ПАРАЛЛЕЛЬНЫЕ МНОГОКАНАЛЬНЫЕ RLS АЛГОРИТМЫ АДАПТИВНОЙ ФИЛЬТРАЦИИ

Джиган В.И.

ГУП НПЦ “ЭЛВИС”, а/я 19, Центральный проспект, Зеленоград, Москва К-460, Россия 124460
Тел.: +7-095-531-1961. Эл. почта: djigan@elvees.ru . Интернет: http://www.elvees.ru

Аннотация: В статье рассматриваются новые SW RLS алгоритмы адаптивной фильтрации, допускающие реализацию с помощью четырех параллельных процессоров. Алгоритмы получены для общего случая многоканальных адаптивных фильтров с неодинаковым числом комплексных весовых коэффициентов в каналах. Семейство алгоритмов включает SW регуляризированные RLS, быстрые RLS и LC RLS алгоритмы, которые могут быть использованы для адаптивной фильтрации нестационарных сигналов.

В работе [1] была представлена обширная библиотека алгоритмов адаптивной фильтрации. Она содержит более 200 алгоритмов, включая рекурсивные алгоритмы по критерию наименьших квадратов со скользящим окном (Sliding Window Recursive Least Squares, SW RLS), SW быстрые (вычислительно эффективные) RLS and SW линейно-ограниченные (Linearly Constrained, LC) RLS алгоритмы [2]. SW RLS алгоритмы предназначены для адаптивной обработки нестационарных сигналов. Получение этих алгоритмов основано на использовании лемм об обращении матриц [3]. В случае скользящего окна, для обращения корреляционной матрицы адаптивного фильтра требуется использовать такие леммы дважды. Это обуславливает примерно удвоенную арифметическую сложность (число арифметических операций за одну итерацию) SW RLS алгоритмов по сравнению с алгоритмами с бесконечным окном (Prewindowed, PW). В силу последовательного характера вычислений в таких алгоритмах, невозможно использовать параллельные процедуры для оценки корреляционной матрицы адаптивного фильтра, определяемой на скользящем окне длиной в L отсчетов.

Учитывая ограниченное число отсчетов, используемых при оценке корреляционной матрицы, и нестационарную природу обрабатываемых сигналов, эта матрица может становиться необратимой, а адаптивный алгоритм – нестабильным. Для борьбы с подобными ситуациями в работах [4, 5] была использована динамическая регуляризация корреляционной матрицы. Этой прием также обуславливает примерно удвоенную арифметическую сложность регуляризированных PW алгоритмов по сравнению с не регуляризированными алгоритмами.

Сложность алгоритмов можно уменьшить в случае использования параллельных вычислений. Для разработки PW регуляризированных RLS, SW RLS или SW LC RLS алгоритмов, допускающих параллельные вычисления, могут быть использованы методы, представленные в работе [6]. В [6] рассмотрены одноканальные SW RLS алгоритмы с действительными весовыми коэффициентами, допускающие вычисления с помощью двух параллельных процессоров. Подобные многоканальные SW LC RLS алгоритмы с комплексными весовыми коэффициентами были рассмотрены в работе [7], а PW регуляризированные RLS алгоритмы – в работе [8].

Приемы [6–8] могут быть также использованы для разработки SW регуляризированных, быстрых и LC RLS алгоритмов для параллельной реализации с помощью четырех процессоров. Один из таких алгоритмов был рассмотрен в [9], а его быстрая версия – в [10].

Настоящая статья представляет новый SW регуляризированный LC RLS алгоритм адаптивной фильтрации, допускающий параллельную реализацию с помощью четырех процессоров. Алгоритм разработан для многоканальных адаптивных фильтров с неодинаковым числом комплексных весовых коэффициентов в каналах. Полная вычислительная процедура алгоритма представлена в табл. 1. Здесь k – номер итерации (отсчета), $\mathbf{ч}_N(k)$ – вектор входных сигналов, $\mathbf{c}_N(k)$ – вектор сигналов регуляризации, $d(k)$ – требуемый сигнал, $\mathbf{h}_N(k)$ – вектор весовых коэффициентов, $N = \sum_{m=1}^{M} N_m$ – общее число весовых коэффициентов, N_m – число весовых коэффициентов в m-м канале, а M – число каналов фильтра. Для получения алгоритма, нужно применить лемму об обращении матриц к регуляризированной корреляционной матрице, определенной на скользящем окне [9], и к матрицам, обусловленным наличием линейных ограничений [4, 7, 11].

