

УДК 681.513.6+681.5.015.52

АЛГОРИТМЫ АДАПТИВНОЙ ПОЛИНОМИАЛЬНОЙ ФИЛЬТРАЦИИ ВО ВРЕМЕННОЙ ОБЛАСТИ

Щербаков М.А., д.т.н., профессор, заведующий кафедрой «Автоматика и телемеханика» Пензенского государственного университета, e-mail: mashcherbakov@yandex.ru

ADAPTIVE POLYNOMIAL FILTERING ALGORITHMS IN THE TIME DOMAIN

Shcherbakov M.A.

Gradient and Newtonian algorithms for adaptive digital polynomial filters (Volterra filters) in the time domain are considered. Estimates of the limits for changing adaptation parameters are given, guaranteeing the convergence of the algorithms. These estimates are expressed through the eigenvalues of the correlation matrix of the higher moments of the input signal. Methods for speeding up the convergence of adaptive nonlinear filtering algorithms are proposed.

Key words: adaptive nonlinear filtering, polynomial filters, discrete Volterra series.

Ключевые слова: адаптивная нелинейная фильтрация, полиномиальные фильтры, дискретные ряды Вольтерра.

Введение

Последние достижения в области цифровой обработки сигналов связаны с разработкой методов адаптивной нелинейной фильтрации [1-4], которые нашли применения в различных областях, таких как идентификация и управление в нелинейных системах [5], нелинейная адаптивная эхо-компенсация [6, 7], нелинейное оценивание [8], фильтрация сигналов и изображений [9]. Среди множества классов нелинейных фильтров широкое распространение получили цифровые полиномиальные фильтры [9-11]. Данный класс нелинейных фильтров, называемых также фильтрами Вольтерра, представляет собой естественное обобщение линейных фильтров [13]. Ввиду того, что полиномиальные фильтры сохраняют свойство линейности относительно своих коэффициентов, принципы построения алгоритмов адаптации данных фильтров во многом схожи с линейным случаем. В то же время имеется ряд особенностей, связанных с выбором параметров адаптации, касающихся различных нелинейных составляющих фильтра.

В настоящее время существует множество различных алгоритмов адаптивной обработки сигналов. Известны различные модификации метода наименьших квадратов, отличающиеся способом формирования целевой функции и направлением поиска [1-3, 12]. Для всех адаптивных алгоритмов характерно присутствие помех, искажающих истинное направление поиска. Данные помехи обусловлены погрешностями статистических оценок и приводят к тому, что алгоритм адаптации сходится лишь в некоторую окрестность оптимальной точки. В связи с этим при исследовании алгоритмов адаптации можно говорить лишь о сходимости в некотором статистическом смысле (с вероятностью, в среднем).

Поэтому, одним из основных вопросов при разработке адаптивных алгоритмов фильтрации является выбор параметров алгоритма, обеспечивающих сходимость и достаточную скорость процесса адаптации.

Рассмотрены градиентные и ньютоновские алгоритмы адаптации цифровых полиномиальных фильтров (фильтров Вольтерра) во временной области. Приведены оценки границ изменения параметров адаптации, гарантирующие сходимость алгоритмов. Данные оценки выражаются через собственные числа корреляционной матрицы высших моментов входного сигнала. Предложены способы ускорения сходимости алгоритмов адаптивной нелинейной фильтрации.

Векторное представление полиномиальных фильтров

Цифровой полиномиальный фильтр M -го порядка определяется отрезком дискретного ряда Вольтерра [13, 14]

$$y(n) = \sum_{m=0}^M y_m(n) = \sum_{i_1=0}^{N-1} \dots \sum_{i_m=0}^{N-1} h_m(i_1, \dots, i_m) \prod_{j=1}^m x(n-i_j), \quad (1)$$

где $y_m(n)$ – составляющая реакции фильтра, характеризующая нелинейность m -го порядка; $h_m(i_1, \dots, i_m)$ – нелинейная импульсная характеристика (ядро) m -го порядка.

Многомерный полиномиальный фильтр (1) может быть охарактеризован $L_M \times 1$ вектором коэффициентов

$$\mathbf{h}^T = [h_0 \mid h_1^T \mid h_2^T \mid \dots \mid h_M^T], \quad (3)$$

где отдельные блоки \mathbf{h}_m^T соответствуют нелинейным членам различных порядков и содержат упорядоченные значения $h_m(i_1, \dots, i_m)$, соответствующие лишь уникальным комбинациям индексов i_1, \dots, i_m . Можно показать, что размерность вектора \mathbf{h} равна числу сочетаний по m с повторениями из N элементов составит

$$L_M = \sum_{m=0}^M C_{N+m-1}^m = C_{N+M}^M, \quad (4)$$

где C_n^m – число сочетаний из n по m .

Пусть вектор отсчетов входного сигнала $\mathbf{x}_n^T = [x(n) \dots x(n-N+1)]$. В соответствии с (4) сформируем вектор произведений отсчетов входного сигнала

$$\chi_n^T = [1 \mid \mathbf{x}_n^T \mid \mathbf{x}_n^{(2)T} \mid \dots \mid \mathbf{x}_n^{(M)T}], \quad (5)$$

образованной из кронекеровских произведений $\mathbf{x}_n^{(m)}$ отсчетов входного сигнала. Данный вектор составлен из

упорядоченных произведений $x(n-i_1) \cdot \dots \cdot x(n-i_m)$, соответствующих лишь уникальным комбинациям (i_1, \dots, i_m) индексов.

Использование введенных векторных обозначений позволяет представить многомерный полиномиальный фильтр (1) в следующей простой векторной форме:

$$y(n) = \mathbf{h}^T \boldsymbol{\chi}_n, \quad (6)$$

линейной относительно вектора \mathbf{h} , содержащего L_M коэффициентов.

Задача оптимальной полиномиальной фильтрации

Вектор \mathbf{h}_{opt} оптимальных коэффициентов полиномиального фильтра определяется путем минимизации среднеквадратической ошибки между выходным $y(n)$ и заданным (опорным) $d(n)$ сигналами на выходе фильтра

$$\varepsilon = M\{d(n) - y(n)\}^2 \quad (7)$$

и сводится к решению матричного уравнения Винера-Хопфа [15]

$$\mathbf{R}_\chi \mathbf{h} = \mathbf{r}_{d\chi}, \quad (8)$$

где $\mathbf{R}_\chi = M\{\boldsymbol{\chi}_n \boldsymbol{\chi}_n^T\}$ – автокорреляционная матрица, а $\mathbf{r}_{d\chi} = M\{d(n)\boldsymbol{\chi}_n\}$ – вектор взаимных корреляций.

