

## УЛУЧШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РОБАСТНОГО АЛГОРИТМА ЭХОКОМПЕНСАЦИИ НА ЦСП ADSP-21160

Назаровский А.Е., Тараканов А.Н.

Ярославский государственный университет им. П.Г. Демидова  
150000, Россия, Ярославль, ул. Советская, 14, Тел. (0852) 79-77-75. E-mail: dcslab@uniyar.ac.ru

Серьезной проблемой в междугородней и международной телефонной связи является возникновение паразитного эхосигнала из-за несогласованности сопротивлений в месте преобразования четырехпроводной линии в двухпроводную. В настоящее время задачу подавления эхосигналов в телефонии можно решать с помощью эхокомпенсаторов. Эхокомпенсаторы обычно строятся на базе цифровых сигнальных процессоров (ЦСП), которые позволяют эффективно реализовывать цифровые адаптивные фильтры. При этом наиболее часто используется адаптивный алгоритм НМНК ввиду простоты его реализации и малых вычислительных затрат. Основные характеристики устройств эхокомпенсации определяются рекомендациями МСЭ-Т G.165 и G.168 [1,2].

В ситуации одновременного разговора нормальная работа алгоритма НМНК нарушается. Этого можно избежать, модифицируя данный алгоритм с помощью методов робастной статистики, что рассмотрено в работе [3]. Как показывалось в [4] одним из узких мест модифицированного алгоритма является процедура вычисления медианного значения в буфере ошибок. Целью данной работы является уменьшение вычислительной сложности процедуры подсчета медианы применительно к модифицированному алгоритму НМНК.

Для задачи вычисления оценки масштаба необходимо для каждого нового обрабатываемого отсчета пересчитывать значение медианы от буфера ошибок [4]. Во всем алгоритме это самая трудоемкая в вычислительном плане процедура, требующая серьезной оптимизации для удовлетворения требований работы алгоритма в реальном времени. Кроме того, необходимо учитывать специфику задачи вычисления оценки масштаба, а именно:

- Алгоритм не должен нарушать порядок следования отсчетов ошибки в буфере ошибок.
- Алгоритм должен учитывать тот факт, что буфер ошибок организован по принципу FIFO, а, следовательно, при обработке каждого последующего отсчета, в буфере ошибок появляется только одно новое значение, а самое старое удаляется.

Существует несколько общепринятых методов вычисления медианы в массиве из  $N$  элементов, которые различаются по некоторым параметрам, таким как:

- Вычислительная сложность
- Затраты памяти (например, под вспомогательный массив, в который копируются значения из буфера ошибок, при использовании алгоритмов, меняющих порядок элементов, или под стек, используемый рекурсивными алгоритмами)
- Простота реализации алгоритма

Среди известных методов можно выделить:

1. вычисление медианы при помощи сортировки (например, используя алгоритм «быстрой сортировки» “quicksort”, или аналогичные). Обычно, это наиболее медленные алгоритмы, кроме того, они меняют порядок элементов в массиве, и поэтому в приложениях они практически не используются. Единственным преимуществом данных алгоритмов является сравнительная простота реализации.

2. вычисление медианы путем поиска  $N/2$  порядковой статистики (например, алгоритм “quickselect”). Это самые быстрые в среднем алгоритмы вычисления медианы. Лучшие из них имеют вычислительную сложность  $O(N)$  [5,6]. Недостатком является нарушение порядка элементов в исходном массиве после работы алгоритма. Поэтому в задаче вычисления оценки масштаба перед работой алгоритма требуется копирование данных из буфера ошибок во временный массив.

3. алгоритмы поиска медианы, которые не меняют порядок элементов в исходном массиве (например, метод Торбена [7]). К сожалению, этот класс алгоритмов является более затратным в вычислительном плане, чем предыдущий.

Исходя из указанного выше, ни один из широко известных алгоритмов не отвечает полностью специфике задачи вычисления оценки масштаба в буфере ошибок. Поэтому была предложена реализация алгоритма, учитывающего особенности данной задачи.

