровке мы фиксируем лишь отдельные значения амплитуд для суперпозиции (множества) волн, что далеко не соответствует «красивой» кривой функции SIN. Поэтому восстанавливая, мы сводим наше восстановление к идеальной «тени» в виде функции SIN для реальной функции поведения волнового процесса. При этом восстанавливается $1/2$ волны. Вторая восстанавливаемая половинка, которая должна быть симметрична первой половинке, относится к значению SIN уже с другим знаком. Таким образом, мы будем получать аппроксимацию некоторой синусоиды, которая близка по своему поведению полуволне реальной волны. Фактически, такой фрагмент, приближаемый традиционным значением SIN, это уже некоторый **идеальный образ** реальной полуволны, который в принципе может и другую «идеальную» функцию поведения для других процессов колебательного характера. И это в случае отсутствия понимания аналитического поведения функции рассматриваемого фрагмента, можно определить, скорей всего, экспериментально, уменьшая шаг квантования и подбирая **«идеальную» функцию** численными методами [2].

2. Алгоритм решения

Введем обозначения:

N – Количество отсчетов для $1/2$ фрагмента волны.

$\Delta_s = S_A - S_B$ – дельта разброса 1/2 фрагмента соответствующего половине цикла волны.

$I = I_1 + I_2 = \Delta_s / 2$ – это дельта разброса на 1/4 фрагмента волны, где:

$$I_1 = \left| \Delta_s / 2 \right| \approx 1/4 \text{ фрагмента волны, а } n_1 = \left| N / 2 \right|$$

$$I_2 = \Delta_s - I_1 \approx 1/4 \text{ фрагмента волны, а } n_2 = N - n_1$$

При I четном $I_1 = I_2$

$\Delta_{i1} = 5400 / n_1$ и $\Delta_{i2} = 5400 / n_2$ – значение шага в градусах и минутах для 1/4 фрагмента волны с учетом количества отсчетов.

$$\Delta_{s1} = I_1 / n_1 \text{ – среднее приращение амплитуды}$$

$$\Delta_{s2} = I_2 / n_2 \text{ – среднее приращение амплитуды}$$

Тогда схематично для первой части 90° углового интервала, вычисление промежуточных значений будет выглядеть следующим образом и представлено в табл. 1.

Таблица 1

Вычисление промежуточных значений

Шаг итерации	Математическая формула	Табличная форма вычисления в целых числах
$\lambda_0 = 0$	$S_0 = 0$	$S_0 = 0$
$\lambda_1 = \lambda_0 - \Delta_i$	$S_1 = \left I_1 \cdot \sin \lambda_1 \right $	$S_1 = \left \left(I_1 \cdot \text{Brad } \lambda_1 \right) / 10000 \right $
...
$\lambda_{n-1} = \lambda_0 + \Delta_i$	$S_{n-1} = \left I_1 \cdot \sin \lambda_{n-2} \right $	$S_{n-1} = \left \left(I_1 \cdot \text{Brad } \lambda_{n-2} \right) / 10000 \right $
$\lambda_n = 5400$	$S_n = I_1, \text{ м. к. } \sin \lambda_n = 1$	$S_n = I_1$

Наглядное объяснение алгоритма представлено на рисунке 1.

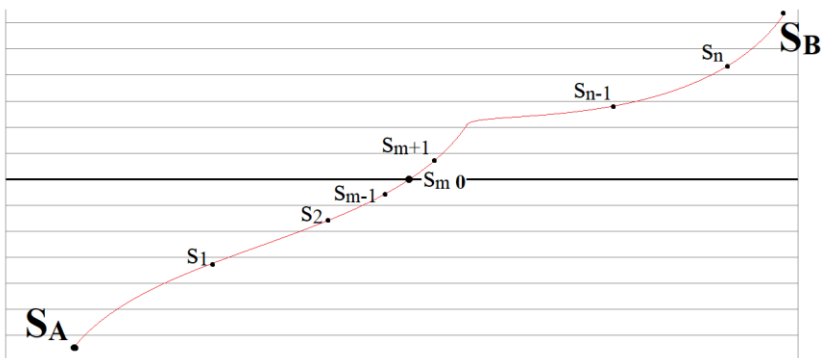


Рис. 1. Наглядное представление алгоритма

3. Процедура восстановления полуволны по характерным точкам

Для восстановления промежуточных значений амплитуд в виде целых чисел без знака была реализована процедура с именем **F@enik** для библиотеки **FEN.dll**, которая для получения максимальной эффективности была написана на языке Ассемблера и использовала табличный способ вычислений геометрических функций. Процедура **F@enik** восстанавливает значения амплитуд в виде целых чисел без знака для волн с длиной до 21600 отсчетов на один фрагмент волны. При этом восстановление производится фрагментами до полуволны. Для полуволны это значение 10800 отсчетов, если больше 10800, то производится выход без результата с сообщением об ошибочном параметре (N) на EAX = -1