Ввиду линейности полиномиального фильтра относительно весовых коэффициентов решение (8) выглядит так же, как и в линейном случае. Отличием является то, что матрицы \mathbf{R}_χ и $\mathbf{r}_{d\chi}$ определяются через корреляции высших порядков. С учетом структуры (5) входного вектора $\boldsymbol{\chi}_n$, данные матрицы могут быть представлены в блочном виде следующим образом:

$$\mathbf{R}_\chi = M\{\boldsymbol{\chi}_n \boldsymbol{\chi}_n^T\} = \begin{bmatrix} \mathbf{R}_{0,0} & \mathbf{R}_{0,1} & \cdots & \mathbf{R}_{0,M} \\ \mathbf{R}_{1,0} & \mathbf{R}_{1,1} & \cdots & \mathbf{R}_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{M,0} & \mathbf{R}_{M,1} & \cdots & \mathbf{R}_{M,M} \end{bmatrix}, \quad (9)$$

$$\mathbf{r}_{d\chi} = M\{d(n)\boldsymbol{\chi}_n\} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_M \end{bmatrix},$$

где блоки $\mathbf{R}_{i,j} = M\{\boldsymbol{\chi}_n^{(i)} \boldsymbol{\chi}_n^{(j)T}\}$ и $\mathbf{r}_i = M\{d(n)\boldsymbol{\chi}_n^{(i)}\}$ являются, соответственно, матрицей автокорреляций и вектором взаимных корреляций. Заметим, что в линейном случае данные матрицы состоят лишь из блоков $\mathbf{R}_{0,0}$,

$$\mathbf{R}_{0,1}, \mathbf{R}_{1,0}, \mathbf{R}_{1,1} \text{ и } \mathbf{r}_0, \mathbf{r}_1.$$

На практике для вычисления среднеквадратической ошибки (7) используются реализации конечной длительности. Определим векторы \mathbf{d} и \mathbf{y} размером $L \times 1$, содержащие, соответственно, L отсчетов заданного и выходного сигналов фильтра. Тогда, вектор \mathbf{y} реакции фильтра может быть представлен в матричной форме

$$\mathbf{y} = \mathbf{X}\mathbf{h}, \quad (10)$$

где \mathbf{h} – вектор (3) коэффициентов фильтра, а \mathbf{X} – матрица входного сигнала фильтра, определяемая соглас-

но (5) в следующем виде:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{\chi}_1^T \\ \vdots \\ \boldsymbol{\chi}_L^T \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^T & \mathbf{x}_1^{(2)T} & \cdots & \mathbf{x}_1^{(M)T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{x}_L^T & \mathbf{x}_L^{(2)T} & \cdots & \mathbf{x}_L^{(M)T} \end{bmatrix}. \quad (11)$$

При конечном объеме выборки вместо (7) можно использовать квадрат евклидовой нормы вектора ошибок $\|\varepsilon\|^2 = \varepsilon^T \varepsilon = (\mathbf{d} - \mathbf{y})^T (\mathbf{d} - \mathbf{y})$.

Как известно [15], минимум данной нормы определяется решением системы нормальных уравнений вида

$$\mathbf{X}^T \mathbf{X} \mathbf{h} = \mathbf{X}^T \mathbf{d} \quad (12)$$

и достигается в точке

$$\mathbf{h}_{opt} = \mathbf{X}^+ \mathbf{d}, \quad (13)$$

где \mathbf{X}^+ является обобщенной обратной (псевдообратной) матрицей для \mathbf{X} .

Если столбцы матрицы \mathbf{X} линейно независимы, что бывает далеко не всегда, то псевдообратная матрица определяется выражением

$$\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T.$$

В частности, для квадратной матрицы $\mathbf{X}^+ = \mathbf{X}^{-1}$. При неполном ранге для определения \mathbf{X}^+ можно воспользоваться сингулярным представлением матрицы $\mathbf{X} = \mathbf{U} \Lambda \mathbf{V}$. Здесь \mathbf{U} и \mathbf{V} являются унитарными матрицами со столбцами, равными собственным векторам, соответственно, матриц $\mathbf{X} \mathbf{X}^T$ и $\mathbf{X}^T \mathbf{X}$, а Λ – диагональной матрицей вида

$$\Lambda = \begin{bmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{bmatrix},$$

где $\mathbf{D} = \text{diag}(\sigma_1, \dots, \sigma_k)$, а σ_i – отличные от нуля сингулярные числа \mathbf{X} . Псевдо-обратная матрица \mathbf{X}^+ , соответствующая матрице \mathbf{X} ранга k , равна

$$\mathbf{X}^+ = \mathbf{V} \mathbf{D}^{-1} \mathbf{U}.$$

В соответствии с (10) и (11) выходной вектор \mathbf{y} формируется путем взвешенного суммирования компонент \mathbf{y}_i , $i = 0, \dots, M$, обусловленных отдельными столбцами матрицы \mathbf{X} . Первый столбец определяет постоянную составляющую y_0 ; столбец из блоков \mathbf{x}_i – линейную составляющую y_1 ; $\mathbf{x}_i^{(2)}$ – нелинейную составляющую y_2 второго порядка, и т.д. до M -го порядка включительно.

Непосредственное вычисление (13) требует наличия полных реализаций входного и выходного сигналов фильтра и из-за большой размерности сопряжено с большим объемом вычислительных затрат. При обработке реальных сигналов на практике требуется постоянно корректировать параметры фильтра, адаптируя его поведение в зависимости от внешней среды, что и обуславливает использование алгоритмов адаптивной фильтрации.

Градиентные алгоритмы адаптации

Градиентный адаптивный алгоритм для класса полиномиальных фильтров определяется дифференцированием среднеквадратической ошибки $\varepsilon = M\{d(n) - y(n)\}^2$ между опорным $d(n)$ и реальным $y(n)$ сигналами на

выходе фильтра по вектору \mathbf{h} коэффициентов и выглядит следующим образом:

$$\mathbf{h}^{[n]} = \mathbf{h}^{[n-1]} - \mu \hat{\nabla}_n = \mathbf{h}^{[n-1]} + \mu \varepsilon^{[n]} \chi_n, \quad (14)$$

где μ – параметр адаптации (шаг); $\varepsilon^{[n]}$ – величина ошибки рассогласования, равная $d(n) - y(n)$; $\hat{\nabla}_n$ – оценка вектора градиента $\partial \varepsilon / \partial \mathbf{h}$; \mathbf{h} – вектор коэффициентов фильтра вида (3).

В силу линейности полиномиальных фильтров относительно коэффициентов внешний вид алгоритма (14) такой же, как и в случае линейной фильтрации. Отличие состоит в том, что здесь вместо вектора \mathbf{x}_n отсчетов входного сигнала фигурирует вектор χ_n , определяемый (5) и включающий в себя наряду с \mathbf{x}_n также векторы $\mathbf{x}_n^{(m)}$, $m = 2, \dots, M$ произведений входных отсчетов, соответствующие различным нелинейным составляющим полиномиального фильтра.

С целью сокращения дальнейших записей будем использовать следующие обозначения: $\chi_{n,0} = 1$, $\chi_{n,m} = \mathbf{x}_n^{(m)}$, $\mathbf{h}_0 = h_0$, $m = 1, \dots, M$. Тогда алгоритм (7) может быть представлен в блочном виде

$$\begin{bmatrix} \mathbf{h}_0^{[n]} \\ \mathbf{h}_1^{[n]} \\ \vdots \\ \mathbf{h}_M^{[n]} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_0^{[n-1]} \\ \mathbf{h}_1^{[n-1]} \\ \vdots \\ \mathbf{h}_M^{[n-1]} \end{bmatrix} + \mu \varepsilon^{[n]} \begin{bmatrix} \chi_{n,0} \\ \chi_{n,1} \\ \vdots \\ \chi_{n,M} \end{bmatrix}. \quad (15)$$

Единственным параметром, с помощью которого можно влиять на поведение алгоритма (15), является величина μ шага. При малых значениях μ скорость адаптации будет невысока, но сходимость алгоритма гарантируется в достаточно малую окрестность оптимальной точки. Возрастание μ приводит к более быстрой адаптации, но с большей предельной ошибкой. Чрезмерное увеличение μ может нарушить устойчивость процесса адаптации, и алгоритм будет расходиться.