Новый алгоритм предназначен для вычисления медианного значения выборки фиксированной длины  $N$ . При этом предполагается, что на каждом шаге в выборке меняется только одно значение, и значение медианы требуется вычислять также на каждом шаге.

Основной идеей алгоритма является включение значений в массив длиной  $N$  в упорядоченном виде (в порядке возрастания или убывания) и своевременное удаление этих значений из данного массива по прошествии  $N$  шагов. Медианное значение соответственно на каждом шаге будет равно значению  $N/2$  элемента данного массива. Для обеспечения работы алгоритма требуется также целочисленный массив длиной  $N$ , хранящий порядок удаления выборок из основного массива, организованный по принципу FIFO, т.е. «первый пришел, первым вышел».

Обозначим основным массив для вычисления медианы как  $A$ , а массив с позициями выборок как  $B$ . Тогда работу алгоритма можно описать следующим образом:

1. Инициализация:
  - 1.1. Основной массив заполняется одинаковыми начальными значениями.
  - 1.2. Массив позиций заполняется нулевыми значениями
2. Для каждого нового значения  $x$  повторять шаги 3-5:
3. Удаление самого «старого» значения из массива  $A$ :
  - 3.1. Удаление из массива  $A$  элемента с индексом  $K=B[0]$ , при этом все элементы  $A$  с индексом большим или равным  $K$  сдвигаются так, что  $A[i]=A[i+1]$  для  $K \leq i \leq N-2$  (если  $K=N-1$ , то ничего сдвигать не требуется, т.к. это последний элемент массива  $A$ ).
  - 3.2. Сдвиг всех элементов из массива  $B$ , так что  $B[i]=B[i+1]$ , при этом  $B[0]$  теряется.
  - 3.3. Каждый элемент полученного массива  $B$  со значением большим, чем позиция удаляемого элемента  $K$  дополнительно уменьшается на 1.
4. Вставка нового значения  $x$  в массив  $A$ :
  - 4.1. Вставка нового элемента  $x$  в массив в соответствии с его значением (например, в порядке возрастания элементов в массиве  $A$ ), в некоторую позицию  $L$ , при этом все элементы  $A$  с индексом больше  $L$  сдвигаются так, что  $A[i]=A[i-1]$  для  $L+1 \leq i \leq N-1$  (при  $L=N-1$ , соответственно, сдвигать ничего не требуется).
  - 4.2. Каждый элемент полученного массива  $B$  со значением большим или равным, чем позиция вставляемого элемента  $L$  дополнительно увеличивается на 1.
  - 4.3. Значение последнего элемента массива  $B$  устанавливается равным позиции вставляемого значения  $L$  ( $B[N-1]=L$ ).
5. Получение медианного значения из  $N/2$  элемента массива  $A$  ( $\text{median}=A[N/2]$ ).

Проверка работы алгоритма робастного НМНК с новой процедурой вычисления медианы проводилась на базе оценочной платы (evaluation board) ADSP-21160 EZ-KIT Lite, основой которой является сигнальный процессор SHARC ADSP-21160, серийно выпускаемый компанией Analog Devices. Данный ЦСП является высокопроизводительным процессором с 32-разрядной арифметикой с плавающей точкой [8]. Выбор процессора обусловлен высокими требованиями модифицированного алгоритма к скорости вычислений, большим объемом внутренней памяти и отсутствием проблем связанных с квантованием коэффициентов фильтров и переполнением.