Таблица 1. Параллельный SW регуляризированный LC RLS алгоритм

Вычисления	Ссылки

$$\textbf{Initialization : } \mathbf{ч}_N(0)=\mathbf{0}_N,\ldots,\mathbf{ч}_N(0-L+1)=\mathbf{0}_N,\mathbf{c}_N(0)=\mathbf{0}_N,\ldots, \tag{1.0}$$
$$\mathbf{c}_N(0-L+1)=\mathbf{0}_N,d(0)=0,\ldots,d(0-L+1)=0,$$
$$\text{Л}\mathbf{I}_N=diag(1,\lambda,\ldots,\lambda^{N_1-1},\ldots,1,\lambda,\ldots,\lambda^{N_m-1},\ldots,1,\lambda,\ldots,\lambda^{N_M-1}),$$
$$\mathbf{R}_N^{-1}(0)=\delta^{-2}\text{Л}\mathbf{I}_N,\mathbf{\Gamma}_{NJ}(0)=\mathbf{R}_N^{-1}(0)\mathbf{C}_{NJ},\mathbf{Q}_{NJ,\mathbf{ч}_D}(0)=\mathbf{\Gamma}_{NJ}(0)\left[\mathbf{C}_{NJ}^H\mathbf{\Gamma}_{NJ}(0)\right]^{-1},$$
$$\mathbf{h}_N(0)=\mathbf{Q}_{NJ}(0)\mathbf{f}_J$$

For $k=1,2,\ldots,K$

Шаги (1.1) – (1.23) параллельного SW регуляризированного RLS алгоритма [10] (1.1)

$$\mathbf{н}_{J,\mathbf{c}_U}(k)=\mathbf{C}_{NJ}^H\mathbf{g}_{N,\mathbf{c}_U}(k) \tag{1.2}$$
$$\mathbf{н}_{J,\mathbf{c}_D}(k)=\mathbf{C}_{NJ}^H\mathbf{g}_{N,\mathbf{c}_D}(k) \tag{1.3}$$
$$\mathbf{н}_{J,\mathbf{ч}_U}(k)=\mathbf{C}_{NJ}^H\mathbf{g}_{N,\mathbf{ч}_U}(k) \tag{1.4}$$
$$\mathbf{н}_{J,\mathbf{ч}_D}(k)=\mathbf{C}_{NJ}^H\mathbf{g}_{N,\mathbf{ч}_D}(k) \tag{1.5}$$
$$\mathbf{x}_{J,\mathbf{c}_U}^H(k)=\xi\mathbf{c}_N^H(k)\mathbf{Q}_{NJ}(k-1) \tag{1.6}$$
$$\mathbf{x}_{J,\mathbf{c}_D}^H(k)=\xi\mathbf{c}_N^H(k-L)\mathbf{Q}_{NJ}(k-1) \tag{1.7}$$
$$\mathbf{x}_{J,\mathbf{ч}_U}^H(k)=\mathbf{ч}_N^H(k)\mathbf{Q}_{NJ}(k-1) \tag{1.8}$$
$$\mathbf{x}_{J,\mathbf{ч}_D}^H(k)=\mathbf{ч}_N^H(k-L)\mathbf{Q}_{NJ}(k-1) \tag{1.9}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_U}}(k)=\mathbf{x}_{J,\mathbf{c}_U}^H(k)\mathbf{н}_{J,\mathbf{c}_U}(k) \tag{1.10}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_D}}(k)=\mathbf{x}_{J,\mathbf{c}_U}^H(k)\mathbf{н}_{J,\mathbf{c}_D}(k) \tag{1.11}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{ч}_U}}(k)=\mathbf{x}_{J,\mathbf{c}_U}^H(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) \tag{1.12}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{ч}_D}}(k)=\mathbf{x}_{J,\mathbf{c}_U}^H(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.13}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_U}}(k)=\mathbf{x}_{J,\mathbf{c}_D}^H(k)\mathbf{н}_{J,\mathbf{c}_U}(k) \tag{1.14}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_D}}(k)=\mathbf{x}_{J,\mathbf{c}_D}^H(k)\mathbf{н}_{J,\mathbf{c}_D}(k) \tag{1.15}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{ч}_U}}(k)=\mathbf{x}_{J,\mathbf{c}_D}^H(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) \tag{1.16}$$
$$\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{ч}_D}}(k)=\mathbf{x}_{J,\mathbf{c}_D}^H(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.17}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{c}_U}}(k)=\mathbf{x}_{J,\mathbf{ч}_U}^H(k)\mathbf{н}_{J,\mathbf{c}_U}(k) \tag{1.18}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{c}_D}}(k)=\mathbf{x}_{J,\mathbf{ч}_U}^H(k)\mathbf{н}_{J,\mathbf{c}_D}(k) \tag{1.19}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_U}}(k)=\mathbf{x}_{J,\mathbf{ч}_U}^H(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) \tag{1.20}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_D}}(k)=\mathbf{x}_{J,\mathbf{ч}_U}^H(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.21}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{c}_U}}(k)=\mathbf{x}_{J,\mathbf{ч}_D}^H(k)\mathbf{н}_{J,\mathbf{c}_U}(k) \tag{1.22}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{c}_D}}(k)=\mathbf{x}_{J,\mathbf{ч}_D}^H(k)\mathbf{н}_{J,\mathbf{c}_D}(k) \tag{1.23}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_U}}(k)=\mathbf{x}_{J,\mathbf{ч}_D}^H(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) \tag{1.24}$$
$$\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_D}}(k)=\mathbf{x}_{J,\mathbf{ч}_D}^H(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.25}$$
$$\varphi^{-1}_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_U}}(k)=1-\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_U}}(k) \tag{1.26}$$
$$\varphi^{-1}_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_D}}(k)=1+\mu\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_D}}(k) \tag{1.27}$$