Как известно [2], в линейном случае сходимость алгоритма вида (14) гарантируется, если параметр μ лежит в диапазоне $0 < \mu < 2 / \lambda_{\max}$, где λ_{\max} – наибольшее собственное значение корреляционной матрицы \mathbf{R}_x входного процесса. Очевидно, что данная оценка будет также справедлива и в нелинейном случае с той лишь разницей, что матрица \mathbf{R}_x заменяется матрицей \mathbf{R}_χ моментов высших порядков вида (9).

Данная матрица включает в себя матрицу \mathbf{R}_x , объединяющую блоки $\mathbf{R}_{0,0}, \mathbf{R}_{0,1}, \mathbf{R}_{1,0}, \mathbf{R}_{1,1}$. Так как значение λ_{\max} не может превышать след матрицы, равный сумме ее диагональных элементов [16], а $\text{tr}(\mathbf{R}_x) < \text{tr}(\mathbf{R}_\chi)$, в нелинейном случае параметр μ может выбираться в меньшем диапазоне, чем в линейном. Это снижает потенциальную скорость сходимости алгоритма (15), что ограничивает его непосредственное использование для адаптации полиномиальных фильтров.

Скорость сходимости может быть увеличена путем

представления полиномиального фильтра в виде компонент с некоррелированными выходами. В этом случае адаптация каждой из них может выполняться независимо, что позволяет снизить размерность исходной задачи и использовать различные параметры μ для адаптации отдельных составляющих фильтра. Полную декорреляцию всех компонент фильтра, обусловленных нелинейностями различных порядков, возможно осуществить с помощью довольно трудоемкой операции ортогонализации [17]. Для входных процессов с симметричной плотностью распределения вероятности, в основном встречающихся на практике, можно предложить более простой способ.

Так как для таких процессов все моменты нечетного порядка равны нулю, в матрице (9) все блоки $\mathbf{R}_{i,j} = 0$, для $i + j = 2k + 1$. Это дает возможность представить систему нормальных уравнений (8), определяющих оптимальный вектор коэффициентов, в виде двух независимых подсистем относительно четных и нечетных моментов:

$$\mathbf{R}_{\chi_\alpha} \mathbf{h}_\alpha = \mathbf{r}_{d\chi_\alpha}, \quad \mathbf{R}_{\chi_\beta} \mathbf{h}_\beta = \mathbf{r}_{d\chi_\beta}. \quad (16)$$

Здесь $\mathbf{R}_{\chi_\alpha} = \chi_{\alpha,n} \chi_{\alpha,n}^T$ и $\mathbf{R}_{\chi_\beta} = \chi_{\beta,n} \chi_{\beta,n}^T$, а входные векторы $\chi_{\alpha,n}$, $\chi_{\beta,n}$ и векторы \mathbf{h}_α , \mathbf{h}_β коэффициентов определяются в виде

$$\chi_{\alpha,n}^T = [1 \mid \chi_{n,2}^T \mid \cdots \mid \chi_{n,M_\alpha}^T],$$

$$\chi_{\beta,n}^T = [\chi_{n,1}^T \mid \chi_{n,3}^T \mid \cdots \mid \chi_{n,M_\beta}^T],$$

$$\mathbf{h}_\alpha^T = [h_0 \mid h_2^T \mid \cdots \mid h_{M_\alpha}^T],$$

$$\mathbf{h}_\beta^T = [h_1^T \mid h_3^T \mid \cdots \mid h_{M_\beta}^T],$$

где $M_\alpha = 2[M/2]$, $M_\beta = 2[M/2] + 1$, а $[M/2]$ означает ближайшее целое, не превосходящее $M/2$.

В соответствии с (16), алгоритм (15) адаптации теперь будет выглядеть следующим образом:

$$\begin{bmatrix} \mathbf{h}_\alpha^{[n]} \\ \mathbf{h}_\beta^{[n]} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_\alpha^{[n-1]} \\ \mathbf{h}_\beta^{[n-1]} \end{bmatrix} + \begin{bmatrix} \mu_\alpha & 0 \\ 0 & \mu_\beta \end{bmatrix} \varepsilon^{[n]} \begin{bmatrix} \chi_{\alpha,n} \\ \chi_{\beta,n} \end{bmatrix} \quad (17)$$

и фактически представляет собой два независимых алгоритма для четных и нечетных составляющих фильтра с параметрами μ_α и μ_β .

Если обозначить через $\lambda_{\alpha,\max}$ и $\lambda_{\beta,\max}$ максимальные собственные значения, соответственно, матриц \mathbf{R}_{χ_α} и \mathbf{R}_{χ_β} , то сходимость алгоритма (17) будет гарантирована при условиях $0 < \mu_\alpha < 2/\lambda_{\alpha,\max}$ и $0 < \mu_\beta < 2/\lambda_{\beta,\max}$, которые с использованием оценки λ_{\max} , через след матрицы могут быть также записаны в виде

$$0 < \mu_\alpha < 2 / \text{tr}(\mathbf{R}_{\chi_\alpha}), \quad 0 < \mu_\beta < 2 \text{tr}(\mathbf{R}_{\chi_\beta}). \quad (18)$$

Так как параметры μ_α и μ_β адаптации в данном случае могут выбираться в более широком диапазоне, чем μ , следует ожидать, что по сравнению с исходным алгоритмом скорость сходимости алгоритма (17) будет выше.

Рассмотрим в качестве примера адаптивный нелинейный фильтр третьего порядка для входного сигнала $x(n)$ в виде гауссовского процесса с нулевым средним и дисперсией σ_x^2 . Для такого процесса все моменты нечетного порядка равны нулю, а моменты четного порядка определяются соотношением [18]

$$M\{x(n_1)x(n_2)\cdots x(n_{2k})\} = \sum \prod M\{x(n_i)x(n_j)\}, \quad (19)$$

где суммирование производится по всем разбиениям совокупности $(1, 2, \dots, 2k)$ на пары, а произведение – по всем парам в каждом разбиении. При этом количество таких разбиений равно $(2k)!/k!2^k$.

Если импульсные характеристики $h_m(n_1, \dots, n_m)$ фильтра имеют длительности N_m , $m=1, 2, 3$, то, согласно (3), размеры соответствующих блоков \mathbf{h}_m вектора коэффициентов будут равны $C_{N_m+m-1}^m$. Возьмем для наглядности $N_1 = N_2 = N_3 = 3$, тогда входные векторы $\chi_{\alpha,n}$ и $\chi_{\beta,n}$ приобретают вид

$$\begin{aligned} \chi_{\alpha,n} &= \left[1 \mid x_1^2 \quad x_2^2 \quad x_3^2 \mid x_1x_2 \quad x_1x_3 \quad x_2x_3 \right], \\ \chi_{\beta,n} &= \left[x_1 \quad x_2 \quad x_3 \mid x_1x_2^2 \quad x_1x_3^2 \quad x_2x_1^2 \quad x_2x_3^2 \quad x_3x_1^2 \quad x_3x_2^2 \quad x_1x_2x_3 \right], \end{aligned}$$

где с целью сокращения записи для обозначения $x(n-i)$ используется x_i .