Обращение производится к библиотеке **DFEN.dll** и представлен в листинге 1.

Листинг

Обращение к библиотеке DFEN.dll

```
F@enik [Sa], [Sb], [N], MFen - для языка C.  
invoke F@enik, [Sa], [Sb], [N], MFen - для Ассемблера
```

Параметры для обращения:

Sa – это 1-е целочисленное значение амплитуды без знака в точке локального экстремума или точки А.

Sb – это 2-е целочисленное значение амплитуды без знака в точке локального экстремума или точки В.

N – количество отсчетов для полуволны между локальными экстремумами.

MFen – соответствует адресу массива в виде последовательности размерностью в слово (16 разрядов), куда следует положить значения амплитуд, кроме Sa, Sb.

Код ответа: EAX = -1 для превышения числа отсчетов более 10800. В остальных случаях любое другое значение.

Учитывается четность или нечетность количества отсчетов. Для четного числа отсчетов виртуальное нулевое значение (нулевое значение без учета сдвига относительно первого значения амплитуды) будет соответствовать средней точке. Для нечетного числа отсчетов виртуальное нулевое значение (нулевое значение без учета сдвига относительно первого значения амплитуды) будет соответствовать значению между двумя «центральными» точками. Восстановление производится на основе таблицы Брадиса [4].

Возможные случаи представлены на рисунке 2.

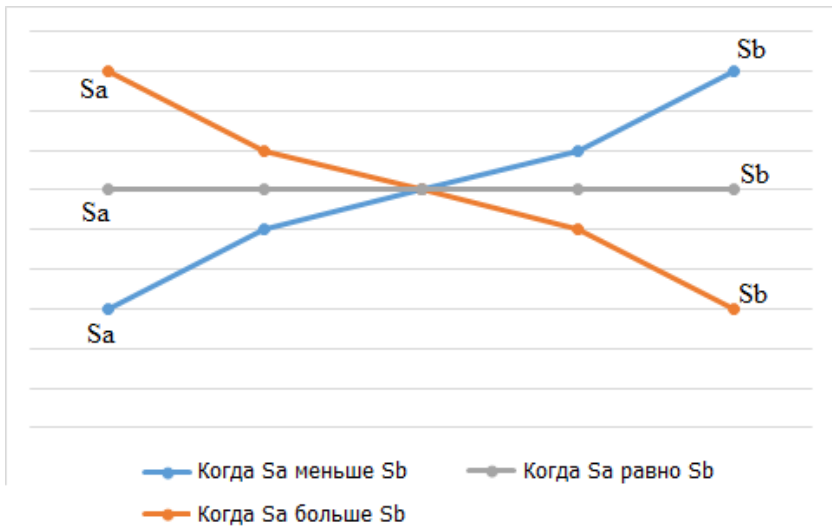


Рис. 2. Всевозможные случаи соотношения характерных точек

Аналогичная процедура F@sik [Sa], [Sb], [N], MFen для языка C была реализована по стандартной схеме обращения к сопроцессору, где обращение сводится к инициализации математического сопроцессора, вычисления с его помощью и преобразование числа с плавающей точкой в целое.

4. Сопоставление использования предложенных алгоритмов восстановления

При анализе способов измерения оказалось, что не всё так однозначно при использовании службы учета времени для замера времени работы двух предложенных вариантов было апробировано несколько временных измерителей и реализованных на различных архитектурах процессоров фирмы Intel. Так, изначально было предложено четыре варианта измерителей:

Библиотека Chrono (C++11) – высокоточный таймер, реализованный через обращение к часам реального времени (RTC). Для работы необходим предустановленный набор библиотек C++11 года.

