

В. И. Д ж и г а н

**ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ
В RLS-АЛГОРИТМАХ АДАПТИВНОЙ
ФИЛЬТРАЦИИ**

Рассмотрен способ представления RLS-алгоритмов адаптивной фильтрации со скользящим окном и регуляризацией корреляционной матрицы адаптивного фильтра, позволяющий реализовывать такие алгоритмы и их частные случаи с помощью параллельных вычислений. Данные алгоритмы математически эквивалентны известным одноименным последовательным и параллельным алгоритмам адаптивной фильтрации. Результаты моделирования демонстрируют работоспособность рассмотренных алгоритмов.

Цифровая обработка сигналов (ЦОС) — одна из областей современной радиотехники — имеет в своем составе ряд научно-технических направлений. Одно из таких направлений — это адаптивная фильтрация [1]. Адаптивные фильтры используются для решения радиотехнических задач широкого класса [2–4]. Качество решения этих задач определяется применяемыми в адаптивных фильтрах алгоритмами. В большинстве таких фильтров в основном используются простейшие (с точки зрения числа арифметических операций) градиентные алгоритмы по критерию наименьшего среднеквадратического отклонения (Least Mean Squares, LMS) или нормализованные LMS (Normalized LMS, NLMS) алгоритмы. Такие алгоритмы привлекательны своей низкой вычислительной сложностью (числом арифметических операций, необходимых для выполнения одной итерации алгоритма в течение одного периода дискретизации обрабатываемых сигналов), но имеют некоторые недостатки, среди которых медленная сходимости, проявляющаяся при обработке нестационарных сигналов, и зависимость длительности переходного процесса и остаточной ошибки на выходе адаптивного фильтра от шага сходимости. Такие недостатки отсутствуют в рекурсивных алгоритмах по критерию наименьших квадратов (Recursive Least Squares, RLS) [5]. Однако вычислительная сложность RLS-алгоритмов значительно превышает вычислительную сложность простых алгоритмов адаптивной фильтрации, что на протяжении многих лет служило весомым препятствием на пути применения сложных алгоритмов в приложениях.

Возможность реализации того или иного алгоритма ЦОС определяется производительностью используемых при этом цифровых сигнальных процессоров (ЦСП). Производительность современных ЦСП уже позволяет эффективно реализовывать вычислительно сложные алгоритмы. Кроме того, современные большие интегральные схемы (БИС) могут содержать несколько ЦСП в одном кристалле [6], что позволяет создавать компактные устройства ЦОС, функционирующие на основе использования параллельных вычислений. Алгоритмы адаптивной фильтрации также могут строиться с использованием параллельных вычислительных процедур. Некоторые из таких алгоритмов рассмотрены в работах [7–14]. Эти алгоритмы имеют достаточно сложное математическое описание.

В настоящей статье рассмотрен простой способ описания RLS- и линейно-ограниченных (Linearly Constrained, LC) RLS-алгоритмов адаптивной фильтрации, позволяющий реализовывать такие алгоритмы с помощью параллельных вычислений, для многоканальных фильтров с неодинаковым числом комплексных весовых коэффициентов в каналах. Возможность использования неодинакового числа весовых коэффициентов в каналах позволяет уменьшать вычислительную нагрузку на ЦСП, если в соответствии с природой решаемой задачи одинаковое число весовых коэффициентов в каналах адаптивного фильтра не требуется. Оценка корреляционной матрицы входных сигналов рассматриваемого фильтра осуществляется на скользящем окне (Sliding Window, SW) с экспоненциальным взвешиванием. Кроме того, при обращении этой матрицы (математической операции, присутствующей в явном или неявном виде во всех RLS-алгоритмах) используется динамическая регуляризация [15] в целях обеспечения устойчивости алгоритма. Многоканальные адаптивные алгоритмы без линейных ограничений, нерегуляризованные алгоритмы со скользящим окном, регуляризованные и нерегуляризованные алгоритмы с бесконечным окном (Prewindowed, PW), одноканальные версии таких алгоритмов или алгоритмы с действительными весовыми коэффициентами — все это частные случаи данного общего решения.

Получение вычислительных процедур большинства RLS-алгоритмов основано на использовании леммы об обращении матриц [1]. С ее помощью осуществляется рекурсивное обращение корреляционной матрицы адаптивного фильтра. Обеспечение следящих свойств таких фильтров при обработке нестационарных сигналов часто осуществляется с помощью экспоненциального взвешивания сигналов и (или) применения скользящего окна. В силу ограниченного числа отсчетов, используемых при оценке корреляционной матрицы адаптивного фильтра на скользящем окне, SW RLS-алгоритмы в ряде случаев мо-

гут становиться нестабильными. Как уже упоминалось, эффективным способом обеспечения стабильной работы адаптивных фильтров является динамическая регуляризация корреляционной матрицы. Регуляризация может применяться во всех RLS-алгоритмах: одноканальных и многоканальных, с бесконечным или скользящим окном, включая быстрые (вычислительно эффективные) RLS-алгоритмы. Скользящее окно увеличивает вычислительную сложность RLS-алгоритмов примерно в 2 раза по сравнению с алгоритмами с бесконечным окном. Динамическая регуляризация также увеличивает примерно в 2 раза вычислительную сложность алгоритмов по сравнению с алгоритмами без регуляризации.

В случае использования нескольких ЦСП, вычислительную нагрузку при реализации сложных алгоритмов адаптивной фильтрации можно уменьшить с помощью параллельных вычислений. Наиболее привлекательным является распараллеливание вычислений, связанных с существующими потоками независимых данных. При этом отсутствуют дополнительные вычислительные затраты для разделения данных на параллельные потоки. В SW RLS- и в регуляризованных PW RLS-алгоритмах имеется два ($F = 2$), а в регуляризованных SW RLS-алгоритмах — четыре ($F = 4$) независимых потоков входных данных, обусловленных модификацией корреляционной матрицы адаптивного фильтра. В известных последовательных RLS-алгоритмах в результате последовательного применения леммы об обращении матрицы эти потоки обрабатываются последовательно. Рассматриваемые далее параллельные RLS-алгоритмы базируются на возможности независимой обработки таких потоков.

Основные приемы получения параллельных RLS-алгоритмов.

Для получения RLS-алгоритмов, допускающих параллельные вычисления, могут быть использованы приемы, приведенные в работе [7]. В этой работе был рассмотрен способ построения одноканальных SW RLS- и быстрых SW RLS-алгоритмов адаптивной фильтрации с действительными весовыми коэффициентами, которые могут быть реализованы с помощью двух параллельных процессоров. Подобные многоканальные SW RLS- и LC SW RLS-алгоритмы с комплексными весовыми коэффициентами были получены в работе [9], а многоканальные регуляризованные PW RLS-алгоритмы — в работе [10]. Многоканальные регуляризованные SW RLS-алгоритмы для параллельной реализации с помощью четырех процессоров были рассмотрены в работах [8, 11–14]. Все эти алгоритмы имеют достаточно сложное математическое описание, являющееся следствием использования сложного выражения леммы для обращения матрицы, модифицируемой на каждой итерации алгоритма адаптивной фильтрации за счет двух или

четырёх потоков обрабатываемых данных. Такая лемма получается путем последовательного двукратного или четырехкратного применения леммы об обращении матрицы [1], модифицируемой за счет одного потока обрабатываемых данных.

Далее рассматривается простой способ описания вычислительных процедур параллельных RLS-алгоритмов адаптивной фильтрации. При таком описании вычислительные процедуры регуляризованных алгоритмов с бесконечным окном, нерегуляризованных или регуляризованных алгоритмов со скользящим окном имеют один и тот же вид, близкий к виду нерегуляризованных RLS-алгоритмов с бесконечным окном. Особенностью представляемого описания параллельных RLS-алгоритмов является замена ряда скалярных и векторных переменных, свойственных последовательным алгоритмам, на векторные и матричные переменные, размерность которых пропорциональна числу F независимых потоков входных данных адаптивного фильтра. Вычисления элементов векторов и столбцов таких матриц не зависят друг от друга, так как определяются только одним из ранее отмеченных независимых потоков входных данных адаптивного фильтра, а значит, могут быть выполнены с помощью F параллельных процессоров.

Блок-схема M -канального адаптивного фильтра, постановка задачи адаптивной фильтрации по критерию наименьших квадратов, а также основные обозначения и определения переменных, используемых в настоящей статье, могут быть найдены в работе [8]. Такой фильтр может иметь неодинаковое число комплексных весовых коэффициентов в каждом из каналов. Наиболее общий случай адаптивной фильтрации с помощью такого фильтра — это LC-фильтрация, при которой накладываются дополнительные ограничения вида $\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J$, где \mathbf{C}_{NJ} и \mathbf{f}_J — матрица и вектор линейных ограничений [16–18], $\mathbf{h}_N(k)$ — вектор весовых коэффициентов адаптивного фильтра.