С учетом (19) матрицы моментов, соответствующие векторам $\chi_{\alpha,n}$ и $\chi_{\beta,n}$, будут выглядеть следующим образом:

$$\mathbf{R}_{\chi_\alpha} = \left[\begin{array}{cccc|c} 1 & \sigma_x^2 & \sigma_x^2 & \sigma_x^2 & \mathbf{0}_{3 \times 3} \\ \sigma_x^2 & 3\sigma_x^4 & \sigma_x^4 & \sigma_x^4 & \\ \sigma_x^2 & \sigma_x^4 & 3\sigma_x^4 & \sigma_x^4 & \\ \sigma_x^2 & \sigma_x^4 & \sigma_x^4 & 3\sigma_x^4 & \\ \hline & \mathbf{0}_{3 \times 3} & & & \mathbf{D}_3(\sigma_x^4) \end{array} \right], \quad (20)$$

$$\mathbf{R}_{\chi_\beta} = \left[\begin{array}{c|c|c|c} \mathbf{D}_3(\sigma_x^2) & \mathbf{D}_3(3\sigma_x^4) & \mathbf{G}_{3 \times 6}(\sigma_x^4) & \mathbf{0}_{3 \times 1} \\ \mathbf{D}_3(3\sigma_x^4) & \mathbf{D}_3(15\sigma_x^6) & \mathbf{G}_{3 \times 6}(3\sigma_x^6) & \mathbf{0}_{3 \times 1} \\ \mathbf{G}_{3 \times 6}^T(\sigma_x^4) & \mathbf{G}_{3 \times 6}^T(3\sigma_x^6) & \mathbf{D}_6(3\sigma_x^6) & \mathbf{0}_{6 \times 1} \\ \hline \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 6} & \sigma_x^6 \end{array} \right], \quad (21)$$

где $\mathbf{D}_L(a)$ обозначает диагональную матрицу размером $L \times L$ с элементами, равными a на диагонали; $\mathbf{0}_{L \times K}$ матрицу $L \times K$, целиком состоящую из нулей, а матрица $\mathbf{G}_{L \times K}(a)$ размером $L \times K$ имеет вид

$$\mathbf{G}_{L \times K}(a) = \begin{bmatrix} a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & a & a & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & a & a \end{bmatrix}$$

и состоит из строк, содержащих сдвинутые пары элементов, равные a .

Зная дисперсию σ_x^2 входного сигнала (мощность), на основании полученных выражений (20) и (21) можно

рассчитать максимальные собственные числа $\lambda_{\alpha,\max}$ и $\lambda_{\beta,\max}$, определяющие границы изменения параметров μ_α и μ_β адаптации. Для определения приближенных границ воспользуемся оценками (18). В результате получим

$$0 < \mu_\alpha < \frac{2}{1+12\sigma_x^4}, \quad 0 < \mu_\beta < \frac{2}{3\sigma_x^2 + 64\sigma_x^6},$$

Заметим, что структура матриц (20) и (21) сохраняется при произвольных значениях N_m , $m=1, 2, 3$, а изменения будут касаться только размеров блоков. Это дает возможность без труда получить следующие оценки диапазона изменения параметров адаптации для общего случая

$$0 < \mu_\alpha < \frac{2}{1+(3N_2+C_{N_2}^2)\sigma_x^4},$$

$$0 < \mu_\beta < \frac{2}{N_1\sigma_x^2 + (1+15N_3+2C_{N_3}^2)\sigma_x^6}.$$

Рассмотренные алгоритмы адаптации вида (15) и (17) предусматривают одновременную коррекцию векторов \mathbf{h}_m , $m=0, 1, \dots, M$, коэффициентов различных порядков и реализуются фильтром с параллельной структурой, показанной на рис. 1 а. Выбор параметра μ для алгоритма (15) определяется матрицей \mathbf{R}_χ большой размерности, равной $C_{N+M}^M \times C_{N+M}^M$. Раздельная адаптация четных и нечетных составляющих фильтра, согласно алгоритму (17), основанная на разбиении матрицы \mathbf{R}_χ на две \mathbf{R}_{χ_α} и \mathbf{R}_{χ_β} , позволяет увеличить скорость сходимости за счет использования различных параметров μ_α и μ_β адаптации для каждой из них. Однако из-за того, что матрицы \mathbf{R}_{χ_α} и \mathbf{R}_{χ_β} содержат моменты различных порядков, диапазон изменения $\lambda_{\min} - \lambda_{\max}$ собственных чисел может все же оставаться большим. Это, как известно [1, 2], приводит к неравномерной сходимости различных компонент (мод) вектора ошибки.

Для выравнивания скорости сходимости целесообразно использовать отдельные параметры μ_m , $m=0, 1, \dots, M$, для каждой составляющей фильтра. Это становится возможным, если вместо параллельной коррекции векторов \mathbf{h}_m коэффициентов фильтра использовать последовательную в направлении увеличения порядка m нелинейности его составляющих.

Теоретической основой такого процесса может стать метод построения оптимального полиномиального фильтра с помощью последовательных приближений [15]. В соответствии с данным методом вектор \mathbf{h}_m коэффициентов, определяемый на m -м этапе процедуры, представляет собой решение системы нормальных уравнений:

$$\mathbf{R}_{m,m}\mathbf{h}_m = \mathbf{M}\{e_{m-1}(n)\chi_{n,m}\}, \quad (22)$$

где частичная ошибка $e_{m-1}(n)$ формируется на основе полученных на предыдущих этапах векторов \mathbf{h}_i , $i=0, 1, \dots, m-1$, и равна

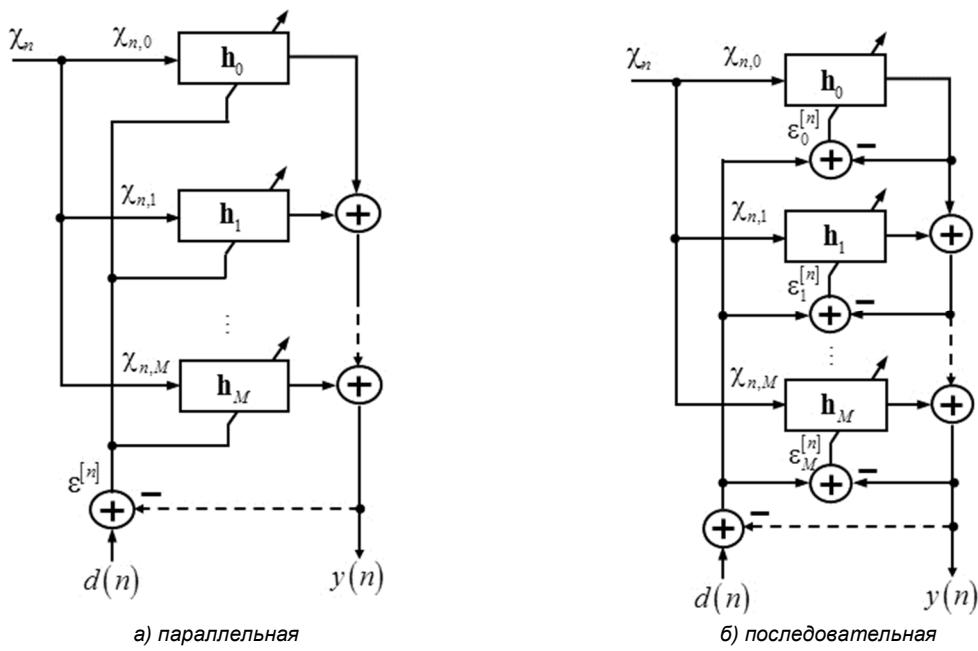


Рис. 1 Структуры адаптивных полиномиальных фильтров

$$e_{m-1}(n) = d(n) - y_{m-1}(n) = d(n) - \sum_{i=0}^{m-1} \mathbf{h}_i^T \boldsymbol{\chi}_{n,i}. \quad (23)$$