$$\varphi^{-1}_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_U}}(k) = 1 - \delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_U}}(k) \tag{1.28}$$

$$\varphi^{-1}_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_D}}(k) = 1 + \mu\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_D}}(k) \tag{1.29}$$

$$\mathbf{w}_{J,\mathbf{c}_U}(k) = \overline{a}_{44}(k)\mathbf{н}_{J,\mathbf{c}_U}(k) - \mu^{0.5}\overline{a}_{34}(k)\mathbf{н}_{J,\mathbf{c}_D}(k) + \overline{a}_{24}(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) - \mu^{0.5}\overline{a}_{14}(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.30}$$

$$\mathbf{w}_{J,\mathbf{c}_D}(k) = -\mu^{-0.5}\overline{a}_{43}(k)\mathbf{н}_{J,\mathbf{c}_U}(k) + \overline{a}_{33}(k)\mathbf{н}_{J,\mathbf{c}_D}(k) - \mu^{-0.5}\overline{a}_{23}(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) + a_{13}(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.31}$$

$$\mathbf{w}_{J,\mathbf{ч}_U}(k) = \overline{a}_{42}(k)\mathbf{н}_{J,\mathbf{c}_U}(k) - \mu^{0.5}\overline{a}_{32}(k)\mathbf{н}_{J,\mathbf{c}_D}(k) + \overline{a}_{22}(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) - \mu^{0.5}\overline{a}_{12}(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.32}$$

$$\mathbf{w}_{J,\mathbf{ч}_D}(k) = -\mu^{-0.5}\overline{a}_{41}(k)\mathbf{н}_{J,\mathbf{c}_U}(k) + \overline{a}_{31}(k)\mathbf{н}_{J,\mathbf{c}_D}(k) - \mu^{-0.5}\overline{a}_{21}(k)\mathbf{н}_{J,\mathbf{ч}_U}(k) + \overline{a}_{11}(k)\mathbf{н}_{J,\mathbf{ч}_D}(k) \tag{1.33}$$