В связи с тем, что указанный ЦСП содержит весьма большой объем внутрикристалльной памяти (4Мбит) а также для простоты реализации алгоритма, все отсчеты сигналов при работе хранились в цифровом виде в памяти сигнального процессора. Результаты работы также записывались в массив в памяти процессора и затем при помощи отладчика пересылались на компьютер. Программа, реализующая указанный алгоритм была написана на языке C с использованием оптимизирующего компилятора из пакета VisualDSP++. Помимо новой процедуры вычисления медианы, программа содержит в своем составе модель дифференциальной системы, адаптивный КИХ-фильтр и детектор двойного разговора (ДДР) по алгоритму Гейгеля [3]. Дифференциальная система моделируется фильтром с конечной импульсной характеристикой (КИХ) 127 порядка. Адаптивный фильтр также является КИХ-фильтром 127 порядка. При моделировании использовались отрезки тестового сигнала, задаваемого в соответствии с рекомендацией МСЭ G.165. Ситуация встречного разговора длится с 2000 по 2200 отсчет. Все отсчеты представлены 16-битными целыми числами. Выходные результаты имеют 32-битную точность.

На рис. 1 представлены зависимости количества машинных циклов процессора для выполнения процедуры подстройки коэффициентов адаптивного фильтра совместно с вычислением медианы по новому алгоритму при разных значениях длины буфера ошибок  $P$ .

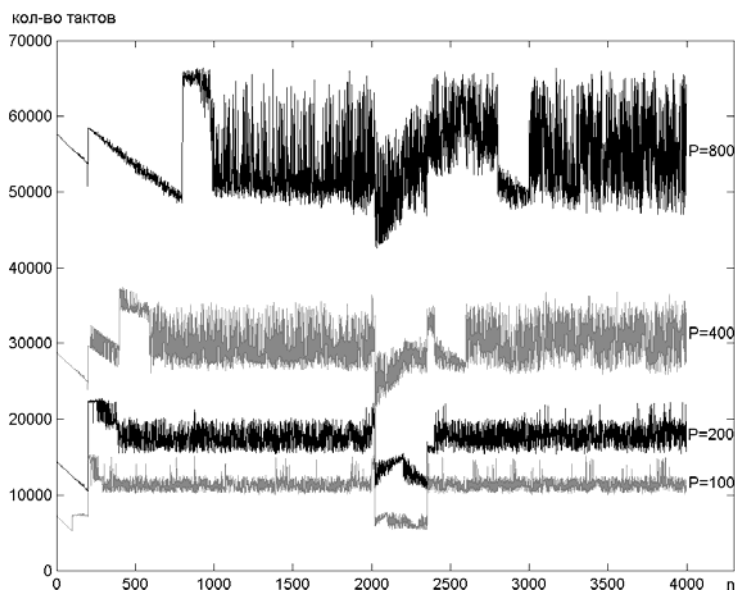


Рис. 1. Зависимость количества тактов в процедуре адаптации от длины буфера ошибок P при обработке тестового сигнала

Предложен новый алгоритм вычисления медианы, учитывающий особенности задачи расчета оценки масштаба в буфере ошибок, возникающей при реализации робастных алгоритмов эхокомпенсации. Рассматриваемый алгоритм имеет вычислительную сложность  $O(N)$ . При этом затраты памяти для работы алгоритма меньше, чем в алгоритме «quickselect», кроме того, предлагаемый алгоритм является более простым для реализации, чем «quickselect». Работа нового алгоритма промоделирована на базе сигнального процессора ADSP-21160.

#### Литература

1. ITU-T Recommendation G.165. Echo cancellers. 1993.
2. ITU-T Recommendation G.168. Digital network echo cancellers. 2000.
3. Брюханов Ю.А., Тараканов А.Н. Усовершенствование адаптивного алгоритма эхо-компенсации // Электросвязь, 2003. №9. С. 38-39.
4. Назаровский А.Е., Тараканов А.Н. Мосеев А.Л. Моделирование работы робастного алгоритма эхо-компенсации на цифровом сигнальном процессоре ADSP-21160 // 6 Международная научно-техническая конференция «Цифровая обработка сигналов и ее применение» DSPA-2004 т.1, стр 286-288
5. Ахо А.А., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. М.: Вильямс, 2000.
6. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 2000.
7. Devillard Nicolas. Fast median search: an ANSI C implementation, 1998
8. ADSP-21160 SHARC Hardware Reference. Analog Devices Inc. November 1999