Подставляя (23) в выражение (22), данную процедуру можно также представить как последовательность решений матричных уравнений вида

$$\mathbf{R}_{m,0} \mathbf{h}_0 + \mathbf{R}_{m,1} \mathbf{h}_1 + \dots + \mathbf{R}_{m,m-1} \mathbf{h}_{m-1} + \mathbf{R}_{m,m} \mathbf{h}_m = \mathbf{r}_m, \quad (24)$$

$$m = 0, 1, \dots, M,$$

относительно \mathbf{h}_m при заданных \mathbf{h}_i , $i = 0, 1, \dots, m-1$.

Применение градиентного метода для решения уравнений (22) приводит к следующему адаптивному алгоритму:

$$\begin{aligned} \varepsilon_m^{[n]} &= d(n) - y_{m-1}(n) - \boldsymbol{\chi}_{n,m}^T \mathbf{h}_m^{[n-1]}, \\ \mathbf{h}_m^{[n]} &= \mathbf{h}_m^{[n-1]} + \mu_m \varepsilon_m^{[n]} \boldsymbol{\chi}_{n,m}, \\ y_m(n) &= y_{m-1}(n) + \boldsymbol{\chi}_{n,m}^T \mathbf{h}_m^{[n]}, \quad m = 0, 1, \dots, M. \end{aligned} \quad (25)$$

Согласно (25), адаптация векторов \mathbf{h}_m происходит последовательно по отдельным алгоритмам с различными параметрами μ_m адаптации для каждого из них.

Формирование очередного вектора $\mathbf{h}_m^{[n]}$ на n -й итерации осуществляется на основе ошибки $\varepsilon_m^{[n]}$, для вычисления которой используется вектор $\mathbf{h}_m^{[n-1]}$, полученный на $(n-1)$ -й итерации, и реакция $y_{m-1}(n)$ фильтра $(m-1)$ -го порядка, зависящая от значений векторов $\mathbf{h}_i^{[n]}$, $i = 0, 1, \dots, m-1$ на n -й итерации. Структура нелинейного фильтра с последовательной адаптацией показана на рис. 1 б.

Для определения условий, гарантирующих сходимость алгоритма адаптации, запишем разностное уравнение относительно средних значений вектора коэффициентов. Вычисляя математическое ожидание от обеих частей уравнения коррекции вектора, получим

$$\mathbf{M}\{\mathbf{h}_m^{[n-1]}\} = \mathbf{M}\{\mathbf{h}_m^{[n-1]}\} + \mu_m \mathbf{M}\{\varepsilon_m^{[n]} \boldsymbol{\chi}_{n,m}\} = \mathbf{M}\{\mathbf{h}_m^{[n-1]}\} +$$

$$+ \mu_m [\mathbf{r}_m - \mathbf{M}\{y_{m-1}(n) \boldsymbol{\chi}_{n,m}\} - \mathbf{M}\{\boldsymbol{\chi}_{n,m} \boldsymbol{\chi}_{n,m}^T \mathbf{h}_m^{[n-1]}\}].$$

Последовательно раскрывая $y_{m-1}(n)$ с помощью последнего выражения из (25) и предполагая, что векторы коэффициентов статистически независимы от входных векторов преобразуем данное уравнение к виду

$$\begin{aligned} \mathbf{M}\{\mathbf{h}_m^{[n]}\} &= \mathbf{M}\{\mathbf{h}_m^{[n-1]}\} + \\ &+ \mu_m \left(\mathbf{r}_m - \sum_{i=0}^{m-1} \mathbf{R}_{m,i} \mathbf{h}_i^{[n]} - \mathbf{R}_{m,m} \mathbf{h}_m^{[n-1]} \right). \end{aligned} \quad (26)$$

Обозначим через $\tilde{\mathbf{h}}_m$, $m = 0, 1, \dots, M$ векторы коэффициентов, являющиеся решением системы уравнений (24), и введем в рассмотрение величины $\mathbf{e}_m^{[n]} = \mathbf{M}\{\mathbf{h}_m^{[n]}\} - \tilde{\mathbf{h}}_m$, характеризующие отклонение средних значений коэффициентов от данного решения. После подстановки в (26) значение \mathbf{r}_m из (24) будем иметь

$$\mathbf{e}_m^{[n]} = \mathbf{e}_m^{[n-1]} - \mu_m \left(\sum_{i=0}^{m-1} \mathbf{R}_{m,i} \mathbf{e}_i^{[n]} - \mathbf{R}_{m,m} \mathbf{e}_m^{[n-1]} \right),$$

$$m = 0, 1, \dots, M.$$

Полученная система разностных уравнений может быть представлена в более компактной форме

$$\mathbf{A} \mathbf{e}_m^{[n]} = (\mathbf{I} - \mu \mathbf{B}) \mathbf{e}_m^{[n-1]}, \quad (27)$$

где $\mathbf{e}_m^{[n]} = [(\mathbf{e}_0^{[n]})^T \mid (\mathbf{e}_1^{[n]})^T \mid \dots \mid (\mathbf{e}_M^{[n]})^T]^T$, а блочно-диагональные матрицы μ , \mathbf{B} и нижняя блочно-треугольная матрица \mathbf{A} имеют следующий вид:

$$\mu = \begin{bmatrix} \mu_0 \mathbf{I}_{L_1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mu_1 \mathbf{I}_{L_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mu_2 \mathbf{I}_{L_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mu_M \mathbf{I}_{L_M} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{R}_{0,0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{1,1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{2,2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{R}_{M,M} \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mu_1 \mathbf{R}_{1,0} & \mathbf{I}_{L_1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mu_2 \mathbf{R}_{2,0} & \mu_2 \mathbf{R}_{2,1} & \mathbf{I}_{L_2} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_M \mathbf{R}_{M,0} & \mu_M \mathbf{R}_{M,0} & \mu_M \mathbf{R}_{M,0} & \dots & \mu_M \mathbf{R}_{M,M-1} & \mathbf{I}_{L_M} \end{bmatrix},$$

где размеры L_m блоков равны $C_{N_m+m-1}^m$, $m=1, \dots, M$.