$$\begin{aligned}\mathbf{Q}'_{NJ}(k) = \big[&\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_{N,\mathbf{c}_U}(k)\mathbf{x}^H_{J,\mathbf{c}_U}(k) + \mu\mathbf{g}_{N,\mathbf{c}_D}(k)\mathbf{x}^H_{J,\mathbf{c}_D}(k)\\ &- \mathbf{g}_{N,\mathbf{ч}_U}(k)\mathbf{x}^H_{J,\mathbf{ч}_U}(k) + \mu\mathbf{g}_{N,\mathbf{ч}_D}(k)\mathbf{x}^H_{J,\mathbf{ч}_D}(k)\big]\\ &\times\big[\ \mathbf{I}_J + \mathbf{w}_{J,\mathbf{c}_U}(k)\mathbf{x}^H_{J,\mathbf{c}_U}(k) - \mu\mathbf{w}_{J,\mathbf{c}_D}(k)\mathbf{x}^H_{J,\mathbf{c}_D}(k)\\ &+ \mathbf{w}_{J,\mathbf{ч}_U}(k)\mathbf{x}^H_{J,\mathbf{ч}_U}(k) - \mu\mathbf{w}_{J,\mathbf{ч}_D}(k)\mathbf{x}^H_{J,\mathbf{ч}_D}(k)\big]\end{aligned} \tag{1.34}$$

$$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ}\left(\mathbf{C}^H_{NJ}\mathbf{C}_{NJ}\right)^{-1}\left[\mathbf{I}_J - \mathbf{C}^H_{NJ}\mathbf{Q}'_{NJ}(k)\right] \tag{1.35}$$

$$\alpha_{N,\mathbf{c}_U}(k) = -\mathbf{h}^H_N(k-1)\xi\mathbf{c}_N(k) \tag{1.36}$$

$$\alpha_{N,\mathbf{c}_D}(k) = -\mathbf{h}^H_N(k-1)\xi\mathbf{c}_N(k-L) \tag{1.37}$$

$$\alpha_{N,\mathbf{ч}_U}(k) = d(k) - \mathbf{h}^H_N(k-1)\mathbf{ч}_N(k) \tag{1.38}$$

$$\alpha_{N,\mathbf{ч}_D}(k) = d(k-L) - \mathbf{h}^H_N(k-1)\mathbf{ч}_N(k-L) \tag{1.39}$$

$$\begin{aligned}\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) &+ \mathbf{g}_{N,\mathbf{c}_U}(k)\alpha^*_{N,\mathbf{c}_U}(k) - \mu\mathbf{g}_{N,\mathbf{c}_D}(k)\alpha^*_{N,\mathbf{c}_D}(k)\\ &+ \mathbf{g}_{N,\mathbf{ч}_U}(k)\alpha^*_{N,\mathbf{ч}_U}(k) - \mu\mathbf{g}_{N,\mathbf{ч}_D}(k)\alpha^*_{N,\mathbf{ч}_D}(k)\end{aligned} \tag{1.40}$$

$$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ}(k)\left[\mathbf{f}_J - \mathbf{C}^H_{NJ}\mathbf{h}'_N(k)\right] \tag{1.41}$$

End for k

В алгоритме используется матрица $\overline{\mathbf{A}} = \mathbf{A}^{-1}$. Структура матрицы $\mathbf{A}$ показана ниже:

$$\mathbf{A} = \begin{bmatrix} \varphi^{-1}_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_D}}(k) & -\mu^{0.5}\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{ч}_U}}(k) & \mu\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{c}_D}}(k) & -\mu^{0.5}\delta_{\mathbf{x}_{\mathbf{ч}_D}\mathbf{н}_{\mathbf{c}_U}}(k) \\ \mu^{0.5}\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_D}}(k) & \varphi^{-1}_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{ч}_U}}(k) & \mu^{0.5}\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{c}_D}}(k) & -\delta_{\mathbf{x}_{\mathbf{ч}_U}\mathbf{н}_{\mathbf{c}_U}}(k) \\ \mu\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{ч}_D}}(k) & -\mu^{0.5}\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{ч}_U}}(k) & \varphi^{-1}_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_D}}(k) & -\mu^{0.5}\delta_{\mathbf{x}_{\mathbf{c}_D}\mathbf{н}_{\mathbf{c}_U}}(k) \\ \mu^{0.5}\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{ч}_D}}(k) & \delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{ч}_U}}(k) & \mu^{0.5}\delta_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_D}}(k) & \varphi^{-1}_{\mathbf{x}_{\mathbf{c}_U}\mathbf{н}_{\mathbf{c}_U}}(k) \end{bmatrix}.$$

Элементы этой матрицы определяются в табл. 1. Подобные матрицы возникают и при вычислении векторов коэффициентов Калмана $\mathbf{g}_{N,\chi_U}(k)$, $\mathbf{g}_{N,\chi_D}(k)$, $\mathbf{g}_{N,\mathbf{c}_U}(k)$ и $\mathbf{g}_{N,\mathbf{c}_D}(k)$ [9, 10]. Обращение данных матриц может быть выполнено с помощью любого доступного метода. Например, использование алгоритмов обращения клеточных матриц [3] позволяет осуществить такое обращение с помощью 54 умножений, 32 сложений и 4 делений.