Решением данной задачи является вектор весовых коэффициентов адаптивного фильтра [18]

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) + \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)], \quad (1)$$

где $\mathbf{R}_N(k)$ и $\mathbf{r}_N(k)$ — корреляционная матрица обрабатываемых сигналов и вектор взаимной корреляции между компонентами вектора входных сигналов адаптивного фильтра $\chi_N(k)$ и требуемым сигналом $d(k)$; k — номер итерации алгоритма; $N = \sum_{m=1}^M N_m$ — общее число

весовых коэффициентов адаптивного фильтра; N_m — число весовых коэффициентов в m -м канале. Здесь и далее один нижний индекс N , J или F используется для обозначения размерности векторов с числом элементов N , J , F и квадратных матриц с числом элементов $N \times N$, $J \times J$ и $F \times F$, а два индекса NJ и NF — для обозначения размерности прямоугольных (нетранспонированных) матриц с числом элементов $N \times J$ и $N \times F$.

Первое слагаемое в уравнении (1) является решением задачи RLS адаптивной фильтрации при отсутствии ограничений. Второе слагаемое обусловлено линейными ограничениями. В соответствии с этим, RLS-алгоритмы, с помощью которых осуществляется рекурсивное вычисление вектора весовых коэффициентов адаптивного фильтра (1), также состоят из двух вычислительных процедур. При этом, в зависимости от числа слагаемых в уравнениях оценки матрицы $\mathbf{R}_N(k)$ и вектора $\mathbf{r}_N(k)$ [8], а также интервалов накопления обрабатываемых данных, участвующих в формировании этих оценок, уравнение (1) может быть использовано для получения PW RLS-алгоритмов, регуляризованных PW RLS-алгоритмов, SW RLS-алгоритмов или регуляризованных SW RLS-алгоритмов. Последовательное применение леммы [1] для обращения матрицы $\mathbf{R}_N(k)$ позволяет получить соответствующие последовательные RLS-алгоритмы [5], а использование приемов [7] — параллельные алгоритмы, ориентированные на реализацию с помощью двух или четырех ЦСП [7–14].

Лемма об обращении матриц [1] имеет вид

$$\mathbf{R}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}\mathbf{A}^{-1}\mathbf{D}\mathbf{B}^{-1},$$

где $\mathbf{A} = \mathbf{D}\mathbf{B}^{-1}\mathbf{C} + 1$ и $\mathbf{R} = \mathbf{B} + \mathbf{C}\mathbf{D}$, а \mathbf{C} и \mathbf{D} — векторы (это позволяет использовать данную лемму для рекурсивного обращения матрицы $\mathbf{R}_N(k)$).

Общий вид леммы

$$\mathbf{R}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}\mathbf{A}^{-1}\mathbf{D}\mathbf{B}^{-1}, \quad (2)$$

где $\mathbf{A} = \mathbf{D}\mathbf{B}^{-1}\mathbf{C} + \mathbf{S}$, а \mathbf{C} и \mathbf{D} — матрицы, приведен в работе [19]. Уравнение (2) лежит в основе рассматриваемых далее RLS-алгоритмов. Для использования этого уравнения необходимо сформировать матрицы $\mathbf{C} = [\mathbf{y}, \mathbf{x}, \mathbf{z}, \mathbf{v}] = \mathbf{X}_{NF}(k) = [\mu^{0,5}\boldsymbol{\chi}_N(k-L), \boldsymbol{\chi}_N(k), \mu^{0,5}\boldsymbol{\xi}\boldsymbol{\rho}_N(k-L), \boldsymbol{\xi}\boldsymbol{\rho}_N(k)]$; $\mathbf{D} = \mathbf{C}^H$ и $\mathbf{S} = \text{diag}(-1, 1, -1, 1)$. Число столбцов F матрицы \mathbf{X}_{NF} и их вид определяют возможные разновидности RLS-алгоритмов.

Уравнение (2) позволяет получить параллельный RLS-алгоритм адаптивной фильтрации.

Шаг 0:	Инициализация : $\chi_N(0) = \mathbf{0}_N, \dots, \chi_N(0 - L + 1) = \mathbf{0}_N,$ $\rho_N(0) = \mathbf{0}_N, \dots, \rho_N(0 - L + 1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0 - L + 1) = 0, \mathbf{X}_{NF}(0) = \mathbf{0}_{NF},$ $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N, \mathbf{h}_N(0) = \mathbf{0}_N,$ $\mathbf{\Lambda}_N = \text{diag}(1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1})$
	For $k = 1, 2, \dots, K$
Шаг 1 :	$\mathbf{G}_{NF}(k) = \frac{\mathbf{R}_N^{-1}(k-1) \mathbf{X}_{NF}(k)}{\lambda \mathbf{S}_F + \mathbf{X}_{NF}^H(k) \mathbf{R}_N^{-1}(k-1) \mathbf{X}_{NF}(k)}$
Шаг 2 :	$\mathbf{R}_N^{-1}(k) = \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{G}_{NF}(k) \mathbf{X}_{NF}^H(k) \mathbf{R}_N^{-1}(k-1)]$
Шаг 3 :	$\alpha_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1) \mathbf{X}_{NF}(k)$
Шаг 4 :	$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{G}_{NF}(k) \alpha_F^H(k)$
	End for k

Данный алгоритм отличается от PW RLS-алгоритма тем, что в параллельном алгоритме $\mathbf{X}_{NF}(k), \mathbf{G}_{NF}(k)$ — это матрицы, а не векторы; $\mathbf{d}_F(k)$ и $\alpha_F(k)$ являются векторами, а не скалярными величинами. Кроме того, знаменатель в уравнении (шаг 2) является не скалярной величиной, а квадратной матрицей с числом элементов $F \times F$. Эта матрица обеспечивает математическую эквивалентность данного параллельного алгоритма и соответствующего последовательного регуляризованного SW RLS-алгоритма.

Эквивалентность означает одинаковое функционирование на всех итерациях алгоритмов при одинаковых входных сигналах и одинаковых параметрах адаптивного фильтра. Здесь $\rho_N(k)$ — вектор сигналов динамической регуляризации; ξ^2 — параметр динамической регуляризации и параметр δ^2 — начальной регуляризации корреляционной матрицы [8], а L — длина скользящего окна, т.е. число выборок обрабатываемых сигналов, участвующих в оценке матрицы $\mathbf{R}_N(k)$ и вектора $\mathbf{r}_N(k)$.

В представленном алгоритме вектор $\mathbf{d}_F(k)$ определяется как $\mathbf{d}_F(k) = [\mu^{0,5} d(k-L), d(k), 0, 0]$, а сигнал ошибки на выходе адаптивного фильтра — как $\alpha_{N,\chi}(k) = d(k) - \mathbf{h}_N^H(k-1) \chi_N(k) = \alpha_F^{(2)}(k)$, где $\alpha_F^{(2)}(k)$ — второй элемент вектора $\alpha_F(k)$. Рассматриваемые в работе векторы, за исключением специально упоминаемых, являются вектор-столбцами. Векторы $\mathbf{d}_F(k)$ и $\alpha_F(k)$ — это векторы-строки. Матрица $\mathbf{G}_{NF}(k)$ представляет собой матрицу коэффициентов Калмана $\mathbf{G}_{NF}(k) = \mathbf{R}_N^{-1}(k) \mathbf{X}_{NF}(k)$.

Вычислительная сложность такого алгоритма пропорциональна $O(N^2F)$ арифметических операций, необходимых для выполнения одной итерации алгоритма.

В силу небольшого числа элементов матрицы в знаменателе уравнения (шаг 2) и аналогичных матриц в других алгоритмах, рассматриваемых далее, вычислительная сложность обращения этих матриц не сказывается на общей вычислительной сложности RLS-алгоритмов при $N \gg 4$. Например, при $F = 4$ использование стандартной процедуры обращения матриц с учетом структуры матрицы позволяет выполнять это обращение с вычислительной сложностью: 88 умножений, 47 сложений и одно деление за одну итерацию алгоритма адаптивной фильтрации. Использование модифицированного приема [20] позволяет выполнять обращение со следующей вычислительной сложностью: 80 умножений, 64 сложения и 4 деления, а использование леммы об обращении клеточных матриц [19, 21], позволяет выполнить такое обращение, используя 54 умножения, 32 сложения и 4 деления за одну итерацию алгоритма адаптивной фильтрации.

Параллельные быстрые RLS-алгоритмы. Рассмотренный способ может быть использован и для получения параллельных быстрых многоканальных RLS-алгоритмов адаптивной фильтрации с вычислительной сложностью $O(NF)$ арифметических операций. Вычислительные процедуры таких алгоритмов получаются путем использования теории линейного предсказания [22] и представления потоков обрабатываемых сигналов в виде матрицы. Полученный таким образом так называемый быстрый алгоритм Калмана (Fast Kalman, FK) приведен ниже.