Матрица \mathbf{B} допускает спектральное представление в виде $\mathbf{B} = \mathbf{U}\Lambda\mathbf{U}^{-1}$, где \mathbf{U} – унитарная матрица, столбцами которой являются собственные векторы \mathbf{B} , а Λ – диагональная матрица собственных значений \mathbf{B} . В силу блочно-диагональной структуры \mathbf{B} матрица Λ определяется через матрицы Λ_m , состоящие из собственных чисел отдельных блоков, и равна $\text{diag}(1, \Lambda_1, \dots, \Lambda_M)$.

Воспользуемся теперь приемом из [2], записав (27) в новой системе координат. Используя подстановку $\mathbf{e}^{[n]} = \mathbf{U}\mathbf{v}^{[n]}$ и учитывая коммутативность матриц μ и \mathbf{U} , получим

$$\mathbf{D}\mathbf{v}^{[n]} = (\mathbf{I} - \mu\Lambda)\mathbf{v}^{[n-1]}, \quad (28)$$

где $\mathbf{D} = \mathbf{U}^{-1}\Lambda\mathbf{U}$. Обозначим $\gamma_0 = 1 - \mu_0$, $\gamma_m = \mathbf{I} - \mu_m\Lambda_m$, $m=1, \dots, M$, тогда матрица $(\mathbf{I} - \mu\Lambda)$ будет равна $\text{diag}(\gamma_0, \gamma_1, \dots, \gamma_M)$. Если параметры μ_m адаптации выбрать из условий

$$0 < \mu_0 < 1, \quad 0 < \mu_m < 2/\lambda_{m,\max}, \quad (29)$$

где $\lambda_{m,\max}$ – максимальное собственное число блока $\mathbf{R}_{m,m}$, то будет справедливо

$$\lim_{n \rightarrow \infty} \gamma_0^n = 0, \quad \lim_{n \rightarrow \infty} \gamma_m^n = \mathbf{0}, \quad m=1, \dots, M. \quad (30)$$

При отсутствии корреляции между различными входными векторами $\chi_{n,i}$ матрицы $\mathbf{R}_{i,j} = \mathbf{0}$ для $i \neq j$. В этом случае $\mathbf{D} = \mathbf{I}$ и из (30) с очевидностью следует, что

$$\lim_{n \rightarrow \infty} \mathbf{v}_m^{[n]} = \mathbf{v}_m^{[0]} \lim_{n \rightarrow \infty} \gamma_m^n = \mathbf{0}.$$

Покажем теперь, что при условиях (29) сходимость алгоритма будет обеспечиваться также и в общем случае. Прежде всего заметим, что матрицы \mathbf{U} и \mathbf{U}^{-1} являются блочно-диагональными так же, как и матрица \mathbf{B} . Непосредственной проверкой можно убедиться, что преобразование $\mathbf{U}^{-1}\Lambda\mathbf{U}$ не изменяет структуру матрицы \mathbf{A} , оставляя ее блочно-треугольной. В результате матрица \mathbf{D} будет отличаться от \mathbf{A} только своими блоками, которые обозначим $\mathbf{D}_{i,j}$.

Сходимость $\mathbf{v}_m^{[n]} \rightarrow \mathbf{0}$ доказывается по индукции.

Для $m=0$ имеем $\mathbf{v}_0^{[n]} = \gamma_0 \mathbf{v}_0^{[n-1]} = \gamma_0^n \mathbf{v}_0^{[0]} \rightarrow \mathbf{0}$. Используя разностное уравнение (28), выразим $\mathbf{v}_1^{[n]}$ через $\mathbf{v}_0^{[0]}$ и $\mathbf{v}_1^{[0]}$ следующим образом:

$$\mathbf{v}_1^{[n]} = \gamma_1^n \mathbf{v}_1^{[0]} - \mathbf{D}_{1,0} \left(\sum_{i=1}^n \gamma_1^{n-i} \gamma_0^i \right) \mathbf{v}_0^{[0]}. \quad (31)$$

Так как все члены в (31) имеют порядок $O(\gamma_0^i \gamma_1^j)$, $i+j=n$, имеем $\mathbf{v}_m^{[n]} \rightarrow \mathbf{0}$. Аналогичным образом можно показать, что

$$\mathbf{v}_m^{[n]} = \gamma_m^n \mathbf{v}_m^{[0]} - \sum_{i=0}^{m-1} \mathbf{D}_{m,i} \sum_{j=1}^n \gamma_m^{n-j} \mathbf{v}_i^{[j]}. \quad (32)$$

Учитывая, что $\mathbf{v}_i^{[j]} \sim O(\gamma_0^{k_0} \gamma_1^{k_1} \dots \gamma_i^{k_i})$, $k_0 + \dots + k_i = j$ для $i < m$, на основании (32) будем иметь $\mathbf{v}_m^{[n]} \rightarrow \mathbf{0}$.

На практике для выбора параметров μ_m адаптации вместо (29) можно воспользоваться оценками $0 < \mu_m < 2/\text{tr}(\mathbf{R}_{m,m})$. В частности, для нелинейного фильтра третьего порядка с некоррелированным гауссовым процессом на входе, имеющим дисперсию σ_x^2 , параметры адаптации будут определяться следующим образом:

$$0 < \mu_0 < 1, \quad 0 < \mu_1 < \frac{2}{N_1 \sigma_x^2}, \quad 0 < \mu_2 < \frac{2}{(3N_2 + C_{N_2}^2) \sigma_x^4},$$

$$0 < \mu_3 < \frac{2}{(1+15N_3 + 2C_{N_3}^2) \sigma_x^6}.$$

Хотя последовательный алгоритм адаптации обладает более высокой скоростью сходимости, чем параллельный, в общем случае нельзя гарантировать, что он сходится к оптимальной точке. Другими словами, $\mathbf{M}\{\mathbf{h}^{[n]}\} = \tilde{\mathbf{h}} \neq \mathbf{h}_{\text{opt}}$, т.к. системы нормальных уравнений (8) и (24), определяющие, соответственно, векторы \mathbf{h}_{opt} и $\tilde{\mathbf{h}}$, отличны друг от друга.

Для достижения необходимой точности без существенной потери скорости адаптации можно совместить последовательный алгоритм с параллельным, используя последний для уточнения текущего вектора коэффициентов. Другим возможным решением является добавление к последовательной схеме на рис. 1 б дополнительной обратной связи (показана пунктиром), периодическое замыкание которой будет приводить к снижению систематической ошибки адаптации.

Рекурсивный алгоритм адаптации с экспоненциальным взвешиванием

В рассмотренных алгоритмах адаптации для оценки вектора градиента использовалось значение ошибки рассогласования лишь в одной (текущей) точке. Как следствие этого, вектор коэффициентов будет иметь значительную случайную составляющую. Другим недостатком градиентных алгоритмов является неравномерная сходимость различных компонент вектора коэффициентов. Согласно [2], такая неравномерность увеличивается с ростом диапазона изменения собственных чисел корреляционной матрицы. В частности, можно показать, что для последовательного алгоритма адаптации постоянные времени τ_m , характеризующие сходимость различных компонент \mathbf{h}_m вектора коэффициентов, будут определяться отношением $\lambda_{m,\max}/2\lambda_{m,\min}$, где $\lambda_{m,\max}$

и $\lambda_{m,\min}$ максимальное и минимальное собственные числа матрицы $\mathbf{R}_{m,m}$.