Как следует из табл. 1, все вычисления алгоритма сгруппированы. Каждая группа содержит четыре вычисления, которые определяются векторами $\mathbf{c}_N(k)$, $\mathbf{c}_N(k-L)$ и $\mathbf{ч}_N(k)$, $\mathbf{ч}_N(k-L)$ и не зависят друг от друга. Такие вычисления могут быть выполнены с помощью четырех процессоров, т. е. параллельно.

Сложность нового алгоритма равна $3NJ^2 + 28NJ + 4J^2 + 18J + 2MJ + 6N + 2M$ умножений, $3NJ^2 + 29NJ + 5J^2 + 13J + 2MJ + 7N + 2M$ сложений и M делений на одну итерацию, где J – число линейных ограничений. Полная оценка сложности также включает $7N^2 + 21N + 2M^2 + 8M$ умножений и $6N^2 + 15N + 2M^2 + 8M$ сложений алгоритма [9], или $M(36N + 8M + 1)$ умножений,

$M(35N+8M+1)$ сложений и M делений алгоритма [10] для вычисления векторов коэффициентов Калмана. Если $N>>4$, то арифметическая сложность операции $\mathbf{A}^{-1}$ становиться несущественной. Вычислительная нагрузка при реализации алгоритма уменьшается в четыре раза в случае использования четырех процессоров.

Таким образом, в статье рассмотрено построение параллельных RLS алгоритмов адаптивной фильтрации и представлена вычислительная процедура такого LC алгоритма. Этот алгоритм и SW регуляризированные LC RLS алгоритмы [2] являются математически эквивалентными, что означает одинаковое функционирование в одинаковых условиях. Алгоритмы могут быть использованы в ряде задач адаптивной фильтрации нестационарных сигналов.

Литература

1. Джиган В.И. Библиотека алгоритмов адаптивной фильтрации // Доклады 6-й Международной конференции «Цифровая обработка сигналов и ее применения (DSPA-2004)». – Москва, 31 марта – 2 апреля 2004. – Том 1. – С. 89–94.
2. Джиган В.И. Семейство RLS и быстрых RLS алгоритмов со скользящим окном для многоканальной адаптивной фильтрации с линейными ограничениями // Доклады 6-й Международной конференции «Цифровая обработка сигналов и ее применения (DSPA-2004)». – Москва, 31 марта – 2 апреля 2004. – Том 1. – С. 83–88.
3. Giordano A.A., Hsu F.M. Least square estimation with application to digital signal processing. Canada, Toronto: John Wiley and Sons, Inc., 1985. – 412 p.
4. Gay S.L. Dynamically regularized fast RLS with application to echo cancellation // Proc. ICASSP'96. – May 1996. – P. 957–960.
5. Джиган В.И. Многоканальные RLS алгоритмы с линейными ограничениями // Известия высших учебных заведений. Электроника. – 2004. – №3. – С. 46–53.
6. Papaodysseus 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.
7. Джиган В.И. Параллельный регуляризированный быстрый RLS алгоритм многоканальной адаптивной фильтрации со скользящим окном и линейными ограничениями // Труды 10-й Международной конференции «Радиолокация, навигация, связь (RLNC-2004)». – Воронеж, 13 – 15 апреля 2004. – Том 1. – С. 132–142.
8. Джиган В.И. Параллельные регуляризированные RLS алгоритмы многоканальной адаптивной фильтрации // Цифровая обработка сигналов. – 2004. – №2.
9. Djigan 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 – October12, 2004. – Vol. 2. – P. 687-691.
10. Djigan 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). – Minsk, November 8 – 12, 2005.
11. Resende L.S., Romano J.M.T., Bellanger M.G. A fast least-squares algorithm for linearly constrained adaptive filtering // IEEE Trans. Signal Processing. – 1996. – Vol. 44. – №5. – P.1168–1174.