Часы реального времени (RTC) – реализован на южном мосте современной вычислительной машины или как отдельный элемент. Имеет высокую погрешность ввиду генерации прерывания равной 1 секунде или в пределах погрешности самого генератора.

Таймер событий высокой точности (HPET) – так же реализован на чипсетах южного моста. Более точен, чем RTC, однако имеет высокую погрешность при поднятии частоты чипсета или шины PCIe.

Счётчик реального времени (TSC) – реализован на процессоре, вводит время с момента запуска компьютера. Счётчик увеличивается даже при глубоком сне вычислительной системы (ACPI C3). Выводит время в тактах независимо от астрономического времени.

После множества экспериментов показывающих большой разброс значения времени выполнения алгоритмов решено использовать счётчик реального времени (TSC) по причине его более высокой устойчивости значений при выполнении замеров. Для обращения к счётчику реального времени была использована инструкция RDTSCP (в системах до 2007 года – RDTSC).

Для тестирования обоих вариантов решено использовать современную операционную систему MS Windows 10 редакции 1709. Алгоритм выполнения сравнения выглядит следующим образом:

1. Отключение в тестируемой системе ненужных (лишних) процессов и служб.

2. Подготовительный запуск тестируемой программы для «прогрева» кэша процессора.

3. Через командную строку, с ожиданием завершения предыдущего экземпляра программы и максимальным приоритетом (приоритетом реального времени) запускает комплекс сравнения. Количество циклов восстановления и количество запусков устанавливается пользователем.

4. Полученные значения суммируются и делятся на количество запусков комплекса сравнения.

Стоит так же добавить, что для достижения максимально точных результатов рекомендуется проделать операции настройки в BIOS'е компьютера, однако не во всех тестируемых системах это возможно, а результаты будут иметь минимальную разброс значений при одних и тех же данных.

Тестирование проводилось для нескольких объемов входных данных 1000, 10000 и 100000 значений амплитуд. На вход обоим вариантам программ поступает соответствующее количество данных. После отработки программы программа-измеритель выводит значение, которое потребовалось для указанного количества данных.

Таким образом были получены результаты, приведённые в таблице 2. Ускорение по отношению к стандартному алгоритму вычисления SIN было троекратным, т. е. в таблице указан коэффициент ускорения для разных процессоров с архитектурой команд Intel. При тестировании были использованы процессоры как от самого младшего класса для встраиваемых систем, так и до процессоров серверного сегмента.

Таблица 2

Отношение стандартного алгоритма восстановления значения амплитуд по характерным точкам к табличному алгоритму

	1000 значений	10000 значений	100000 значений
Atom Z3736F	2	2,247	3,385
Pentium B960	3,589	3,291	3,379
Core i5 3210M	3,46	3,393	3,385
Xeon E5 2670v2	3,219	3,322	3,329

На основе изложенных данных можно сделать некоторые выводы, а именно:

1. Восстановление на основе таблицы имеет преимущество по скорости выполнения по сравнению с восстановлением стандартным средствам информационных волн.

2. Восстановление на основе таблицы может быть вариативным. То есть, возможно восстановление и получение информационных структур не только по таблице Брадиса, но и по той таблице, которая будет лучше восстанавливать информационные структуры для конкретной волновой структуры.

3. Реализованный способ замера продемонстрировал свою устойчивость результатов при различных запусках и может быть использован для измерения производительности других алгоритмов.

Список литературы

1. Мелентьев П. В. Приближенные вычисления / П. В. Мелентьев; М. : Госиздательство физматлитературы, 1962. – 388 с.
2. Балакирев, Н.Е. Качественная оценка и мера при распознавании информационного содержания волн / Н. Е. Балакирев // Материалы XIX Международной научно-методической конф. «Информатика: проблемы, методология, технологии» (Воронеж, 14-15 февраля 2019 г.). – Воронеж, 2019. – С. 1435- 1440.
3. Intel® 64 and IA-32 architectures software developer's manual combined volumes 2A, 2B, 2C, and 2D: Instruction set reference, A-Z [Электронный ресурс] : документация фирмы Intel. – Режим доступа : <https://software.intel.com/content/dam/develop/external/us/en/documents-tps/325383-sdm-vol-2abcd.pdf>
4. Брадис В. Четырёхзначные математические таблицы для средней школы / В. Брадис; М.: Учпедгиз, 1953