Улучшение отмеченных свойств процесса адаптации возможно за счет использования алгоритмов ньютоновского типа с экспоненциальным взвешиванием, называемых также рекурсивными алгоритмами наименьших квадратов [2]. Хотя данные алгоритмы требуют больших вычислительных затрат по сравнению с градиентными, они обладают более высокой и равномерной скоростью сходимости, а также менее подвержены влиянию шумов [1, 2].

Рассмотрим последовательный процесс адаптации полиномиальных фильтров, основанный на использовании данного класса алгоритмов. Пусть на n -й итерации уже получены компоненты $\mathbf{h}_0^{[n]}, \mathbf{h}_1^{[n]}, \dots, \mathbf{h}_{m-1}^{[n]}$ вектора коэффициентов. Вычисление очередного вектора $\mathbf{h}_m^{[n]}$ осуществляется путем минимизации взвешенной суммы квадратов ошибок:

$$\min \sum_{i=0}^n \alpha^{n-i} e_m^2(i), \quad (33)$$

где постоянная α «забывания» определяет степень влияния предыдущих значений ошибки и выбирается в диапазоне $0 \leq \alpha \leq 1$, а частичная ошибка $e_m(i)$ определяется выражением (23).

Решение $M+1$ задач вида (33) для различных значений $m = 0, 1, \dots, M$ приводит к следующей системе нормальных уравнений:

$$\sum_{i=0}^m \mathbf{R}_{m,i}^{[n]} \mathbf{h}_i^{[n]} = \mathbf{r}_m^{[n]}, \quad m = 0, 1, \dots, M, \quad (34)$$

которая отличается от (24) тем, что вместо точных значений матриц $\mathbf{R}_{m,j}$ и векторов \mathbf{r}_m содержит их оценки вида

$$\mathbf{R}_{m,j}^{[n]} = \sum_{i=0}^n \alpha^{n-i} \boldsymbol{\chi}_{i,m} \boldsymbol{\chi}_{i,j}^T, \quad \mathbf{r}_m^{[n]} = \sum_{i=0}^n \alpha^{n-i} d(n) \boldsymbol{\chi}_{i,m}.$$

Данные оценки могут определяться рекурсивно с помощью выражений

$$\mathbf{R}_{m,j}^{[n]} = \alpha \mathbf{R}_{m,j}^{[n-1]} + \boldsymbol{\chi}_{n,m} \boldsymbol{\chi}_{n,j}^T, \quad \mathbf{r}_m^{[n]} = \alpha \mathbf{r}_m^{[n-1]} + d(n) \boldsymbol{\chi}_{n,m}. \quad (35)$$

Для построения алгоритма адаптации решим (34) относительно $\mathbf{h}_m^{[n]}$, считая $\mathbf{h}_0^{[n]}, \mathbf{h}_1^{[n]}, \dots, \mathbf{h}_{m-1}^{[n]}$ известными

$$\mathbf{h}_m^{[n]} = \mathbf{P}_m^{[n]} \left(\mathbf{r}_m^{[n]} - \sum_{i=0}^{m-1} \mathbf{R}_{m,i}^{[n]} \mathbf{h}_i^{[n]} \right), \quad (36)$$

где $\mathbf{P}_m^{[n]}$ представляет собой матрицу, обратную $\mathbf{R}_{m,m}^{[n]}$. С учетом (35) данное выражение можно представить в виде

$$\begin{aligned} \mathbf{h}_m^{[n]} &= \mathbf{P}_m^{[n]} \left[\alpha \left(\mathbf{r}_m^{[n-1]} - \sum_{i=0}^{m-1} \mathbf{R}_{m,i}^{[n-1]} \mathbf{h}_i^{[n]} \right) + \right. \\ &\quad \left. + \boldsymbol{\chi}_{n,m} \left(d(n) - \sum_{i=0}^{m-1} \boldsymbol{\chi}_{n,i}^T \mathbf{h}_i^{[n]} \right) \right] = \\ &= \mathbf{P}_m^{[n]} \left[\alpha \boldsymbol{\Theta}_{n,m} + \boldsymbol{\chi}_{n,m} e_{m-1}(n) \right], \end{aligned} \quad (37)$$

$$\text{где } \boldsymbol{\Theta}_{n,m} = \mathbf{r}_m^{[n-1]} - \sum_{i=0}^{m-1} \mathbf{R}_{m,i}^{[n-1]} \mathbf{h}_i^{[n]}.$$

Используя для вычисления $\mathbf{P}_m^{[n]}$ рекурсивный метод обращения матриц [2], можно показать, что

$$\begin{aligned} \mathbf{P}_m^{[n]} &= \frac{1}{\alpha} \left(\mathbf{P}_m^{[n-1]} - \mathbf{s}_m^{[n]} \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \right), \\ \mathbf{s}_m^{[n]} &= \frac{\mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}}{\alpha + \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}}. \end{aligned} \quad (38)$$

Подставим выражение (38) для $\mathbf{P}_m^{[n]}$ в (37) и выполним следующие преобразования:

$$\begin{aligned} \mathbf{h}_m^{[n]} &= \mathbf{P}_m^{[n-1]} \boldsymbol{\Theta}_{n,m} - \mathbf{s}_m^{[n]} \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\Theta}_{n,m} + \\ &+ \frac{1}{\alpha} \left(\mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m} - \frac{\mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m} \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}}{\lambda + \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}} \right) e_{m-1}(n) = \\ &= \mathbf{P}_m^{[n-1]} \boldsymbol{\Theta}_{n,m} - \mathbf{s}_m^{[n]} \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\Theta}_{n,m} + \mathbf{s}_m^{[n]} e_{m-1}(n). \end{aligned}$$

Для упрощения полученного выражения предположим, что $\mathbf{h}_i^{[n]} \approx \mathbf{h}_i^{[n-1]}$ для $i = 0, 1, \dots, m-1$. Тогда на основании (36) имеем $\mathbf{P}_m^{[n-1]} \boldsymbol{\Theta}_{n,m} \approx \mathbf{h}_m^{[n-1]}$, и алгоритм адаптации можно представить в виде

$$\begin{aligned} \mathbf{e}_m^{[n]} &= d(n) - y_{m-1}(n) - \boldsymbol{\chi}_{n,m}^T \mathbf{h}_m^{[n-1]}, \\ \mathbf{P}_m^{[n]} &= \frac{1}{\alpha} \left(\mathbf{P}_m^{[n-1]} - \mathbf{s}_m^{[n]} \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \right), \\ \mathbf{s}_m^{[n]} &= \frac{\mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}}{\alpha + \boldsymbol{\chi}_{n,m}^T \mathbf{P}_m^{[n-1]} \boldsymbol{\chi}_{n,m}}, \\ \mathbf{h}_m^{[n]} &= \mathbf{h}_m^{[n-1]} + \mu \mathbf{s}_m^{[n]} \mathbf{e}_m^{[n]}, \\ y_m(n) &= y_{m-1}(n) + \boldsymbol{\chi}_{n,m}^T \mathbf{h}_m^{[n]}, \quad m = 0, 1, \dots, M. \end{aligned} \quad (39)$$

По сравнению с (25) данный алгоритм предусматривает рекурсивное вычисление матрицы $\mathbf{P}_m^{[n]}$, на что дополнительно затрачивается приблизительно $O(L_m^2)$ операций умножения. В то же время, поскольку при вычислении вектора $\mathbf{s}_m^{[n]}$, определяющего направление поиска, используется взвешенное суммирование предшествующих отсчетов входного сигнала, процесс адаптации по алгоритму (39) имеет более гладкий характер и обеспечивает меньшие флуктуации при приближении к оптимальной точке. При этом величина шумовой составляющей коэффициентов фильтра зависит от памяти фильтра и определяется постоянной α .