Параллельный FK-алгоритм

Шаг 0 :	Инициализация: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0 - L + 1) = \mathbf{0}_N,$ $\boldsymbol{\rho}_N(0 - L + 1) = \mathbf{0}_N, d(0) = 0, \dots, d(0 - L + 1) = 0,$ $\mathbf{X}_{NF}(0) = \mathbf{0}_{NF}, \mathbf{h}_N(0) = \mathbf{0}_N, E_N^{f(m)}(0) = \delta^2,$ $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N, \mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N, m = 1 : M,$ $\mathbf{G}_{NF}^{(M)}(1) = \mathbf{0}_{NF}$
	For $k = 1, 2, \dots, K$ For $m = M, M - 1, \dots, 1$
Шаг 1 :	$\boldsymbol{\alpha}_F^{f(m)}(k) = \mathbf{x}_F^{(m)}(k) - \mathbf{h}_N^{f(m)H}(k-1)\mathbf{X}_{NF}^{(m)}(k)$
Шаг 2 :	$\boldsymbol{\alpha}_F^{b(m)}(k) = \mathbf{x}_F^{(m)}(k - N_m) - \mathbf{h}_N^{b(m)H}(k-1)\mathbf{X}_{NF}^{(m-1)}(k)$
Шаг 3 :	$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \mathbf{G}_{NF}^{(m)}(k)\boldsymbol{\alpha}_F^{f(m)H}(k)$
Шаг 4 :	$\mathbf{e}_F^{f(m)}(k) = [\mathbf{x}_F^{(m)}(k) - \mathbf{h}_N^{f(m)H}(k)\mathbf{X}_{NF}^{(m)}(k)]\mathbf{S}_F$
Шаг 5 :	$E_N^{f(m)}(k) = \lambda E_N^{f(m)}(k-1) + \mathbf{e}_F^{f(m)}(k)\boldsymbol{\alpha}_F^{f(m)H}(k)$
Шаг 6 :	$\mathbf{G}_{(N+1)}^{(m)}(k) = \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k) \end{bmatrix} \frac{\mathbf{e}_F^{f(m)}(k)}{E_N^{f(m)}(k)} + \begin{bmatrix} \mathbf{0}_F^T \\ \mathbf{G}_{NF}^{(m)}(k) \end{bmatrix}$

$$\begin{array}{l}
\text{Шаг 7 : } \mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \{ \mathbf{G}_{(N+1)F}^{(m)} \} = \begin{bmatrix} \tilde{\mathbf{Q}}_{NF}^{(m)}(k) \\ \tilde{\mathbf{q}}_F^{(m)}(k) \end{bmatrix} \\
\text{Шаг 8 : } \mathbf{G}_{NF}^{(m-1)}(k) = \left[\tilde{\mathbf{Q}}_{NF}^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) \tilde{\mathbf{q}}_F^{(m)}(k) \right] \times \\
\times \left[\mathbf{I}_F - \boldsymbol{\alpha}_F^{b(m)H}(k) \tilde{\mathbf{q}}_f^{(m)}(k) \right]^{-1} \\
\text{Шаг 9 : } \mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \mathbf{G}_{NF}^{(m-1)}(k) \boldsymbol{\alpha}_F^{b(m)H}(k) \\
\text{End for } m \\
\text{Шаг 10 : } \boldsymbol{\alpha}_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1) \mathbf{X}_{NF}(k) \\
\text{Шаг 11 : } \mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{G}_{NF}^{(0)}(k) \boldsymbol{\alpha}_F^H(k) \\
\text{Шаг 12 : } \mathbf{G}_{NF}^{(M)}(k+1) = \mathbf{G}_{NF}^{(0)}(k) \\
\text{End for } k
\end{array}$$

В данном алгоритме \mathbf{I}_F — единичная матрица, а векторы $\mathbf{x}_F^{(m)}(k) = [\mu^{0,5} x_m(k-L), x_m(k), \mu^{0,5} \xi \rho_m(k-L), \xi \rho_m(k)]$, $\mathbf{x}_F^{(m)}(k-N_m) = [\mu^{0,5} x_m(k-N_m-L), x_m(k-N_m), \mu^{0,5} \xi \rho_m(k-N_m-L), \xi \rho_m(k-N_m)]$, $\boldsymbol{\alpha}_F^{f(m)}(k)$, $\mathbf{e}_F^{f(m)}(k)$, $\mathbf{q}_F^{(m)}(k)$ и $\mathbf{e}_F(k)$ являются вектор-строками. Матрицы $\mathbf{X}_{NF}^{(m)}(k) = [\mu^{0,5} \boldsymbol{\chi}_N^{(m)}(k-L), \boldsymbol{\chi}_N^{(m)}(k), \mu^{0,5} \xi \boldsymbol{\rho}_N^{(m)}(k-L), \xi \boldsymbol{\rho}_N^{(m)}(k)]$ формируются из векторов:

$$\begin{array}{l}
\boldsymbol{\chi}_N^{(0)}(k) = \boldsymbol{\chi}_N(k), \\
\boldsymbol{\chi}_N^{(1)}(k) = \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k), \dots, \right. \\
\left. \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T, \\
\vdots \\
\boldsymbol{\chi}_N^{(m)}(k) = \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \right. \\
\left. \mathbf{x}_{N_m}^T(k-1), \mathbf{x}_{N_{m+1}}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T, \\
\vdots \\
\boldsymbol{\chi}_N^{(M)}(k) = \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \right. \\
\left. \mathbf{x}_{N_m}^T(k-1), \dots, \mathbf{x}_{N_M}^T(k-1) \right]^T; \\
\boldsymbol{\chi}_N^{(0)}(k-L) = \boldsymbol{\chi}_N(k-L), \\
\boldsymbol{\chi}_N^{(1)}(k-L) = \left[\mathbf{x}_{N_1}^T(k-L-1), \mathbf{x}_{N_2}^T(k-L), \dots, \right. \\
\left. \mathbf{x}_{N_m}^T(k-L), \dots, \mathbf{x}_{N_M}^T(k-L) \right]^T,
\end{array}$$

$$\begin{aligned}
& \vdots \\
& \boldsymbol{\chi}_N^{(m)}(k-L) = \left[\mathbf{x}_{N_1}^T(k-L-1), \mathbf{x}_{N_2}^T(k-L-1), \dots, \right. \\
& \left. \mathbf{x}_{N_m}^T(k-L-1), \mathbf{x}_{N_{m+1}}^T(k-L), \dots, \mathbf{x}_{N_M}^T(k-L) \right]^T, \\
& \vdots \\
& \boldsymbol{\chi}_N^{(M)}(k-L) = \left[\mathbf{x}_{N_1}^T(k-L-1), \mathbf{x}_{N_2}^T(k-L-1), \dots, \right. \\
& \left. \mathbf{x}_{N_m}^T(k-L-1), \dots, \mathbf{x}_{N_M}^T(k-L-1) \right]^T; \\
& \\
& \boldsymbol{\rho}_N^{(0)}(k) = \boldsymbol{\rho}_N(k), \\
& \boldsymbol{\rho}_N^{(1)}(k) = [\mathbf{p}_{N_1}^T(k-1), \mathbf{p}_{N_2}^T(k), \dots, \mathbf{p}_{N_m}^T(k), \dots, \\
& \mathbf{p}_{N_M}^T(k)]^T, \\
& \vdots \\
& \boldsymbol{\rho}_N^{(m)}(k) = [\mathbf{p}_{N_1}^T(k-1), \mathbf{p}_{N_2}^T(k-1), \dots, \mathbf{p}_{N_m}^T(k-1), \\
& \mathbf{p}_{N_{m+1}}^T(k), \dots, \mathbf{p}_{N_M}^T(k)]^T, \\
& \vdots \\
& \boldsymbol{\rho}_N^{(m)}(k) = [\mathbf{p}_{N_1}^T(k-1), \mathbf{p}_{N_2}^T(k-1), \dots, \\
& \mathbf{p}_{N_m}^T(k-1), \dots, \mathbf{p}_{N_M}^T(k-1)]^T; \\
& \\
& \boldsymbol{\rho}_N^{(0)}(k-L) = \boldsymbol{\rho}_N(k-L), \\
& \boldsymbol{\rho}_N^{(1)}(k-L) = [\mathbf{p}_{N_1}^T(k-L-1), \mathbf{p}_{N_2}^T(k-L), \dots, \\
& \mathbf{p}_{N_m}^T(k-L), \dots, \mathbf{p}_{N_M}^T(k-L)]^T, \\
& \vdots \\
& \boldsymbol{\rho}_N^{(m)}(k-L) = [\mathbf{p}_{N_1}^T(k-L-1), \mathbf{p}_{N_2}^T(k-L-1), \dots, \\
& \mathbf{p}_{N_m}^T(k-L-1), \mathbf{p}_{N_{m+1}}^T(k-L), \dots, \mathbf{p}_{N_M}^T(k-L)]^T, \\
& \vdots \\
& \boldsymbol{\rho}_N^{(m)}(k-L) = [\mathbf{p}_{N_1}^T(k-L-1), \mathbf{p}_{N_2}^T(k-L-1), \dots, \\
& \mathbf{p}_{N_m}^T(k-L-1), \dots, \mathbf{p}_{N_M}^T(k-L-1)]^T.
\end{aligned}$$

В быстром RLS-алгоритме матрицы $\mathbf{S}_{N+1}^{(m)}$ и $\mathbf{T}_{N+1}^{(m)}$ являются перестановочными. Умножение векторов на эти матрицы не требует арифметических операций. Некоторые детали аналогичного способа

построения быстрых многоканальных RLS-алгоритмов адаптивной фильтрации с неодинаковым числом весовых коэффициентов в каналах адаптивного фильтра с использованием перестановочных матриц рассмотрены в работе [23]. Перестановочные матрицы для случая $M = 3$ приведены в работе [8].

Вычислительные процедуры параллельных версий других быстрых RLS-алгоритмов приведены ниже: Fast Transversal Filter (FTF), Fast a Posteriori Error Sequential Technique (FAEST) и стабилизированный FAEST-алгоритма [5, 22, 24].

Параллельный FTF-алгоритм

Шаг 0 :	Инициализация: $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\chi}_N(0 - L + 1) = \mathbf{0}_N,$ $\boldsymbol{\rho}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\rho}_N(0 - L + 1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0 - L + 1) = 0, \mathbf{X}_{NF}(0) = \mathbf{0}_{NF},$ $\mathbf{h}_N(0) = \mathbf{0}_N, E_N^{f(m)}(0) = \delta^2, E_N^{b(m)}(0) = \delta^2 \lambda^{-N_m},$ $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N, \mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N, m = 1 : M,$ $\tilde{\mathbf{T}}_{NF}^{(m)}(1) = \mathbf{0}_{NF}, \Phi_F^{(M)}(1) = \mathbf{S}_F$
	For $k = 1, 2, \dots, K$ For $m = M, M - 1, \dots, 1$
Шаг 1 :	$\boldsymbol{\alpha}_F^{f(m)}(k) = \mathbf{x}_F^{(m)}(k) - \mathbf{h}_N^{f(m)H}(k - 1)\mathbf{X}_{NF}^{(m)}(k)$
Шаг 2 :	$\mathbf{e}_F^{f(m)}(k) = \boldsymbol{\alpha}_F^{f(m)}(k)\Phi_F^{(m)}(k)$
Шаг 3 :	$\boldsymbol{\eta}_F^{(m)}(k) = \frac{\boldsymbol{\alpha}_F^{f(m)}(k)}{E_N^{f(m)}(k - 1)}$
Шаг 4 :	$\tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) = \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k - 1) \end{bmatrix} \boldsymbol{\eta}_F^{(m)}(k) + \begin{bmatrix} \mathbf{0}_F^T \\ \tilde{\mathbf{T}}_{NF}^{(m)}(k) \end{bmatrix}$
Шаг 5 :	$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \{ \tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) \} = \begin{bmatrix} \tilde{\mathbf{Q}}_{NF}^{(m)}(k) \\ \tilde{\mathbf{q}}_F^{(m)}(k) \end{bmatrix}$
Шаг 6 :	$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k - 1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m)}(k) \mathbf{e}_F^{f(m)H}(k)$
Шаг 7 :	$\boldsymbol{\alpha}_F^{b(m)}(k) = \tilde{\mathbf{q}}_F^{(m)}(k) E_N^{b(m)}(k - 1)$
Шаг 8 :	$\tilde{\mathbf{T}}_{NF}^{(m-1)}(k) = \tilde{\mathbf{Q}}_{NF}^{(m)}(k) + \mathbf{h}_N^{b(m)}(k - 1) \tilde{\mathbf{q}}_F^{(m)}(k)$
Шаг 9 :	$E_N^{f(m)}(k) = \lambda E_N^{f(m)}(k - 1) + \mathbf{e}_F^{f(m)}(k) \boldsymbol{\alpha}_F^{f(m)H}(k)$
Шаг 10 :	$\tilde{\Phi}_F^{(m)}(k) = \Phi_F^{(m)}(k) \left[\mathbf{I}_F - \frac{\boldsymbol{\alpha}_F^{f(m)H}(k) \mathbf{e}_F^{f(m)}(k)}{E_N^{f(m)}(k)} \right]$
Шаг 11 :	$\Phi_F^{(m-1)}(k) = [\mathbf{I}_F - \lambda^{-1} \tilde{\Phi}_F^{(m)}(k) \tilde{\mathbf{q}}_F^{(m)H}(k) \boldsymbol{\alpha}_F^{b(m)}(k)]^{-1} \tilde{\Phi}_F^{(m)}(k)$
Шаг 12 :	$\mathbf{e}_F^{b(m)}(k) = \boldsymbol{\alpha}_F^{b(m)}(k) \Phi_F^{(m-1)}(k)$
Шаг 13 :	$E_N^{b(m)}(k) = \lambda E_N^{b(m)}(k - 1) + \mathbf{e}_F^{b(m)}(k) \boldsymbol{\alpha}_F^{b(m)H}(k)$
Шаг 14 :	$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k - 1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m-1)}(k) \mathbf{e}_F^{b(m)H}(k)$ End for k

Параллельный FTF-алгоритм

Шаг 15 :	$\alpha_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1)\mathbf{X}_{NF}(k)$
Шаг 16 :	$\mathbf{e}_F(k) = \alpha_F(k)\Phi_F^{(0)}(k)$
Шаг 17 :	$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1}\tilde{\mathbf{T}}_{NF}^{(0)}(k)\mathbf{e}_F^H(k)$
Шаг 18 :	$\tilde{\mathbf{T}}_{NF}^{(M)}(k+1) = \tilde{\mathbf{T}}_{NF}^{(0)}(k), \Phi_F^{(M)}(k+1) = \Phi_F^{(0)}(k)$
	End for k

Параллельный FAEST-алгоритм отличается от параллельного FTF-алгоритма тем, что в первом рекурсивно вычисляются матрицы $[\Phi_F^{(m)}(k)]^{-1}$, а во втором – матрицы $\Phi_F^{(m)}(k)$. Аналогичные матрицы в последовательных быстрых RLS-алгоритмах отсутствуют. Они становятся скалярными величинами, известными в теории адаптивной фильтрации как отношения правдоподобия.

Параллельный FAEST-алгоритм

Шаг 0 :	Инициализация: $\chi_N(0) = \mathbf{0}_N, \dots, \chi_N(0-L+1) = \mathbf{0}_N,$ $\rho_N(0) = \mathbf{0}_N, \dots, \rho_N(0-L+1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0-L+1) = 0, \mathbf{X}_{NF}(0) = \mathbf{0}_{NF},$ $\mathbf{h}_N(0) = \mathbf{0}_N, E_N^{f(m)}(0) = \delta^2, E_N^{b(m)}(0) = \delta^2\lambda^{-N_m},$ $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N, \mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N, m = 1 : M,$ $\tilde{\mathbf{T}}_{NF}^{(m)}(1) = \mathbf{0}_{NF}, \Phi_F^{(M)}(1) = \mathbf{S}_F$
	For $k = 1, 2, \dots, K$
	For $m = M, M-1, \dots, 1$
Шаг 1 :	$\alpha_F^{f(m)}(k) = \mathbf{x}_F^{(m)}(k) - \mathbf{h}_N^{f(m)H}(k-1)\mathbf{X}_{NF}^{(m)}(k)$
Шаг 2 :	$\mathbf{e}_F^{f(m)}(k) = \alpha_F^{f(m)}(k)\Phi_F^{(m)}(k)$
Шаг 3 :	$\boldsymbol{\eta}_F^m(k) = \frac{\alpha_F^{f(m)}(k)}{E_N^{f(m)}(k-1)}$
Шаг 4 :	$\tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) = \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k-1) \end{bmatrix} \boldsymbol{\eta}_F^m(k) + \begin{bmatrix} \mathbf{0}_F^T \\ \tilde{\mathbf{T}}_{NF}^{(m)}(k) \end{bmatrix}$
Шаг 5 :	$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \{ \tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) \} = \begin{bmatrix} \tilde{\mathbf{Q}}_{NF}^{(m)}(k) \\ \tilde{\mathbf{q}}_F^{(m)}(k) \end{bmatrix}$
Шаг 6 :	$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1}\tilde{\mathbf{T}}_{NF}^{(m)}(k)\mathbf{e}_F^{f(m)H}(k)$
Шаг 7 :	$\alpha_F^{b(m)}(k) = \tilde{\mathbf{q}}_F^{(m)}(k)E_N^{b(m)}(k-1)$
Шаг 8 :	$\tilde{\mathbf{T}}_{NF}^{(m-1)}(k) = \tilde{\mathbf{Q}}_{NF}^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1)\tilde{\mathbf{q}}_F^{(m)}(k)$
Шаг 9 :	$[\bar{\Phi}_F^{(m)}(k)]^{-1} = [\Phi_F^{(m)}(k)]^{-1} + \lambda^{-1}\alpha_F^{f(m)H}(k)\boldsymbol{\eta}_F^m(k)$
Шаг 10 :	$E_N^{f(m)}(k) = \lambda E_N^{f(m)}(k-1) + \mathbf{e}_F^{f(m)}(k)\alpha_F^{f(m)H}(k)$
Шаг 11 :	$[\Phi_F^{(m-1)}(k)]^{-1} = [\bar{\Phi}_F^{(m)}(k)]^{-1} - \lambda^{-1}\alpha_F^{b(m)H}(k)\mathbf{q}_F^{(m)}(k)$
Шаг 12 :	$\mathbf{e}_F^{b(m)}(k) = \alpha_F^{b(m)}(k)\Phi_F^{(m-1)}(k)$