Можно показать, что система разностных уравнений относительно векторов ошибок $\mathbf{e}_m^{[n]} = \mathbf{M}\{\mathbf{h}_m^{[n]}\} - \tilde{\mathbf{h}}_m$, характеризующая свойства сходимости алгоритма (39) в среднем будет определяться выражением (27), если в матрицах \mathbf{M} , \mathbf{A} , \mathbf{B} произвести формальную подстановку $\mu_m = \mu \mathbf{P}_m^{[n]}$. Тогда на основании (29) условия, гарантирующие сходимость алгоритма, принимают вид $0 < \mu < 2/\lambda_{m,\max}$, где $\lambda_{m,\max}$ – максимальное собственное число произведения матриц $\mathbf{P}_m^{[n]} \mathbf{R}_{m,m}$. Так как $\mathbf{P}_m^{[n]} \approx \mathbf{R}_{m,m}^{-1}$, можно считать, что $\lambda_{m,\max} \approx 1$ для всех m . Таким образом, в данном случае сходимость алгоритма

будет иметь равномерный характер относительно составляющих фильтра различного порядка m независимо от разброса собственных чисел корреляционных матриц $\mathbf{R}_{m,m}$.

Заключение

Рассмотрены алгоритмы адаптивной нелинейной фильтрации для класса полиномиальных фильтров (фильтров Вольтерра) во временной области. Свойство линейности полиномиальных фильтров относительно своих коэффициентов позволяет, с одной стороны, в значительной степени использовать принципы построения алгоритмов адаптации линейных фильтров, а с другой, имеет ряд особенностей, связанных с выбором параметров адаптации для различных нелинейных составляющих фильтра.

Показано, что для нелинейных фильтров диапазон выбора параметра адаптации градиентных алгоритмов сужается по сравнению с линейным случаем и определяется корреляционными моментами высших порядков. Для ускорения скорости сходимости градиентных алгоритмов адаптации для процессов с симметричной плотностью распределения вероятности вместо проведения трудоемкой операции ортогонализации предложен алгоритм, основанный на раздельной адаптации четных и нечетных нелинейных составляющих фильтра. Данный подход рассмотрен на примере нелинейного фильтра третьего порядка, для которого получены оценки допустимых границ параметров адаптации.

С целью выравнивания скорости сходимости отдельных нелинейных составляющих фильтра предложен алгоритм последовательной адаптации, основанный на выполнении итераций в направлении увеличения порядка нелинейности составляющих фильтра. Доказана сходимость такого алгоритма и получены оценки для выбора его параметров. Показано, что наряду с увеличением скорости сходимости, последовательный алгоритм адаптации в то же время не гарантирует сходимость к оптимальной точке. Для достижения необходимой точности без существенной потери скорости адаптации предложена комбинированная схема, основанная на совместном использовании последовательного и параллельного алгоритмов адаптации, используя последний для уточнения текущего вектора коэффициентов фильтра. Другим возможным решением является добавление к последовательной схеме дополнительной обратной связи, периодическое замыкание которой приводит к снижению систематической ошибки адаптации.

Увеличение и выравнивание скорости сходимости процесса адаптации может быть достигнуто за счет использования алгоритмов ньютоновского типа (рекурсивных алгоритмов наименьших квадратов). Предложен рекурсивный алгоритм адаптации полиномиальных

фильтров с экспоненциальным взвешиванием, обеспечивающий, наряду с возрастанием скорости сходимости, сглаживание случайных флуктуаций при приближении к оптимальной точке, а также выравнивание скорости сходимости относительно нелинейных составляющих фильтра.

Литература

1. Адаптивные фильтры / Под ред. К.Ф.Н. Коуэна, П.М. Гранта. М.: Мир, 1988. 392 с.
2. Уидроу Б., Стирнз С. Адаптивная обработка сигналов. М.: Радио и связь, 1989. 440 с.
3. Haykin S. Adaptive filter theory. NJ: Prentice-Hall, 1991. 936 с.
4. Джиган В.И. Адаптивная фильтрация сигналов: теория и алгоритмы. М.: Техносфера, 2013. 528 с.
5. Doyle III F. J., Pearson R. K., Ogunnaike B.A. Identification and control using Volterra models. London: Springer-Verlag, 2002. 318 с.
6. Hansler E., Schmidt G. Acoustic echo and noise control: A practical approach. Hoboken, NJ: Wiley, 2004. 472 с.
7. Меньшиков Б.Н., Приоров А.Л. Нелинейная эхо-компенсация на базе адаптивного полиномиального фильтра Вольтерра с динамически перестраиваемой структурой. Цифровая обработка сигналов. 2006. № 3. С. 20-25.
8. Степанов О.А., Нейросетевые алгоритмы в задаче нелинейного оценивания. Взаимосвязь с байесовским подходом. Навигация и управление движением: материалы докладов XI конференции молодых ученых. 2009. С. 39-65.
9. Mitra S. K., Sicuranza G. L. Nonlinear image processing. Academic Press, 2001. 455 с.
10. Mathews V.J., Sicuranza G.L. Polynomial signal processing. New York: John Wiley & Sons Interscience publication, 2000. 472 с.
11. Щербаков М.А., Стешенко В.Б., Губанов Д.А. Цифровая полиномиальная фильтрация в реальном масштабе времени: алгоритмы и пути реализации на современной элементной базе. Цифровая обработка сигналов. 2000. № 1. С. 19-26.
12. Mathews V. J. Adaptive polynomial filters. IEEE Signal Processing Magazine. 1991. № 7, pp. 10-26.
13. Пупков К.А., Капалин В.И., Ющенко А.С. Функциональные ряды в теории нелинейных систем.. М.: Наука. 1976. 448 с.
14. Shcherbakov M. A. A Recursive Algorithm of Digital Polynomial Filtering. 2020 Moscow Workshop on Electronic and Networking Technologies (MWENT), March 11-13, 2020, Moscow, Russia, 4 p. DOI: 10.1109/MWENT47943.2020.9067458.
15. Щербаков М.А. Построение оптимальных нелинейных фильтров методом последовательных приближений. Аналитическая механика, устойчивость и управление: Труды X Международной Четаевской конференции. Т. 3. Секция 3. Управление. Ч. II. Казань, 12-16 июня 2012 г. Казань: Изд-во Казан. гос. техн. ун-та, 2012. С. 472-482.
16. Ланкастер П. Теория матриц. М.: Наука, 1978. 280 с.
17. Pottmann M. Application of general multi-model approach for identification of highly nonlinear processes – a case study / M. Pottmann, H. Unbehauen, D.E. Seborg. Int. Journal of Control. 1993. V. 57. № 1. P. 97-120.
18. Мармарелис В. Мармарелис М. Анализ физиологических систем. Метод белого шума. М.: Мир. 1981. 480 с.