Шаг 13 :	$E_N^{b(m)}(k) = \lambda E_N^{b(m)}(k-1) + \mathbf{e}_F^{b(m)}(k) \boldsymbol{\alpha}_F^{b(m)H}(k)$
Шаг 14 :	$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m-1)}(k) \mathbf{e}_F^{b(m)H}(k)$ End for k
Шаг 15 :	$\boldsymbol{\alpha}_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1) \mathbf{X}_{NF}(k)$
Шаг 16 :	$\mathbf{e}_F(k) = \boldsymbol{\alpha}_F(k) \boldsymbol{\Phi}_F^{(0)}(k)$
Шаг 17 :	$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m-1)}(k) \mathbf{e}_F^H(k)$
Шаг 18 :	$\tilde{\mathbf{T}}_{NF}^{(M)}(k+1) = \tilde{\mathbf{T}}_{NF}^{(0)}(k), \boldsymbol{\Phi}_F^{(M)}(k+1) = \boldsymbol{\Phi}_F^{(0)}(k)$ End for k

Параллельная версия стабилизированного FAEST-алгоритма [24]

Шаг 0 :	Инициализация: $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\chi}_N(0-L+1) = \mathbf{0}_N,$ $\boldsymbol{\rho}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\rho}_N(0-L+1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0-L+1) = 0,$ $\mathbf{X}_{NF}(0) = \mathbf{0}_{NF}, \mathbf{h}_N(0) = \mathbf{0}_N,$ $E_N^{f(m)}(0) = \delta^2, E_N^{b(m)}(0) = \delta^2 \lambda^{-N_m},$ $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N, \mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N, m = 1 : M,$ $\tilde{\mathbf{T}}_{NF}^{(M)}(1) = \mathbf{0}_{NF}, \boldsymbol{\Phi}_F^{(M)}(1) = \mathbf{S}_F$
	For $k = 1, 2, \dots, K$ For $m = M, M-1, \dots, 1$
Шаг 1 :	$\boldsymbol{\alpha}_F^{f(m)}(k) = \mathbf{x}_F^{(m)}(k) - \mathbf{h}_N^{f(m)H}(k-1) \mathbf{X}_{NF}^{(m)}(k)$
Шаг 2 :	$\mathbf{e}_F^{f(m)}(k) = \boldsymbol{\alpha}_F^{f(m)}(k) \boldsymbol{\Phi}_F^{(m)}(k)$
Шаг 3 :	$\boldsymbol{\eta}_F^{(m)}(k) = \frac{\boldsymbol{\alpha}_F^{f(m)}(k)}{E_N^{f(m)}(k-1)}$
Шаг 4 :	$\tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) = \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k-1) \end{bmatrix} \boldsymbol{\eta}_F^{(m)}(k) + \begin{bmatrix} \mathbf{0}_F^T \\ \tilde{\mathbf{T}}_{NF}^{(m)}(k) \end{bmatrix}$
Шаг 5 :	$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \{ \tilde{\mathbf{T}}_{(N+1)F}^{(m)}(k) \} = \begin{bmatrix} \tilde{\mathbf{Q}}_{NF}^{(m)}(k) \\ \tilde{\mathbf{q}}_F^{(m)}(k) \end{bmatrix}$
Шаг 6 :	$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m)}(k) \mathbf{e}_F^{f(m)H}(k)$
Шаг 7 :	$E_N^{f(m)}(k) = \lambda E_N^{f(m)}(k-1) + \mathbf{e}_F^{f(m)}(k) \boldsymbol{\alpha}_F^{f(m)H}(k)$
Шаг 8 :	$[\boldsymbol{\Phi}_F^{(m)}(k)]^{-1} = [\boldsymbol{\Phi}_F^{(m)}(k)]^{-1} + \lambda^{-1} \boldsymbol{\alpha}_F^{f(m)H}(k) \boldsymbol{\eta}_F^{(m)}(k)$
Шаг 9 :	$\boldsymbol{\alpha}_F^{b(m)}(k) = \mathbf{x}_F^{(m)}(k - N_m) - \mathbf{h}_N^{b(m)H}(k-1) \mathbf{X}_{NF}^{(m-1)}(k)$
Шаг 10 :	$\mathbf{q}^{(m)}(k) = \frac{\boldsymbol{\alpha}_F^{b(m)}(k)}{E_N^{b(m)}(k-1)}$
Шаг 11 :	$\tilde{\boldsymbol{\alpha}}_F^{b(m)}(k) = \tilde{\mathbf{q}}_F^{(m)}(k) E_N^{b(m)}(k-1)$

Шаг 12 :	$\alpha_F^{b(1)(m)}(k) = K_1 \alpha_F^{b(m)}(k) + (1 - K_1) \tilde{\alpha}_F^{b(m)}(k)$
Шаг 13 :	$\alpha_F^{b(2)(m)}(k) = K_2 \alpha_F^{b(m)}(k) + (1 - K_2) \tilde{\alpha}_F^{b(m)}(k)$
Шаг 14 :	$\alpha_F^{b(5)(m)}(k) = K_5 \alpha_F^{b(m)}(k) + (1 - K_5) \tilde{\alpha}_F^{b(m)}(k)$
Шаг 15 :	$\mathbf{t}_F^{(m)}(k) = K_4 \mathbf{q}_F^{(m)}(k) + (1 - K_4) \tilde{\mathbf{q}}_F^{(m)}(k)$
Шаг 16 :	$\tilde{\mathbf{T}}_{NF}^{(m-1)}(k) = \tilde{\mathbf{Q}}_{NF}^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) \mathbf{t}_F^{(m)}(k)$
Шаг 17 :	$[\hat{\Phi}_F^{(m-1)}(k)]^{-1} = \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m-1)H}(k) \mathbf{X}_{NF}^{(m-1)}(k) + \mathbf{S}_F$
Шаг 18 :	$[\tilde{\Phi}_F^{(m-1)}(k)]^{-1} = [\bar{\Phi}_F^{(m)}(k)]^{-1} - \lambda^{-1} \alpha_F^{b(5)(m)H}(k) \tilde{\mathbf{q}}_F^{(m)}(k)$
Шаг 19 :	$[\Phi_F^{(m-1)}(k)]^{-1} = K_3 [\hat{\Phi}_F^{(m-1)}(k)]^{-1} + (1 - K_3) [\tilde{\Phi}_F^{(m-1)}(k)]^{-1}$
Шаг 20 :	$\mathbf{e}_F^{b(1)(m)}(k) = \alpha_F^{b(1)(m)}(k) \tilde{\Phi}_F^{(m-1)}(k)$
Шаг 21 :	$\mathbf{e}_F^{b(2)(m)}(k) = \alpha_F^{b(2)(m)}(k) \tilde{\Phi}_F^{(m-1)}(k)$
Шаг 22 :	$E_N^{b(m)}(k) = \lambda E_N^{b(m)}(k-1) + \mathbf{e}_F^{b(2)(m)}(k) \alpha_F^{b(2)(m)H}(k)$
Шаг 23 :	$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(m-1)}(k) \mathbf{e}_F^{b(1)(m)H}(k)$
	End for m
Шаг 24 :	$\alpha_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1) \mathbf{X}_{NF}(k)$
Шаг 25 :	$\mathbf{e}_F(k) = \alpha_F(k) \Phi_F^{(0)}(k)$
Шаг 26 :	$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(0)}(k) \mathbf{e}_F^H(k)$
Шаг 27 :	$\tilde{\mathbf{T}}_{NF}^{(M)}(k+1) = \tilde{\mathbf{T}}_{NF}^{(0)}(k), \Phi_F^{(M)}(k+1) = \Phi_F^{(0)}(k)$
	End for k

Рассмотренные вычислительные процедуры могут быть использованы в качестве алгоритмов безусловной адаптивной фильтрации (без ограничений) или в качестве процедур вычисления матрицы коэффициентов Калмана $\mathbf{G}_{NF}(k) = \mathbf{G}_{NF}^{(0)}(k) = \lambda^{-1} \tilde{\mathbf{T}}_{NF}^{(0)}(k) \Phi_F^{(0)}(k)$ в рассматриваемых далее линейно-ограниченных (LC) RLS-алгоритмах.

Параллельные LC RLS-алгоритмы. В параллельных LC RLS-алгоритмах используется лемма (2) не только для вычисления матрицы $\mathbf{R}_N^{-1}(k)$, но и для вычисления матриц $\Gamma_{NJ}(k) = \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}$, $\Psi_J^{-1}(k) = [\mathbf{C}_{NJ}^H \Gamma_{NJ}(k)]^{-1}$ и $\mathbf{Q}_{NJ}(k) = \Gamma_{NJ}(k) \Psi_J^{-1}(k)$, обусловленных линейными ограничениями. Эти вычисления базируются на приемах, аналогичных рассмотренным в работе [18]. Таким образом, можно последовательно получить три разновидности параллельных LC RLS-алгоритмов, условно обозначаемых как алгоритмы № 1, № 2 и № 3.

Шаг 0 :	Инициализация: $\boldsymbol{\chi}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\chi}_N(0 - L + 1) = \mathbf{0}_N,$ $\boldsymbol{\rho}_N(0) = \mathbf{0}_N, \dots, \boldsymbol{\rho}_N(0 - L + 1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0 - L + 1) = 0,$ $\mathbf{X}_{NF}(0) = \mathbf{0}_{NF}, \mathbf{R}_N^{-1}(0) = \delta^{-2} \boldsymbol{\Lambda}_N,$ $\boldsymbol{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}, \boldsymbol{\Psi}_J^{-1}(0) = [\mathbf{C}_{NJ}^H \boldsymbol{\Gamma}_{NJ}(0)]^{-1},$ $\mathbf{h}_N(0) = \boldsymbol{\Gamma}_{NJ}(0) \boldsymbol{\Psi}_J^{-1}(0) \mathbf{f}_J,$ $\boldsymbol{\Lambda}_N = \text{diag}(1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1})$
	For $k = 1, 2, \dots, K$
Шаг 1 :	Вычисление $\mathbf{G}_{NF}(k)$
Шаг 2 :	$\mathbf{V}_{JF}(k) = \mathbf{C}_{NJ}^H \mathbf{G}_{NF}(k)$
Шаг 3 :	$\mathbf{N}_{JF}^H(k) = \mathbf{X}_{NF}^H(k) \boldsymbol{\Gamma}_{NJ}(k - 1)$
Шаг 4 :	$\boldsymbol{\Omega}_J(k) = \frac{\boldsymbol{\Psi}_J^{-1}(k - 1) \mathbf{V}_{JF}(k)}{\mathbf{I}_F - \mathbf{N}_{JF}^H(k) \boldsymbol{\Psi}_J^{-1}(k - 1) \mathbf{V}_{JF}(k)}$
Шаг 5 :	$\boldsymbol{\Psi}_J^{-1}(k) = \lambda [\boldsymbol{\Psi}_J^{-1}(k - 1) + \boldsymbol{\Omega}_J(k) \mathbf{N}_{JF}^H(k) \boldsymbol{\Psi}_J^{-1}(k - 1)]$
Шаг 6 :	$\boldsymbol{\Gamma}_{NJ}(k) = \lambda^{-1} [\boldsymbol{\Gamma}_{NJ}(k - 1) - \mathbf{G}_{NF}(k) \mathbf{N}_{JF}^H(k)]$
Шаг 7 :	$\boldsymbol{\alpha}_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k - 1) \mathbf{X}_{NF}(k)$
Шаг 8 :	$\mathbf{h}_N(k) = \mathbf{h}_N(k - 1) + [\mathbf{G}_{NF}(k) - \lambda \mathbf{G}_{NJ, \boldsymbol{\chi}}(k) \boldsymbol{\Omega}_J(k)] \boldsymbol{\alpha}_F^H(k)$ End for k

Модификацией алгоритма № 1 является алгоритм № 2. Отличие между этими алгоритмами заключается в выполнении шага 8, который для алгоритма № 2 определяется как $\mathbf{h}_N(k) = \mathbf{P}_N(k) [\mathbf{h}_N(k - 1) + \mathbf{G}_{NF}(k) \boldsymbol{\alpha}_F^H(k)] + \mathbf{a}_N(k)$, где $\mathbf{P}_N(k) = \mathbf{I}_N - \boldsymbol{\Gamma}_{NJ}(k) \boldsymbol{\Psi}_J^{-1}(k) \mathbf{C}_{NJ}^H$ и $\mathbf{a}_N(k) = \boldsymbol{\Gamma}_{NJ}(k) \boldsymbol{\Psi}_J^{-1}(k)$ (см. шаги 8 и 9 алгоритма № 2). Здесь $\mathbf{P}_N(k)$ – динамически изменяемая проекционная матрица, благодаря которой условия ограничений $\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_j$ удовлетворяются на каждой итерации алгоритма № 2 в отличие от алгоритма № 1, где такие ограничения задаются лишь при инициализации, а в процессе функционирования могут перестать удовлетворяться в силу накопления ошибок округления. Из-за наличия матрицы $\mathbf{P}_N(k)$ вычислительная сложность алгоритма № 2 пропорциональна $O(N^2)$, что не зависит от способа вычисления матрицы $\mathbf{G}_{NF}(k)$ (быстрого или не быстрого).

Шаг 0 :	Инициализация: $\chi_N(0) = \mathbf{0}_N, \dots, \chi_N(0 - L + 1) = \mathbf{0}_N,$ $\rho_N(0) = \mathbf{0}_N, \dots, \rho_N(0 - L + 1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0 - L + 1) = 0,$ $\mathbf{X}_{NF}(0) = \mathbf{0}_{NF}, \mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N,$ $\mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}, \mathbf{\Psi}_J^{-1}(0) = [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1},$ $\mathbf{h}_N(0) = \mathbf{\Gamma}_{NJ}(0) \mathbf{\Psi}_J^{-1}(0) \mathbf{f}_J,$ $\mathbf{\Lambda}_N = \text{diag}(1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1})$
	For $k = 1, 2, \dots, K$
Шаг 1 :	Вычисление $\mathbf{G}_{NF}(k)$
Шаг 2 :	$\mathbf{V}_{JF}(k) = \mathbf{C}_{NJ}^H \mathbf{G}_{NF}(k)$
Шаг 3 :	$\mathbf{N}_{JF}^H(k) = \mathbf{X}_{NF}^H(k) \mathbf{\Gamma}_{NJ}(k - 1)$
Шаг 4 :	$\mathbf{\Omega}_J(k) = \frac{\mathbf{\Psi}_J^{-1}(k - 1) \mathbf{V}_{JF}(k)}{\mathbf{I}_F - \mathbf{N}_{JF}^H(k) \mathbf{\Psi}_J^{-1}(k - 1) \mathbf{V}_{JF}(k)}$
Шаг 5 :	$\mathbf{\Psi}_J^{-1}(k) = \lambda [\mathbf{\Psi}_J^{-1}(k - 1) + \mathbf{\Omega}_J(k) \mathbf{N}_{JF}^H(k) \mathbf{\Psi}_J^{-1}(k - 1)]$
Шаг 6 :	$\mathbf{\Gamma}_{NJ}(k) = \lambda^{-1} [\mathbf{\Gamma}_{NJ}(k - 1) - \mathbf{G}_{NF}(k) \mathbf{N}_{JF}^H(k)]$
Шаг 7 :	$\alpha_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k - 1) \mathbf{X}_{NF}(k)$
Шаг 8 :	$\mathbf{P}_N(k) = \mathbf{I}_N - \mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k) \mathbf{C}_{NJ}^H$
Шаг 9 :	$\mathbf{a}_N(k) = \mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k) \mathbf{f}_J$
Шаг 10 :	$\mathbf{h}_N(k) = \mathbf{P}_N(k) [\mathbf{h}_N(k - 1) + \mathbf{G}_{NF}(k) \alpha_F^H(k)] + \mathbf{a}_N(k)$
	End for k

Упрощение LC RLS-алгоритма № 2 достигается в алгоритме № 3, в котором, вместо рекурсивного вычисления матриц $\mathbf{\Gamma}_{NJ}(k)$ и $\mathbf{\Psi}_J^{-1}(k)$, вычисляется матрица $\mathbf{Q}_{NJ}(k)$.

Шаг 0 :	Инициализация: $\chi_N(0) = \mathbf{0}_N, \dots, \chi_N(0 - L + 1) = \mathbf{0}_N,$ $\rho_N(0) = \mathbf{0}_N, \dots, \rho_N(0 - L + 1) = \mathbf{0}_N,$ $d(0) = 0, \dots, d(0 - L + 1) = 0,$ $\mathbf{X}_{NF}(0) = \mathbf{0}_{NF}, \mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N,$ $\mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}, \mathbf{Q}_{NJ}(0) = \mathbf{\Gamma}_{NJ}(0) [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1},$ $\mathbf{h}_N(0) = \mathbf{Q}_{NJ}(0) \mathbf{f}_J,$ $\mathbf{\Lambda}_N = \text{diag}(1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1})$
	For $k = 1, 2, \dots, K$
Шаг 1 :	Вычисление $\mathbf{G}_{NF}(k)$
Шаг 2 :	$\mathbf{V}_{JF}(k) = \mathbf{C}_{NJ}^H \mathbf{G}_{NF}(k)$
Шаг 3 :	$\mathbf{N}_{JF}^H(k) = \mathbf{X}_{NF}^H(k) \mathbf{Q}_{NJ}(k - 1)$

Шаг 4 :	$\mathbf{Q}'_{NJ}(k) = [\mathbf{Q}_{NJ}(k-1) - \mathbf{G}_{NF}(k)\mathbf{N}_{JF}^H(k) \times$ $\times \left[\mathbf{I}_J + \frac{\mathbf{V}_{JF}(k)\mathbf{N}_{JF}^H(k)}{\mathbf{I}_F - \mathbf{N}_{JF}^H(k)\mathbf{V}_{JF}(k)} \right]]$
Шаг 5 :	$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ}(\mathbf{C}_{NJ}^H\mathbf{C}_{NJ})^{-1}[\mathbf{I}_J - \mathbf{C}_{NJ}^H\mathbf{Q}'_{NJ}(k)]$
Шаг 6 :	$\boldsymbol{\alpha}_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1)\mathbf{X}_{NF}(k)$
Шаг 7 :	$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mathbf{G}_{NF}(k)\boldsymbol{\alpha}_F^H(k)$
Шаг 8 :	$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ}(k)[\mathbf{f}_J - \mathbf{C}_{NJ}^H\mathbf{h}'_N(k)]$
	End for k

Рассмотренные LC RLS-алгоритмы также являются параллельными, поскольку в таких алгоритмах вычисление всех матриц с числом столбцов, равным F , может быть выполнено с помощью F параллельных процессоров, так как эти вычисления не зависят друг от друга и определяются независимыми элементами матриц $\mathbf{X}_{NF}(k)$.

Моделирование. Поскольку все рассмотренные параллельные алгоритмы являются математически эквивалентными одноименным последовательным алгоритмам, то физическая зависимость между независимыми параллельными вычислениями на самом деле существует. Она обеспечивается с помощью квадратных матриц с размером $F \times F$, фигурирующих в выражениях параллельного RLS-алгоритма (шаг 1), параллельного FK-алгоритма (шаг 8); параллельных FTF-алгоритма и FAEST-алгоритма (шаг 11); параллельного FAEST-алгоритма и шагов 4 параллельных LC RLS-алгоритмов № 1, № 2 и № 3 (шаг 18).

В силу эквивалентности параллельных и последовательных алгоритмов, результаты моделирования, подтверждающие работоспособность рассмотренных в статье алгоритмов, полностью совпадают с результатами, представленными в работе [5]. Эти результаты также приводятся далее. На рис. 1, *a* и *б* сравниваются многоканальный вариант LC PW RLS-алгоритма и многоканальный нерегуляризованный LC SW RLS-алгоритм при решении задачи идентификации трехканального линейного фильтра и речевых (нестационарных) входных сигналов.

Из рис. 1, *a* видно, что ограничение 0 дБ амплитудно-частотной характеристики (АЧХ) адаптивного фильтра, задаваемое на частотах 1 и 2 кГц, обеспечивается обоими алгоритмами. Вертикальные стрелки указывают на частоты ограничений, а горизонтальная штриховая линия обозначает уровень ограничения АЧХ. Однако в алгоритмах со скользящим окном (благодаря следящим свойствам этих алгоритмов) такой параметр, как Echo Return Loss Enhancement (ERLE), достигает более высокого значения, чем в алгоритмах с бесконечным окном (см.

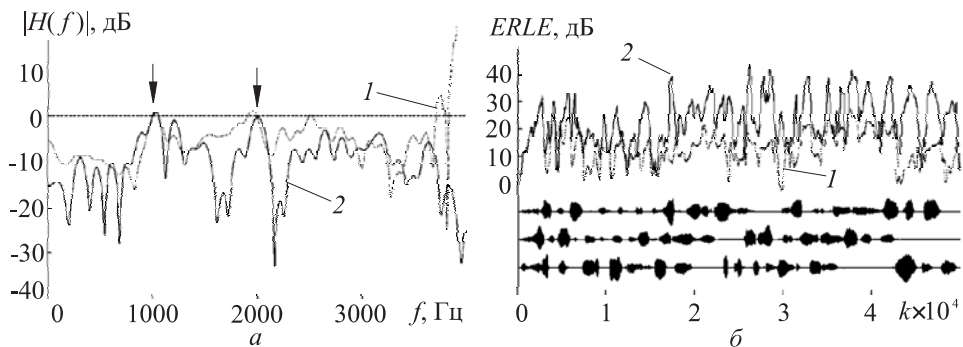


Рис. 1. Результаты моделирования (АЧХ — а и ERLE — б) линейно-ограниченного RLS-алгоритма без регуляризации с бесконечным (1) и со скользящим (2) окном

рис. 1). Параметр ERLE является одним из показателей качества функционирования алгоритмов адаптивной фильтрации в задачах идентификации и представляет собой отношение энергии эхо-сигнала $d(k)$ к энергии подавленного эхо-сигнала $\alpha_{N,\chi}(k)$, измеряемых на каждой итерации алгоритма в пределах некоторого скользящего окна B как

$$ERLE(k) = 10 \log_{10} \left(\frac{\sum_{i=k-B+1}^k d^2(i)}{\sum_{i=k-B+1}^k \alpha_{N,\chi}^2(i)} \right).$$

Из рис. 1, б видно, что в рассмотренных примерах значения параметра ERLE в случае RLS-алгоритма с бесконечным окном примерно на 20 дБ ниже, чем в случае RLS-алгоритма со скользящим окном. Это объясняется тем, что при обработке нестационарных сигналов во втором случае корреляционная матрица сигналов вычисляется более точно. В рассматриваемых примерах длина скользящего окна L в адаптивном алгоритме равнялась выборкам, что при частоте дискретизации сигналов, равной 8 кГц, примерно соответствует 30 мс (интервалу стационарности речевых сигналов). Значения ERLE также рассчитывались на скользящем окне $B = 256$.

Улучшение разработанных алгоритмов достигается путем использования динамической регуляризации при обращении корреляционной матрицы (рис. 2, а и б). Из рис. 2, а видно, что заданные ограничения АЧХ также выполняются и в случае регуляризованного алгоритма, а из рис. 2, б следует, что при использовании регуляризации достигаемое значение ERLE в среднем не меньше, чем при отсутствии регуляризации.

Заключение. Таким образом, в настоящей работе рассмотрен простой способ описания RLS- и LC RLS-алгоритмов в виде однотипных параллельных вычислительных процедур независимо от возможных вариантов модификации корреляционной матрицы адаптивного

фильтра. Данные алгоритмы эквивалентны одноименным последовательным алгоритмам и одноименным параллельным алгоритмам. Различие между параллельными и последовательными алгоритмами заключается в вычислительных процедурах, а также в возможности или невозможности реализации этих процедур с помощью параллельных вычислений. Полное число арифметических операций в одноименных последовательных и параллельных алгоритмах примерно одинаково. Однако при использовании процессоров для реализации параллельных алгоритмов вычислительная нагрузка на один ЦСП уменьшается в F раз. Рассмотренные алгоритмы и их частные случаи могут быть использованы во всех традиционных приложениях адаптивной обработки сигналов. Эти алгоритмы были разработаны в качестве элементов библиотеки [25] БИС серии “МУЛЬТИКОР” [6].

СПИСОК ЛИТЕРАТУРЫ

1. S a y e d A. H. Fundamentals of adaptive filtering. – NJ, Hoboken: John Wiley and Sons, Inc., 2003. – 1125 p.
2. В e n e s t y J., H u a n g Y. (Eds). Adaptive signal processing: applications to real-world problems. – Berlin, Heidelberg, New York, Springer-Verlag, 2003. – 356 p.
3. П е р в а ч е в С. В., П е т р о в А. И. Адаптивная фильтрация сообщений. – М.: Радио и связь, 1991. – 160 с.
4. П и с т о л ь к о р с А. А., Л и т в и н о в О. С. Введение в теорию адаптивных антенн. – М.: Наука, 1991. – 200 с.
5. Д ж и г а н В. И. Многоканальные RLS- и быстрые RLS-алгоритмы адаптивной фильтрации // Успехи современной радиоэлектроники. – 2004. – № 11. – С. 48–77.
6. П е т р и ч к о в и ч Я. Я., С о л о х и н а Т. В. Цифровые сигнальные контроллеры “МУЛЬТИКОР” – новые отечественные серии систем на кристалле // Докл. 6-й Международной конф. “Цифровая обработка сигналов и ее применения (DSPA-2004)” (Москва, 31 марта–2 апреля 2004 г.). – М., 2004. – Т. 1. – С. 8–15.

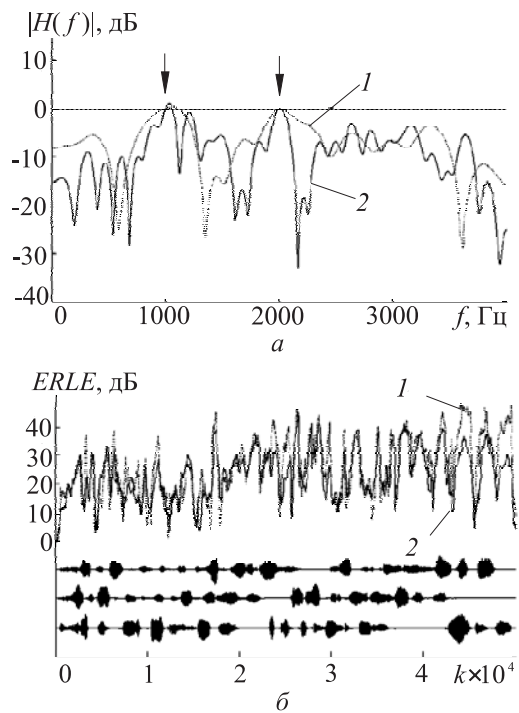


Рис. 2. Результаты моделирования (АЧХ — а и ERLE — б) линейно-ограниченного RLS-алгоритма со скользящим окном с регуляризацией 1 и без регуляризации (2)

7. P a r a o d y s s e u s C. A robust, parallelizable, $O(m)$, a posteriori recursive least squares algorithm for efficient adaptive filtering // IEEE Trans. Signal Processing. – 1999. – Vol. 47. – № 9. – P. 2552–2558.
8. Д ж и г а н В. И. Многоканальный быстрый RLS-алгоритм адаптивной фильтрации для параллельной реализации с помощью четырех процессоров // Вестник МГТУ им. Н.Э. Баумана. Серия “Приборостроение”. – 2005. – № 1. – С. 83–98.
9. Д ж и г а н В. И. Параллельный регуляризованный быстрый RLS-алгоритм многоканальной адаптивной фильтрации со скользящим окном и линейными ограничениями // Тр. 10-й Международной конф. “Радиолокация, навигация, связь (RLNC-2004)” (Воронеж, 13–15 апреля 2004 г.). – Воронеж, 2004. – Т. 1. – С. 132–142.
10. Д ж и г а н В. И. Параллельные регуляризованные RLS-алгоритмы многоканальной адаптивной фильтрации // Цифровая обработка сигналов. – 2004. – № 2. – С. 7–13.
11. Д ж и г а н В. И. RLS-алгоритм адаптивной фильтрации для параллельной реализации с помощью четырех процессоров // Цифровая обработка сигналов. – 2004. – № 3. – С. 2–8.
12. D j i g a n V. I. Multichannel RLS adaptive filtering algorithm for parallel implementation by means of four processors // Proceedings of the 4-th International Scientific and Practical Conference “Internet-Science-Education-2004 (ISE-2004)” (Baku-Vinnytsia-Tyrnovo, September 29–October 12, 2004). – Vinnytsia, 2004. – Vol. 2. – P. 687–691.
13. D j i g a n V. I. Parallelizable multichannel SW fast RLS algorithm for implementation by means of four processors // Proceedings of the Second International Conference on Information Systems and Technology (IST-2004) (November 8–12, 2004, Minsk, Belarus). – Minsk, 2004. – Vol. 2. – P. 100–105.
14. Д ж и г а н В. И. Параллельные вычисления в адаптивной обработке сигналов // Материалы 13-й Международной науч.-техн. конф. “Проблемы передачи и обработки информации в сетях и системах телекоммуникаций”. – Рязань, 2004. – С. 40–42.
15. G a y S. L. Dynamically regularized fast RLS with application to echo cancellation // Proceedings of the International Conference on Acoustic Speech and Signal Processing (ICASSP-96), (May 7–9, 1996, Atlanta, USA). – Atlanta, 1996. – P. 957–960.
16. F r o s t O. L. An algorithm for linearly constrained adaptive array processing // Proceedings of the IEEE. – 1972. – Vol. 60. – August. – P. 926–935.
17. T a k a o K., F u j i t a M., N i s h i T. An adaptive antenna array under directional constraint // IEEE Trans. Antennas and Propagation. – 1976. – Vol. 24. – № 5. – P. 662–669.
18. R e s e n d e L. S., R o m a n o J. M. T., B e l l a n g e r M. G. A fast least-squares algorithm for linearly constrained adaptive filtering // IEEE Trans. Signal Processing. – 1996. – Vol. 44. – № 5. – P. 1168–1174.
19. G i o r d a n o A. A., H s u F. M. Least square estimation with application to digital signal processing. - Canada, Toronto: John Wiley and Sons, Inc., 1985. – 412 p.
20. M a z u c h T., K o z a n e k J. New recurrent algorithm for a matrix inversion // Journal of Computational and Applied Mathematics. – 2001. – Vol. 136, № 1–2. – P. 219–226.
21. Л а н ш о ц К. Практические методы прикладного анализа. Пер. с англ. М.З. Кайнера / Под ред. А.М. Лопшица. – М.: Государственное издательство физико-математической литературы, 1961. – 524 с.

22. Zelniker G., Taylor F. J. Advanced digital signal processing: theory and applications. – New York: Marcel Dekker, Inc., 1994. – 666 p.
23. Glentis G. A., Kalouptsidis N. Fast adaptive algorithms for multichannel filtering and system identification // IEEE Trans. Signal Processing. – 1992. – Vol. 40. – № 10. – P. 2433–2458.
24. Сlock D. T. M., Kailath T. Numerically stable fast transversal filters for recursive least squares adaptive filtering // IEEE Trans. Signal Processing. – 1991. – Vol. 39. – № 1. – P. 92–114.
25. Джиган В. И. Двести пятьдесят алгоритмов адаптивной фильтрации // Материалы 13-й Международной науч.-техн. конф. “Проблемы передачи и обработки информации в сетях и системах телекоммуникаций”. – Рязань, 2004. – С. 42–44.

Статья поступила в редакцию 8.02.2005

Виктор Иванович Джиган родился в 1958 г., окончил в 1984 г. Московский институт электронной техники (МИЭТ). Канд. техн. наук, член РНТОРЭС им. А.С. Попова и института IEEE (США), ведущий научный сотрудник ГУП НПЦ “ЭЛВИС” (Зеленоград, г. Москва). Автор около 120 научных работ в области адаптивной обработки сигналов, цифровой обработки речи и цифровой передачи данных.



V.I. Djigan (b. 1958) graduated from the Moscow Institute of Electronic Engineering in 1984. Ph. D. (Eng.), Russian Popov’s Radio Society member, IEEE senior member (M’96, SM’2004), principal researcher of “ELVEES” R&D Center (Zelenograd, Moscow region). Author of about 120 publications in the field of adaptive signal processing, digital speech processing and digital communication.

**В издательстве МГТУ им. Н.Э. Баумана
в 2006 г. вышла в свет книга**

Суржиков С.Т.

Физическая механика газовых разрядов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 640 с.: 384 ил. (Компьютерные модели физической механики).

Рассмотрены методы компьютерного моделирования электроразрядных процессов и динамики частично ионизованных газов, которые используются в задачах физической механики, физики газовых разрядов и аэрофизики. Основное внимание уделено решению двумерных задач физической механики тлеющих разрядов в аэрокосмических приложениях.

Для научных сотрудников и инженеров, работающих в области физической газовой динамики, физики низкотемпературной плазмы и газовых разрядов, а также для студентов и аспирантов физико-технических специальностей университетов.

По вопросам приобретения обращаться по тел. 433-82-98;
e-mail: surg@ipmnet.ru